

Leandro Quingerski

**KE-IoT: UMA PROPOSTA DE MODELO BASEADO EM
CONHECIMENTO PARA AMBIENTES DE INTERNET DAS
COISAS (IoT)**

Dissertação submetida ao Programa de
Pós-Graduação em Engenharia e
Gestão do Conhecimento da
Universidade Federal de Santa
Catarina para a obtenção do Grau de
Mestre em Engenharia e Gestão do
Conhecimento.
Orientador: Prof. Mario Antonio
Ribeiro Dantas, Dr.
Co-orientador: Prof.^a Patricia de Sá
Freire, Dr.^a

Florianópolis
2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Quingerski, Leandro

KE-IoT: UMA PROPOSTA DE MODELO BASEADO EM
CONHECIMENTO PARA AMBIENTES DE INTERNET DAS COISAS
(IoT) / Leandro Quingerski ; orientador, Mario
Antonio Ribeiro Dantas, coorientadora, Patricia de
Sá Freire, 2019.
172 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Centro Tecnológico, Programa de Pós
Graduação em Engenharia e Gestão do Conhecimento,
Florianópolis, 2019.

Inclui referências.

1. Engenharia e Gestão do Conhecimento. 2.
Internet das Coisas. 3. Ontologias. 4. Web
Semântica. 5. Sistemas de Conhecimento. I. Ribeiro
Dantas, Mario Antonio. II. Freire, Patricia de Sá.
III. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Engenharia e Gestão do
Conhecimento. IV. Título.

Leandro Quingerski

**KE-IoT: UMA PROPOSTA DE MODELO BASEADO EM
CONHECIMENTO PARA AMBIENTES DE INTERNET DAS
COISAS (IoT)**

Esta dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia e Gestão do Conhecimento e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina.

Florianópolis, 15 de março de 2019.

Prof.^a Gertrudes Aparecida Dandolini, Dr.^a
Coordenadora do Curso

Banca examinadora:

Prof.^a Patricia de Sá Freire, Dr.^a
Coorientadora

Prof. José Leomar Todesco, Dr.
Universidade Federal de Santa Catarina

Prof. Fernando Alvaro Ostuni Gauthier, Dr.
Universidade Federal de Santa Catarina

Prof. Alex Sandro Roschildt Pinto, Dr.
Universidade Federal de Santa Catarina

*“Nenhum problema pode ser resolvido
a partir do mesmo nível de consciência
que o criou”*

(Albert Einstein).

AGRADECIMENTOS

Agradeço primeiramente a meus pais, Sr. Arnaldo e Sra. Josemeri, por sempre acreditarem em mim, me apoiarem na busca de meus sonhos e pelo exemplo de pessoas que são para mim.

A todos meus familiares, amigos e colegas de trabalho, grato pela compreensão enquanto estive focado para alcançar mais este objetivo e principalmente por toda a motivação e confiança compartilhada comigo, isso foi muito importante nesta jornada.

Ao orientador Prof. Dr. Mario A. R. Dantas e co-orientadora Prof.^a Dr.^a Patricia de Sá Freire, me sinto honrado pela amizade e muito agradecido por me instruírem e apoiarem neste processo, passando conhecimentos que vão além da elaboração deste trabalho, mas que também levo para a vida.

A todo corpo docente do EGC, meus parabéns por formarem este time de altíssimo nível e meu eterno agradecimento por todo o conhecimento e experiência compartilhada, vocês são os verdadeiros “ativos de conhecimento” do EGC. E aos meus colegas do programa, desejo sucesso em seus desafios e que levem a sabedoria adquirida e o “orgulho de pertencer” sempre a diante.

Por fim, meus agradecimentos a Universidade Federal de Santa Catarina (UFSC) pelo espaço e oportunidade concedida na obtenção deste título.

RESUMO

A expansão da quantidade e variedade de dispositivos conectados à internet já ultrapassou o número de habitantes do planeta e continua a crescer em grande escala, diante deste cenário surge a área da Internet das Coisas e com isso uma série de novas aplicações e inovações começam a emergir, em âmbito social, organizacional e governamental, tendo aplicações em diversas áreas, como: saúde, transporte, agricultura, energia, indústria, entre outras. Diante desta revolução tecnológica gerada pela IoT, uma série de desafios necessitam ser superados, entre eles, a garantia de interoperabilidade é um dos pontos principais, pois com o imenso volume de dados gerados, e dos diversos protocolos, redes de comunicação e formatos utilizados é necessário que a interoperabilidade seja trabalhada para permitir que a IoT aconteça. O uso de ontologias e tecnologias da web semântica permite gerar a interoperabilidade necessária para atender a esta demanda latente da IoT e tem tido papel chave nas iniciativas dadas neste sentido, além disso, também possibilita criar maior significado aos dados brutos gerados pelos dispositivos conectados, o que permite a geração de novas informações e a descoberta de conhecimentos a partir da implantação de uma base de conhecimento deste contexto. Neste sentido, este estudo propõe um modelo que aborda o cenário da IoT com uma ótica da Engenharia do Conhecimento (EC), propondo um modelo baseado em conhecimento para ambientes de IoT apoiado principalmente no uso de ontologias e web semântica, porém, expandindo para possibilidades de utilização de outras técnicas da EC, como métodos de resolução de problemas, aprendizagem de máquina, KDD, entre outros. A partir do desenvolvimento de um ambiente simulado de IoT, foi possível validar conceitos, fluxos e alguns dos principais requisitos que compõem o modelo proposto, sendo gerada uma base de conhecimento com os dados de dispositivos conectados (agente simulado). Durante o processo de desenvolvimento do modelo proposto e do ambiente simulado, foi possível coletar resultados experimentais que permitem a evolução dos mesmos e a expansão para diversas áreas de aplicação, pois o modelo foi elaborado com uma abordagem generalizada, o que permite realizar especializações de acordo com as aplicações a serem implementadas.

Palavras-chave: Engenharia do Conhecimento. Sistemas de Conhecimento. Ontologias. Web semântica. Internet das Coisas. IoT.

ABSTRACT

The expansion of the quantity and variety of devices connected to the Internet has already surpassed the number of inhabitants of the planet and continues to grow on a large scale, before this scenario emerges the Internet of Things area and with this a series of new applications and innovations begin to emerge, in a social, organizational and governmental context, having applications in several areas, such as: health, transportation, agriculture, energy, industry, among others. Faced with this technological revolution generated by IoT, a series of challenges need to be overcome, among them, the guarantee of interoperability is one of the main points, because with the huge volume of data generated, and the various protocols, communication networks and formats used Interoperability is required to enable IoT to take place. The use of ontologies and semantic web technologies allows the generation of interoperability needed to meet this latent demand of IoT and has played a key role in the initiatives taken in this regard, and also allows to create greater meaning to the raw data generated by the connected devices, the which allows the generation of new information and the discovery of knowledge from the implementation of a knowledge base of this context. In this sense, this study proposes a model that approaches the IoT scenario with a view of Knowledge Engineering (EC), proposing a knowledge-based model for IoT environments supported mainly in the use of ontologies and semantic web, however, expanding to possibilities of other EC techniques, such as problem solving methods, machine learning, KDD, among others. From the development of a simulated IoT environment, it was possible to validate concepts, flows and some of the main requirements that make up the proposed model, generating a knowledge base with the data of connected devices (simulated agent). During the process of development of the proposed model and the simulated environment, it was possible to collect experimental results that allow the evolution of the same and the expansion for several application areas, since the model was elaborated with a generalized approach, which allows to realize specializations according to with the applications to be implemented.

Keywords: Knowledge Engineering. Knowledge-Based Systems. Ontologies. Semantic Web. Internet of Things. IoT.

LISTA DE FIGURAS

Figura 1 – Técnicas de EC.....	32
Figura 2 – Evolução dos conceitos de SC.....	34
Figura 3 – Arquitetura genérica de um SC.....	35
Figura 4 – Modelo da arquitetura da web semântica.....	41
Figura 5 – Camadas para construção do conhecimento na web semântica	42
Figura 6 – Conectividades da IoT.....	48
Figura 7 – Conceito de Internet of Everything.....	49
Figura 8 – Projeção do total de dispositivos conectados e da população mundial entre os anos de 2003 a 2020.....	52
Figura 9 – Total de dispositivos conectados separados por tipo entre os anos de 2004 a 2018.....	53
Figura 10 – Transformação de dados em sabedoria relacionados aos benefícios à humanidade.....	54
Figura 11 – Integração do conhecimento no contexto da IoT.....	54
Figura 12 – Tecnologias associadas à IoT.....	56
Figura 13 – Visão de mundo inteligente no contexto da IoT.....	63
Figura 14 – Modelo de arquitetura proposta pela ITU-T.....	68
Figura 15 – Ontologia SSN para anotação semântica de observações de sensores.....	75
Figura 16 – Framework conceitual SEG 3.0.....	78
Figura 17 – Arquitetura de SWoT baseada na metodologia SEG 3.0.....	80
Figura 18 – Tecnologias atuais da web semântica para IoT.....	81
Figura 19 – Tecnologias da web mais atuais no contexto da WoT.....	82
Figura 20 – Arquitetura SOA aplicada como middleware semântico.....	83
Figura 21 – Etapas DSRM.....	90
Figura 22 – Detalhamento da metodologia de pesquisa.....	92
Figura 23 – Arquitetura do modelo proposto.....	103
Figura 24 – Diagrama de domínio do modelo KE-IoT.....	113
Figura 25 – Diagrama de sequência de Registro de Dispositivo IoT.....	114
Figura 26 – Diagrama de sequência para autenticação de Dispositivos IoT e Usuários da aplicação.....	115
Figura 27 – Diagrama de sequência do envio de dados do Dispositivo IoT.....	115
Figura 28 – Diagrama de sequência de solicitações de demanda de usuários (genérico).....	116

Figura 29 – Cenário tecnológico implementado (ambiente de simulação)	120
Figura 30 – Estrutura JSON para envio ao serviço web de registro	124
Figura 31 – Exemplo de JSON recebido na resposta sucesso na autenticação	125
Figura 32 – Exemplo de envio do token no cabeçalho da requisição HTTP	125
Figura 33 – Estrutura JSON de envio ao serviço web de inserção de dados	126
Figura 34 – SPARQL de consulta à dispositivos na base de conhecimento	128
Figura 35 – Inicialização do servidor HTTP	128
Figura 36 – Requisição ao serviço web de registro	129
Figura 37 – Requisição ao serviço web de autenticação	129
Figura 38 – Requisição ao serviço web de envio de dados (simulação do Dispositivo IoT)	130
Figura 39 – SPARQL de inserção de dados de observação de sensores (simulados)	131
Figura 40 – Requisição ao serviço de consulta de dados de um dispositivo IoT	131
Figura 41 – Demonstração da consulta de dados do dispositivo IoT via navegador de internet	132
Figura 42 – Aderência do modelo proposto x ambiente de simulação	133
Figura 43 – Relação dos fluxos propostos x serviços web disponibilizados	134

LISTA DE QUADROS

Quadro 1 – Referências do EGC relacionadas ao tema do trabalho.....	29
Quadro 2 – Definições de IoT.....	46
Quadro 3 – Aplicações da IoT.....	61
Quadro 4 – Arquiteturas e especificações para IoT.....	66
Quadro 5 – Detalhamento dos serviços web disponibilizados.....	123

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CoAP	Constrained Application Protocol
EC	Engenharia do Conhecimento
EGC	Engenharia e Gestão do Conhecimento
EO	Engenharia de Ontologias
EPC	Electronic Product Codification
ES	Engenharia de Software
ETSI	European Telecommunications Standards Institute
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IEEE	Institute of Electrical and Electronics Engineers
IERC	European Research Cluster on the Internet of Things
IoE	Internet of Everything
IoT	Internet of Things
IP	Internet Protocol
ITU	International Telecommunication Union
ITU-T	Telecommunication Standardization Sector of ITU
JSON	JavaScript Object Notation
M2M	Machine-to-Machine
MQTT	Message Queuing Telemetry Transport
NFC	Near Field Communication
OGC	Open Geospatial Consortium
OWL	Ontology Web Language
P2M	People-to-Machine
P2P	People-to-People
PPEGC	Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento
RDF	Resource Description Framework
RDFs	Resource Description Framework Schema
REST	Representational State Transfer
RFID	Radio-Frequency Identification
SC	Sistemas de Conhecimento
SOA	Service-Oriented Architecture
SOSA	Sensor, Observation, Sample and Actuator
SPARQL	SPARQL Protocol and RDF Query Language
SSN	Semantic Sensor Network
SWoT	Semantic Web of Things

TCP	Transmission Control Protocol
TGS	Teoria Geral de Sistema
UFSC	Universidade Federal de Santa Catarina
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language
WoT	Web of Things

SUMÁRIO

1	INTRODUÇÃO	23
1.1	CONTEXTUALIZAÇÃO DO PROBLEMA.....	25
1.2	OBJETIVOS	25
1.2.1	Objetivo Geral	25
1.2.2	Objetivos Específicos.....	26
1.3	JUSTIFICATIVA.....	26
1.4	DELIMITAÇÃO E LIMITAÇÃO DO TRABALHO	27
1.5	ADERÊNCIA DO TEMA AO EGC	28
1.6	ESTRUTURA DO TRABALHO	29
2	FUNDAMENTAÇÃO TEÓRICA.....	31
2.1	ENGENHARIA DO CONHECIMENTO	31
2.1.1	Sistemas de Conhecimento	33
2.1.2	Ontologias	36
2.1.3	Web semântica.....	38
2.2	INTERNET DAS COISAS (IoT).....	45
2.2.1	Conceitos, definições e temas relacionados	45
2.2.2	História e evolução	51
2.2.3	Tecnologias.....	55
2.2.4	Áreas de aplicação e suas convergências	59
2.2.5	Arquiteturas e modelos de referência.....	63
2.2.6	Requisitos e modelos conceituais.....	67
2.3	ONTOLOGIAS E WEB SEMÂNTICA PARA IoT	72
2.3.1	Modelos de ontologias para IoT	72
2.3.2	Web semântica aplicada na IoT	76
3	METODOLOGIA	87
3.1	CARACTERIZAÇÃO DA PESQUISA.....	87
3.2	PROCEDIMENTOS METODOLÓGICOS	88
3.3	ETAPAS E ATIVIDADES DA PESQUISA	91
3.3.1	Etapa 1 – Identificar o problema e sua motivação	92
3.3.2	Etapa 2 – Definir os objetivos da solução	94
3.3.3	Etapa 3 – Projetar e desenvolver	95

3.3.4	Etapa 4 – Demonstrar	97
3.3.5	Etapa 5 – Avaliar	98
3.3.6	Etapa 6 – Comunicar	100
3.4	CONSIDERAÇÕES DO CAPÍTULO	100
4	KE-IoT: UMA PROPOSTA DE MODELO BASEADO EM CONHECIMENTO PARA AMBIENTES DE INTERNET DAS COISAS (IoT)	101
4.1	CAMADAS	103
4.1.1	Dispositivo IoT	103
4.1.2	Segurança e Confiabilidade	105
4.1.3	Gerenciamento	105
4.1.4	Redes e protocolos de comunicação	105
4.1.5	Serviços de suporte	106
4.1.6	Apresentação	107
4.1.7	Conhecimento	108
4.2	REQUISITOS GERAIS	109
4.3	MODELO DE DOMÍNIO	112
4.4	FLUXOS BÁSICOS DE COMUNICAÇÃO	113
4.5	APLICAÇÃO DA WEB SEMÂNTICA E ONTOLOGIAS	116
4.6	CONSIDERAÇÕES DO CAPÍTULO	118
5	AMBIENTE SIMULADO E RESULTADOS EXPERIMENTAIS	119
5.1	TECNOLOGIAS E INFRAESTRUTURA UTILIZADA ...	120
5.2	SIMULAÇÃO DO DISPOSITIVO IOT	121
5.3	SERVIÇOS WEB	122
5.4	BASE DE CONHECIMENTO	126
5.5	TESTES E RESULTADOS	128
5.6	ADERÊNCIA AO MODELO PROPOSTO	132
6	CONCLUSÃO E TRABALHOS FUTUROS	135
	REFERÊNCIAS	139
	APÊNDICE A – Inicializador API	151
	APÊNDICE B – Serviço web de registro	153
	APÊNDICE C – Serviço web de autenticação	157
	APÊNDICE D – Serviços web de envio e de obtenção de dados	159

ANEXO A – Lista de requisitos comuns para IoT 163

1 INTRODUÇÃO

É estimado que atualmente existam cerca de 50 bilhões de dispositivos conectados à internet, tendo em média, seis por habitante do planeta (EVANS, 2011). E a previsão é de que nos próximos seis anos aproximadamente 100 bilhões de dispositivos estejam conectados (ROSE; ELDRIDGE; CHAPIN, 2015).

Diante desta revolução tecnológica nasce a área da IoT, possibilitando que objetos físicos também possuam uma existência virtual, permitindo que estes se conectem com pessoas ou até a outros objetos, a qualquer momento, em qualquer lugar, através de qualquer caminho, rede ou serviço (SUNDMAEKER et al., 2010; EVANS, 2011).

Muitas aplicações relacionadas a IoT estão sendo implementadas, gerando soluções em diversos domínios e ambientes, por exemplo, é possível melhorar a qualidade de vida dos seres humanos, nas casas, durante viagens, em situações de doenças, nos locais de trabalho, ao realizarem exercícios (ATZORI et al. 2010). Atuando também no contexto de cidades e ambientes inteligentes, veículos autônomos e conectados em rede, indústrias, logística, automação industrial, entre outras (FRIESS, 2013; MANYIKA, 2015). As aplicações são variadas e impactam em âmbito social, organizacional e governamental, abrangendo várias áreas como saúde, transporte, agricultura, energias, indústria e turismo (ATZORI et al. 2010; FRIESS, 2013; MANYIKA, 2015).

Devido a diversidade de dispositivos e dados que estão sendo gerados na IoT, um dos desafios principais a serem superados neste cenário é promover a interoperabilidade entre estas comunicações (VERMESAN et al. 2013; SERRANO et al., 2015; SZILAGYI, WIRA, 2016).

Neste sentido, ontologias e tecnologias da web semântica tem sido utilizadas na solução do problema de interoperabilidade em diversas áreas e aplicá-las na resolução desta demanda da IoT tem se demonstrado uma boa solução (TOMA; SIMPERL; HENCH, 2009; HACHEM; TEIXEIRA; ISSARNY, 2011; BARNAGHI et al., 2012; VERMESAN et al. 2013; SERRANO et al., 2015; SZILAGYI; WIRA, 2016).

Além disso, como uso de anotações semânticas e de ontologias nos dados brutos advindos da IoT é possível representar e expandir conhecimentos e conceitos envolvidos, gerando uma base de conhecimento do contexto de aplicação. E desta maneira, desenvolver soluções mais genéricas e adaptáveis, inferir novos conhecimentos e aplicar métodos de resolução de problemas e de aprendizado de máquina, por exemplo. Contudo, torna-se um ambiente adequado para que a

Engenharia do Conhecimento (EC) seja trabalhada no contexto da IoT. (BECHHOFFER, et al. 2002; FELICÍSSIMO, et al; 2003; SASIETA, BEPPLER, PACHECO, 2011; NAZÁRIO; DANTAS; TODESCO, 2014).

Através de buscas em bases científicas foram encontradas iniciativas relacionadas a padronização de modelos para construção de soluções para IoT. E foram identificadas técnicas e métodos para a modelagem e desenvolvimento de Sistemas de Conhecimento (SC) baseados em ontologias e tecnologias da web semântica (SASIETA, BEPPLER, PACHECO, 2011; NAZÁRIO; DANTAS; TODESCO, 2014). A partir do material estudado foi possível a proposição do modelo apresentado neste trabalho.

O modelo KE-IoT utiliza uma abordagem baseada no uso de ontologias e web semântica aplicadas em ambientes de IoT, o qual especifica um modelo genérico e os principais aspectos de um arquitetura IoT, considerando também um processo de anotação semântica e publicação de dados em uma base de conhecimento. O modelo detalha camadas conceituais, diagrama de domínio, listagem de requisitos, fluxos de comunicação e etapas para o desenvolvimento da anotação semântica de dados, com o objetivo de contemplar os principais assuntos a serem considerados na implementação de SC para IoT.

A proposição do modelo visa criar uma base de conhecimento em grafos, no formato de triplas RDF, considerando a aplicação de anotações semânticas com base na ontologia SOSA/SSN, em dados gerados no contexto da IoT, como sensores, atuadores e tags de identificação. Também é abordado um processo para expansão e vinculação com ontologias externas e de outros contextos, no intuito de promover um ambiente que possua interoperabilidade semântica e que favoreça a aplicação de métodos e técnicas da EC para descoberta de novos conhecimentos, por exemplo.

A partir do modelo proposto foi desenvolvido um ambiente simulado onde foi possível coletar resultados experimentais que foram analisados e relatados neste estudo, gerando considerações importantes para a evolução e expansão do modelo e do próprio ambiente simulado.

Contudo, este estudo aproxima as áreas de pesquisa da IoT e EC, e apresenta uma ótica favorável para que possa ser trabalhada a Engenharia e Gestão do Conhecimento (EGC) no contexto de soluções para IoT.

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA

A IoT nasceu com o objetivo de integrar o mundo físico com o mundo virtual usando a Internet como o meio para se comunicar e trocar informações. E para fornecer uma comunicação e interação contínua entre os objetos do mundo real, a qualquer momento e em qualquer lugar no futuro, é preciso resolver os problemas complexos de interoperabilidade atuais (KATAZONOV et al., 2008; BARNAGHI et al., 2012; VERMESAN et al., 2013; SERRANO et al., 2015; SZILAGYI, WIRA, 2016).

Nas práticas da Engenharia do Conhecimento são tratados pontos de aquisição, integração, recuperação e descoberta de conhecimentos, através de dados e informações (SASIETA, BEPLER, PACHECO, 2011; NAZÁRIO; DANTAS; TODESCO, 2014). Entre as técnicas existentes estão as ontologias que possibilitam atribuir maior significado aos dados e informações de um domínio, permitindo inferir novos conhecimentos e aplicar métodos de resolução de problemas (GÓMEZ-PÉREZ et al, 2004; NAZÁRIO; DANTAS; TODESCO, 2014).

As ontologias também permitem a modelagem semântica necessária para resolver problemas de interoperabilidade, e tem sido fundamentais no papel de resolver desafios impostos por estes ambientes diversificados da IoT, e na efetivação da própria web semântica (GÓMEZ-PÉREZ et al, 2004; WANG et al, 2012; NAZÁRIO; DANTAS; TODESCO, 2014).

Diante dos variados modelos, protocolos, técnicas, linguagens, ferramentas e tecnologias encontradas, nestas áreas, a questão de pesquisa se definiu em: **“Quais aspectos e tecnologias devem ser consideradas na implementação de Sistemas de Conhecimento para ambientes de IoT?”**.

1.2 OBJETIVOS

Neste tópico são apresentados os objetivos desta pesquisa, sendo separados respectivamente, em geral e específicos.

1.2.1 Objetivo Geral

Propor um modelo baseado em conhecimento para ambientes de Internet das Coisas.

1.2.2 Objetivos Específicos

Para atender ao objetivo geral foram estabelecidos os objetivos específicos da pesquisa, detalhados a seguir:

- Especificação detalhada considerando artefatos e diagramas para representação dos conhecimentos envolvidos nestes ambientes.
- Ambiente de simulação de um cenário da IoT de acordo com especificações do modelo proposto para realização de testes.
- Base de conhecimento no formato de grafos com dados do contexto da IoT com representação semântica.

Considera-se que ao alcançar todos os objetivos específicos estabelecidos, o objetivo geral estará sendo atendido.

1.3 JUSTIFICATIVA

As projeções futuras para IoT até o ano de 2025 é que existam cerca de 100 bilhões de dispositivos conectados e um impacto de mais de 11 trilhões de dólares na economia global (ROSE; ELDRIDGE; CHAPIN, 2015). Com as perspectivas futuras quanto a expansão de dispositivos conectados, surgem diversos desafios a serem superados para que este ambiente se torne real, são questões que estão gerando uma série de pesquisas no campo da Internet das Coisas. Sendo assim, a IoT se mostra como uma área emergente que não somente requer desenvolvimento de infraestruturas e tecnologias, mas também de novos serviços capazes de suportar aplicações múltiplas, escaláveis e interoperáveis (BALDINI et al., 2015; SERRANO et al., 2015).

A modelagem semântica tornou-se fundamental para resolver o problema da interoperabilidade para a Internet das Coisas, dada a natureza distribuída e heterogênea das “Coisas” (WANG, Wei et al, 2012). E isto é possível através de práticas da EC, onde através da aplicação de ontologias e tecnologias e linguagens da web semântica, se torna possível a geração de uma base de conhecimento a partir destes dados, permitindo que técnicas e métodos aplicados na construção de SC possam ser utilizados neste contexto para agregar maior inteligência às soluções de IoT (GÓMEZ-PÉREZ et al, 2004; WANG, Wei et al, 2012; NAZÁRIO; DANTAS; TODESCO, 2014).

Sistemas de Conhecimento têm sido utilizados em aplicações envolvendo Internet das coisas, extração de conhecimento, simuladores

de processos, processos de aprendizagem, avaliação de sistemas de transportes, avaliação de risco, entre outras aplicações. (MARTÍNEZ, 2017 apud MENDES et al., 2016; RIBEIRO et al., 2016; RAUTA et al., 2016; WERNKE et al., 2017; PARK et al., 2017; YANG, YANG e COHEN, 2017; KARAPANTELAKIS et al., 2017; CHEN et al., 2017).

As ferramentas disponíveis de IoT, em termos de suporte ao desenvolvimento, opções de plataforma e opções de conectividade, são muitas. No entanto, os projetistas de sistemas precisam evoluir para enfrentar os desafios impostos pela visão da IoT (CHUAH, 2014). E desta maneira é compreendido que se técnicas da EC, como ontologias e tecnologias da web semântica, tem tido papel chave na tentativa de superar o desafio de interoperabilidade causado pela IoT, a área da Engenharia do Conhecimento e seus especialistas também precisam evoluir no sentido de atender às necessidades impostas por este novo cenário.

Portanto, neste estudo é proposto um modelo para ambientes de IoT que agrega uma ótica do Conhecimento aos aspectos principais inerentes em soluções da IoT, visando facilitar que a EGC possa acontecer nestes cenários.

1.4 DELIMITAÇÃO E LIMITAÇÃO DO TRABALHO

Esta pesquisa não pretende realizar uma revisão sistemática da literatura sobre os termos envolvidos, nem gerar novas teorias sobre o tema, mas conceituar uma proposta e aplicá-la de forma prática, utilizando tecnologias existentes na atualidade para resolução de problemas reais e com isso coletar os resultados obtidos para análises e conclusões.

No escopo deste trabalho, o modelo proposto foi construído através da definição de uma abordagem generalizada do domínio da IoT, não tratando assuntos de aplicações para áreas específicas nem a proposição de novas ontologias, métodos ou metodologias, sendo priorizada a utilização de práticas já conhecidas neste contexto.

Quanto a implementação do ambiente simulado, esta é restrita ao desenvolvimento a nível de software, e não envolve implementações de sistemas embarcados nem o uso de equipamentos físicos. Além disso, dado a amplitude da área de IoT, o ambiente não contempla todos os requisitos propostos no modelo, tendo foco principal na implementação

de requisitos de segurança e no processo de anotação semântica para a geração da base de conhecimento, e a execução de testes.

1.5 ADERÊNCIA DO TEMA AO EGC

Este trabalho se enquadra na linha de pesquisa Teoria e Prática em Engenharia do Conhecimento, a qual aborda metodologias e tecnologias da EC e da Inteligência Computacional e suas relações com a gestão e com a mídia do conhecimento (EGC, 2018).

A EC objetiva buscar por ferramental computacional, apoiar a processos de aquisição e de representação do conhecimento, e permitir que esses processos possam ser geridos e utilizados pelas três áreas do programa: engenharia, gestão e mídia do conhecimento (NAZÁRIO; DANTAS; TODESCO, 2014; CECI, 2015). De forma mais concreta, a EC, com base em suas metodologias, técnicas e ferramentas, apoia e fornece suporte aos processos e instrumentos da GC (NAZÁRIO; DANTAS; TODESCO, 2014).

De acordo com Studer et al. (1998), com o surgimento da gestão do conhecimento, a engenharia do conhecimento baseou-se no paradigma de modelagem, em que a representação, a utilização e a reutilização, assim como o compartilhamento de métodos e modelos formais, passaram a ser requisitos essenciais dos Sistemas de Conhecimento.

Contudo, a EGC está diretamente relacionada com os assuntos tratados nesta dissertação, pois visa permitir a integração, reuso e compartilhamento de conhecimentos, a partir da utilização de ontologias e de tecnologias da web semântica aplicadas em dados oriundos de ambientes de IoT, formando uma base de conhecimento a partir dos mesmos.

Neste contexto, entre os estudos do PPGEGC/UFSC consultados, diversos abordam assuntos como ontologias, SC e web semântica, bem como, a proposição ou aplicação de modelos e metodologias que circundam o tema de EC. Vale ressaltar que também possuem alguns estudos que tratam do tema Cidades Inteligentes (Smart Cities) e Big Data e que estes também possuem convergência com a área de IoT (FRIESS, 2013; MANYIKA, 2015). E por este motivo, também foram incluídas como referências do EGC relacionadas a esta dissertação, e estão apresentados no Quadro 1.

Dentre as publicações do EGC apresentadas, se destaca o trabalho de Nazário (2015), por tratar de ambientes ubíquos e aplicação de sensores em ambiente simulado, este é o estudo que possui maior aproximação ao tema desta dissertação.

Quadro 1 – Referências do EGC relacionadas ao tema do trabalho

Autor	Título	Publicação	Ano
RIBEIRO JÚNIOR, Divino Ignácio.	Modelo de sistema baseado em conhecimento para apoiar processos de tomada de decisão em ciência e tecnologia	Tese	2011
NAPOLI, Márcio.	Aplicação de ontologias para apoiar operações analíticas sobre fontes estruturadas e não estruturadas	Dissertação	2011
SCHNEIDER, Viviane.	Método de Modelagem do Contexto Estratégico para Sistemas baseados em Conhecimento	Dissertação	2013
BRAGLIA, Israel.	Um Modelo Baseado em Ontologia e Extração de Informação como Suporte ao Design Instrucional na Geração de Mídias do Conhecimento	Tese	2014
BOTELHO, Mauricio.	Aplicação de Ontologias na Organização de Conteúdos para Apoio a Equipes de Desenvolvimento de Software	Dissertação	2015
SILVA, Thales do Nascimento da.	Um Modelo Baseado em Ontologia para Suporte a Tarefa Intensiva em Conhecimento de Recomendação	Dissertação	2015
NAZARIO, Débora Cabral.	CUIDA - Um Modelo de Conhecimento de Qualidade de Contexto Aplicado aos Ambientes Ubíquos Internos em Domicílios Assistidos	Tese	2015
KLEIN, Vinicius Barreto.	Uma Proposta de Modelo Conceitual para Uso de Big Data E Open Data para Smart Cities	Dissertação	2016
WOSZEZENK, Cristiane Raquel.	Modelo Para Descoberta de Conhecimento Baseado em Associação Semântica e Temporal Entre Elementos Textuais	Tese	2016
ANDERLE, Daniel Fernando.	Modelo de Conhecimento para Representação Semântica de Smart Cities com foco nas Pessoas	Tese	2017
MARTINEZ, Diego Jessie.	Sistema Baseado em Conhecimento (SBC) de Apoio à Capacitação Organizacional	Dissertação	2017

Fonte: O autor (2019)

Apesar dos estudos relacionados encontrados na base do EGC, ainda não constam pesquisas que abordam especificamente ambientes de IoT, sendo um dos constructos principais deste trabalho.

1.6 ESTRUTURA DO TRABALHO

Esta dissertação está organizada em seis capítulos, sendo apresentada, a seguir, uma breve descrição de cada um deles:

- No capítulo 1, é apresentada a introdução do trabalho através de uma descrição das principais motivações e justificativas que levaram à execução desta pesquisa, sendo mostrada também, a problemática que foi abordada. Além disso, também apresenta-se os estudos do PPGECC que possuem vínculo com os assuntos tratados neste trabalho.

- O capítulo 2 apresenta a fundamentação teórica dos assuntos tratados neste trabalho, sendo mostrados os principais conceitos envolvidos, bem como os métodos, técnicas e tecnologias utilizadas no desenvolvimento do modelo e ambiente simulado.
- No capítulo 3, a metodologia utilizada para execução desta pesquisa e elaboração do conteúdo é descrita detalhadamente em nível de etapas e atividades.
- O capítulo 4 apresenta a proposição do modelo que foi elaborado a partir da pesquisa bibliográfica e documental realizada, sendo especificado de forma descritiva e detalhada.
- O capítulo 5, mostra de forma detalhada o ambiente simulado que foi desenvolvido para realização de testes e obtenção dos resultados experimentais, que também estão inseridos neste capítulo.
- Por fim, o capítulo 6 demonstra as conclusões diante da pesquisa realizada, verificando os objetivos atribuídos e os resultados obtidos. E também apresenta as possibilidades de trabalhos futuros que podem ser realizados a partir desta pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem por objetivo apresentar os fundamentos imprescindíveis para a realização desta pesquisa, a partir do conteúdo coletado através da revisão bibliográfica e documental executada, o que permitiu o aprofundamento nos conceitos envolvidos, dando a base para a proposição do modelo e a implementação do ambiente de simulação.

Inicialmente são apresentados os tópicos sobre Engenharia do Conhecimento, abordando assuntos como Sistemas de Conhecimento, Ontologias e Web semântica, mostrando os métodos e tecnologias essenciais para as realização desta pesquisa.

Posteriormente, apresenta-se a fundamentação do tema Internet das Coisas, abordando diferentes aspectos do conceito bem como tecnologias, arquiteturas e áreas de abrangência da IoT. Neste tópico também é apresentada a IoT sob uma lente do Conhecimento, e por fim, são apresentadas ontologias e tecnologias da web semântica relacionadas ao cenário da IoT.

2.1 ENGENHARIA DO CONHECIMENTO

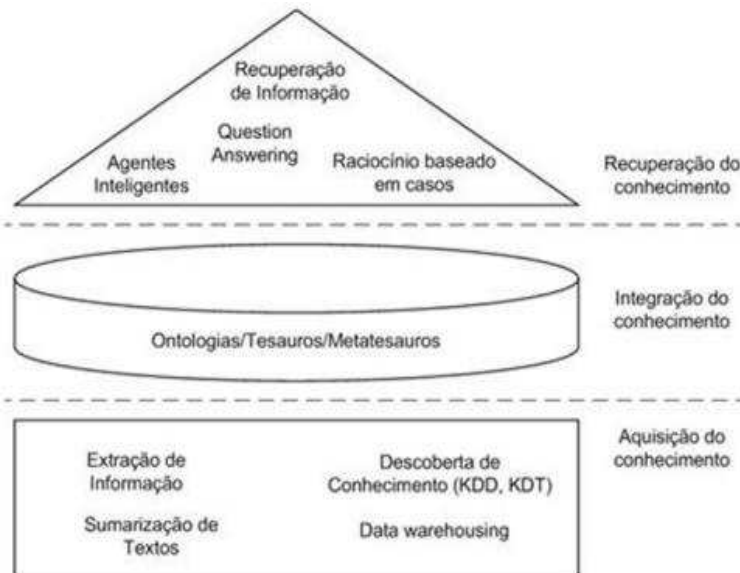
A Engenharia do Conhecimento (EC) surge através dos primeiros trabalhos na área de Inteligência Artificial (IA), no contexto de sistemas especialistas, ainda nos anos setenta. E com a popularidade dos sistemas baseados em conhecimento, tornou-se necessário uma abordagem sistemática para a construção destes sistemas, semelhante às metodologias da Engenharia de Software (SCHREIBER, 2008; DEL ÁGUILA; PALMA; TUNEZ, 2014; NAZÁRIO; DANTAS; TODESCO, 2014). Desta maneira a EC evolui como disciplina para o desenvolvimento de teoria, métodos, linguagens, processos, padrões e ferramentas para o desenvolvimento de aplicações intensivas em conhecimento, orientando sobre quando e como aplicar técnicas específicas de apresentação de conhecimento para resolver problemas específicos (STUDER et al.,1998; SCHREIBER, 2008; DEL ÁGUILA; PALMA; TUNEZ, 2014).

Inicialmente, o conceito de EC esteve baseado no paradigma de mineração/extração do conhecimento advindos de especialistas, sendo passados diretamente para os sistemas computacionais. Posteriormente, o conceito mudou para o paradigma de modelagem do conhecimento (DEL ÁGUILA; PALMA; TUNEZ, 2014).

Entre as metodologias de EC, citam-se: CommonKADs, SPEDE, MOKA, Suite PCPACK4, XP.K, RapidOWL, MIKE e VITAL. Estas possuem objetivos que convergem, no sentido de estabelecerem processos e artefatos que apoiam o desenvolvimento de SC (NAZÁRIO; DANTAS; TODESCO, 2014).

As diversas técnicas de EC podem ser separadas em três fases: Aquisição do conhecimento, Integração do conhecimento e Recuperação do conhecimento. Como técnicas de Aquisição do conhecimento, citam-se: Extração de Informação, Sumarização de Textos, Data warehouse e processos de descoberta de conhecimento (KDD e KDT). Quanto a integração de conhecimento, algumas das técnicas que podem apoiar com algumas formas de modelagem e representação do conhecimento, são: vocabulários controlados, taxonomias, tesouros, metatesouros e ontologias. Sobre a fase de Recuperação do conhecimento, citam-se as técnicas de: Recuperação de Informação, Question Answering, Raciocínio Baseado em Casos e Agentes Inteligentes (SASIETA, BEPLER, PACHECO, 2011; NAZÁRIO; DANTAS; TODESCO, 2014). A Figura 1 apresenta técnicas da EC, separadas pelas fases comentadas.

Figura 1 - Técnicas da EC



Fonte: SASIETA, BEPLER, PACHECO (2011, p.10)

No entanto, a EC surgiu para orientar o desenvolvimento de SC, independente do contexto de domínio e de aplicação em que os mesmos serão utilizados. Neste sentido, a EC integra metodologias, técnicas e ferramentas para construção de SC e que dão suporte aos processos da gestão do conhecimento (STUDER et. al., 1998; NAZÁRIO; DANTAS; TODESCO, 2014).

2.1.1 Sistemas de Conhecimento

O desenvolvimento de um Sistema de Conhecimento, também conhecido na literatura como Sistema Baseado em Conhecimento, abrange diversas áreas do conhecimento, dentre as quais se destacam a Teoria Geral de Sistemas (TGS), Engenharia do Conhecimento (EC), Engenharia de Ontologias (EO) e a Engenharia de Software (ES). Sendo que a TGS fornece uma base teórica para o desenvolvimento de sistemas, a EC visa estruturar o conhecimento, de acordo com os parâmetros do sistema de conhecimento, a EO objetiva representar formalmente o conhecimento e a ES fornece atividades para desenvolver a parte tecnológica de um SC (SCHNEIDER, 2013).

Segundo Ribeiro Junior (2010) apud Schreiber, Akkermans et al. (2002, p.14), a evolução dos SC partem de um paradigma de transposição do conhecimento, para o de modelagem e gestão do conhecimento, se transformando em um área interdisciplinar, tendo os últimos avanços dados no sentido de geração de novos métodos de representação do conhecimento como ontologias e web semântica, conforme é apresentado na Figura 2.

O desenvolvimento de sistemas baseados em conhecimento tem como objetivo separar explicitamente o conhecimento de domínio do restante do sistema. Com a codificação de conceitos por um especialista do domínio, forma-se a base de conhecimento que é utilizada para representação dos conhecimentos, bem como a aplicação de métodos de resolução de problemas (STUDER et. al., 1998).

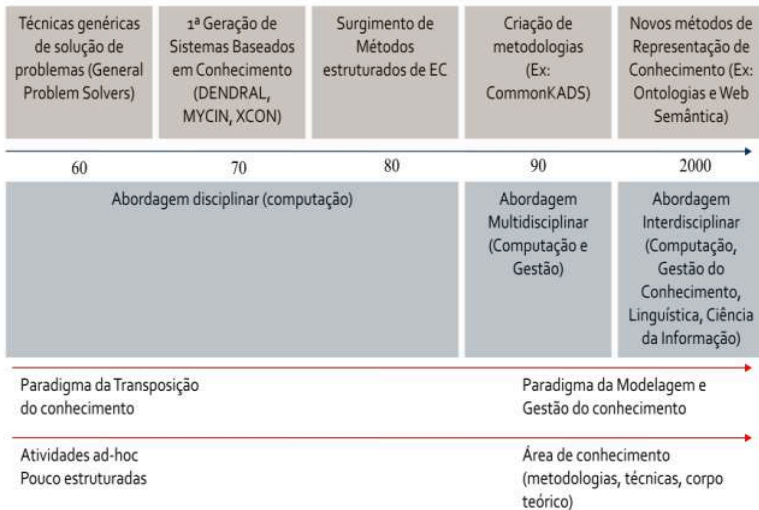
Knublauch (2002) apresenta a visão de uma arquitetura de Sistema de Conhecimento que considera interações entre as camadas de ontologias e de base de conhecimento, as quais interagem entre si e com a camada de métodos de resolução de problemas, fornecendo uma interface de acesso a estes conteúdos e ainda realizando comunicação com

base com dados sobre o contexto de aplicação, conforme é possível visualizar na Figura 3.

Para Brachman e Levesque (2004, p.7) “o que define um SC não é somente o uso de formalismos lógicos, mas a presença de uma Base de Conhecimento, ou seja, uma coleção de estruturas simbólicas representando as crenças e regras durante a operação do sistema”.

Segundo Schneider (2013) apud Nicolini (2006), uma característica dos sistemas baseados em conhecimento está no processo de concepção diferenciado, que é realizado por meio da EC, ao invés dos processos tradicionais de engenharia de software.

Figura 2 - Evolução dos conceitos de SC



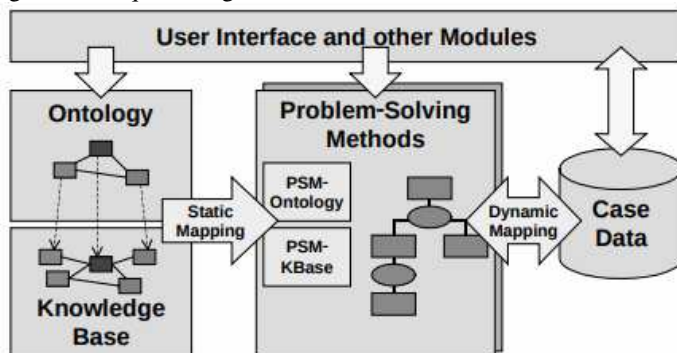
Fonte: Ribeiro Junior (2010) apud Schreiber, Akkermans et al. (2002, p.14)

O processo de construção de um SC pode ser visto como uma atividade de modelagem, considerando o desenvolvimento de um modelo computacional com o objetivo de solucionar problemas de forma comparável a um especialista humano do domínio. Não se trata de criar um modelo cognitivo para simular processos de cognição do especialista, mas de criar um modelo com base no conhecimento existente, que possibilite chegar a resultados similares ao do especialista (STUDER et. al., 1998; NAZÁRIO; DANTAS; TODESCO, 2014).

Ribeiro Junior (2010) apud Schreiber et al. (2002) afirma que o que caracteriza melhor um SC é o fato de possuir processos de modelagem

que usam representações explícitas de conhecimento. E Martínez (2017) apud Ribeiro et al. (2016) ainda comenta que SC utiliza-se dos dados disponíveis como insumo à geração de informação estratégica ou de novos conhecimentos para a tomada de decisões.

Figura 3 – Arquitetura genérica de um SC



Fonte: Knublauch (2002, p.46)

Contudo, sistemas baseados em conhecimento têm sido utilizados em aplicações envolvendo Internet das coisas, extração de conhecimento, simuladores de processos, processos de aprendizagem, avaliação de sistemas de transportes, avaliação de risco, entre outras aplicações. (MARTÍNEZ, 2017, p.24 apud MENDES et al., 2016; RIBEIRO et al., 2016; RAUTA et al., 2016; WERNKE et al., 2017; PARK et al., 2017; YANG, YANG e COHEN, 2017; KARAPANTELAKIS et al., 2017; CHEN et al., 2017).

Os Sistemas de Conhecimento têm se tornado cada vez mais necessários, devido ao aumento constante de dados e informações. A integração e interoperabilidade entre estes sistemas podem ser citados como desafios nesta área. A Engenharia do Conhecimento tem o objetivo de auxiliar o desenvolvimento de Sistemas Baseados em Conhecimento, sendo que a EC utiliza o paradigma de modelagem do conhecimento existente nas pessoas, arquivos e sistemas para um formato computacional (NAZÁRIO; DANTAS; TODESCO, 2014, p. 6).

As ontologias visam representar e formalizar dados em projetos de SC, para que seja possível sua utilização, tanto por sistemas computacionais como por pessoas (SCHNEIDER, 2013). E desta maneira, a utilização de ontologias é um elemento chave na construção de um SC (MARTÍNEZ, 2017).

2.1.2 Ontologias

As ontologias permitem que o conhecimento seja explicitado, formalizando uma visão relevante do mundo (modelo de domínio) e tornando este modelo passível de processamento e interpretação por parte dos computadores (BLOEHDORN et al., 2006).

Segundo Gruber (1993), ontologias são definidas como especificações formais e explícitas, para compartilhamento de conceitos de um domínio específico de conhecimento. Em um contexto de aplicação, a ontologia constitui em uma conformidade para representação de modelos comuns que são compartilhados entre diversas entidades, administrativas ou não (STUDER et al., 1998; LAMHARHAR et al., 2016).

De modo geral, ontologias são esquemas de metadados, onde pode apresentar uma visão abstrata e simplificada de um mundo que se pretende representar para algum propósito (BECHHOFER, 2002). A estrutura da ontologia é baseada na descrição de conceitos e dos relacionamentos semânticos entre eles. Desta maneira, ontologias fornecem uma linguagem que permite o processamento por computadores, e maior eficácia no entendimento entre homem e máquina, bem como a integração de recursos de internet de maneira inteligente, possibilitando a realização de buscas rápidas e aprimoradas, e facilitando a comunicação em um ambiente heterogêneo de recursos e serviços (BECHHOFER, 2002; FELICÍSSIMO, et al; 2003).

As ontologias podem ser representadas, basicamente, de duas formas: graficamente e formalmente. A representação gráfica é utilizada para a compreensão humana, enquanto que a representação formal é usada para que a ontologia seja consumida por máquinas. Para representar ontologias graficamente é possível utilizar grafos, estrutura de árvore, UML, entre outras. Já, para representação formal de ontologias, as linguagens mais populares para descrever ontologias de

maneira formal são RDF/RDF-S e OWL (PEREIRA et. al, 2017, p. 70-71).

Segundo Junior (2010) as ontologias se constituem em uma forma de criar bases de conhecimento utilizáveis por SCs, sendo que uma Base de Conhecimento é um artefato de EC que reúne a representação do conhecimento no domínio de aplicação, de maneira formal, clara, com escopo e aplicação bem definidos, permitindo seu uso em processos de inferência.

Os conceitos de uma ontologia representam a descrição de qualquer coisa, como uma ação, uma função, uma tarefa, um processo ou raciocínio, dentre outros. Estes conceitos se convertem em Classes que geralmente são organizadas em uma taxonomia, onde é possível aplicar mecanismos como herança (MARTÍNEZ, 2017).

Segundo Kiryakov (2006) na estrutura formal de uma ontologia são consideradas as relações de quatro elementos, representados por $O = \{C, R, I, A\}$, onde:

- C - é o conjunto de classes que representam os conceitos em um dado domínio de interesse;
- R - é o conjunto de relações ou associações entre os conceitos do domínio;
- I - é o conjunto de instâncias derivadas das classes, ou ainda, os exemplos de conceitos representados em uma ontologia;
- A - é o conjunto de axiomas do domínio, que servem para modelar restrições e regras inerentes às instâncias.

Segundo Názario, Dantas e Todesco (2015, p. 59) apud Gómes Pérez e Corcho, quanto à expressividade, as ontologias são classificadas pelo tipo de linguagem de representação utilizada e pelos elementos que a constituem, sendo classificadas em:

Ontologias de menor expressividade - são ontologias que modelam informação de um determinado domínio (conceitos e sua taxonomia), sem incluir axiomas e restrições. Este tipo de ontologia não requer um nível de expressividade elevado, o que, por outro lado, dificulta o processo de raciocinar em computadores;

Ontologias de maior expressividade: são ontologias que requerem um alto nível de

expressividade para incorporar axiomas e restrições, facilitando os processos de inferência computacional neste tipo de ontologia.

Uma classificação dos tipos de ontologias é apresentada por Guarino (1998), por se tratar de uma classificação simples, porém consistente e consideravelmente difundida na academia é a considerada nesta dissertação. Segundo o autor, as ontologias podem ser classificadas em 4 tipos:

1. Ontologias Genéricas: tratam de conceitos mais amplos, independente de um problema ou domínio específico;
2. Ontologias de Domínio: envolvem conceitos e vocabulários relativos a domínios particulares;
3. Ontologias de Tarefas: descrevem tarefas ou atividades genéricas, que podem contribuir na resolução de problemas, independente do domínio que ocorrem, por exemplo, processos de vendas ou diagnóstico.
4. Ontologias de Aplicação: abordam conceitos que dependem de um domínio e de uma tarefa específica;

A utilização de ontologias no entanto, permite representar explicitamente a semântica dos dados, onde as ontologias fornecem um entendimento comum e compartilhado de um domínio, que pode ser comunicado através de pessoas e máquinas, possibilitando melhorar a integração e reuso de recursos, tornando-se fator chave para o desenvolvimento da web semântica (NAZÁRIO; DANTAS; TODESCO, 2014).

2.1.3 Web semântica

O conceito da Web Semântica é ideia do criador original da WorldWide Web, Tim Berners-Lee, segundo ele o princípio da Web Semântica é tecer uma Web que não apenas conecte documentos uns aos outros, mas também reconheça o significado das informações contidas nesses documentos (BERNERS-LEE et al., 2001).

Segundo Hendler, Berners-Lee e Miller (2002) a Web Semântica fornece uma infraestrutura que permite não apenas páginas da Web, mas também bancos de dados, serviços, programas, sensores, dispositivos pessoais e até eletrodomésticos a consumir e produzir dados na Web.

Para Nazário, Dantas e Todesco (2015) a web semântica consiste em uma nova Internet que viabiliza a representação e busca de

conhecimento pelos computadores, sendo projetada e construída para ser entendida também por máquinas.

Contudo, o objetivo da web semântica é trazer estrutura para o conteúdo das páginas web, aumentando o seu significado, criando um ambiente onde agentes de software possam navegar página a página e cumprir tarefas para os seus usuários. Portanto, os computadores devem ter acesso a coleções estruturadas de informações e conjuntos de regras de inferência, podendo usá-las para conduzir o raciocínio automatizado (BERNERS-LEE et al., 2001; NAZÁRIO; DANTAS; TODESCO, 2015).

Segundo Van Harmelen (2004), são dois os motivadores principais para criação da Web Semântica, descritos a seguir:

- O primeiro é a integração de dados, que é um gargalo significativo em muitos aplicativos de TI, pois as soluções atuais para este problema são principalmente ad-hoc: cada vez, um mapeamento específico é feito entre os modelos (esquemas) das fontes de dados envolvidas; se a semântica das fontes de dados fosse descrita de uma maneira interpretável pela máquina, os mapeamentos poderiam ser construídos pelo menos de maneira semiautomática (BERNERS-LEE, 2001; HENDLER, BERNERS-LEE e MILLER, 2002; VAN HARMELEN, 2004).
- O segundo motivador é o suporte mais inteligente para usuários finais. Se os programas de computador podem inferir consequências de informações na Web, eles podem oferecer um melhor suporte para encontrar informações, selecionar fontes de informação, personalizar informações, combinar informações de diferentes fontes e assim por diante (BERNERS-LEE, 2001; HENDLER, BERNERS-LEE e MILLER, 2002; VAN HARMELEN, 2004).

Contudo ferramentas presentes na Web Semântica, como Portais Semânticos, Wikis Semânticos, entre outros, são consideradas ferramentas de EC voltadas para web, desta maneira considera-se que a EC está presente na Web Semântica, através da utilização da técnica de ontologia. Sendo que as tecnologias semânticas devem ser exploradas para revelar relações ocultas dentro das informações disponíveis (NAZÁRIO; DANTAS; TODESCO, 2014).

As tecnologias da Web semântica permitem que as pessoas criem armazenamentos de dados na Web e vocabulários, e escrevam regras para

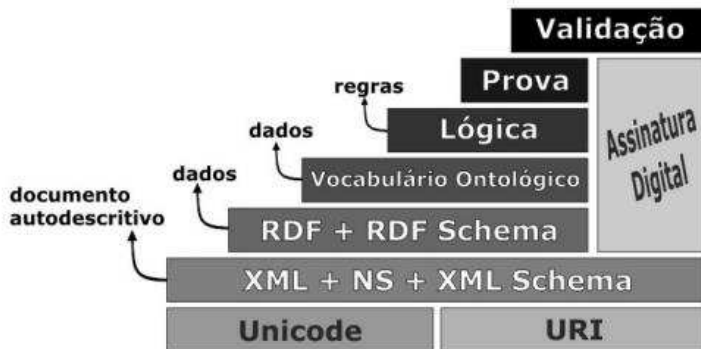
o tratamento de dados. Sendo isso viabilizado através de padrões como o Semantic Web¹ da W3C, que engloba: o modelo de dados RDF, os padrões RDF Schema e OWL e especificação do protocolo e linguagem de consulta SPARQL, por exemplo (JARA et al., 2014). Segundo Jara et al. (2014, p. 8) “se uma solução usa semântica, mas não usa esses padrões, não pode fazer parte do conceito da Web semântica”.

Berners-Lee (2001) propõe uma arquitetura para web semântica, conforme apresentado na Figura 4, sendo este modelo adaptado em 2006. Onde é possível visualizar os tratamentos de cada camada proposta, passando de uma visão sintática para semântica (BERNERS-LEE et al., 2001; BERNERS-LEE et al., 2006; BRAGA; RAMOS; COELHO, 2007).

A primeira camada, mais abaixo, é a base para a Web Semântica onde um conjunto de caracteres Unicode e a URI (Uniform Resources Identifier) é utilizada para referenciar os recursos disponíveis na rede. Na segunda camada, a linguagem de marcação XML (eXtensible Markup Language) associada aos recursos de NS (NameSpaces) e XML Schema, estas servem para descrever documentos que podem ser compreendidos por pessoas e software. A partir deste ponto, todas as camadas seguintes estão estruturadas sobre a camada XML e tiram proveito dela. Assim, a estrutura seguinte (RDF + RDF Schema - Resource Description Framework) é uma camada de metadados que é construída sobre a de XML e que visa a interoperabilidade entre as aplicações, representando os dados sob a forma de triplas através de descrições de recursos, propriedades e valor (BERNERS-LEE et al., 2006; BRAGA; RAMOS; COELHO, 2007).

¹ Disponível em: <<https://www.w3.org/standards/semanticweb/>>. Acesso em: agosto de 2018.

Figura 4 - Modelo da arquitetura da web semântica



Fonte: Berners-Lee (2001)

Quanto a camada de ontologias, ela trata da definição um vocabulário comum (ontológico) para que o conteúdo semântico possa ser compartilhado entre as aplicações. Neste nível, a linguagem padrão OWL (Ontology Web Language) é usada para representar as ontologias e agregar maior conteúdo semântico aos documentos utilizando recursos do RDF e do XML das camadas inferiores (BERNERS-LEE et al., 2006; BRAGA; RAMOS; COELHO, 2007).

As últimas camadas, alicerçadas sobre a estrutura já existente na parte inferior da arquitetura, implementam a definição de regras e possibilitam a prova e a validação de inferências realizadas por agentes de software que fazem uso dos recursos de toda a estrutura existente para a representação do conhecimento (BERNERS-LEE et al., 2006; BRAGA; RAMOS; COELHO, 2007).

Os autores de Braga, Ramos e Coelho (2007) ainda apresentam uma visão funcional da arquitetura da web semântica e mostram três camadas que são a base para a construção do conhecimento, conforme demonstrada na Figura 5. Os procedimentos exigem linguagens, recursos, regras e agentes inteligentes que dão suporte a todo este funcionamento, conforme o detalhamento de Braga, Ramos e Coelho (2007, p. 3) a seguir:

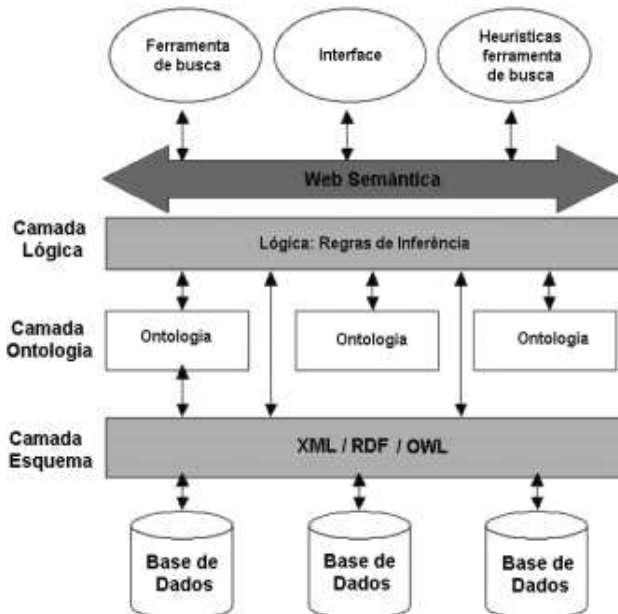
Camada Esquema. É o primeiro passo para definir a Web Semântica, sendo responsável por estruturar os dados e definir seu significado para que possa elaborar raciocínio lógico.

Camada Ontologia. Tem como finalidade definir padrões e as relações entre os dados. Nesse momento, dá-se o entendimento comum e compartilhado de um domínio.

Camada Lógica. A partir dessa camada é possível definir os relacionamentos de informações e as inferências de conhecimento da Web Semântica. Composta por um conjunto de regras de inferência agentes poderão utilizá-la para relacionar e processar informações.

Por uma questão de interoperabilidade é obrigatório que seja aplicada uma semântica bem definida para garantir que tanto o publicador quanto o consumidor destas informações, realmente compartilhem os mesmos significados. Uma contribuição fundamental da Web Semântica, portanto, está em fornecer um conjunto de padrões mundiais que abrem a possibilidade de operar com recursos heterogêneos, fornecendo uma ponte de sintaxe e vocabulários comum entre as partes (UREN, et al., 2006).

Figura 5 – Camadas para construção do conhecimento na web semântica



Fonte: Braga, Ramos e Coelho (2007, p. 3) apud (Afonso, 2001)

O Resource Description Framework (RDF) é a especificação do W3C para projetar a descrição e modelar informações para os recursos da Web, permitindo a interconexão de dados. O modelo de dados RDF consiste em um grafo, que basicamente é composto por uma unidade básica de informação chamada tripla, a qual é formada pela relação sujeito-predicado-objeto. Um sujeito denota um recurso, um predicado denota a relação entre o sujeito e um objeto e o objeto denota o valor do sujeito relacionado (W3C RDF WORKING GROUP, 2014; JARA et al., 2014).

O RDF possui recursos que facilitam a mesclagem de dados, mesmo se os esquemas subjacentes forem diferentes, e também suporta a evolução dos esquemas ao longo do tempo, sem exigir que todos os consumidores de dados sejam alterados. Este se utiliza de URIs para representar sujeitos e predicados, a fim de assegurar a inequívoca informação em toda a Web (W3C RDF WORKING GROUP, 2014).

Uma coleção de instruções RDF representa um grafo que fornece uma representação de contexto de conhecimento, o que pode ser persistente usando bancos de dados relacionais ou armazenado em formato de triplas (JARA et al., 2014; SERRANO et al., 2015; PEREIRA, 2017). Contudo, consiste em uma estrutura de descrição flexível e independente de plataforma que pode ser usada em diferentes estágios da representação de informações, o que torna a implementação consistente e muito mais fácil (SERRANO et al., 2015).

O uso de RDF está aumentando, mas ele é genérico por definição e usando esse modelo simples, ele permite que dados estruturados e semiestruturados sejam misturados, expostos e compartilhados entre diferentes aplicativos. No entanto, novos idiomas baseados em RDF foram desenvolvidos para adicionar recursos específicos do aplicativo como parte da definição da linguagem, como o RDF-S (BRICKLEY; GUHA; MCBRIDE; 2004; SERRANO et al., 2015).

O RDF Schema (RDFS) consiste em uma linguagem de descrição para RDF e é composta de um conjunto de classes com determinadas propriedades, que geram um vocabulário base para linguagens mais expressivas como a OWL (Web Ontology Language) (JARA et al., 2014). O RDFS estende a expressividade do RDF ao permitir descrever grupos de recursos (classes) e suas relações utilizando o conceito de triplas (BRICKLEY; GUHA; MCBRIDE; 2004; PEREIRA; 2017).

A linguagem de ontologia da web (OWL), recomendada pela W3C, permite que significado e semântica sejam melhor expressos se comparada à outras linguagens como: RDF e RDFS, sendo uma linguagem para definir e instanciar ontologias, baseada em lógica descritiva (BECHHOFER et al., 2004; W3C OWL WORKING GROUP et al, 2012; PEREIRA, 2017).

A OWL pode ser usada para representar explicitamente o significado de termos e das relações entre estes, em vocabulários. Sendo estes descrevem propriedades e classes: entre outras relações entre as classes, como: cardinalidade, igualdade, tipificação mais rica de propriedades, características de propriedades e classes enumeradas (BECHHOFER et al., 2004).

Atualmente a OWL é a linguagem mais utilizada para representar ontologias formalmente, apresenta variantes da linguagem que lidam com a escalabilidade e a expressividade das ontologias e permite que aplicações com diferentes propósitos sejam construídas (PEREIRA, 2017). Portanto, foi utilizada na representação das ontologias aplicadas nesta pesquisa.

Quanto a linguagem SPARQL pode ser usada para expressar consultas em diversas fontes de dados, sejam os dados armazenados nativamente como RDF ou visualizados como RDF por meio de middleware (W3C SPARQL WORKING GROUP et al., 2013). Contém recursos para consultar padrões de grafos obrigatórios e opcionais, juntamente com suas conjunções e disjunções. O SPARQL também suporta teste de valor extensível e restringe consultas pelo grafo RDF de origem. Os resultados das consultas SPARQL podem ser conjuntos de resultados ou de grafos RDF (W3C SPARQL WORKING GROUP et al., 2013).

Contudo é necessário que um servidor de dados com suporte ao armazenamento destas triplas RDF seja utilizado para que se torne possível o armazenamento e recuperação destes dados. Neste sentido, o OpenLink Virtuoso Server² é um Servidor de Dados multipropósito e multiprotocolo (Híbrido) que atende a este requisito, entre outras funcionalidades conforme detalhado a seguir por Martinez (2017, p. 49):

² Disponível em: <<https://virtuoso.openlinksw.com/>>. Acesso em: agosto de 2018.

- Gerenciamento de Dados de Tabelas Relacionais (Columnar ou Coluna-Armazena SQL RDBMS)
- Gerenciamento de Dados em Grafos de propriedades relacionais (SPARQL RDF based Quad Store)
- Gerenciamento de conteúdo (HTML, TEXTO, TURTLE, RDF/XML, JSON, JSON-LD, XML)
- Implementação de Dados Abertos Ligados Cinco Estrelas (Servidor de Dados Ligados Baseado em RDF)
- Servidor de Aplicações Web

Desta maneira, optou-se por utilizar o OpenLink Virtuoso Server no desenvolvimento do ambiente simulado deste estudo, pois além do armazenamento das triplas RDF ele permite a recuperação de resultados através da linguagem SPARQL.

2.2 INTERNET DAS COISAS (IoT)

Nesta seção estão abordados os principais assuntos que circundam o tema Internet das Coisas. No primeiro tópico, são apresentados os conceitos, definições e temas relacionados a IoT, seguindo por uma abordagem histórica e evolutiva sobre o assunto.

Posteriormente, são apresentadas as tecnologias utilizadas na implementação de sistemas de IoT, mostrando na sequência uma perspectiva dos modelos e arquiteturas encontradas. E por fim, apresenta-se as áreas de aplicação da IoT e suas respectivas convergências.

2.2.1 Conceitos, definições e temas relacionados

Atualmente a IoT se tornou um termo popular para descrever cenários nos quais a conectividade com a internet e a capacidade de computação se estendem a uma variedade de objetos, dispositivos, sensores e demais itens do dia-a-dia (ROSE; ELDRIDGE; CHAPIN, 2015).

No Quadro 2 são apresentadas as principais definições encontradas para o termo IoT, sendo apresentadas relacionando as respectivas fontes.

Quadro 2 – Definições de IoT

Definição	Fonte
Uma infraestrutura de rede global e dinâmica com recursos de autoconfiguração baseados em protocolos padrões e interoperáveis de comunicação, onde “coisas” físicas e virtuais têm identidades, atributos físicos e personalidades virtuais e usam interfaces inteligentes, e são perfeitamente integrados à rede de informações.	Sundmaecker et al. (2010, p. 43)
IoT é simplesmente o ponto no tempo onde o número de “coisas ou objetos” conectados superou a quantidade da população mundial.	Evans (2011, p. 2)
Uma infraestrutura global para a sociedade da informação, possibilitando serviços avançados ao interconectar coisas (físicas e virtuais) com base em tecnologias de informação e comunicação interoperáveis existentes e em evolução. NOTA 1 – Através da exploração das capacidades de identificação, captura de dados, processamento e comunicação, a IoT faz uso total das coisas para oferecer serviços a todos os tipos de aplicações, assegurando ao mesmo tempo que os requisitos de segurança e privacidade sejam cumpridos. NOTA 2 – De uma perspectiva mais ampla, a IoT pode ser percebida como uma visão com implicações tecnológicas e sociais.	ITU-T Y. 2060 (2012, p. 4)
A Internet das Coisas se desenvolve a partir da Internet atual, criando uma rede abrangente e auto organizável de objetos físicos conectados, identificáveis e endereçáveis, permitindo o desenvolvimento de aplicativos em setores verticais-chave através do uso de chips incorporados.	Schindler et al. (2013, p. 17)
IoT: A interconexão, via Internet, de dispositivos computacionais embarcados em objetos do dia a dia, que possibilitam o envio e recebimento de dados.	Oxford ³ (2018)

Fonte: O autor (2019)

³

Disponível

em:

<https://en.oxforddictionaries.com/definition/internet_of_things> Acesso em: setembro de 2018.

Pela definição apresentada pelo dicionário Oxford (2018) a Internet é considerada como um componente da IoT, mostrando uma conformidade com a definição estabelecida por Schindler et al. (2013).

Quanto as definições de Sundmaecker et al. (2010) e a Recomendação ITU-T Y. 2060 (2012) estas se mostram congruentes, porém, esta segunda apresenta uma abordagem com uma visão social, enquanto a primeira tem um enfoque mais no mundo físico e suas comunicações, padrões e representações virtuais. Vale ressaltar que, ambas são adotadas pelas instituições European Research Cluster on the Internet of Things (IERC) e International Telecommunications Union (ITU).

Já a abordagem dada por Evans (2011), apresenta uma visão diferenciada em relação a todas as outras definições, relacionando um fato interessante ao surgimento do conceito, apesar de que também considera dispositivos conectados à Internet, como os outros autores.

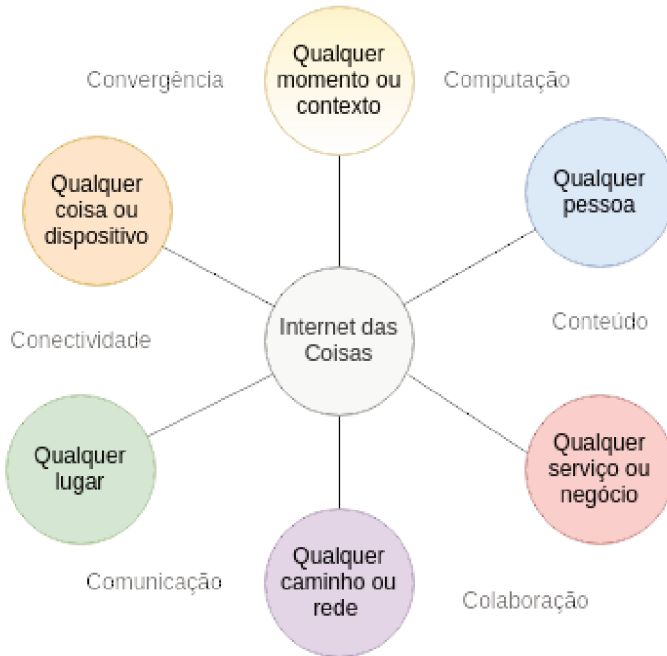
A Internet das Coisas permite que pessoas e coisas possam estar conectados a qualquer momento, em qualquer lugar, com qualquer coisa e qualquer um, idealmente usando qualquer caminho e/ou rede e qualquer serviço (SUNDMAEKER et al., 2010; VERMESAN et al., 2011; ITU-T Y.4000/Y.2060, 2012; ZEINAB; ELMUSTAFA, 2017). Toda esta interação implica em abordar elementos como: Conteúdo, Colaboração, Comunicação, Conectividade e Convergência; criando um contexto onde existe a interconexão perfeita entre pessoas e coisas e/ou entre coisas e coisas, conforme apresenta a Figura 6 (SUNDMAEKER et al., 2010).

Portanto, a IoT implica em uma interação simbiótica dos mundos real e físico e digital e virtual, onde entidades físicas possuem partes digitais e representação virtual na Internet; as coisas tornam-se conscientes do contexto e podem sentir, comunicar, interagir, trocar dados, informação e conhecimento (Sundmaecker et al. 2010; UCKELMANN; HARRISON; MICHAHELLES, 2011; FRIESS, 2013; GUILLEMIN et al., 2014; SERRANO et al., 2015).

Através do uso de algoritmos inteligentes de tomada de decisão em aplicações de software, com base nas informações coletadas, podem ser dadas respostas rápidas e apropriadas a fenômenos físicos. Estes algoritmos inteligentes podem ser utilizados para descartar dados irrelevantes, analisar padrões a partir de dados históricos, determinar características e correlações semânticas e estáticas, bem como, aumentar a escalabilidade e robustez nos ambientes de IoT (Sundmaecker et al. 2010;

UCKELMANN; HARRISON; MICHAELLES, 2011; ROSE; ELDRIDGE; CHAPIN, 2015).

Figura 6 – Conectividades da IoT



Fonte: Tradução do autor com base em Sundmaecker et al. (2010).

Além do termo IoT, também são encontradas pesquisas relacionadas que utilizam termos como Internet of objects, Web of things, Ubiquidade computacional (ubiquitous computing), Cyber physical systems (CPS), Computação pervasiva, Wireless Sensor Network, Internet of Nano Things (IoNT) e também como Internet of Everything (IoE) (EVANS, 2012; EZECHINA; OKWARA; UGBOAJA, 2015; MIRAZ et al., 2015; KHODADADI; DASTJERDI; BUYYA, 2016; ZEINAB; ELMUSTAFA, 2017).

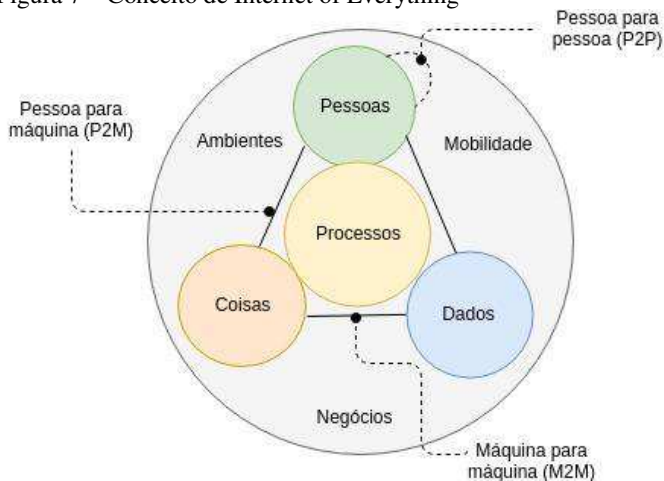
Para Atzori et al. (2010), a IoT pode ser separada em três paradigmas, sendo: orientado a internet (middleware), orientado as “coisas” (sensores) e orientado a semântica (conhecimento). Apesar deste tipo de delimitação ser necessária devido à natureza interdisciplinar do assunto, o autor afirma que a utilidade da IoT somente pode ser desencadeada em um domínio de aplicação em que os três paradigmas se cruzam.

Evans (2012) aborda o conceito da IoE que expande o conceito de IoT, com uma abordagem mais generalizada, sendo que enquanto na sua concepção a IoT é baseada em sensores físicos, o conceito de IoE aborda uma visão que abrange quatro componentes principais, sendo: Pessoas, Processos, Dados e Coisas, conforme é possível visualizar na Figura 7.

A (IoE) reúne pessoas, processos, dados e coisas para tornar as conexões em rede mais relevantes e valiosas do que nunca, transformando informações em ações que criam novos recursos, experiências mais ricas e oportunidades econômicas sem precedentes para empresas, indivíduos e países (EVANS, 2012, p. 3).

Apesar da definição fornecida sobre IoE, é possível observar pelos conceitos e definições apresentadas, a grande convergência entre estes dois termos. Segundo Khodadadi, Dastjerdi e Buyya (2016, p. 7) “a IoT é um facilitador nas comunicações entre máquina-máquina, humano-máquina e humano-humano”. Desta maneira, considera-se que apesar do termo IoE ter um enfoque maior na participação humana no processo, o conceito de IoT também aborda esta visão (KHODADADI; DASTJERDI; BUYYA, 2016).

Figura 7 – Conceito de Internet of Everything



Fonte: Tradução do autor com base em Evans (2012).

O fato é de que a visão do que exatamente a Internet das Coisas é, e o que será sua arquitetura final, ainda está divergindo e sendo foco de diversas pesquisas (Vermesan et al., 2011)

Diferentes nomes estão sendo dados a Internet das Coisas, mas os objetivos são o mesmo, em sentido amplo. Entre os termos usados para Internet das Coisas incluem os nomes: Internet de Objetos, Inteligência Integrada, Dispositivos Conectados e Tecnologia Onipotente, Onisciente e Onipresente. Além destes, também citam-se:

(1) Integrações de computação e processos físicos (Cyber Physical Systems), em que traz a abordagem dos mundos real e virtual juntos; (2) Computação Pervasiva sendo um ambiente de computador em que praticamente todos os objetos tem poder de processamento com conexões sem fio ou com fio para uma rede global; (3) Computação ubíqua ou tecnologia Calma, onde a tecnologia se torna praticamente invisível em nossas vidas; (4) Interação Máquina a Máquina significa nenhuma intervenção humana enquanto dispositivos estão se comunicando fim a fim; (5) HumanComputer Interaction envolve o estudo, planejamento e projeto de interação entre pessoas e computadores; (6) A Ambient Intelligence que é uma tecnologia em desenvolvimento que tornará cada vez mais nosso ambiente cotidiano sensível e responsivo. (MADAKAM; RAMASWAMY; TRIPATHI; 2015, p. 166)

Outros temas conhecidos como Cloud, Edge e Fog computing, Big Data, BI, Web semântica, Ambientes inteligentes, Robótica e Segurança e Privacidade também estão relacionadas ao contexto da IoT, deixando clara a interdisciplinaridade deste assunto, uma vez que pode ser encontrado em diferentes campos da ciência (Sundmaeker et al., 2010; UCKELMANN; HARRISON; MICHAHELLES, 2011; ZEINAB; ELMUSTAFA, 2017).

2.2.2 História e evolução

O termo Internet das Coisas foi usado primeiramente por Kevin Ashton em 1999, para descrever um sistema onde objetos do mundo físico, as chamadas “coisas”, pudessem se conectar à Internet. Nesta oportunidade, o termo foi utilizado em uma apresentação para ilustrar o poder desta conexão através de tags de identificação por radiofrequência (RFID), tendo como objetivo rastrear mercadorias sem a necessidade de intervenção humana, em cadeias corporativas de suprimentos (ASHTON, 2009; ROSE; ELDRIDGE; CHAPIN, 2015; KHODADADI; DASTJERDI; BUYYA, 2016).

Se tivéssemos computadores que soubessem tudo o que havia para saber sobre as coisas - usando dados que eles coletaram sem qualquer ajuda nossa - poderíamos rastrear e contar tudo e reduzir muito o desperdício, a perda e o custo. Saberíamos quando as coisas precisavam ser substituídas, consertadas ou recuperadas, e se eram frescas ou passadas. Precisamos capacitar os computadores com seus próprios meios de coletar informações, para que possam ver, ouvir e cheirar o mundo por si mesmos, em toda a sua glória aleatória. RFID e tecnologia de sensores permitem que os computadores observem, identifiquem e entendam o mundo - sem as limitações dos dados inseridos pelo ser humano (ASHTON, 2009, p. 1).

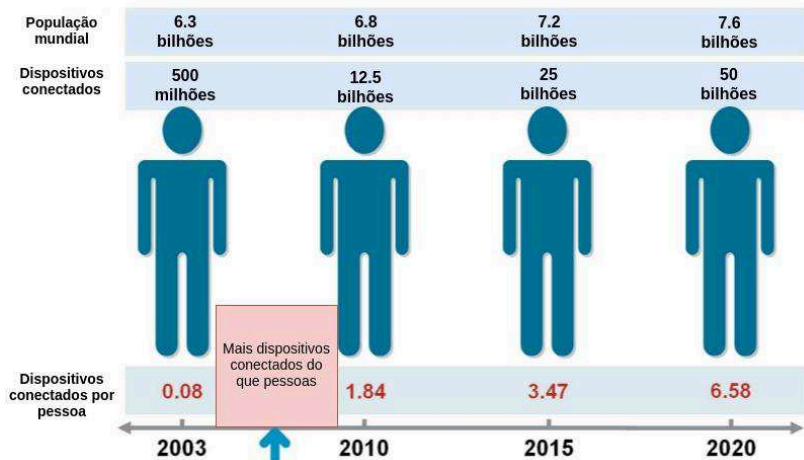
Apesar disso, a primeira publicação que faz menção ao termo IoT, foi de David Brock em 2001, em um paper sobre Codificação Eletrônica de Produtos (EPC) da Auto-ID Center, empresa da qual Kevin Ashton é co-fundador design (MADAKAM; RAMASWAMY; TRIPATHI, 2015).

A partir deste momento, um número crescente de pesquisadores e profissionais seguiram esta visão, conforme é possível encontrar em livros, conferências e simpósios relacionados ao tema (UCKELMANN; HARRISON; MICHAHELLES, 2011). Para Evans (2011), o surgimento da Internet das Coisas (IoT) ocorreu entre os anos de 2008 e 2009, pois foi o momento em que o número de dispositivos conectados à internet, passou a ser maior que a população mundial. A Figura 8 demonstra que no ano de 2003 havia aproximadamente 6,3 bilhões de pessoas vivendo no planeta e 500 milhões de dispositivos conectados à internet, já em

2010, com o rápido crescimento do uso de dispositivos como smartphones e tablets, o número de dispositivos já havia superado o número de habitantes na terra, sendo 12,5 bilhões de dispositivos, enquanto havia aproximadamente 6,8 bilhões de pessoas. Contudo, a previsão é que esta diferença seja ainda maior até o ano de 2020, existindo cerca de 50 bilhões de dispositivos conectados à internet com uma população mundial estimada em 7.6 bilhões (EVANS, 2011).

No entanto, vale ressaltar que as projeções futuras até o ano 2025 é que haja cerca de 100 bilhões de dispositivos conectados, gerando um impacto de mais de 11 trilhões de dólares na economia global (ROSE; ELDRIDGE; CHAPIN, 2015).

Figura 8 – Projeção do total de dispositivos conectados e da população mundial entre os anos de 2003 a 2020.



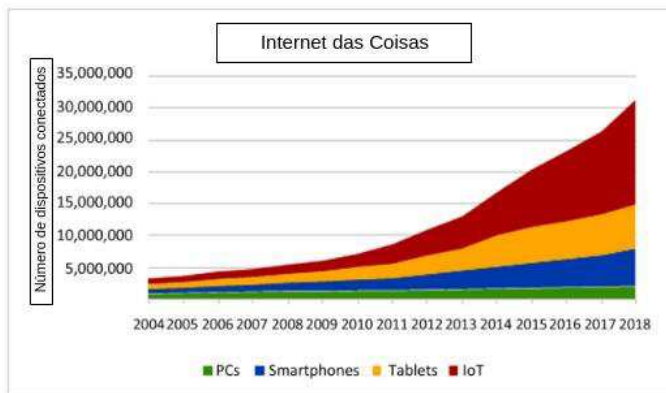
Fonte: Tradução do autor com base em Evans (2011).

Khodadadi, Dastjerdi e Buyya (2016) apud BI Intelligence apresentam ainda o crescimento de dispositivos conectados entre os anos de 2004 a 2018, que considera quatro grupos, sendo eles: Computadores pessoais (PC), Smartphones, Tablets e dispositivos IoT. Sendo possível visualizar um crescimento maior para dispositivos IoT, conforme apresentado na Figura 9.

Com as perspectivas futuras apresentadas quanto a expansão de dispositivos conectados, surgem diversos desafios a serem vencidos para que este ambiente se torne real, são questões que estão gerando uma série de pesquisas no campo da Internet das Coisas. Sendo assim, a IoT se

mostra como uma área emergente que não somente requer desenvolvimento de infraestruturas e tecnologias, mas também de novos serviços capazes de suportar aplicações múltiplas, escaláveis e interoperáveis (BALDINI et al., 2015; SERRANO et al., 2015).

Figura 9 – Total de dispositivos conectados separados por tipo entre os anos de 2004 a 2018.



Fonte: Tradução do autor com base em Khodadadi; Dastjerdi; Buyya (2016) apud BI Intelligence (2015).

Contudo, a IoT representa na realidade uma nova evolução da internet. Considerando o impacto que a internet já promoveu em áreas como educação, comunicação, negócios, ciência, governo e sociedade, claramente a torna como uma das mais importantes e poderosas criações da história da humanidade. Nesse contexto, atualmente a internet está dando um enorme salto em sua capacidade de coletar, analisar e distribuir dados, aumentando o valor da informação criada pelas interconexões entre máquinas e humanos. A IoT possibilita processar e transformar dados em sabedoria, através das informações e conhecimentos gerados, o que impacta diretamente na maneira em como vivemos e nos relacionamos com os objetos do dia-a-dia, trazendo benefícios à humanidade e sociedade, conforme demonstrado na Figura 10 (EVANS, 2011; VERMESAN, 2011).

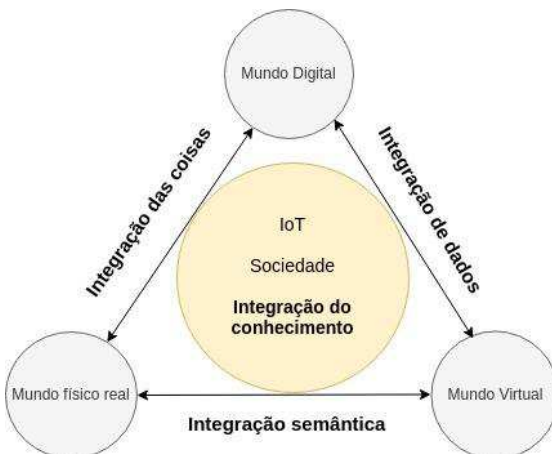
Figura 10 – Transformação de dados em sabedoria relacionados aos benefícios à humanidade



Fonte: Adaptada com base em Ackoff (1989), Vermesan (2011) e Evans (2011)

Para Sundmaeker et al. (2010), os benefícios da IoT para a sociedade e a possibilidade de integrar e gerar novos conhecimentos acontece através da integração dos mundos físico, digital e virtual. Sendo que entre o físico e digital, acontece no sentido de integrar as coisas (objetos físicos), já a interação entre os mundos digital e virtual se dá a partir da integração a nível de dados, e quanto ao físico e virtual, a integração acontece em nível da semântica, conforme apresentado na Figura 11 (SUNDMAEKER; 2010; EVANS, 2011; VERMESAN, 2011).

Figura 11 – Integração do conhecimento no contexto da IoT



Fonte: Tradução do autor com base em SUNDMAEKER et al. (2010).

No tópico a seguir são apresentadas as tecnologias principais que englobam a área da IoT.

2.2.3 Tecnologias

Segundo Vermesan et al. (2013) a Internet das Coisas fornece soluções baseadas na integração de tecnologias da informação, no que se refere a hardware e software usados para armazenar, recuperar e processar dados e tecnologias de comunicação, o que inclui sistemas eletrônicos usados para comunicação entre indivíduos e grupos.

Quando se trata sobre os eletrônicos, Mcewen e Cassimally (2013) apoiam a divisão na IoT entre sensores e atuadores. O sensor consiste em um dispositivo que tem a capacidade de medir alguma propriedade física do mundo real, e o atuador se trata de um dispositivo que tem a capacidade de executar uma operação ou controle de um sistema ou em uma entidade física do mundo real. Dentro ambas as categorias, os componentes eletrônicos podem conversar com o computador de várias maneiras (HACHEM; TEIXEIRA; ISSARNY, 2011; MCEWEN; CASSIMALLY, 2013).

A possibilidade de usuários trazerem objetos físicos para a esfera do mundo cibernético foi devido a criação de diferentes tecnologias de marcação, tais como: NFC, RFID e código de barras 2D. Estas tecnologias permitiram que objetos físicos fossem identificados e conectados a internet (MADAKAM; RAMASWAMY; TRIPATHI, 2015).

Atualmente microcontroladores como Arduino e Raspeberry Pi tem sido bastante utilizados para prototipação de soluções IoT, sendo utilizados até em aplicações em produção. Com a capacidade de processamento, memória, fonte de energia e comunicação em rede é possível controlar sensores e atuadores e gerar dispositivos inteligentes conectados à internet com um custo relativamente baixo (MCEWEN; CASSIMALLY, 2013; SANTOS et al., 2016).

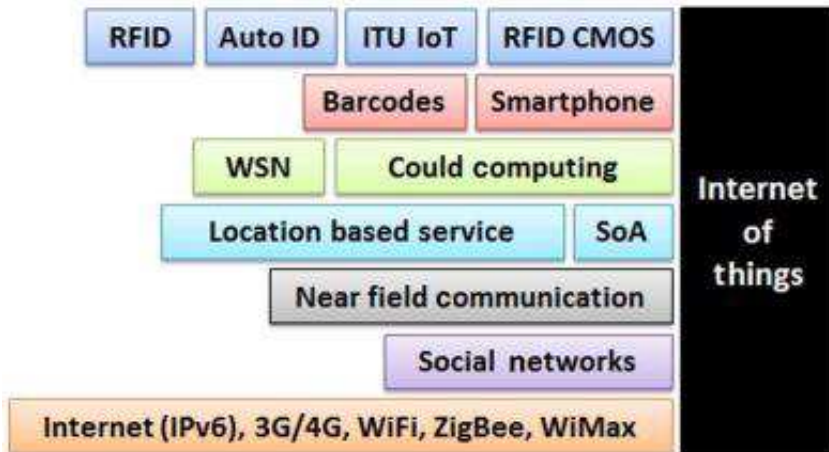
Quanto a tecnologias de comunicação, há uma mistura heterogênea delas para acomodar a diversidade da IoT, e que precisam ser adaptadas para atender às necessidades de aplicações, como eficiência de energia, segurança e confiabilidade. Alguns exemplos de padrões nessas categorias incluem tecnologias com e sem fio, como: Ethernet, Wi-Fi, Bluetooth, ZigBee e Z-Wave (VERMESAN et al., 2013). Protocolos como CoAP e MQTT para comunicação dos dispositivos com servidores,

bem como arquiteturas orientadas a serviço (SOA) e APIs vem sendo bastante utilizadas em aplicações de IoT (DA XU, HE, LI, 2014).

Da Xu, He e Li (2014) apresentam as principais tecnologias envolvidas na área da IoT, sendo possível visualizá-las na Figura 12.

A IoT requer não apenas o desenvolvimento de infra-estrutura, mas também a implantação de novos serviços capazes de suportar múltiplos aplicativos escaláveis (baseados em nuvem) e interoperáveis (multi-domínio). Na corrida de projetar a IoT as comunidades acadêmicas da indústria de Tecnologia da Informação e Comunicação (TIC) perceberam que um problema comum da IoT a ser enfrentado é a interoperabilidade das informações e serviços (SERRANO et al., 2015). Neste contexto, as tecnologias semânticas tem papel chave, uma vez que possibilita o compartilhamento, descoberta e reuso de objetos virtuais como serviços na nuvem, de forma interoperável e flexível (VERMESAN et al., 2013).

Figura 12 – Tecnologias associadas à IoT



Fonte: Da Xu, He e Li (2014)

Segundo Schindler (2013), IoT e big data habilitam-se reciprocamente: enquanto os sensores são capazes de fornecer os grandes volumes de dados fundamentais para análise de dados da IoT, perceber o potencial de várias aplicações de IoT depende da análise e feedback contínuo de grandes fluxos de dados através de big data (SCHINDLER, 2013).

A utilização de computação em nuvem atende a requisitos de processamento e armazenamento sob demanda, podendo ser usada para

analisar dados gerados por objetos IoT (ROSE; ELDRIDGE; CHAPIN, 2015; Khodadadi; Dastjerdi; Buyya, 2016). Porém, para Khodadadi, Dastjerdi e Buyya (2016) as abordagens atuais de análise de dados concentram-se principalmente em lidar com Big Data, no entanto, o processamento de dados gerados a partir de milhões de sensores e dispositivos em tempo real é mais desafiador (Khodadadi; Dastjerdi; Buyya, 2016).

Soluções que utilizam apenas a computação em nuvem (Cloud computing) e Big data para processamento e armazenamento, nem sempre atendem à escalabilidade e a restrições de latência de aplicativos em tempo real, por exemplo. Requisitos de processamento em tempo real e o aumento do poder computacional dos dispositivos de ponta, como roteadores, switches e pontos de acesso, levam ao surgimento do paradigma Edge ou Fog Computing, o que pode ser tratado como uma extensão da computação em nuvem que visa manter os mesmos recursos da nuvem, como rede, computação, virtualização e armazenamento, mas também atende aos requisitos de aplicativos que exigem baixa latência, requisitos específicos de qualidade de serviço e Acordo de Nível de Serviço (SLA) (Khodadadi; Dastjerdi; Buyya, 2016).

No tocando a relação entre computação em nuvem e IoT, Schindler (2013, p. 10-11) expande uma visão com algumas das principais possibilidades e vantagens com a convergência destas áreas, conforme apresentado nos tópicos a seguir:

- Os sensores de IoT produzirão quantidades de dados sem precedentes, a coleta, o armazenamento, o processamento combinado e a disponibilidade onipresente se tornará cada vez mais importante. As instalações de computação em nuvem são ideais para fornecer isso e, assim, garantir que os benefícios da IoT sejam distribuídos da forma mais ampla possível. A consequência é que novos modelos de negócios e serviços e novos aplicativos podem ser alcançados a um custo menor.
- As instalações em nuvem podem coordenar a coleta de dados específicos apropriados para o monitoramento de eventos e para a detecção de problemas emergentes, projetando e implementando estratégias de detecção em várias entidades e locais da IoT.

- Os cálculos baseados em nuvem podem ser usados para instruir dispositivos IoT a executar ações de maneira coordenada e distribuída.
- Os aplicativos baseados em nuvem podem ser criados e implantados para usar na aquisição de dados dos sensores.
- O compartilhamento de recursos de hardware (como sensores) por meio da nuvem, no entanto, pode aumentar as preocupações relacionadas à privacidade e à continuidade do serviço.

No entanto, sensores, atuadores, tecnologias de marcação (RFID, NFC, etc.), tecnologias de comunicação e de interoperabilidade, nanotecnologia e tecnologia integrada de inteligência com servidor são a base para possibilitar que uma variedade de objetos físicos e dispositivos possam ser associados à internet, e que cooperem entre si para realização de um objetivo comum (VERMESAN et al., 2013; DA XU, HE, LI, 2014; MADAKAM; RAMASWAMY; TRIPATHI, 2015). E tecnologias como Big data e Cloud e Fog Computing apoiam no fornecimento de um ambiente adequado para atender requisitos de escalabilidade, alta disponibilidade, análises e processamento de grande volume de dados, entre outros (Khodadadi; Dastjerdi; Buyya, 2016).

Segundo Vermesan et al. (2013, p. 20) os principais aspectos tecnológicos que devem convergir para permitir o funcionamento de uma aplicação de IoT, sendo eles:

- Arquitetura IoT
- Identificação
- Comunicação
- Tecnologia de redes
- Descoberta de rede (Network Discovery)
- Softwares e algoritmos
- Tecnologias de hardware
- Processamento de dados e sinais
- Motores de busca e descoberta
- Gestão de rede
- Armazenamento e baixo consumo de energia
- Segurança, confiabilidade e privacidade
- Interoperabilidade
- Padronização

Contudo, existem diversos cenários de aplicações para IoT e a convergência destas tecnologias estão ligadas diretamente aos requisitos de cada uma destas aplicações, sendo exploradas as áreas e convergências no tópico a seguir.

2.2.4 Áreas de aplicação e suas convergências

Com o avanço da infraestrutura e da comunicação sem fio, a proliferação de novos dispositivos de comunicação e o surgimento do paradigma da computação em nuvem, estão surgindo diversos aplicativos baseados na Internet das Coisas (SARKAR, 2016). Estes aplicativos, distribuídos em diversos dispositivos, estão mais integrados ao ambiente de negócios do que nunca, gerando novas oportunidades para atender aos requisitos de negócios, como criar novos serviços em tempo real com base em dados do mundo físico, obter insights sobre processos e relacionamentos complexos, lidar com incidentes, abordar a degradação ambiental (poluição, desastres, aquecimento global, etc.), monitorar atividades humanas (saúde, movimentos, etc.), melhorar a integridade da infraestrutura (energia, transporte, etc.) e abordar questões de eficiência energética (medição inteligente de energia em edifícios, consumo eficiente de veículos, etc.) (SUNDMAEKER et al., 2010; VERMESAN et al., 2013).

Por exemplo, câmeras inteligentes e acessíveis em rede podem ser colocadas em locais estratégicos em um conjunto de prédios ou em ruas, medidores inteligentes podem ser instalados em uma rede elétrica, minúsculos dispositivos embarcados podem ser usados para monitorar a saúde, veículos em uma cidade podem ser equipados com sensores baseados em GPS, e sensores sem fio estáticos podem ser incorporados em aparelhos modernos como uma televisão ou uma geladeira. Esses dispositivos habilitados para rede podem executar processos distribuídos, que por sua vez podem coordenar, trocar dados e tomar decisões críticas em tempo real. Espera-se que tal sistema seja implantado uma vez e esteja operacional para sempre (VERMESAN et al., 2013; SARKAR, 2016).

As potencialidades oferecidas pela IoT possibilitam o desenvolvimento de um grande número de aplicações, das quais apenas uma parte muito pequena está atualmente disponível para a nossa sociedade. Entre as aplicações possíveis, podemos distinguir entre aqueles diretamente

aplicáveis ou mais próximos de nossos habitus atuais e aqueles futuristas, que só podemos imaginar no momento, uma vez que as tecnologias e / ou nossas sociedades não estão prontas para sua implantação. (ATZORI et al., 2010)

Muitos são os domínios e os ambientes nos quais aplicações de IoT podem melhorar a qualidade de vida do ser humano, como: em casa, durante uma viagem, em situação de doença, no trabalho, quando realiza corridas ou exercícios na academia, por exemplo. Esses ambientes estando são equipados com objetos com inteligência primitiva, na maioria das vezes sem nenhum recurso de comunicação. Dar a esses objetos a possibilidade de se comunicar uns com os outros e elaborar as informações percebidas a partir do entorno implica em diferentes ambientes com uma ampla gama de aplicações que podem ser implantadas (ATZORI et al., 2010).

Segundo Chui; Loffler e Roberts (2010), os tipos de aplicações da IoT podem ser separados em dois grupos principais, sendo eles: Informação e análise e Automação e controle. Cada grupo conta com três tipos de aplicações que remetem ao seu grupo, sendo para Informação e análise os itens: Monitoramento de comportamento, sensibilidade ao contexto e tomada de decisões baseada em dados. Para as aplicações de Automação e controle, os itens são: Otimização de processos, otimização de consumo de recursos e autonomia de sistemas complexos.

Para Evans (2011), a IoT pode ser entendida como uma rede de redes:

A IoT é composta por uma coleção dispersa de redes diferentes e específicas. Os carros de hoje, por exemplo, têm várias redes para controlar a função do motor, os recursos de segurança, os sistemas de comunicação e assim por diante. Edifícios comerciais e residenciais também têm vários sistemas de controle para aquecimento, ventilação e ar condicionado; serviço telefônico; segurança; e iluminação. À medida que a IoT evoluir, essas redes e muitas outras serão conectadas com recursos adicionais de segurança, análise e gerenciamento. Isso permitirá que a IoT se torne ainda mais poderosa no que ela pode ajudar as pessoas a alcançar (EVANS, 2011, p. 4)

Para Atzori et al. (2010) os domínios de aplicação de IoT podem ser agrupados entre os domínios de: transporte e logística, setor de saúde, ambientes inteligentes (casa, escritório, etc.) e no pessoal e social. Contudo, Friess (2013) apresenta as áreas potenciais para aplicação da IoT separando em: Cidades Inteligentes (Smart Cities), Carros Inteligentes e mobilidade, Casas Inteligentes e gravação assistida (assisted living), Indústrias inteligentes, Saúde pública, Energia e ambientes de proteção, agricultura e turismo tem ganhado grande atenção.

No entanto, Manyika (2015) apresenta uma separação dos aplicativos de IoT por meio de uma lente de cenários, onde considera o contexto dos ambientes físicos nos quais sistemas IoT podem ser implantados, e captura maneiras pelas quais eles criam valor para todas as partes envolvidas em cada cenário (empresas, consumidores, trabalhadores). Sendo assim, ele define nove cenários que capturam o uso de IoT em locais como residências, escritórios, fábricas, locais de trabalho (mineração, petróleo e gás e construção) e cidades, conforme apresentado no Quadro 3.

Quadro 3 – Aplicações para IoT

Aplicação	Descrição	Exemplos
Humano	Dispositivos acoplados ao corpo humano	Dispositivos (vestíveis ou ingeríveis) para monitorar e manter a saúde e bem estar. Gerenciamento de doenças. Aumento de produtividade.
Residências	Construções onde pessoas vivem	Controladores de residências e sistemas de segurança.
Ambientes de venda	Espaços onde consumidores realizam compras	Lojas, bancos, restaurantes, arenas, etc. Qualquer lugar onde pessoas frequentam e compram. Otimização de inventários.
Escritórios	Ambientes onde trabalhadores colaboram	Gestão de energia e segurança em escritórios. Melhoria de produtividade, incluindo trabalhadores remotos.

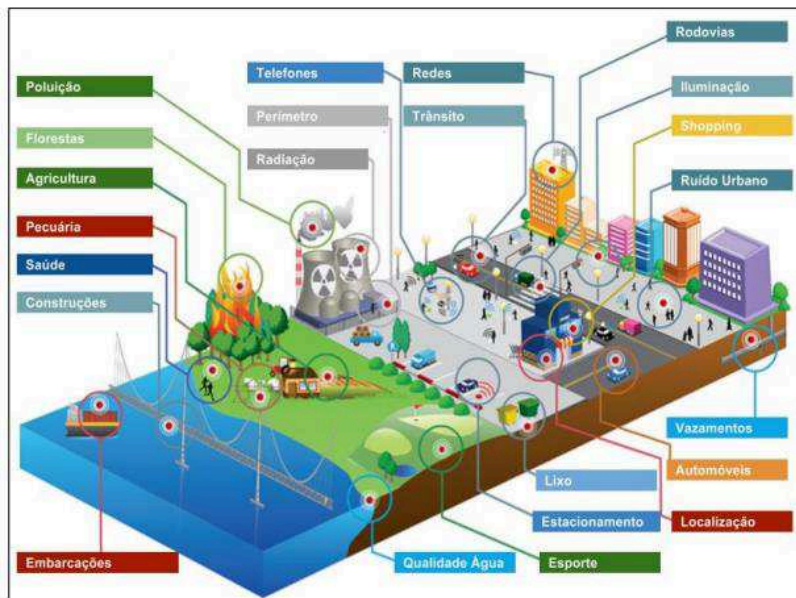
(continuação)

Aplicação	Descrição	Exemplos
Fábricas	Ambientes de produção padronizada	Lugares com rotinas repetitivas de trabalho, incluindo fazendas e hospitais, operações eficientes, otimização de uso de equipamentos e inventário.
Locais de trabalho	Ambientes de produção customizada	Minação, petróleo e gás, construção; operações eficientes, manutenção preditiva, saúde e segurança
Veículos	Sistemas em veículos móveis	Veículos incluindo carros, caminhões, embarcações, aviões e trens; manutenção baseada na condição, design baseado no uso, análise de pré-venda
Cidades	Ambientes urbanos	Espaços públicos e infraestrutura urbana; controle de tráfego adaptativo, monitoramento de ambiente, gestão de recursos
Ambientes externos	Entre ambiente urbano e outras características externas	Ambientes externos incluindo rodovias, veículos autônomo, navegação aérea, roteirização em tempo real, navegação integrada e rastreamento de embarcações

Fonte: Tradução do autor com base em Manyika (2015).

Portanto, a IoT pode ser considerada um fenômeno que contempla diferentes contextos, conectando diversos ambientes e possibilitando interação entre eles (MANYIKA, 2015; LIBELIUM, 2018). O que pode ser chamado de mundo inteligente, conforme é possível visualizar no cenário apresentado na Figura 13 (LIBELIUM, 2018).

Figura 13 - Visão de mundo inteligente no contexto da IoT



Fonte: Tradução do autor com base em Libelium (2018).

A seguir são apresentadas as arquiteturas e modelos de referência para IoT que foram utilizadas na elaboração do modelo proposto nesta dissertação.

2.2.5 Arquiteturas e modelos de referência

Uma arquitetura holística para sistemas de IoT visa garantir uma operação perfeita de seus componentes (a confiabilidade é considerada o fator de design mais importante na IoT) e vincular os mundos: físico e virtual. Portanto, é necessário considerar a possibilidade de projetar recuperação de falhas e escalabilidade (SCHINDLER, 2013; Khodadadi; Dastjerdi; Buyya, 2016). Além disso, como a mobilidade e a mudança

dinâmica de localização se tornaram parte integrante dos sistemas de IoT com o uso disseminado de smartphones, por exemplo, as arquiteturas de ponta precisam ter um certo nível de adaptabilidade para lidar adequadamente com interações dinâmicas em todo o ecossistema da IoT (Khodadadi; Dastjerdi; Buyya, 2016).

Até o momento, as aplicações de IoT foram baseadas em implementações de software fragmentadas para sistemas e casos de uso específicos. A grande necessidade de arquiteturas de referência para IoT tornou-se tangível com o crescente número de iniciativas voltadas para arquiteturas padronizadas. Essas iniciativas visam facilitar a interoperabilidade, simplificar o desenvolvimento e facilitar a implementação. E tal referência pode servir como uma diretriz genérica geral, pois nem todos os aplicativos de domínio exigem cada detalhe para a implementação em um cenário real (WEYRICH; EBERT, 2016).

Arquiteturas de referência da IoT lidam com seus requisitos e formam um conjunto genérico de funcionalidades, estruturas de informação e mecanismos. A consideração adicional de entidades e sua interação levam a um modelo de referência. Esse modelo integra aspectos das entidades relacionadas, como usuários humanos, implementações de dispositivos e estruturas de servidor, e fornece uma visão mais completa, ou modelos, da configuração geral e de sua implementação de domínio. A arquitetura e o modelo ajudam a descrever e mapear tecnologias para casos de negócios (WEYRICH; EBERT, 2016).

A arquitetura da IoT está surgindo de maneira evolucionária, a partir de uma variedade de contribuições desconectadas, pois elas vêm de várias partes interessadas com origens e objetivos diferentes. Importante, atualmente não há arquitetura única emergente. Uma arquitetura de IoT pode ser considerada consistindo em vários pilares de suporte, abordados a seguir

- um modelo de arquitetura, com seus princípios conceituais para processamento distribuído, comunicações e armazenamento, com suas interações frequentemente em execução em links de rádio, que requerem espectro adequado (licenciado ou isento de licença)
- esquemas de identificação, com sistemas de nomeação e endereçamento, para rotular e localizar as coisas (ou objetos), esquemas que são flexíveis e interoperáveis para trabalhar em vários

namespaces, talvez exigindo tradutores de endereços (de rede) ou diretórios de nomenclatura

- segurança e privacidade, que precisam ser incorporadas ao projeto arquitetônico desde o primeiro dia - e não adicionadas posteriormente, como a Internet experimentou
- padrões que sustentam a IoT e formam a base para sua arquitetura técnica (SCHINDLER, 2013, p. 48)

Contudo, arquiteturas e modelos de referência dão uma visão geral de todo o sistema subjacente, portanto, sua vantagem sobre outras arquiteturas depende de fornecer um nível de abstração melhor e maior, que, conseqüentemente, oculta restrições específicas e detalhes de implementação (Khodadadi; Dastjerdi; Buyya, 2016).

Atualmente ainda não existe um modelo conceitual padronizado que caracterize e padronize as várias funções de um sistema IoT, várias iniciativas de modelos de Arquitetura de Referência da IoT foram lançadas, e outras estão em desenvolvimento (SCHINDLER, 2013; VERMESAN et. al., 2013; DA XU; HE; LI, 2014).

Torkaman e Seyyedi (2016) apresentam um levantamento e análise de arquiteturas quatro arquiteturas de IoT, comentando as principais características e abordagens de cada uma, sendo elas IoT-RA proposta pela ISO, das recomendações da ITU-T e dos modelos WSO2 proposto por Fremantle (2015) e ETSI.

Das iniciativas relacionadas a arquiteturas generalizadas para IoT que foram identificadas nesta pesquisa, existem propostas desenvolvidas por instituições e organizações governamentais e também por grandes players do mercado, sendo estas apresentadas no Quadro 4.

Como nesta pesquisa o foco está em abordagens generalizadas de ambientes de IoT, as arquiteturas que abordam aplicações de áreas específicas não foram consideradas neste estudo, como exemplo das especificações da ETSI, que são focadas na comunicação M2M (TORKAMAN; SEYYEDI; 2016).

Quadro 4 – Arquiteturas e especificações para IoT

Nome	Fonte
A Reference Architecture For The Internet of Things - WSO2	FREMANTLE, 2015.
Information technology – Internet of Things Reference Architecture (IoT RA)	ISO/IEC 30141. Disponível em: < https://www.iso.org/standard/65695.html >. Acesso em: novembro de 2018.
IoT-A	BASSI et al., 2011.
ITU-T	ITU –T Y.4000/Y.2060, 2012 e ITU-T Y.4100/Y.2066, 2014.
Wise-IoT	Wise-IoT. Disponível em: < http://wise-iot.eu/wp-content/uploads/2018/09/D1.3-Wise-IoT-Updated-Architecture-PU-V1.0.pdf >. Acesso em: outubro de 2018.
Intel - IoT Platform Reference Architecture	Intel IoT Plataforma RA. Disponível em: < https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/iot-platform-reference-architecture-paper.pdf >. Acesso em: setembro de 2018.
Microsoft Azure IoT Reference Architecture	Microsoft Azure IoT RA. Disponível em: < https://azure.microsoft.com/pt-br/blog/azure-iot-reference-architecture-2-1-release/ >. Acesso em: setembro de 2018.
Web of Things (WoT)	GUINARD; TRIFA, 2009.
Semantic Web of Things (SWoT)	GYRARD; BONNET e BOUDAUD, 2014.

Fonte: O autor (2019)

Na seção a seguir são comentados os requisitos e modelos conceituais considerados nesta pesquisa.

2.2.6 Requisitos e modelos conceituais

Uma visão simplificada dos requisitos de uma arquitetura de IoT é apresentada por Weyrich e Ebert (2016), onde comenta que os requisitos gerais são:

- Conectividade e comunicação, podendo envolver conectividade um-a-um (unicast) ou para vários usuários (multicast e anycast) na coleta de dados e disseminação de informações.
- O gerenciamento de dispositivos deve fornecer soluções quando um dispositivo é adicionado ou uma configuração é alterada, e esta deve ser possível propagar para outros dispositivos.
- A coleta, análise e atuação de dados são relevantes para a extração de informações e conhecimentos e na oferta de serviços.
- A escalabilidade é importante para lidar com grandes volumes de processamento em diferentes ambientes da IoT.

Os recursos de segurança são necessários para fornecer confiança e privacidade e são necessários em todos os aspectos da IoT.

Contudo, dos modelos de arquiteturas e especificações encontradas que foram comentadas no tópico anterior, identificou-se que os documentos que disponibilizam um conteúdo detalhado e uma abordagem mais genérica sobre soluções de IoT, sendo utilizada como diretriz principal na elaboração do modelo proposto as seguinte recomendações da ITU-T:

- [ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), Overview of the Internet of things.
- [ITU-T Y.4100] Recommendation ITU-T Y.4100/Y.2066 (2014), Common requirements of the Internet of things.

Segundo Y.4100/Y.2066 (2014) os requisitos funcionais de IoT referem-se aos requisitos relacionados aos agentes de IoT, ou seja, entidades que são externas à IoT e que interagem com a IoT. Estes requisitos funcionais são categorizados da seguinte forma:

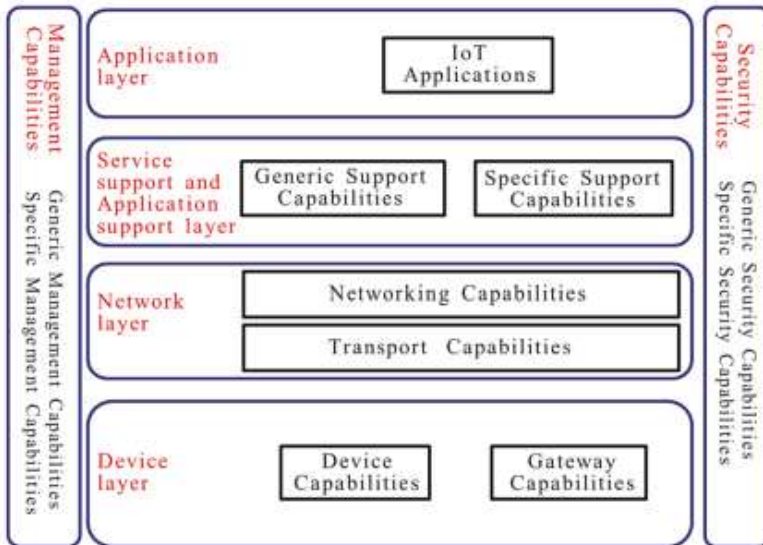
- Suporte à aplicação
- Serviço

- Comunicação
- Dispositivo
- Gerenciamento de dados
- Proteção de segurança e privacidade

No ANEXO A é apresentada uma tradução do detalhamento dos requisitos da recomendação ITU-T Y.4100/Y.2066 (2014).

Quanto ao modelo de referência da IoT sugerido na recomendação ITU-T Y.4000/Y.2060 (2012) e apresentado na Figura 14. Este é composto de duas camadas verticais, sendo elas: de gestão e de segurança; e quatro camadas horizontais: a de aplicação, suporte de serviço e camada de suporte à aplicação, de rede e do dispositivo.

Figura 14 – Modelo de arquitetura proposta pela ITU-T



Fonte: ITU-T Y.4000/Y.2060 (2012, p. 7)

Uma tradução do autor, segundo ITU-T Y.4000/Y.2060 (2012) sobre as camadas da arquitetura é apresentada nos tópicos a seguir:

- Aplicação: A camada de aplicativo contém aplicativos IoT.

- Suporte de serviço e camada de suporte à aplicação: A camada de suporte a serviços e suporte a aplicativos consiste nos dois agrupamentos de recursos a seguir:
 - Recursos genéricos de suporte: os recursos genéricos de suporte são recursos comuns que podem ser usados por diferentes aplicativos de IoT, como processamento ou armazenamento de dados. Esses recursos também podem ser invocados por recursos específicos de suporte, por exemplo, para construir outros recursos específicos de suporte.
 - Capacidades específicas de suporte: As capacidades específicas de suporte atendem aos requisitos das aplicações. Eles podem consistir em vários agrupamentos de recursos, a fim de fornecer diferentes funções de suporte para diferentes aplicativos de IoT.
- Rede: Esta camada consiste nos dois tipos de recursos a seguir:
 - Recursos de rede: fornecem funções de controle relevantes de conectividade de rede, como funções de controle de recursos de acesso e transporte, gerenciamento de mobilidade ou autenticação, autorização e auditoria (AAA).
 - Capacidades de transporte: foco no fornecimento de conectividade para o transporte de serviços de IoT e informações de dados específicos da aplicação, bem como, o transporte de informações gerenciais de controle.
- Dispositivo: Os recursos da camada de dispositivo podem ser categorizados logicamente em dois tipos de recursos:
 - capacidades do dispositivo: Os recursos do dispositivo incluem, mas não estão limitados a Interação direta com a rede de comunicação: os dispositivos podem coletar e carregar informações diretamente (ou seja, sem usar recursos de gateway) para a rede de comunicação e podem receber informações diretamente (por exemplo, comandos) da rede de comunicação. Interação indireta com a rede de comunicação: os dispositivos podem coletar e carregar informações indiretamente na rede de comunicação,

ou seja, através de recursos de gateway, mas também podem receber indiretamente informações (por exemplo, comandos) da rede de comunicação. E rede ad-hoc: os dispositivos podem ser capazes de construir redes de maneira ad-hoc em alguns cenários que precisam de maior escalabilidade e implantação rápida.

- capacidades de gateway: Os recursos de gateway suportam dispositivos conectados através de diferentes tipos de tecnologias com ou sem fio, como um barramento de rede de área (CAN), ZigBee, Bluetooth ou Wi-Fi. Na camada de rede, os recursos de gateway podem se comunicar por meio de várias tecnologias, como rede telefônica pública comutada (PSTN), segunda geração ou terceira geração (2G ou 3G), redes de evolução de longo prazo (LTE), Ethernet ou por Digital Subscriber Lines (DSL). Quanto a conversão de protocolo: existem duas situações em que os recursos de gateway são necessários. Uma situação é quando as comunicações na camada do dispositivo usam protocolos diferentes na própria camada de dispositivo, por exemplo, protocolos de tecnologia ZigBee e protocolos de tecnologia Bluetooth. A outra é quando as comunicações envolvendo a camada de dispositivo e a camada de rede usam protocolos diferentes, por exemplo, um protocolo de tecnologia ZigBee na camada de dispositivo e um protocolo de tecnologia 3G na camada de rede.
- Gerenciamento: De maneira semelhante às redes de comunicação tradicionais, os recursos de gerenciamento de IoT abrangem as classes tradicionais de falha, configuração, contabilidade, desempenho e segurança (FCAPS), ou seja, gerenciamento de falhas, gerenciamento de configuração, gerenciamento contábil, gerenciamento de desempenho e gerenciamento de segurança. Os recursos de gerenciamento de IoT podem ser categorizados em recursos de gerenciamento genéricos e recursos de gerenciamento específicos.
- Segurança: Existem dois tipos de recursos de segurança: recursos genéricos de segurança e capacidades de

segurança específicas. Quanto aos recursos genéricos de segurança são independentes dos aplicativos e eles incluem:

- na camada de aplicação: autorização, autenticação, confidencialidade dos dados da aplicação e proteção de integridade, proteção de privacidade, auditoria de segurança e antivírus;
- na camada de rede: autorização, autenticação, dados com confidencialidade e de proteção da integridade;
- na camada do dispositivo: autenticação, autorização, validação da integridade do dispositivo, controle de acesso, confidencialidade dos dados e proteção da integridade.

A partir do estudo dos modelos e arquiteturas de IoT levantados identifica-se que algumas visões se convergem com as especificações da ITU-T, e até mesmo abordam conteúdos específicos em maiores detalhes ou até mesmo que não são tratados no conteúdo selecionado da ITU-T, portanto, também foram consideradas neste estudo.

Neste sentido, na especificação da Intel são apresentadas características de Fog Computing, e de processamento de alto volume de dados. O que converge com os requisitos gerais apresentados anteriormente por Weyrich e Ebert (2016).

Já o modelo Wise-IoT apresenta uma visão mais detalhada sobre questões de aplicação semântica, sendo estas também consideradas no modelo, principalmente no tocante a camada de semântica.

Os autores Guinard e Trifa (2009) apresentam o conceito de Web das Coisas (WoT), visando uma solução para o problema de interoperabilidade da IoT, onde dão uma definição deste termo como “uma representação digital de um objeto físico acessível por meio de uma API Web RESTful”. O modelo proposto por eles impõe implementações técnicas que permitem que as “coisas” integrem facilmente a Web, por exemplo, com o uso do formato JSON. Neste contexto, a WoT necessariamente se comunica por protocolo HTTP e implementa uma API usando o estilo arquitetural RESTful (GUINARD; TRIFA, 2009; SZILAGYI; WIRA, 2016).

Guinard et. al (2011) ainda comenta sobre a importância e benefícios em aplicar técnicas da web semântica no modelo da WoT,

porém, não apresenta maiores detalhes sobre como aplicá-las mostrando apenas como possibilidades de trabalhos futuros.

2.3 ONTOLOGIAS E WEB SEMÂNTICA PARA IoT

Nesta seção são apresentados os modelos de ontologias e estudos relacionados a aplicação da web semântica em ambientes de IoT, mostrando arquiteturas e requisitos relacionados a estes ambientes e que foram utilizados na elaboração do modelo proposto apresentado no capítulo 4 .

Observamos que as diretrizes da web semântica geralmente não são conhecidas pela comunidade IoT, o que impede a automação ou reutilização de conhecimento de domínio (ontologias, conjuntos de dados e regras), enquanto inicialmente uma ontologia foi projetada para ser facilmente compartilhada e reutilizada (GYRARD; BONNET E BOUDAUD, 2014, p-1).

No tópico a seguir são apresentados as principais ontologias que tem sido utilizadas na realização de pesquisas e projetos sobre IoT, conforme o levantamento bibliográfico realizado.

2.3.1 Modelos de ontologias para IoT

Foram identificadas diversas iniciativas de ontologias para domínios de IoT, algumas mais genéricas e outras para aplicações mais específicas. Contudo, ontologias em IoT foram desenvolvidas para vários usos, incluindo a descrição de sensores e redes de sensores, recursos e serviços de IoT, “Coisas” inteligentes, entre outras (BARNAGHI et al., 2012; DESAI, SHETH, ANANTHARAM, 2015).

Na IoT é normal pensar em vários dispositivos conectados (inteligentes ou não, essa não é uma restrição para conectá-los), e estes possuem a capacidade de “conversarem” no sentido de trocarem dados ou informações. Neste contexto, a interoperabilidade semântica significa ter um ponto único de referência no nível de dados, sendo através de grafo de dados, e da representação dos dados a nível de ontologias. Esta é uma questão que pode ser resolvida entre os diferentes modelos de ontologias existentes, ou através da geração de uma específica, fundindo e mapeando os termos e conceitos envolvidos, podendo haver também protocolos ou

modelos externos sendo utilizadas nesta ontologia específica (SERRANO et al., 2015).

Um trabalho inicial na definição de interfaces e descrições comuns para dados relacionados à IoT é fornecido pelo grupo Sensor Web Enablement (SWE) da OGC. As principais especificações definidas pela OGC são a de Observation & Measurement (O&M), que define um modelo para codificar observações e medições de dados de sensores e o Sensor Model Language (SensorML), que é um modelo para descrever sistemas e processos de sensores. Ambos oferecem também um schema XML para estruturação dos dados (BOTTS et al., 2006; BARNAGHI et al., 2012; DESAI, SHETH, ANANTHARAM, 2015).

A OGC ainda define o padrão Sensor Observation Service (SOS) que é um modelo de serviços que fornece uma interface web padrão para solicitar, filtrar e recuperar observações e informações de sistema de sensores (BOTTS et al., 2006; BARNAGHI et al., 2012; DESAI, SHETH, ANANTHARAM, 2015).

Henson et al. (2009) apresenta o SemSOS, o qual demonstra um modelo que expande o padrão SOS da OGC e visa fornecer a semântica em alto nível de abstração para anotação de dados de sensores, visando apoiar na formação de uma base de conhecimento. O SemSOS é um componente fundamental na proposta da Semantic Sensor Web proposta por Desai, Shet, Anantharam (2015), que é apresentado no próximo tópico 2.3.2;

Para Barnaghi et al. (2012) os modelos e interfaces fornecidas pela OGC definem um framework para trabalhar com sensores em ambientes heterogêneos, porém, a representação destes modelos da SWE estão codificados em XML e isto gera uma limitação significativa na definição e associação dos elementos para promover a interoperabilidade semântica. Ainda, segundo a W3C, esses padrões não estão integrados e alinhados com as recomendações de padrões W3C Semantic Web, que são os principais impulsionadores para a criação e manutenção de um grafo global e densamente interconectado de dados.

Contudo, a W3C em conjunto com OGC, desenvolveu as ontologias chamadas de Sensor, Observation, Sample, and Actuator (SOSA) e Semantic Sensor Network (SSN), para modelagem de dispositivos e plataformas de sensores, abrangendo os conceitos de ambientes e observações, dando mais um passo na direção de alcançar a interoperabilidade entre ambientes de IoT (GYRARD; BONNET; BOUDAUD, 2014; DESAI, SHETH, ANANTHARAM, 2015). Sendo

que o SOSA funciona um bloco de construção central para o SSN, e este expande o escopo original do SSO, proposto pela OGC, para além de sensores e suas observações, incluindo classes e propriedades para atuadores e amostras (W3C SSN, 2017).

A SSN é uma ontologia para descrever os sensores e suas observações, os procedimentos envolvidos, as características de interesse estudadas, as amostras usadas para fazê-lo e as propriedades observadas, bem como os atuadores. O SSN segue uma arquitetura de modularização horizontal e vertical, incluindo uma ontologia central leve, mas auto-suficiente, chamada SOSA (Sensor, Observation, Sample e Actuator) para suas classes e propriedades elementares. Com seu escopo com diferentes axiomas, a SSN e SOSA são capazes de suportar uma ampla gama de aplicações e casos de uso, incluindo imagens de satélite, monitoramento científico em larga escala, infraestruturas industriais e domésticas, sensoriamento social, ciência social, engenharia de ontologias orientada a observação e a Web of Things (W3C SSN, 2017).

A W3C SSN (2017) aborda uma perspectiva de sensores, com foco no que sente, como sente e o que é sentido, através de três perspectivas, sendo elas:

- Observação: com foco em dados de observação e metadados relacionados;
- Sistema: aborda sistemas de sensores e implantações;
- Características e propriedades: contempla o que sente uma propriedade em específico ou quais observações tem sido feitas sobre uma propriedade.

Na Figura 15 é apresentada uma visão geral das principais classes e propriedades que estão especificamente relacionadas às observações de modelagem. Os axiomas da SOSA são mostrados em verde, enquanto os axiomas da SSN são mostrados em azul (W3C SSN, 2017).

Na seção a seguir são tratados os assuntos da web semântica que foram encontrados e estão relacionados com a área da IoT.

2.3.2 Web semântica aplicada na IoT

Existe um consenso de que a web semântica é a ferramenta apropriada para abordar a diversidade da Internet das Coisas (TOMA; SIMPERL; HENCH, 2009; HACHEM; TEIXEIRA; ISSARNY, 2011; BARNAGHI et al., 2012; SZILAGYI; WIRA, 2016).

A IoT pode se beneficiar da web semântica e uso de ontologias para questões de interoperabilidade, armazenamento, integração, acesso e abstração de dados, resolução e interpretação semântica, a busca e descoberta de recursos e escalabilidade (DESAI; SHETH; ANANTHARAM, 2015; SRIRAM; SHETH, 2015; SZILAGY; WIRA, 2016).

“A semântica formal permite a gestão do conhecimento e a troca de dados de uma maneira interpretável por máquinas. Isso torna as tecnologias semânticas uma chave para superar problemas comuns de modelagem, troca de modelos e interoperabilidade que precisam ser resolvidos ao longo do ciclo de vida destes sistemas da IoT” (SZILAGYI; WIRA, 2016, p. 1)

Um problema importante é a heterogeneidade inerente em sistemas de computação ubíquos, como ambientes de IoT, devido à natureza dos componentes conectados, dos diferentes padrões, formatos de dados e protocolos envolvidos, o que cria obstáculos significativos para a interoperabilidade entre seus componentes (KATAZONOV et al., 2008).

Segundo Barnaghi et al. (2012) o fornecimento de interoperabilidade entre as “Coisas” na IoT é um dos requisitos mais fundamentais para suportar o endereçamento, rastreamento e descoberta de objetos, bem como a representação, armazenamento e troca de dados e informações. E atentam sobre a necessidade de compartilhar e integrar informações em diferentes domínios para inferir novos conhecimentos.

Em IoT, dados binários e modelos de dados sintáticos oferecem interoperabilidade limitada em níveis de descrição de dados e recursos. Ao utilizar anotações semânticas e adicionar descrições de metadados em diferentes partes de um sistema de IoT, garante que os dados originados de diversos recursos e em formas heterogêneas possam se tornar

acessíveis e processáveis por vários domínios e usuários (SERRANO et al., 2015). Contudo, as tecnologias semânticas são reivindicadas como uma abordagem qualitativamente mais forte de interoperabilidade do que as abordagens de padrões contemporâneos (LASSILA, 2005).

Contudo, para ajudar a resolver problemas de interoperabilidade entre sistemas de IoT, causados pela natureza heterogênea e distribuída das “Coisas”, a comunidade de IoT passou a adotar tecnologias da web semântica para esse objetivo, e com isso vários formatos e modelos de ontologias para anotar e descrever os dados da IoT foram desenvolvidas (BARNAGHI et al., 2012; GYRARD; BONNET; BOUDAUD, 2014).

Através de descrições e anotações semânticas é possível representar dispositivos, objetos e eventos do mundo real, modelos de serviços e processos de negócios. Estas dão suporte ao gerenciamento automatizado e a interação entre os diferentes componentes dos sistemas de IoT (BARNAGHI et al., 2012; GYRARD; BONNET; BOUDAUD, 2014; SZILAGY; WIRA, 2016).

De acordo com Szilagy e Wira (2016) são dois tipos principais de aplicações da web semântica que podem auxiliar nos ambientes de IoT, sendo através da representação e descrição semântica dos dispositivos, de suas capacidades e ambiente; e da anotação semântica dos dados produzidos por estes dispositivos.

Serviços baseados em semântica são necessários na IoT para oferecer suporte ao provisionamento de serviços autônomos. Os mecanismos para implementar serviços baseados em semântica incluem anotação semântica de serviço, acesso semântico de serviço e troca semântica entre serviços. NOTA - A anotação semântica do serviço pode permitir a descrição do mesmo. E o acesso semântico aos serviços pode ser usado para acessar outros serviços por meio de interfaces semânticas. A troca semântica entre os serviços pode permitir o fornecimento e a troca de semântica entre os serviços para suportar a criação automática de novos serviços (ITU-T Y.4100/Y.2066, p. 9).

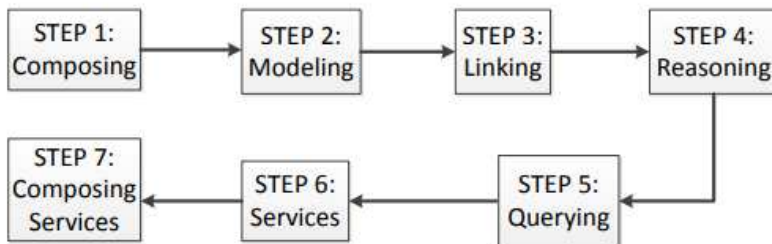
Sheth et al. (2008) foram os primeiros a integrar a semântica em redes de sensores e o chamaram de "Semantic Sensor Web", neste estudo eles demonstram um cenário com dados de sensores sendo coletados e através da anotação semântica com OWL e RDF, reuso e integração de

modelos de ontologias de Sensores, Geolocalização, Temporalidade e Clima. E realizam os processos de anotação semântica em dados coletados de sensores, vinculam estes com ontologias de outros domínios, e aplicam raciocínio baseado em regras através da especificação proposta pela W3C, Semantic Web Rule Language (SWRL), para inferir novos conhecimentos ao cenário de estudo (SHETH et al., 2008; GYRARD; BONNET; BOUDAUD, 2014).

Os autores Pfisterer et al. (2011), através do projeto SPITFIRE introduzem o conceito da Web Semântica das Coisas (SwoT) e apresentam uma arquitetura com base principalmente no modelo Semantic Sensor Web de Sheth et al. (2008).

Segundo Gyrard, Bonnet e Boudaoud (2014, p. 1) “Semantic Web of Things (SWoT) é um novo campo que combina a Internet das Coisas e tecnologias da web semântica”. Gyrard e Serrano (2016) propõe uma metodologia para SWoT, chamada de SEG 3.0⁵, a qual é baseada em metodologias da engenharia da ontologia, principalmente na metodologia NeON, e tem por objetivo estabelecer as etapas necessárias para alcançar a interoperabilidade semântica e permitir a recuperação de conhecimentos, conforme apresentado na Figura 16.

Figura 16 – Framework conceitual SEG 3.0



Fonte: Gyrard e Serrano (2014, p. 3)

O detalhamento de cada passo proposto por Gyrard e Serrano (2016, p. 3-4) na metodologia SEG 3.0 são descritos a seguir:

- 1) A composição permite unificar dados heterogêneos provenientes de diferentes projetos e usar diferentes formatos de dados (por exemplo, CSV, Excel) ou termos

⁵ Disponível em: <<https://fiesta-iot-tools.appspot.com/>>. Acesso em: novembro de 2018.

diferentes (por exemplo, temperatura ou temp). Requerendo um dicionário comum para unificar termos empregados para descrever dados.

- 2) A modelagem permite anotar semanticamente os dados com tecnologias da Web semântica (por exemplo, RDF, RDFS e OWL). Esta etapa emprega modelos, vocabulários e ontologias para unificar os dados, uma etapa obrigatória para os processos seguintes.
- 3) Vinculação permite enriquecer os dados com outros conjuntos de dados RDF para obter informações adicionais. Ele explora a ideia de Linked Data e Linked Vocabularies.
- 4) O raciocínio permite atualizar o banco de dados e armazenar triplas com informações adicionais, por exemplo, usando o mecanismo de raciocínio, por exemplo, mecanismo de inferência baseado em regras. Esta etapa promove abstração de alto nível dos dados do sensor e explora a ideia de regras vinculadas.
- 5) A consulta permite consultar os conjuntos de dados RDF através da linguagem SPARQL com base nas ontologias usadas nas etapas anteriores. É uma etapa essencial para obter dados e criar serviços e aplicativos para usuários finais.
- 6) Os serviços permitem fornecer acesso a dados mais inteligentes para os usuários finais. Os dados estão disponíveis por meio de APIs interoperáveis ou serviços da Web (por exemplo, serviços da RESTful). Esses serviços da web retornam o resultado fornecido pelo mecanismo de consulta SPARQL.
- 7) A composição dos serviços permite a construção de aplicações complexas, compondo vários serviços juntos. Isso pode ser alcançado através do uso de serviços web ou serviços da web semânticos.

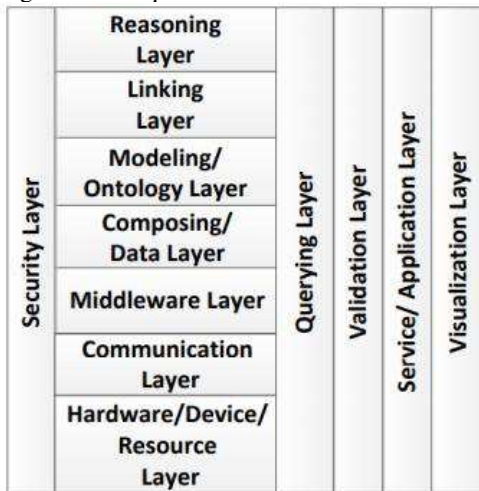
Uma arquitetura de aplicação IoT que baseia-se na metodologia SEG 3.0 é apresentada por Gyrard e Serrano (2016), a qual é separada em doze camadas, sendo estas detalhadas a seguir:

- Camada de Hardware: é específica para IoT e Smart Cities para obter dados produzidos pelos dispositivos.
- Camada de Comunicação: envia dados para a Web para acessar facilmente dados dos dispositivos.

- Camada Middleware: harmoniza plataformas existentes
- Camada de Dados: unifica dados provenientes de fontes e projetos heterogêneos
- Camada de Ontologias: modelos de dados em um maneira unificada
- Camada de Ligação: enriquece dados com outros conjuntos de dados
- Camada de Raciocínio: deduz conhecimentos de alto nível a partir dos dados.
- Camada de Segurança: abrange todas as demais camadas
- Camada de Consulta: para selecionar um subconjunto de dados ou ontologias específicas, regras, etc.
- Camada de Validação: valida as etapas anteriores para verificar se a interoperabilidade é garantida.
- Camada de Serviços: fornece serviços interoperáveis para facilitar a composição dos mesmos
- A Camada de Visualização: fornece interfaces de usuário amigáveis, interoperáveis e reutilizáveis para exibir dados selecionados e enriquecidos.

A arquitetura comentada na lista acima é demonstrada na Figura 17.

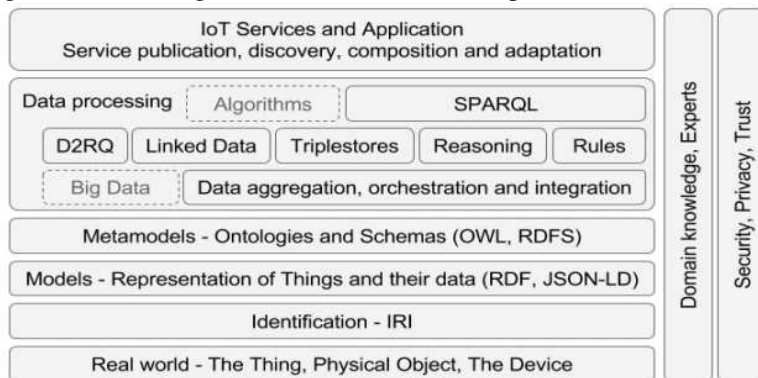
Figura 17 - Arquitetura de SWoT baseada na metodologia SEG 3.0



Fonte: Gyrard e Serrano (2016, p. 4)

Szilagyi e Wira (2016, p. 6) demonstram uma visão das tecnologias de web semântica separadas pelos diferentes aspectos da IoT, baseando-se nas visões apresentadas nos estudos de Vermesan et al. (2013) e BARNAGHI et al. (2012), e apresentada através da Figura 18.

Figura 18 - Tecnologias atuais da web semântica para IoT



Fonte: Szilagyi e Wira (2016, p. 6)

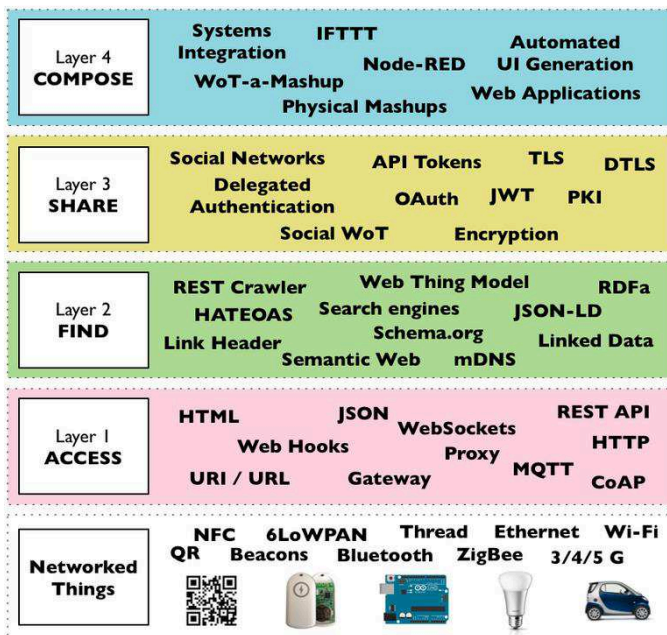
Quanto as tecnologias apresentadas por Guinard e Trifa (2016) incluem uma visão da web semântica e apresentam uma arquitetura separada em quatro camadas, sendo elas: Acesso, Busca, Compartilhamento e Composição. Sendo que para cada camada são atribuídas as principais tecnologias relacionadas ao contexto da WoT, por exemplo, a camada de acesso cria uma API para que a camada dispositivos possa conectar e enviar dados, enquanto a camada de busca utiliza a API da camada de acesso para tornar os serviços disponíveis e localizáveis na web. Esta visão é apresentada na Figura 19.

Buckley (2006), Katasonov (2008) e Hribernik et al. (2011) consideram uma visão da aplicação da web semântica em ambientes de IoT, através da implementação de middlewares que funcionam como componentes que são acomodados nas arquiteturas de IoT. Buckley (2006) comenta que para interoperabilidade requer a existência de algum middleware para atuar como a cola que une componentes heterogêneos juntos, oferecendo interfaces de programação, comunicações e outros serviços às aplicações, simplificando a criação de aplicativos e mudando de abordagens estáticas de programação para um padrão configurável e com capacidade de composição dinâmica dos componentes envolvidos.

“A Web of Things está essencialmente ligada ao papel das tecnologias da Web em facilitar o uso de aplicativos e serviços para a IoT, ou seja, objetos físicos e sua representação virtual. Isso inclui sensores e atuadores, bem como objetos físicos marcado com um código de barras ou Near Field Communication (NFC). Algumas tecnologias relevantes da Web incluem HTTP para acessar Serviços RESTful e nomeação de objetos como base para dados vinculados e descrições ricas, e APIs JavaScript para objetos virtuais que atuam como proxies para objetos reais.” (MINERVA; BIRU; ROTONDI, 2015, p. 21)

O termo middleware também é utilizado no trabalho de Gyrard e Serrano (2016), porém, se difere de Buclkey (2006), Katasonov (2008) e Hribernik et al. (2011), por dividir a atribuição do middleware com outras camadas.

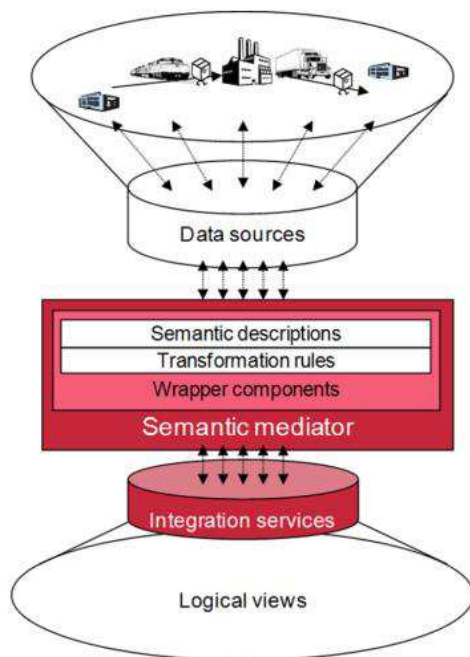
Figura 19 - Tecnologias da web mais atuais no contexto da WoT



Fonte: Guinard e Trifa (2016, p. 5)

No entanto, Hribernik et al. (2011) apresenta uma arquitetura orientada a serviços através de uma visão que incorpora um mediador semântico baseado em ontologias, conforme demonstrado na Figura 20. E também comenta que “um mediador semântico deve ser capaz de extrair conhecimentos sobre as estruturas de dados de fontes subjacentes e, posteriormente, transformar, decompor e recompor em solicitações de dados de acordo com estes conhecimentos” (HRIBERNIK et al., 2011, p. 150).

Figura 20 – Arquitetura SOA aplicada como middleware semântico



Fonte: HRIBERNIK et al. (2011, p. 150).

Vale ressaltar o framework denominado CityPulse, proposto por Puiu et al. (2016), que aborda o suporte à criação de serviços para cidades inteligentes por meio de um sistema distribuído que envolve descoberta semântica, análise e interpretação de dados em grande escala e em tempo real, para dados da IoT e de mídias sociais. Segundo os autores, o objetivo da proposta é permitir a integração de dados entre domínios, integrando dados multimodais, de qualidade mista, incertos e

incompletos para criar informações confiáveis e continuamente adaptá-los com as técnicas de processamento de dados para atender à qualidade dos requisitos de informações dos usuários finais.

No trabalho de Hribernick et al. (2011) uma arquitetura orientada a serviços é sugerida, identificando uma relação com as especificações dadas no contexto da WoT, uma vez que esta determina utilização do protocolo RESTful sob o protocolo HTTP. Porém, no modelo de Hribernick observa-se uma ótica mais genérica, que considera diversos protocolos de comunicação, além do que é sugerido no contexto da WoT apresentado por Guinard e Trifa (2009) e Guinard e Trifa (2016), que focam somente no uso de tecnologias da web.

Quanto aos modelos de Spitfire a metodologia SEG 3.0, estes possuem convergência, sendo que o segundo foi apoiado pelos resultados do primeiro trabalho.

Segundo Serrano et al. (2015, p. 31) os principais requisitos para implementar a web semântica são:

- Integração: Suporte a múltiplos dispositivos (sensores e atuadores) e tipos de fontes de dados relevantes (independentemente de fabricante e localização).
 - Ativar o compartilhamento escalável e integração de fontes de dados distribuídas.
 - Todos os aplicativos de IoT envolvem vários dispositivos heterogêneos. Orquestrar os dispositivos para formular automaticamente fluxos de trabalho compostos, conforme exigido pelos aplicativos do usuário final.
- Anotação: Possibilitar a vinculação (automatizada) de fontes de dados relevantes.
 - A vinculação de fontes de dados facilita a integração de aplicativos e a reutilização de dados.
 - Permitir interações entre dispositivos e serviços da IoT.
 - Construído sobre os padrões (ou seja, ontologia padrão W3C SSN) para descrição de sensores e demais dispositivos.
- Gestão: Permitir a criação e o gerenciamento de sensores virtuais e de ICOs virtuais com

base na composição e fusão de fluxos provenientes de várias fontes de dados (ICO).

- Desenvolvimento e integração de aplicativos envolve múltiplas fontes de dados distribuídas e heterogêneas para serem processadas em paralelo.
- A definição e o gerenciamento de sensores virtuais facilitam a integração de aplicativos.
- **Descoberta:** Fornecer os meios para descobrir e selecionar ICOs e fontes de dados referentes a solicitações de aplicativos (de acordo com suas capacidades)
 - Os usuários finais precisam de uma interface de alto nível para ser acessada.
 - Fornecer os meios para descrever e formular serviços e aplicativos IoT de acordo com as descrições de alto nível.
 - Fornecer recursos de visualização (configuráveis) de várias fontes de dados integradas (em um formato de mashup).
- **Análise e Raciocínio:** Fornecer ferramentas analíticas e de raciocínio além dos recursos de nível semântico.
 - A IoT aborda ambientes de grande escala com inúmeras ICOs com diferentes funcionalidades e recursos.
 - As aplicações do usuário final envolvem o monitoramento de sensores virtuais e / ou físicos
- **Visualização:** Otimizar o uso de recursos (armazenamento, ciclo de computação, sensor utilização) em vários usuários que compartilham esses recursos.
 - Vários aplicativos envolvem interações objeto-a-objeto (por exemplo, M2M) ou interações entre serviços; tais interações podem ser definidas explicitamente (ou seja, por usuários finais) ou derivadas implicitamente (com base no contexto da aplicação).

Contudo Serrano et al. (2015) ainda comenta que os principais requisitos relativos a dados na IoT, são:

- Representação estruturada dos conceitos identificados.
- Mecanismos de raciocínio
- Descrições interpretáveis por máquinas
- Mecanismos de acesso as descrições dos recursos
- Interações automatizadas e integração horizontal com aplicações existentes

No entanto, a listagem dos requisitos da Serrano et al. (2015) e a visão apresentada por ele em relação a semântica e interoperabilidade de dados apoiam na fundamentação do modelo proposto, apresentado no Capítulo 4 .

3 METODOLOGIA

Este capítulo apresenta a metodologia de pesquisa utilizada para execução deste trabalho, sendo descrita a caracterização da pesquisa, seguido da apresentação dos procedimentos metodológicos separados e detalhados em etapas e atividades necessárias para execução desta pesquisa.

3.1 CARACTERIZAÇÃO DA PESQUISA

A presente pesquisa enquadra-se no paradigma funcionalista de Morgan e Smirich (1980) onde predomina neste paradigma uma concepção sistêmica e sincrônica do mundo social orientada pra consecução de objetivos.

Segundo Silva e Menezes (2005) é possível categorizar a pesquisa quanto aos seus objetivos, natureza, abordagem do problema e em relação aos procedimentos técnicos utilizados, conforme nos tópicos a seguir:

- Quanto a natureza esta pesquisa pode ser considerada pesquisa aplicada, pois objetiva gerar conhecimentos de aplicação prática, propondo avanços no conhecimento científico tanto na área de gestão e engenharia do conhecimento, mas também causando o avanço tecnológico ao citar ou utilizar tecnologias existentes na proposição e verificação de um modelo baseado em conhecimento para ambiente de Internet das Coisas.
- Sobre a abordagem do problema está pode ser classificada como qualitativa pois considera que há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito. A interpretação dos fenômenos e a atribuição de significados são básicas no processo de pesquisa qualitativa, onde o processo e seu significado são os focos principais da abordagem.
- Referente aos objetivos pode ser classificada como pesquisa exploratória pois visa proporcionar maior familiaridade com o problema no intuito de torná-lo explícito ou a construir hipóteses sob, inclusive envolvendo levantamento bibliográfico e estudo de caso. Também se classifica como pesquisa descritiva pois descreve as características de

determinado fenômeno, estabelecendo relações entre as variáveis envolvidas.

- Quanto aos procedimentos técnicos da pesquisa, é considerada pesquisa bibliográfica pois foi elaborada a partir de material já publicado, constituído principalmente de livros e artigos de periódicos e pesquisa documental no sentido de utilizar materiais encontrados na web que não receberam tratamento analítico prévio.

A seguir são apresentados os procedimentos metodológicos desta pesquisa.

3.2 PROCEDIMENTOS METODOLÓGICOS

Na execução desta pesquisa foi utilizada a metodologia de ciência do projeto ou do artificial, conhecida como Design Science Research Methodology (DSRM). As ciências do artificial tratam de como as coisas devem ser para funcionar e atingir determinados objetivos, sendo utilizadas para criar e projetar artefatos que tenham as propriedades desejadas para alcançar estes objetivos (DRESCH; LACERDA; JÚNIOR, 2015).

Contudo, os objetivos da Design Science são divididos em projetar, criar e avaliar artefatos de tecnologia da informação que tem por objetivo a resolução de problemas reais, onde através de um processo bem definido resolve problemas observados e então comunica os resultados ao público interessado (PEFFERS et al., 2007).

Para March e Smith (1995, p. 253) “o Design Research tenta criar coisas que servem a propósitos humanos e é orientado para a tecnologia e seus produtos são avaliados de acordo com critérios de valor ou de utilidade”. Segundo a abordagem de Dresch, Lacerda e Júnior (2015) apud Van Aken (2004) corrobora com essa afirmação, uma vez que para ele o principal objetivo da Design Science é desenvolver conhecimento para a geração e desenvolvimento de artefatos.

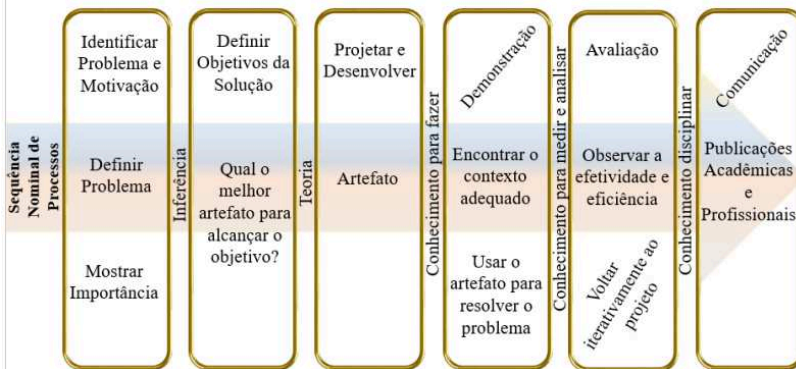
Segundo Peffers et al. (2007) são seis as etapas do processo de metodologia DSRM. A seguir são detalhadas estas etapas, de acordo com Ribeiro (2018, p. 81-82) apud Peffers et al. (2007):

1. Identificar o problema e sua motivação: definição do problema de pesquisa específico e justifica-se sua solução;
2. Definir os objetivos da solução: definem-se objetivos da solução que foi proposta;
3. Projetar e Desenvolver: aqui cria-se o artefato, que segundo March e Smith (1995) pode ser um: constructo ou conceitos que compõem o vocabulário de um domínio, e constituem uma conceituação utilizada para descrever os problemas dentro do domínio e para especificar as respectivas soluções; modelo que é um conjunto de proposições ou declarações que demonstram o relacionamento entre os constructos, podem ser visualizados como uma representação de como as coisas são, e nas atividades de design representam situações como problema e solução; método que é um conjunto de passos, sendo um algoritmo ou orientação usado para executar uma tarefa e por fim instanciação que é concretização de um artefato em seu ambiente de modo que operacionalizam constructos, modelos e métodos. Nesta etapa de projetar e desenvolver artefatos, abordagens como algoritmos computacionais, representações gráficas, protótipos, maquetes em escala, entre outros, podem ser utilizadas;
4. Demonstrar: etapa responsável pela demonstração de uso do artefato, resolvendo um ou mais aspectos do problema por meio de um experimento, simulação, estudo de caso, prova formal entre outras atividades apropriadas;
5. Avaliar: nesta etapa, é feita observação e mensuração de como o artefato atende à solução do problema, fazendo comparações a partir de métricas e técnicas de análises, dos objetivos propostos com os resultados observados na utilização do artefato. Ainda nesta etapa, segundo Lacerda et al. (2013) deve ser definido um processo de verificação do comportamento do artefato, sendo necessário:
 - a) explicitar o ambiente interno, externo e os objetivos;
 - b) explicitar como o artefato pode ser testado; e

- c) descrever os procedimentos que medem os resultados;
6. Comunicar: quando apropriado, faz-se a divulgação, para outros pesquisadores e outras audiências, do problema e sua relevância, do artefato que foi concebido, da sua utilidade e seu ineditismo, da efetividade e rigor do projeto.

As seis etapas comentadas estão apresentadas na Figura 21.

Figura 21 - Etapas DSRM



Fonte: Ribeiro (2018) com base em Peffers et al. (2007)

Quanto aos artefatos que podem ser projetados, são classificados em quatro tipos segundo March e Smith (1995), sendo eles:

- **Constructos:** Os constructos ou conceitos, formam o vocabulário de um domínio. Eles permitem uma conceituação que pode ser usada para descrever problemas de domínio e para especificar as respectivas soluções.
- **Modelos:** Um modelo é um conjunto de propostas ou declarações que expressam as relações entre os conceitos relacionados e pode ser visto como uma descrição ou representação de como as coisas são em situações de problema e solução.
- **Métodos:** Um método é um conjunto de passos (um algoritmo ou orientação) usado para executar uma tarefa. Os métodos podem ser ligados aos modelos, nos quais as etapas do método podem utilizar partes do modelo como uma entrada que o compõe. Os métodos são, muitas vezes,

utilizados para traduzir um modelo ou representação em um curso para resolução de um problema.

- Instanciações: Uma instanciação é a concretização de um artefato em seu ambiente. Instanciações operacionalizam constructos, modelos e métodos, podendo demonstrar a viabilidade e a eficácia dos modelos e métodos que estão sendo contemplados.

Apesar de ter uma etapa de avaliação final do artefato não é dispensado que em cada etapa do da DSRM, sejam realizadas avaliações parciais dos resultados. Isso se faz necessário, para se certificar de que a pesquisa está no sentido dos objetivos propostos (DRESCH; LACERDA; JÚNIOR, 2015).

3.3 ETAPAS E ATIVIDADES DA PESQUISA

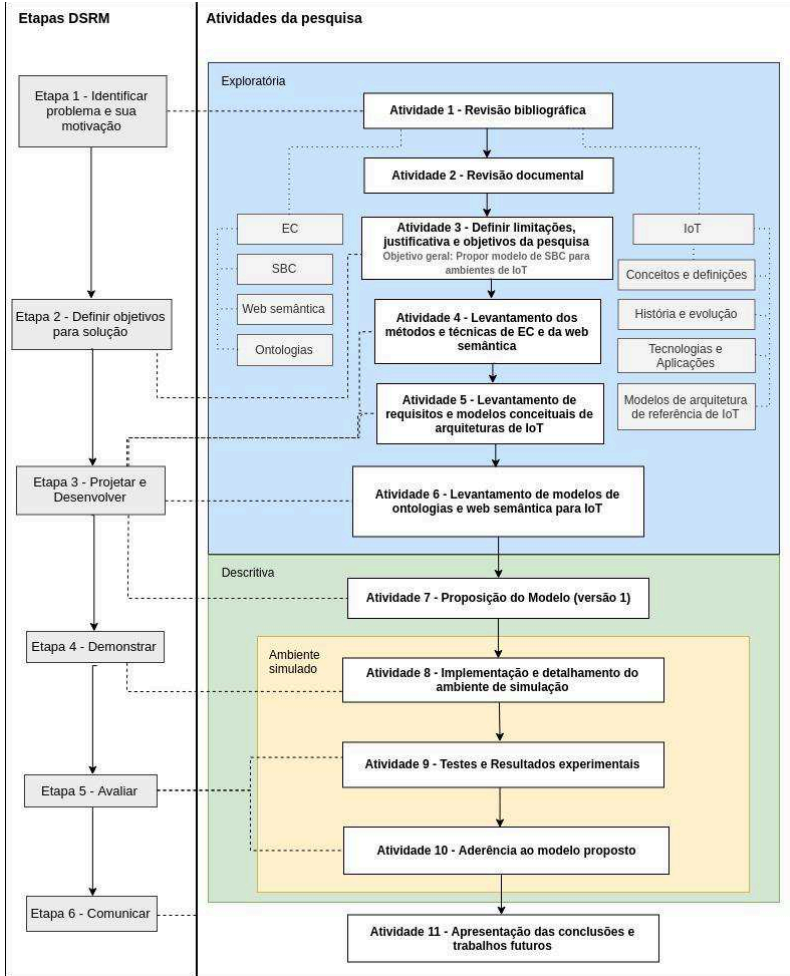
Seguindo as etapas da metodologia DSRM, foram definidas e relacionadas as atividades necessárias para a execução de cada etapa desta pesquisa, conforme é apresentado na Figura 22.

As etapas estão sequencialmente interligadas e relacionadas, algumas com uma ou mais atividades, das quais foram necessárias serem executadas para o cumprimento de cada etapa da pesquisa. Sendo que as Atividade 1 e 2, estão no contexto da Etapa 1, a atividade 3 está ligada a Etapa 2, as Atividade 4 a 7 estão vinculadas à Etapa 3, sobre a Atividade 8 é executada na Etapa 4. Quanto as Atividades 9 e 10 estas estão inseridas no contexto da Etapa 5. Na Etapa 6 está ligada à atividade 11.

As atividades da pesquisa estão apresentadas em formato sequencial, sendo que o processo se inicia pelo objetivo geral proposto e passa por todas as atividades necessárias para atingir este objetivo. Também são apontados os principais vínculos entre as atividades e assuntos abordados (itens na cor cinza, ligados na atividade 1).

As linhas pontilhadas ligam os conceitos envolvidos, abordando tanto atividades quanto os assuntos pesquisados na revisão bibliográfica. Também foi realizado um agrupamento conceitual das atividades relacionadas a parte exploratória (caixa azul) e a parte descritiva (caixa verde) contemplada neste estudo.

Figura 22 – Detalhamento da metodologia de pesquisa



Fonte: O autor (2019)

Nas seções abaixo são detalhadas cada etapa e suas principais atividades e tarefas realizadas em cada uma delas.

3.3.1 Etapa 1 – Identificar o problema e sua motivação

Nesta etapa foram realizadas as atividades de revisão bibliográfica e revisão documental, sendo levantado o material que apoiou na

identificação da motivação (Capítulo 1) e na problematização e questão da pesquisa, expostos na seção 1.1.

E principalmente as atividades desta etapa foram essenciais na formação de uma base teórica para realização deste estudo, possibilitando a elaboração do Capítulo 2 e fornecendo a fundamentação necessária para a execução das demais etapas desta pesquisa.

A seguir o detalhamento das duas atividades executadas nesta etapa.

3.3.1.1 Atividade 1 – Revisão bibliográfica

Foram realizadas buscas nas bases Scopus, Web of Science e IEEE pelo conjunto de termos em inglês: “Iot” e “KBS”, “IoT e “KE”, “IoT” e “Knowledge” e “IoT”, “Knowledge” e “Engineering”. Além destas buscas, também foram consultados o termos: “IoT” e “Architechure”, “IoT” e “Ontolog*” e por fim, “IoT”, “Semantic” e “Web. Em todas estas buscas comentadas foram aplicados filtro por título do artigo.

Contudo, para aprofundamento nos assuntos relacionados a EC, SC, Ontologias e Web Semântica, também foram realizadas buscas na base de teses e dissertações do EGC, separadamente para cada termo. Os estudos selecionados da base do EGC estão apresentados na seção 1.5.

Com o conteúdo levantado nas buscas realizadas foi possível selecionar o material a partir da eliminação de duplicidades e a leitura dos resumos, utilizando como critério de seleção de análise as pesquisas que demonstraram abordagens relacionadas a modelos, arquiteturas, frameworks, técnicas e tecnologias relacionadas aos constructos desta dissertação: EC e IoT.

Vale ressaltar que também foram encontradas referências e citações de publicações da Mckley, Springer e Elsevier que possuem relação direta aos assuntos da pesquisa e que foram buscadas pontualmente e referenciadas neste trabalho. E também algumas referências a documentos e recomendações disponíveis na web, sendo neste caso, realizadas buscas e revisão do conteúdo, conforme detalhado no tópico a seguir.

3.3.1.2 Atividade 2 – Revisão documental

Foram encontrados documentos em sites de organizações públicas e privadas que possuem iniciativas de estudos na área de IoT, sendo as seguintes fontes utilizadas:

- ETSI: <https://www.etsi.org/technologies/internet-of-things>
- IERC: <http://www.internet-of-things-research.eu/>
- ITU-T: <https://www.itu.int/pub/T-REC>
- União Européia – programa H2020: <https://iot-epi.eu/>
- Wise-IoT (parceria do grupo H2020 e o governo da Coreia do Sul): <http://wise-iot.eu/en/deliverables/>
- FIESTA-IoT: <http://fiesta-iot.eu/>
- Intel:
<https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/iot-platform-reference-architecture-paper.pdf>
- Microsoft Azure IoT Reference Architecture (Versão 2.1):
<https://azure.microsoft.com/pt-br/blog/azure-iot-reference-architecture-2-1-release/>

Da listagem comentada, as duas últimas são iniciativas privadas e as demais são de instituições e grupos de pesquisas na área da IoT. Estes foram utilizados de forma a apoiar o embasamento da fundamentação teórica e do modelo proposto, sendo referenciadas nos demais capítulos deste trabalho.

3.3.2 Etapa 2 – Definir os objetivos da solução

Com a realização das atividades da etapa anterior, foi possível identificar vários aspectos sobre os assuntos abordados, bem como, os principais desafios a serem superados e as projeções futuras de pesquisas nas áreas envolvidas. Neste sentido, esta etapa contempla a execução da Atividade 3.

3.3.2.1 Atividade 3 – Definir limitações, justificativa e objetivos da pesquisa

A partir da identificação do problema e a motivação foi possível a definição dos objetivos da solução, sendo atribuído um objetivo geral e três específicos. Sendo que o cumprimento dos objetivos específicos

devem possibilitar alcançar ao objetivo geral determinado, estes estão expostos na seção 1.2.

Nesta etapa também foi determinada as delimitações e justificativa para realização desta pesquisa, sendo apresentadas respectivamente nas seções 1.3 e 1.4 deste trabalho.

3.3.3 Etapa 3 – Projetar e desenvolver

O artefato a ser projetado e desenvolvido trata-se de um modelo com abordagem de SC para ambientes de IoT, visando oferecer uma solução que apresente uma ótica voltada para a EC e que considere os requisitos inerentes a estes ambientes e a geração de uma base de conhecimento deste contexto. Todavia, o intuito é permitir que processos da EC como de aquisição, integração e recuperação de conhecimentos possam ser viabilizados nestes ambientes.

Nesta seção são apresentadas as principais atividades realizadas para a construção do modelo proposto, sendo elas de: levantamento de modelos, métodos e técnicas da EC e Web Semântica, levantamento de requisitos e modelos conceituais de arquiteturas de IoT, levantamento de modelos de ontologias e da web semântica para IoT e, por fim, a de proposição do modelo. Sendo estas atividades detalhadas nos tópicos a seguir desta seção.

3.3.3.1 Atividade 4 – Levantamento de modelos, métodos e técnicas da EC e web semântica

Foi realizado um levantamento dos principais métodos, técnicas e ferramentas da EC, utilizados no desenvolvimento de um SC. E a seleção das tecnologias a serem aplicadas e consideradas no ambiente de simulação foi voltada ao objetivo de atender principalmente ao requisito de interoperabilidade que é inerente em soluções de IoT e a geração de uma base de conhecimento.

A partir da revisão bibliográfica e documental realizada, foram analisadas e selecionadas as tecnologias que possibilitam a interoperabilidade e integração e reuso do conhecimento. Os resultados desta atividade estão descritos na seção 2.1.

3.3.3.2 Atividade 5 – Levantamento de requisitos e modelos conceituais de arquiteturas de IoT

A partir da pesquisa bibliográfica e documental realizada foram identificadas as principais arquiteturas de referência para IoT do cenário mundial, desenvolvidas por grandes organizações privadas e governamentais.

A abordagem de seleção das arquiteturas IoT a serem consideradas nesta etapa foram as que apresentam uma abordagem generalizada deste ambientes. Sendo possível durante a análise levantar os requisitos, modelos lógicos e de comunicação e os principais conceitos e tecnologias envolvidas, o que embasou a elaboração do modelo proposto. Os resultados desta atividade estão apresentado na seção 2.2.

3.3.3.3 Atividade 6 – Levantamento de modelos de ontologias e web semântica para IoT

Através da identificação das referências e iniciativas que correlacionam os constructos envolvidos, como modelos de ontologias e tecnologias da web semântica aplicados em ambientes de IoT, foi realizada a execução desta atividade. Tendo os resultados obtidos mostrados na seção 2.3.

Quanto ao levantamento e seleção dos modelos de ontologias a serem considerados na proposição do modelo foi com base no grau de generalização dos mesmos, ou seja, ontologias que servem no contexto geral de domínio da IoT. Portanto, ontologias relacionadas ao tema mas que abordam um contexto específico de aplicação da IoT, não foram analisadas detalhadamente.

No tocante a tecnologias da web semântica foram consideradas as essenciais para geração da proposta desejada e que se demonstram consolidadas, sendo encontradas a utilização no material selecionado e em recomendações reconhecidas da área. Neste contexto, também foi proposta a utilização da metodologia SEG 3.0 baseada na Engenharia de Ontologias (EO) que é considerada em pesquisas relacionadas a IoT.

3.3.3.4 Atividade 7 – Proposição do modelo

Esta atividade da pesquisa consiste no detalhamento da especificação e modelagem da proposta, considerando os requisitos funcionais e não-funcionais deste contexto e uma arquitetura em camadas

com detalhamento de suas interações e tecnologias envolvidas para seu funcionamento.

Para tal, foram utilizados métodos da Engenharia de Software (ES) como análise de requisitos e a linguagem UML para o desenvolvimento dos diagramas e modelos propostos, sendo possível apresentar em diferentes perspectivas os principais conceitos e fluxos de comunicação envolvidos na implementação de um SC para ambientes de IoT.

A atividade de proposição do modelo considera uma abordagem genérica de SC para ambientes de IoT, visando promover um ambiente favorável para que a Engenharia do Conhecimento possa ser trabalhada. Vale ressaltar que a execução desta atividade somente foi possível ser executada a partir dos resultados obtidos nas atividades anteriores.

Como resultado da execução desta atividade foi possível elaborar o capítulo 4, onde apresenta-se a especificação do modelo proposto de forma detalhada.

3.3.4 Etapa 4 – Demonstrar

Nesta etapa foi realizada a definição de um ambiente hipotético, onde ocorre a simulação de aquisição de dados de sensores de veículos, considerando na implementação as diretrizes impostas pelo modelo proposto, como os requisitos, fluxos de comunicação e a geração de uma base de conhecimento com estes dados. A Atividade 8 está envolvida nesta etapa e detalhada no tópico a seguir.

3.3.4.1 Atividade 8 – Implementação e detalhamento do ambiente de simulação

A implementação foi realizada por meio da definição do cenário envolvido no ambiente simulado. Contudo, algumas restrições foram necessárias serem aplicadas nesta atividade, no sentido de viabilizar uma versão de um ambiente de simulação para realização de testes, sendo elas:

- Ambiente simulado em estação local de trabalho
- Os dispositivos conectados foram simulados virtualmente, não sendo tratados neste estudo a utilização de dispositivos físicos e nem a implementação de software embarcado.
- Foram implementados requisitos essenciais para verificação principalmente do funcionamento do processo de anotação semântica e dos fluxos entre as camadas conceituais do

modelo proposto, para formação de uma base de conhecimento de grafos no formato de triplas RDF.

A seguir um detalhamento das principais tarefas envolvidas na implementação do ambiente simulado:

- Modelagem da arquitetura de implementação e comunicação entre as camadas, conforme os principais requisitos e fluxos expostos no modelo proposto, conforme demonstrada na introdução do capítulo 5 (ver Figura 29).
- Definição das tecnologias a serem utilizadas para criação do ambiente de simulação, conforme apresentado na seção 5.1.
- Desenvolvimento de agente simulador de dispositivos conectados: Aplicação desenvolvida em Node.JS para simulação de dispositivo conectado à camada de Apresentação (exposto na seção 5.2)
- Desenvolvimento de fluxos de comunicação propostos no modelo a partir da disponibilização de uma API que fornece serviços web via HTTP (exposto na seção 5.3).
- Implantação e disponibilização da base de grafos no formato de triplas RDF, conforme demonstrada na seção 5.4.

Na implementação do ambiente simulado foi utilizado o método de desenvolvimento iterativo-incremental. E para o detalhamento do modelo de implementação adotado neste ambiente, baseou-se na linguagem UML e na especificação dos serviços web implementados.

Na etapa a seguir apresentam-se os procedimentos realizados a avaliação do ambiente simulado desenvolvido nesta etapa.

3.3.5 Etapa 5 – Avaliar

Na ótica do modelo proposto, foi avaliada a aplicabilidade em decorrência da implementação do ambiente simulado, onde foram considerados conceitos, requisitos, fluxos de comunicação do modelo proposto. Sendo nesta etapa, realizadas as atividades de testes e resultados experimentais e a de aderência ao modelo conforme detalhadas nos tópicos a seguir.

3.3.5.1 Atividade 9 – Testes e resultados experimentais

Com a implementação do ambiente simulado foram realizados testes de requisições aos serviços web disponibilizados e a coleta dos resultados obtidos, contudo, os testes englobam procedimentos de anotação semântica e de publicação e recuperação de conhecimentos da base gerada.

As ferramentas utilizadas para realização das requisições comentadas são o software Postman⁶ e o navegador Google Chrome⁷. Um detalhamento dos resultados obtidos na realização desta atividade é exposto na seção 5.6.

3.3.5.2 Atividade 10 – Aderência ao modelo proposto

Nesta etapa foi possível observar a aderência dos artefatos e da especificação proposta no modelo em relação ao ambiente simulado implementado, bem como das tecnologias e processos adotados na construção deste ambiente.

Foram realizadas avaliações no sentido do funcionamento da arquitetura proposta como um todo, avaliando a implementação e interações em cada uma de suas camadas. E também foi realizada a avaliação no tocante a integração e recuperação de conhecimentos, através de consultas via SPARQL à base de conhecimento gerada na etapa de demonstração da pesquisa.

A verificação do modelo foi direcionada à camada de Conhecimento, onde foram executadas consultas nos dados armazenados na base de conhecimento gerada, a partir de consultas SPARQL. A verificação das demais camadas também foi realizada parcialmente, porém, estas já são separações conhecidas em diversos modelos publicados, conforme apresentado na fundamentação teórica, e por este motivo não foi o enfoque principal desta verificação.

⁶ Disponível em: <<https://www.getpostman.com/>>. Acesso em: outubro de 2018.

⁷ Disponível em: <<https://www.google.com/chrome/>>. Acesso em: outubro de 2018.

3.3.6 Etapa 6 – Comunicar

Esta etapa contempla a Atividade 11, tratando-se da apresentação de todo o conteúdo envolvido na pesquisa, sendo exposta por meio da elaboração desta dissertação e da publicação de artigos a serem realizadas em periódicos.

3.3.6.1 Atividade 11 – Apresentação das conclusões e trabalhos futuros

Como resultado principal da execução desta atividade realizou-se a descrição do capítulo 6, onde são sintetizadas as principais conclusões em relação aos objetivos impostos nesta pesquisa e é exposto um levantamento dos trabalhos futuros identificados em relação aos assuntos tratados no trabalho.

3.4 CONSIDERAÇÕES DO CAPÍTULO

Este capítulo apresenta as atividades necessárias para realização desta pesquisa, relacionando as atividades planejadas com as etapas da metodologia DSRM.

Contudo, devido a magnitude interdisciplinar dos temas envolvidos nesta pesquisa, considera-se necessária a utilização de métodos de ES, EC e EO para atingir aos objetivos propostos.

4 KE-IoT: UMA PROPOSTA DE MODELO BASEADO EM CONHECIMENTO PARA AMBIENTES DE INTERNET DAS COISAS (IoT)

O modelo KE-IoT (Knowledge Engineering to Internet of Things) visa especificar os principais aspectos que devem ser considerados no desenvolvimento de um SC, em uma abordagem de aplicação de ontologias e tecnologias da web semântica, para cenários de aplicação da IoT. A proposta fornece uma contextualização de um ambiente genérico de IoT e apresenta diretrizes para implementação.

O modelo segue as recomendações e conceitos da IoT estabelecidos pela ITU-T através das publicações Y.4000/Y.2060 (2012) e Y.4100/Y.2066 (2014). Além de princípios da WoT relativos a utilização de tecnologias e arquitetura orientada à serviços apresentados por Guinard e Trifa (2009), e da SWoT demonstrada por Gyrard, Bonnet e Boudaoud (2014), o qual tem enfoque na aplicação de semântica para ambientes de domínio da IoT. E também está apoiado nos projetos CityPulse proposto por Puiu et al. (2016), o SPITFIRE de Pfisterer et al. (2011) e Semantic Sensor Web de Sheth et al. (2008) que auxiliam no entendimento do funcionamento técnico e abordagem mais prática de aplicações na área da IoT.

Também são consideradas as visões de mediadores semânticos comentadas por Buckley (2006), Katasonov (2008) e Hribernik et al. (2011), no tocante ao processo de semântica ser aplicado como um componente que se acopla a uma arquitetura existente.

Sobre os requisitos são considerados os da recomendação Y.4100/Y.2066 (2014) da ITU-T e a listagem fornecida por Serrano et al. (2015), sendo que a primeira apoia com requisitos gerais de ambientes de IoT e a segunda aborda uma listagem relacionada aos dados e semântica envolvidos neste cenário.

As aplicações da IoT expostas por Manyika (2015) oferecem a visão dos possíveis cenários de utilização para o modelo. E no tocante as camadas conceituais e aspectos tecnológicos que circundam a IoT, como: protocolos, linguagens e formatos; foram consultados os modelos da IoT-A de Bassi et al. (2013), WSO2 proposto por Fremantle (2015) e a arquitetura de referência da Wise-IoT⁸. Além disso, também é

⁸ Disponível em: <<http://wise-iot.eu/wp-content/uploads/2018/09/D1.3-Wise-IoT-Updated-Architecture-PU-V1.0.pdf>>. Acessado em: outubro de 2018.

considerado o levantamento de tecnologias de Da Xu, He e Li (2014) e os documentos disponibilizados pelas organizações Intel e Microsoft para complementação e maior detalhamento de conceitos e tecnologias da IoT.

Esta proposta também se apoia em especificações da W3C, como a: Web semântica⁹, RDF, RDFs, OWL e também considera a utilização de um modelo de ontologia base para dados da IoT, sendo o padrão SOSA/SSN proposto pela W3C e OGC. Todavia, o processo que envolve a modelagem e anotação semântica é fundamentado nas etapas da metodologia SEG 3.0 de Gyrard e Serrano (2016) baseada na Engenharia de Ontologias e que visa fornecer a interoperabilidade semântica para ambientes de IoT e Cidades Inteligentes.

Contudo, a proposta reúne os principais elementos identificados de forma integrada, visando fornecer uma solução que promova a interoperabilidade semântica em ambientes de IoT e possibilite a representação dos conhecimentos envolvidos neste contexto.

A abordagem principal do modelo está em apresentar os principais componentes da IoT e as comunicações necessárias para formar uma base de conhecimento que permita um ambiente favorável para que EC possa ser trabalhada no contexto da IoT. E que a partir da base gerada seja possível, por exemplo, a inferência de novos conhecimentos e aplicação de métodos de resolução de problemas, aprendizagem de máquina, entre outras práticas da EC que podem atender tanto aos processos de aquisição, integração, reuso e descoberta de conhecimentos (SASIETA, BEPPLER, PACHECO, 2011; NAZÁRIO; DANTAS; TODESCO, 2014)

O detalhamento da especificação do modelo se inicia com a apresentação de uma visão separada em camadas, o que auxilia no agrupamento dos conceitos e principais assuntos que envolvem um ambiente de IoT.

Posteriormente, são apresentados os requisitos funcionais e não-funcionais do modelo, seguidos de um diagrama de domínio que apresenta as principais entidades envolvidas e suas respectivas relações.

Por fim, mostra-se um detalhamento com diagramas dos principais fluxos existentes entre as camadas, apresentando os sentidos em que os mesmos acontecem, bem como maiores detalhes técnicos para auxiliar na implementação deste modelo.

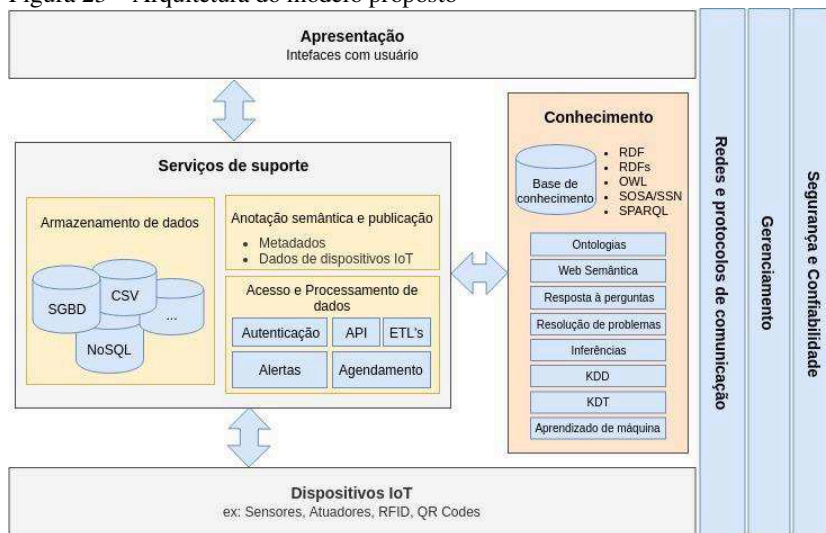
⁹ Disponível em: <<https://www.w3.org/standards/semanticweb/>>. Acessado em: novembro de 2018.

4.1 CAMADAS

O modelo possui uma arquitetura que é dividida em sete camadas lógicas, sendo que cada uma traz uma abordagem específica de um aspecto do modelo, na Figura 23 é possível visualizar um diagrama que demonstra estas camadas com suas respectivas características básicas e interações com as demais camadas.

As camadas de Segurança e Confiabilidade, Gerenciamento e Redes e protocolos de comunicação, são camadas verticais e seus recursos são necessários às demais camadas da arquitetura para o devido funcionamento e comunicação do ambiente como um todo.

Figura 23 – Arquitetura do modelo proposto



Fonte: O autor (2019)

Nos tópicos a seguir é apresentado um detalhamento dos principais aspectos envolvidos em cada camada que envolve a construção de um SC para ambientes de IoT, seguindo as diretrizes expostas por este modelo.

4.1.1 Dispositivo IoT

A camada de dispositivos da IoT engloba os objetos físicos do mundo real e as suas representações no mundo virtual. Existem diversos

tipos de dispositivos considerados dentro da IoT, com diferentes recursos e serviços, variando suas capacidades de comunicação, processamento e armazenamento de dados e informações. Basicamente é possível separar os dispositivos IoT em três tipos, sendo: Sensores, Atuadores e Tags de identificação.

Existem diferentes maneiras de interações e cenários de aplicações, onde dispositivos IoT funcionam independentes ou interligados, e trocam informações em rede entre eles ou com outros servidores.

A capacidade de comunicação com a rede pode ser realizada diretamente, quando o dispositivo possuir este recurso, ou indiretamente, quando o dispositivo depende de um gateway para realizar a comunicação. Esta interação pode ainda acontecer em dois sentidos: o dispositivo enviando informações pela rede; ou o dispositivo recebendo comandos ou informações via rede.

Dispositivos com a capacidade de gateway devem suportar vários tipos de conexões e protocolos de comunicação, bem como podem agregar diversos dispositivos e recursos. A comunicação pode acontecer por meio de redes com e sem fio, e utilizar diferentes protocolos, sendo necessário em alguns realizar conversões e autenticações para efetivar a comunicação.

Os dispositivos IoT que possuem a capacidade de gateway, geralmente necessitam de software embarcado que é responsável por realizar a implementação das comunicações de envio e recebimento de dados com o servidor.

Existe uma lista extensa de dispositivos de IoT aplicados a variados tipos de solução, como exemplo: Smart Cities, Smart TVs, Wearables (dispositivos vestíveis no corpo humano), Smartphones, Carros conectados, Câmeras inteligentes com IP, etc. Microcontroladores como Raspberry PY e Arduino são exemplos de dispositivos usados para prototipagem e até em produção, e possuem função de gateway com a possibilidade de interconectar sensores, atuadores e tags e comunicar os dados com a camada de Processamento e Armazenamento.

Não existem limitações quanto a tamanho, localização, formato ou origem dos dispositivos de IoT, alguns dispositivos podem ter o tamanho de um microchip, enquanto outros o tamanho de um veículo.

Neste modelo, a interação da camada de dispositivo IoT é efetivada através do envio e recebimento de dados por meio da camada de Serviços de suporte.

4.1.2 Segurança e Confiabilidade

A atribuição desta camada é garantir a Autenticação, Autorização e Auditoria (AAA), sendo que: a autenticação verifica a identidade digital dos usuários conectados, a autorização garante que um usuário autenticado somente tenha acesso aos recursos autorizados e, por fim, a auditoria refere-se a coleta de informações sobre o uso dos recursos de um sistema pelos usuários, também chamada de contabilidade.

Neste camada, questões de confidencialidade, proteção e validação de integridade, privacidade e antivírus também devem ser consideradas, de acordo com os requisitos das demais camadas e do domínio de aplicação em que a solução será implementada.

4.1.3 Gerenciamento

A camada de gerenciamento possibilita a gestão e monitoramento dos recursos envolvidos em todas as outras camadas, abrangendo a gestão de falhas, configuração, contabilidade, desempenho e de segurança. Por exemplo, para a camada de Dispositivos IoT esta camada pode permitir a ativação e desativação do equipamento, atualização de firmware, monitoramento de consumo de energia; na camada de Serviços de suporte, através o gerenciamento do uso de recursos de processamento, memória e rede dos servidores; ou na camada de Apresentação realizando o monitoramento da navegação e registro das ações dos usuários com a interface, entre outras.

Geralmente nesta camada são registrados dados de log e gerados indicadores estatísticos que permitem a realização da gestão e monitoramento dos recursos envolvidos, variando sua aplicabilidade de acordo com os requisitos necessários a cada uma das camadas, bem como das aplicações específicas para IoT a serem implementadas.

4.1.4 Redes e protocolos de comunicação

A camada de rede é responsável por realizar o transporte dos dados, informações e conhecimentos entre todas as outras camadas. Sendo que, a comunicação pode interconectar máquinas e humanos, podendo ocorrer nos sentidos de: M2M (ex: Dispositivo enviando dados para um servidor), P2M (ex: uma pessoa acessando dados de um

dispositivo) e P2P (ex: uma pessoa compartilhando dados e informações com outra pessoa).

Quanto ao conteúdo transportado este abrange os dados gerados pelos dispositivos de IoT, informações de gerenciamento e controle, bem como o fornecimento de dados, informações e conhecimentos solicitados pelas aplicações de IoT. Além de envio de comandos aos dispositivos.

Neste contexto, diversos protocolos de comunicação podem ser utilizados em diferentes situações, e a comunicação pode ocorrer por meio de rede com ou sem fio, ou até mesmo utilizando as duas maneiras na mesma solução, entre eles, citam-se: TCP/IP, UDP, Satellite, Wi-Fi, Ethernet e telefonia (3G, 4G) CoAP, MQTT, IPv6, 6LoWPAN, HTTP, ZigBee, CAN, Bluetooth, RFID, NFC, entre outros.

4.1.5 Serviços de suporte

Esta camada abrange a extração, processamento, armazenamento e recuperação dos dados e informações coletadas da camada de Dispositivos IoT.

A extração e processamento dos dados pode acontecer por meio do uso de bibliotecas e códigos computacionais que possibilitam a conexão e aquisição dos dados dos dispositivos conectados e o tratamento necessário para o devido armazenamento. Portanto, nesta camada podem existir diversos agentes ativos e recursos disponíveis, que realizam a coleta, processamento e armazenamento de dados.

A coleta e processamento inicial dos dados dos dispositivos IoT pode acontecer por meio de recursos sendo requisitados aos dispositivos e estes respondendo aos comandos, ou através de soluções embarcadas que comunicam com um servidor para conectar e enviar seus dados.

Geralmente são necessárias adaptações, conversões ou até mesmo correções nos dados, antes de seu armazenamento. Além disso, é importante que o conteúdo armazenado esteja estruturado com as relações e formatos adequados para facilitar a consulta dos mesmos, nesse sentido a geração de metadados também pode ser considerada.

O processamento dos dados, quando necessário, pode acontecer no sentido de agrupar e resumir as informações, e gerar novas informações a partir do cruzamento dos dados.

O armazenamento ou publicação dos dados dos dispositivos geralmente acontece por meio do uso de SGBD, bases NoSQL, arquivos de sistema (ex: CSV ou XLS), mecanismos de publish/subscriber como MQTT ou CoAP, entre outros.

De acordo com os requisitos da aplicação IoT a ser implementada e infraestrutura disponível deve ser analisada a utilização dos ambientes de fog e cloud computing, sendo possível reduzir a carga de transmissão, processamento e armazenamento dos dados na cloud através da fog ou edge computing.

Além disso, esta camada também é responsável por fornecer a interface para a camada de Apresentação consumir os dados, informações e conhecimentos coletados e gerados a partir dos dispositivos de IoT, bem como, responder a todas às solicitações demandas pelos usuários e dispositivos conectados. E para a camada de Conhecimento, na recuperação e descoberta de novas relações e conhecimentos.

Os serviços disponíveis devem atender aos requisitos de diferenciados tipos de aplicações de IoT, sendo necessário serviços específicos para cada domínio de aplicação. Ainda assim, pode ser necessário o agrupamento e detalhamento destes serviços em vários recursos, a fim de fornecer diferentes funções de suporte.

Aplicações orientadas a serviços (SOA) seguindo o conceito de RESTful sobre HTTP tem sido bastante usadas na disponibilização dos recursos através de uma API. Porém, outras formas também podem ser utilizadas para cumprir tais objetivos.

Desta maneira, esta camada se comunica com as camadas de Dispositivo Iot, Apresentação e de Conhecimento.

4.1.6 Apresentação

A camada de Apresentação fornece funcionalidades de interação com o usuário, possibilitando o acesso aos dados, informações e conhecimentos gerados a partir do conteúdo coletado e processado dos dispositivos IoT conectados.

Também abrange a apresentação de informações de gerenciamento e monitoramento dos recursos do ambiente de IoT como um todo, respeitando o perfil e permissões de acessos dos usuários.

Nesta camada podem ser acessados os conteúdos através de diversas interfaces, como o uso de: portais web e de intranet, painéis de gestão (dashboards), ferramentas de BI e de análise estatística, sistemas corporativos, aplicativos de smartphone, redes sociais, etc.

A apresentação pode atender as necessidades de diversos tipos de usuários, com o fornecimento de respostas e conteúdos solicitados, de forma específica à área de aplicação em que a solução IoT estiver sendo

aplicada, como exemplo citam-se: Cidades Inteligentes, Indústrias, Energia, Corpo Humano, Residências, Ambientes de venda, Escritórios, Fábricas, Locais de trabalho, Veículos, entre outro.

4.1.7 Conhecimento

A camada de conhecimento é atribuída a responsabilidade de tratar da implementação de métodos e técnicas da EC, podendo abranger processos como o de Aquisição, Integração, Descoberta e Recuperação de conhecimentos.

Vários aspectos da EC são passíveis de implementação em um ambiente de IoT, sendo necessária a identificação dos requisitos específicos de cada solução para definição quais técnicas utilizar e aonde e quando aplicá-las.

Com a implementação de uma base de conhecimento se torna possível, por exemplo, a aplicação de métodos de aprendizado e raciocínio de máquina, resolução de problemas, mecanismos de inferência, de respostas à perguntas (querying answer), entre outros. Neste contexto também se considera a utilização de agentes de inteligência artificial e de métodos e técnicas de descoberta de conhecimentos como KDT e KDD.

Contudo, para desenvolvimento de um SC é fundamental que exista uma base de conhecimento para que seja possível a aplicação das técnicas da EC comentadas. Desta maneira, a geração de um BC é um dos requisitos principais a serem atendidos nesta camada.

Em ambientes da IoT o uso de tecnologias da web semântica e modelos de ontologias tem sido aplicados para a representação de conhecimentos e publicação de dados da IoT. Através da semântica aplicada nestes dados é possível a representação de conhecimentos sobre observações feitas por um sensor, por exemplo. Ou até mesmo aplicar regras de inferência nestes dados, como exemplo, uma lógica que baseada em conhecimentos representados na BC e em dados de observação de um sensor, acione automaticamente uma atuação em um ambiente de IoT.

Com o uso de ontologias, vocabulários compartilhados e da web semântica é possível gerar a base de conhecimento desejada sobre o contexto da solução, permitindo a base necessária para que um SC para IoT possa ser implementado. Contudo, vale ressaltar, que um alto nível de interoperabilidade também pode ser alcançado a partir desta BC, o qual é uma dos requisitos principais de ambientes da IoT.

A criação da base de conhecimento proposta neste modelo considera um processo de anotação semântica e publicação dos dados e

informações no formato de triplas RDF em uma base de grafos. E utiliza como modelos e vocabulários base o: RDFs, OWL e SOSA/SSN. Desta maneira é necessário a disponibilização de um endpoint que permita que consultas SPARQL sejam realizadas para armazenamento, atualização e recuperação dos conhecimentos representados semanticamente.

Entre os dados envolvidos citam-se informações sobre o contexto do ambiente e dos usuários conectados, a representação de metadados, recursos e configurações disponíveis, a representação dos dados e informações obtidos de dispositivos de IoT do ambiente, entre outros. Todavia, esta camada está relacionada a atribuição de significado semântico aos dados e informações que abrangem a solução IoT em que o SC será implementado, portanto, pode ser expandida de acordo com o ambiente a ser implementado.

Esta camada é apoiada pela camada de Serviços de suporte, no sentido de que os dados advindos dos dispositivos IoT são comunicados a partir desta, ela se torna o ponto em que o processo de anotação e publicação dos dados com semântica pode ser executado. Maiores detalhes sobre estes fluxos são apresentados na seção 4.4, e sobre o processo de anotação e modelagem semântica na seção 4.5.

A camada de Serviços de suporte além possibilitar manter os dados da base de conhecimento atualizados, também dá o suporte para a recuperação de conhecimentos por meio de consultas SPARQL. Um exemplo do fluxo deste processo é apresentado no diagrama da Figura 28.

Contudo, agentes de software automatizados podem realizar este procedimento considerando fluxos alternativos ao do envio de dados do dispositivo IoT à camada de serviços de suporte, sendo que este processo de aquisição pode ser realizado de outra maneira, identifica-se a possibilidade de neste ponto, utilizar técnicas da EC de aquisição do conhecimento, como KDT e KDD, apesar destes assuntos não são serem abordados detalhadamente neste modelo.

Na próxima seção são apresentadas as características e requisitos gerais considerados neste modelo.

4.2 REQUISITOS GERAIS

Os requisitos estão separados em funcionais e não-funcionais e agrupados entre as camadas do modelo, sendo expostos requisitos que englobam o cenário da IoT, a aplicação das ontologias e web semântica e a interoperabilidade semântica.

A seguir são listados e detalhados os requisitos não funcionais do modelo proposto:

- Interoperabilidade: Capacidade de comunicação dos dados entre ambientes heterogêneos.
- Escalabilidade: Suportar grande número de dispositivos e usuários conectados.
- Disponibilidade: Manter o funcionamento dos serviços e comunicações com alta disponibilidade.
- Adaptabilidade: Permitir adicionar novas tecnologias ao ambiente.
- Confiabilidade: Manter a segurança nas comunicações.
- Gerenciamento: Possibilitar acompanhar o desempenho das operações realizadas.

Apresenta-se abaixo uma listagem dos requisitos funcionais principais que devem ser considerados na implementação de ambientes de IoT considerando também as características de implementação de um SC, sendo apresentados de modo separado pelas respectivas camadas propostas no tópico anterior:

- Segurança e Confiabilidade
 - Autenticação e Autorização de acessos
 - Garantia de segurança na comunicação necessária ocorra com segurança
 - Identificação dos dispositivos e usuários conectados
 - Gestão das permissões e configurações dos usuários e dispositivos conectados
 - Permitir controle e auditoria dos registros de acesso
- Redes e protocolos de comunicação
 - Possibilidade de comunicação entre dispositivos IoT (M2M)
 - Permitir que pessoas se comuniquem com dispositivos IoT (P2M)
 - Possibilidade de comunicação entre pessoas (P2P)
- Gerenciamento
 - Monitoramento e gerenciamento dos serviços, usuários e dispositivos conectados
 - Monitoramento do consumo de processamento, memória e rede dos servidores
 - Gerenciamento de dados e permissões de acesso com segurança

- Apresentação de estatísticas e indicadores de desempenho
- Dispositivos IoT
 - Validação de integridade
 - Conectividade com as coisas
 - Monitoramento do equipamento
 - Configurações de conexão com servidor
 - Mobilidade
- Serviços de suporte
 - Controle de acesso aos dados das bases
 - Validação dos dados
 - Transformação de dados
 - Armazenamento dos dados
 - Gerenciamento de um grande volume de dados
 - Recuperação de dados da camada de Conhecimento via consultas SPARQL.
 - Consulta à dados de fontes distintas, exemplo: SGBD, NoSQL, CSV, entre outros.
 - Utilizar interfaces programáveis como API com serviços RESTful
- Apresentação
 - Interface de autenticação e controle das permissões do usuários
 - Interfaces que consideram as necessidades do usuário da aplicação
 - Visualização da situação dos dispositivos conectados
 - Visualização dos dados e informações históricas dos dispositivos conectados.
 - Recebimento automatizados de alertas pré-programados
 - Informações em tempo real
- Conhecimento:
 - Aplicar anotação semântica nos dados coletados
 - Vinculação de dados com modelos externos de ontologias e schemas
 - Utilizar processos de engenharia de ontologias para modelagem do domínio
 - Utilizar ontologia SOSA/SSN como padrão para anotação dos dados de dispositivos IoT

- Aplicar ontologias específicas relativas ao contexto da aplicação.
- Armazenamento de triplas no padrão RDF
- Recuperação de conhecimento via consulta SPARQL
- Descoberta de conhecimentos por meio de inferências

A seguir apresenta-se o modelo de domínio e uma descrição das entidades envolvidas.

4.3 MODELO DE DOMÍNIO

As principais entidades envolvidas neste modelo proposto são apresentadas no diagrama da Figura 24, onde demonstra as relações e comunicações entre as principais camadas do modelo.

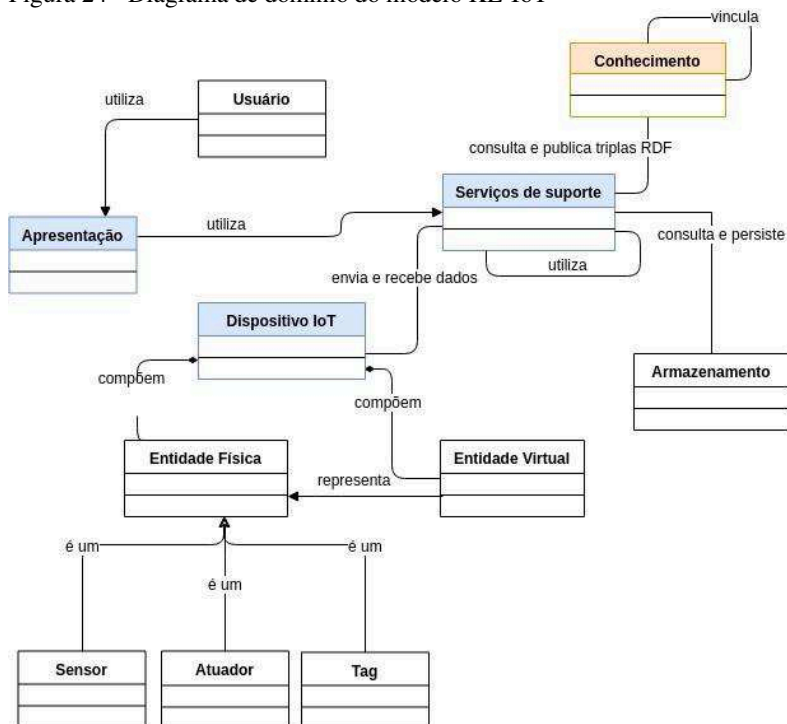
A camada de Conhecimento é representada na cor laranjada, e as camadas de Apresentação, Dispositivo IoT e Serviços de suporte estão na cor azul. As entidades em branco, são representações de entidades conceituais que estão envolvidas no modelo e são apresentadas para ampliar o entendimento do contexto.

Sobre as camadas de Redes e protocolos de comunicação, Gerenciamento e Segurança não foram apresentadas no diagrama, pois são camadas verticais e circundam todas as demais camadas do modelo.

Contudo, é possível visualizar no diagrama que o dispositivo IoT é composto por uma entidade virtual e uma física, sendo que a virtual pode existir através de registro do dispositivo em uma base de dados, por exemplo, e a física tem relação direta com os objetos do mundo real, como a utilização de sensores, atuadores ou tags de identificação.

O usuário da aplicação também é representado neste diagrama, apresentando relação direta com a camada de Apresentação, pois é por meio desta que o usuário tem acesso ao ambiente IoT e todos os serviços e recursos disponibilizados.

Figura 24 - Diagrama de domínio do modelo KE-IoT



Fonte: O autor (2019)

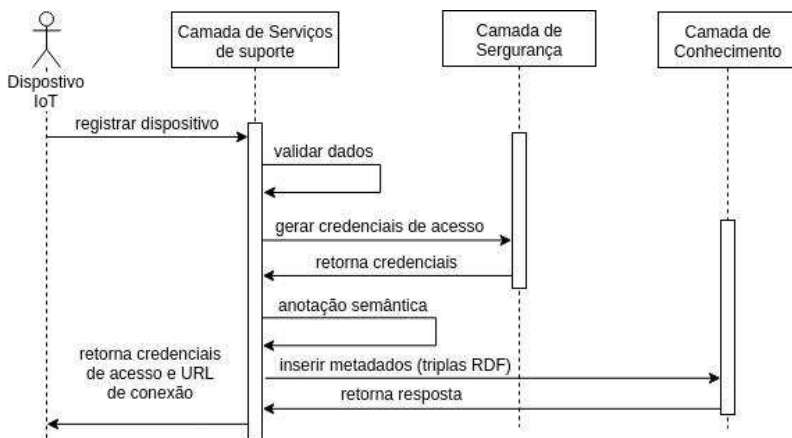
A camada de Serviços de suporte funciona como um núcleo central que atende às solicitações da camada de Apresentação e do Dispositivo IoT, realizando recuperação e processamento de registros da camada de Conhecimento, bem como, de outros tipos de bases de armazenamento que podem ser utilizadas para responder a estas demandas solicitadas.

4.4 FLUXOS BÁSICOS DE COMUNICAÇÃO

Neste tópico são apresentados alguns cenários para exemplificar os fluxos de informações entre as camadas do modelo. Sendo detalhados exemplos de casos de uso relativos respectivamente aos processos de: identificação e registro, autenticação, publicação de dados e consulta de dados.

O processo de identificação e registro do dispositivo IoT é a primeira etapa para conectar o dispositivo IoT ao ambiente de aplicação. Nesta etapa o dispositivo realiza comunicação com o servidor, passando informações sobre os recursos disponíveis e metadados do dispositivo. Desta maneira, é gerado um código único para registro do dispositivo dentro do ambiente IoT e são realizados os procedimentos de anotação semântica e persistência de triplas na base de conhecimento com os metadados do dispositivo. Após a identificação e registro do dispositivo, o servidor deve responder com as credenciais para autenticação e as configurações de comunicação para envio dos dados gerados pelo dispositivo IoT. Na Figura 25 é possível visualizar um exemplo da comunicação entre as respectivas camadas do modelo que são envolvidas durante este procedimento.

Figura 25 – Diagrama de sequência de Registro de Dispositivo IoT

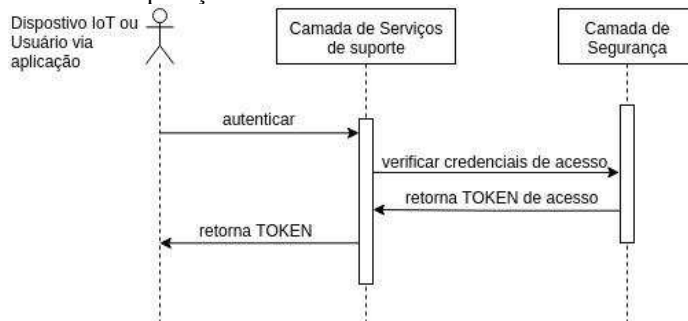


Fonte: O autor (2019)

O processo de autenticação e autorização acontece através do envio de usuário e senha ao servidor, e este deve realizar a validação do mesmo, e responder positivamente se as credenciais forem válidas, ou negativamente caso o usuário ou senha estiverem incorretos ou o dispositivo não possua mais autorização ao ambiente. No caso da resposta positiva, ainda se espera que o servidor forneça informações relativas aos recursos que o usuário autenticado possui autorização de acesso. Este mecanismo pode ser utilizado tanto para conexões de usuários através de acesso pela Camada de Apresentação, como para Camada de Dispositivo

IoT, sendo considerados como os atores no diagrama apresentando na Figura 26.

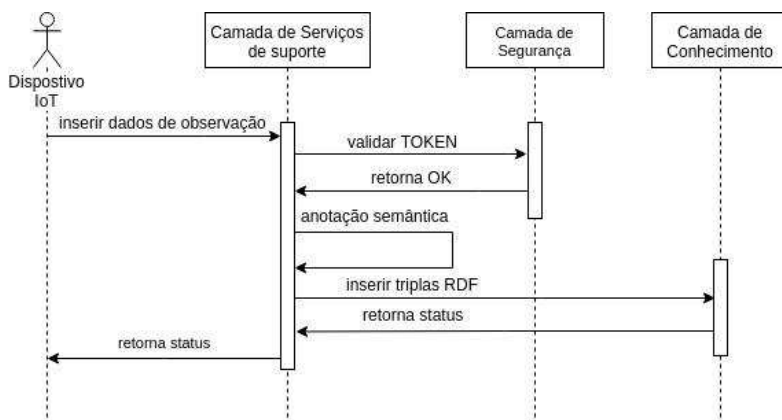
Figura 26 – Diagrama de seqüência para autenticação de Dispositivos IoT e Usuários da aplicação



Fonte: O autor (2019)

Na Figura 27 é apresentado o fluxo referente ao serviço de recebimento e publicação de dados, neste caso é considerado o recebimento dos dados de um dispositivo IoT pela camada de Serviços de suporte, o qual valida primeiramente a credencial de acesso com a camada de Segurança, e posteriormente realiza a validação dos dados e a anotação semântica para armazenamento das triplas RDF na Camada de Conhecimento.

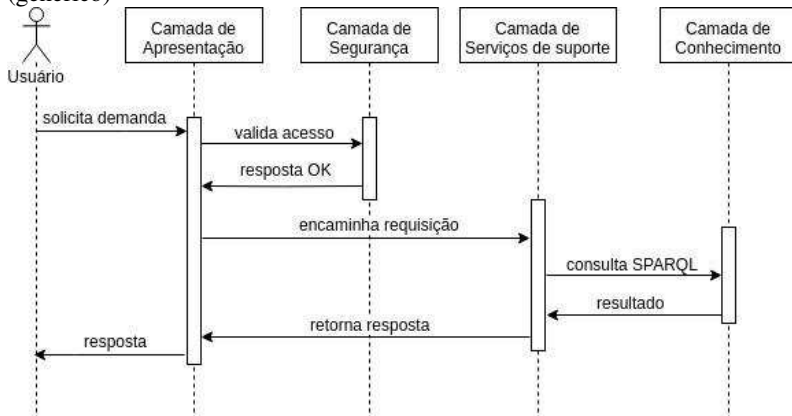
Figura 27 - Diagrama de seqüência do envio de dados do Dispositivo IoT



Fonte: O autor (2019)

O diagrama da Figura 28 demonstra um exemplo de solicitação de demanda por um usuário conectado ao sistema via camada de Apresentação, sendo realizada a validação do acesso e posteriormente a requisição da demanda.

Figura 28 - Diagrama de sequência de solicitações de demanda de usuários (genérico)



Fonte: O autor (2019)

A seguir apresenta-se informações para realização do processo de anotação semântica demonstrado nos fluxos de registro do dispositivo IoT e no de publicação de dados.

4.5 APLICAÇÃO DA WEB SEMÂNTICA E ONTOLOGIAS

A aplicação de tecnologias da web semântica e ontologias tem papel chave na proposta deste modelo, uma vez que este trata da geração de uma base de grafos, no formato de triplas RDF, que permite a representação de conhecimentos dos dados de observações oriundos da Camada de Dispositivo IoT, bem como, do domínio da aplicação como um todo.

Através de anotações semânticas para identificação e gerenciamento dos dispositivos de IoT conectados é possível especificar as capacidades e metadados dos dispositivos IoT, visando facilitar a descoberta e inter-relação de seus recursos com outros elementos conectados, incluindo usuários, de forma automatizada.

A construção e manutenção da ontologia utilizada na solução deve adotar uma metodologia da engenharia de ontologias que considere a reutilização de ontologias existentes, evitando, por exemplo, a geração de diferentes vocabulários para a mesma finalidade.

Contudo, é recomendada a utilização da metodologia SEG 3.0 por ser desenvolvida com base em metodologias da engenharia de ontologias, e ter como foco o fornecimento da interoperabilidade semântica e da representação dos conhecimentos envolvidos no contexto da IoT e Cidades Inteligentes. As etapas desta metodologia está detalhada a seguir:

Composição: Realizar a identificação das fontes de dados, formatos e tratamentos necessários.

1. Modelagem: Definir e aplicar modelo, vocabulários e ontologias para unificar os dados.
2. Vínculação: Permitir que os dados sejam vinculados a outros conjuntos de dados RDF, possibilitando obter informações adicionais.
3. Resolução: Aplicar regras que possibilitem o raciocínio de máquina, como mecanismos de inferência.
4. Consulta: Recuperar dados da base de conhecimento por meio de consulta SPARQL baseada no modelo de ontologia que está sendo utilizado.
5. Serviços: Disponibilizar serviços que facilitem a realização de consultas aos dados da base de conhecimento.
6. Composição de serviços: Agregar serviços no intuito de prover novos serviços a partir da composição dos serviços existentes.

As três primeiras etapas dão a base para que a anotação semântica inicial possa ser realizada, sendo possível a partir destas anotar e armazenar os dados em formato RDF na base de grafos. As demais etapas, não influenciam diretamente nesta fase de anotação semântica, uma vez que para execução das etapas de Consulta, Serviços e Composição de serviços os dados já necessitam estarem armazenados na base de conhecimento em formato de triplas. E no caso da etapa de Resolução, esta também pode ser executada após o armazenamento destes dados no grafo, apesar do papel importante desta na realização de descoberta e inferência de novos conhecimentos, por exemplo.

Contudo, um repositório de armazenamento de triplas RDF em formato de grafos deve ser disponibilizado, possibilitando a inserção,

atualização, remoção e recuperação dos conhecimentos representados no grafo através da linguagem SPARQL. Neste contexto, diversos formatos para serialização dos dados podem ser utilizados como os padrões Turtle ou RDF-a, por exemplo.

4.6 CONSIDERAÇÕES DO CAPÍTULO

O modelo proposto visa integrar as visões e diretrizes seguidas pela comunidade da IoT, e incorporar uma abordagem voltada para construção de SC baseados em ontologias e tecnologias da web semântica.

A ideia principal está em possibilitar um cenário que considere os diversos aspectos tecnológicos da IoT, como segurança, gerenciamento, escalabilidade, interoperabilidade, protocolos, entre outros. E que aplique uma ótica do conhecimento, abordando metodologias, ontologias e tecnologias da web semântica envolvidas para geração de uma base de grafos RDF.

Entende-se que as iniciativas dadas no sentido da SWoT possuem relação direta com a construção de um SC para ambientes de IoT, pois estudam o uso de ontologias e da web semântica aplicadas neste cenário. Porém, estas iniciativas não apresentam um detalhamento maior sobre os demais aspectos da IoT, e se apresentam principalmente no intuito de solucionar o desafio de interoperabilidade inerente à ambientes de IoT, mas entende-se que com uma base de conhecimento deste contexto se torna possível que sejam trabalhadas questões relativas a EGC.

5 AMBIENTE SIMULADO E RESULTADOS EXPERIMENTAIS

Nesta seção são apresentados os detalhes do ambiente simulado que foi desenvolvido para verificação do modelo proposto neste trabalho. O cenário criado trata-se de um ambiente hipotético que simula uma aplicação de IoT para veículos, onde dispositivos simulados de IoT com capacidade de aquisição de dados e comunicação com servidor HTTP, enviam dados de observações de sensores ao ambiente IoT, bem como, metadados sobre os recursos disponibilizados.

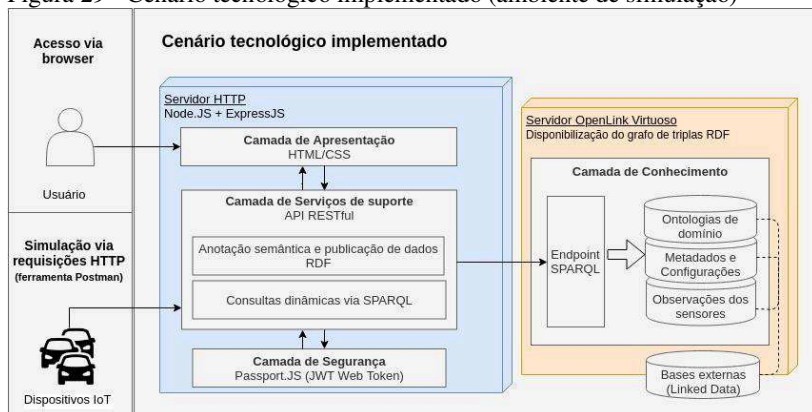
O objetivo principal do desenvolvimento deste ambiente está em possibilitar a verificação do modelo proposto no tocante as diretrizes impostas, considerando os conceitos, requisitos e fluxos de comunicação especificados. E a partir da geração de uma base de conhecimento de grafos no formato de triplas RDF, com informações simuladas deste contexto, seja possível realizar testes de recuperação de conhecimentos via consultas SPARQL e a obtenção de resultados em interface web.

Contudo, não são utilizados equipamentos físicos nem veículos reais na implementação deste ambiente, sendo estes simulados através de requisições HTTP via ferramenta Postman.

A Figura 29 apresenta o cenário lógico desenvolvido, mostrando as comunicações de usuários e dispositivos com o servidor HTTP que implementa requisitos relativos as seguintes camadas: Apresentação, Serviços de suporte, Segurança e de Conhecimento (relacionado a parte de Anotação semântica), sendo que alguns dos requisitos desta última camada, também são contemplados com a implantação da base de conhecimento de grafos RDF, através do servidor OpenLink Virtuoso.

Quanto a camada de redes e protocolos de comunicação, ela está presente na comunicação que acontece entre os servidores deste cenário, e na comunicação dos usuários ou dispositivos IoT com o servidor HTTP.

Figura 29 - Cenário tecnológico implementado (ambiente de simulação)



Fonte: O autor (2019)

Nos tópicos a seguir estão detalhadas as tecnologias e configurações utilizadas no desenvolvimento deste ambiente de simulação, seguido de um detalhamento do funcionamento do dispositivo IoT simulado, dos serviços web disponibilizados, e das ontologias usadas, e por fim, são apresentados os testes realizados e os resultados obtidos seguido de uma verificação da aderência do ambiente simulado desenvolvido em relação ao modelo proposto.

5.1 TECNOLOGIAS E INFRAESTRUTURA UTILIZADA

As tecnologias de desenvolvimento web que foram utilizadas na implementação da API e na visualização de informações via navegador, são Node.JS e bibliotecas em Javascript como a Passport para tratar de questões relativas a autenticação e geração de tokens de acesso e a Express.JS que fornece um servidor HTTP com mecanismos de rotas para disponibilização dos serviços web implementados.

Quanto aos formato dos dados trafegados neste ambiente, foi definido como formato padrão o JSON na comunicação entre as camadas de Dispositivo IoT, Serviços de suporte, Apresentação e de Conhecimento.

Sobre a base de conhecimento, esta foi gerada a partir da implantação do OpenLink Virtuoso Server, da criação de uma base de

grafos no formato de triplas RDF e da disponibilização de um endpoint SPARQL para acesso a mesma.

No processo de anotação semântica e publicação dos dados para geração da base de conhecimento, foram usados os modelos OWL, RDF-S e SOSA/SSN. Sendo realizadas as inserções na base através da biblioteca Virtuoso SPARQL Client, a qual realiza a conexão com o endpoint criado e executa queries SPARQL.

Referente a rede utilizada trata-se do padrão Ethernet TCP/IP, sendo disponibilizados serviços web no modelo de API RESTful sob o protocolo HTTP.

E o computador utilizado na implementação deste ambiente possui processador Intel i5 2,8Ghz, com 8GB de memória, 1 terabyte de disco rígido e sistema operacional Linux Ubuntu 16.04.4 LTS.

5.2 SIMULAÇÃO DO DISPOSITIVO IOT

A camada de Dispositivo IoT neste ambiente é simulada e faz parte de um contexto onde é suposta a implantação de um equipamento conectado em veículos, onde enviam dados ao servidor referentes a observações de sensores de velocidade, temperatura e odômetro do veículo conectado.

Esta simulação acontece através da ferramenta Postman, que possibilita a realização de requisições customizadas e programadas ao servidor, via protocolo HTTP. Quanto aos dados enviados nestas requisições, são preenchidos manualmente em formato JSON.

Sobre as informações de autenticação necessárias ao ambiente, como o token e usuário e senha, são inseridas manualmente na ferramenta de acordo com os fluxos definidos para cada serviço web disponibilizado. Desta maneira, é possível realizar a simulação da comunicação de um dispositivo IoT com o servidor, considerando a implementação de mecanismos de segurança e privacidade, como o uso de tokens de acesso.

Contudo, é considerado que o dispositivo IoT que for conectar ao ambiente simulado, possua a capacidade de cumprir com as especificações dos serviços web disponibilizados e os formatos previamente definidos, sendo estes assuntos tratados no modelo proposto e especificados com maiores detalhes da implementação, no tópico a seguir.

5.3 SERVIÇOS WEB

A camada de Serviços de suporte foi implementada a partir da disponibilização de uma API web seguindo os princípios de uma arquitetura orientada à serviços no modelo RESTful, a qual utiliza o protocolo HTTP na comunicação na porta 4200. Esta API implementa os fluxos de comunicação propostos no modelo, alguns dos requisitos básicos pertinentes ao ambiente simulado, incluindo requisitos relativos a segurança.

Para implementação de mecanismos de segurança e privacidade no servidor HTTP, através do uso de tokens, foi utilizado o formato JSON Web Token. Sendo que em toda autenticação de sucesso no servidor, um novo token é gerado e neste pode conter diversas informações, como exemplo, os recursos em que o usuário e/ou dispositivo IoT que executou login está autorizado a acessar, bem como, informações sobre URL de conexão para envio de dados.

O formato JSON é o padrão definido para todos os serviços implementados neste servidor. E para cada recurso (URI) disponibilizado, o token gerado deve ser enviado junto na requisição, somente assim os serviços permitirão a execução da ação solicitada, caso contrário, o mecanismo de segurança implementado com Passport.JS irá identificar a falta de permissão necessária para acesso ao recurso e irá realizar o bloqueio da chamada, retornando detalhes do problema ocorrido. Exceto os serviços web de registro e o de autenticação, estes devem ser requisitados em um processo anterior a geração do token, devido à necessidade prévia de identificação e autorização do dispositivo ao ambiente desenvolvido.

Contudo, na processo de codificação e decodificação do token é necessário que uma senha seja definida para realizar tal procedimento, para este ambiente a senha é “iotSimulatorPlataformKey”, conforme é possível ver em maiores detalhes no APÊNDICE C.

Os serviços web desenvolvidos e disponibilizados no servidor HTTP estão detalhados no Quadro 5.

Quadro 5 – Detalhamento dos serviços web desenvolvidos

URI	Método HTTP	Descrição	Resposta
/api/register	POST	registra o dispositivo armazenando seu código e realizando anotações semânticas dos metadados do dispositivo.	usuário e senha são retornados na resposta da chamada, para que o dispositivo possa realizar a autenticação no ambiente IoT desenvolvido.
/api/authentication/login	POST	autentica dispositivo IoT e fornece as autorizações de acesso e configurações. recebe usuário e senha e valida na base de dados	Positiva: Retorna o token e a URL em que o dispositivo deve enviar seus dados, para que fique dinâmico o destino dos dados a cada autenticação. Negativa: Retorna mensagem de erro.
/api/device-data	POST	salva dados recebidos do dispositivo IoT: realiza anotações semântica e publicar dados na base de conhecimento (triple store).	Positiva: Retorna mensagem de sucesso Negativa: Retorna mensagem de erro com detalhes sobre o problema.
/api/device-data/:código	GET	obtem dados mais atuais de um sensor via consulta SPARQL na base de conhecimento	Retorna objeto JSON com os resultados da consulta

Fonte: O autor (2019)

O serviço web de registro do dispositivo IoT tem por objetivo “conectar” o dispositivo ao ambiente IoT desenvolvido. O registro é efetivado a partir do envio dos metadados do dispositivo, em formato JSON previamente definido, conforme apresentado na Figura 30.

Figura 30 – Estrutura JSON para envio ao serviço web de registro

```
1 {  
2   "code": "ITXLLSS2424",  
3   "sensors": {  
4     "velocity": "true",  
5     "temperature": "true",  
6     "odometer": "true"  
7   }  
8 }
```

Fonte: O autor (2019)

Uma vez que o dispositivo realizar o registro, será gerado um código de identificação que é retornado na resposta da requisição, juntamente com as credenciais (usuário e senha) de acesso, para que seja possível a realização da autenticação e obtenção do token que possibilitará a comunicação com o serviço de envio de dados.

O serviço web de autenticação tem por objetivo atender tanto a camada de Dispositivo IoT quanto a camada de Apresentação. Sendo que os dispositivos necessitam da autenticação para enviar os dados de sensores e informações relativas aos recursos disponíveis, e também usuários poderão estar autenticando para acessar informações da camada de serviços de suporte. Ambas as comunicações devem ocorrer através do envio de token gerado pelo serviço de autenticação.

Na requisição da autenticação é necessário o envio dos parâmetros de usuário e senha, para que sejam validados pelo servidor HTTP. Em caso de sucesso na validação, um token no formato Bearer¹⁰ é repassado na resposta à requisição deste serviço, e caso as credenciais fornecidas não sejam válidas, uma mensagem de erro é retornada. Um exemplo da mensagem JSON de resposta com o token gerado e com a URL para envio dos dados, no caso de sucesso na autenticação, é demonstrado na Figura 31.

¹⁰ Disponível em: <<https://tools.ietf.org/html/rfc6750>>. Acesso em: novembro de 2018.

Figura 33 - Estrutura JSON de envio ao serviço web de inserção de dados

```
1 {  
2   "velocity": "60",  
3   "temperature": "26",  
4   "odometer": "200000",  
5   "datetime": "2019-01-20 08:10:00"  
6 }
```

Fonte: O autor (2019)

O serviço de obtenção dos dados mais atuais dos sensores foi desenvolvido para funcionar de forma dinâmica de acordo com o parâmetro recebido na URI da requisição. Este parâmetro trata-se do código de identificação do dispositivo IoT, sendo utilizado na consulta SPARQL realizada na base de conhecimento.

Os serviços de envio e obtenção de dados foram implementados no mesmo arquivo, em funções separadas, conforme é possível visualizar estes métodos no APÊNDICE D.

5.4 BASE DE CONHECIMENTO

A criação da base de conhecimento foi a partir da implantação do servidor OpenLink Virtuoso na porta 8890, e da criação de um grafo (dataset) para armazenamento das triplas RDF e geração da base com representações de conhecimentos do contexto através da utilização de ontologias.

O objetivo deste dataset é armazenar informações dos recursos oferecidos (metadados) de cada dispositivo IoT conectado ao ambiente, bem como, armazenar os dados gerados pelas observações registradas pelo mesmo.

O procedimento de anotação semântica para geração da base está apoiado principalmente no uso de classes e propriedades da ontologia SOSA/SSN, e também os modelos RDFs e OWL conforme proposto no modelo.

Os modelos de ontologias utilizados são apresentados com seus prefixos e respectivas URLs, sendo eles:

- owl: <<http://www.w3.org/2002/07/owl#>>
- rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>
- rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>

- ssn: <<http://www.w3.org/ns/ssn/>>
- sosa: <<http://www.w3.org/ns/sosa/>>

As classes do modelo SOSA/SSN que foram consideradas nos procedimentos de anotação semântica implementados, em formato de URL, são os seguintes:

- <http://www.w3.org/ns/sosa/Platform>
- <http://www.w3.org/ns/sosa/FeatureOfInterest>
- <http://www.w3.org/ns/sosa/Sensor>
- <http://www.w3.org/ns/sosa/ObservableProperty>
- <http://www.w3.org/ns/sosa/Actuator>
- <http://www.w3.org/ns/sosa/ActuableProperty>

Sobre as propriedades utilizadas, também estão apresentadas no formato de URLs, na lista a seguir:

- <http://www.w3.org/ns/sosa/hosts>
- <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
- <http://www.w3.org/ns/sosa/isObservedBy>
- <http://www.w3.org/ns/sosa/isSampleOf>
- <http://www.w3.org/ns/ssn/deployedOnPlatform>
- <http://www.w3.org/ns/sosa/isHostedBy>
- <http://www.w3.org/ns/sosa/observes>
- <http://www.w3.org/ns/sosa/hosts>

Os scripts que realizam a anotação semântica estão implementados nos serviços de registro e de inserção de dados, estes podem ser visualizados em maiores detalhes no. Quanto ao serviço de autenticação, este está exposto no APÊNDICE C

No serviço de consulta de dados dos dispositivos é executado um script SPARQL, onde o código do dispositivo é um parâmetro recebido de forma dinâmica na URL da requisição, conforme é possível visualizar na Figura 34. E desta maneira, a consulta é realizada e os dados encontrados relativos ao dispositivo especificado são retornados no formato JSON. O código completo com todos os detalhes da implementação deste serviço, pode ser encontrado no APÊNDICE D.

Figura 34 – SPARQL de consulta à dispositivos na base de conhecimento

```

var query = `
  SELECT ?dispositivo, ?valor, ?property, ?sensor, ?date WHERE {
    ?dispositivo <http://www.w3.org/ns/sosa/hasFeatureOfInterest> <dispositivo/+ deviceCode +> ;
    <http://www.w3.org/ns/sosa/hasSimpleResult> ?valor ;
    <http://www.w3.org/ns/sosa/observedProperty> ?property ;
    <http://www.w3.org/ns/sosa/madeBySensor> ?sensor ;
    <http://www.w3.org/ns/sosa/resultTime> ?date .
  } ;

```

Fonte: O autor (2019)

A partir dos serviços web desenvolvidos e disponibilizados na API é possível a realização de testes e coleta de resultados, sendo estes apresentados de forma detalhada no tópico a seguir.

5.5 TESTES E RESULTADOS

A realização dos testes se inicia com a publicação da API e seus respectivos serviços web, para tal, basta a execução do comando “npm run start” no diretório do projeto, conforme é possível visualizar na Figura 35. O código relativo a inicialização desta API está apresentado no APÊNDICE A.

Figura 35 – Inicialização do servidor HTTP

```

leandro@ahura-nb-leandro:~/Workspace/egc/iot-simulator$ npm run start
> iot-simulator@1.0.0 start /home/leandro/Workspace/egc/iot-simulator
> supervisor ./server/http-server

Running node-supervisor with
  program './server/http-server'
  --watch ''
  --extensions 'node,js,/server/http-server'
  --exec 'node'

Starting child process with 'node ./server/http-server'
Watching directory '/home/leandro/Workspace/egc/iot-simulator' for changes.
Press rs for restarting the process.

```

Fonte: O autor (2019)

Com a API disponibilizada, os testes executados são relativos a realização de requisições HTTP aos serviços web disponibilizados que foram apresentados no tópico anterior, utilizando a ferramenta Postman para simulação do Dispositivo IoT e Usuários conectados. No caso do serviço de consulta também é apresentado um caso de requisição via navegador de internet.

Na Figura 36 é possível visualizar a requisição realizada ao serviço web de registro do dispositivo IoT, apresentando os dados enviados e o respectivo resultado obtido.

Figura 36 – Requisição ao serviço web de registro

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:4200/api/register
- Request Body (JSON):**

```

1 {
2   "code": "ITXLLS2424",
3   "sensors": {
4     "velocity": "true",
5     "temperature": "true",
6     "odometer": "true"
7   }
8 }

```
- Response Body (JSON):**

```

1 {
2   "username": "ITXLLS2424",
3   "password": "security",
4   "url": "https://localhost:4200/authentication/login"
5 }

```
- Status:** 200 OK, Time: 509 ms, Size: 483 B

Fonte: O autor (2019)

Os detalhes da requisição realizada ao serviço web de autenticação estão apresentados na Figura 37. Sendo que as credenciais de acesso utilizadas na execução da simulação do Dispositivo IoT, foram geradas a partir do serviço de registro, mostrado anteriormente. Todavia, identifica-se que este serviço também pode ser utilizado para autenticação de Usuários via Aplicação, sendo que neste caso necessita da implementação de mecanismos de controle de sessão na camada da Aplicação.

Figura 37 – Requisição ao serviço web de autenticação

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:4200/api/authentication/login
- Request Body (Form):**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	ITXLLS2424	
<input checked="" type="checkbox"/> password	security	
Key	Value	Description
- Response Body (JSON):**

```

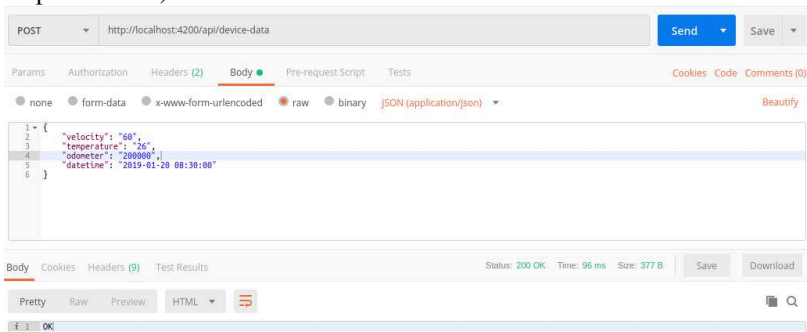
1 {
2   "message": "ok",
3   "user": {
4     "dataset": "http://localhost:4200/device-data",
5     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IjoiITXLLS2424IiwiaWF0IjoiMTUxOTUxODUwMjUwMTYxLnQzZmNFTTc0S2VzbnVhQmQyYjYxVjI1b3Q1fXZlbnQ"
6   }
7 }

```
- Status:** 202 Accepted, Time: 11 ms, Size: 612 B

Fonte: O autor (2019)

Na Figura 38 é possível visualizar a requisição realizada ao serviço web de envio de dados, o qual é responsável pela validação do formato recebido e execução do procedimento de anotação semântica e publicação de triplas RDF, via SPARQL, na base de conhecimento. O endereço de envio desta requisição é informado na propriedade “dataset”, do JSON recebido na resposta da requisição de autenticação, apresentada anteriormente.

Figura 38 – Requisição ao serviço web de envio de dados (simulação do Dispositivo IoT)



Fonte: O autor (2019)

Para o serviço de envio de dados também apresenta-se o SPARQL de inserção destes dados, o qual é gerado no processamento do serviço e escrito no terminal do servidor HTTP, conforme demonstrado na Figura 39.

Sobre o serviço de consulta de dados do dispositivo, este é dinâmico pois de acordo com o valor do parâmetro de código do dispositivo enviado na URL da requisição, é que são obtidos os dados via consulta SPARQL e retornados em formato JSON.

Figura 39 - SPARQL de inserção de dados de observação de sensores (simulados)

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX ogp: <http://www.w3.org/ns/ogp#>
PREFIX sosa: <http://www.w3.org/ns/sosa/>

INSERT IN GRAPH <http://localhost:8890/device-metadata> {
  <dispositivo/ITXLLS2424/sensor/velocimetro/observacao/2019-01-20T10:30:00.000Z sosa:resultTime "2019-01-20T10:30:00.000Z"^^xsd:dateTimeStamp .
  <dispositivo/ITXLLS2424/sensor/velocimetro/observacao/2019-01-20T10:30:00.000Z sosa:hasSimpleResult "00"^^xsd:decimal .
  <dispositivo/ITXLLS2424/sensor/velocimetro/observacao/2019-01-20T10:30:00.000Z sosa:observedProperty v:velocidade .
  <dispositivo/ITXLLS2424/sensor/velocimetro/observacao/2019-01-20T10:30:00.000Z sosa:madeBySensor <dispositivo/ITXLLS2424/sensor/velocimetro .
  <dispositivo/ITXLLS2424/sensor/velocimetro/observacao/2019-01-20T10:30:00.000Z sosa:hasFeatureOfInterest <dispositivo/ITXLLS2424 .
  <dispositivo/ITXLLS2424/sensor/odometro/observacao/2019-01-20T10:30:00.000Z sosa:resultTime "2019-01-20T10:30:00.000Z"^^xsd:dateTimeStamp .
  <dispositivo/ITXLLS2424/sensor/odometro/observacao/2019-01-20T10:30:00.000Z sosa:hasSimpleResult "20000"^^xsd:decimal .
  <dispositivo/ITXLLS2424/sensor/odometro/observacao/2019-01-20T10:30:00.000Z sosa:observedProperty <llometragem .
  <dispositivo/ITXLLS2424/sensor/odometro/observacao/2019-01-20T10:30:00.000Z sosa:madeBySensor <dispositivo/ITXLLS2424/sensor/odometro .
  <dispositivo/ITXLLS2424/sensor/odometro/observacao/2019-01-20T10:30:00.000Z sosa:hasFeatureOfInterest <dispositivo/ITXLLS2424 .
  <dispositivo/ITXLLS2424/sensor/temperaturaMotor/observacao/2019-01-20T10:30:00.000Z sosa:resultTime "2019-01-20T10:30:00.000Z"^^xsd:dateTimeStamp .
  <dispositivo/ITXLLS2424/sensor/temperaturaMotor/observacao/2019-01-20T10:30:00.000Z sosa:hasSimpleResult "20"^^xsd:decimal .
  <dispositivo/ITXLLS2424/sensor/temperaturaMotor/observacao/2019-01-20T10:30:00.000Z sosa:observedProperty <temperatura .
  <dispositivo/ITXLLS2424/sensor/temperaturaMotor/observacao/2019-01-20T10:30:00.000Z sosa:madeBySensor <dispositivo/ITXLLS2424/sensor/temperaturaMotor .
  <dispositivo/ITXLLS2424/sensor/temperaturaMotor/observacao/2019-01-20T10:30:00.000Z sosa:hasFeatureOfInterest <dispositivo/ITXLLS2424 .
}

[[{"head":{"link":[]},"vars":{"callret-0"},"results":{"distinct":false,"ordered":true,"bindings":[{"callret-0":{"type":"literal","value":"Insert into <http://localhost:8890/device-metadata>, 15 (or less) triples -- done"}]}}]]

```

Fonte: O autor (2019)

Na Figura 40 é apresentada a consulta realizada com o valor “ITXLLS2424” para este parâmetro, possibilitando acessar o histórico de registros de observações relativos ao respectivo veículo.

Figura 40 - Requisição ao serviço de consulta de dados de um dispositivo IoT

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:4200/api/device-data/ITXLLS2424
- Params:** Authorization
- Headers:** Content-Type: application/json, Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IjY2...
- Status:** 200 OK, Time: 118 ms, Size: 3.29 KB
- Body (JSON):**

```

1- {
2-   "head": {
3-     "link": [],
4-     "vars": [
5-       "dispositivo",
6-       "valor",
7-       "property",
8-       "sensor",
9-       "date"
10-    ]
11-  },
12-   "results": {
13-     "distinct": false,
14-     "ordered": true,
15-     "bindings": [
16-       {
17-         "dispositivo": {
18-           "type": "uri",
19-           "value": "dispositivo/ITXLLS2424/sensor/velocimetro/observacao/2019-01-20T10:10:00.000Z"
20-         },
21-         "valor": {
22-           "type": "typed-literal",
23-           "datatype": "http://www.w3.org/2001/XMLSchema#decimal",
24-           "value": "00"
25-         }
26-       }
27-     ]
28-   }
29- }

```

Fonte: O autor (2019)

Ainda sobre o serviço de consulta de dados de um veículo específico, é realizado o teste da visualização dos resultados já impressos em um página HTML no browser, conforme demonstrado na Figura 41.

Figura 41 – Demonstração da consulta de dados do dispositivo IoT via navegador de internet

```
{
  "head": {
    "link": [],
    "vars": [
      "dispositivo",
      "valor",
      "property",
      "sensor",
      "date"
    ],
    "results": {
      "distinct": false,
      "ordered": true,
      "bindings": [
        {
          "dispositivo": {
            "type": "uri",
            "value": "dispositivo/ITXLL552424/sensor/velocimetro/observacao/2019-01-20T10:10:00.000Z"
          },
          "type": "typed-literal",
          "datatype": "http://www.w3.org/2001/XMLSchema#decimal",
          "value": "500",
          "property": "value"
        },
        {
          "type": "uri",
          "value": "velocidade",
          "sensor": {
            "type": "uri",
            "value": "dispositivo/ITXLL552424/sensor/velocimetro",
            "date": {
              "type": "typed-literal",
              "datatype": "http://www.w3.org/2001/XMLSchema#dateTimeStamp",
              "value": "2019-01-20T10:10:00.000Z"
            }
          },
          "dispositivo": {
            "type": "uri",
            "value": "dispositivo/ITXLL552424/sensor/odometro/observacao/2019-01-20T10:10:00.000Z",
            "valor": {
              "type": "typed-literal",
              "datatype": "http://www.w3.org/2001/XMLSchema#decimal",
              "value": "200000",
              "property": "value"
            },
            "type": "uri",
            "value": "kilometragem",
            "sensor": {
              "type": "uri",
              "value": "dispositivo/ITXLL552424/sensor/odometro",
              "date": {
                "type": "typed-literal",
                "datatype": "http://www.w3.org/2001/XMLSchema#dateTimeStamp",
                "value": "2019-01-20T10:10:00.000Z"
              }
            },
            "dispositivo": {
              "type": "uri",
              "value": "dispositivo/ITXLL552424/sensor/temperatura/observacao/2019-01-20T10:10:00.000Z",
              "valor": {
                "type": "typed-literal",
                "datatype": "http://www.w3.org/2001/XMLSchema#decimal",
                "value": "26",
                "property": "value"
              },
              "type": "uri",
              "value": "temperatura",
              "sensor": {
                "type": "uri",
                "value": "dispositivo/ITXLL552424/sensor/temperaturaMotor",
                "date": {
                  "type": "typed-literal",
                  "datatype": "http://www.w3.org/2001/XMLSchema#dateTimeStamp",
                  "value": "2019-01-20T10:10:00.000Z"
                }
              }
            }
          ]
        }
      ]
    }
  }
}
```

Fonte: O autor (2019)

Com os resultados obtidos na requisição da consulta realizada é possível a descoberta de outros recursos que também podem ser implementados como serviços web na API, o que permite uma navegação dinâmica e fluida entre os resultados e recursos oferecidos através de links e páginas HTML, por exemplo.

5.6 ADERÊNCIA AO MODELO PROPOSTO

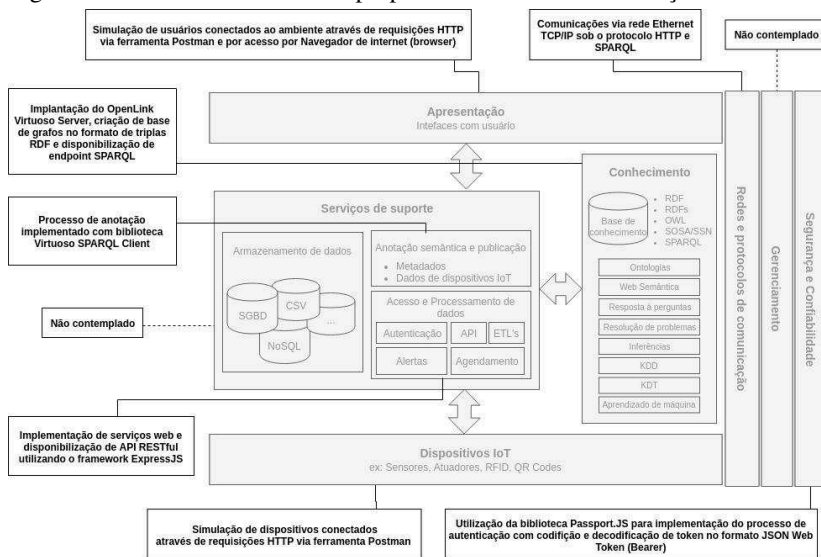
Com as implementações do ambiente simulado, a realização dos testes e coleta dos resultados comentados nos tópicos anteriores deste capítulo, é possível identificar as camadas do modelo contempladas no desenvolvimento do ambiente, bem como, os principais aspectos de implementação abordados em cada uma delas, sendo este conteúdo apresentado na Figura 42.

Vale ressaltar que, não foi realizada a implementação de requisitos da camada de Gerenciamento, uma vez que o objetivo principal do ambiente simulado consiste na geração da base de conhecimento, estes não foram contemplados. E pelo mesmo motivo, também não foi considerada a utilização do armazenamento de dados proposto na camada de Serviços de suporte.

Em relação a aderência dos fluxos de comunicação propostos no modelo com os serviços web do ambiente simulado, foi constatado que todos os fluxos desenvolvidos atendem ao que o modelo propõe, sendo desenvolvido um serviço web para cada fluxo proposto, conforme consta na Figura 43.

Contudo, a parte que envolve a camada de Segurança apesar de ser detalhada separadamente nos diagramas de sequência do modelo proposto, estas possuem também uma existência lógica separada no ambiente simulado, porém, estão implementadas dentro das lógicas da camada de Serviços de suporte.

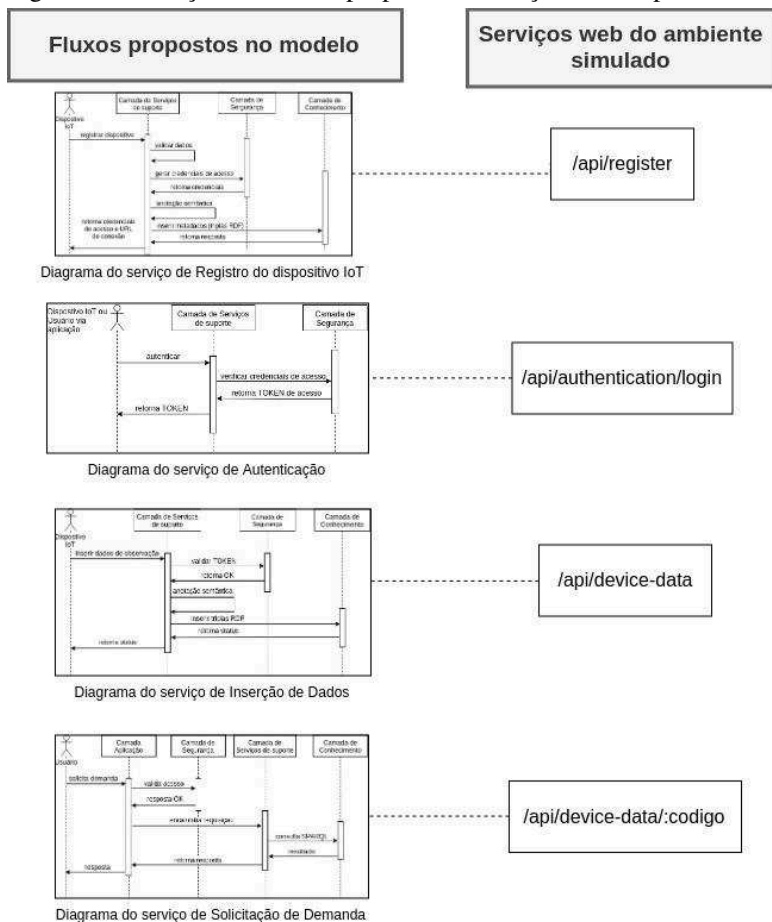
Figura 42 – Aderência do modelo proposto x ambiente de simulação



Fonte: O autor (2019)

Sobre os procedimentos de anotação semântica detalhado no tópico 4.5 do modelo proposto, foram realizadas as etapas de Composição, Modelagem, Vinculação, Resolução, Consulta e Serviços.

Figura 43 – Relação dos fluxos propostos x serviços web disponibilizados



Fonte: O autor (2019)

E quanto ao modelo de domínio exposto na proposta, apenas a entidade física do dispositivo IoT e a parte de Armazenamento é que não foram contempladas, as demais foram implementadas, conforme exposto anteriormente na Figura 42.

6 CONCLUSÃO E TRABALHOS FUTUROS

A proposta de um modelo com uma abordagem generalizada demonstra a possibilidade de realizar especializações de acordo com requisitos específicos do domínio em que o modelo pretende ser aplicado. E a partir do ambiente simulado e das tecnologias implementadas, constata-se que a ótica de Conhecimento aplicada ao modelo e o processo de anotação semântica baseado em ontologias agregam na formação da base de conhecimento desejada, com dados de dispositivos IoT e de contexto em que o mesmo se aplica.

O modelo proposto apresenta características atribuídas a um SC e demonstra que ambientes de IoT podem se beneficiar da aplicação de ontologias e tecnologias da web semântica, no sentido de atender a demanda de interoperabilidade semântica necessária a estes ambientes e principalmente de promover um cenário em que os processos da EC, como de aquisição, integração e recuperação de conhecimentos, possam ser aplicados. Como exemplo, métodos de inferência, descoberta e recuperação de conhecimentos realizados com a aplicação de regras semânticas e de consultas SPARQL na base de conhecimento gerada.

Quanto as especificações OWL, RDF e RDFs se demonstram essenciais no cenário da web semântica e a aplicação da ontologia SOSA/SSN é satisfatória ao atender os principais conceitos envolvidos nos ambientes de IoT, o que permite a representação, compartilhamento e reuso de conhecimentos através da web. Além disso, com a utilização do padrão SOSA/SSN como base no processo de anotação semântica, possibilita que os dados sejam publicados em um modelo já reconhecido no âmbito de ontologias de representação da IoT, demonstrando uma alta capacidade de interoperabilidade e vinculação destes dados com outros modelos de ontologias, ambientes de IoT ou datasets disponibilizados na web. Ressalta-se que modelos de ontologias de contextos específicos de acordo com a solução a ser implementada podem ser utilizados para ampliar a representação dos conhecimentos envolvidos.

Sobre o desenvolvimento do ambiente simulado para IoT, utilizando tecnologias de Javascript como Node.JS e bibliotecas relacionadas, se mostrou uma iniciativa diferenciada das plataformas existentes. Além disso, o ambiente se mostrou atender aos requisitos básicos para implementação da base de conhecimento em grafos no formato de triplas RDF.

De acordo com os objetivos definidos previamente nesta pesquisa, foi possível identificar que a geração de artefatos tecnológicos como os diagramas, especificação das camadas conceituais e dos fluxos de comunicação, apoiam na contextualização dos principais aspectos a serem considerados na implementação de um SC para ambientes de IoT.

No modelo proposto neste trabalho, foi possível apresentar uma abordagem conjunta onde os conceitos e técnicas de EC estão relacionadas à sistemas de IoT e ao contemplar a geração de uma base de conhecimento, o mesmo aborda uma das características fundamentais atribuídas à SC.

Pelo fato do modelo abordar um contexto generalizado para ambientes de IoT, se mostrou possível ampliar os horizontes deste trabalho para as diversas áreas de aplicação da IoT, como transportes, logística, saúde, energia, indústria, entre outras. Contudo, algumas possibilidades de trabalhos futuros foram identificadas e estão relatadas nos tópicos a seguir:

- Especialização do modelo para áreas de domínio de aplicação da IoT, como Smart Cities.
- Validação do modelo a partir de métodos quantitativos, e métodos formais de consultas a especialistas, por exemplo.
- Modelagem de uma ontologia para o modelo KE-IoT a partir de metodologias da engenharia de ontologias.
- Ampliação do modelo e ambiente de simulação para aplicação de outras técnicas da EC, como exemplo KDD, KDT, métodos de raciocínio de máquina e resolução de problemas baseados em caso, entre outros.
- Utilização de equipamentos físicos como microcontroladores Arduino ou Raspeberry Pi no lugar do dispositivo IoT simulado. E aplicação de outros protocolos de comunicação utilizados em ambientes de IoT, como MQTT ou CoAP, possibilitando testar mecanismos de comunicação do tipo publish/subscriber diretamente no meio física.
- Análise de desempenho do OpenLink Virtuoso Server em comparação com diferentes bases de armazenamento de triplas.
- Aprofundamento do modelo considerando abordagens de áreas correlatas, como: Big Data, Cloud e Fog Computing, entre outras.

- Implementação de algoritmos de inferência de dados para o ambiente simulado desenvolvido.

REFERÊNCIAS

ACKOFF, Russell L. From data to wisdom. *Journal of applied systems analysis*, v. 16, n. 1, p. 3-9, 1989.

ASHTON, Kevin et al. That 'internet of things' thing. *RFID journal*, v. 22, n. 7, p. 97-114, 2009.

ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The internet of things: A survey. *Computer networks*, v. 54, n. 15, p. 2787-2805, 2010.

BALDINI, Gianmarco et al. Iot governance, privacy and security issues. Position paper, European Research Cluster on the Internet of Things, 2015.

BARNAGHI, Payam et al. Semantics for the Internet of Things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, v. 8, n. 1, p. 1-21, 2012.

BASSI, Alessandro et al. *Enabling things to talk*. Springer-Verlag GmbH, 2013.

BECHHOFER, Sean et al. The semantics of semantic annotation. In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, Berlin, Heidelberg, 2002. p. 1152-1167.

BECHHOFER, Sean et al. OWL web ontology language reference. *W3C recommendation*, v. 10, n. 02, 2004.

BERMUDEZ-EDO, M. et al. IoT-Lite: A Lightweight Semantic Model for the Internet of Things and its Use with Dynamic Semantics. *Personal and Ubiquitous Computing*, 2017.

BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora. The semantic web. *Scientific american*, v. 284, n. 5, p. 34-43, 2001.

BERNERS-LEE, Tim et al. A framework for web science. *Foundations and Trends® in Web Science*, v. 1, n. 1, p. 1-130, 2006.

BLOEHDORN, Stephan et al. Ontology evolution. *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, p. 51-70, 2006.

BOTTS, Mike et al. OGC® sensor web enablement: Overview and high level architecture. In: *International conference on GeoSensor Networks*. Springer, Berlin, Heidelberg, 2006. p. 175-190.

BRACHMAN, R.; LEVESQUE, H. The Tradeoff Between Expressiveness and Tractability. In: *Knowledge Representation and Reasoning*. 2004. p. 327-348.

BRAGA, M. de M.; RAMOS JÚNIOR, Hélio Santiago; COELHO, T. de F. Aplicações de ontologias na recuperação de informações jurídicas na web semântica. In: *Jornadas Argentinas de Informática-Simposio Argentino de Informática y Derecho*. 2007.

BRICKLEY, Dan; GUHA, Ramanathan V.; MCBRIDE, Brian. RDF vocabulary description language 1.0: RDF Schema. W3C Recommendation (2004). URL <http://www.w3.org/tr/2004/rec-rdf-schema-20040210>, 2004.

BUCKLEY, John. From RFID to the Internet of Things: Pervasive networked systems. European Union Directorate for Networks and Communication Technologies, 2006.

CECI, Flávio. Um modelo baseado em casos e ontologia para apoio à tarefa intensiva em conhecimento de classificação com foco na análise de sentimentos. 2015.

CHELLAPPAN, V.; SIVALINGAM, K. M. Security and privacy in the Internet of Things. In: *Internet of Things*. 2016. p. 183-200.

CHUAH, Jun Wei. The Internet of Things: An overview and new perspectives in systems design. In: *2014 International Symposium on Integrated Circuits (ISIC)*. IEEE, 2014. p. 216-219.

CHUI, Michael; LÖFFLER, Markus; ROBERTS, Roger. The internet of things. *McKinsey Quarterly*, v. 2, n. 2010, p. 1-9, 2010.

COMPTON, Michael et al. The SSN ontology of the W3C semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, v. 17, p. 25-32, 2012.

DA XU, Li; HE, Wu; LI, Shancang. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, v. 10, n. 4, p. 2233-2243, 2014.

DESAI, Pratikkumar; SHETH, Amit; ANANTHARAM, Pramod. Semantic gateway as a service architecture for iot interoperability. In: *Mobile Services (MS), 2015 IEEE International Conference on*. IEEE, 2015. p. 313-319.

DEL ÁGUILA, Isabel M.; PALMA, José; TÚNEZ, Samuel. Milestones in software engineering and knowledge engineering history: A comparative review. *The Scientific World Journal*, v. 2014, 2014.

DRESCH, Aline; LACERDA, Daniel Pacheco; JÚNIOR, José Antonio Valle Antunes. *Design science research: método de pesquisa para avanço da ciência e tecnologia*. Bookman Editora, 2015.

EGC. Linhas de Pesquisa [Internet]. [citado em 2019 mar 15]. Disponível em: <<http://www.egc.ufsc.br/pesquisas/linhas-de-pesquisa/>>.

EVANS, Dave. The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, v. 1, n. 2011, p. 1-11, 2011.

EVANS, Dave. The internet of everything: How more relevant and valuable connections will change the world. *Cisco IBSG*, p. 1-9, 2012.

EZECHINA, M. A.; OKWARA, K. K.; UGBOAJA, C. A. U. The Internet of Things (Iot): A Scalable Approach to Connecting Everything. *The International Journal of Engineering and Science*, v. 4, n. 1, p. 09-12, 2015.

FELICÍSSIMO, Carolina Howard et al. Geração de Ontologias Subsidiada pela Engenharia de Requisitos. In: WER. 2003. p. 255-269.

FREMANTLE, Paul. A reference architecture for the internet of things. WSO2 White paper, 2015.

FRIESS, Peter. Driving European internet of things research. RIVER PUBLISHERS SERIES IN COMMUNICATIONS, p. 1-6, 2013.

GÓMEZ-PÉREZ; Asunción; CORCHO, Oscar; FERNANDEZ-LOPEZ, Mariano. Ontological engineering. Computing Reviews 45, v. 8, p. 478-479, 2004.

GYRARD, A.; BONNET, C. Semantic Web best practices: Semantic Web Guidelines for domain knowledge interoperability to build the Semantic Web of Things. 2014.

GYRARD, Amelie; BONNET, Christian; BOUDAUD, Karima. Domain knowledge Interoperability to build the semantic web of things. In: W3C Workshop on the Web of Things. 2014. p. 1-5.

GYRARD, Amelie; SERRANO, Martin. Connected smart cities: Interoperability with seg 3.0 for the internet of things. In: Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on. IEEE, 2016. p. 796-802.

GRUBER, Thomas R. A translation approach to portable ontology specifications. Knowledge acquisition, v. 5, n. 2, p. 199-220, 1993.

GUARINO, Nicola. Formal ontology and information systems. In: Proceedings of FOIS. 1998. p. 81-97.

GUILLEMIN, Patrick et al. Internet of Things Global Standardization—State of Play. Internet of Things—From Research and Innovation to Market Deployment, p. 143-197, 2014.

GUINARD, Dominique; TRIFA, Vlad. Towards the web of things: Web mashups for embedded devices. In: Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain. 2009.

GUINARD, Dominique et al. From the internet of things to the web of things: Resource-oriented architecture and best practices. In: Architecting the Internet of things. Springer, Berlin, Heidelberg, 2011. p. 97-129.

HACHEM, Sara; TEIXEIRA, Thiago; ISSARNY, Valérie. Ontologies for the internet of things. In: Proceedings of the 8th Middleware Doctoral Symposium. ACM, 2011. p. 3.

HENDLER, James; BERNERS-LEE, Tim; MILLER, Eric. Integrating applications on the semantic web. JOURNAL-INSTITUTE OF ELECTRICAL ENGINEERS OF JAPAN, v. 122, n. 10, p. 676-680, 2002.

HENSON, Cory A. et al. SemSOS: Semantic sensor observation service. In: 2009 International Symposium on Collaborative Technologies and Systems. IEEE, 2009. p. 44-53.

HRIBERNIK, Karl A. et al. A service-oriented, semantic approach to data integration for an Internet of Things supporting autonomous cooperating logistics processes. In: Architecting the internet of things. Springer, Berlin, Heidelberg, 2011. p. 131-158.

ITU-T Y.4000/Y.2060, Overview of the Internet of things, 2012. Disponível em: <<https://www.itu.int/rec/T-REC-Y.2060-201206-I>>. Acessado em: agosto de 2018.

ITU-T Y.4100/Y.2066, Common requirements of the Internet of things, 2014. Disponível em: <<https://www.itu.int/rec/T-REC-Y.2066-201406-I>> . Acessado em: agosto de 2018.

JARA, Antonio J. et al. Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence. International Journal of Web and Grid Services, v. 10, n. 2-3, p. 244-272, 2014.

RIBEIRO JUNIOR, Divino Ignácio. Modelo de sistema baseado em conhecimento para apoiar processos de tomada de decisão em ciência e tecnologia. 2010.

KATASONOV, Artem et al. Smart Semantic Middleware for the Internet of Things. *Icinco-Icso*, v. 8, p. 169-178, 2008.

KHODADADI, Farzad; DASTJERDI, Amir Vahid; BUYYA, Rajkumar. Internet of Things: an overview. In: *Internet of Things*. 2016. p. 3-27.

KIRYAKOV, Atanas. Ontologies for knowledge management. *Semantic Web Technologies: Trends and Research in Ontology based Systems*, p. 115-138, 2006.

KNUBLAUCH, Holger. An agile development methodology for knowledge-based systems including a Java framework for knowledge modeling and appropriate tool support. 2002. Tese de Doutorado. Universität Ulm.

LAMHARHAR, Hind et al. A Semantic Meta-Modelling Approach for Smart Government: Service Discovery Based on Conceptual Structures. *International Journal of Conceptual Structures and Smart Applications (IJCSSA)*, v. 4, n. 2, p. 72-93, 2016.

LASSILA, Ora. Using the semantic web in mobile and ubiquitous computing. In: *IFIP Working Conference on Industrial Applications of Semantic Web*. Springer, Boston, MA, 2005. p. 19-25.

LIBELIUM. Libelium Comunicaciones Distribuidas. 2013. Acessado em Novembro 20, 2018 em:
<http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/#show_infographic>

MADAKAM, Somayya; RAMASWAMY, R.; TRIPATHI, Siddharth. Internet of Things (IoT): A literature review. *Journal of Computer and Communications*, v. 3, n. 05, p. 164, 2015.

MANYIKA, James. The Internet of Things: Mapping the value beyond the hype. McKinsey Global Institute, 2015.

MARCH, Salvatore T.; SMITH, Gerald F. Design and natural science research on information technology. *Decision support systems*, v. 15, n. 4, p. 251-266, 1995.

MARTÍNEZ, Diego Jessie. Sistema baseado em conhecimento (SBC) de apoio à capacitação organizacional. 2017.

MCEWEN, Adrian; CASSIMALLY, Hakim. *Designing the internet of things*. John Wiley & Sons, 2013.

MINERVA, Roberto; BIRU, Abyi; ROTONDI, Domenico. Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative*, v. 1, p. 1-86, 2015.

MIRAZ, Mahdi H. et al. A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT). In: *Internet Technologies and Applications (ITA)*, 2015. IEEE, 2015. p. 219-224.

MORGAN, Gareth; SMIRCICH, Linda. The case for qualitative research. *Academy of management review*, v. 5, n. 4, p. 491-500, 1980.

NAZÁRIO, Debora Cabral; DANTAS, Mario Antonio Ribeiro; TODESCO, Jose Leomar. Knowledge engineering: survey of methodologies, techniques and tools. *IEEE Latin America Transactions*, v. 12, n. 8, p. 1553-1559, 2014.

NAZÁRIO, Débora Cabral; DANTAS, Mário Antônio Ribeiro; TODESCO, José Leomar. Modelo de Conhecimento de Qualidade de Contexto para Ambientes Ubíquos baseado em Ontologia. *International Journal of Knowledge Engineering and Management (IJKEM)*, v. 4, n. 9, p. 51-86, 2015.

PEFFERS, Ken et al. A design science research methodology for information systems research. *Journal of management information systems*, v. 24, n. 3, p. 45-77, 2007.

PEREIRA, Larissa Mariany Freiberger et al. OGDPub: Ontologia de Domínio para Publicação de Dados Abertos por Municípios Brasileiros. In: *ONTOBRAS*. 2016. p. 163-168.

PFISTERER, Dennis et al. SPITFIRE: toward a semantic web of things. *IEEE Communications Magazine*, v. 49, n. 11, p. 40-48, 2011.

PIIU, Dan et al. Citypulse: Large scale data analytics framework for smart cities. *IEEE Access*, v. 4, p. 1086-1108, 2016.

ROSE, Karen; ELDRIDGE, Scott; CHAPIN, Lyman. The internet of things: An overview. *The Internet Society (ISOC)*, p. 1-50, 2015.

RIBEIRO, Alessandro Costa. MODELO DE RECONHECIMENTO DE PADRÕES EM IDEIAS USANDO TÉCNICAS DE DESCOBERTA DE CONHECIMENTO EM TEXTOS. 2018. Dissertação de Mestrado. Universidade Federal de Santa Catarina.

SANTOS, Bruno P. et al. Internet das coisas: da teoria a prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2016.

SARKAR, S. Internet of Things—robustness and reliability. In: *Internet of Things*. 2016. p. 201-218.

SASIETA, Héctor Andrés Melgar; BEPPLER, Fabiano Duarte; PACHECO, Roberto Carlos dos Santos. A memória organizacional no contexto da engenharia do conhecimento. *DataGramZero-Revista de Informação*, v. 12, n. 3, 2011.

SCHINDLER, Helen Rebecca et al. Europe's policy options for a dynamic and trustworthy development of the Internet of Things. 2013.

SCHNEIDER, Viviane et al. Método de modelagem do contexto estratégico para sistemas baseados em conhecimento. 2013.

SCHREIBER, Guus. Knowledge engineering. *Foundations of Artificial Intelligence*, v. 3, p. 929-946, 2008.

SERRANO, Martín et al. Internet of Things iot semantic interoperability: Research challenges, best practices, recommendations and next steps. *IERC: European Research Cluster on the Internet of Things*, Tech. Rep, 2015.

SHETH, Amit; HENSON, Cory; SAHOO, Satya S. Semantic sensor web. *IEEE Internet computing*, v. 12, n. 4, 2008.

SILVA, E. L.; MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação: UFSC. 2005.

SRIRAM, Ram D.; SHETH, Amit. Internet of things perspectives. *IT Professional*, v. 17, n. 3, p. 60-63, 2015.

STUDER, Rudi et al. Knowledge engineering: principles and methods. *Data and knowledge engineering*, v. 25, n. 1, p. 161-198, 1998.

SUNDMAEKER, Harald et al. Vision and challenges for realising the Internet of Things. *Strategic Research Agenda, European Commission*, v. 3, n. 3, p. 39-82, 2010.

SURESH, P. et al. A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment. In: *Science Engineering and Management Research (ICSEMR), 2014 International Conference on. IEEE, 2014. p. 1-8.*

SZILAGYI, Ioan; WIRA, Patrice. Ontologies and Semantic Web for the Internet of Things-a survey. *IECON, IEEE, 2016.*

TOMA, Ioan; SIMPERL, Elena; HENCH, Graham. A joint roadmap for semantic technologies and the internet of things. In: *Proceedings of the Third STI Roadmapping Workshop, Crete, Greece. 2009. p. 140-53.*

TORKAMAN, Atefeh; SEYYEDI, M. A. Analyzing IoT Reference Architecture Models. *Int. J. Comput. Sci. Softw. Eng.*, v. 5, n. 8, p. 154-160, 2016.

TSCHOFENIG, H. et al. Architectural considerations in smart object networking. 2015.

UCKELMANN, Dieter; HARRISON, Mark; MICHAHELLES, Florian. An architectural approach towards the future internet of things. In: *Architecting the internet of things. Springer, Berlin, Heidelberg, 2011. p. 1-24.*

UREN, Victoria et al. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: science, services and agents on the World Wide Web*, v. 4, n. 1, p. 14-28, 2006.

VAN HARMELEN, Frank. The semantic web: What, why, how, and when. *IEEE Distributed Systems Online*, v. 5, n. 3, 2004.

VERMESAN, Ovidiu et al. Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends*, v. 1, n. 2011, p. 9-52, 2011.

VERMESAN, Ovidiu et al. Internet of things strategic research and innovation agenda. *RIVER PUBLISHERS SERIES IN COMMUNICATIONS*, p. 7-151, 2013.

WANG, Wei et al. A comprehensive ontology for knowledge representation in the internet of things. In: *Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012 IEEE 11th International Conference on. IEEE, 2012. p. 1793-1798.

W3C OWL WORKING GROUP et al. OWL 2 Web Ontology Language: Document Overview (Second edition). W3C Recommendation, 2012. Disponível em: <<https://www.w3.org/TR/owl2-overview/>>. Acesso em: setembro de 2018.

W3C RDF WORKING GROUP et al. Resource description framework (RDF), 2014. Disponível em: <<http://www.w3.org/RDF/>>. Acesso em: setembro de 2018.

W3C SPARQL WORKING GROUP et al. SPARQL 1.1 Overview. W3C Recommendation. W3C, 2013. Disponível em: <<https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/>>. Acesso em: outubro de 2018.

W3C SSN Ontology: Document Overview. W3C Recommendation, 2017. Disponível em: <<https://www.w3.org/TR/vocab-ssn/>>. Acesso em: setembro de 2018.

WEYRICH, Michael; EBERT, Christof. Reference architectures for the internet of things. *IEEE Software*, v. 33, n. 1, p. 112-116, 2016.

WU, Weigang; YANG, Zhiwei; LI, Keqin. Internet of Vehicles and applications. In: *Internet of Things*. 2016. p. 299-317.

ZEINAB, Kamal Aldein Mohammed; ELMUSTAFA, Sayed Ali Ahmed. Internet of Things applications, challenges and related future technologies. *World Scientific News*, v. 2, n. 67, p. 126-148, 2017.

APÊNDICE A – Inicializador API

```

var express = require('express');
var path = require('path');

var bodyParser = require('body-parser');

//Routes created to WebAPI
var registerRoutes = require('./services/register');
var authenticationRoutes = require('./services/authentication');
var deviceDataRoutes= require('./services/device-data');

var cookieParser = require('cookie-parser');
var app = express();

app.use(function(req, res, next) {
    res.setHeader('Access-Control-Allow-Headers', "X-
Requested-With,Content-Type,Authorization");
    res.setHeader('Access-Control-Allow-Origin', '*');
    res.setHeader('Access-Control-Allow-Methods',
    'GET,PUT,POST,DELETE,PATCH,OPTIONS');
    next();
});

app.use(bodyParser.json({ limit: '10mb' }));
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cookieParser());

app.set('view engine', 'html');
app.engine('html', require('ejs').renderFile);

// Rotas
app.use('/api/register', registerRoutes);
app.use('/api/authentication', authenticationRoutes);
app.use('/api/device-data', deviceDataRoutes);
// Inicial page from WebAPI
var catalogRoute = express.Router().get('/', function(req, res, next) {
    res.send(
        <h1>WEB - API</h1><p/>

```

```
<h2>CATÁLOGO DE RECURSOS:</h2>
<p/>
<ul>
  <a href="/api/register"><li>/api/register</li></a>
  <a href="/api/authentication"><li>/api/authentication</li></a>
  <a href="/api/device-data"><li>/api/device-data</li></a>
</ul>
  `);
});
app.use('/', catalogRoute);
// catch 404 and forward to error handler
app.use(function(req, res, next) {
  var err = new Error('Not Found');
  err.status = 404;
  next(err);
});
// error handlers
app.use(function(err, req, res, next) {
  res.status(err.status || 500);
  res.render('error', {
    message: err.message,
    error: {}
  });
});
module.exports = app;
```


APÊNDICE B – Serviço web de registro

```

var express = require("express");
var router = express.Router();
const { Client, Node, Text, Data, Triple } = require('virtuoso-sparql-client');
const endpointSPARQL = new Client('http://localhost:8890/sparql');

//URL que o dispositivo vai receber para
const urlSendData = "http://localhost:8890/device-metadata";
router.post('/', registerDevice);

/* Função de registro do dispositivo IoT */
function registerDevice(req, res, next) {
  endpointSPARQL.setOptions(
    "application/json",
    {
      //SSN ontology
      "ssn": "http://www.w3.org/ns/ssn/",
//SOSA ontology
      "sosa": "http://www.w3.org/ns/sosa/"
    },
    "http://localhost:8890/device-metadata"
  );
  var deviceMetadata = req.body;
  var tripleRDF = [];
  //Plataforma - IoT Simulator
  var iotPlatform = new Node('iot-simulator');
  //Entidade de observação
  var featureOfInterest = new Node("veiculo");
  //Validação do objeto JSON recebido
  if (deviceMetadata.code) {
    //URI: Código do equipamento
    var deviceResource = "dispositivo/" + deviceMetadata.code;
    var deviceNode = new Node(deviceResource);
    //URI: Placa do veículo
    var vehicleNode = new Node("veiculo/" + deviceMetadata.code);
    //Sensors
  }
}

```

```

    var velocitySensor = new Node(deviceResource +
"/sensor/velocimetro");
    var odometerSensor = new Node(deviceResource +
"/sensor/odometro");
    var temperatureSensor = new Node(deviceResource +
"/sensor/temperaturaMotor");
    //Properties
    var velocityProperty = new Node("velocidade");
    var odometerProperty = new Node("kilometragem"); //odometro
    var temperatureProperty = new Node("temperatura");

/*****
 * Anotações semânticas e Geração de triplas RDF *
 *****/

//Instância do dispositivo IoT e veículo
tripleRDF.push( new Triple(deviceNode, "rdf:type", "sosa:Platform" ) );
tripleRDF.push( new Triple(featureOfInterest, "rdf:type",
"sosa:FeatureOfInterest" ) );
tripleRDF.push( new Triple(vehicleNode, "rdf:type",
"sosa:FeatureOfInterest" ) );
tripleRDF.push( new Triple(deviceNode, "rdf:type",
"sosa:FeatureOfInterest" ) );
tripleRDF.push( new Triple(deviceNode, "owl:sameAs",
vehicleNode) );

//Instância dos Sensores & Atuadores e suas respectivas propriedades
tripleRDF.push( new Triple(velocitySensor, "rdf:type", "sosa:Sensor" ) );
tripleRDF.push( new Triple(velocityProperty, "rdf:type",
"sosa:ObservableProperty" ) );

tripleRDF.push( new Triple(odometerSensor, "rdf:type",
"sosa:Sensor" ) );
tripleRDF.push( new Triple(odometerProperty, "rdf:type",
"sosa:ObservableProperty" ) );
tripleRDF.push( new Triple(temperatureSensor, "rdf:type",
"sosa:Sensor" ) );
tripleRDF.push( new Triple(temperatureProperty, "rdf:type",
"sosa:ObservableProperty" ) );
//Vinculo com a Plataforma IoT
tripleRDF.push( new Triple(iotPlatform, "sosa:hosts", deviceNode)
);

```

```

tripleRDF.push( new Triple(vehicleNode,
"ssn:deployedOnPlatform", deviceNode) );
tripleRDF.push( new Triple(vehicleNode, "sosa:isSampleOf",
featureOfInterest) );

//Relações entre Sensores & Atuadores com o Dispositivo IoT
tripleRDF.push( new Triple(velocitySensor, "sosa:isHostedBy",
deviceNode) );
tripleRDF.push( new Triple(odometerSensor, "sosa:isHostedBy",
deviceNode) );
tripleRDF.push( new Triple(temperatureSensor,
"sosa:isHostedBy", deviceNode) );

// //DeviceIoT > Hosts > Sensors
tripleRDF.push( new Triple(deviceNode, "sosa:hosts",
velocitySensor) );
tripleRDF.push( new Triple(deviceNode, "sosa:hosts",
odometerSensor) );
tripleRDF.push( new Triple(deviceNode, "sosa:hosts",
temperatureSensor) );

//FeatureOfInterest > isObservedBy > Sensors
tripleRDF.push( new Triple(vehicleNode, "sosa:isObservedBy",
velocitySensor) );
tripleRDF.push( new Triple(vehicleNode, "sosa:isObservedBy",
odometerSensor) );
tripleRDF.push( new Triple(vehicleNode, "sosa:isObservedBy",
temperatureSensor) );

//Relações entre Sensores e Propriedades de observação
tripleRDF.push( new Triple(velocitySensor, "sosa:observes",
velocityProperty) );
tripleRDF.push( new Triple(odometerSensor, "sosa:observes",
odometerProperty) );
tripleRDF.push( new Triple(temperatureSensor, "sosa:observes",
temperatureProperty) );

endpointSPARQL.getLocalStore().bulk(tripleRDF);
endpointSPARQL.store(true)

```

```
.then((result)=>{
    //Retorna credenciais de acesso
    var credenciais = {
        username:deviceMetadata.code,
password: 'security',
        url: 'http://localhost:4200/authentication/login'

    };
    console.log(JSON.stringify(result));
    //montar JSON
    res.send(credenciais);
})
.catch((err) => {
    console.log(err);
    res.send('ERRO: ' + JSON.stringify(err));
});
} else {
    res.send('ERRO: Objeto inválido, verificar estrutura do JSON
enviado');
}
}
module.exports = router;
```

APÊNDICE C – Serviço web de autenticação

```

var express = require("express");
var router = express.Router();
var jwt = require('jsonwebtoken');
var passport = require("passport");
var passportJWT = require("passport-jwt");
var resourceSendData = "http://localhost:4200/device-data";
var ExtractJwt = passportJWT.ExtractJwt;
var JwtStrategy = passportJWT.Strategy;
var jwtOptions = { }
jwtOptions.jwtFromRequest =
ExtractJwt.fromAuthHeaderAsBearerToken();
jwtOptions.secretOrKey = 'iotSimulatorPlatformKey';
var strategy = new JwtStrategy(jwtOptions, function(jwt_payload, next)
{
    var user = { username: jwt_payload.username, resource:
resourceSendData };
    next(null, user, jwt_payload);
});
passport.use(strategy);
router.get("/", function(req, res) {
console.log('chegou Login ');
res.json({ message: "Express is up!" });
});
router.post('/login', login);
/* Handlers functions */
function login(req, res, next) {
var username = req.body.username;
var password = req.body.password;
    if (username && password) {
        var token = jwt.sign({ username: username },
jwtOptions.secretOrKey);
        res.status(202).json({ message: "ok", user: { dataset:
resourceSendData, token: token } });
    } else {
        res.status(401).json({ message: "usuário ou senha incorreto" });
    }
}
module.exports = router;

```


APÊNDICE D – Serviços web de envio e de obtenção de dados

```

var express = require("express");
var router = express.Router();
var passport = require("passport");
var bodyParser = require('body-parser');
var app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

const {Client, Node, Text, Data, Triple} = require('virtuoso-sparql-client');
const endpointSPARQL = new Client('http://localhost:8890/sparql');

router.get('/:codigo', getDeviceData);
router.post('/', passport.authenticate('jwt', { session: false }),
insertDeviceData);

//Properties
var velocityProperty = new Node("velocidade");
var odometerProperty = new Node("kilometragem"); //odometro
var temperatureProperty = new Node("temperatura");

/* Handlers functions */
function getDeviceData(req, res, next) {
var deviceCode = req.params.codigo;
endpointSPARQL.setOptions(
  "application/json",
  {
    //SSN ontology
    "ssn": "http://www.w3.org/ns/ssn/",
    //SOSA ontology
    "sosa": "http://www.w3.org/ns/sosa/"
  },
  "http://localhost:8890/device-metadata"
);

var query = `

```

```

SELECT ?dispositivo, ?valor, ?property, ?sensor, ?date WHERE {
?dispositivo
<http://www.w3.org/ns/sosa/hasFeatureOfInterest><dispositivo/ +
deviceCode + > ;
<http://www.w3.org/ns/sosa/hasSimpleResult> ?valor ;
<http://www.w3.org/ns/sosa/observedProperty> ?property ;
<http://www.w3.org/ns/sosa/madeBySensor> ?sensor ;
<http://www.w3.org/ns/sosa/resultTime> ?date .

```

```

};

```

```

//Exemplo para incluir filtro de data
//FILTER (?date >= "2019-01-
20T10:10:00.000Z"^^xsd:dateTimeStamp)
endpointSPARQL.query(query, true)
  .then((results) => {
    console.log(JSON.stringify(results));
    res.send(results);
  })
  .catch((err) => {
    console.log(err);
  });
}
/* Handlers functions */
function insertDeviceData(req, res, next) {
  endpointSPARQL.setOptions(
    "application/json",
    {
      //SSN ontology
      "ssn": "http://www.w3.org/ns/ssn/",
      //SOSA ontology
      "sosa": "http://www.w3.org/ns/sosa/"
    },
    "http://localhost:8890/device-metadata"
  );
  //var deviceMetadata = req.body;
  var tripleRDF = [];
  //código do dispositivo
  var id = req.user.username;
  //JSON recebido
  var data = req.body;

```



```

//datetime e código da observação realizada
var resultTime = new Date(data.datetime).toISOString();
/*****
 * Anotações semânticas e Geração de triplas RDF *
*****/
var deviceResource = "dispositivo/" + id;
var deviceNode = new Node(deviceResource);
//Sensors
var velocityDeviceSensor = new Node(deviceResource +
"/sensor/velocimetro");
var odometerDeviceSensor = new Node(deviceResource +
"/sensor/odometro");
var temperatureDeviceSensor = new Node(deviceResource +
"/sensor/temperaturaMotor");
var velocityObservation = new Node(deviceResource +
"/sensor/velocimetro/observacao/" + resultTime);
var odometerObservation = new Node(deviceResource +
"/sensor/odometro/observacao/" + resultTime);
var temperatureObservation = new Node(deviceResource +
"/sensor/temperaturaMotor/observacao/" + resultTime);
//Dados observados pelo dispositvo IoT
tripleRDF.push( new Triple(velocityObservation, "sosa:resultTime",
new Data(resultTime, "xsd:dateTimeStamp") ) );
tripleRDF.push( new Triple(velocityObservation,
"sosa:hasSimpleResult", new Data(data.velocity, "xsd:decimal" ) ) );
tripleRDF.push( new Triple(velocityObservation,
"sosa:observedProperty", velocityProperty ) );
tripleRDF.push( new Triple(velocityObservation,
"sosa:madeBySensor", velocityDeviceSensor ) );
tripleRDF.push( new Triple(velocityObservation,
"sosa:hasFeatureOfInterest", deviceNode ) );
tripleRDF.push( new Triple(odometerObservation, "sosa:resultTime",
new Data(resultTime, "xsd:dateTimeStamp") ) );
tripleRDF.push( new Triple(odometerObservation,
"sosa:hasSimpleResult", new Data(data.odometer, "xsd:decimal") ) );
tripleRDF.push( new Triple(odometerObservation,
"sosa:observedProperty", odometerProperty ) );
tripleRDF.push( new Triple(odometerObservation,
"sosa:madeBySensor", odometerDeviceSensor ) );

```

```

    tripleRDF.push( new Triple(odometerObservation,
"sosa:hasFeatureOfInterest", deviceNode) );
    tripleRDF.push( new Triple(temperatureObservation,
"sosa:resultTime", new Data(resultTime, "xsd:dateTimeStamp") ) );
    tripleRDF.push( new Triple(temperatureObservation,
"sosa:hasSimpleResult", new Data(data.temperature, "xsd:decimal") ) );
    tripleRDF.push( new Triple(temperatureObservation,
"sosa:observedProperty", temperatureProperty) );
    tripleRDF.push( new Triple(temperatureObservation,
"sosa:madeBySensor", temperatureDeviceSensor) );
    tripleRDF.push( new Triple(temperatureObservation,
"sosa:hasFeatureOfInterest", deviceNode) );
    endpointSPARQL.getLocalStore().bulk(tripleRDF);
    endpointSPARQL.store(true)
    .then((result)=> {
        console.log(JSON.stringify(result));
        res.send("OK");
    })
    .catch((err) => {
        console.log(err);
        res.send('ERRO: ' + JSON.stringify(err));
    }); } module.exports = router;

```

ANEXO A – Lista de requisitos comuns para IoT

Abaixo consta uma tradução realizada pelo autor, referente a lista de requisitos comuns para IoT da recomendação ITU-T Y.4100/Y.2066 (2014, p. 10-19).

Número do requisito	Categoria de exigência	Descrição do requisito	Resumo do requisito
N1	Não funcional	A interoperabilidade é necessária para ser assegurada entre implementações em ambientes heterogêneas de IoT.	Interoperabilidade
N2	Não funcional	Escalabilidade é necessária para ser suportado em IoT, a fim de lidar com uma grande quantidade de dispositivos, aplicativos e usuários.	Escalabilidade
N3	Não funcional	Confiabilidade nas capacidades da IoT, como confiabilidade nas capacidades de comunicação, serviço e gerenciamento de dados da IoT.	Confiabilidade
N4	Não funcional	A IoT é necessária para fornecer alta disponibilidade em serviço provisionamento, gerenciamento de dados, comunicação, sensoriamento e acionamento de coisas.	Alta disponibilidade

N5	Não funcional	Adaptabilidade às novas tecnologias emergentes no futuro é necessária em IoT	Adaptabilidade
N6	Não funcional	A capacidade de gerenciamento precisa ser suportada em IoT para garantir operações normais	Capacidade de gerenciamento
A1	Suporte à aplicação	Interfaces programáveis precisam ser padronizadas para fornecer acesso aberto aos recursos de suporte a aplicativos	Interfaces programáveis padronizadas
A2	Suporte à aplicação	O gerenciamento de grupo, incluindo exibição, criação, modificação, exclusão de grupos de IoT e exibição, adição, modificação, exclusão de membros do grupo de IoT, é necessário para ser suportado em IoT.	Gerenciamento de grupo .
A3	Suporte à aplicação	Para suportar a marcação de hora global em IoT, é necessária uma sincronização de tempo confiável.	Sincronização de tempo confiável
A4	Suporte à aplicação	A colaboração entre serviços ou entre dispositivos com o mesmo objetivo de acessar aplicativos IoT é necessária	Colaboração
A5	Suporte à aplicação	O gerenciamento de usuários é necessário, incluindo criação, autenticação, autorização e contabilidade de usuários de IoT	Gerenciamento de usuários

A6	Aplicação	A contabilidade do uso de recursos de IoT é necessária para cada aplicação	Contabilidade de uso de recursos
S1	Serviço	A priorização de serviços é necessária para satisfazer os diferentes requisitos de serviço de diferentes grupos de usuários de IoT.	Priorização de serviços
S2	Serviço	Serviços baseados em semântica são necessários na IoT para suportar o provisionamento de serviços autônômicos	Serviços baseados em semântica
S3	Serviço	A composição do serviço é necessária para suportar a criação flexível de serviços em IoT	Composição do serviço
S4	Serviço	Serviços de mobilidade são necessários, para que a IoT possa suportar mobilidade de serviços, mobilidade de usuários e mobilidade de dispositivos.	Serviços de mobilidade
S5	Serviço	Alta confiabilidade e segurança são necessárias quando os serviços de conectividade do corpo humano são fornecidos.	Serviços de conectividade do corpo humano altamente confiáveis e seguros.

S6	Serviço	Serviços autônomos são necessários, para que a IoT possa ativar captura automática, comunicação e processamento de dados de coisas com base em regras configuradas por provedores de serviços ou customizadas por usuários de IoT.	Serviços autônomos
S7	Serviço	Serviços baseados em local e contextaware são necessários, para que a IoT permite serviços flexíveis, personalizados e autônomos, com base nas informações de localização e no contexto relacionado das coisas e / ou usuários.	Serviços baseados em localização e com reconhecimento de contexto
S8	Serviço	O gerenciamento de serviços é necessário para que o provisionamento de serviços possa ser suportado de maneira altamente disponível e confiável	Gestão de serviços
S9	Serviço	Serviços de descoberta são necessários, para que os usuários, serviços e dispositivos e dados de coisas podem ser descobertos por provedores de serviços ou usuários de IoT.	Descoberta de serviços
S10	Serviço	O suporte à assinatura de serviço é necessário, para que a IoT possa fornecer um meio para permitir que o usuário da IoT assine os serviços necessários e os	Suporte de assinatura de serviço

		dados associados das coisas	
S11	Serviço	Necessidade de nomenclatura e endereçamento padronizado de serviços e das coisas	Nomenclatura e endereçamento padronizado
S12	Serviço	Para armazenar e processar uma grande quantidade de dados (big data), são necessários recursos de armazenamento virtual e processamento	Recursos de armazenamento e processamento virtuais
C1	Comunicação	A IoT é necessária para suportar comunicações baseadas em eventos, periódicas e automáticas entre dispositivos ou entre usuários de IoT.	Modos de comunicação baseados em eventos, periódicos e automáticos
C2	Comunicação	O suporte do modo de comunicação unicast é necessário (por exemplo, para comunicações entre usuários ou dispositivos de IoT). O suporte do multicast, broadcast e anycast modos de comunicação são necessários, para que a IoT possa fornecer vários serviços de comunicação dentro de um grupo de usuários ou dispositivos de IoT (por exemplo, para dar suporte à colaboração entre usuários ou dispositivos de IoT).	Suporte aos modos de comunicação: unicast, multicast, broadcast e anycast.

C3	Comunicação	O suporte de comunicações iniciadas pelo dispositivo é necessário satisfazer os requisitos das comunicações automáticas	Suporte de comunicações iniciadas pelo dispositivo
C4	Comunicação	O controle de erros para comunicações é necessário, para que a IoT seja capaz, por exemplo, de lidar com interferências entre dispositivos.	Controle de erros para comunicações
C5	Comunicação	A IoT é necessária para fornecer tratamento e entrega de mensagens com tempo crítico.	Comunicações de tempo crítico
C6	Comunicação	Os recursos de autoconfiguração, autocorreção, auto-otimização e autoproteção no nível da rede são necessários na IoT.	Rede autônoma
C7	Comunicação	A comunicação sensível ao conteúdo é necessária para que, por exemplo, a IoT pode fornecer um suporte para seleção de caminho / e roteamento de comunicações baseadas em conteúdo.	Comunicação com reconhecimento de conteúdo
C8	Comunicação	A IoT é necessária para suportar interações baseadas em localização entre os atores da IoT.	Comunicação baseada em localização

C9	Comunicação	As comunicações podem ocorrer na camada do dispositivo [ITU-T Y.2060] através de vários tipos de tecnologias com ou sem fio, como o barramento da rede de área do controlador (CAN), ZigBee, Bluetooth, WiFi, etc	Suporte para tecnologias de comunicação relacionadas a dispositivos heterogêneos
C10	Comunicação	As comunicações podem ocorrer na camada de rede [ITU-T Y.2060] através de vários tipos de tecnologias, como o segundo geração / terceira geração (2G / 3G), evolução a longo prazo (LTE), Ethernet, linha de assinante digital (DSL), etc.	Suporte para tecnologias de comunicação relacionadas à rede heterogênea.
D1	Dispositivo	A IoT é necessária para apoiar o estabelecimento da conectividade entre uma coisa e a IoT com base no identificador da coisa.	A conectividade baseada em identificação entre a coisa e a IoT
D2	Dispositivo	Suporte de monitoramento remoto, controle e configuração de dispositivos é necessário para que a capacidade de gerenciamento do dispositivo em IoT seja aumentada.	Monitoramento remoto, controle e configuração de dispositivos
D3	Dispositivo	O recurso Plug and Play é necessário para ser suportado no IoT a fim de permitir configurações de	Capacidade plug and play

		dispositivos baseados em semântica em tempo real.	
D4	Dispositivo	A notificação automática do status das coisas e suas alterações é necessária para monitorar as coisas em tempo hábil.	Monitorar as coisas em tempo hábil
D5	Dispositivo	A IoT é necessária para suportar a mobilidade das coisas.	Mobilidade do dispositivo
D6	Dispositivo	A verificação de integridade do dispositivo é necessária para suportar alta disponibilidade de dispositivos	Verificação de integridade do dispositivo
DM1	Gestão de dados	A IoT é necessária para suportar o armazenamento de dados das coisas com base em regras e políticas predefinidas.	Suporte ao armazenamento de dados das coisas
DM2	Gestão de dados	A fusão e mineração de dados com base em regras e políticas predefinidas precisam ser suportadas.	Suporte ao processamento dos dados das coisas
DM3	Gestão de dados	A IoT é necessária para fornecer informações históricas sobre as coisas	Consulta de dados históricos
DM4	Gestão de dados	O controle de acesso dos dados por seus proprietários deve ser suportados em IoT, para que os usuários de IoT possam ter a capacidade de controlar como seus dados	O controle de acesso a dados pelos proprietários dos dados

		são expostos a outros usuários de IoT	
DM5	Gestão de dados	A IoT é necessário fornecer acesso a fontes de dados externas, por exemplo, bancos de dados de saúde, fora da IoT.	Troca de dados com fontes externas
DM6	Gestão de dados	A IoT é necessária para fornecer verificação de integridade e gerenciamento do ciclo de vida dos dados, para que a IoT seja capaz de fornecer alta disponibilidade e confiabilidade dos dados das coisas.	Verificação de integridade e gerenciamento do ciclo de vida dos dados das coisas
DM7	Gestão de dados	Anotação semântica dos dados e o acesso semântico aos dados das coisas é necessário para que possa ser suportada consultas automáticas.	Anotação semântica e acesso semântico aos dados
DM8	Gestão de dados	O armazenamento, a transferência e a agregação de dados de coisas precisam ser executados automaticamente de acordo com os requisitos dos usuários ou aplicativos da IoT	Armazenamento semântico, transferência e agregação de dados

SP1	Segurança e proteção de privacidade	A IoT é necessária para oferecer suporte à capacidade de comunicação segura, confiável e protegida por privacidade.	Segurança na comunicação
SP2	Segurança e proteção de privacidade	A IoT é necessária para fornecer recursos de gerenciamento de dados seguros, confiáveis e protegidos por privacidade.	Segurança no gerenciamento
SP3	Segurança e proteção de privacidade	A IoT é necessária para fornecer capacidade de prestação de serviços protegida, confiável e protegida por privacidade	Segurança no fornecimento de serviços
SP4	Segurança e proteção de privacidade	A integração de diferentes políticas e técnicas de segurança relacionadas à variedade de dispositivos e redes de usuários na IoT é necessária	Integração de diferentes políticas de segurança e tecnologias relacionadas
SP5	Segurança e proteção de privacidade	Antes que um dispositivo (ou um usuário de IoT) possa acessar a IoT, a autenticação mútua e a autorização são necessárias de acordo com as políticas de segurança predefinidas.	Autenticação e autorização mútua
SP6	Segurança e proteção de privacidade	Qualquer acesso a dados ou tentativa de acessar aplicativos IoT deve ser totalmente transparente, rastreável e reproduzível de acordo com as leis e regulamentações apropriadas.	Auditoria de segurança