

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA  
E TECNOLOGIA FLUMINENSE**

**PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À  
ENGENHARIA E GESTÃO**

**GEOVANA DA SILVA BATISTA**

**REPRESENTAÇÃO SEMÂNTICA FORMAL DE INTERFACE E USO  
DE SENSORES E ATUADORES**

**Campos dos Goytacazes/RJ  
2019**

# **INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FLUMINENSE**

**PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À  
ENGENHARIA E GESTÃO**

**GEOVANA DA SILVA BATISTA**

**REPRESENTAÇÃO SEMÂNTICA FORMAL DE INTERFACE E USO  
DE SENSORES E ATUADORES**

Mark Douglas de Azevedo Jacyntho  
(Orientador)

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

Campos dos Goytacazes, RJ  
Junho 2019

Biblioteca Anton Dakitsch  
CIP - Catalogação na Publicação

B333r      Batista, Geovana da Silva  
             Representação semântica formal de interface e uso de sensores e  
             atuadores / Geovana da Silva Batista - 2019.  
             105 f.: il. color.

             Orientador: Mark Douglas de Azevedo Jacyntho

             Dissertação (mestrado) -- Instituto Federal de Educação, Ciência e  
             Tecnologia Fluminense, Campus Campos Centro, Curso de Mestrado  
             Profissional em Sistemas Aplicados à Engenharia e Gestão, Campos dos  
             Goytacazes, RJ, 2019.  
             Referências: f. 93 a 99.

             1. Internet das Coisas. 2. Web Semântica. 3. Linked Data. I. Jacyntho,  
             Mark Douglas de Azevedo, orient. II. Título.

**Instituto Federal de Educação, Ciência e Tecnologia Fluminense**  
**Programa de Pós-graduação em Sistemas Aplicados à Engenharia e Gestão**

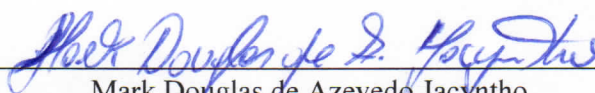
**GEOVANA DA SILVA BATISTA**

**REPRESENTAÇÃO SEMÂNTICA FORMAL DE INTERFACE E USO  
DE SENSORES E ATUADORES**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

APROVADA EM 26/06/2019.

Banca Examinadora:



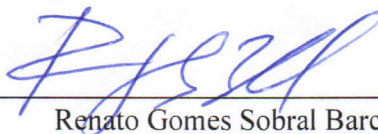
---

Mark Douglas de Azevedo Jacyntho  
(Doutor em Informática) IF Fluminense  
(Orientador)



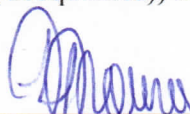
---

Adriana Pereira de Medeiros  
(Doutora em Informática) UFF



---

Renato Gomes Sobral Barcellos  
(Doutor em Geociências (Geoquímica)) IF Fluminense



---

Luiz Gustavo Lourenço Moura  
(Doutor em Engenharia de Sistemas e Computação) IF Fluminense



## AGRADECIMENTOS

Em 1 Tessalonicenses 5:18, diz que em tudo devemos dar Graças, pois é a vontade de Deus em Cristo Jesus para conosco. Primeiramente, meu agradecimento a Deus, meu Senhor e Amigo fiel. Por toda bênção em forma de aprendizado, esforço, sustento, capacitação e alegria. A ti toda honra e glória para sempre.

Aos meus pais queridos, por ser fazerem presentes, me auxiliando, encorajando, intercedendo, me aconselhando sabiamente em todos os momentos. Por meio do exemplo deles aprendi a agradecer, confiar, lutar, cair, levantar, perder e vencer.

Aos meus irmãos Gilson Junior e Gisela Batista. Ele que é tão cheio do amor de Deus, cuidando de cada etapa para que não me faltasse nada, me cobrindo de incentivo e amor. Ela tão decidida, amiga, me fazendo acreditar a cada instante que os sonhos são possíveis de realizar. A sua doença, não foi mais forte que nosso amor. Hoje, podemos juntos, meus irmãos, comemorar a conclusão deste trabalho.

Aos meus filhos amados, Antonella e Bernardo. Pelas madrugadas em que acordavam e me diziam: Mãe, vai dar tudo certo! Pelo tudo bem mãe, quanto aos passeios, filmes, e festas interrompidos. Vocês me enchem de vida, de amor, de orgulho, coragem e determinação.

À minha querida cunhada Monica e meu querido cunhado André que sempre me apoiaram e incentivaram, deixando minha caminhada mais suave.

As minhas sobrinhas Sophia e Valentina, pela graciosidade, pelo carinho e doçura quando a tia precisava.

Ao professor Mark Douglas Jacyntho pela oportunidade de crescimento intelectual e pessoal, me ofertando gentilmente, generosamente e incansavelmente, seus ensinamentos e suas imprescindíveis contribuições, permitindo a realização deste trabalho, bem como me ensinando a ser uma profissional melhor. Meu sincero, obrigada.

À querida Tamiris de Souza Rangel, pelo apoio, empatia e incentivo, concedendo gentilmente seu trabalho para as minhas pesquisas.

Aos meus amigos, fiéis, que a cada passo do meu caminhar, estão como anjos a me ajudar. Em especial aos meus queridos Ranielli, Leandro Arêas, Janaina, Rejane, Ralph, Cleber, Luciana, Everton (Veveto, sem sua carona, não seria possível realizar algumas disciplinas obrigatórias do curso). Há tantos outros amigos que pelo conhecimento técnico (Anderson Veiga) ou incentivos e/ou orações (Michelle Freitas, Beth Rocha, Maria Eugênia, Bruna Pessanha, Camila, Raisa, Frederico, André, Fabrício, Áquila, Wesley, Ianne, Garcez, Anderson Lima, Dilmar, Edina) fizeram parte da caminhada.

Por fim, a todos que contribuíram de alguma forma e que torceram pelo meu sucesso na realização deste trabalho que representa uma etapa importante da minha formação.

“Não fui eu que ordenei a você? Seja forte e corajoso! Não se apavore nem desanime, pois o Senhor, o seu Deus, estará com você por onde você andar”.

Josué 1:9

## RESUMO

No contexto da computação pervasiva ou ubíqua, surge sua mais nova geração conhecida por IoT (*Internet of Things* ou Internet das Coisas, em português). IoT corresponde a interoperabilidade entre sensores e atuadores, denominados "coisas", onipresentes em infraestruturas, equipamentos, telefones celulares, redes de sensores sem fio, cidades inteligentes, etc. Não obstante, nos dias de hoje, o verdadeiro potencial desta plethora de dispositivos inteligentes é extremamente subexplorado. Devido à enorme quantidade de dados de sensores produzidos, há uma necessidade real de que estes dados estejam acessíveis na Web e que agentes de software consigam interpretá-los. O conceito de WoT (*Web of Things* ou Web das Coisas, em português) vem como extensão natural da IoT, possibilitando a integração das coisas inteligentes na Web. Porém, a medida que a tecnologia se consolida, mais e mais coisas inteligentes são conectadas e disponibilizadas na Web. Deste modo, torna-se um grande desafio pesquisar e encontrar tais coisas para poder então utilizá-las. Como primeiro passo, temos que catalogar e publicar a descrição ou especificação dessas coisas inteligentes em uma Web Semântica inteligível por máquinas, usando um vocabulário formal (ontologia) comum. Entretanto, não basta somente ter esses exemplares catalogados, com suas especificações (tipo, serviço oferecido, fabricante, localização física, etc.). Uma vez encontrada a coisa inteligente na Web, é preciso saber como usá-la. Para tal, como segundo passo, temos que descrever explicitamente a interface de acesso do dispositivo. O objetivo primário deste trabalho é justamente contemplar este segundo passo. Em outras palavras, propor uma abordagem para descrever formalmente a interface de acesso de sensores e atuadores (parâmetros, protocolos de acesso, formatos de dados, etc.), utilizando as tecnologias e padrões da Web Semântica, propostos pelo consórcio W3C. A fim de corroborar a proposta, desenvolver também uma aplicação semântica *Linked Data*, cuja informação publicada seja compreensível tanto por agentes computacionais quanto por humanos, contribuindo, pois, para a construção da crescente Web Semântica das Coisas (SWoT).

**Palavras-chave:** Internet das Coisas, Web Semântica, *Linked Data*.

## ABSTRACT

In the context of pervasive or ubiquitous computing, emerges its newest generation known as IoT (Internet of Things or Internet das Coisas, in Portuguese). IoT corresponds to the interoperability between sensors and actuators, called "things", omnipresent in infrastructures, equipment, cell phones, wireless sensor networks, intelligent cities, etc. Nevertheless, today the true potential of this plethora of devices is extremely underexploited. Due to the huge amount of sensor data produced, there is a real need for this data to be accessible on the Web and software agents to be able to interpret it. The concept of WoT (Web of Things) comes as a natural extension of IoT, enabling the integration of smart things on the Web. But as technology consolidates, more and more smart things are connected and available on the Web. In this way, it becomes a great challenge to search and find such things so that you can use them. As a first step, we have to catalog and publish the description or specification of these intelligent things in a machine-intelligible Semantic Web using a common formal vocabulary (ontology). However, it is not enough to have these catalogs, with their specifications (type, service offered, manufacturer, physical location, etc.). Once you find the smart thing on the Web, you need to know how to use it. For this, as a second step, we must explicitly describe the access interface of the device. The primary objective of this research is to address this second step. In other words, propose an approach to formally describe the access interface of sensors and actuators (parameters, access protocols, data formats, etc.), using the technologies and standards of Semantic Web, proposed by the W3C consortium. In order to corroborate the proposal, we also develop a *Linked Data* semantic application, whose published information is understandable by both computational and human agents, thus contributing to the construction of the growing Semantic Web of Things (SWoT).

**Keywords:** Internet of Things, Semantic Web, *Linked Data*.

## LISTA DE FIGURAS

Figura 1	Exemplo de uma tripla RDF.....	22
Figura 2	Grafo RDF com três triplas de um mesmo recurso.....	23
Figura 3	Grafo RDF relacionando dois recursos.....	23
Figura 4	Grafo RDF relacionando um recurso a um blank node.....	24
Figura 5	Semântica baseada em RDF usada em OWL.....	27
Figura 6	Exemplo em sintaxe RDF/XML do uso da propriedade owl:sameAs.....	28
Figura 7	Linked Open Data Cloud Diagram.....	30
Figura 8	Exemplo de tripla RDF.....	31
Figura 9	Exemplo de consulta SPARQL.....	32
Figura 10	Consulta SPARQL com mais de um resultado.....	33
Figura 11	Trajectoria metodológica.....	36
Figura 12	Gráfico de Venn da Análise Bibliográfica.....	39
Figura 13	Aspectos da Descrição de sensores e atuadores.....	41
Figura 14	Macro-arquitetura da aplicação.....	46
Figura 15	Cenário retratando as questões para descoberta de uma “coisa inteligente”.....	47
Figura 16	Características de uma Thing Description.....	48
Figura 17	Arquitetura abstrata do W3C WoT.....	49
Figura 18	Visão geral da ontologia WoT.....	51
Figura 19	Modelo para descrição de Segurança.....	54
Figura 20	Modelo para descrição de protocolo.....	55
Figura 21	Modelo para descrição do sensor e de sua Interface de acesso.....	56
Figura 22	Dados de localização de sensor.....	57
Figura 23	Modelo para descrição do Padrão de Interação – Property.....	58
Figura 24	Modelo para descrição de Link.....	60
Figura 25	Modelo para descrição do DataSchema.....	61

Figura 26	Modelo para descrição do Padrão de Interação – Action.....	62
Figura 27	Modelo para descrição do Padrão de Interação – Event.....	63
Figura 28	Modelo para descrição do atuador e de sua Interface de acesso.....	65
Figura 29	Funcionalidades atualizadas da Aplicação.....	66
Figura 30	Elementos que compõem o cenário IoT.....	67
Figura 31	Exemplo de criação de cadastro de Segurança.....	68
Figura 32	Exemplo de edição do cadastro de Segurança.....	69
Figura 33	Grafo RDF subjacente ao cadastro de Segurança.....	69
Figura 34	Exemplo de criação do cadastro de Protocolo de Comunicação.....	70
Figura 35	Exemplo de edição do cadastro de Protocolo de Comunicação.....	71
Figura 36	Consulta SPARQL para Dbpedia.....	71
Figura 37	Grafo RDF subjacente ao cadastro de Protocolo de Comunicação.....	72
Figura 38	Tela de criação do modelo de sensor DHT22.....	73
Figura 39	Tela de criação de exemplar de sensor DHT22.....	73
Figura 40	Tela de edição Parte I do exemplar DHT22_Sensor.....	74
Figura 41	Grafo RDF subjacente a tela de edição ParteI do exemplar DHT22_Sensor.	75
Figura 42	Propriedades de localização de exemplar do modelo DHT22.....	75
Figura 43	Tela simplificada de edição Parte II do sensor.....	76
Figura 44	Descrição do Padrão de Interação Property.....	77
Figura 45	Parte do grafo RDF subjacente a tela de edição Parte II do exemplar DHT22 _Sensor com ênfase no padrão de interação Property.....	78
Figura 46	Tela de criação do modelo de atuador LED_Rgb.....	79
Figura 47	Tela de criação do atuador LED_Rgb_Actuator.....	79
Figura 48	Tela de edição Parte I do exemplar LED_Rgb_Actuator.....	80
Figura 49	Grafo RDF subjacente a tela de edição Parte I do exemplar LED_Rgb_Actuator.....	81
Figura 50	Grafo RDF subjacente ao cadastro de atuador com foco nas propriedades de Localização.....	82

Figura 51	Tela simplificada de edição parte II do atuador.....	82
Figura 52	Descrição do Padrão de Interação Property do Atuador.....	83
Figura 53	Parte do grafo RDF subjacente a tela de edição Parte II do exemplar LED Rgb Actuator com ênfase no padrão de interação Property.....	84
Figura 54	Descrição do Padrão de Interação Action do Atuador.....	85
Figura 55	Parte do grafo RDF subjacente a tela de edição Parte II do exemplar LED Rgb Actuator com ênfase no padrão de interação Action.....	86
Figura 56	Grafo RDF subjacente ao cadastro de Segurança na Aplicação semântica em Turtle.....	100
Figura 57	Grafo RDF subjacente ao cadastro de Protocolo de Comunicação na Aplicação semântica em Turtle.....	101
Figura 58	Grafo RDF subjacente ao cadastro de um exemplar de Sensor na Aplicação semântica em Turtle.....	102
Figura 59	Grafo RDF subjacente ao cadastro de um exemplar de Atuador na Aplicação semântica em Turtle.....	104

## LISTA DE QUADROS

Quadro 1	Áreas de Conhecimento e Critérios de Busca.....	38
Quadro 2	Área de Conhecimento e a quantidade de artigos encontrados.....	39
Quadro 3	Número de publicações anual envolvendo as 3 áreas de conhecimento....	40



## LISTA DE TABELAS

Tabela 1	Classes e propriedades das ontologias RDF e RDFS.....	26
Tabela 2	Classes e propriedades da ontologia OWL.....	28
Tabela 3	Descrição atual dos recursos que podem ser cadastrados na Aplicação....	42
Tabela 4	Ontologias adotadas na aplicação.....	44
Tabela 5	Tipos de recursos.....	45
Tabela 6	Classes e Propriedades da ontologia WoT.....	50
Tabela 7	Dispositivos que compõem o cenário IoT.....	67
Tabela 8	Definição dos Prefixos.....	68

## LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface  
CRUD - Create Read Update Delete  
DB - DBpedia Ontology  
DCTERMS - Dublin Core  
FOAF - Friend of a friend  
GR - Good Relations  
HTML - HyperText Markup Language  
HTTP - HyperText Transfer Protocol  
IoT - Internet of Things  
JSON-LD - JavaScript Object Notation for *Linked Data*  
LOD - Linked Open Data  
LOV - Linked Open Vocabularies  
ORG - The Organization Ontology  
OWL - Ontology Web Language  
PROV - Provenance Ontology  
RDF - Resource Description Framework  
RDFa - Resource Description Framework in Attributes  
RDFS - Resource Description Framework Schema  
RFID - Radio-Frequency Identification  
SCHEMA - Schema.Org  
SOSA - Sensor, Observation, Sample and Actuator  
SPARQL - SPARQL Protocol and RDF Query Language  
SSN - Semantic Sensor Network Ontology  
SSN-SYSTEM - Semantic Sensor Network Ontology and System Capability module  
SWoT – Semantic Web of Things  
UML - Unified Modelling Language  
URI - Uniform Resource Identifier  
VCARD - VCard Ontology  
W3C - World Wide Web Consortium  
WoT - Web of Things  
WoT IG - Web of Things Interest Group  
WoT WG - Web of Things Working Group  
XML - Extensible Markup Language

## SUMÁRIO

1. INTRODUÇÃO .....	15
1.1. Contextualização.....	15
1.2. Objetivo Geral .....	17
1.3. Objetivos Específicos .....	18
1.4. Justificativa .....	18
1.5. Contribuições.....	19
1.6. Tipo de Pesquisa .....	19
1.7. Estrutura do documento .....	20
2. FUNDAMENTAÇÃO TEÓRICA.....	21
2.1. Web Semântica .....	21
2.2. Resource Description Framework - RDF .....	21
2.2.1. Sintaxe RDF.....	24
2.3. Ontologias e Inferências .....	24
2.3.1. Metaontologia RDF-Schema – RDFS .....	25
2.3.2. MetaOntologia Web Ontology Language – OWL.....	26
2.4. <i>Linked Data</i> .....	29
2.5. Linguagem SPARQL.....	31
2.6. Internet of Things (IoT).....	33
2.7. Web of Things (WoT) .....	34
2.8. Semantic Web of Things (SWoT) .....	35
3. TRAJETÓRIA METODOLÓGICA.....	36
3.1. Revisão Bibliográfica .....	36
3.2. Elicitação de requisitos .....	40
3.3. Modelo ontológico.....	42

3.3.1. Seleção de ontologias .....	43
3.4. Desenvolvimento da Aplicação .....	45
3.4.1. Seleção do framework .....	45
4. VICINITY ONTOLOGY MODEL FOR WEB OF THINGS (WOT ONTOLOGY).....	47
5. MODELO ONTOLOGICO DA APLICAÇÃO .....	53
5.1. Módulo Security (Segurança) .....	53
5.2. Módulo Communication Protocol (Protocolo de Comunicação) .....	54
5.3. Módulo Sensor (Sensor) .....	55
5.4. Módulo Actuator (Atuador) .....	64
6. APLICAÇÃO SEMÂNTICA <i>LINKED DATA</i> .....	66
6.1. Alguns dispositivos comumente usados em configurações IoT .....	67
6.1.1. Cadastro de Segurança (Security).....	68
6.1.2. Cadastro de Protocolo de Comunicação (Communication Protocol) .....	70
6.1.3. Cadastro de Sensor (Sensor) .....	72
6.1.4. Cadastro de Atuador (Actuator).....	78
7. TRABALHOS RELACIONADOS .....	87
8. CONSIDERAÇÕES FINAIS .....	89
9. RESULTADOS.....	90
10. TRABALHOS FUTUROS .....	92
REFERÊNCIAS BIBLIOGRÁFICAS .....	93
APÊNDICE.....	100

## 1. INTRODUÇÃO

### 1.1. Contextualização

A possibilidade de objetos físicos do nosso cotidiano observarem o mundo por meio de sensores e atuadores e, posteriormente, analisar, armazenar ou até trocar informações, existia, há décadas, apenas no campo da ficção científica. Nos dias atuais, tais objetos são nomeados como “coisas inteligentes” e podem ser encontrados em diferentes campos de atuação, como: saúde, monitoramentos espaciais, ambientais, em automação e produção industrial, transporte inteligente de pessoas e bens etc. A viabilidade de comunicação entre o mundo real e o virtual dessas coisas inteligentes deu origem ao conceito de IoT (*Internet of Things* /Trad. Internet das Coisas). A IoT tornou-se um dos desenvolvimentos mais promissores e emocionantes em tecnologia e negócios (GUINARD & TRIFA, 2016).

Para Li *et al.* (2015), todas as áreas de aplicação, como também, nossas atitudes serão influenciadas diretamente devido a este novo universo de pequenos, senão, minúsculos computadores com sensores e interfaces de comunicação integrados em objetos cotidianos. A IoT abrange comunicação dos tipos Pessoa-Pessoa (P2P), Pessoa-Objeto (P2O) e Objeto-Objeto (O2O) (WANG *et al.*, 2017).

Para Serena *et al.* (2017), a particular heterogeneidade da IoT (tipo de coisa inteligente, protocolos de comunicação, formatos de dados e até mesmo os padrões da IoT (DATTA, 2015) e a quantidade de coisas inteligentes conectadas são questões evolutivas. Conforme informações da consultoria *Gartner*<sup>1</sup>, no ano de 2017, funcionavam cerca de 8,4 bilhões de coisas conectadas mundo afora. Para o ano de 2020, o crescimento deverá ser para 20 bilhões (MARQUES, 2017). Mediante este cenário, os usuários da IoT não estão cientes de todos os aspectos possíveis, de todas as plataformas e de todas as coisas inteligentes existentes e espalhadas geograficamente.

O conceito de WoT (*Web of Things* /Trad. Web das Coisas) vem como próximo passo evolutivo da IoT, onde ocorre a integração das coisas inteligentes na Web, por meio de reutilização e ambientação de tecnologias e padrões normalmente utilizados para o conteúdo da Web. Neste contexto, as coisas inteligentes passam a ser acessadas da mesma forma que qualquer página Web, ou seja, por meio de um endereço Web (URL) sobre o protocolo HTTP. Acrescentando, existe ainda o problema da Web convencional que é voltada para consumo humano, disponibilizando conteúdo de forma não estruturada, textual, fazendo com que as

---

<sup>1</sup> <https://www.gartner.com>

máquinas apenas exibam os documentos Web, sem, no entanto, compreender a semântica do conteúdo exibido (AZEVEDO & JACYNTHO, 2014). A Web hoje possui em média vinte bilhões de páginas, um número relativamente pequeno em relação a suposição feita acerca das coisas inteligentes que estarão disponíveis como parte da IoT, em futuro próximo. Se levarmos em consideração o problema de ausência de semântica da Web convencional no contexto da Internet das Coisas, onde é previsto um tamanho de 100 até 6000 vezes maior que a Web atual, é notório que os quesitos busca e indexação se tornam complexos. Deste modo, a semântica das coisas processável por máquina é um fator preponderante para a viabilidade e crescimento IoT (CUNHA, 2010). Assim, se faz necessário um modelo formal de representação de conhecimento (ontologia) comum para descrever as coisas, as suas características e capacidades, complementa Serena *et al.* (2017).

De acordo com Santarem Segundo e Coneglian (2016), ontologia define uma área de conhecimento de forma estruturada e formal, por meio de classes (conceitos), propriedades, relações, restrições, axiomas e instâncias. Para a extensão da Web original, conhecida como Web Semântica (ou Web do Conhecimento, ou ainda Web de Dados Ligados), ontologias têm papel fundamental, tornando a Web Semântica um veículo muito poderoso para descrição, processamento, reúso e compartilhamento de informações (BATISTA & JACYNTHO, 2017). Para o domínio de IoT, o uso de ontologias vem se destacando devido a benefícios, como: a formalização deste conhecimento; a representação de relacionamentos complexos entre os padrões IoT; a possibilidade de integração da ontologia de padrões de IoT com outras ontologias; o conhecimento inteligível por máquinas, que podem efetuar consultas precisas sobre o conhecimento explicitado pela ontologia; o compartilhamento e extensão dos conhecimentos sobre padrões de IoT de forma mais estruturada; o aprofundamento nos conhecimentos representados, permitindo aos desenvolvedores obter o conhecimento no nível de profundidade desejado (desde apenas uma lista de padrões relacionados até detalhes técnicos mais específicos) (COSSINI & RESENDE, 2016).

O consórcio W3C<sup>2</sup>, uma comunidade internacional que define os padrões da Web, dirigida pelo inventor da Web e Diretor *Sir. Tim Berners-Lee*, se refere a IoT como a representação virtual de um grupo de objetos ligados à Internet, interagindo com demais sistemas e serviços ainda na Internet, e com as pessoas. Essa comunicação possibilita que qualquer coisa esteja conectada a qualquer momento e em qualquer lugar (COSSINI &

---

<sup>2</sup> <https://www.w3.org/>

RESENDE, 2016). O WoT WG<sup>3</sup> (*Web of Things Working Group* / Trad. Grupo de Trabalho Web das Coisas) do W3C, possui seu processo de padronização e nasceu dos estudos e pesquisas do WoT IG<sup>4</sup> (*Web of Things Interest Group* / Trad. Grupo de Interesses Web das Coisas). Este grupo de trabalho tem como objetivo estudar e desenvolver serviços e aplicações que utilizem tecnologias IoT e WoT, e assim também definir novos padrões de interoperabilidade entre os dispositivos IoT e os serviços que os conectam. Seguindo as especificações do WG, tem-se o Projeto VICINITY<sup>5</sup>, onde pode-se contar com ontologias ligadas ao domínio IoT. O principal objetivo do projeto é fazer com que os usuários compartilhem informações de suas coisas inteligentes, sem perder o controle sobre elas. Em contribuição, aplicações *Linked Data* - criadas conforme o conceito *Linked Data* proposto por Tim Bernes-Lee (2006) para publicação de dados e metadados estruturados na Web - são desenvolvidas, tornando possíveis a publicação, reutilização e o compartilhamento dos dados, combinando as ontologias já consagradas em seu desenvolvimento, proporcionando avanço tanto no estudo da Web Semântica quanto da IoT, culminando na chamada SWoT (*Semantic Web of Things* / Trad. Web Semântica das Coisas).

## 1.2. Objetivo Geral

O objetivo geral deste trabalho se concentra na proposta de uma abordagem para descrever formalmente a interface de acesso de coisas inteligentes (sensores e atuadores), bem como a extensão da aplicação semântica *Linked Data* proposta por Rangel (2018), seguindo as especificações da W3C WG e os princípios *Linked Data*.

Na aplicação semântica *Linked Data* proposta por Rangel (2018), é possível catalogar e publicar na Web, coisas inteligentes descrevendo-as formalmente (tipo, serviço oferecido, fabricante, localização física, etc.), permitindo que sensores e atuadores sejam pesquisados e encontrados, de forma precisa, na Web.

Uma vez pesquisado e encontrado o sensor (ou atuador) do qual se queira fazer uso, é preciso saber, detalhadamente, "como" usá-lo. O objetivo deste trabalho é justamente descrever formalmente, por meio de ontologias de referência, "como" fazer uso da coisa inteligente. Em outras palavras, descrever a interface de acesso da coisa inteligente, identificando seus parâmetros, protocolos de acesso e formatos de dados. Desta forma, tantos agentes de software

---

<sup>3</sup> <https://www.w3.org/WoT/WG/>

<sup>4</sup> <https://www.w3.org/WoT/IG/>

<sup>5</sup> <http://vicinity.iot.linkeddata.es>

quanto humanos compreendem os dados contidos na aplicação, de forma a possibilitar o reuso e compartilhamento dos mesmos por toda Web de Dados Ligados.

**Delimitação da pesquisa:** A proposta do trabalho/aplicação desenvolvida consiste em descrever, utilizando as tecnologias e padrões da Web Semântica, a interface de acesso da coisa inteligente, identificando seus parâmetros, protocolos de acesso e formatos de dados. O armazenamento semântico dos dados de observação dos sensores (ou atuadores) não faz parte do escopo deste trabalho, sendo considerado o próximo passo a ser contemplado na continuação prevista para este trabalho.

### 1.3. Objetivos Específicos

Para concretização do objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Utilizar as ontologias consagradas como padrão de facto para uma melhor representação do domínio IoT;
- Garantir uma aplicação com uma interface simples de uso pelos usuários finais;
- Atestar a validade da aplicação com exemplos de dispositivos usados em cenários IoT;
- Contribuir tanto para expansão da Web de Dados Ligados quanto a Internet das Coisas e, por conseguinte, para a nova Web Semântica das Coisas;

### 1.4. Justificativa

A medida que novas pessoas acessam a Web, novas coisas inteligentes são conectadas e disponibilizadas nela; do mesmo modo que cada pessoa possui sua particularidade (nome, endereço físico, profissão, etc.) as coisas inteligentes também possuem (tipo de coisa inteligente, protocolos de comunicação, formato de dados, etc.). Deste modo, torna-se um grande desafio descobrir tais coisas inteligentes a ponto de interagir com êxito com elas (SERENA *et al.*, 2017).

Outro ponto de atenção está no fato de que a maioria das buscas por essas coisas na Web, tem os humanos como atores responsáveis, onde os mesmos buscam as informações, fazem a consulta e integração delas, e as interpretam para tomada de decisão, de forma manual, via mecanismos de busca, como Google. Um processo lento, tedioso e suscetível a erros (AZEVEDO & JACYNTHO, 2014). Tim Bernes-Lee *et al.* (2001) propuseram uma extensão da Web atual, tornando o conhecimento semântico da informação publicada explícito para agentes de softwares, que desta forma, conseguem buscar tais informações, inferir novas, além



de ligá-las a outras que já se encontram na web. Deste modo, seria primordial que estas coisas inteligentes estivessem catalogadas nesta nova Web Semântica. Porém, tê-las semanticamente catalogadas na Web não é suficiente. Um segundo passo muito importante é descrever a forma de acesso a essas coisas, permitindo saber que tipo de coisa será acessado (sensor, atuador), qual protocolo de comunicação é utilizado (p.ex.: COAP, HTTP), que tipo de dados é retornado (p.ex.: JSON), enfim descrever semanticamente a interface de acesso da coisa inteligente.

Com o intuito de fazer parte deste novo paradigma da Web, atuando diretamente no domínio de IoT, que já faz parte de uma nova era de tecnologia de comunicação, que este trabalho apresenta uma proposta para descrição semântica *Linked Data* da interface de acesso de sensores e atuadores, seguindo as especificações da W3C WG. Descrição formalizada por meio de ontologias *Linked Data* consagradas, possibilitando melhor representação do domínio estudado por meio de um vocabulário comum, bem como permitindo que informações sejam interpretadas e inferidas por máquinas. A viabilidade e os benefícios da proposta são corroborados por meio de uma aplicação com exemplos de dispositivos usados em cenários IoT, onde coisas inteligentes têm suas interfaces de acesso formalmente descritas.

### 1.5. Contribuições

As principais contribuições deste trabalho incluem:

- Modelo ontológico para a descrição formal interface de acesso aos sensores e atuadores (seleção de ontologias *Linked Data*);
- Aplicação semântica *Linked Data* para cadastro de metadados de sensores e atuadores na Web por meio do modelo ontológico anterior;
- Legado de como desenvolver uma aplicação semântica *Linked Data*.

### 1.6. Tipo de Pesquisa

Pela perspectiva dos procedimentos técnicos realizados, este trabalho encontra-se classificado como pesquisa bibliográfica, onde é feita por meio de um levantamento de referências teóricas já conferidas, e publicadas por meios escritos e eletrônicos, tais como artigos científicos, livros, páginas de *web sites*. Quanto a natureza da pesquisa, este trabalho é classificado como pesquisa aplicada, pois é voltado para aquisição de conhecimentos decorrentes de uma aplicação numa situação específica conforme destaca Appolinário (2009).

Quanto à abordagem do problema, a pesquisa é qualitativa através dos dados levantados. Com relação aos objetivos, a pesquisa desenvolvida, classifica-se como exploratória. Uma vez

que seu objetivo é proporcionar maior familiaridade com o problema, tornando-o mais explícito ou a construir hipóteses (GIL, 2008).

### **1.7. Estrutura do documento**

Neste capítulo inicial foram apresentadas as principais ideias desta dissertação, descrevendo o contexto, os objetivos, a justificativa do trabalho e metodologia de pesquisa. Além deste capítulo introdutório, o trabalho é composto pelos seguintes capítulos: o Capítulo 2 (Fundamentação Teórica) apresenta aspectos teóricos relacionados a Web Semântica, e seus componentes, além de outros conceitos também relevantes ao trabalho como IOT e suas variantes; no Capítulo 3 (Trajetória Metodológica) são apresentadas as etapas definidas na metodologia do trabalho, bem como os procedimentos realizados em cada uma destas etapas; o Capítulo 4 (Vicinity Ontology Model for Web of Things (Wot Ontology)) descreve a ontologia parte do Projeto VICINITY, que formaliza a base para descrição da interface de acesso a uma coisa inteligente; no Capítulo 5 (Modelo Ontológico da Aplicação) é abordada a utilização integrada das ontologias selecionadas na descrição de cada requisito funcional da aplicação; o Capítulo 6 (Desenvolvimento do Aplicação) descreve as telas da aplicação semântica *Linked Data* para descrição da interface e uso de sensores e atuadores, por meio do cadastro de uma “coisa inteligente”, sendo esta, participante de um cenário realista, atestando a aplicação; no Capítulo 7 (Trabalhos Relacionados) são elencados alguns trabalhos relacionados ao domínio explorado neste trabalho; no Capítulo 8 (Considerações Finais) são reiteradas as contribuições do trabalho, bem como as conclusões; no Capítulo 9 (Resultados) são apresentados os resultados obtidos no trabalho; por fim, no Capítulo 10 (Trabalhos Futuros) apresenta as propostas de trabalhos futuros para continuidade e aprimoramento do trabalho.

## 2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos relacionados à fundamentação teórica do trabalho. Inicialmente, das seções 2.1 a 2.5, conceitos centrais e tecnologias associadas à Web Semântica e, por fim, nas seções 2.6, 2.7 e 2.8 têm-se uma elucidação acerca dos domínios de Internet das Coisas, Web das Coisas e Web Semântica das Coisas.

### 2.1. Web Semântica

Uma das finalidades iniciais da Web era a troca de informação entre pessoas, com os computadores sendo participantes na comunicação, auxiliando os usuários. Atualmente, os computadores na Web têm o papel somente no direcionamento e entrega de informações, sem compreensão do conteúdo das páginas, uma vez que a informação não é estruturada, voltada para consumo por pessoas e não máquinas. Por esta razão, a Web original é nomeada como Web de documentos e não de dados. Assim, os computadores ajudam limitadamente no acesso e processamento da informação, necessitando que as funções de extração e interpretação dessa informação sejam realizadas a cargo dos usuários (DIAS & SANTOS, 2003).

A Web Semântica, também conhecida como Web de Dados ou ainda Web do Conhecimento, consiste em uma extensão evolutiva da Web, permitindo que a informação seja alcançada em níveis muito mais profundos, atribuindo significância formal explícita aos conteúdos pesquisados e relacionando-os ao contexto em que são usados. As informações são estruturadas de tal forma que as máquinas podem lê-las e compreendê-las tanto quanto as pessoas, sem ambiguidade (NATH & ISWARY, 2015).

Para o correto intercâmbio de dados estruturados, a Web Semântica é definida, basicamente, por três componentes, a saber: *Resource Description Framework* (RDF) - modelo de dados; sintaxes; ontologias - semântica explícita (ANTONIOU *et al.*, 2012).

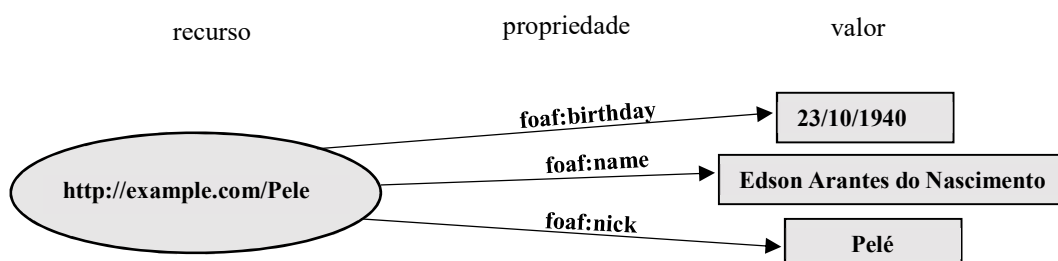
### 2.2. Resource Description Framework - RDF

Segundo Jacyntho (2012), RDF é um modelo de dados em grafo, adequado para descrever recursos Web por meio de triplas (recurso-propriedade-valor). Cada tripla é formada por:

- **Nó de origem (Recurso ou Sujeito):** qualquer entidade do mundo real (eventos, cidades, leis, escolas, etc.) que se almeje descrever, identificada por um endereço Web, uma espécie de identificador global, conhecido como URI (*Uniform Resource Locator*);



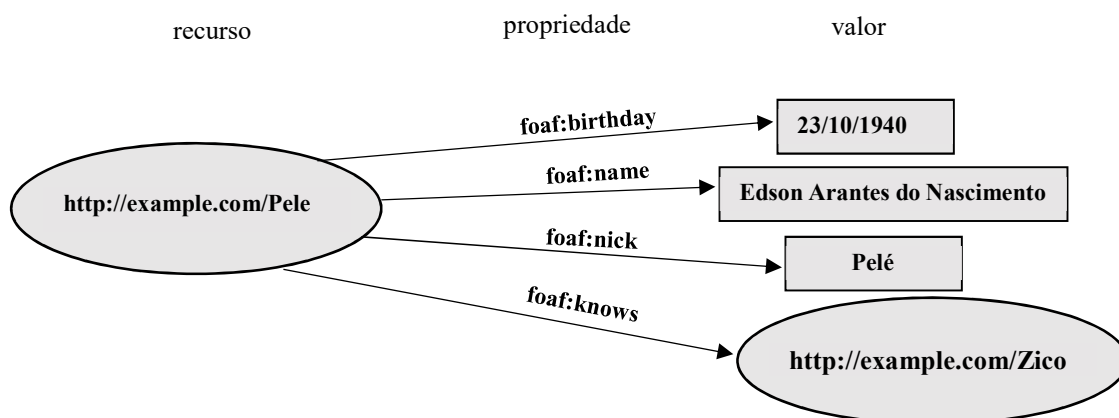
Um recurso pode ter mais de uma propriedade atrelada a ele, ou seja, afirmações compostas sobre um mesmo recurso. A seguir, na Figura 2, é exemplificado o grafo referente a frase “Pelé nasceu em 1940, seu nome completo é Edson Arantes do Nascimento e seu apelido é Pelé.”



**Figura 2** - Grafo RDF com três triplas de um mesmo recurso.

Fonte: Elaboração Própria

Por definição, objetos podem conter recurso ou valor literal (texto). Na Figura 3, temos um recurso cujo URI é “http://example.com/Pele” relacionado a um outro recurso com o URI “http://example.com/Zico”. A descrição da relação é feita pela propriedade “foaf:knows”. Note que o valor para a propriedade “foaf:knows” no grafo é representado pelo formato oval, significando um recurso. Para as demais propriedades, os valores são representados pelo formato retangular, significando um valor literal.

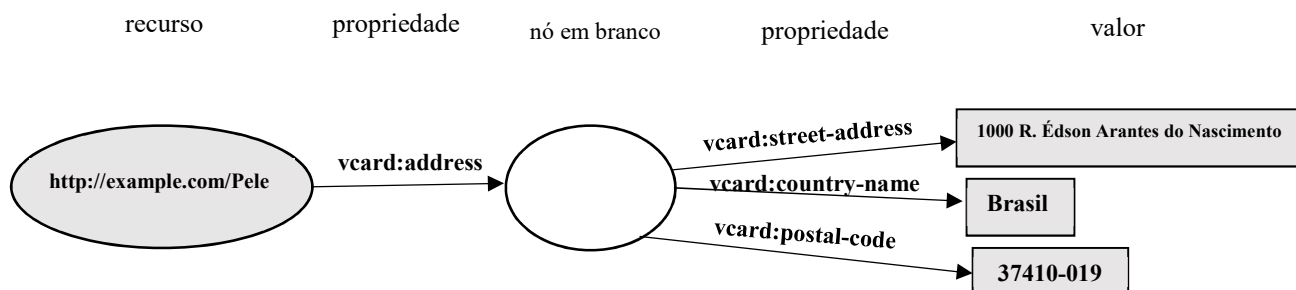


**Figura 3** - Grafo RDF relacionando dois recursos.

Fonte: Elaboração Própria

Contudo, em algumas situações, um recurso pode ser descrito sem um identificador. Para tais recursos (sem um URI que os especifica) são representados em RDF como recursos anônimos ou nós em branco (*blank nodes*). Estes podem surgir como sujeitos ou objetos em uma tripla fazendo referência a recurso sem “nome” explícito por um URI. A Figura 4 descreve o endereço de um recurso cujo URI é “http://example.com/Pele”. Para efetuar a descrição, o recurso foi relacionado ao nó em branco por meio da propriedade “vcard:address”, da ontologia

vCard (IANNELLA & MCKINNEY, 2014). O referido nó em branco possui várias propriedades (logradouro, país, Cep) que em conjunto descrevem o endereço do recurso “http://example.com/Pele”.



**Figura 4** - Grafo RDF relacionando um recurso a um *blank node*.

Fonte: Elaboração Própria

Dentre os benefícios de utilização do padrão RDF, destaca-se: ser um modelo semiestruturado flexível para criação de metadados; conceder sintaxes específicas para a descrição dos recursos e propriedades na Web; consentir que aplicações se comportem de forma inteligente e automatizada sobre as informações publicadas na Web, dado que, sua significância é explícita e, portanto, mais facilmente inteligível por máquina (SOUZA & ALVARENGA, 2004).

### 2.2.1. Sintaxe RDF

Vale ressaltar que o RDF não é um formato de dados, mas sim um modelo de dados para descrever recursos sob a forma de sujeito, predicado, objeto. Para efetivamente publicar um grafo RDF na Web, deve-se usar uma sintaxe RDF. Isso significa tomar as triplas que compõem um grafo RDF e usar uma sintaxe específica para publicá-las como um arquivo. Dentre as sintaxes utilizadas e padronizadas temos: RDF/XML<sup>6</sup>; RDFa<sup>7</sup>, Turtle<sup>8</sup>, N-Triples<sup>9</sup>, RDF/JSON<sup>10</sup> (HEATH & BIZER, 2011).

### 2.3. Ontologias e Inferências

Na Web de Dados, ou Web semântica, é proposto uma implantação explícita da semântica das informações, isto é, dotado de contexto e vocabulário comuns. Para tal, têm-se usado ontologias, aptas a oferecer essa semântica. Para Santarem Segundo e Coneglian (2016),

<sup>6</sup> <http://www.w3.org/TR/rdf-syntax-grammar/>

<sup>7</sup> <http://www.w3.org/TR/xhtml-rdfa-primer/>

<sup>8</sup> <http://www.w3.org/TeamSubmission/turtle/>

<sup>9</sup> [http://www.mkbergman.com/wp-content/themes/ai3/files/2009Posts/Advantages\\_Myths\\_RDF\\_090422.pdf/](http://www.mkbergman.com/wp-content/themes/ai3/files/2009Posts/Advantages_Myths_RDF_090422.pdf/)

<sup>10</sup> <http://www.mkbergman.com/938/what-is-a-reference-concept/>

ontologias representam uma área de conhecimento de forma estruturada e formal, por meio de classes (conceitos), propriedades, relações, restrições, axiomas e instâncias. Para Guarino *et al* (2009), as ontologias são como um tipo especial de objeto de informação ou artefato computacional. Tais ontologias computacionais são um meio de modelar formalmente a estrutura de um sistema, ou seja, as entidades e relações relevantes que emergem da sua observação e que são úteis para nossos propósitos.

Para construir a semântica dos domínios, faz-se necessário referir suas interações lógicas. As expressões que auxiliam essa operação são os axiomas (regras lógicas), definidos nas ontologias, capazes de permitir inferências, fator relevante na semântica. Assim, por meio dos axiomas, um agente de software é capaz de inferir dados implícitos, a partir de dados existentes formalizados com a ontologia.

As ontologias precisam de uma linguagem para transformar sua estrutura conceitual em implementação (SANTAREM SEGUNDO & CONEGLIAN, 2016). Consideradas como linguagens padrão, RDF-Schema (RDFS) (D. BRICKLEY & GUHA, 2004) e *Web Ontology Language* (OWL) (DEAN & SCHREIBER, 2004) são linguagens reconhecidas como meta-ontologias, pois são ontologias RDF utilizadas para geração de novas ontologias RDF para uma área de conhecimento específico (AZEVEDO & JACYNTHO, 2014).

### **2.3.1. Metaontologia RDF-Schema – RDFS**

É possível revalidar, conforme os exemplos anteriores, que o RDF possibilita caminhos para descrição de recursos por meio de propriedades com seus devidos valores, porém, nenhuma propriedade foi criada por meio do RDF e sim, foram utilizados vocabulários já criados como o “FOAF”, “vCard”.

A metaontologia RDF-Schema (RDFS) é um vocabulário que complementa o RDF e insere uma camada que caracteriza algumas particularidades que adicionam semântica a dados definidos em RDF. Em outras palavras, Santarem Segundo (2010) reporta que RDFS oferece ao RDF capacidade de criação de estruturas como hierarquias, propriedades e subpropriedades.

RDFS tem construtores que possibilitam, por exemplo, descrever que determinados URIs indicam propriedades de recursos, ou que determinados recursos identificados por URIs pertencem a uma dada classe. A relação entre uma instância e sua classe é indicada através da propriedade “type” de RDF. Restrições de tipos podem ser definidas sobre os sujeitos e os

objetos das triplas, por meio da especificação de domínios (*rdfs:domain*<sup>11</sup>) e contradomínios (*rdfs:range*<sup>12</sup>) para cada um dos tipos. Na Tabela 1 a seguir é possível visualizar as classes e propriedades da metaontologia RDFS:

**Tabela 1** - Classes e propriedades das ontologias RDF e RDFS.

Nome da classe	Descrição
rdfs: Resource	O recurso de classe, tudo.
rdfs: Literal	A classe de valores literais.
rdf: XMLLiteral	A classe de valores literais XML.
rdfs: Class	A classe de classes.
rdf: Property	A classe de propriedades RDF.
rdfs: Datatype	A classe de tipos de dados RDF.
Nome da propriedade	Descrição
rdf: type	Indica a classe de uma instância.
rdfs: subclassOf	O indica a super classe de uma subclasse.
rdfs: subPropertyOf	Indica a super classe de uma propriedade.
rdfs: domain	O domínio de uma propriedade (classes).
rdfs: range	O alcance (possíveis valores) de uma propriedade.
rdfs: label	Etiqueta em linguagem natural.
rdfs: comment	Comentário.
rdfs: seeAlso	Mais informações sobre o recurso.
rdfs: isDefinedBy	A definição do recurso assunto.
rdf: value	Instância rdf:Property usada na descrição de valores estruturados.

Fonte: Elaboração Própria

### 2.3.2. MetaOntologia Web Ontology Language – OWL

RDF especifica um modelo de dados, RDFS estende RDF permitindo assim um primeiro conjunto de restrições (restringir o número de diferentes interpretações possíveis), sobre as triplas definidas, além de inferências que deduzem triplas não declaradas de forma explícita (LAUFER, 2015).

Web Ontology Language (OWL) é uma linguagem que estende RDF e RDFS e oferece um conjunto muito mais amplo de tipos de restrições ao conjunto de triplas definidas. Além disso, são oferecidos diversos construtores que permitem, entre outros, a construção de classes complexas a partir de outras definições de classes, e encadeamento de propriedades. A

<sup>11</sup> P rdfs:domain {C1, C2, ... Cn} indica que qualquer recurso que tenha a propriedade P é instância de {C1 Ç C2 Ç ... Ç Cn}.

<sup>12</sup> P rdfs:range {C1, C2, ... Cn} indica que os valores da propriedade P são instâncias de {C1 Ç C2 Ç ... Ç Cn}.

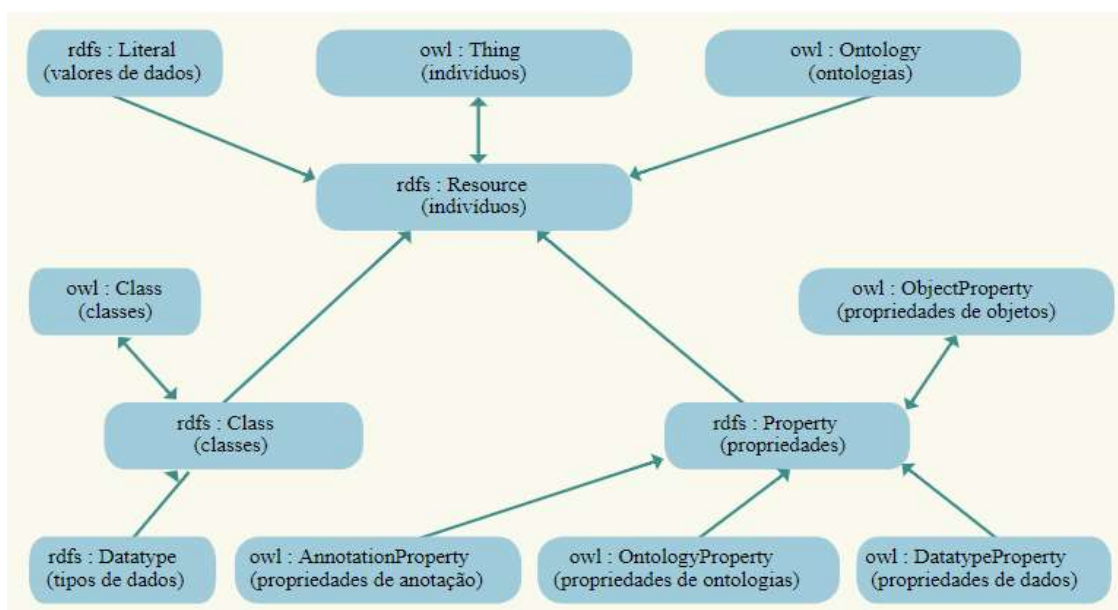


linguagem possui três sub-linguagens (OWL-Lite, OWL-DL, OWL-Full), cada uma mais expressiva que a anterior. Ontologias fornecem meios para modelar as relações entre as entidades em um domínio de interesse. Em OWL existem três tipos de entidades:

- **Instâncias** – representam os recursos (são também chamados de indivíduos).
- **Classes** – definem conjuntos de instâncias, de indivíduos.
- **Propriedades** – representam relações binárias entre duas instâncias ou entre uma instância e um literal.

Em OWL, uma classe é qualquer recurso que possua o valor *owl:Class* na propriedade *rdf:type*, sendo que *owl:Class* é uma subclasse de *rdfs:Class*. Classe são definidas por um conjunto de axiomas, ou seja, conjunto de instâncias com características comuns.

Por basear-se em vocabulário RDF, pode ser usado em combinação com RDF-Schema. Assim, os termos criados em RDFS continuam válidos em OWL (DEAN & SCHREIBER, 2004). Na Figura 5, é possível visualizar as relações entre os vocabulários RDFS e OWL.



**Figura 5-** Semântica baseada em RDF usada em OWL.

Fonte: SCHNEIDER (2012)

Em OWL, as propriedades são separadas em duas categorias distintas: *object properties* (propriedades de objeto), que relacionam indivíduos a indivíduos, ou seja, definem relacionamento entre classes e *datatype properties* (propriedades de tipos de dados), que relacionam indivíduos a dados literais. Essas duas categorias são subpropriedades de *rdfs:Property*.

Uma especial propriedade de OWL é a *owl:sameAs*, ela define que dois URIS distintos identificam o mesmo indivíduo. A seguir, tem-se na Figura 6, um exemplo em que dois URI's se referem à mesma pessoa:

```
<rdf:Description rdf:about="#William_Jefferson_Clinton">
  <owl:sameAs rdf:resource="#BillClinton"/>
</rdf:Description>
```

**Figura 6** – Exemplo em sintaxe RDF/XML do uso da propriedade *owl:sameAs*.

Fonte: w3.org<sup>13</sup>

Atualmente existe uma versão estendida da OWL (primeira versão publicada em 2004), nomeada como OWL2<sup>14</sup>. Esta inclui maior expressividade, um modelo de dados mais simples e serialização, assim como uma coleção de sub-linguagens, além de, aproveitar todos os elementos da OWL. Na Tabela 2 abaixo, é possível visualizar algumas das diversas classes e propriedades presentes no vocabulário OWL.

**Tabela 2** - Classes e propriedades da ontologia OWL.

<b>Classes</b>	owl:Class
	owl:DataRange
	owl:DatatypeProperty
	owl:DeprecatedClass
	owl:DeprecatedProperty
	owl:Nothing
	owl:ObjectProperty
	owl:Restriction
	owl:SymmetricProperty
	owl:TransitiveProperty
owl:Thing	
<b>Propriedades</b>	owl:differentFrom
	owl:sameAs
	owl:equivalentClass
	owl:equivalentProperty
	owl:someValuesFrom
	owl:allValuesFrom
	owl:InverseFunctionalProperty
owl:FunctionalProperty	

Fonte: Elaboração Própria

<sup>13</sup> <http://www.w3.org/TR/owl-ref/>

<sup>14</sup> <https://www.w3.org/TR/owl2-overview/>

## 2.4. *Linked Data*

A expressão *Linked Data* corresponde ao conjunto de melhores práticas ou princípios para publicação e ligação de dados estruturados em RDF na Web, conforme destaca Tim Bernes-Lee *et al* (2006). São quatro princípios, a saber:

1. Nomear recursos disponíveis na Web por meio de URIs;
2. Utilizar HTTP URIs para que humanos ou máquinas procurem e acessem esses endereços, usando o protocolo HTTP;
3. Quando um URI de um recurso for dereferenciado (acessado), retornar informações úteis em RDF e seus padrões, detalhando explicitamente o recurso;
4. Incluir links RDF para outros URIs, de maneira que tanto humanos ou máquinas possam navegar por estes links na Web encontrando mais recursos (*Linked Data Mashup*).

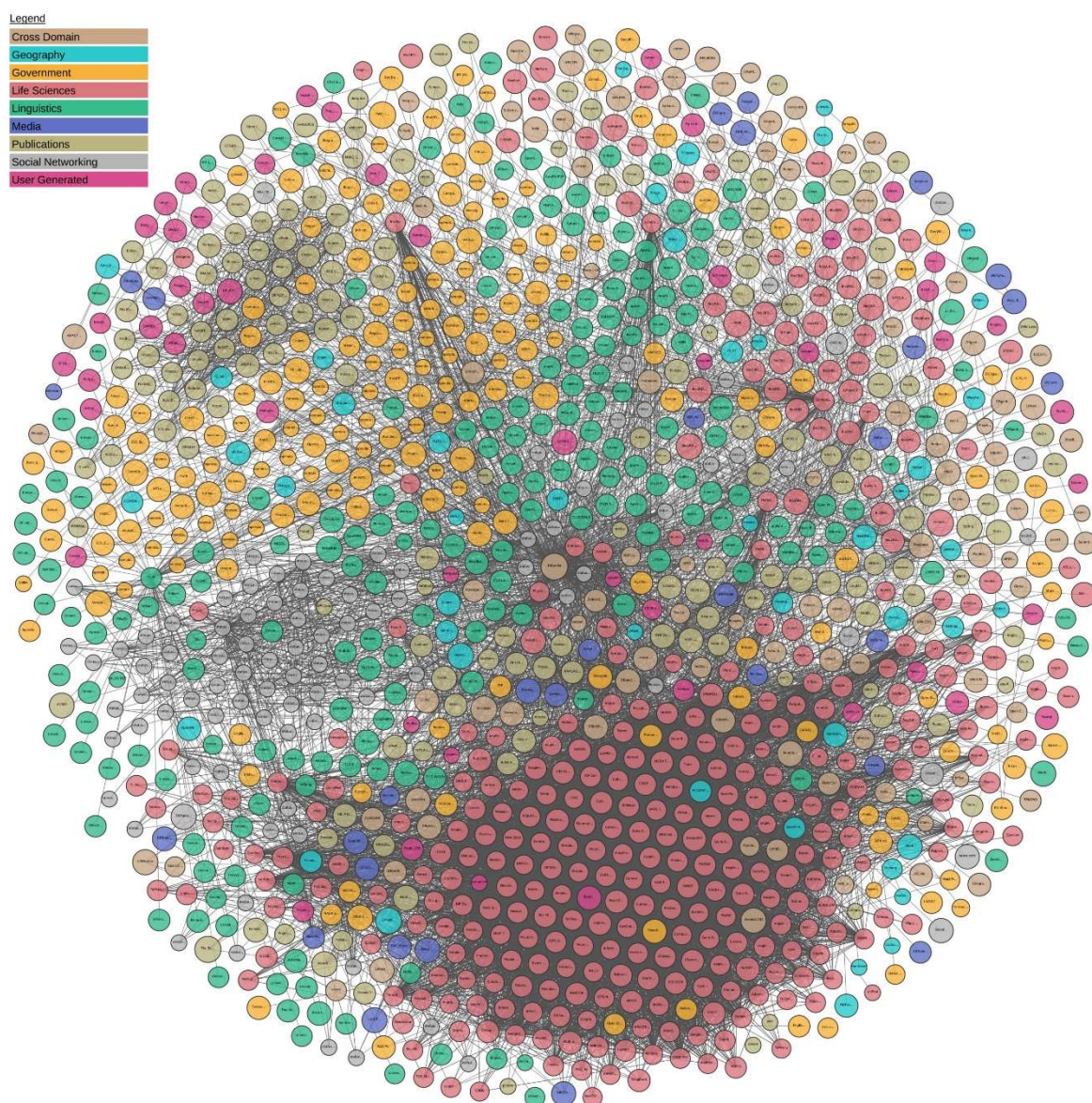
O W3C, desde 2007, apoia o projeto *Linked Open Data* (LOD) (BERNES-LEE, 2009), que vem transformando dados na Web pertencentes a licenças públicas em dados RDF, de acordo com os princípios *Linked Data*. O projeto é aberto a qualquer pessoa que publica dados de acordo com os princípios *Linked Data*. Seu objetivo é maximizar a Web, expondo conjuntos de dados abertos em formato RDF na Web e determinando links RDF entre itens de fontes de dados distintas (HEATH & BIZER, 2011).

Por estarem dentro dos padrões, estas fontes de dados ganham espaço no *Linked Open Data Cloud Diagram* ou *Nuvem LOD* (ABELE & McCRAE, 2017), local onde as fontes de dados *Linked Data* são listadas em uma espécie de diagrama. A medida que cresce a utilização da fonte de dados por outras aplicações, maior é o destaque no diagrama. Como fonte de dados mais reusada na nuvem LOD tem-se a famosa DBpedia (BIZER *et al*, 2009), uma versão *Linked Data* do conteúdo da *Wikipedia*.

A medida que dados de áreas de conhecimento são publicados como *Linked Data*, é conferido a estes uma categoria específica, como: *Linked Government Data*, *Enterprise Linked Data*, *Statistical Linked Data*, *GeoLinked Data*, *Linked Sensor Data*, etc.

Atualmente, com o proveito do uso de *Linked Data*, foram criadas aplicações semânticas pela ótica de *Linked Data Mashups* (LDM), disponibilizando novas funções para estas aplicações, por meio de combinações dos dados em bases de dados distintas disponíveis na Web.

A Figura 7 mostra a topologia da Web de Dados Ligados, onde várias fontes de dados (círculos) são interligadas por triplas RDF. O conjunto de dados contém atualmente 1.234 conjuntos de dados com 16.136 *links* (em junho de 2018), deste modo, não é possível explorar de maneira satisfatória o conteúdo presente na imagem. Assim, para melhor visualização e exploração do conteúdo, segue o endereço para consulta < <https://lod-cloud.net/>>.



The Linked Open Data Cloud from lod-cloud.net



**Figura 7 - Linked Open Data Cloud Diagram.**  
Fonte: lod-cloud<sup>15</sup>

<sup>15</sup> <https://lod-cloud.net/>





```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
?person foaf:name ?name .}

```

**Figura 9** – Exemplo de consulta SPARQL.

Fonte: Elaboração Própria

Com base na consulta acima realizada, seriam retornados todos os nomes, o resultado é apenas o valor “Marta”, pois temos apenas uma tripla. Este valor é armazenado na variável *?name*. Na variável *?person*, é armazenado o valor “*http://www.example.com/person01*”. Abaixo, os quatro tipos de consulta do oferecidos pela linguagem de consulta SPARQL:

- **SELECT:** Possui como retorno uma lista tabular, conforme uma condição definida (cláusula *WHERE*), semelhante com o *SELECT* do SQL; sintaxe: *SELECT ?<variavel> WHERE { <condição> .}*
- **CONSTRUCT:** Possui como retorno um novo grafo, conforme uma condição definida (cláusula *WHERE*);
- **ASK:** Possui como retorno um valor booleano (*TRUE* ou *FALSE*), conforme a existência ou não do padrão gráfico definido (cláusula *WHERE*) na base consultada;
- **DESCRIBE:** Possui como retorno um grafo conforme suas restrições definidas na clausula *WHERE*. Excelente alternativa para recuperar todas as triplas onde um recurso é citado;

As variáveis SPARQL começam com um caractere “?” e podem ser atribuídas em qualquer uma das três posições de uma tripla (sujeito, predicado, objeto) no conjunto de dados RDF. SPARQL aceita consultas compostas por mais de um *triple pattern*, formando um *graph pattern* (padrão de grafo). Na Figura 10, é realizada uma nova consulta SPARQL usando *graph pattern* e retornando mais de um resultado.

The screenshot shows a SPARQL playground interface. At the top right, the endpoint is 'http://sparql-playground.sib.swiss/sparql'. Below this, there is a list of prefixes: 'PREFIX dbo:<http://dbpedia.org/ontology/>', 'PREFIX dbp:<http://dbpedia.org/property/>', 'PREFIX dbpedia:<http://dbpedia.org/resource/>', 'PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>', 'PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>', 'PREFIX tto:<http://example.org/tuto/ontology#>', 'PREFIX ttr:<http://example.org/tuto/resource#>', and 'PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>'. The query is: 'select ?person ?pet where { ?person rdf:type dbo:Person . ?person tto:pet ?pet . }'. Below the query, it says 'Query time is 0.602[s] for 3 rows'. The results are shown in a table with two columns: 'person' and 'pet'. The rows are: 'ttr:John' and 'ttr:LunaCat', 'ttr:John' and 'ttr:TomCat', and 'ttr:William' and 'ttr:RexDog'.

```

Hide prefixes
endpoint: http://sparql-playground.sib.swiss/sparql
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX dbp:<http://dbpedia.org/property/>
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>

select ?person ?pet where {
  ?person rdf:type dbo:Person .
  ?person tto:pet ?pet .
}

Query time is 0.602[s] for 3 rows

person      pet
ttr:John    ttr:LunaCat
ttr:John    ttr:TomCat
ttr:William ttr:RexDog

```

**Figura 10** – Consulta SPARQL com mais de um resultado.  
Fonte: sparql-playground.sib<sup>20</sup>

Na Figura 10, são selecionadas duas variáveis *?person* e *?pet*. A variável *?person* está associada por meio da propriedade *rdf:type* ao objeto *dbo:Person*. Neste caso, também é desejado que a *?person* seja conectada a um objeto *?pet* através do predicado *tto:pet*. É possível observar que somente pessoas que tenham pelos menos um pet (animal de estimação) fazem parte do conjunto de resultados.

SPARQL admite uma série de filtros e operadores que possibilita realizar consultas complexas ao conjunto de triplas armazenadas. Vários bancos de dados de triplas oferecem pontos de acesso via Web (URLs), que aceitam o protocolo SPARQL e sua linguagem de consulta. Esses pontos de acesso são nomeados como SPARQL *endpoints*. Um SPARQL endpoint aceita consultas e retorna os resultados via HTTP (SCHNEIDER, 2011).

Devido a documentação do SPARQL ser extensa, aqui foram exemplificadas apenas algumas funcionalidades da forma de consulta *SELECT*. A documentação completa pode ser encontrada em PRUD'HOMMEAUX E SEABORNE (2008).

## 2.6. Internet of Things (IoT)

Demonstrando uma nova geração de computação ubíqua, representando a onipresença da computação em produtos de consumo e no dia a dia das pessoas, o termo IoT (*Internet of Things* /Trad. Internet das Coisas) tem sido empregado como alusão. Objetos comuns como automóveis, telefones, geladeiras, televisores, relógios dispõem da capacidade de conexão,

<sup>20</sup> parql-playground.sib.swiss

comunicação, e acesso à Internet embutida, demonstrando variadas perspectivas de uso, como por exemplo, comando à distância, personalização, automação e análises de desempenho.

As operações se tornam viáveis por meio dos elementos envolvidos, nomeados como *Smart Things* (Trad. Coisas Inteligentes) como: sensores, atuadores, mini placas processadoras, interface de comunicação com ou sem fio. Deste modo, os objetos físicos passam a ter a habilidade de processamento e comunicação de dados e conseqüentemente, podem “sentir” o ambiente, “perceber” seu estado e o de outros, além de trocar, requisitar, fornecer, delegar, gerenciar informações (BARACHO *et al*, 2017).

A inserção de tais coisas inteligentes na Internet traz vastas possibilidades de novas aplicações, fazendo uso dos serviços e informações presentes. No entanto, a maior parte dessas coisas nos dias de hoje são conectadas à internet, sendo muito pouco na Web, fazendo uso de softwares e interfaces proprietárias, tornando oneroso o surgimento de aplicações que realizam a integração dos dados e serviços providos pelos distintos dispositivos (GUINARD, 2010).

## **2.7. *Web of Things* (WoT)**

O cenário IoT segue com uma crescente perspectiva. Nos próximos anos, bilhões de objetos estarão aptos para utilização. A próxima etapa será fazer uso de um ecossistema e de uma infraestrutura para construir aplicações para o IoT, desarmando a prática de “um dispositivo, um protocolo, um aplicativo”. Nesse contexto tem-se a Web, onde, tais coisas inteligentes participam do mesmo universo, possibilitados pelo reúso de protocolos e padrões, sem comprometer a segurança ou desempenho, tornando os dados e serviços oferecidos por tais coisas mais acessíveis a um maior número de usuários simultaneamente.

A essa aplicabilidade da Web para o domínio da Internet das coisas damos o nome de *Web of Things* (Trad. Web das Coisas). É no nível de aplicação, acima da camada de conectividade de rede que ocorre a integração dos dispositivos na Web. A Web das Coisas faz uso de padrões da Web existentes e já conhecidos na programação Web (ex.: REST, HTTP, JSON), Web semântica (JSON-LD, Microdados, etc.), Web em tempo real (*Websockets*) e a Web social (redes sociais) (GUINARD & TRIFA, 2016).

Porém, não basta publicar as informações oriundas de tais coisas inteligentes na Web. Para maior aproveitamento e compreensão destas informações, se faz necessário o uso dos conceitos de Web semântica, viabilizando as máquinas o acesso e a compreensão de tais informações, contribuindo para utilização efetiva destes objetos, bem como, acrescentando desempenho na resolução de problemas.



## **2.8. *Semantic Web of Things (SWoT)***

Fornecer descrição semântica processável a “coisas inteligentes” é fundamental e gera pesquisas mais conclusivas sem necessidade de assistência humana. Porém, é necessário superar o desafio de como interligar, reconhecer e armazenar todas as informações geradas pela Web Semântica das Coisas (SWoT). Essas informações incluem, mas não se limitam a informações sobre os objetos em si, o tipo de etiqueta do objeto, o produtor, dados sobre o estado de origem, manipulação, data de produção, trajetória de transporte do produto, avaliações de outros consumidores e onde ele é vendido.

Uma forma de associar e organizar essas informações se faz por meio da Web Semântica, possibilitando a descrição das propriedades, das ligações dos dados e relação entre objetos, utilizando o modelo de dados e linguagem baseados em RDF para a interconexão de dados gerados pela IoT. Assim, motores de busca conseguem reunir objetos com propriedades comuns, abrindo novas dimensões da informação e do conhecimento. Outro desafio se faz na criação de um vocabulário padrão para que os dados referentes as coisas inteligentes não sejam gerados em vários RDF's distintos (CUNHA, 2010).

### 3. TRAJETÓRIA METODOLÓGICA

As etapas da metodologia do trabalho são destacadas na Figura 11.



**Figura 11** – Trajetória metodológica.

Fonte: Adaptado de Rangel (2018)

Como etapa inicial, foi realizada uma revisão bibliográfica para levantamento do estado da arte do tema, fazendo uso da base de dados *SCOPUS (Elsevier)*, com acesso por meio do Portal de Periódicos Capes. A busca teve como foco trabalhos que relacionassem as seguintes áreas de interesse: IoT, Web Semântica e *Linked Data*. Posteriormente o estudo acerca do tema, uma nova etapa foi iniciada com a elicitação dos requisitos funcionais da aplicação semântica a ser desenvolvida. A partir da análise dos requisitos, iniciou-se a etapa de construção dos modelos ontológicos. Em seguida, a aplicação semântica *Linked Data* foi desenvolvida, fazendo uso da plataforma *Linked Data Callimachus*<sup>21</sup>. Encerrando, têm-se exemplos de uso realistas para corroborar a eficácia da aplicação.

#### 3.1. Revisão Bibliográfica

O termo “*Internet das Coisas*” foi pronunciado por um especialista chamado Kevin Ashton, em uma reunião, em 1991. Anos mais tarde, este mesmo especialista publicou o artigo intitulado como “*That 'Internet of Things' thing.*”, lançado no *RFID Journal*<sup>22</sup>. A mensagem que o artigo passou foi da possível existência de computadores que conhecessem “tudo” de “todas” as coisas no geral, sem a necessidade de informações inseridas pelos humanos neles. Mais precisamente, que os computadores fossem capacitados para “sentir” o mundo real sem a

<sup>21</sup> <http://callimachusproject.org/>

<sup>22</sup> <https://www.rfidjournal.com/>

ajuda dos humanos. Estes, poderiam utilizar RFID<sup>23</sup> (*Radio-Frequency IDentification* Trad. identificação por rádio Frequência) e sensores para tal proposta (ASHTON, 1999).

No período em que o artigo foi lançado, muitas questões foram levantadas acerca das tecnologias, pois, como seria possível realizar conexão de todas as coisas do mundo? Atualmente já é possível acompanhar, na literatura, mudanças significativas, como: custo mais baixo das rádios, surgimento do IPV6<sup>24</sup>, maior cobertura de dados móveis, criações de vocabulários sobre o domínio IoT, etc. Assim sendo, como exemplo, o Grupo de Internet das Coisas da Cisco (IOTG)<sup>25</sup> prevê que haverá mais de 50 bilhões de dispositivos conectados em 2020 (LOPEZ, 2013). No ano de 2009, o W3C deu início ao projeto para criação da ontologia SSN (*Semantic Sensor Network Ontology*)<sup>26</sup> para descrição de dados de sensores em termos de suas capacidades técnicas, processo de medição, observações e implantação seguindo os padrões da Web Semântica (BARNAGHI *et al.*, 2011). É possível hoje, já obter uma nova versão da ontologia SSN publicada em 2017, incluindo termos para descrição de sensores, atuadores e amostradores (HALLER *et al.* 2017). O W3C WG vem trabalhando na definição e padronização de um modelo formal comum para descrição da coisa inteligente, nomeada como TD (*Web of Things (WoT) Thing Description*)<sup>27</sup>. Atualmente, muitos trabalhos envolvendo a área de IoT, WoT, e SWoT começaram a despontar. Os avanços no desenvolvimento de dispositivos físicos, ontologias e os incentivos para publicação de dados ligados na Web refletiram em um crescente número de aplicações *Linked Data* relacionadas à área de IoT nos últimos anos.

Com base nos vocabulários que definem o método (ontologias WoT), o objeto de pesquisa (sensores, atuadores, IoT) e o objetivo do presente estudo (descrição da interface de acesso as coisas inteligentes), foi feita uma pesquisa referente aos artigos indexados a base de dados *Scopus (Elsevier)*, com acesso por meio do Portal de Periódicos Capes. A escolha desta base deve-se à sua representatividade e abrangência, com mais de 21 mil periódicos de todas as áreas de conhecimento. Quanto ao recorte temporal, a pesquisa contempla uma década (por serem temas bastante novos), ou seja, o período de 2008 a 2018, conforme Quadro 1.

---

<sup>23</sup> [http://dbpedia.org/page/Radio-frequency\\_identification](http://dbpedia.org/page/Radio-frequency_identification)

<sup>24</sup> <http://ipv6.br/>

<sup>25</sup> [http://www.cisco.com/web/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf)

<sup>26</sup> <https://www.w3.org/TR/vocab-ssn/>

<sup>27</sup> <https://w3c.github.io/wot-thing-description/>

**Quadro 1 - Áreas de Conhecimento e Critérios de Busca**

GRUPOS	A	TITLE-ABS-KEY ("Internet of Things" OR "IoT") AND ( LIMIT-TO ( PUBYEAR, 2018 ) OR LIMIT-TO ( PUBYEAR, 2017 ) OR LIMIT-TO ( PUBYEAR, 2016 ) OR LIMIT-TO ( PUBYEAR, 2015 ) OR LIMIT-TO ( PUBYEAR, 2014 ) OR LIMIT-TO ( PUBYEAR, 2013 ) OR LIMIT-TO ( PUBYEAR, 2012 ) OR LIMIT-TO ( PUBYEAR, 2011 ) OR LIMIT-TO ( PUBYEAR, 2010 ) OR LIMIT-TO ( PUBYEAR, 2009 ) OR LIMIT-TO ( PUBYEAR, 2008 ))
	B	TITLE-ABS-KEY ("Semantic WEB" OR "WEB 3.0" OR "web of knowledge") AND ( LIMIT-TO ( PUBYEAR, 2018 ) OR LIMIT-TO ( PUBYEAR, 2017 ) OR LIMIT-TO ( PUBYEAR, 2016 ) OR LIMIT-TO ( PUBYEAR, 2015 ) OR LIMIT-TO ( PUBYEAR, 2014 ) OR LIMIT-TO ( PUBYEAR, 2013 ) OR LIMIT-TO ( PUBYEAR, 2012 ) OR LIMIT-TO ( PUBYEAR, 2011 ) OR LIMIT-TO ( PUBYEAR, 2010 ) OR LIMIT-TO ( PUBYEAR, 2009 ) OR LIMIT-TO ( PUBYEAR, 2008 ))
	C	TITLE-ABS-KEY ("Linked Data" OR "Linked Open Data") AND ( LIMIT-TO ( PUBYEAR, 2018 ) OR LIMIT-TO ( PUBYEAR, 2017 ) OR LIMIT-TO ( PUBYEAR, 2016 ) OR LIMIT-TO ( PUBYEAR, 2015 ) OR LIMIT-TO ( PUBYEAR, 2014 ) OR LIMIT-TO ( PUBYEAR, 2013 ) OR LIMIT-TO ( PUBYEAR, 2012 ) OR LIMIT-TO ( PUBYEAR, 2011 ) OR LIMIT-TO ( PUBYEAR, 2010 ) OR LIMIT-TO ( PUBYEAR, 2009 ) OR LIMIT-TO ( PUBYEAR, 2008 ))
GRUPOS	$A \cap B$	TITLE-ABS-KEY ("internet of things" OR "IOT") AND ( TITLE-ABS-KEY ("Semantic WEB" OR "WEB 3.0" OR "web of knowledge") AND ( LIMIT-TO ( PUBYEAR, 2018 ) OR LIMIT-TO ( PUBYEAR, 2017 ) OR LIMIT-TO ( PUBYEAR, 2016 ) OR LIMIT-TO ( PUBYEAR, 2015 ) OR LIMIT-TO ( PUBYEAR, 2014 ) OR LIMIT-TO ( PUBYEAR, 2013 ) OR LIMIT-TO ( PUBYEAR, 2012 ) OR LIMIT-TO ( PUBYEAR, 2011 ) OR LIMIT-TO ( PUBYEAR, 2010 ) OR LIMIT-TO ( PUBYEAR, 2009 ) OR LIMIT-TO ( PUBYEAR, 2008 ))
	$A \cap C$	TITLE-ABS-KEY ("Internet of Things" OR "IoT") AND TITLE-ABS-KEY ("Linked Data" OR "Linked Open Data") AND ( LIMIT-TO ( PUBYEAR, 2018 ) OR LIMIT-TO ( PUBYEAR, 2017 ) OR LIMIT-TO ( PUBYEAR, 2016 ) OR LIMIT-TO ( PUBYEAR, 2015 ) OR LIMIT-TO ( PUBYEAR, 2014 ) OR LIMIT-TO ( PUBYEAR, 2013 ) OR LIMIT-TO ( PUBYEAR, 2012 ) OR LIMIT-TO ( PUBYEAR, 2011 ) OR LIMIT-TO ( PUBYEAR, 2010 ) OR LIMIT-TO ( PUBYEAR, 2009 ) OR LIMIT-TO ( PUBYEAR, 2008 ))
	$B \cap C$	TITLE-ABS-KEY ("Semantic WEB" OR "WEB 3.0" OR "web of knowledge") AND TITLE-ABS-KEY ("Linked Data" OR "Linked Open Data") AND ( LIMIT-TO ( PUBYEAR, 2018 ) OR LIMIT-TO ( PUBYEAR, 2017 ) OR LIMIT-TO ( PUBYEAR, 2016 ) OR LIMIT-TO ( PUBYEAR, 2015 ) OR LIMIT-TO ( PUBYEAR, 2014 ) OR LIMIT-TO ( PUBYEAR, 2013 ) OR LIMIT-TO ( PUBYEAR, 2012 ) OR LIMIT-TO ( PUBYEAR, 2011 ) OR LIMIT-TO ( PUBYEAR, 2010 ) OR LIMIT-TO ( PUBYEAR, 2009 ) OR LIMIT-TO ( PUBYEAR, 2008 ))
	$A \cap B \cap C$	TITLE-ABS-KEY ("Internet of Things" OR "IoT") AND TITLE-ABS-KEY ("Semantic WEB" OR "WEB 3.0" OR "web of knowledge") AND TITLE-ABS-KEY ("Linked Data" OR "Linked Open Data") AND ( LIMIT-TO ( PUBYEAR, 2018 ) OR LIMIT-TO ( PUBYEAR, 2017 ) OR LIMIT-TO ( PUBYEAR, 2016 ) OR LIMIT-TO ( PUBYEAR, 2015 ) OR LIMIT-TO ( PUBYEAR, 2014 ) OR LIMIT-TO ( PUBYEAR, 2013 ) OR LIMIT-TO ( PUBYEAR, 2012 ) OR LIMIT-TO ( PUBYEAR, 2011 ) OR LIMIT-TO ( PUBYEAR, 2010 ) OR LIMIT-TO ( PUBYEAR, 2009 ) OR LIMIT-TO ( PUBYEAR, 2008 ))

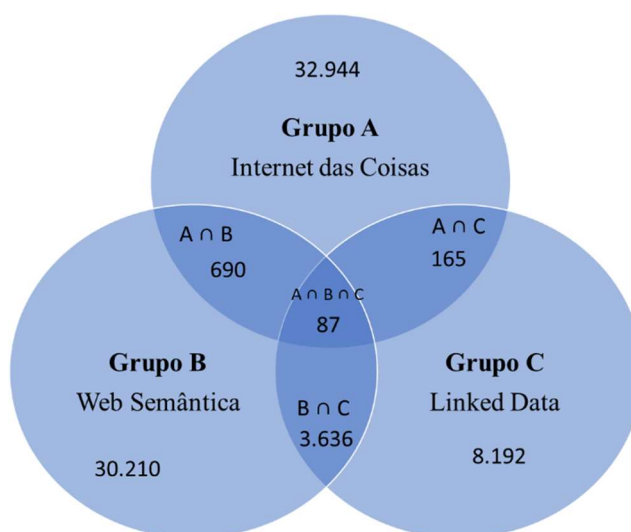
Fonte: Elaboração própria

Foram identificados, conforme cada área de conhecimento, um número expressivo de artigos, como pode ser visto no Quadro 2, a seguir. Logo em seguida, é possível acompanhar uma nova etapa, constituída das relações entre as áreas de conhecimento, onde foi desenvolvido um Gráfico de *Venn* para melhor ilustrar o resultado obtido, conforme Figura 12.

**Quadro 2** - Área de Conhecimento e a quantidade de artigos encontrados.

Áreas de Conhecimento	Quantidade de Artigos
Grupo A	32.944
Grupo B	30.210
Grupo C	8.192

Fonte: Elaboração própria

**Figura 12:** Gráfico de *Venn* da Análise Bibliográfica.

Fonte: Elaboração própria

Acompanhando a Figura 12, observa-se as relações entre as áreas de conhecimento exploradas. Como resultado da pesquisa, foram retornados 87 documentos caracterizando a relação dos três (3) Grupos. O intuito da busca foi encontrar pesquisas relevantes acerca do domínio estudado, sobre os conceitos Web Semântica e *Linked Data* aplicados à área de Internet das Coisas. A seguir, no Quadro 3, é possível acompanhar anualmente, o número de publicações da pesquisa efetuada.

**Quadro 3** - Número de publicações anual envolvendo as 3 áreas de conhecimento.

Ano	Número de Publicações
2017	22
2016	23
2015	15
2014	12
2013	7
2012	5
2011	2
2009	1

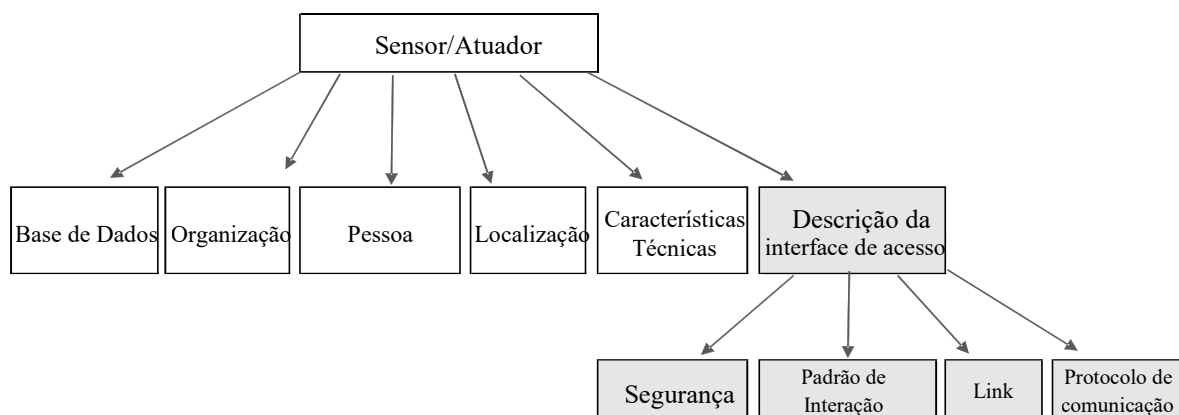
Fonte: SCOPUS (2018)

Os trabalhos retornados na busca, conforme Quadro 3, refletem a esforços contínuos nos últimos anos, para desenvolvimentos e/ou aperfeiçoamento de vocabulários acerca do domínio IoT, bem como, aplicações semânticas atuantes neste domínio. É possível, no Capítulo 7, acompanhar os diversificados trabalhos referentes a esta pesquisa.

### 3.2. Elicitação de requisitos

A diversidade de trabalhos encontrados em vários campos de atuação como computação, engenharia, medicina, biologia, reflete o amplo universo em que o domínio IoT está envolvido e a necessidade de padronização para provimento de sucesso na interoperabilidade dos objetos envolvidos nesse domínio. Assim, a análise de artigos constitui-se como o ponto de partida da etapa de elicitação de requisitos, sobretudo, a análise do trabalho realizado por Rangel (2018), uma vez que o presente trabalho dá continuidade ao por ela realizado, bem como a ontologia, parte do Projeto VICINITY, nomeada como *Vicinity ontology model for Web of Things (WoT ontology)* (SERENA et al., 2017), que formaliza a base para descrição da interface de acesso a uma coisa inteligente, objetivo deste trabalho.

Estudos de diversificados trabalhos na área e reuniões periódicas com o orientador foram realizados a fim de elencar os diferentes requisitos. É importante reiterar que este trabalho, como continuação de Rangel (2018), trata da descrição formal da interface de acesso a uma coisa inteligente (sensores e atuadores). A Figura 13 apresenta os diferentes aspectos adotados para tal descrição.



**Figura 13:** Aspectos da Descrição de sensores e atuadores.  
Fonte: Adaptado de Rangel (2018).

Na Figura 13, é possível visualizar os vários aspectos (Base de Dados, Organização, Pessoa, Localização, Características Técnicas) presentes na aplicação realizada no trabalho de Rangel (2018), como também, o conteúdo presente na coloração cinza, correspondente aos aspectos adicionais, determinados neste trabalho.

Na Tabela 3, são apresentadas as informações e/ou exemplos de uso dos recursos previstos no catálogo. Todos os recursos foram declarados a fim de prover maior entendimento acerca do estado atual da aplicação.

Os recursos indicados na Tabela 3 associados a este trabalho são: Sensor, Atuador, Segurança, Protocolo de Comunicação. Vale destacar que no recurso Sensor e recurso Atuador, ocorre além do cadastro presente no Trabalho de Rangel (2018) acerca das descrições técnicas, o cadastro também da interface de acesso. Por mais que sensores e atuadores sejam cadastrados de maneira independente, tem-se como propósito disponibilizar dados dos dispositivos em questão para possibilitar que agentes de software promovam a integração dos mesmos e obtenham como resultado a solução de diversos problemas. Os demais recursos descritos foram reaproveitados do trabalho de Rangel (2018) e apresentados para melhor compreensão.

**Tabela 3** - Descrição atual dos recursos que podem ser cadastrados na Aplicação.

Recurso	Descrição/Exemplo
Sensor	Exemplar físico instalado que detecta e mede uma propriedade física (p. ex.: acelerômetro, etc.). É considerado uma coisa inteligente e tem a sua interface de acesso descrita.
Modelo de Sensor	Classe que descreve as características de um modelo de sensor. Um sensor físico é uma instância de uma classe (modelo).
Propriedade de sistema	Característica que afeta a função principal do sistema (p. ex.: acurácia, precisão, frequência, etc.)
Propriedade de operação	Característica que influencia na operação normal de um sistema (p. ex.: tensão de alimentação)
Propriedade de sobrevivência	Característica que garante que o sistema funcione sem danos (p. ex.: tempo de vida)
Tipo de Propriedade	Classe que representa um tipo propriedade que pode ser medida (p. ex.: temperatura, vazão, pressão, nível, etc.). Um Tipo de propriedade é uma classe.
Propriedade	Propriedade de um objeto de interesse (p. ex.: temperatura do ar, temperatura da sala de estar, vazão do rio Paraíba, nível do tanque da cozinha, etc.). Uma Propriedade é uma instância da classe Tipo de Propriedade. Procedimento Método que especifica funcionamento de sensor ou atuador (p. ex.: fluxo de trabalho)
Objeto de interesse	Um objeto cuja propriedade está sendo estimada no decurso de uma observação (p. ex.: Ao medir a temperatura de sala de estar, a sala de estar é o objeto de interesse).
Implantação	Implantação de um sistema (ou dispositivo), para um propósito específico, em uma plataforma (p. ex.: sensor instalado em na parede da sala de estar.)
Plataforma	Entidade que hospeda outras entidades, particularmente sensores, atuadores ou outras plataformas (p. ex.: sensores, atuadores ou outras plataformas (p. ex.: um telefone móvel, a parede da sala de
Pessoa	Pessoas relacionadas a um dispositivo (p. ex.: responsável pelo sensor).
Organização	Instituições sociais (p. ex.: escolas, instituições governamentais, etc.) Base de dados Fontes de dados de sensores na Web
Atuador	Exemplar físico instalado que controla algum mecanismo ou muda o estado de um sistema (p. ex.: motor). É considerado uma coisa inteligente e tem a sua interface de acesso descrita.
Modelo de Atuador	Classe que descreve as características de um modelo de atuador. Um atuador físico é uma instância de uma classe (modelo).
Segurança	Classe usada para fornecer informações de metadados de acesso (auto-contidas) da Coisa para transmitir informações com segurança através de todos os seus recursos.
Protocolo de Comunicação	Acordo entre as partes em comunicação sobre como a comunicação deve proceder.

Fonte: Adaptado de Rangel (2018)

### 3.3. Modelo ontológico

Como próxima etapa, já de posse do conhecimento sobre os requisitos funcionais da aplicação, ontologias consagradas foram estudadas e selecionadas. Com a identificação das ontologias para o trabalho, conhecendo suas classes e propriedades, foi possível descrever formalmente os recursos. O uso combinado destas ontologias, é apresentado no Capítulo 5.



### 3.3.1. Seleção de ontologias

Como fator categórico para o desenvolvimento da aplicação, encontra-se a fase de seleção das ontologias (vocabulários). Duas premissas são fundamentais: (1) conhecer as ontologias aplicáveis aos tipos de recursos envolvidos no trabalho; (2) verificar o reconhecimento ou abrangência de uso destas ontologias na Web. No catálogo *Linked Open Vocabularies for Internet of Things (LOV4IoT)*<sup>28</sup> são encontrados aproximadamente 518 projetos de pesquisa baseados em ontologia para IoT e seus domínios de aplicação. No catálogo *Linked Open Vocabularies (LOV)*<sup>29</sup>, constam as ontologias *Linked Data* de maior reuso, que vem se tornando padrão de *facto*. Assim, fazendo uso dos catálogos LOV4IoT e LOV, foi possível selecionar ontologias de interesse, de acordo com cada funcionalidade da aplicação, todas elas desenvolvidas em consonância com os padrões estabelecidos pelo W3C. O vocabulário que norteia este trabalho é a ontologia, parte do Projeto VICINITY, nomeada como *Vicinity ontology model for Web of Things (WoT ontology)*, que formaliza a base para descrição da interface de acesso a uma coisa inteligente. Como o trabalho também conta com recursos já desenvolvidos na aplicação de Rangel (2018), para fins de conhecimento tem a ontologia *Semantic Sensor Network (SSN)*, que compreende a maior parte dos termos utilizados para descrever tais dispositivos. Além da ontologia SSN, outras ontologias foram adotadas para descrever os demais recursos ou requisitos funcionais. A Tabela 4 abrange as ontologias utilizadas na aplicação.

---

<sup>28</sup> <https://lov4iot.appspot.com/?p=ontologies>

<sup>29</sup> <http://lov.okfn.org/dataset/lov/>

**Tabela 4 - Ontologias adotadas na aplicação.**

Prefixo	Nome da ontologia	Descrição/Finalidade
foaf	Friend of a friend (BRICKLEY & MILLER, 2014)	Descreve pessoas e organizações
prov	Provenance Ontology (LEBO, SAHOO & MCGUINNESS, 2013)	Vocabulário que permite descrever a proveniência (origem) de dados publicados na Web para que a confiabilidade da informação seja avaliada.
rdfs	RDF Schema (BRICKLEY & GUHA, 2014)	Oferece padrões para modelagem de dados em RDF. Metavocabulário para criação de vocabulários simples, contendo apenas classes e propriedades.
rdf	Resource Description Foundation (CYGANIAK, WOOD & LANTHALER, 2014)	Vocabulário contendo conceitos elementares e sintaxe abstrata do modelo RDF.
ssn	Semantic Sensor Network Ontology (HALLER et al., 2017)	Descreve sensores, atuadores, observações e conceitos relacionados
vcard	vCard (IANNELLA & MCKINNEY, 2014)	Descreve pessoas e organizações.
sosa	Sensor, Observation, Sample and Actuator (HALLER et al., 2017)	Vocabulário contendo conceitos elementares da ontologia SSN
owl	Web Ontology Language (DEAN & SCHREIBER, 2004)	Metaontologia para criação de ontologias, sobretudo, baseadas em lógica descritiva.
ssnsystem	System Capability module (HALLER et al., 2017)	Extensão da ontologia SSN para descrição de sistemas em termos de suas capacidades, faixas de operação e de sobrevivência.
schema	Schema.org (SCHEMA.ORG, 2011)	Vocabulário multidomínio, criado pela Google, Yahoo, Microsoft e Yandex para descrever entidades, seus relacionamentos e ações.
dcterms	Dublin Core (HAKALA, 2000)	Descreve metadados genéricos.
wot	Web of Things ontology (SERENA et al., 2017)	Descreve o domínio <i>Web of Things</i>
om	Ontology of units of Measure (GKOUTOS et al., 2012)	Fornecer classes, instâncias e propriedades que representam os diferentes conceitos usados para definir e usar medidas e unidades.
unit	QUDT Units of Measure Individuals (MASTERS, 2010)	O Vocabulário de Unidades QUDT é uma especificação RDF / OWL de unidades comuns de medida nos domínios da ciência e engenharia. Todas as Unidades SI estão incluídas.

Fonte: Adaptado de Rangel (2018)

Cada funcionalidade identificada na etapa de elicitação de requisitos representa o cadastro de um novo recurso que pode consistir em uma instância ou subclasse de uma dada classe de uma ou mais ontologias selecionadas. Assim, cada um destes recursos é descrito de acordo com o tipo ou a classe à qual pertence, sob o ponto de vista das ontologias e

metaontologias empregadas. A Tabela 5 apresenta os recursos que podem ser cadastrados na aplicação e a definição dos seus “tipos”.

**Tabela 5** – Tipos de recursos.

Recursos	Tipos
Sensor (Sensor)	?SensorModel sosa:Sensor ssn:System wot:Thing
Atuador (Actuator)	?ActuatorModel sosa:Actuator ssn:System wot:Thing
Segurança (Security)	wot:Security
Protocolo de Comunicação (Communication Protocol)	wot:CommunicationProtocol

Fonte: Elaboração Própria

Na Tabela 5, tipos com "?" na frente do respectivo nome, significa que se trata de uma classe definida pelo usuário da aplicação. Enquanto tipos sem "?" no nome, se trata de uma classe pré-definida em alguma ontologia. Por exemplo, ao catalogar um dado sensor, tal dispositivo deve ser associado ao seu modelo, ou seja, deve estar atrelado a um tipo de modelo. Como modelos de sensores consistem em classes criadas pelo usuário na aplicação, são identificados como sendo “?SensorModel”. Para qualificar as classes e instâncias, ou seja, para descrever os recursos apresentados na Tabela 5, diversas propriedades das ontologias são utilizadas. Maiores detalhes sobre cadastro das características técnicas e funcionalidades de sensores/atuadores podem ser obtidos em Rangel (2018). Já os detalhes de cadastro da interface de uso dos sensores/atuadores, objeto deste trabalho, serão detalhados nos próximos capítulos.

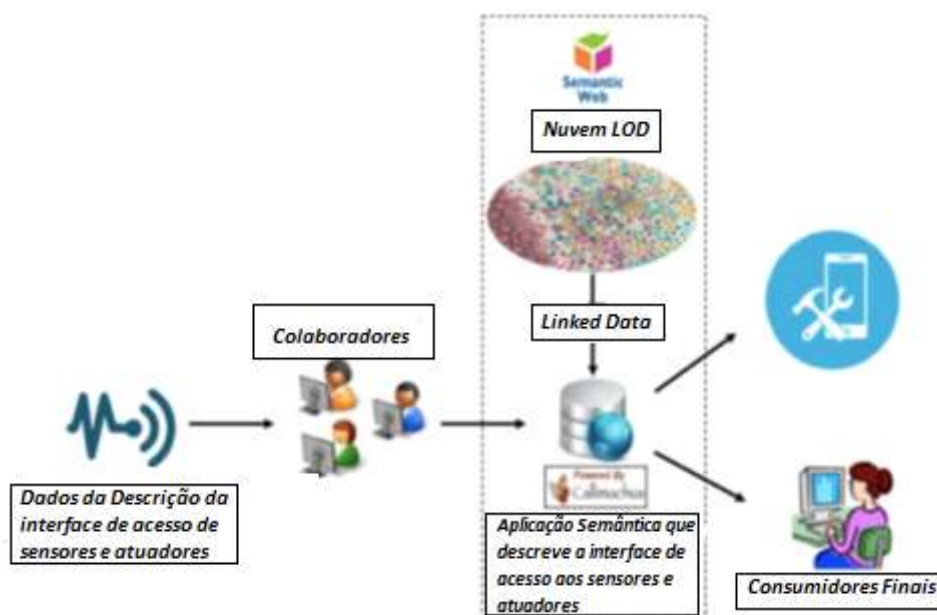
### 3.4. Desenvolvimento da Aplicação

Esta etapa da metodologia envolve a seleção do *framework* para desenvolvimento do catálogo semântico proposto, no caso, a plataforma *Linked Data Callimachus*, e o desenvolvimento da aplicação em si. Nesta última etapa, faz-se uso de cenário de uso realista para demonstrar o processo de desenvolvimento e corroborar a eficácia da aplicação. Os detalhes sobre a construção desta aplicação são apresentados no capítulo 6.

#### 3.4.1. Seleção do *framework*

Dado que o objetivo deste trabalho é desenvolver uma aplicação *Linked Data* para catalogação de sensores e atuadores na Web, optou-se pelo uso da plataforma *Callimachus*. A plataforma facilita o desenvolvimento de aplicações semânticas *Linked Data* por seguir padrões

da Web e os princípios *Linked Data*, oferecendo: armazenamento de grafos RDF, consultas SPARQL, HTML5 com RDFa, CSS3, Javascript, *templates* HTML CRUD (*Create, Read, Update, Delete*) semiprontos, ambiente de desenvolvimento integrado e ambiente para visualização e publicação de RDF e HTML na Web. Além disso, oferece um SPARQL *endpoint* e um API RESTful podendo ser utilizado para construir aplicações complexas e dinâmicas com conteúdo RDF. A Figura 14 apresenta uma visão macro da arquitetura da aplicação proposta.



**Figura 14** – Macro-arquitetura da aplicação.

Fonte: Adaptado de Rangel (2018)

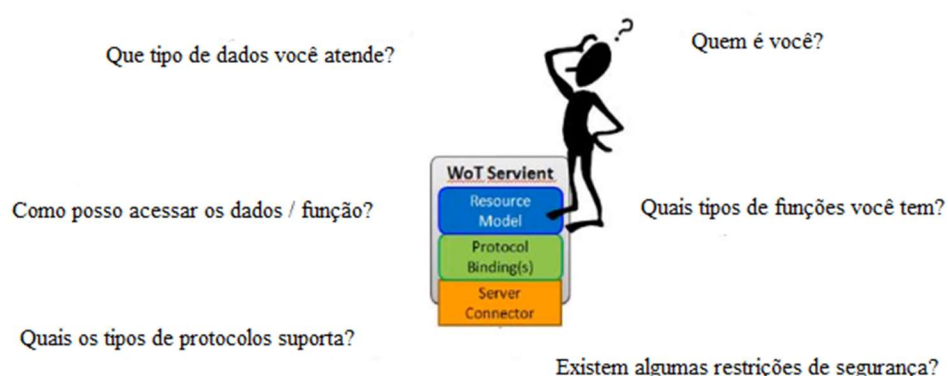
Na Figura 14 é possível visualizar o funcionamento da aplicação. O intuito é que colaboradores distribuídos geograficamente cadastrem metadados de sensores e atuadores, bem como, sua descrição de acesso, seguindo os padrões da Web Semântica, interligando tais dados a dados de outras bases semânticas na Web. Deste modo, é possível com a interface de acesso cadastrada, tornar disponível informações de sensores e atuadores para propiciar que agentes de software realizem a integração das informações.

#### 4. VICINITY ONTOLOGY MODEL FOR WEB OF THINGS (WOT ONTOLOGY)

No contexto da WoT, a descoberta de uma “coisa inteligente”, pode ser imaginada como uma busca por página Web: os usuários emitem os critérios de pesquisa que podem resultar na descoberta de resultados relevantes ou até mesmo desconhecidos. Fatalmente, para se obter os resultados válidos, se faz necessário ter os meios para descrever os recursos, de maneira que a busca, com os critérios de pesquisa fornecidos, possa ser satisfeita de forma precisa. Porém, a descoberta de “coisas inteligentes” no universo IoT carece de formatos comuns e sintaxes para efeito em um escopo global. Com isso, um modelo formal comum para descrever essas “coisas inteligentes”, suas características e habilidades se faz necessário (SERENA *et al*, 2017).

A W3C *Web of Things* (WoT) foi criada para permitir a interoperabilidade entre as plataformas de IoT e domínios de aplicação. Fornece mecanismos para descrever formalmente interfaces IoT permitindo que dispositivos e serviços de IoT se comuniquem entre si, independentemente de sua implementação subjacente e de diversos protocolos de rede.

A WoT pretende fazer com que tudo que envolva o ecossistema da IoT faça parte da Web. Assim todas as “coisas inteligentes” acessíveis por meio de uma interface Web poderão fazer parte da WoT. Contudo, essas interfaces deverão ser abastecidas de detalhes, características destas coisas inteligentes, proporcionando ao usuário não apenas saber a quais dispositivos estão se referindo, mas também, onde e como alcançá-los na Web. Na Figura 15, é possível visualizar o cenário.

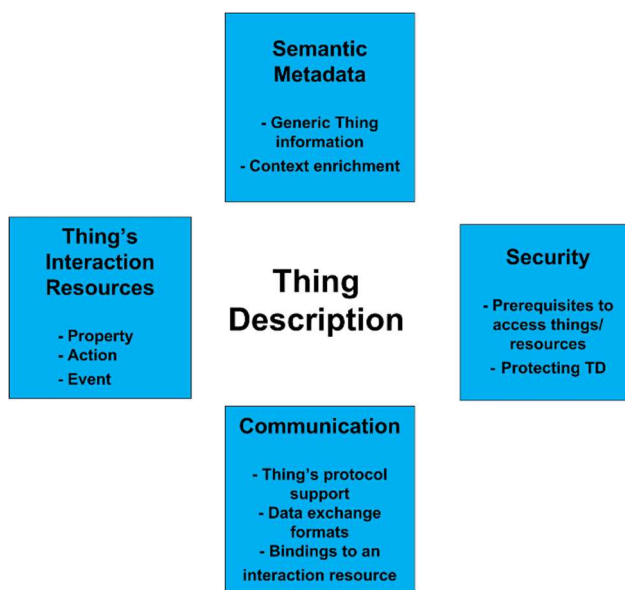


**Figura 15** – Cenário com as questões para descoberta de uma “coisa inteligente”.  
Fonte: Adaptado de W3C WoT Thing Description<sup>30</sup>

<sup>30</sup> <https://www.w3.org/2015/spatial/wiki/images/2/20/Td-wot-introduction-for-SDWG-TPAC-2016.pptx>

O documento “*Web of Things (WoT) Architecture*<sup>31</sup>” (Trad. Arquitetura da Web das Coisas), disponibilizado pela W3C em 2017, derivado de um conjunto de casos de uso, está focado no escopo da padronização do W3C WoT. Consiste em três blocos de construção: *WoT Thing Description*<sup>32</sup> (Trad. Descrição da Coisa de WoT); *WoT Binding Templates*<sup>33</sup> (Trad. modelos de vinculação WoT) e *WoT Scripting API* (Trad. API de scripts da WoT)<sup>34</sup>.

O bloco central, *WoT Thing Description* (TD), consiste em metadados semânticos da coisa, um padrão ou modelo de interação (Propriedade, Ação e Evento), um esquema semântico para tornar os modelos de dados compreensíveis à máquina e recursos para *Web Linking* para expressar as relações entre coisas. Deste modo, definir uma TD para um dispositivo existente permite que o dispositivo participe da WoT sem precisar fazer nenhuma modificação no próprio dispositivo. Em suma, a TD fornece metadados que descrevem como comunicar com cada dispositivo (coisa) em particular. Na Figura 16, é possível visualizar os elementos que compõem uma TD.



**Figura 16** – Características de uma *Thing Description*.  
Fonte: Adaptado de W3C WoT Thing Description<sup>35</sup>

Na Figura 17 são apresentadas as possíveis formas de se descobrir TD. Para as “coisas inteligentes” próximas, por exemplo, usando Bluetooth, “coisas inteligentes” na mesma rede local e “coisas inteligentes” listadas em um repositório, por exemplo, um hub ou serviço

<sup>31</sup> <https://www.w3.org/TR/wot-architecture/>

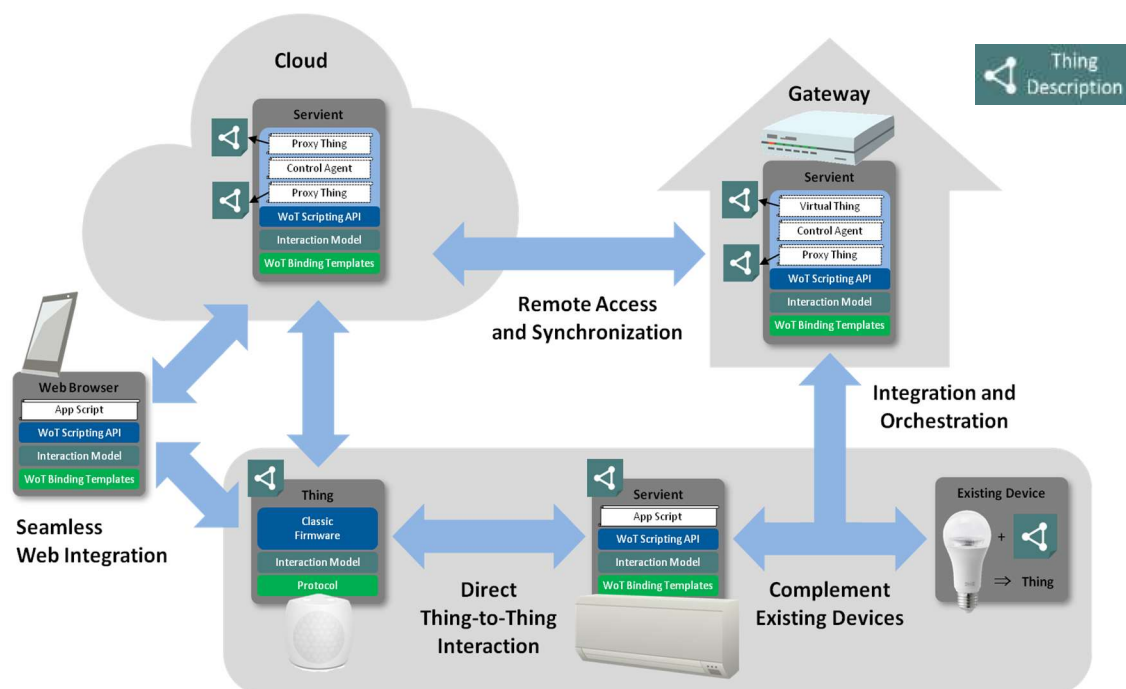
<sup>32</sup> <https://www.w3.org/TR/wot-thing-description/>

<sup>33</sup> <https://w3c.github.io/wot-binding-templates/>

<sup>34</sup> <https://w3c.github.io/wot-scripting-api/>

<sup>35</sup> <https://www.w3.org/2015/spatial/wiki/images/2/20/Td-wot-introduction-for-SDWG-TPAC-2016.pptx>

baseado em nuvem. Os metadados permitem pesquisar as “coisas inteligentes” com base em suas descrições.



**Figura 17** – Arquitetura abstrata do W3C WoT.

Fonte: [W3C.github.io](https://w3c.github.io)<sup>36</sup>

TD é um bloco de construção central em um sistema habilitado para Web das Coisas e pode ser considerado como o ponto de entrada dessa “coisa inteligente” conectada, assim como um “index.html”. Baseado em suas especificações, foi desenvolvida uma ontologia, parte do Projeto VICINITY, nomeada como *WoT ontology (Vicinity ontology model for Web of Things)*, que formaliza a base para descrição da interface de acesso a uma coisa inteligente.

O Projeto VICINITY, financiado pelo Programa-Quadro Horizonte 2020 da União Européia para Pesquisa e Inovação, apresenta como principal objetivo fornecer aos proprietários de infraestruturas de IoT uma interoperabilidade descentralizada. Ou seja, conecta diferentes objetos inteligentes em uma “rede social” chamada vizinhança virtual, onde os proprietários de infraestrutura controlam seus dispositivos e dados compartilhados. A abordagem de interoperabilidade VICINITY depende de ontologias que serão exploradas em toda a infraestrutura VICINITY.

A *WoT ontology* conta com atuação de pesquisadores que fazem parte do WoT IG e WoT WG, estando assim de acordo com a visão do W3C. A *WoT Ontology* possui os conceitos intimamente ligados a *WoT Thing Description* como por exemplo: “coisa inteligente”,

<sup>36</sup> <https://w3c.github.io/wot-architecture/>

nomeada na ontologia como *Thing* (podendo ser uma entidade física ou virtual, como: sensor, atuador, uma sala, um grupo de dispositivos, etc.); três tipos de interação com a coisa inteligente - "Propriedade" (status de um Led, por exemplo), "Ação" (por exemplo, acender ou apagar o Led) e "Evento" (para o Led, por exemplo, a emissão de um alerta para aplicação acerca do Led). WoT *Ontology* foi desenvolvida para definir “o quê” (um dispositivo), “como” (o protocolo de comunicação, formato de dados) e “onde” (*endpoint*) as coisas podem ser descobertas e acessadas na WoT. A seguir, na Tabela 6, é apresentada a divisão da WoT *Ontology* mediante suas classes e propriedades. Em seguida, na Figura 18, uma visão geral da ontologia.

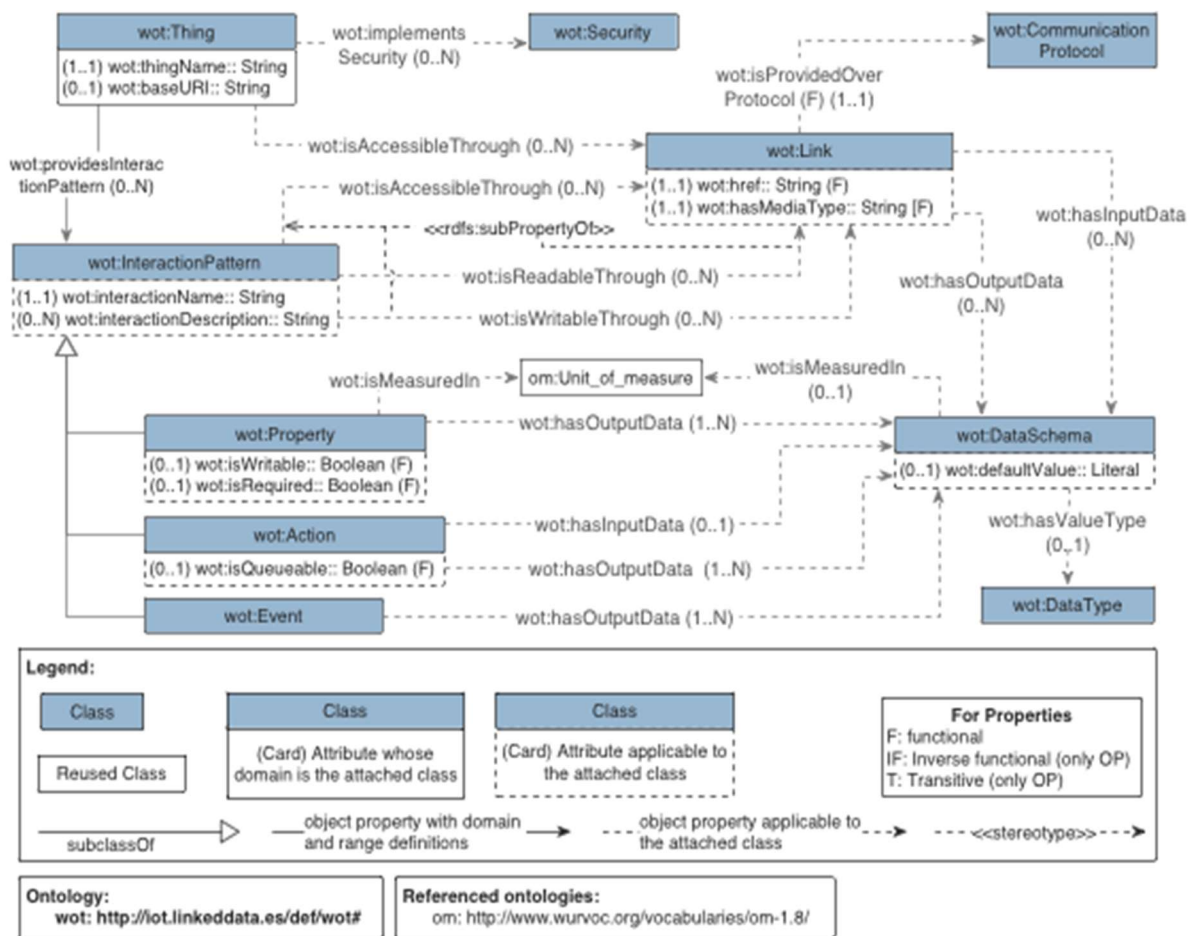
Vale notar que para arquitetura Web of Things (WoT), três entidades básicas (coisa WoT, WoT Cliente e WoT Servient) são definidas, podendo ser uma entidade coisa física como um sensor ou atuador, pode ser um cliente sendo um navegador ou um aplicativo no smartphone ou um serviente, uma entidade que pode ser visto como uma combinação de um cliente e um servidor (MCCOOL & RESHETOVA, 2018). No escopo deste trabalho as coisas inteligentes estudadas e descritas formalmente são sensor e atuador.

**Tabela 6** – Classes e Propriedades da ontologia WoT.

	Nome	Total
<b>Classes</b>	wot:Action, wot:Communication protocol wot:Data schema, wot:Data type wot:Event, wot:Interaction pattern wot:Link, wot:MediaType wot:Property, wot:Security wot:Thing, wot:Unit of measure	12
<b>Object Properties</b>	wot:describes thing, wot:hasInputData wot:hasOutputData, wot:hasPriorityEndpoint wot:hasValueType, wot:implementsSecurity wot:isAccessibleThrough, wot:isDescribedBy wot:isMeasuredIn, wot:isProvidedOverProtocol wot:isReadableThrough, wot:isWritableThrough wot:providedBy, wot:providesInteractionPattern	14
<b>Datatype Properties</b>	wot:baseURI, wot:defaultValue wot:hasMediaType, wot:href wot:interactionName, wot: interactionPatternDescription wot:isQueueable, wot:isRequired wot:isWritable, wot:thingName	10

Fonte: Elaboração Própria





**Figura 18:** Visão geral da ontologia WoT.

Fonte: Wot Ontology<sup>37</sup>

Na Figura 18 é apresentada a visualização gráfica da WoT *Ontology*, além disso, é possível visualizar uma breve descrição do processo de desenvolvimento e a disponibilidade dos recursos. Os principais conceitos definidos na ontologia, são *wot:Thing*, *wot:InteractionPattern*, *wot:DataSchema* e *wot:Link*. Vale a pena notar que a classe *wot:Thing* define coisas apenas no contexto da WoT.

De acordo ainda com a Figura 18, uma coisa inteligente particular está ligada aos padrões de interação que ela fornece por meio da propriedade *wot:providesInteractionPattern*. Um padrão de interação (Propriedade, Ação e Evento) é representado pelas classes *wot:Property*, *wot:Action* e *wot:Event*, respectivamente. Uma coisa inteligente ou um padrão de interação pode ser relacionado a um ou mais *endpoint*, representado na classe *wot:Link*. Isto ocorre diretamente ou através de seus padrões de interação por meio da propriedade *wot:isAccessibleThrough*. As principais informações fornecidas pela classe *wot:Link* são sobre a localização na Web na qual o serviço é fornecido, indicado pela propriedade *wot:href* e um

<sup>37</sup> <http://iot.linkeddata.es/def/wot/index-en.html>

tipo de mídia gerado, indicado pela propriedade *wot:mediaType*. A propriedade *wot:isProvidedOverProtocol* relaciona um *endpoint* com o protocolo de comunicação em que é servido, identificado pela classe *wot:CommunicationProtocol*. Uma interação pode ter dados de entrada ou saída associados ou ambos. Para essa modelagem, as propriedades *wot:hasInputData* e *wot:hasOutputData* foram criadas. Essas propriedades permitem a conexão de um dado padrão de interação para uma instância que será vinculada a um determinado tipo de dados e uma certa unidade de medida, identificada pela classe reusada *om:Unit\_of\_measure*, por meio das propriedades *wot:hasValueType* e *wot:isMeasuredIn* respectivamente. O classe *wot:Security* é opcional e pode ser usada para fornecer informações de metadados de acesso da Coisa (p. ex. OAuth<sup>38</sup> que é um padrão aberto para autorização, utilizado para permitir que os usuários da Internet possam fazer logon em sites de terceiros usando suas contas do Google, Facebook, Microsoft, Twitter, etc) para transmitir informações com segurança através de todos os seus recursos.

A WoT *ontology* está em desenvolvimento, juntamente com outras ontologias que compõem o domínio IoT pertencentes ao Projeto VICINITY, como: *Vicinity core model*<sup>39</sup>, *Vicinity WoT mappings model*<sup>40</sup>, *Vicinity adapters model*<sup>41</sup>. Novos conceitos podem ser incluídos ou estendidos a estas ontologias. O presente trabalho conta com a versão de revisão: 0.011 da WoT *ontology*.

---

<sup>38</sup> <http://dbpedia.org/page/OAuth>

<sup>39</sup> <http://iot.linkeddata.es/def/core#>

<sup>40</sup> <http://iot.linkeddata.es/def/wot-mappings#>

<sup>41</sup> <http://iot.linkeddata.es/def/adapters#>

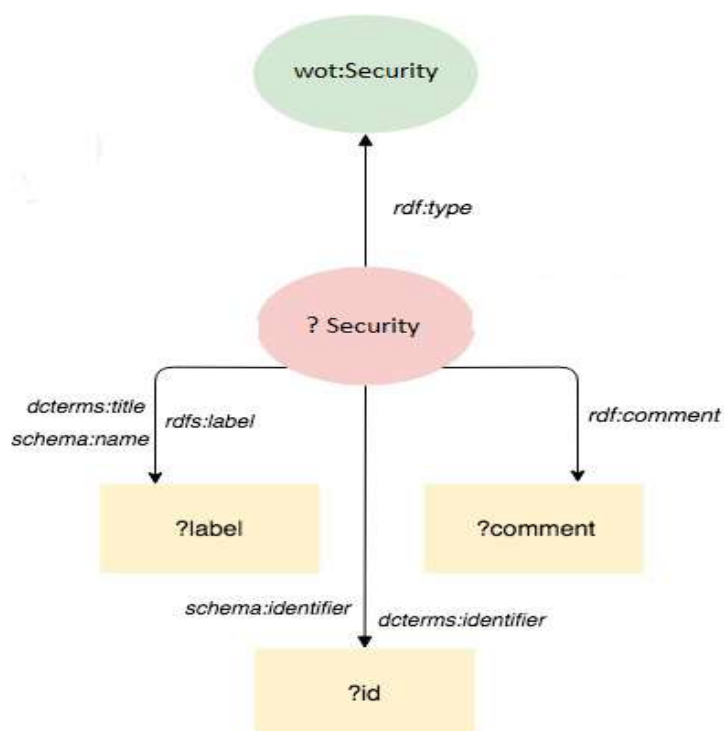
## 5. MODELO ONTOLOGICO DA APLICAÇÃO

Neste capítulo é apresentada a utilização integrada das ontologias selecionadas na descrição de cada requisito funcional especificado na etapa de elicitação de requisitos conforme item 3.2 do Capítulo 3. Os modelos são apresentados como *graph patterns* da linguagem SPARQL. Assim como em grafos RDF, em *graph patterns*, por convenção, elipses representam recursos e retângulos representam objetos cujos valores são literais. O símbolo "?" indica variáveis. Alguns modelos foram criados integralmente no escopo deste trabalho, outros são extensões de modelos originalmente criados no trabalho de Rangel (2018). Neste último caso, para destacar o acréscimo ao modelo original, as elipses dos recursos criados ou adaptados por este trabalho serão exibidas com o contorno em linha pontilhada.

### 5.1. Módulo Security (Segurança)

Para W3C, a WoT deve permitir o uso das melhores práticas em segurança e privacidade mesmo não sendo geralmente garantidas. Para questão de segurança, o sistema deve preservar sua integridade e funcionalidade mesmo quando estiver sujeito a ataques. Já para privacidade significa que o sistema deve manter a confidencialidade das informações pessoalmente identificáveis. Informações de segurança relacionadas as coisas precisam ser descritas em sua TD. Isso inclui informações para autenticação / autorização e criptografia.

De acordo com a WoT Ontology, a classe *wot:Security* é o recurso de segurança (opcional) usado para fornecer informações de metadados de acesso da coisa para transmitir informações com segurança através de todos os seus recursos. A Figura 19 apresenta o modelo ontológico criado para descrição de Segurança.



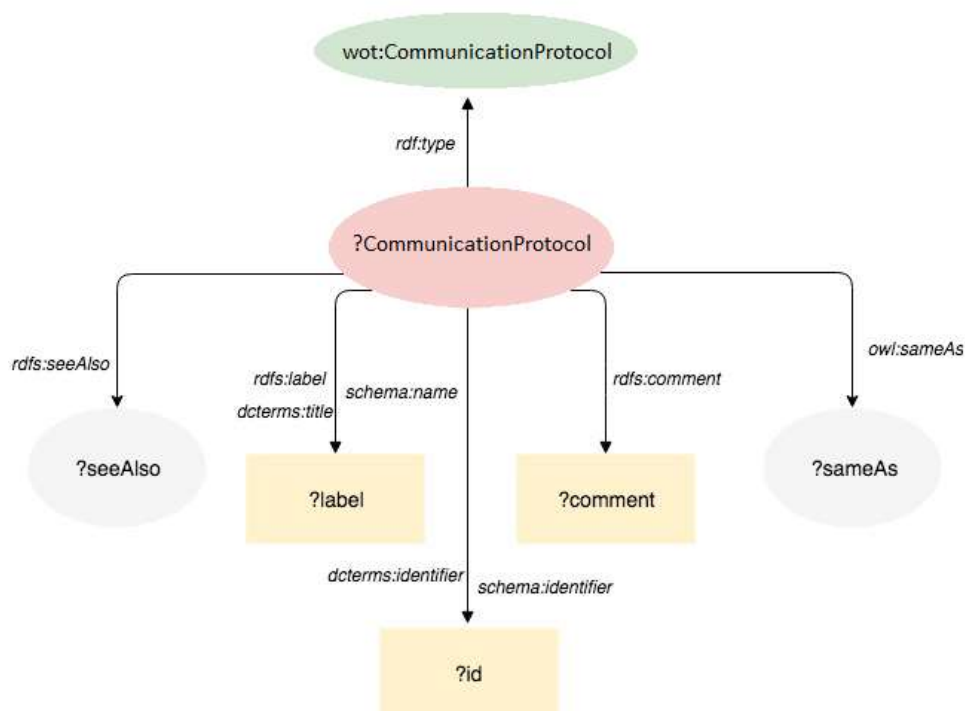
**Figura 19:** Modelo para descrição de Segurança.  
Fonte: Elaboração Própria

De acordo com o modelo de descrição de segurança, o recurso (*?Security*) é definido pela classe *wot:Security* por meio da propriedade *rdf:type*. As propriedades *schema:identifier* e *dcterms:identifier* são usadas para definir o URI do recurso. Assim, quando um usuário cadastrar um mecanismo de segurança, estará cadastrando uma instância de *wot:Security*. O usuário tem, ainda, a opção de adicionar um nome (*?label*) para tal recurso por meio das propriedades *rdfs:label*, *schema:name* e *dcterms:tittle* e associar um comentário sobre o recurso em questão através da propriedade *rdfs:comment*.

## 5.2. Módulo Communication Protocol (Protocolo de Comunicação)

A IoT usa uma variedade de protocolos para acessar dispositivos, já que nenhum protocolo é apropriado em todos os contextos. Então torna-se desafiador para WoT permitir interações com a infinidade de diferentes plataformas de IoT.

A iteração de uma coisa, seja ela, uma propriedade, uma ação ou um evento, pode estar disponível em protocolos diferentes. Na *WoT Ontology*, a classe *wot:CommunicationProtocol* é o item referente a protocolo. A Figura 20 apresenta o modelo ontológico criado para descrição do protocolo.



**Figura 20:** Modelo para descrição de protocolo.

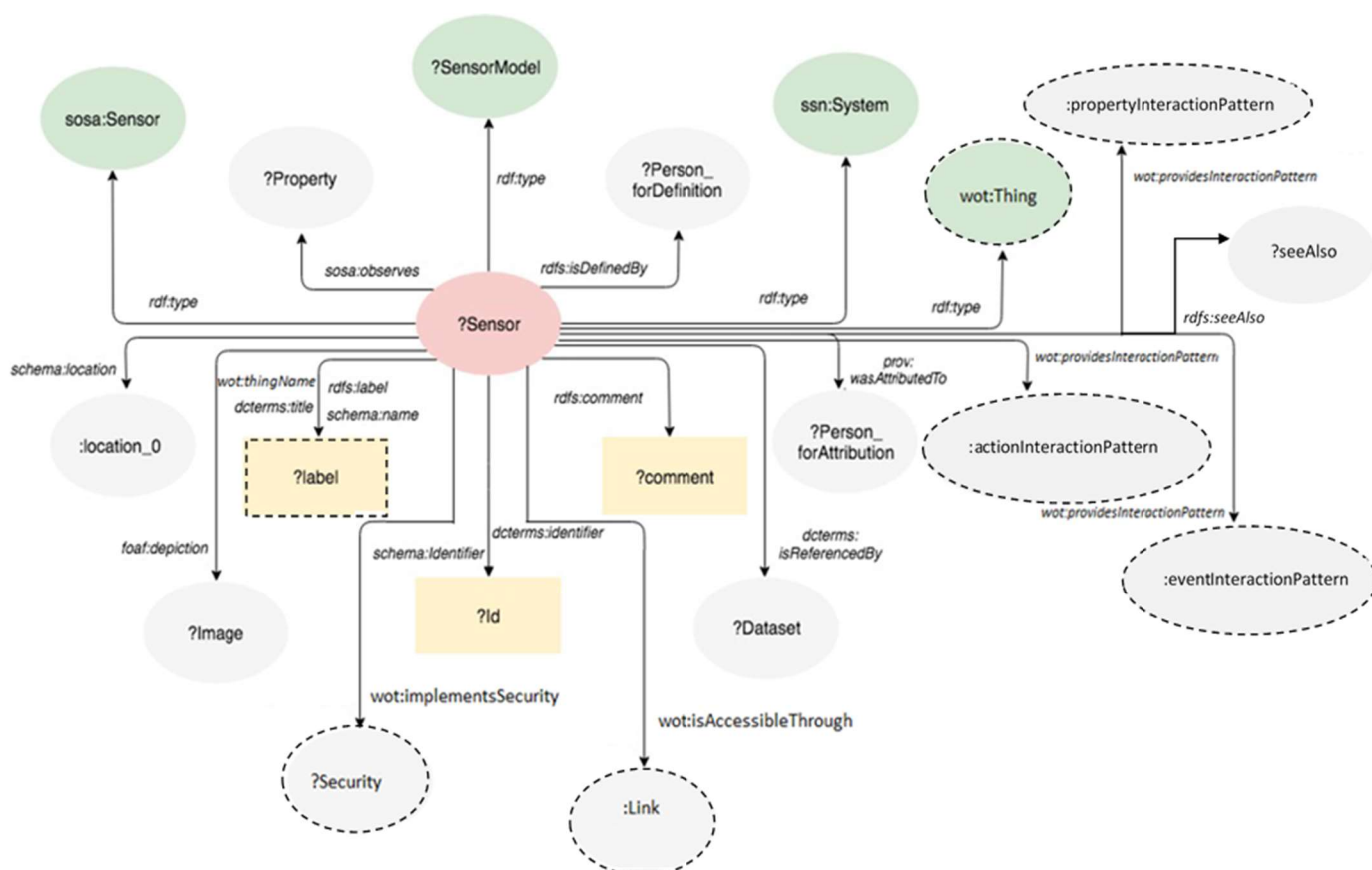
Fonte: Elaboração Própria

Na Figura 20, o recurso *CommunicationProtocol* (*?CommunicationProtocol*) é definido pela classe *wot:CommunicationProtocol*, por meio da propriedade *rdf:type*. Deste modo, ao cadastrar um novo protocolo, o usuário estará cadastrando instância da classe *wot:CommunicationProtocol*. As propriedades *schema:identifier* e *dcterms:identifier* são utilizadas para definir o URI do recurso. Ainda no modelo apresentado na Figura 20, protocolos podem também ser relacionados a outros recursos por meio das propriedades *owl:sameAs* e *rdfs:seeAlso*, realizando o *Linked Data Mashup*, seguindo o quarto princípio *Linked Data*. A tripla formada pela propriedade *owl:sameAs* informa que os recursos associados representam o mesmo protocolo, ou seja, são equivalentes. Já a propriedade *rdfs:seeAlso*, representa a existência de informação adicional sobre o recurso descrito (*?CommunicationProtocol*). Como em todo módulo criado neste trabalho, o usuário conta com a opção de adicionar um nome (*?label*) para o recurso (*?CommunicationProtocol*) por meio das propriedades *rdfs:label*, *schema:name* e *dcterms:title*, bem como associar um comentário através da propriedade *rdfs:comment*.

### 5.3. Módulo Sensor (Sensor)

O Sensor representa uma coisa inteligente do mundo físico. É conceituado como um dispositivo que recebe um estímulo, geralmente físico, químico ou biológico e que pode ser transformado em outra grandeza física para fins de medição e/ou monitoramento. (BALBINOT & BRUSAMARELLO, 2011). No trabalho de Rangel (2018), o modelo ontológico para

descrever os sensores foi criado para facilitar a identificação e integração dos sensores, descrevendo-os em termos de localização, fabricante, propriedade observada, modelo etc. A Figura 21 apresenta o modelo ontológico adaptado de Rangel (2018) para descrição da interface de acesso aos sensores. Ou seja, não somente a descrição técnica do dispositivo ou “coisa inteligente” é representada, mas também sua interface de acesso (TD).



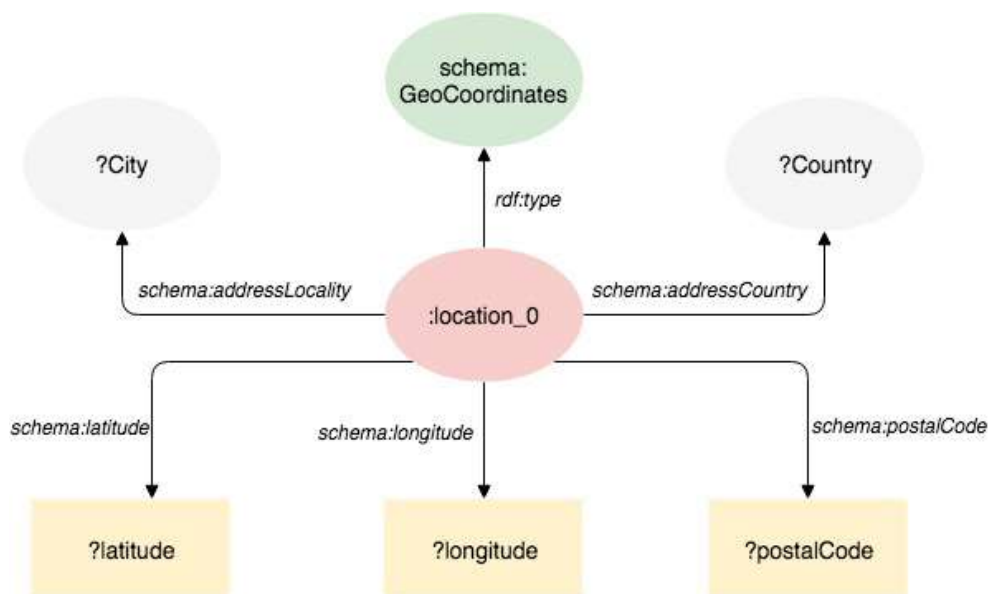
**Figura 21** - Modelo para descrição do sensor e de sua Interface de acesso.

Fonte: Adaptado de Rangel (2018)

Conforme modelo apresentado na Figura 21, sensores são definidos como recurso do tipo *?SensorModel* (modelo de sensor, classe OWL criada previamente pelo usuário na aplicação de Rangel (2018)). Como exemplo, tem-se a classe de modelo de sensor nomeada como DHT22 – Modelo de Sensor de temperatura). O módulo de sensor aqui neste trabalho representado, é uma instância do modelo de sensor *?SensorModel*. Note também que, ao cadastrar um novo sensor, o usuário cadastrará, uma instância das classes *sosa:Sensor*, *ssn:System*, e *wot:Thing*.

É possível observar ainda na Figura 21, que o recurso sensor, é representado pela variável *?Sensor* com o seu URI gerado por meio das propriedades *schema:identifier* e *dcterms:identifier*. A aplicação prevê também um nome para o sensor, representado pelas

propriedades *wot:thingName*, *rdfs:label*, *dcterms:title* e *schema:name* que são usadas para relacionar o sensor ao valor recebido na variável *?label*. Além disso, um comentário e uma imagem podem ser inseridos no cadastro do sensor por meio das propriedades *rdfs:comment* e *foaf:depiction*, respectivamente. O elemento *:location\_0* trata-se de nó em branco (recurso anônimo) contendo as propriedades de localização do sensor. Maiores informações, do modelo para descrição dados de localização de sensores na Figura 22.



**Figura 22** - Dados de localização de sensor.

Fonte: Rangel (2018)

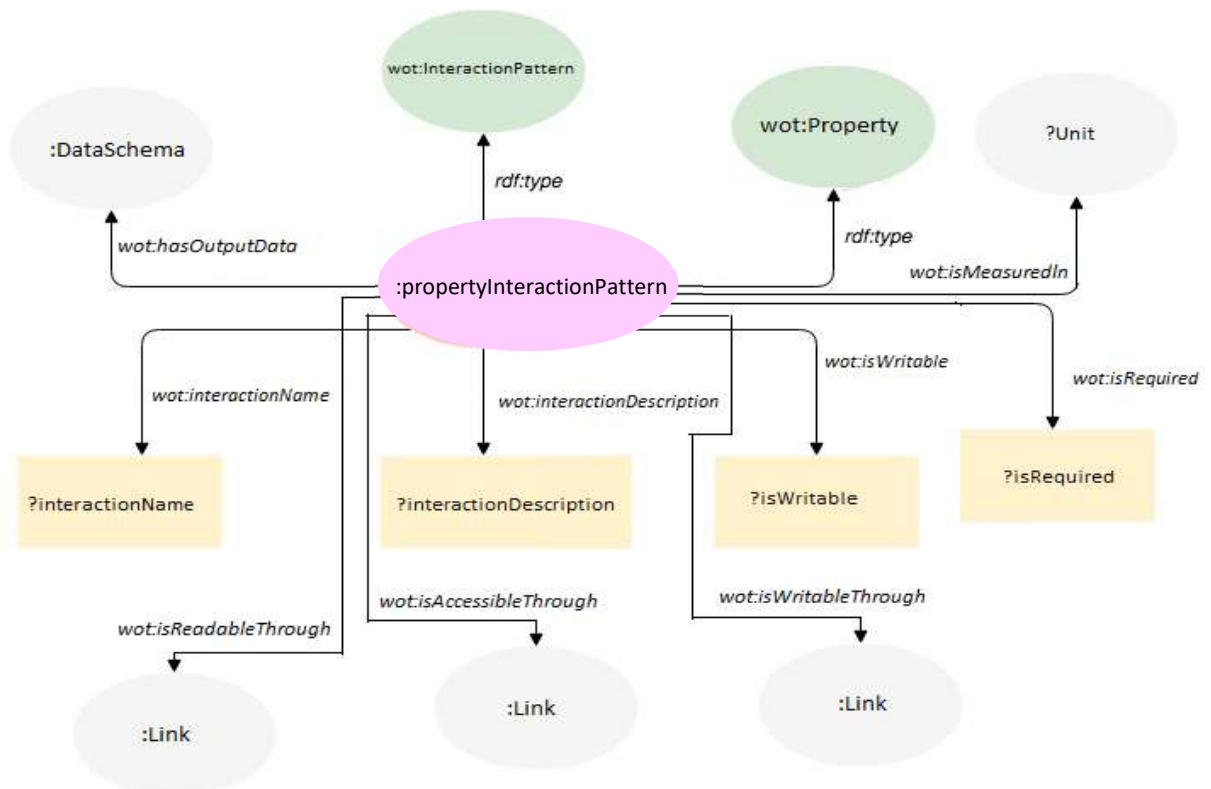
Conforme a Figura 22, variadas propriedades da ontologia Schema.org são usadas para descrever a localização de sensores. Informações sobre a latitude, longitude e código postal são referidas por meio das propriedades *schema:latitude*, *schema:longitude* e *schema:postalCode*, nessa ordem. Além disto, as propriedades *schema:addressLocality* e *schema:addressCountry* são utilizadas para associar sensores à cidade (região) e país onde encontram-se inseridos. Os valores dos objetos nas propriedades *schema:addressLocality* e *schema:addressCountry* são recursos, ou seja, elementos que possuem URI. Os recursos *?City* e *?Country* podem ter como origem bases de dados de terceiros - *mashup* (p. ex.: base de dados *DBpedia*).

Sensores observam certas propriedades (de objetos de interesse genéricos ou específicos), tais como: temperatura do ar, temperatura da sala 245, etc. Deste modo, é possível associar tais propriedades (*?Property* - instâncias de *ssn:Property*), ao cadastro da “coisa”. A relação de propriedades (*?Property*) com o sensor é declarada pela propriedade *sosa:observes*. Sensores podem, também, ser associados a pessoas – sejam estas responsáveis por tais

dispositivos, ou outras que apenas fazem o seu cadastro na aplicação. A relação de um sensor (?Sensor) com a pessoa responsável pelo seu registro (?Person\_forDefinition) é dada pela propriedade *rdfs:isDefinedBy*. No entanto, a propriedade *prov:wasAttributedTo* representa a relação de um sensor com a pessoa responsável pelo dispositivo (?Person\_forAttribution). O modelo proposto prevê ainda, o cadastro de bases de dados de sensores (?Dataset), sejam elas semânticas ou não (definidas como instâncias da classe *dcat:Dataset*), e sua ligação a sensores, cujos dados são utilizados - o relacionamento entre tais recursos é definido pela propriedade *dcterms:isReferencedBy*.

Na Figura 21, conforme relatado, é possível também visualizar a descrição da interface de acesso do sensor, que é justamente a contribuição deste trabalho. Para tal, o sensor (?Sensor), provê três tipos de interação, representados na Figura 21 pelos nós em branco (recursos anônimos) *:propertyInteractionPattern*, *:actionInteractionPattern* e *:eventInteractionPattern*. A relação do recurso (?Sensor) com cada uma interação acontece por meio da propriedade *wot:providesInteractionPattern*. Maiores detalhes das propriedades de cada padrão de interação são exibidos nas Figuras 23, 26-27.

### Padrão de Interação de Propriedade (:propertyInteractionPattern)



**Figura 23** - Modelo para descrição do Padrão de Interação - *Property*.

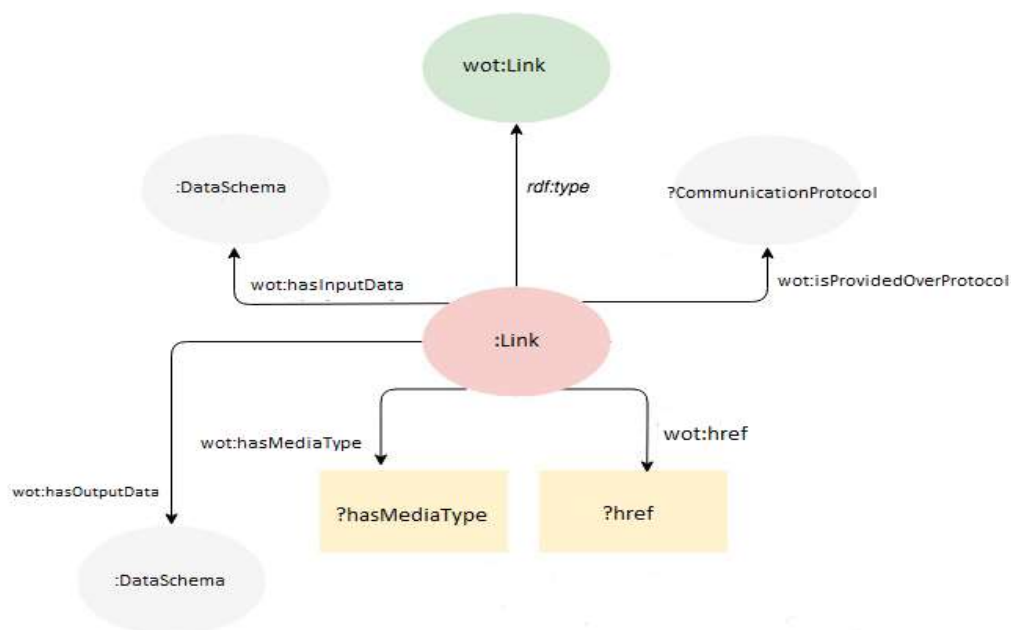
Fonte: Elaboração Própria



Na Figura 23, o recurso anônimo *:propertyInteractionPattern* representa o acesso a uma propriedade do dispositivo (p.ex.: a temperatura observada por sensor). O acesso a uma propriedade é definido no modelo como instância das classes *wot:InteractionPattern* e *wot:Property* por meio da propriedade *rdf:type*. Na WoT Ontology, a classe *wot:Property* é uma subclasse da Classe *wot:InteractionPattern*. Deste modo, quando o nó em branco é tipificado como *wot:Property* também, acaba sendo tipificado como *wot:InteractionPattern* e por ser uma classe filha, todas as propriedades pertencentes a classe *wot:InteractionPattern*, pertencem também a classe *wot:Property*. São elas: *wot:interactionName*, *wot:interactionDescription*, *wot:isAccessibleThrough*, *wot:isReadableThrough* e *wot:isWritableThrough*. Para as duas primeiras propriedades, tem-se como valor, um literal, representado pelas variáveis *?interactionName* e *?interactionDescription*. Assim, é possível atribuir um nome para interação, bem como um comentário respectivamente. Para as demais propriedades, tem-se como valor de propriedade, nós em branco. Por esses nós, é possível identificar os *endpoints* a partir dos quais um padrão de interação é acessível, lido ou escrito, identificadas na ontologia WoT como a classe *wot:Link* (Figura 24). Para o tipo de interação *Property*, é possível saber, ainda, se a propriedade em questão é obrigatória ou não (*?isRequired*) e se é modificável ou não (*?isWritable*). Ainda na Figura 23, o recurso *:propertyInteractionPattern* declara o tipo de dados de saída, por meio da propriedade *wot:hasOutputData* cujo valor é um nó em branco (*:DataSchema*), além de fornecer a unidade de medida (*?Unit*) utilizada na interação, através da propriedade *wot:isMeasuredIn*. Para a unidade de medida, tem-se um recurso (*?Unit*) instância de uma classe reusada, da ontologia *unit* do Catálogo QUDT<sup>42</sup>.

---

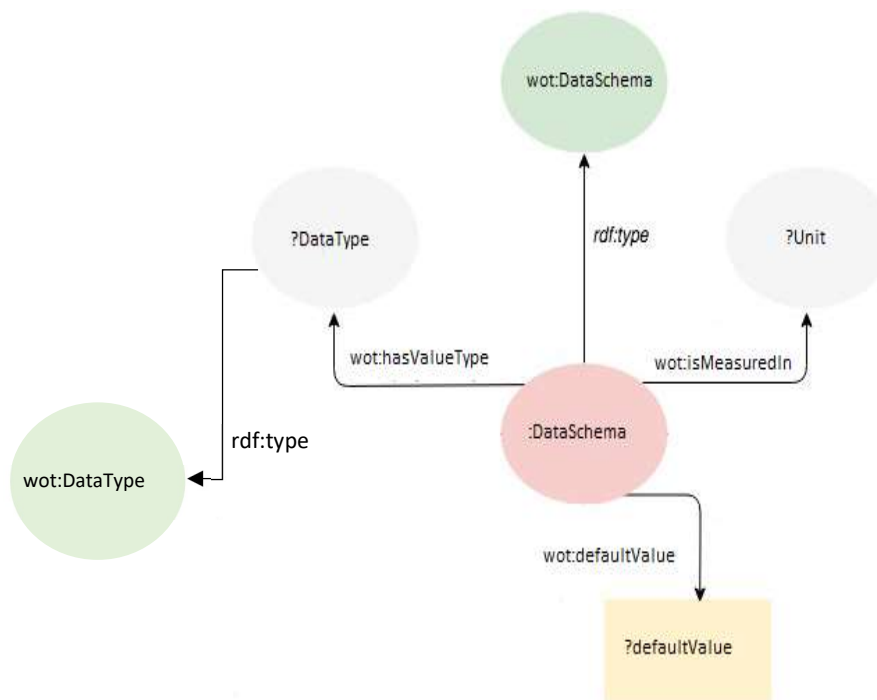
<sup>42</sup> <http://www.linkedmodel.org/catalog/qudt/1.1/index.html>



**Figura 24** - Modelo para descrição de *Link*.

Fonte: Elaboração Própria

A classe *wot:Link* representa um *link* para um endpoint (*ponto de acesso*), de acordo com um protocolo de comunicação (p.ex.: HTTP), a partir do qual podemos interagir com o dispositivo, quer seja para leitura de dados, quer seja para escrita de dados. Na Figura 24, tem-se o elemento *:Link* que consiste em nó em branco (recurso anônimo) tipificado como *wot:Link* pela propriedade *rdf:type*. Cada link tem no mínimo dois atributos: *URL* e *mediatype* (tipo de mídia ou arquivo), representados pelas propriedades *wot:href* e *wot:hasMediaType*, pela *WoT ontology*, as duas são classificadas como *Datatype Property*. Deste modo, as instâncias correspondentes *?href* e *?hasMediaType* são valores literais do tipo *string* das propriedades já mencionadas. A propriedade de objeto *wot:isProvidedOverProtocol* relaciona um *endpoint* com o protocolo de comunicação em que é servido, identificado como *?CommunicationProtocol*. Uma interação pode ter dados de entrada e/ou saída associados. Para isto tem-se as propriedades *wot:hasInputData* e *wot:hasOutputData* que permitem a conexão de um dado padrão de interação a um *:DataSchema* (esquema de dados) que, conforme descrito na Figura 25, consiste basicamente em um tipo de dados e uma unidade de medida.



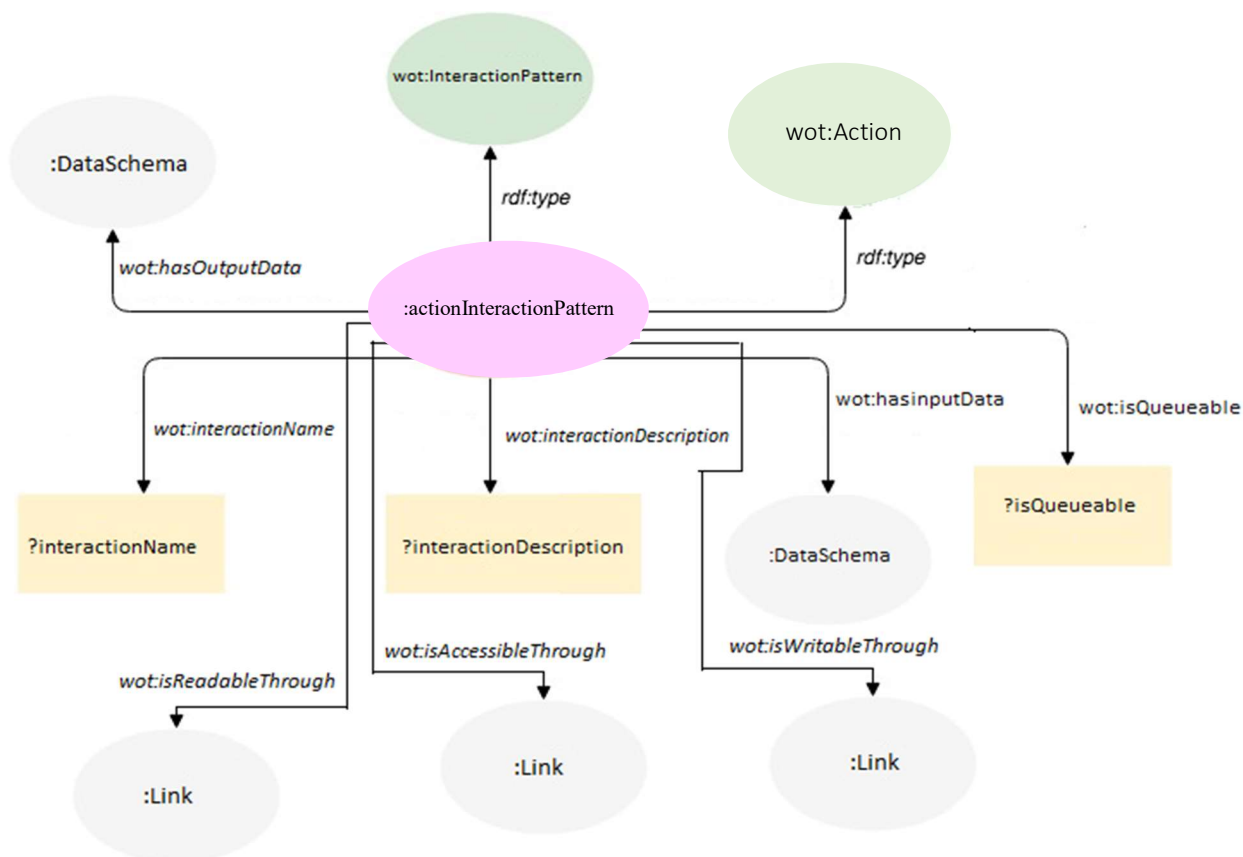
**Figura 25** - Modelo para descrição do *DataSchema*.

Fonte: Elaboração Própria

Um esquema de dados, representado pelo nó em branco *:DataSchema*, é uma instância da classe *wot:DataSchema*, relacionado pela propriedade *rdf:type*. Um esquema de dados é definido por um tipo de dados (*wot:hasValueType*), uma unidade de medida (*wot:isMeasuredIn*) e, opcionalmente, um valor default (*wot:defaultValue*). O valor é um literal (p.ex.: *false*) e o tipo de dados um recurso (p.ex.: *xsd:boolean*). O recurso *?DataType* é uma instância da classe *wot:DataType*.

Os modelos representando os tipos de interação (Propriedade, Ação e Evento) são semelhantes. A seguir, são apresentadas as Figuras 26 e 27 representando os padrões de interação Ação e Evento, respectivamente.

### Padrão de Interação de Ação (:actionInteractionPattern)



**Figura 26** - Modelo para descrição do Padrão de Interação – *Action*.

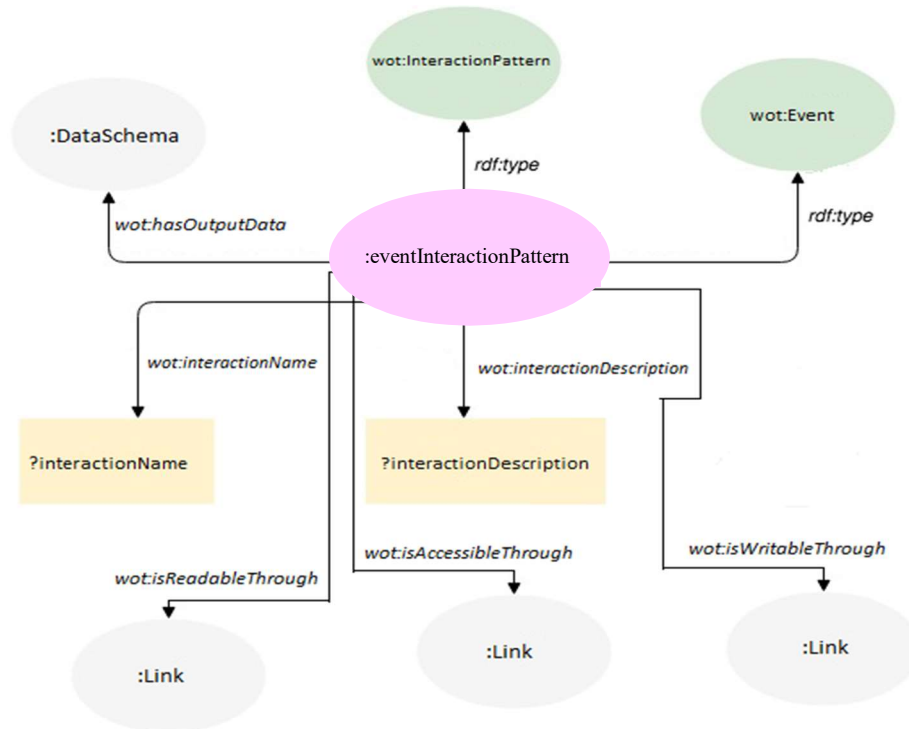
Fonte: Elaboração Própria

Na Figura 26, o recurso anônimo *:actionInteractionPattern* representa o disparo de uma ação no dispositivo (em geral, um atuador). Este tipo de acesso é definido no modelo como instância das classes *wot:InteractionPattern* e *wot:Action* por meio da propriedade *rdf:type*. Na *Wot Ontology*, a classe *wot:Action* é subclasse da classe *wot:InteractionPattern*. Deste modo, o nó em branco (*:actionInteractionPattern*) é tipificado como *wot:Action* e também, acaba sendo tipificado como *wot:InteractionPattern*. Além disso, por ser uma classe filha, todas as propriedades pertencentes à classe *wot:InteractionPattern*, pertencem também a classe *wot:Action*. São elas: *wot:interactionName*, *wot:interactionDescription*, *wot:isAccessibleThrough*, *wot:isReadableThrough* e *wot:isWritableThrough*. Os detalhes destas propriedades são os mesmos descritos para o padrão de interação de propriedade.

Conforme o W3C WoT, o padrão de interação definido conforme a classe *wot:Action*, visa alterações ou processos em um dispositivo que leva um certo tempo para serem concluídos (isto é, ações que não podem ser aplicadas instantaneamente, como por exemplo, alterações de propriedades). As ações podem conter até dois esquemas de dados (*:DataSchema*), um para

parâmetros de entrada opcionais, relacionado pela propriedade *wot:hasInputData*, um para resultados de saída, relacionado pela propriedade *wot:hasOutputData*. Como exemplo de utilização deste tipo de interação, tem-se o controle de um atuador. Ainda é possível provê enfileiramento de ações, por meio da propriedade *wot:isQueueable*, que possui como valor, um literal do tipo *Boolean*.

### Padrão de Interação de Evento (:eventInteractionPattern)



**Figura 27** - Modelo para descrição do Padrão de Interação – *Event*.

Fonte: Elaboração Própria

Na Figura 27, o recurso anônimo *:eventInteractionPattern* representa a captura de um evento gerado pelo dispositivo. Este recurso é uma instância das classes *wot:InteractionPattern* e *wot:Event* por meio da propriedade *rdf:type*. Conforme a *Wot Ontology*, a classe *wot:Event* é subclasse da classe *wot:InteractionPattern*. Assim, quando o nó em branco é tipificado como *wot:Event*, também acaba sendo tipificado como *wot:InteractionPattern* e, por ser uma classe filha, todas as propriedades pertencentes a classe *wot:InteractionPattern*, pertencem também a classe *wot:Event*. São elas: *wot:interactionName*, *wot:interactionDescription*, *wot:isAccessibleThrough*, *wot:isReadableThrough* e *wot:isWritableThrough*. Os detalhes destas propriedades são os mesmos descritos para o padrão de interação de propriedade.

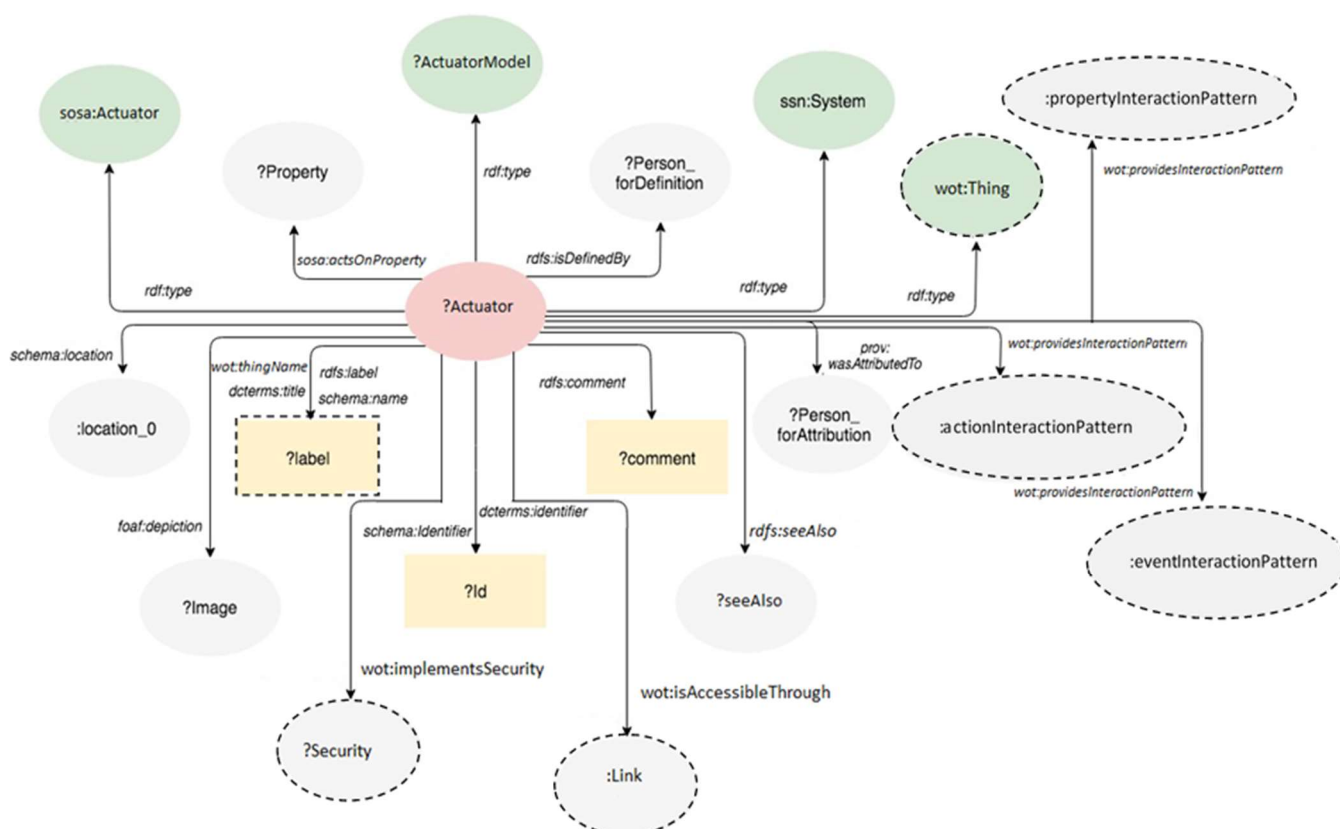
Para W3C WoT, um evento permite que um mecanismo seja notificado por um dispositivo em uma determinada condição. Ele descreve interações de *push* assíncronas

(interações descritas sem a necessidade de espera pelas demais) iniciadas pelo dispositivo. Nesta interação, não o estado, mas as transições de estado (ou seja, eventos) são comunicadas. Tais eventos são disparados mesmo não sendo solicitados. A interação ainda conta com a declaração do tipo de dados de saída, realizada por meio da propriedade *wot:hasOutputData* cujo valor é um nó em branco (*:DataSchema*). Um alarme ou amostras de uma série de tempo que são enviadas regularmente são exemplos deste tipo de interação.

#### **5.4. Módulo Actuator (Atuador)**

Atuador é um dispositivo que produz movimento, atendendo a comandos que podem ser manuais, elétricos ou mecânicos. Como exemplo, pode-se citar atuadores de movimento induzido por cilindros pneumáticos (pneumática), cilindros hidráulicos (hidráulica), motores hidráulicos e motores pneumáticos (dispositivos rotativos com acionamento de diversas naturezas). Estes mecanismos transformam, em geral, a energia de entrada (diversas naturezas) em movimentos que se pode considerar energia cinética. Pela visão da ontologia WoT, baseado na W3C WoT, tanto os atuadores quanto os sensores são “coisas” inteligentes. Segundo Rangel (2018), no contexto da ontologia SSN os atuadores são dispositivos capazes de mudar o estado de um ambiente.

O modelo proposto para descrição da interface de acesso de atuadores é similar ao utilizado para descrever a interface de acesso dos sensores. Devido às semelhanças dos modelos, apenas os elementos que não estão inclusos no módulo “Módulo Sensor” serão apresentados a seguir. Basicamente, são os elementos previamente descritos em Rangel (2018). Porém, a Figura 28, apresenta o modelo de atuadores já com a descrição da interface de acesso definida.



**Figura 28** - Modelo para descrição do atuador e de sua Interface de acesso.

Fonte: Adaptado de Rangel (2018)

Atuadores, conforme Figura 28, são definidos pelas classes *wot:Thing*, *ssn:System*, *sosa:Actuator* e pela classe *?ActuatorModel* (módulo modelo de atuador, criado previamente pelo usuário na aplicação de Rangel (2018)). O URI de um dado atuador é gerado tendo como base o valor da propriedade *schema:identifier* e *dcterms:identifier*. Além disso, propriedades (de atuação) podem ser vinculadas ao cadastrado de atuadores através da propriedade *sosa:actsOnProperty*. Ao associar um valor à propriedade *sosa:actsOnProperty*, tal valor é definido como *sosa:ActuableProperty* por inferência. De forma análoga aos sensores, atuadores podem ser associados a propriedades (de atuação) de objeto de interesse genéricas ou específicas. A aplicação prevê também, um (nome), para o atuador, representado pelas propriedades *wot:thingName*, *rdfs:label*, *dcterms:title* e *schema:name* que são usadas para relacionar o atuador ao valor recebido na variável *?label*. A descrição da interface de acesso ao atuador, é análoga à do sensor, descrita anteriormente, bastando acompanhar todo o processo detalhado nas Figuras 21-27.

## 6. APLICAÇÃO SEMÂNTICA *LINKED DATA*

Realizadas as fases de eliciação de requisitos e da modelagem ontológica correspondente, a aplicação semântica *Linked Data* para descrição da interface e uso de sensores e atuadores foi desenvolvida no framework *Callimachus*. A aplicação aqui desenvolvida é uma nova versão da aplicação de Rangel (2018), acrescentando as novas funcionalidades pertinentes. Esta seção apresenta esta nova versão da aplicação, por meio do cadastro de um sensor e um atuador comumente utilizados em projetos de IoT.

A Figura 29, por meio de um diagrama de casos de uso UML, traz as funcionalidades da aplicação de Rangel (2018), juntamente com as funcionalidades acrescentadas por este trabalho, destacadas na cor cinza. Assim, é possível, uma visualização geral de todos os recursos que podem ser cadastrados na aplicação até o presente trabalho. Em seguida, na Figura é possível visualizar as telas correspondentes a cada funcionalidade.



**Figura 29** – Funcionalidades atualizadas da Aplicação.

Fonte: Adaptado de Rangel (2018)

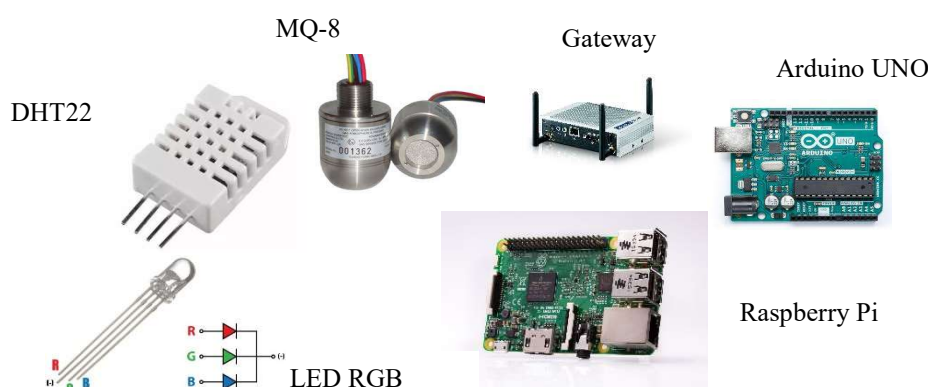


Seguindo a mesma abordagem de Rangel (2018), para facilitar a compreensão, o grafo RDF gerado como resultado de cada cadastro de recurso será apresentado graficamente (instanciação do correspondente *graph pattern* descrito anteriormente). Nos Apêndices A-D deste trabalho é possível conferir as versões em sintaxe *Turtle* dos grafos RDF subjacentes aos cadastros.

O código fonte da aplicação pode ser encontrado em <[https://github.com/App\\_Semantic/WoT\\_Description](https://github.com/App_Semantic/WoT_Description)>.

### 6.1. Alguns dispositivos comumente usados em configurações IoT

Na Figura 30, é possível observar alguns dispositivos (coisas inteligentes) comumente usados em projeto de IoT.



**Figura 30** – Elementos que compõem o cenário IoT.  
Fonte: Elaboração Própria.

Para os elementos que compõem a Figura 30, tem-se a Tabela 7.

**Tabela 7** – Dispositivos que compõem o cenário IoT.

Elemento	Descrição
DHT22	Sensor que observa temperatura e umidade
MQ-8	Sensor que observa gás
LED RGB	Atuador Led capaz de transformar eletricidade em energia luminosa
Raspberry Pi 3	Plataforma que hospeda os sensores e atuadores
Arduino UNO	Plataforma que hospeda os sensores e atuadores
Gateway	Usado para conectar vários nós de IoT e dispositivos que executam protocolos diferentes

Fonte: Elaboração Própria

Como forma de demonstração do uso da aplicação e as telas desenvolvidas para o cadastro de cada um dos requisitos funcionais, serão apresentados os cadastros do sensor DHT22 e do atuador LED RGB, dentre os dispositivos físicos apresentados na Tabela 7 (Figura 30). Também serão apresentados os cadastros dos recursos associados (Segurança, Protocolo

de comunicação). Como pode ser visto na Tabela 8, para facilitar a demonstração dos grafos RDF resultantes dos exemplos realistas, foram criados prefixos para simplificar os URIs dos recursos inseridos na aplicação e de terceiros (p. ex.: recursos reusados da fonte de dados DBpedia).

**Tabela 8**– Definição dos Prefixos.

Prefixo	URI
sec	<i>http://localhost:8080/Resource/Security/</i>
comm	<i>http://localhost:8080/Resource/CommunicationProtocol/</i>
act	<i>http://localhost:8080/Resource/Actuator/</i>
unit	<i>http://qudt.org/vocab/unit/#</i>
pattern	<i>http://localhost:8080/Resource/Sensor/</i>
db	<i>http://dbpedia.org/resource/</i>

Fonte: Elaboração Própria

### 6.1.1. Cadastro de Segurança (Security)

A Figura 31 apresenta o cadastro de Segurança.

The screenshot shows a web form for creating security information. The form has the following fields and values:

- ID (URI):** JWT
- Label:** JSON Web Token
- Comment:** são credenciais, que podem conceder acesso a recursos.

A green 'Create' button is located at the bottom of the form.

**Figura 31** - Exemplo de criação de cadastro de Segurança.

Fonte: Elaboração própria

Com base na Figura 31, tem-se o valor “JWT” inserido no campo “ID (URI)”. Neste campo acontece a construção do URI do recurso. Na tela de criação do novo recurso, além do identificador único (URI), o usuário pode definir, ainda, um nome e um comentário para o recurso conforme os campos “Label” e “Comment”, respectivamente. Além disso, para cada

tela de cadastro e edição, uma breve descrição acerca do recurso é exibida. Assim, o usuário antes de cadastrar e/ou editar, fica informado acerca do recurso.

Na tela de edição, podem ser modificadas apenas as propriedades “Label” e “Comment”, com exceção do valor inserido no campo “ID (URI)”. A tela de edição do recurso cadastrado é conferida na Figura 32.

O cadastro do recurso realizado conforme as Figuras 31 e 32, gera um grafo RDF subjacente que fica armazenado na base de dados da aplicação. Este grafo é apresentado graficamente na Figura 33.

## JSON Web Token

---

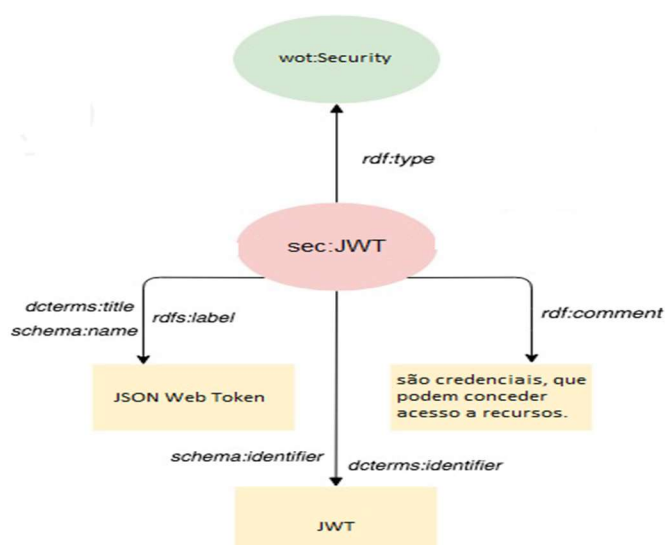
### Security Informations

The (optional) security field can be used to provide access metadata (self-contained) information of the Thing for securely transmitting information via all its resources

**Label**

**Comment**

**Figura 32** - Exemplo de edição do cadastro de Segurança.  
Fonte: Elaboração própria



**Figura 33** - Grafo RDF subjacente ao cadastro de Segurança.  
Fonte: Elaboração própria

Na Figura 33 é possível visualizar os valores de cada uma das propriedades, utilizadas para descrever segurança, relacionados durante o processo de cadastro. Na tela de edição é possível modificar ou alterar os valores disponibilizados no referido grafo, com exceção ao das propriedades *schema:identifier* e *dcterms:identifier*.

### 6.1.2. Cadastro de Protocolo de Comunicação (Communication Protocol)

A Figura 34 exemplifica o cadastro de uma novo Protocolo de Comunicação.

The screenshot shows a web form titled "Communication Protocol". Underneath, there's a section "Communication Protocol Information" with a brief definition: "A protocol is an agreement between the communicating parties on how communication is to proceed." The form has three main input fields: "ID (URI)" containing "SOAP", "Label" containing "Simple Object Access Protocol", and "Comment" containing "SOAP (Simple Object Access Protocol, em português Protocolo Simples de Acesso a". A green "Create" button is positioned at the bottom of the form.

**Figura 34** - Exemplo de criação do cadastro de Protocolo de Comunicação.

Fonte: Elaboração própria

Com base na Figura 34, tem-se o valor “SOAP” inserido no campo “ID (URI)”. Neste campo acontece a construção do URI do recurso. Na tela de criação do novo recurso, além do identificador único (URI), o usuário pode definir, ainda, um nome e um comentário para o recurso conforme os campos “Label” e “Comment” respectivamente. Além disso, para cada tela de cadastro e edição, uma breve descrição acerca do recurso é exibida. Assim, o usuário antes de cadastrar e/ou editar, fica informado acerca do recurso.

Na tela de edição (Figura 35), podem ser modificadas as propriedades “Label” e “Comment”, com exceção do valor inserido no campo “ID (URI)”. Além disso, é possível visualizar o campo “Same as at Dbpedia.org”, em que um protocolo cadastrado pode ser relacionado a um recurso da base de dados DBpedia (por meio da propriedade *owl:sameAs*). Deste modo, o recurso cadastrado e o recurso associado são equivalentes, em bases de dados distintas, seguindo o quarto princípio *Linked Data*. No campo “See Also at Dbpedia.org” é possível obter maiores informações sobre o recurso. Para realizar a associação com recurso da

DBpedia, usa-se a consulta SPARQL da Figura 36, na qual a palavra-chave digitada no campo é comparada com a propriedade *rdfs:label* do recurso da DBpedia.

A tela de edição do recurso cadastrado é conferida na Figura 35, logo a seguir.

**Communication Protocol Information**

A protocol is an agreement between the communicating parties on how communication is to proceed.

**Label**

Simple Object Access Protocol

**Comment**

SOAP (Simple Object Access Protocol, em português Protocolo Simples de Acesso a Objeto) é um protocolo baseado...

**See Also at DBpedia.org:**

SOAP

**Same as at DBpedia.org:**

SOAP

Save Cancel Delete

**Figura 35** - Exemplo de edição do cadastro de Protocolo de Comunicação.  
Fonte: Elaboração própria

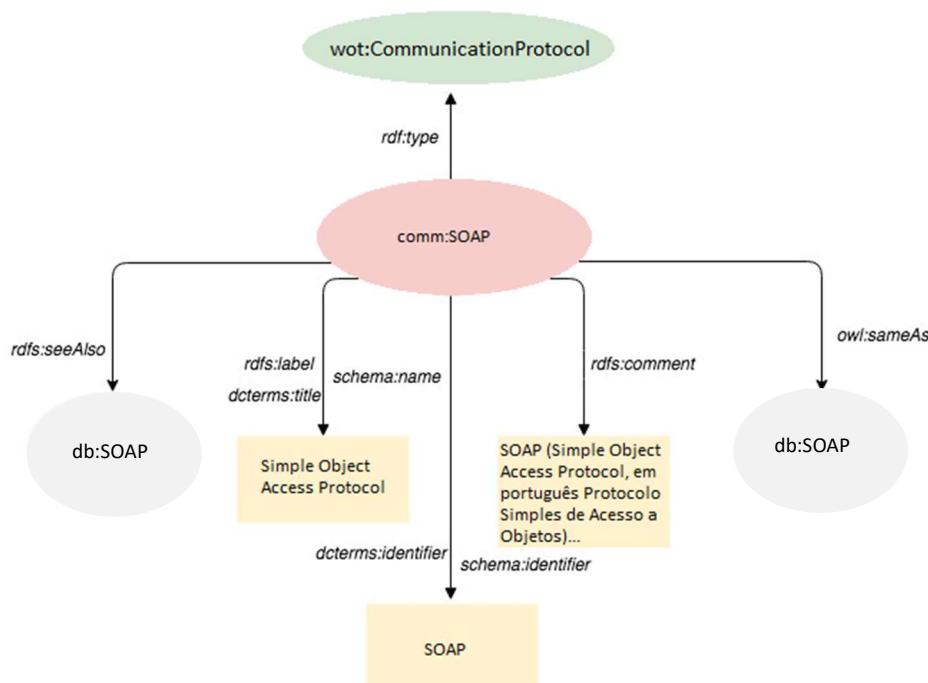
```

1 #
2 # @Cache-Control:
3 # @infer true
4 # @view
5 #
6 #
7 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
8 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
9 PREFIX dbo: <http://dbpedia.org/ontology/>
10 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
11
12 SELECT ?resource ?label
13
14 {
15   ?resource rdfs:label ?label .
16
17   FILTER (regex(str(?resource), "dbpedia.org", "i") &&
18           (lang(?label) = '' || lang(?label) = 'pt') &&
19           regex(str(?label), "$keyword", "i")) .
20
21 }
22 ORDER BY ?label

```

**Figura 36** - Consulta SPARQL para DBpedia.  
Fonte: Elaboração própria

O cadastro do recurso, cujo nome é “SOAP” apresentada nas Figuras 34 e 35 gera um grafo RDF subjacente que fica armazenado na base de dados da aplicação. O grafo é visualizado graficamente na Figura 37.



**Figura 37** - Grafo RDF subjacente ao cadastro de Protocolo de Comunicação.  
Fonte: Elaboração própria

### 6.1.3. Cadastro de Sensor (Sensor)

Segundo Rangel (2018), o modelo ontológico para descrição de modelos (tipos) de sensores prevê a associação das características técnicas dos dispositivos. Assim, o cadastro dos modelos demanda diferentes recursos, tais como procedimentos, propriedades de observação, capacidade de sistema, faixa de operação etc. Reiterando, cada modelo de sensor cadastrado estende a ontologia SSN, gerando uma classe OWL cujas instâncias serão os exemplares deste modelo a serem cadastrados posteriormente.

Partindo do modelo de sensor DHT22 (Figura 38) previamente cadastrado no catálogo semântico de Rangel (2018), realizou-se o cadastro de sensor nesta nova versão da aplicação.

## New Sensor Model

**ID (URI)**

**Label**

**Comment**

**Figura 38** - Tela de criação do modelo de sensor DHT22.  
Fonte: Rangel (2018)

O cadastro de sensores requer diversos outros recursos (pessoas, organizações, bases de dados, propriedades de observação, etc.). Por este trabalho se tratar da descrição da interface de acesso de um dispositivo, seja um sensor ou atuador, toda construção da interface de acesso foi desenvolvida na mesma tela de edição de sensores e atuadores de Rangel (2018). A Figura 39 apresenta a inserção de um novo exemplar de sensor. Vale ressaltar que esta tela de cadastro permanece como a mesma apresentada em Rangel (2018).

## New Sensor

**ID (URI)**

**Label**

**Comment**

**Figura 39**- Tela de criação de exemplar de sensor DHT22.  
Fonte: Rangel (2018)

Na Figura 39, o valor “DHT22\_Sensor” inserido no campo ID (URI) compõe o URI do exemplar cadastrado. Na tela de criação do novo recurso, além do URI, o usuário pode definir, ainda, um nome e um comentário para o recurso conforme os campos “Label” e “Comment”

respectivamente. Vale ressaltar que ao salvar o novo exemplar por esta tela, é gerado RDF já com as seguintes informações referentes à interface de acesso ao sensor: nome (*wot:thingName*), pelo valor recebido no campo Label, e a tipificação do exemplar como sendo uma instância da classe *wot:Thing*. A tela de edição será mostrada em partes, para melhor entendimento.

### Parte I: Edição das características técnicas do Sensor

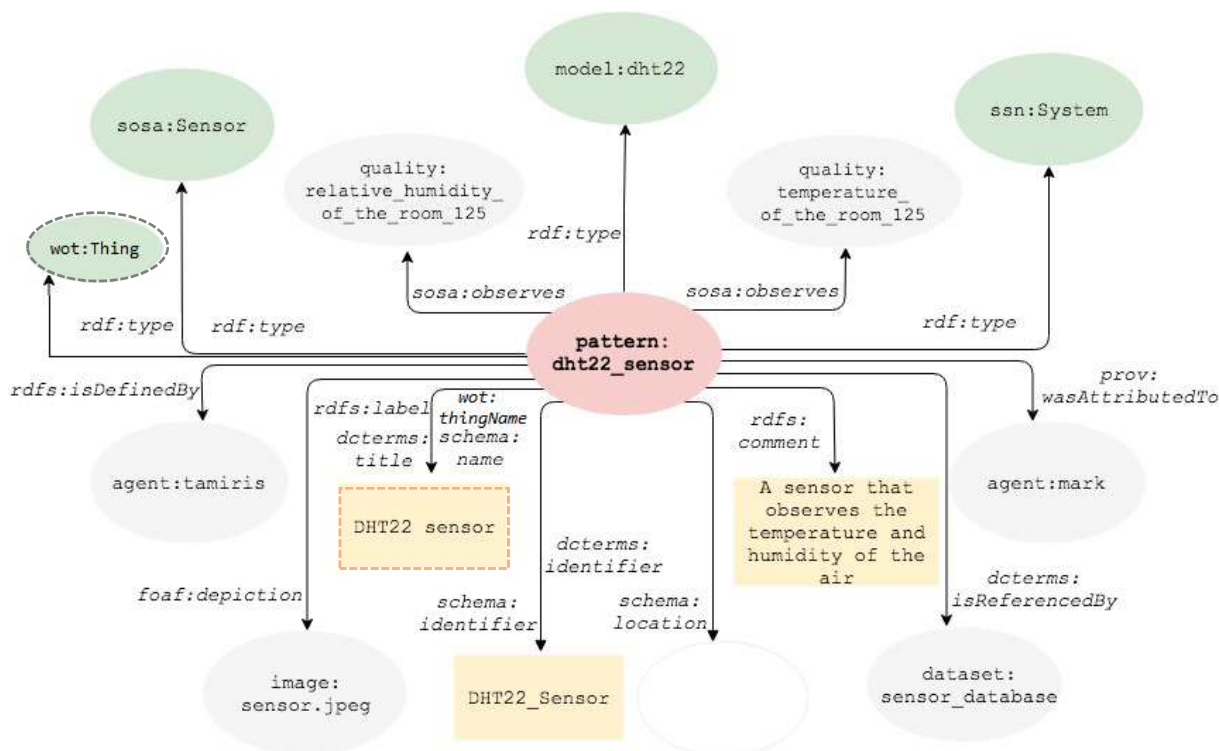
Esta primeira parte contempla a edição das características técnicas do sensor, a saber: modelo do exemplar cadastrado, propriedades de observação, dados de observação, informações sobre localização física, responsável pelo sensor, imagem do sensor, etc. Deste modo, a tela de edição - parte I de um exemplar de sensor, neste caso o “DHT22\_Sensor”, permanece conforme Rangel (2018). Na Figura 40 é possível visualizar a edição do referido exemplar. O grafo RDF subjacente a este exemplar é mostrado na Figura 41.

## DHT22 Sensor

<p><b>Label:</b> DHT22 Sensor</p> <p><b>Model:</b> DTH22</p> <p><b>Observes (Property):</b> Temperature of the room 125 Relative Humidity of the room 125</p> <p><b>Dataset:</b> Sensor_Database</p> <p><b>Attributed to:</b> Mark Zuckerberg</p> <p><b>Registered by:</b> Tamiris Sousa</p> <p><b>Comment</b> A sensor that observes the temperature and humidity of the air.</p>	<p><b>Postal code:</b> 28030490</p> <p><b>Latitude:</b> -21.762472</p> <p><b>Longitude:</b> -41.333265</p> <p><b>City:</b> Campos dos Goytacazes</p> <p><b>Country:</b> Brasil</p> <p><b>Select an image of the system</b> </p> <p>Temperature and humidity sensor      LCD</p> <p>PIR sensor      The PI      The camera</p>
--	---

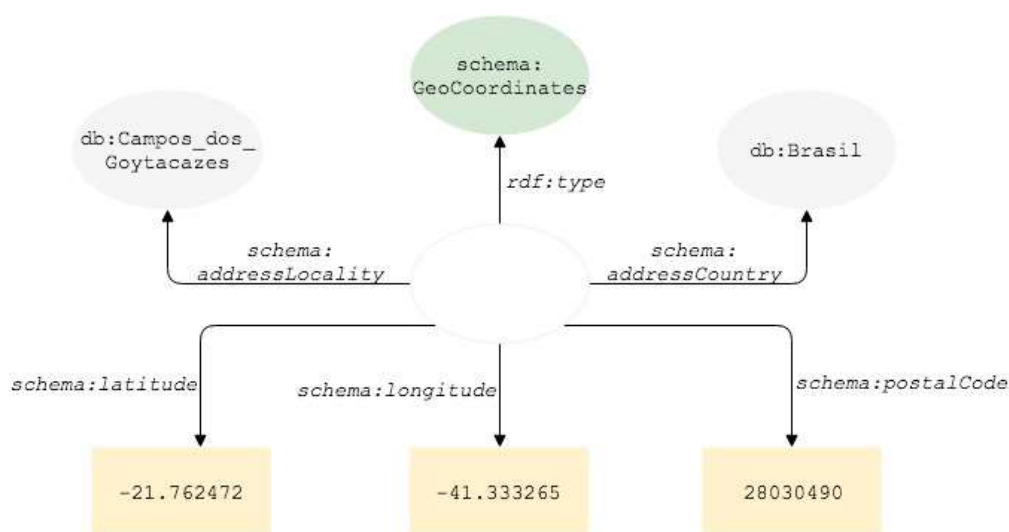
**Figura 40-** Tela de edição Parte I do exemplar DHT22\_Sensor.  
Fonte: Rangel (2018)





**Figura 41-** Grafo RDF subjacente a tela de edição ParteI do exemplar DHT22\_Sensor.  
Fonte: Adaptado de Rangel (2018)

Na Figura 41 é possível verificar as propriedades das ontologias utilizadas para descrever os exemplares de sensor. É possível evidenciar, os diferentes recursos usados para contextualizar o sensor cadastrado. Como exemplo, a propriedade de observação: “quality:relative\_humidity\_of\_the\_room\_125”. Os valores referentes a localização do exemplar cadastrado estão na Figura 42.



**Figura 42-**Propriedades de localização de exemplar do modelo DHT22.  
Fonte: Rangel (2018)

## Parte II: Edição da Interface de acesso do Sensor

A segunda parte da tela de edição de exemplar de Sensor consiste na contribuição central deste trabalho. Nela, é possível, visualizar todos os elementos que compõem a descrição da interface de acesso do exemplar de sensor. Na Figura 43, é possível visualizar a tela ainda sem a inserção de valores.

### DESCRIPTION OF ACCESS INTERFACE OF SENSORS

**Exemplar Data**

ImplementsSecurity:

URI\_Base

Base URI

The Think is Accessible Through Link

Define Link +

**Interaction Pattern**

An interaction in the context of the Web of Things is an exchange of data between a Web client and a Thing. This data can be either given as input by the client, returned as output by the Thing or both. Three interaction patterns have been defined so far. Property, Action and Event.

**Information about the Property Interaction Pattern**

Property provides readable and/or writeable data that can be static (e.g., supported mode, rated output voltage, etc.) or dynamic (e.g., current fill level of water, minimum recorded temperature, etc.).

Property +

**Information about the Action Interaction Pattern**

The Action interaction pattern targets changes or processes on a Thing that take a certain time to complete (i.e., actions cannot be applied instantaneously like property writes). Examples include an LED fade in, moving a robot, brewing a cup of coffee, etc. Usually, ongoing Actions are modelled as Task resources, which are created when an Action invocation is received by the Thing.

Action +

**Information about the Event Interaction Pattern**

The Event interaction pattern enables a mechanism to be notified by a Thing on a certain condition.

Event +

Save Cancel Delete

**Figura 43** – Tela simplificada de edição Parte II do sensor.

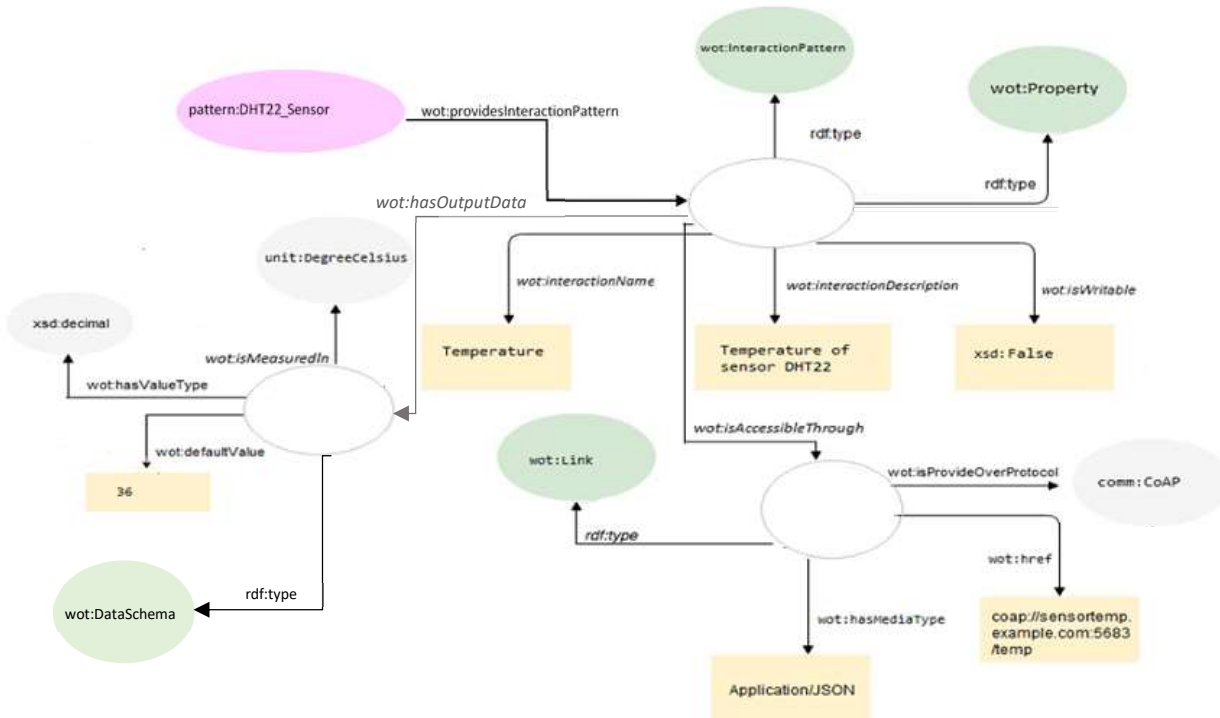
Fonte: Elaboração Própria

A descrição da interface de acesso requer alguns metadados semânticos sobre o sensor, padrão ou modelo de interação (Propriedade, Ação e Evento), esquema semântico para tornar os modelos de dados compreensíveis à máquina e recursos para *Web Linking* a fim de expressar as relações entre dispositivos (Figura 17). Não necessariamente todos os campos serão preenchidos quando se realiza o cadastro (vide cardinalidade Figura 19). Como exemplo, um sensor pode estar associado a um tipo de padrão de interação ou até mesmo a vários. A tela da Figura 44 mostra os elementos utilizados para efetuar a descrição da interface de acesso do exemplar “DHT22\_Sensor”. O padrão de interação mostrado é *wot:Property*, ou seja, ao escolher na tela, o padrão de interação “Property” é gerado um recurso anônimo (nó em branco) definido no modelo como instância das classes *wot:InteractionPattern* e *wot:Property* por meio da propriedade *rdf:type*. Para melhor compreensão e apresentação, são visualizadas as telas particionadas.

**Figura 44** – Descrição do Padrão de Interação *Property*.

Fonte: Elaboração Própria

O padrão de interação "Property", conforme a Figura 44, recebe um nome pelo campo "Interaction Name" e uma breve descrição da interação é fornecida conforme o campo "Interaction Description". Deste modo, é possível saber que o DHT22\_Sensor fornece dados de temperatura por meio deste Padrão de Interação. Trata-se de uma propriedade não editável, conforme atribuição do valor "False" para o campo "Is Writable:" Note que o Padrão de Interação está acessível por meio de um Link, cujo endereço está definido como: "coap://sensortemp.example.com:5683/temp" e o tipo de mídia gerado é um "Application/JSON", emitindo assim, uma representação JSON dos recursos. Além disso, a interação é provida pelo protocolo "COAP", um recurso previamente cadastrado na aplicação pelo usuário. O DataType (Tipo de Dado) é definido na Tela de "Output Data" onde é permitido a declaração do tipo de dado e a unidade de medida fornecida pela Interação Property. Assim, para o "Default Value", foi inserido o valor "36" e declarado sendo um "xsd:Decimal" (tipo primitivo, predefinido pelo XML Schema) e a unidade de medida definida foi "Degree Celsius". O grafo RDF gerado a partir dos dados apresentados na tela de Edição Parte II referentes a descrição do DHT22\_Sensor é apresentado na Figura 45.



**Figura 45** – Parte do grafo RDF subjacente a tela de edição Parte II do exemplar DHT22\_Sensor com ênfase no padrão de interação "Property".

Fonte: Elaboração Própria

#### 6.1.4. Cadastro de Atuador (Actuator)

As telas de cadastro e edição para o modelo proposto para descrição da interface de acesso de atuadores é muito similar ao utilizado para descrever a interface de acesso dos sensores. Mesmo sendo muito similar, optou-se por mostrar o cadastro de um atuador por razões de completude.

Partindo do modelo de atuador LED\_Rgb (Figura 46) previamente cadastrado no catálogo semântico de Rangel (2018), realizou-se o cadastro do atuador. Deste modo, cadastrando um exemplar de atuador na aplicação, temos uma instância da classe OWL correspondente ao modelo de atuador previamente cadastrado.

## New Actuator Model

**ID (URI)**

**Label**

**Comment**

**Create**

**Figura 46** - Tela de criação do modelo de atuador LED\_Rgb.  
Fonte: Elaboração Própria

O cadastro de atuadores requer diversos outros recursos (pessoas, organizações, bases de dados, propriedades de observação etc.). Conforme já destacado no cadastro de sensores, por este trabalho se tratar da descrição da interface de acesso, seja de um sensor ou atuador, toda construção da interface de acesso foi desenvolvida na mesma tela de edição de sensores e atuadores de Rangel (2018). A Figura 47 apresenta a inserção de um novo exemplar de atuador. Vale ressaltar que, o *layout* da tela de cadastro permanece como a mesma apresentada em Rangel (2018).

## New Actuator

**ID (URI)**

**Label**

**Comment**

**Create**

**Figura 47** - Tela de criação do atuador LED\_Rgb\_Actuator.  
Fonte: Elaboração Própria

Na Figura 47, o valor “LED\_Rgb\_Actuator” inserido no campo ID (URI) compõe o URI do exemplar cadastrado. Na tela de criação do novo recurso, além do identificador único (URI), o usuário pode definir, ainda, um nome e um comentário para o recurso conforme os campos “Label” e “Comment” respectivamente. Vale ressaltar que ao salvar o novo exemplar

por esta tela, é gerado RDF já com as seguintes informações referentes à interface de acesso: nome (*wot:thingName*), pelo valor recebido no campo Label, e a tipificação do exemplar como sendo uma instância de *wot:Thing*. A tela de edição, será mostrada em partes, para melhor compreensão.

### Parte I: Edição das características técnicas do Atuador

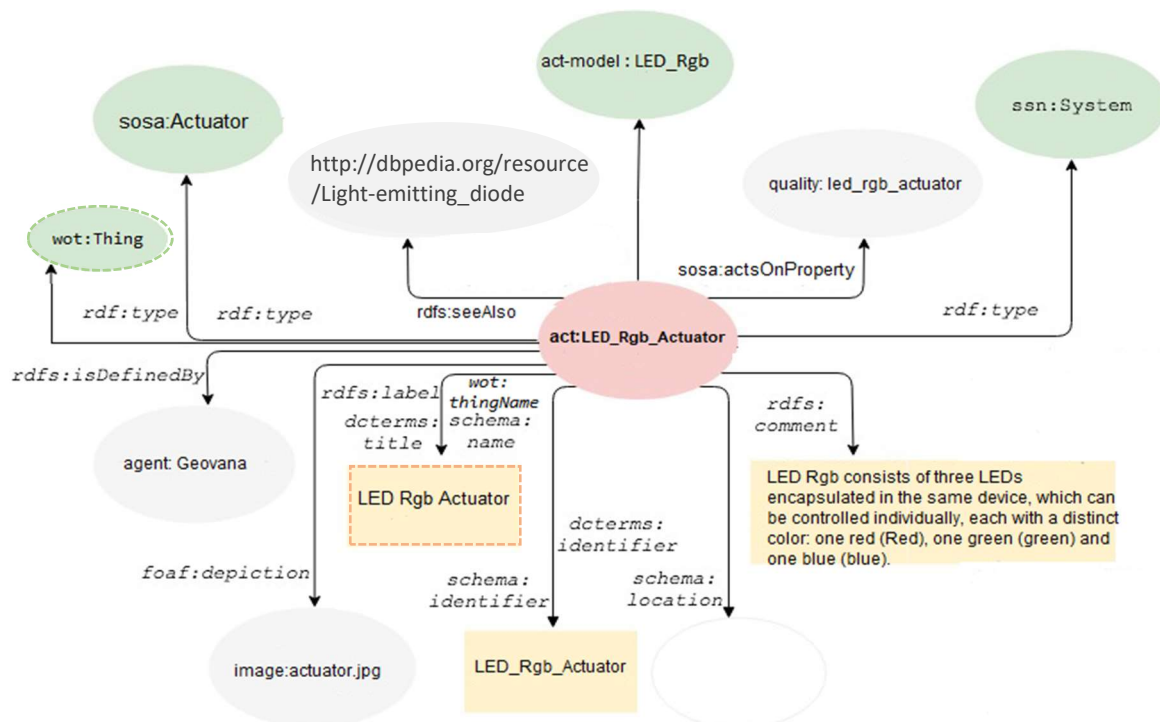
Esta primeira parte compreende a edição das características técnicas do atuador, aqui representado pelo “LED\_Rgb\_Actuator”, a saber: modelo do exemplar cadastrado, propriedades de atuação, informações sobre localização física, responsável pelo atuador, imagem do atuador etc. Na Figura 48 é possível visualizar a edição referente ao exemplar gerado. Para os dados gerados nas telas de criação e edição Parte I referentes ao exemplar LED\_Rgb\_Actuator, o correspondente grafo RDF subjacente é mostrado na Figura 49.

### LED\_Rgb\_Actuator

<b>Label:</b> <input type="text" value="LED Rgb Actuator"/>	<b>Postal code:</b> <input type="text" value="28030490"/>
<b>Acts on Property:</b> <input type="text" value="Content of LED_Rgb_Actuator"/>	<b>Latitude:</b> <input type="text" value="-21782472"/>
<b>Model:</b> <input type="text" value="LED_Rgb"/>	<b>Longitude:</b> <input type="text" value="-41333285"/>
<b>See Also:</b> <input type="text" value="http://dbpedia.org/resource/Light-emitting_diode"/>	<b>City:</b> <input type="text" value="Campos dos Goytacazes"/>
	<b>Country:</b> <input type="text" value="Brasil"/>
	<b>Comment</b> <input type="text" value="LED Rgb consists of three LEDs encapsulated in the same device, which can be controlled individually, each with a distinct"/>
	<b>Registered by:</b> <input type="text" value="Geovana da Silva Batista"/>
	<b>Select an image of the system</b> 

**Figura 48-** Tela de edição Parte I do exemplar LED\_Rgb\_Actuator.  
 Fonte: Elaboração Própria

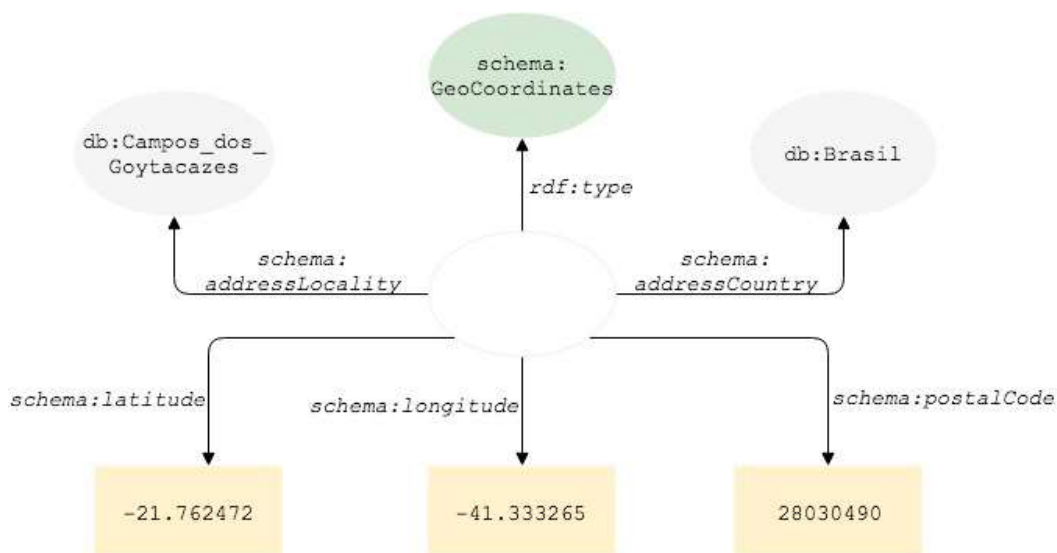




**Figura 49-** Grafo RDF subjacente a tela de edição Parte I do exemplar `LED_Rgb_Actuator`.  
Fonte: Elaboração Própria

Na Figura 49, além de ser uma instância de `sosa:Actuator`, `ssn:System`, o atuador “`LED_Rgb_Actuator`” é, ainda, em uma instância da classe “`LED_Rgb`” que é seu modelo. Por ser uma instância do modelo “`LED_Rgb`”, herda todas as características associadas a este modelo, tais como informações sobre procedimentos, fabricante e propriedades de objeto de interesse. Não bastante, o modelo também trata de uma instância da classe `wot:Thing`, confirmando sua identidade como sendo uma “coisa” do domínio WoT e assim, possibilitando realizar toda a sua descrição da interface de acesso.

No caso das propriedades de localização de atuadores, o modelo adotado para descrever tais informações é equivalente ao adotado para sensores. Além disso, neste exemplo em particular, os valores das propriedades de localização envolvidas no cadastro do atuador são os mesmos utilizados no cadastro do sensor “`DHT22_Sensor`”, tendo em vista que tais dispositivos fazem parte da mesma instalação WoT. A Figura 50 apresenta parte do grafo RDF subjacente ao cadastro de atuador “`LED_Rgb_Actuator`” com foco nas propriedades de localização.



**Figura 50** - Grafo RDF subjacente ao cadastro de atuador com foco nas propriedades de Localização.

Fonte: Rangel (2018)

## Parte II: Edição da Interface de acesso do Atuador

A segunda parte da tela de edição de atuadores, da mesma maneira dos sensores, consiste na principal contribuição deste trabalho. Nela, é possível, visualizar todos os elementos que compõem a descrição da interface de acesso ao atuador. Na figura 51, é possível visualizar a tela ainda sem a inserção de valores.

### DESCRIPTION OF ACCESS INTERFACE OF ACTUATORS

#### Exemplar Data

ImplementsSecurity:

URI\_Base

The Think is Accessible Through Link

Define Link +

#### Interaction Pattern

An interaction in the context of the Web of Things is an exchange of data between a Web client and a Thing. This data can be either given as input by the client, returned as output by the Thing or both. Three interaction patterns have been defined so far: Property, Action and Event.

##### Information about the Property Interaction Pattern

Property provides readable and/or writeable data that can be static (e.g., supported mode, rated output voltage, etc.) or dynamic (e.g., current fill level of water, minimum recorded temperature, etc.).

Property +

##### Information about the Action Interaction Pattern

The Action interaction pattern targets changes or processes on a Thing that take a certain time to complete (i.e., actions cannot be applied instantaneously like property writes). Examples include an LED fade in, moving a robot, brewing a cup of coffee, etc. Usually, ongoing Actions are modelled as Task resources, which are created when an Action invocation is received by the Thing.

Action +

##### Information about the Event Interaction Pattern

The Event interaction pattern enables a mechanism to be notified by a Thing on a certain condition.

Event +

**Figura 51** – Tela simplificada de edição parte II do atuador.

Fonte: Elaboração Própria



Assim com sensor, a descrição do atuador requer alguns metadados semânticos, padrão ou modelo de interação (Propriedade, Ação e Evento), esquema semântico para tornar os modelos de dados compreensíveis à máquina e recursos para *Web Linking* a fim de expressar as relações entre dispositivos. Não necessariamente todos os campos serão preenchidos quando se realiza o cadastro. Como exemplo, um atuador pode estar associado a um tipo de padrão de interação ou até mesmo a vários ou até mesmo fazer uso de mais de um tipo de protocolo para descrever tais interações. As telas das Figuras 52 e 54 mostram os elementos utilizados para descrever a interface de acesso do exemplar de atuador “LED\_Rgb\_Actuador”. Os padrões de interação associados são: *wot:Property* e *wot:Action*, ou seja, ao escolher na tela os padrões de interação “Property” e “Action” são gerados dois recursos anônimos (nós em branco) que são instâncias das classes *wot:InteractionPattern/wot:Property* e *wot:InteractionPattern/wot:Action*, respectivamente. A associação ocorre por meio da propriedade *rdf:type*. As telas destes dois padrões de interação e os correspondentes grafos RDF podem ser vistos nas Figuras 52-53 e 54-55, respectivamente. Para melhor compreensão e apresentação, são visualizadas as telas particionadas.

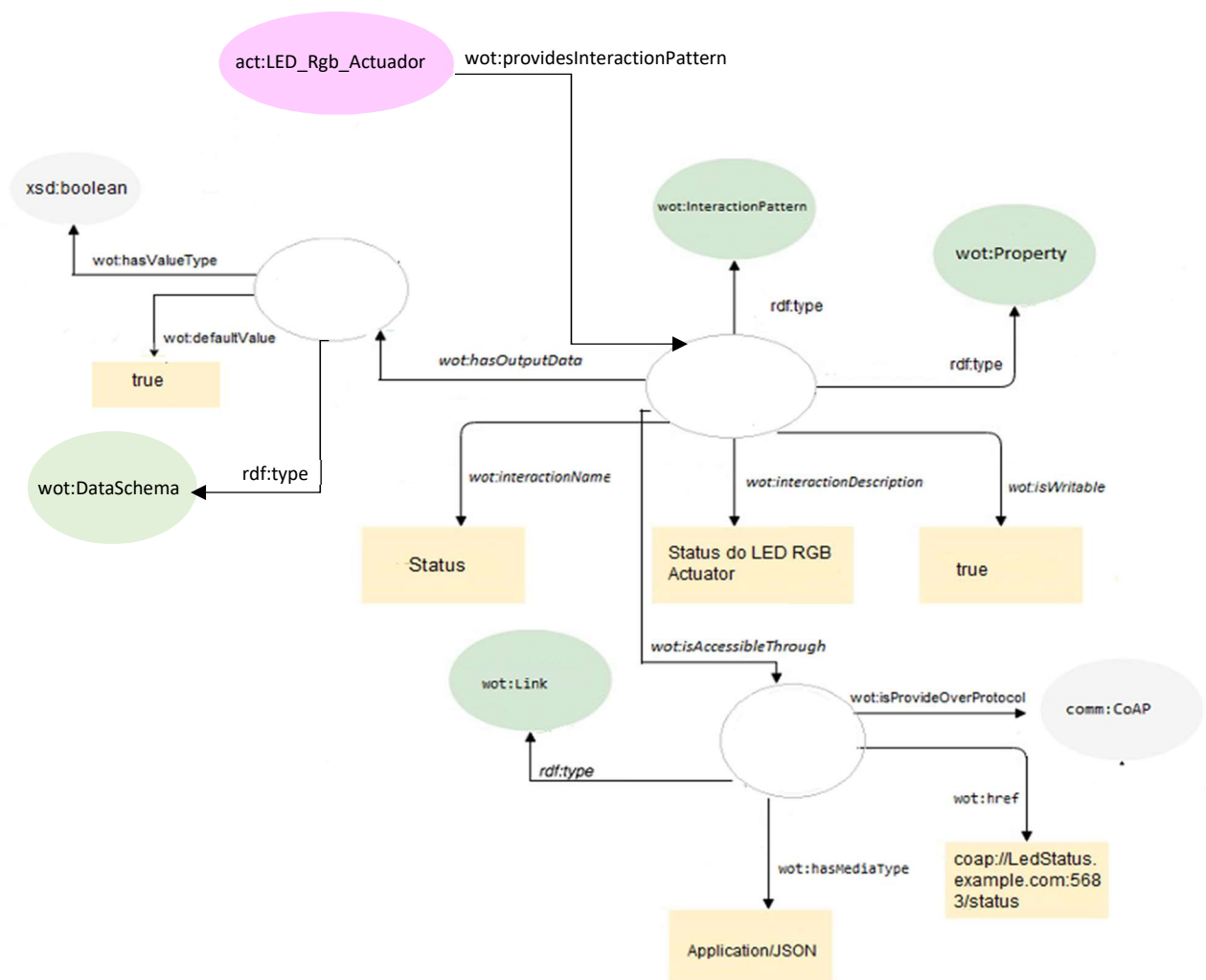
The screenshot shows a web-based configuration interface for a WOT Property interaction pattern. It is divided into three main sections:

- Information about the Property Interaction Pattern:** Contains a descriptive text about the property, a 'Property +' icon, an 'Interaction Name' field with the value 'Status', an 'Interaction Description' field with the value 'Satus do LED RGB Actuador', and an 'Is Writable' dropdown menu set to 'True'.
- Property's Output Data +:** Features a text input field containing 'true', a 'Unit of measurement:' dropdown menu with options like 'Abampere', 'Abampere per Square C', 'Abcoulomb', 'Abcoulomb per Square (', and 'Abfarad', and a 'Data Type:' dropdown menu with options like 'boolean', 'byte', 'date', 'dateTime', and 'decimal'. A red 'Remove Output Data' button is located at the bottom of this section.
- The Property Interaction Pattern is Accessible Through Link:** Includes a 'Define Link +' icon, a text input field for the URI 'coap://LedStatus.exempl', a dropdown menu for the media type 'application/json', and an 'Is Provided Over Protocol:' dropdown menu with 'CoAP' selected. Other options include 'Hypertext Transfer Protoc', 'Internet Protocol', and 'Simple Object Access P'.

**Figura 52** – Descrição do Padrão de Interação *Property* do Atuador.  
Fonte: Elaboração Própria

O padrão de interação "Property", conforme a Figura 52, recebe um nome pelo campo “Interaction Name” e uma breve descrição da interação é fornecida conforme o campo “Interaction Description”. Trata-se de uma propriedade editável, conforme atribuição do valor

“True” para o campo “Is Writable:”. Deste modo, é possível saber que o "LED\_Rgb\_Actuador" possui uma propriedade "Status" cujo valor pode ser alterado. A descrição dos dados de saída da Interação "Property" ocorre por meio da opção “Property’s Output Data” onde é permitido a declaração do tipo de dado e a unidade de medida fornecida pela Interação "Property". Deste modo, para o “Default Value”, foi inserido o valor “true” e declarado sendo um “xsd:Boolean” (tipo primitivo, predefinido pelo XML Schema). Note que o Padrão de Interação está acessível por meio de um Link, cujo endereço está definido como: “coap://LedStatus.example.com:5683/status” e o tipo de mídia gerado é um “Application/JSON”, emitindo assim, uma representação JSON dos recursos. Além disso, a interação é provida pelo protocolo “COAP”, que é um recurso previamente cadastrado na aplicação pelo usuário. O grafo RDF gerado a partir dos dados apresentados na tela de Edição Parte II referentes a descrição da interação "Property" do "LED\_Rgb\_Actuador" são apresentados na Figura 53.



**Figura 53** – Parte do grafo RDF subjacente a tela de edição Parte II do exemplar LED Rgb Actuator com ênfase no padrão de interação Property.

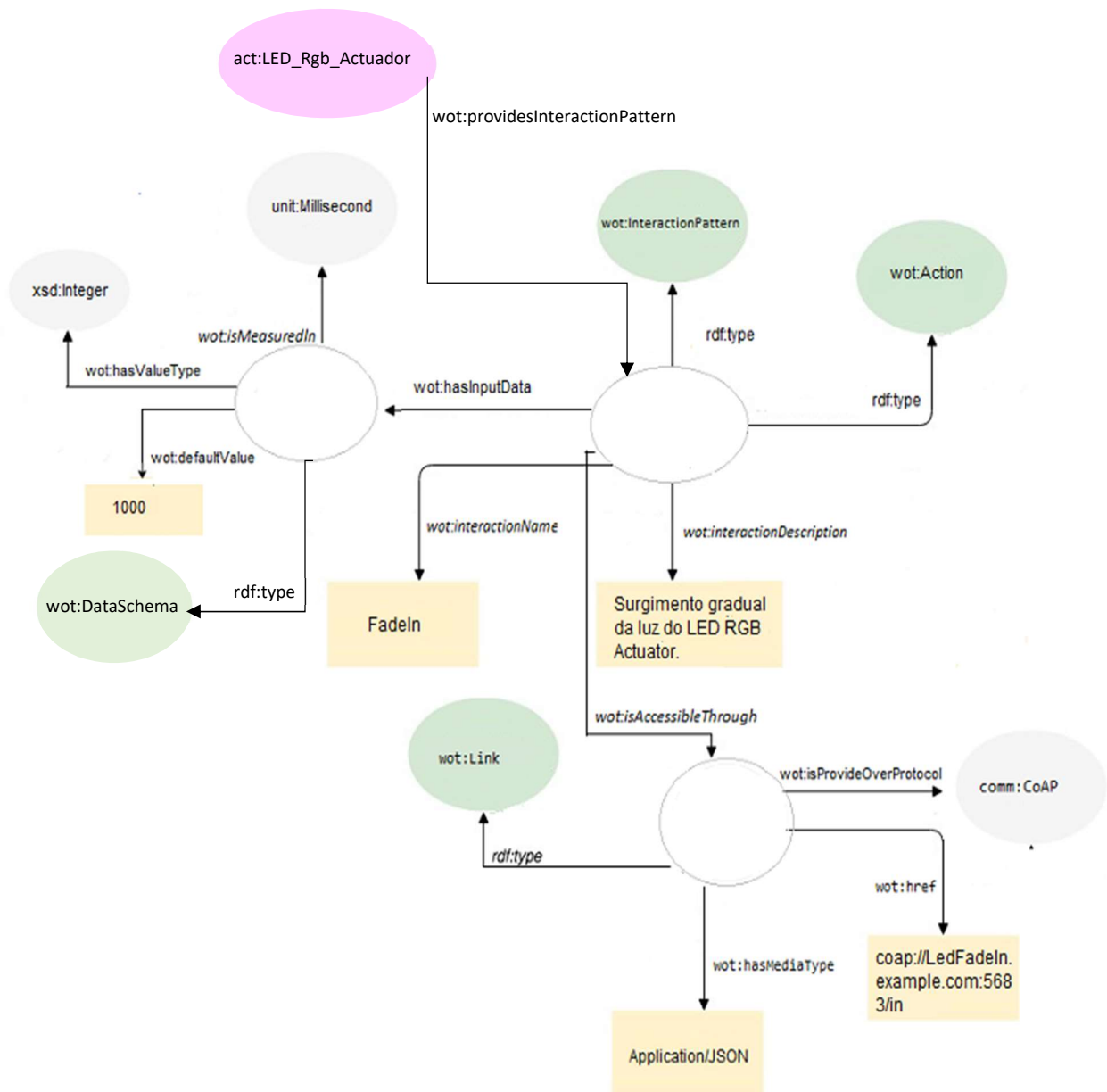
Fonte: Elaboração Própria

A seguir, na Figura 54 é possível acompanhar a tela de Edição Parte II referente à Interação Action.

**Figura 54** – Descrição do Padrão de Interação *Action* do Atuador.  
Fonte: Elaboração Própria

O padrão de interação "Action" visa disparar uma ação no dispositivo que leva certo tempo para ser concluída. Deste modo, conforme a Figura 54, a interação "Action" recebe um nome pelo campo "Interaction Name" e uma breve descrição da interação é fornecida conforme o campo "Interaction Description". Em suma, é possível saber que o "LED\_Rgb\_Actuator" realiza uma operação de "Fade In", ou seja, um acendimento gradual da luz. A descrição dos dados de entrada da Interação "Action" ocorre por meio da opção "Action's Input Data" onde é permitido a declaração do tipo de dado e a unidade de medida. Assim, para o "Default Value", foi inserido o valor "1000" e declarado sendo um "xsd:Integer" (tipo primitivo, predefinido pelo XML Schema) e a unidade de medida definida foi "Millisecond". Note que o Padrão de Interação está acessível por meio de um Link, cujo endereço está definido como "coap://LedFadeIn.example.com:5683/in" e o tipo de mídia gerado é um "Application/JSON", emitindo assim, uma representação JSON dos recursos. Além disso, a interação é provida pelo protocolo "COAP", que é um recurso previamente cadastrado na aplicação pelo usuário. O

grafo RDF gerado a partir dos dados apresentados na tela de Edição Parte II referentes a descrição da interação "Action" do "LED\_Rgb\_Actuador" é na Figura 55.



**Figura 55** – Parte do grafo RDF subjacente a tela de edição Parte II do exemplar LED Rgb Actuador com ênfase no padrão de interação Action.

Fonte: Elaboração Própria

## 7. TRABALHOS RELACIONADOS

A diversidade de trabalhos encontrados em vários campos de atuação como computação, engenharia, medicina, biologia, reflete o amplo universo em que as áreas de atuação estão envolvidas. Com o objetivo de padronização para provimento de sucesso na interoperabilidade dos objetos envolvidos nesse domínio, têm-se a necessidade, conforme já abordado neste trabalho, da criação de um model formal comum para representação do domínio estudado.

O W3C WG, atendendo as recomendações de publicação do W3C, apresentou em 2017 a ontologia *Web of Things (WoT) Thing Description*, um modelo formal e uma representação comum para descrição da “coisa” na *Web of Things (WoT)*. Uma descrição da coisa diz respeito aos metadados e as interfaces das coisas, onde uma coisa é uma abstração de uma entidade física ou virtual que fornece interações e participa na Web das coisas. Complementando, tem-se o projeto VICINITY, seguindo as especificações do grupo de Interesses do W3C no domínio de IoT, com a criação de várias ontologias (*Ontology model for Web of Things, Vicinity core model e Vicinity WoT mappings model*) com o intuito de modelar o domínio de WoT, representando os termos principais do domínio, além de descrever a ligação entre os dados oriundos de recursos Web, permitindo uma interoperabilidade no contexto IoT.

Corroborando a relevância da criação de uma representação comum para o Domínio IoT, tem-se ainda, criada pelo W3C, a ontologia *Semantic Sensor Network (SSN)*. Uma ontologia significativa que proporciona detalhadamente a descrição de sensores (e atuadores) e conceitos relacionados a IoT. Bermudez *et al* (2016), com a apresentação do IOT-Lite, propõe um modelo semântico leve para IoT, tratando-se de uma instância da Ontologia (SSN), onde o objetivo é proporcionar aos usuários finais, opções de consultas simples acerca dos dados da IoT.

Outra ontologia para modelagem da WoT é a *Semantic Web of Things (SWoT)* baseado em ontologia (SWoT-O)<sup>43</sup> desenvolvida no contexto SWoT4CPS (WU *et al*, 2017), que faz representação dos principais elementos da WoT, porém, de forma mais restrita do que definido pelo W3C WG. Esta ontologia abrange apenas os atuadores e não descreve a forma de acesso aos dados destes dispositivos. No trabalho de Allan e Noll (2010), é possível encontrar uma ontologia sobre os conceitos da WoT, porém esta não considera a modelagem e as ações de padrão de interação. Já a ontologia proposta por Charpenay *et al* (2016) apresenta um

---

<sup>43</sup> <https://github.com/minelabwot/SWoT/blob/master/swot-o.owl>

vocabulário para WoT que visa definir conceitos de IoT usando termos da Web, incluindo os dois conceitos identificados como os principais recursos WoT: *Thing Description* (TD) e Interação, que foi elaborada primeiramente pelo W3C IG para WoT, porém, não fornece mecanismos para indicar onde e como os valores dos recursos podem ser acessados e interpretados, além de se tratar de uma ontologia demasiadamente pesada para alguns casos de uso.

No trabalho do Zhukov *et al* (2017), é proposto o desenvolvimento de uma API sobre IoT baseado em ontologias, utilizando *Linked Data* em função do problema de incompatibilidade existente em inúmeras API's no domínio estudado. O trabalho requer algumas considerações nas questões de implementação, levando em consideração o fato da enorme variedade de dispositivos existentes, tal projeto exige um alto poder de computação.

Kamilaris e Ali (2017) propõem um motor de busca, nomeado como WoTS2E, para a Web Semântica das Coisas. O objetivo é reunir a Web Semântica e a IoT / WoT, associando informações semanticamente anotadas para dispositivos físicos habilitados para Web, serviços e seus dados, para integração contínua de dados e uma melhor compreensão das informações do mundo real. Baseado no rastreamento da Web, propõem uma maneira escalável e flexível de descobrir, em tempo real e em todo o mundo, dispositivos embutidos conectados na Web, bem como a semântica dos dados. Ainda não está em uso, faltando a conclusão do desenvolvimento de uma API.

Por fim, validando a proposta das aplicações *Linked Data* para IoT, tem-se o trabalho de Rangel e Jacyntho (2018), com a criação de uma aplicação semântica *Linked Data* para catalogação semântica de sensores, como forma de validação do modelo para descrição de sensores utilizando ontologias *Linked Data*. Nesta aplicação, por meio das classes de ontologias (SSN, Foaf, Geonames, etc.) os sensores são catalogados, com suas especificações (nome do sensor, dados de localização (país, longitude, latitude), informações técnicas condições de funcionamento, precisão, sensibilidade, etc.).

A realidade da IoT vem ao encontro das abordagens relevantes realizadas neste trabalho. As pesquisas, aplicações, ferramentas auxiliares contribuem ainda mais para um melhor aproveitamento futuro deste universo IoT.

## 8. CONSIDERAÇÕES FINAIS

O poderio da *Web* das Coisas (WoT) ganhará força quando existirem bastante aplicações aptas a reusar informações, provenientes de diversas fontes na *Web* e publicar e compartilhar seu conteúdo com outras aplicações. Tal propósito terá um maior desempenho mediante as criações de Aplicações semânticas *Linked Data*, permitindo a catalogação e publicação da descrição ou especificação das "coisas inteligentes" em uma Web Semântica inteligível por máquinas, fazendo uso de vocabulário formal (ontologia) comum.

O trabalho consistiu em um complemento do trabalho de Rangel (2018), uma aplicação *Linked Data*, que permite que sensores e atuadores possam ser pesquisados e encontrados, de forma precisa, na Web. Deste modo, o objetivo deste trabalho se concentrou em descrever formalmente "como" fazer uso desses dispositivos. Em outras palavras, descrever a interface de acesso do dispositivo, identificando seus parâmetros, protocolos de acesso e formatos de dados. Uma aplicação inteligível por seres humanos e máquinas e enriquecidos por outras bases de dados da nuvem LOD, colaborando com o reúso de informação na Web, tornando as pesquisas mais precisas e possibilitando melhores análises sobre os dados. Como mais uma contribuição importante deste trabalho tem-se os conceitos fundamentais da nova Web envolvidos: modelo de dados RDF, modelos de representação do conhecimento (ontologias) e linguagem de consulta SPARQL.

O vocabulário que norteou este trabalho é a ontologia, parte do Projeto VICINITY, nomeada como *Vicinity ontology model for Web of Things (WoT ontology)*, que formaliza a base para descrição da interface de acesso a uma coisa inteligente. Como o trabalho também contou com recursos já desenvolvidos na aplicação de Rangel (2018), para fins de conhecimento, a ontologia *Semantic Sensor Network (SSN)*, que compreende a maior parte dos termos utilizados para descrever tais dispositivos.

Além da ontologia SSN, outras ontologias foram adotadas para descrever os demais recursos ou requisitos funcionais. Assim, a aplicação semântica *Linked Data* desenvolvida contribui para que tais ontologias se desenvolvam ainda mais. Deste modo, à medida que se utiliza a aplicação semântica, informações ficam disponíveis na base de dados da própria aplicação a fim de serem reutilizadas por humanos ou agentes de software, pela ótica do *Linked Data Mashup (LDM)*.

## 9. RESULTADOS

Mediante o abrangente cenário da IoT e WoT, os usuários não estão cientes de todos os aspectos possíveis, de todas as plataformas e de todas as coisas inteligentes existentes e espalhadas geograficamente. Uma forma inicial de dirimir a problemática é catalogar e publicar esses exemplares na Web de forma que tanto humanos quanto máquinas possam buscá-los e usá-los. Como segundo passo, descrever a forma como essas "coisas inteligentes" são acessadas. Assim, será possível saber que tipo de dispositivo está sendo usado, qual o protocolo envolvido, que formato de dados.

O presente trabalho alcançou o objetivo proposto que consistiu na abordagem para descrever formalmente a interface de acesso de "coisas inteligentes" (sensores e atuadores), bem como no desenvolvimento de uma aplicação semântica *Linked Data*, seguindo as especificações da W3C WG e os princípios *Linked Data*, propostos por Berners-Lee et al. (2006): URIs são utilizadas para nomear os recursos cadastrados na aplicação, as URIs possuem acesso HTTP, retornam arquivo RDF e são conectadas a outros recursos (ou URIs) na Web. Links semânticos (mashup) foram realizados, utilizando a fonte de dados DBpedia, objeto central da nuvem LOD. A reutilização de informações desse repositório valorizou a aplicação, uma vez que aumentou o universo da base de conhecimento.

A seleção das ontologias destacou-se como uma etapa essencial do trabalho. A utilização de ontologias consagradas, partes do catálogo LOV, para uma melhor representação do domínio IoT enriqueceu o trabalho, uma vez que, projetos de aplicações de dados ligados, visam essencialmente produzir informação interoperável. No mais, a utilização da base do catálogo semântico de Rangel(2018), adicionado ao uso da ontologia WoT Ontology, parte do Projeto VICINITY, que serviu de fonte principal para o desenvolvimento da aplicação *Linked Data* proposta neste trabalho, para descrição da interface de acesso de coisas inteligentes (sensores e atuadores), diferenciou a aplicação desenvolvida de outras identificadas na revisão bibliográfica. A aplicação foi construída de modo a atender o contexto dos sensores e atuadores, seguindo o trabalho de Rangel (2018). Deste modo, atendendo, com maior completude, os sistemas Internet das Coisas ou Web das Coisas, atestado por exemplos realistas, trazendo elementos do domínio IoT.

Na reprodução dos exemplos, ficou claro que a aplicação é capaz de descrever os problemas apresentados contemplando os aspectos propostos neste trabalho, tais como descrição da interface de acesso de uma "coisa inteligente", bem como a informação pública



ser consumida tanto por humanos, mas, sobretudo, por agentes de software que possam pesquisar por coisas inteligentes na Web e usá-las, de forma autônoma, sem assistência humana.

As concepções iniciais deste trabalho foram publicadas em artigo científico e podem ser consultadas em Batista & Jacyntho (2018). Desde sua publicação, novos resultados foram obtidos e refletem a evolução do desenvolvimento deste trabalho. Esforços foram concentrados na análise e estudo de ontologias, tendo como resultado a inclusão de novos termos para descrição dos recursos. Por meio destes e demais avanços futuros, uma vez disponível na Web de Dados, recursos poderão ser cadastrados na aplicação.

Por fim, por se tratar de um tema bastante novo, este trabalho deixa um legado valioso de diretrizes de como criar uma aplicação semântica *Linked Data* para cadastro da interface e uso de sensores e atuadores, que pode ser utilizado no desenvolvimento de trabalhos aplicações similares futuras.

## 10. TRABALHOS FUTUROS

No decorrer do desenvolvimento do trabalho, chances de avanços significativos foram observados. Como futuras investidas, deve compreender na adição de novas ontologias, como *Vicinity core model*, *Vicinity WoT mappings model*, *Vicinity adapters model*, relacionadas ao domínio estudado com suas respectivas propriedades a fim de criar ligações, proporcionando relacionamentos com outras aplicações semânticas de WoT já existentes na *Web*, trocando e buscando informações desejadas. Além de um estudo mais exploratório da ontologia, *WoT Ontology*, podendo enriquecer ainda mais o processo de descrição da interface e uso dos sensores e atuadores.

Avanços futuros devem contemplar, ainda o registro semântico dos dados observados, usando a parte de "Observation" da ontologia SSN, dentre outras. Sendo indispensável pensar em uma arquitetura *Big Data* para os dados obtidos em tempo real ou quase real. Para esta finalidade, tem-se a tecnologia de semantic blockchain<sup>44</sup> (RUTA et al., 2017) como importante opção a ser estudada.

O campo de segurança é tratado de forma opcional na aplicação. Este, tem como intenção, ser usado para fornecer informações de metadados de acesso (auto-contidas) do dispositivo para transmitir com segurança informações através de todos os seus recursos. Deste modo, em futuros trabalhos, podem conter um maior estudo acerca do tema, bem como, criação de algumas funcionalidades que envolve o tema, como por exemplo, definição de perfis de usuários para cadastrar ou obter tais dados dos sensores e atuadores.

Com o intuito de continuar a fazer parte deste novo paradigma da *Web*, atuando diretamente no domínio de IoT, que já faz parte de uma nova era de tecnologia de comunicação, espera-se que as melhorias futuras possam levar a um produto final totalmente funcional, um portal semântico de sensores e atuadores, utilizado em larga escala, por muitos usuários geograficamente distribuído, cuja informação publicada seja compreensível tanto por agentes computacionais, quanto por humanos, contribuindo, pois, para a construção da crescente *Web Semântica das Coisas (SWoT)*.

---

<sup>44</sup> Semantic Blockchain é um banco de dados distribuído (P2P) que mantém uma lista encadeada de blocos de registros de dados padrão (em geral, RDF), de crescimento contínuo e praticamente imune à adulteração e modificação

## REFERÊNCIAS BIBLIOGRÁFICAS

- ANDREAS KAMILARIS, ANDREAS PITSILLIDES, FRANCESC X. PRENAFETA-BOLD, MUHAMMAD INTIZAR ALI, "A Web of Things based eco-system for urban computing - towards smarter cities", *Telecommunications (ICT) 2017 24th International Conference on*, pp. 1-7, 2017.
- ALAM, S., NOLL, J.: A semantic enhanced service proxy framework for internet of things. In: Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, IEEE Computer Society (2010) 488-495.
- ANTONIOU, G. et al, 2012. A Semantic Web Primer, 3rd edition. The MIT Press.
- APPOLINÁRIO, F. Metodologia da ciência: filosofia e prática da pesquisa. São Paulo: Cengage Learning, 2009.
- ASHTON, K. "That 'Internet of Things' Thing", *RFID Journal*, 22 June 2009.
- AZEVEDO, R.S. AND JACYNTHO, M.D., 2014. Um Modelo Baseado em Ontologias *Linked Data* para Catalogação de Projetos de Software. In 12ª Conferência Ibero-Americana WWW/Internet 2014, Porto, Portugal.
- BALBINOT, A.; BRUSAMARELLO, V.J. Instrumentação e Fundamentos de Medidas 2a. ed. LTC, 2011
- BARACHO, Renata MA; JUNIOR, Mário L. Pereira; ALMEIDA, Maurício B. Ontologia, Internet das Coisas e Modelagem da Informação da Construção (BIM): Estudo Exploratório e a Inter-relação entre as Tecnologias.

BATISTA AND JACYNTHO, MD., 2016. CATALOGAÇÃO SEMÂNTICA DE EVENTOS COM BASE NOS PRINCÍPIOS DO *LINKED DATA*. In 15ª Conferência Ibero-Americana WWW/Internet 2014, Porto, Portugal.

BATISTA AND JACYNTHO, MD., 2018. WEB SEMÂNTICA EM MONITORAMENTO AMBIENTAL: DESCRIÇÃO FORMAL DE INTERFACE DE ACESSO DE SENSORES E ATUADORES. XXI ENMC – Encontro Nacional de Modelagem Computacional e IX ECTM – Encontro de Ciências e Tecnologia de Materiais. Búzios, RJ – 08 a 11 Outubro 2018.

BERMUDEZ-EDO, MARIA et al. IoT-Lite: a lightweight semantic model for the Internet of Things. In: Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld), 2016 Intl IEEE Conferences. IEEE, 2016. p. 90-97.

BERNERS-LEE, T. et al. Creating a Science of the Web. *Science*, v. 313, n. 5788, p. 769–771, 11 ago. 2006.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. *Scientific American*, v. 284, n. 5, p. 34–43, 2001.

BERNERS-LEE, T. Putting government data online. <http://www.w3.org/DesignIssues/GovData.html>, 2009.

BIZER, C.; HEATH, T.; BERNERS-LEE, T. *Linked Data* - The story so far. *International Journal on Semantic Web and Information Systems*, v. 5, n. 3, p. 1–22, 2009.

BIZER, C., Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of

data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

CASTILLO, Juan Carlos Martin. *Instalaciones Domóticas*. Madri: Editex, 2009.

CHARPENAY, V., KABISCH, S., KOSCH, H.: Introducing thing descriptions and interactions: An ontology for the web of things. In: *Proceedings of the 1st Workshop on SemanticWeb technologies for the Internet of Things (SWIT) at ISWC*. (2016).

CLARK, K. G.; FEIGENBAUM, L.; TORRES, E. **SPARQL Protocol for RDF**. 2008. Disponível em: < <http://www.w3.org/TR/rdf-sparql-protocol/> >. Acesso em: 20 dez. 2018.

COSSINI, F. REZENDE, M. N. (2016). Building an ontology for the internet of things (iot) standards. 13CONTECSI/PS-4114.

CUNHA, C.L.M. 2010. *Redes Sociais Dirigidas ao Contexto das Coisas*. Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

DATTA, S.K., FERREIRA DA COSTA, R.P., BONNET, C.: Resource discovery in Internet of Things: Current trends and future standardization aspects. In: *WF-IOT 2015, IEEE 2nd World Forum on Internet of Things*, Milan, Italy. (12 2015).

DEAN, M.; SCHREIBER, G. *OWL Web Ontology Language Reference*. W3C Recommendation 10 February 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-ref-20040210/>> Acessado em: 07 de dezembro de 2018.

DIAS, T. D; SANTOS, N. *Web Semântica: Conceitos Básicos e Tecnologias Associadas*. *Cadernos do IME Série Informática*, 14:p. 25–38, 2003.

- D. BRICKLEY and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema - W3C Recommendation. <http://www.w3.org/TR/rdf-schema/>, 2004.
- GIL, A. C. Como elaborar projetos de pesquisa. 5. ed. São Paulo: Atlas. 2010.
- GKOUTOS, G. V., SCHOFIELD, P. N., & HOEHNDORF, R. (2012). The Units Ontology: a tool for integrating units of measurement in science. *Database : the journal of biological databases and curation*, 2012, bas033.
- GUARINO, N., OBERLE, D., e STAAB, S. 2009. O que é uma ontologia? Em S. Staab e R. Studer (eds.), *Handbook em ontologias*, Second Edition. Springer Verlag: 1-17.
- GUINARD, DOMINIQUE D. AND TRIFA, Vlad M. (2016). *Building the Web of Things. With examples in Node.js and Raspberry Pi*. ISBN 9781617292682, 344 pages.
- GUINARD, Dominique; TRIFA, Vlad; WILDE, Erik. A resource oriented architecture for the web of things. In: *Internet of Things (IOT)*, 2010. IEEE, 2010. p. 1-8.
- HAKALA, Juha. Dublin core metadata initiative. In: *CRIS*. 2000.
- HALLER, A., Cox, K.J. S., Phuoc, D. L., Taylor, K., Lefrançois, M. SSN: The Semantic Sensor Network Ontology. W3C 19 October 2017. W3C Recommendation. URL: <https://www.w3.org/TR/vocab-ssn/>
- HEATH T. AND BIZER, C. 2011. *Linked Data: Evolving the Web into a Global Data Space*, 1ª Edição. Morgan & Claypool.
- JACYNTHO, M. D., 2012. Um modelo de bloqueio multigranular para RDF. Tese (Doutor) Departamento de Informática, PUC-RIO, Rio de Janeiro, Brasil, pp. 47-59.

- LAUFER, C. . Guia de Web Semântica. São Paulo: Ceweb - Centro de Estudos sobre Tecnologias Web, 2015 (Monografia em Ciência da Computação).
- LI, Shancang; DA XU, Li; ZHAO, Shanshan. The internet of things: a survey. *Information Systems Frontiers*, v. 17, n. 2, p. 243-259, 2015.
- LOPEZ Research 2013. “Uma introdução à Internet da Coisas (IoT)” LLC 2269 Chestnut Street #202 San Francisco, CA 94123.
- MARQUES, F. (2017). O Brasil da Internet das Coisas. ED. 259, Disponível em < <http://revistapesquisa.fapesp.br/2017/09/21/o-brasil-da-internet-das-coisas/>>.  
Acessado em 10.01.2018.
- MASTERS, James E. (2010). QUDT Units Vocabulary (without dimensions) Version 1.1. Disponível em < <http://qudt.org/1.1/vocab/unit/>> Acessado em 23.12.2018.
- NATH, K. AND ISWARY, R., 2015. What Comes after Web 3.0? Web 4.0 and the Future. *International Conference on Computing and Communication Systems (I3CS'15)*. Shillong, India, Vol 2.
- PRUD'HOMMEAUX, E.; SEABORNE, A. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008. Disponível em < <http://www.w3.org/TR/rdf-sparql-query/>>. Acessado em 04 de dezembro de 2011.
- RANGEL, Tamiris de Sousa. 2018. Catalogação semântica de sensores e atuadores com base nos princípios *Linked Data*. Dissertação (Mestrado em Sistemas Aplicados a Engenharia e Gestão) Instituto Federal Fluminense. Campos dos Goytacazes, RJ.

RANGEL AND JACYNTHO, MD., 2016. PROPOSTA DE MODELO BASEADO EM ONTOLOGIAS *LINKED DATA* PARA CATOLOGAÇÃO DE SENSORES. In 15ª Conferência Ibero-Americana WWW/Internet 2014, Porto, Portugal.

RUTA, M., SCIOSCIA, F., LEVA, S., CAPURSO, G., Di SCIASCIO, E. (2017). Semantic Web blockchain to improve scability in the internet of things. *Open Journal of Internet Of Things (OJIOT)*, 3(1), 46-61.

SANTAREM SEGUNDO, J. E. Representação Iterativa: um modelo para Repositórios Digitais. São Paulo: Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, 2010.

SANTAREM SEGUNDO, J. E. AND CONEGLIAN, C. S. 2016. Web semântica e ontologias: um estudo sobre construção de axiomas e uso de inferências. *Informação & Informação*, Vol. 21, No. 2, pp. 217 – 244.

SCHEMA.ORG. about page - schema.org. Disponível em: <<https://schema.org/docs/about.html>>.

SERENA, Fernando; POVEDA-VILLALÓN, María; GARCÍA-CASTRO, Raúl. Semantic Discovery in the Web of Things. In: *International Conference on Web Engineering*. Springer, Cham, 2017. p. 19-31.

SOUZA, R. R. AND ALVARENGA, L., 2004. A Web Semântica e suas contribuições para a ciência da informação. *Ciência da Informação*, Vol. 33, No. 1, pp. 132-141.

SCHNEIDER, M. *OWL 2 Web Ontology Language: RDF-Based Semantics - 2nd edition*. W3C, 2012. Disponível em: <http://www.w3.org/TR/owl2-rdf-based-semantics/>. Acesso: 29 de Janeiro, 2019.



WANG, Y.-P. Eric et al. A primer on 3GPP narrowband Internet of Things. IEEE Communications Magazine, v. 55, n. 3, p. 117-123, 2017.

WU, Z., Xu, Y., Yang, Y., Zhang, C., Zhu, X., Y., Y. : For a semantic web of things: Semantic annotation hybrid, extraction and structure for the reasoning of the cyberphysical system. sensors 17 (2) (2017) 403

ZHUKOV, Vyacheslav; KOMAROV, Mikhail. Semantic control method of the Internet of Things based on Linked Open Data. In: Business Informatics (CBI), 2017 IEEE 19th Conference on. IEEE, 2017. p. 1-4.

## APÊNDICE

Este apêndice tem como objetivo apresentar os grafos RDF (em sintaxe Turtle) gerados como resultado dos cadastros dos recursos apresentados nos capítulos 6.

### A. Descrição de Segurança “JSON Web Token”

```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix schema: <http://schema.org/> .
@prefix wot: <http://iot.linkeddata.es/def/wot#> .
<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/json+web+token>
  calli:administrator <http://localhost:8080/auth/groups/admin> ;
  dc:created "2019-04-01T09:24:24.267000+00:00"^^xsd:dateTime ;
  rdfs:comment "são credenciais, que podem conceder acesso a recursos. " ;
  calli:editor <http://localhost:8080/auth/groups/power>, <http://localhost:8080/auth/groups/staff> ;
  rdfs:label "JSON Web Token" ;
  calli:reader <http://localhost:8080/auth/groups/system>, <http://localhost:8080/auth/groups/public> ;
  dc:title "JSON Web Token" ;
  calli:subscriber <http://localhost:8080/auth/groups/everyone> ;
  prov:wasGeneratedBy <http://localhost:8080/callimachus/changes/2019/05/07/t16a90afd1e8x13529#provenance> ;
  schema:identifier "JSON Web Token" ;
  dc:identifier "JSON Web Token" ;
  schema:name "JSON Web Token" ;
  a <http://localhost:8080/ProjetoFinal/WoTDescription/Classes/Security/Security>,
    wot:Security ;
<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/json+web+token?describe>
  foaf:primaryTopic <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/json+web+token> ;
  a <http://www.w3.org/ns/ldp#RDFSource> .

```

**Figura 56** - Grafo RDF subjacente ao cadastro de Segurança na Aplicação semântica em Turtle.

Fonte: Elaboração própria

## B. Descrição de Protocolo de Comunicação “Simple Object Access Protocol”

```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix schema: <http://schema.org/> .
@prefix wot: <http://iot.linkeddata.es/def/wot#> .

<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/soap?describe>
  a <http://www.w3.org/ns/ldp#RDFSsource> ;
  foaf:primaryTopic <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/soap> .

<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/soap>
  calli:administrator <http://localhost:8080/auth/groups/admin> ;
  calli:editor <http://localhost:8080/auth/groups/power>, <http://localhost:8080/auth/groups/staff> ;
  calli:reader <http://localhost:8080/auth/groups/public>, <http://localhost:8080/auth/groups/system> ;
  calli:subscriber <http://localhost:8080/auth/groups/everyone> ;
  a <http://localhost:8080/ProjetoFinal/WoTDescription/Classes/CommunicationProtocol/CommunicationProtocol> ,
    wot:CommunicationProtocol;
  rdfs:label "Simple Object Access Protocol"^^xsd:string ;
  prov:wasGeneratedBy <http://localhost:8080/callimachus/changes/2019/03/10/t16964681b9ex26493#provenance> ;
  rdfs:comment "SOAP (Simple Object Access Protocol, em português Protocolo Simples de Acesso a Objetos) é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída. Ele se baseia na Linguagem de Marcação Extensível (XML) para seu formato de mensagem, e normalmente baseia-se em outros protocolos da camada de aplicação, mais notavelmente em chamada de procedimento remoto (RPC) e Protocolo de transferência de hipertexto (HTTP), para negociação e transmissão de mensagens. SOAP pode formar a camada base de uma pilha de protocolos de serviços Web, fornecendo um arcabouço básico de mensagens sob o qual se podem construir os serviços Web."^^xsd:string ;
  dc:created "2019-03-10T16:08:53.459Z"^^xsd:dateTime ;
  dc:modified "2019-03-10T16:12:03.830Z"^^xsd:dateTime ;
  dc:title "Simple Object Access Protocol"^^xsd:string ;
  dc:identifier "SOAP"^^xsd:string ;
  rdfs:seeAlso <http://dbpedia.org/resource/SOAP>
  owl:sameAs <http://dbpedia.org/resource/SOAP> ;
  schema:identifier "SOAP"^^xsd:string ;
  schema:name "Simple Object Access Protocol"^^xsd:string .

```

**Figura 57** - Grafo RDF subjacente ao cadastro de Protocolo de Comunicação na Aplicação semântica em Turtle.  
Fonte: Elaboração própria

### C. Descrição de um exemplar de Sensor “DHT22 Sensor”

```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix schema: <http://schema.org/> .
@prefix wot: <http://iot.linkeddata.es/def/wot#/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .

<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/dht22_sensor?describe>
  a <http://www.w3.org/ns/ldp#RDFSSource> ;
  foaf:primaryTopic <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/dht22_sensor> .

<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/dht22_sensor>
  calli:administrator <http://localhost:8080/auth/groups/admin> ;
  calli:editor <http://localhost:8080/auth/groups/power>, <http://localhost:8080/auth/groups/staff> ;
  calli:reader <http://localhost:8080/auth/groups/public>, <http://localhost:8080/auth/groups/system> ;
  calli:subscriber <http://localhost:8080/auth/groups/everyone> ;
  a sosa:Sensor ,
    ssn:System ,
    wot:Thing , <http://localhost:8080/Resource/SensorModel/dht22>,
<http://localhost:8080/ProjetoFinal/WoTDescription/Classes/Sensor/Sensor> ;
  rdfs:label "DHT22 sensor"^^xsd:string ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2019/05/30/t16b02833aa6x17094#provenance> ;
  rdfs:comment "A sensor that observes the temperature and humidity on the air."^^xsd:string ;
  dc:created "2019-04-01T08:01:36.224Z"^^xsd:dateTime ;
  dc:modified "2019-04-01T09:20:43.947Z"^^xsd:dateTime, "2019-04-01T09:26:20.725Z"^^xsd:dateTime,
"2019-04-02T12:05:38.918Z"^^xsd:dateTime, "2019-04-02T15:52:55.043Z"^^xsd:dateTime, "2019-04-
02T15:54:44.528Z"^^xsd:dateTime, "2019-04-02T17:04:15.682Z"^^xsd:dateTime, "2019-04-
04T01:05:04.383Z"^^xsd:dateTime, "2019-04-04T01:29:03.028Z";
  foaf:depiction <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/sensor.jpg> ;
  dc:title "DHT22 sensor"^^xsd:string ;
  dc:identifier "DHT22_Sensor"^^xsd:string ;
  schema:identifier "DHT22_Sensor"^^xsd:string ;
  schema:location [
    a schema:GeoCoordinates ;
    schema:addressCountry <http://dbpedia.org/resource/Brazil> ;
    schema:addressLocality <http://dbpedia.org/resource/Campos_dos_Goytacazes> ;
    schema:latitude "-21.762472"^^xsd:string ;
    schema:longitude "-41.333265"^^xsd:string ;
    schema:postalCode "28030490"^^xsd:string
  ] ;

```

```

schema:name "DHT22 sensor"^^xsd:string ;
rdfs:isDefinedBy <http://localhost:8080/Resource/Person/tamiris> ;
prov:wasAttributedTo <http://localhost:8080/Resource/Person/mark> ;
sosa:observes <http://localhost:8080/Resource/Property/relative_humidity_of_the_room_125>,
<http://localhost:8080/Resource/Property/temperature_of_the_room_125> ;
wot:providesInteractionPattern [
  a wot:InteractionPattern, wot:Property ;
  wot:interactionDescription "Temperature of sensor DHT22"^^xsd:string ;
  wot:interactionName "Temperature"^^xsd:string ;
  wot:isWritable false^^xsd:boolean ;
  wot:hasOutputData [
    a wot:DataSchema ;
    wot:isMeasuredIn <http://qudt.org/vocab/unit#DegreeCelsius> ;
    wot:hasValueType xsd:decimal ;
    wot:defaultValue "36"^^xsd:string
  ]
]
wot:isAccessibleThrough [
  a wot:Link ;
  wot:isProvidedOverProtocol <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/CoAP_1> ;
  wot:hasMediaType "Application/JSON"^^xsd:string ;
  wot:href "coap://sensortemp.example.com:5683/temp"^^xsd:string
]
] ;
wot:implementsSecurity <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/json+web+token> .

```

**Figura 58** - Grafo RDF subjacente ao cadastro de um exemplar de Sensor na Aplicação semântica em Turtle.

Fonte: Elaboração própria

### D. Descrição de um exemplar de Atuador “LED\_Rgb\_Actuator”

```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix schema: <http://schema.org/> .
@prefix wot: <http://iot.linkeddata.es/def/wot#> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .

<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/led_rgb_actuator?describe>
  a <http://www.w3.org/ns/ldp#RDFSSource> ;
  foaf:primaryTopic <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/led_rgb_actuator> .

<http://localhost:8080/ProjetoFinal/WoTDescription/Resources/led_rgb_actuator>
  calli:administrator <http://localhost:8080/auth/groups/admin> ;
  calli:editor <http://localhost:8080/auth/groups/power>, <http://localhost:8080/auth/groups/staff> ;
  calli:reader <http://localhost:8080/auth/groups/public>, <http://localhost:8080/auth/groups/system> ;
  calli:subscriber <http://localhost:8080/auth/groups/everyone> ;
  a sosa:Actuator,
    ssn:System ,
    wot:Thing , http://localhost:8080/Resource/ActuatorModel/led_rgb ,

<http://localhost:8080/ProjetoFinal/WoTDescription/Classes/Actuator/Actuator> ;
  rdfs:label "LED Rgb Actuator"^^xsd:string ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2019/05/30/t16b02833aa6x17597#provenance> ;
  rdfs:comment "LED Rgb consists of three LEDs encapsulated in the same device, which can be controlled
individually, each with a distinct color: one red (Red), one green (green) and one blue
(blue)."^^xsd:string ;
  dc:created "2019-04-07T19:59:16.013Z"^^xsd:dateTime ;
  dc:modified "2019-04-07T20:21:19.208Z"^^xsd:dateTime, "2019-04-10T06:48:49.243Z"^^xsd:dateTime,
"2019-05-07T06:27:50.461Z"^^xsd:dateTime, "2019-05-07T16:56:34.325Z"^^xsd:dateTime;
  foaf:depiction <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/ledrgb.jpg> ;
  dc:title "LED Rgb Actuator"^^xsd:string ;
  dc:identifier "LED_Rgb_Actuator"^^xsd:string ;
  schema:identifier "LED_Rgb_Actuator"^^xsd:string ;
  schema:location [
    a schema:GeoCoordinates ;
    schema:addressCountry <http://dbpedia.org/resource/Brazil> ;
    schema:addressLocality <http://dbpedia.org/resource/Campos_dos_Goytacazes> ;
    schema:latitude "-21762472"^^xsd:string ;
    schema:longitude "-41333265"^^xsd:string ;
    schema:postalCode "28030490"^^xsd:string
  ] ;
  schema:name "LED Rgb Actuator"^^xsd:string ;

```

```

rdfs:isDefinedBy <http://localhost:8080/Resource/Person/Geovana> ;
wot:implementsSecurity <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/json+web+token> ;
sosa:actsOnProperty
<http://localhost:8080/Resource/Property/content_of_Led_Rgb_Actuator> .
rdfs:seeAlso <http://dbpedia.org/resource/Lightemitting_diode/> ;
wot:providesInteractionPattern [
  a wot:InteractionPattern, wot:Property ;
  wot:hasOutputData [
    a wot:DataSchema ;
    wot:defaultValue "true"^^xsd:string ;
    wot:hasValueType xsd:boolean
  ] ;
  wot:interactionDescription "Satus do LED RGB Actuator"^^xsd:string ;
  wot:interactionName "Status"^^xsd:string ;
  wot:isAccessibleThrough [
    a wot:Link ;
    wot:hasMediaType "application/json"^^xsd:string ;
    wot:href "coap://LedStatus.example.com:5683/status"^^xsd:string ;
    wot:isProvidedOverProtocol <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/CoAP_1>
  ] ;
  wot:isWritable true^^xsd:boolean ;
], [
  a wot:Action, wot:InteractionPattern ;
  wot:hasInputData [
    a wot:DataSchema ;
    wot:defaultValue "1000"^^xsd:string ;
    wot:hasValueType xsd:integer ;
    wot:isMeasuredIn <http://qudt.org/vocab/unit#MilliSecond>
  ] ;
  wot:interactionDescription "Surgimento gradual da luz do LED RGB Actuator."^^xsd:string ;
  wot:interactionName "FadeIn"^^xsd:string ;
  wot:isAccessibleThrough [
    a wot:Link ;
    wot:hasMediaType "application/json"^^xsd:string ;
    wot:href "coap://LedFadeIn.example.com:5683/in"^^xsd:string ;
    wot:isProvidedOverProtocol <http://localhost:8080/ProjetoFinal/WoTDescription/Resources/CoAP_1>
  ] ;
  ns1:isQueueable false^^xsd:boolean ;
] ;
wot:thingName "LED Rgb Actuator"^^xsd:string .

```

**Figura 59** - Grafo RDF subjacente ao cadastro de um exemplar de Atuador na Aplicação semântica em Turtle.

Fonte: Elaboração própria