

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

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

**TAMIRIS DE SOUSA RANGEL**

**CATALOGAÇÃO SEMÂNTICA DE SENSORES E ATUADORES COM  
BASE NOS PRINCÍPIOS LINKED DATA**

**Campos dos Goytacazes/RJ  
2018**

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

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

**TAMIRIS DE SOUSA RANGEL**

**CATALOGAÇÃO SEMÂNTICA DE SENSORES E ATUADORES COM  
BASE NOS PRINCÍPIOS LINKED DATA**

Mark Douglas Jacyntho de Azevedo  
(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  
2018

Biblioteca Anton Dakitsch  
CIP - Catalogação na Publicação

R196c Rangel, Tamiris de Sousa  
CATALOGAÇÃO SEMÂNTICA DE Sensores e Atuadores  
COM BASE NOS PRINCÍPIOS LINKED DATA  
/ Tamiris de Sousa Rangel - 2018.  
197 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, 2018.  
Referências: f. 171 a 174.

1. Web Semântica. 2. Linked Data. 3. Sensores. 4. Atuadores. I. de  
Azevedo Jacyntho, Mark Douglas, orient. II. Título.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
FLUMINENSE

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

TAMIRIS DE SOUSA RANGEL

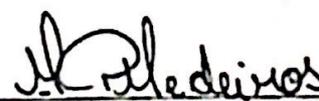
CATALOGAÇÃO SEMÂNTICA DE SENSORES E ATUADORES COM  
BASE NOS PRINCÍPIOS LINKED DATA

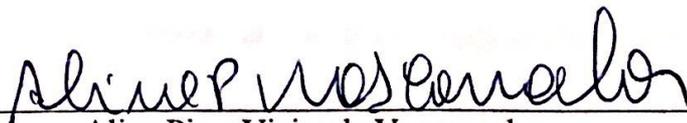
Dissertação apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à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

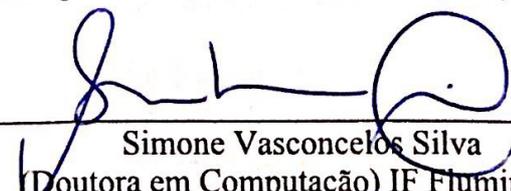
Aprovada em 06 de Junho de 2018

Banca Examinadora:

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

  
Adriana Pereira de Medeiros  
(Doutora em Informática) UFF

  
Aline Pires Vieira de Vasconcelos  
(Doutora em Engenharia de Sistemas e Computação) IF Fluminense

  
Simone Vasconcelos Silva  
(Doutora em Computação) IF Fluminense

## **AGRADECIMENTOS**

Ao professor Mark Douglas Jacyntho pelas contribuições valiosas que permitiram a realização desta dissertação.

Aos professores membros da banca pelas contribuições para melhoria deste trabalho.

A minha mãe, Ana, que mesmo não tendo tido acesso ao estudo que eu tive, sempre demonstrou sabedoria para me incentivar e guiar em todos os momentos da minha vida.

As minhas irmãs e amigos que direta ou indiretamente fizeram parte da minha formação.

Ao Instituto Federal Fluminense por oportunizar o acesso à educação gratuita e de qualidade.

“Educação é o que fica depois de ter esquecido o que foi aprendido na escola”.

“Education is what remains after one has forgotten what one has learned in school”.

Albert Einstein

## RESUMO

A necessidade de monitoração, coleta e análise de dados dos ambientes tem sido cada vez mais demandada. Isso porque a resolução de muitos problemas do cotidiano depende de informações das condições ambientais de diversas regiões ou sistemas. Assim sendo, a sociedade moderna tem estado cada vez mais permeada por dispositivos de sensoriamento. Em alguns casos, é possível que para resolver um problema de forma eficiente, uma aplicação necessite consumir informações de sensores alocados em regiões geograficamente distribuídas, nas quais a descoberta e o acesso aos dispositivos não sejam nada triviais. Este trabalho tem como objetivo, portanto, desenvolver uma aplicação Web semântica para catalogar sensores, geograficamente distribuídos, descrevendo suas características/especificações e serviços oferecidos. Além disso, dada sua estreita relação com sensores, a aplicação deve incluir também os atuadores. Com o advento da Web Semântica (Web 3.0), o ideal é que os dados desta aplicação possuam semântica explícita, formalmente definida por ontologias, seguindo os padrões e tecnologias definidos pelo consórcio W3C<sup>1</sup>, sendo, portanto, inteligíveis por humanos e, sobretudo, por agentes de software. Para tal, este trabalho apresenta um estudo sobre as tecnologias e padrões da Web semântica, seguido de seleção criteriosa de ontologias para, finalmente, desenvolver uma aplicação semântica que atenda o campo de sensores e atuadores. Nesse cenário, a proposta deste trabalho consiste no desenvolvimento de uma aplicação para catalogação semântica de sensores e atuadores na Web, com base nos princípios *Linked Data* enunciados por Sir Tim Berners-Lee, inventor da Web. Como resultado, tem-se uma base de dados ligados abertos e um portal semântico descrevendo os sensores e atuadores na Web, contendo informações inteligíveis por seres humanos e, sobretudo, por máquinas, facilitando, sobremaneira, a descoberta, a seleção e o uso combinado de sensores e atuadores instalados mundo afora. Exemplos realistas de uso do catálogo semântico proposto demonstram que o objetivo do trabalho foi alcançado, à medida que sensores e recursos relacionados puderam ser descritos (semanticamente) com expressividade significativa por meio do uso de diferentes ontologias. Além disso, as informações catalogadas puderam ser enriquecidas através de *links* semânticos com outras fontes de dados na Web, perfazendo a prática *Linked Data*.

**Palavras-chave:** Sensores, Atuadores, Web Semântica, *Linked Data*

---

<sup>1</sup> <https://www.w3.org>

## ABSTRACT

The need for monitoring, collection and analysis of environments' data has been increasingly demanded. The reason for this is that the resolution for many everyday problems depends on information about environmental conditions from different regions or systems. Therefore, modern society has been increasingly permeated by sensing devices. In some cases, in order to solve a problem in an efficient manner, applications may need to consume information from sensors in geographically distributed regions, in which discovery and access to devices are by no means trivial. Therefore, this work aims to develop a semantic Web application to catalog geographically distributed sensors and actuators, describing its features/specifications and services offered. In addition, given the close relationship to sensors, actuators should also be included. With the advent of the Semantic Web (Web 3.0), the ideal application would be the one that publishes its data with explicit semantics, formally defined by ontologies. This way, it is crucial that these types of applications follow the standards and technologies defined by the W3C consortium so that published data can be understood by humans and, mainly, by software agents. To this end, this work presents a research on the semantic Web technologies and its standards, followed by careful selection of ontologies, to finally develop a semantic application that meets the field of sensors and actuators. This research proposes the development of an application for semantic cataloging of sensors and actuators on the Web, based on the Linked Data principles enunciated by Sir Tim Berners-Lee, inventor of the Web. As outcome, we have an open linked database and a semantic portal describing sensors and actuators on the Web, containing information intelligible by humans and, above all, by machines, greatly facilitating the discovery, selection and combined use of these sensors and actuators installed around the world. Realistic examples using the proposed semantic catalog demonstrate that the objective of the work was achieved, as sensors and related resources could be described (semantically) with significant expressiveness through the use of different ontologies. In addition, the published information could be enriched through semantic links with other datasets on the Web, making the *Linked Data* practice.

**Keywords:** Sensors, Actuators, Semantic Web, Linked Data

## LISTA DE FIGURAS

Figura 1 - Exemplo de grafo RDF .....	20
Figura 2 - Exemplo de grafo RDF contendo nó em branco .....	20
Figura 3 - Exemplo de descrição de recurso na ontologia RDFS .....	23
Figura 4 - Relações entre classes das ontologias RDFS e OWL .....	24
Figura 5 - Restrição na propriedade <i>hasParent</i> .....	24
Figura 6 - Tripla gerada com uso da restrição na propriedade <i>owl:allValuesFrom</i> .....	25
Figura 7 - LOD Cloud Diagram, Fevereiro de 2017.....	27
Figura 8 - Exemplo de tripla RDF .....	28
Figura 9 - Exemplo de consulta SPARQL.....	28
Figura 10 - Exemplo de consulta SPARQL com mais de um resultado .....	29
Figura 11 - Principais etapas do trabalho.....	31
Figura 12 - Resultado de pesquisa na base de dados SCOPUS .....	32
Figura 13 - Aspectos adotados para descrição de sensores.....	33
Figura 14 - Funcionalidades do catálogo .....	34
Figura 15 - Macro-arquitetura da aplicação.....	46
Figura 16 - Compartilhamento de dados do catálogo proposto na Web.....	47
Figura 17 - Módulos da ontologia Semantic Sensor Network Ontology (SSN).....	49
Figura 18 - Tipos de sistemas na ontologia SSN .....	50
Figura 19 - Classes e propriedades da ontologia SSN na perspectiva de observação .....	51
Figura 20 - Classes e propriedades da ontologia SSN na perspectiva da atuação .....	52
Figura 21 - Subclasses de <i>ssn:Property</i> na ontologia SSN.....	52
Figura 22- Hierarquia de classes e subclasses da ontologia SSN .....	53
Figura 23 - Modelo para descrição de pessoas .....	54
Figura 24 - Modelo para descrição de organizações.....	56
Figura 25 - Modelo para descrição de dados de localização de organizações.....	57
Figura 26 - Tesouro de tipos de organização .....	58
Figura 27 - Modelo para descrição de base de dados .....	59
Figura 28 - Modelo para descrição de formatos de publicação .....	60
Figura 29 - Modelo para descrição de objetos de interesse .....	61
Figura 30 - Modelo para descrição de tipos de propriedades .....	62
Figura 31 - Modelo para descrição de propriedades associadas a objeto de interesse .....	64
Figura 32 - Modelo para descrição de procedimentos .....	65
Figura 33 - Modelo para descrição de parâmetros de entrada .....	66
Figura 34 - Modelo para descrição de parâmetros de saída.....	67
Figura 35 - Modelo para descrição de propriedades de sistema .....	67
Figura 36 - Modelo para descrição de propriedades de operação.....	68
Figura 37 - Modelo para descrição de propriedades de sobrevivência.....	69
Figura 38 - Modelo para descrição de modelo de sensor.....	70
Figura 39 - Restrição na propriedade <i>ssn-system:hasSystemCapability</i> .....	73
Figura 40 - Modelo para descrição das condições de funcionamento .....	74
Figura 41 - Restrição na propriedade <i>ssn-system:hasOperatingRange</i> .....	76
Figura 42 - Restrição na propriedade <i>ssn-system:hasSurvivalRange</i> .....	78
Figura 43 - Restrição na propriedade <i>sosa:observes</i> .....	79
Figura 44 - Restrição na propriedade <i>ssn:implements</i> .....	80
Figura 45 - Restrição nas propriedades <i>dcterms:creator</i> e <i>foaf:maker</i> .....	81
Figura 46 - Modelo para descrição de sensor .....	82
Figura 47 - Dados de localização de sensor.....	83
Figura 48 - Modelo para descrição de modelos de atuadores.....	85
Figura 49 - Restrição na propriedade <i>sosa:actsOnProperty</i> .....	86
Figura 50 - Modelo para descrição de atuadores .....	87
Figura 51 - Modelo para descrição de plataformas.....	89
Figura 52 - Modelo para descrição de implantação .....	90

Figura 53 - Implantação Web das Coisas .....	92
Figura 54 - Exemplo de criação de cadastro de pessoa .....	94
Figura 55 - Exemplo de edição de cadastro de pessoa.....	95
Figura 56 - Consulta SPARQL de pessoas .....	95
Figura 57 - Grafo RDF subjacente ao cadastro de uma nova pessoa.....	96
Figura 58 - Tela de criação de organização .....	97
Figura 59 - Tela de edição de organização .....	98
Figura 60- Grafo RDF subjacente ao cadastro de organização.....	99
Figura 61 - Propriedades de localização da organização catalogada.....	100
Figura 62 - Tela de criação de Base de dados de sensores .....	100
Figura 63 - Tela de edição de Bases de dados .....	101
Figura 64 - Grafo RDF subjacente ao cadastro de Bases de dados .....	102
Figura 65 - Grafo RDF subjacente ao cadastro de Bases de dados com foco na descrição arquivo que contém os dados de observação de sensores.....	103
Figura 66 - Tela de criação de um novo Objeto de Interesse.....	104
Figura 67 -Tela de edição de um Objeto de Interesse.....	104
Figura 68 - Grafo RDF do objeto de interesse “air” (ar) .....	105
Figura 69 - Tela de criação de um novo Tipo de propriedade .....	106
Figura 70 - Tela de edição de um Tipo de propriedade .....	106
Figura 71-RDF subjacente ao cadastro de um novo Tipo de propriedade.....	107
Figura 72- Tela de criação de uma nova Propriedade de Objeto de Interesse .....	108
Figura 73- Tela de edição de uma nova Propriedade de Objeto de Interesse .....	108
Figura 74 - Grafo RDF subjacente ao cadastro de propriedade de objeto de interesse “Temperature of the air” .....	109
Figura 75- Tela de criação de um novo Procedimento .....	110
Figura 76- Tela de edição de um novo Procedimento .....	110
Figura 77 - Grafo RDF de procedimento do modelo de sensor DHT22.....	111
Figura 78- Tela de criação de nova propriedade de sistema .....	112
Figura 79-Tela de edição de propriedade de sistema.....	113
Figura 80- Grafo RDF subjacente ao cadastro de propriedade de sistema .....	113
Figura 81- Tela de criação do modelo de sensor DHT22 .....	114
Figura 82- Parte da tela de edição do modelo de sensor DHT22.....	115
Figura 83- Tela de edição do modelo de sensor DHT22 com foco na capacidade de sistema .....	116
Figura 84- Parte da tela de edição do modelo de sensor DHT22 com foco na capacidade de sistema e condições normais de funcionamento .....	117
Figura 85- Parte da tela de edição do modelo de sensor DHT22 com foco na propriedade de sistema “Precision” .....	118
Figura 86-Tela de edição do modelo DHT22 com foco na propriedade de sistema “Hysteresis” .....	119
Figura 87- Tela de edição do modelo DHT22 com foco na faixa de operação .....	120
Figura 88- Tela de edição do modelo DHT22 com foco na condição de funcionamento para faixa de operação .....	121
Figura 89-Tela de edição do modelo DHT22 com foco na propriedade de operação “Operating Power Range” .....	122
Figura 90-Tela de edição do modelo de sensor DHT22 com foco na faixa de sobrevivência	123
Figura 91-Tela de edição de modelo de sensores com foco na propriedade de sobrevivência .....	124
Figura 92 - Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 .....	125
Figura 93-Parte do grafo RDF subjacente ao cadastro do modelo DHT22 com foco na capacidade de sistema e sua condição de temperatura.....	126
Figura 94- Parte do grafo RDF subjacente ao cadastro do modelo DHT22 com foco na capacidade de sistema e sua condição de umidade.....	127

Figura 95-Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de sistema “Precision” .....	128
Figura 96- Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de sistema “Hysteresis” .....	129
Figura 97- Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na faixa de operação e condição de temperatura .....	130
Figura 98- Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na faixa de operação e condição normal de umidade .....	131
Figura 99-Grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de operação “OperatingPowerRange” .....	132
Figura 100- Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na faixa de sobrevivência.....	133
Figura 101- Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de sobrevivência “System Lifetime” .....	134
Figura 102- Tela de criação de exemplar de sensor DHT22 .....	135
Figura 103- Tela de edição de exemplar de sensor DHT22.....	135
Figura 104 - Grafo RDF do exemplar do modelo de sensor DHT22.....	136
Figura 105-Propriedades de localização de exemplar do modelo DHT22 .....	137
Figura 106- Tela de criação de um novo modelo de atuador.....	137
Figura 107-Parte da tela de edição de modelo de atuador .....	138
Figura 108-Grafo RDF subjacente ao cadastro de modelo de atuador .....	139
Figura 109-Tela de criação de um novo atuador .....	140
Figura 110- Tela de edição de atuador.....	141
Figura 111-Grafo RDF subjacente ao cadastro de atuador.....	142
Figura 112- Grafo RDF subjacente ao cadastro de atuador com foco nas propriedades de localização.....	143
Figura 113- Tela de criação de plataforma .....	143
Figura 114- Tela de edição de plataforma .....	144
Figura 115- Grafo RDF subjacente ao cadastro da Plataforma “Raspberry Pi” .....	144
Figura 116- Tela de criação de implantação WoT.....	145
Figura 117- Tela de edição de implantação WoT.....	146
Figura 118- Grafo RDF subjacente ao cadastro de Implantação WoT .....	147
Figura 119 - Parte do grafo RDF subjacente ao cadastro do modelo de sensor IP68.....	149
Figura 120 - Parte da descrição da capacidade de sistema do modelo IP68.....	150
Figura 121 - Parte do grafo RDF da capacidade de sistema do modelo IP68 com foco na propriedade de sistema “Accuracy” .....	151
Figura 122 - Parte do grafo RDF da capacidade de sistema como foco na propriedade “Sensitivity” .....	152
Figura 123 - Parte do grafo RDF da capacidade de sistema como foco na propriedade “Resolution” para a propriedade “Temperatura do ar” .....	153
Figura 124 - Parte do grafo RDF da capacidade de sistema como foco na propriedade “Resolution” para a propriedade “Bateria do modelo IP68” .....	154
Figura 125 - Grafo RDF do objeto de interesse “IP68” .....	155
Figura 126 - Grafo RDF contendo a descrição das propriedades <i>quality:battery_of_ip68</i> e <i>quality-type:battery</i> .....	155
Figura 127 - Grafo RDF do objeto de interesse “IP68 sensor”.....	156
Figura 128 - Grafo RDF contendo a descrição das propriedades <i>quality:battery_of_ip68_sensor</i> e <i>quality-type:battery</i> .....	157
Figura 129 - Grafo RDF do exemplar do modelo IP68 .....	158
Figura 130 - Grafo RDF da implantação do sensor IP68.....	159
Figura 131 - Visão geral do grafo RDF do modelo de sensor PIR 555-28027.....	160
Figura 132 - Grafo RDF do modelo de sensor PIR (555-28027) com foco na propriedade de sistema “Sensitivity” .....	161
Figura 133 - Grafo RDF do modelo de sensor PIR (555-28027) com foco na propriedade de operação “OperatingPowerRange” .....	162

Figura 134 - Grafo RDF do exemplar do modelo PIR (555-28027).....	163
Figura 135 - Grafo RDF subjacente ao cadastro de pessoa no catálogo semântico em Turtle.....	175
Figura 136 - Grafo RDF subjacente ao cadastro de organização no catálogo semântico em Turtle.....	176
Figura 137 - Grafo RDF subjacente ao cadastro de base de dados no catálogo semântico em Turtle.....	177
Figura 138 - Grafo RDF subjacente ao cadastro do objeto de interesse “ar” no catálogo semântico em Turtle.....	178
Figura 139 - Grafo RDF subjacente ao cadastro de tipo de propriedade “Temperatura” no catálogo semântico em Turtle.....	179
Figura 140 - Grafo RDF subjacente ao cadastro de propriedade de objeto de interesse “Temperatura do ar” em Turtle.....	180
Figura 141 - Grafo RDF subjacente ao cadastro de propriedade de objeto de interesse “Umidade do ar” em Turtle.....	180
Figura 142 - Grafo RDF subjacente ao cadastro de procedimento do modelo de sensor DHT22 em Turtle.....	181
Figura 143 - Grafo RDF subjacente ao cadastro de propriedade de sistema “Histerese” em Turtle.....	182
Figura 144 - Grafo RDF subjacente ao cadastro do modelo de sensor DHT22 em Turtle (continua).....	183
Figura 145 - Grafo RDF subjacente ao cadastro do exemplar do modelo DHT22 em Turtle.....	185
Figura 146 - Grafo RDF subjacente ao cadastro do modelo de atuador em Turtle.....	186
Figura 147 - Grafo RDF subjacente ao cadastro de atuador em Turtle.....	187
Figura 148 - Grafo RDF subjacente ao cadastro de implantação WoT em Turtle.....	188
Figura 149- Grafo RDF subjacente ao cadastro de plataforma em Turtle.....	189
Figura 150 - Grafo RDF subjacente ao cadastro do modelo de sensor IP68 em Turtle (continua).....	190
Figura 151 - Grafo RDF subjacente ao cadastro do objeto de interesse “Modelo IP68” em Turtle.....	191
Figura 152 - Grafo RDF subjacente ao cadastro da propriedade “Bateria do modelo IP68” em Turtle.....	192
Figura 153 - Grafo RDF subjacente ao cadastro do objeto de interesse “Sensor IP68” em Turtle.....	192
Figura 154 - Grafo RDF subjacente ao cadastro da propriedade “Bateria do sensor IP68” em Turtle.....	193
Figura 155 - Grafo RDF subjacente ao cadastro do exemplar IP68 em Turtle.....	194
Figura 156 - Grafo RDF subjacente ao cadastro da implantação do sensor IP68 em Turtle.....	195
Figura 157 - Grafo RDF subjacente ao cadastro do modelo de sensor PIR em Turtle.....	196
Figura 158 - Grafo RDF subjacente ao cadastro do sensor PIR em Turtle.....	197

## LISTA DE TABELAS

Tabela 1 - Classes e propriedades da ontologia RDFS .....	23
Tabela 2 - Tipos de restrição em OWL.....	26
Tabela 3 - Número de publicações envolvendo as áreas de interesse entre 2009 e 2017 .....	33
Tabela 4 - Descrição dos recursos que podem ser cadastrados na plataforma .....	35
Tabela 5 - Ontologias adotadas para descrição de recursos.....	37
Tabela 6 - Tipos de recursos .....	38
Tabela 7 - Domínios e contradomínios das propriedades da ontologia DCAT utilizadas nos modelos.....	39
Tabela 8- Domínios e contradomínios das propriedades da ontologia Dublin Core utilizadas nos modelos .....	40
Tabela 9- Domínios e contradomínios das propriedades da ontologia FOAF utilizadas nos modelos .....	40
Tabela 10- Domínios e contradomínios das propriedades da ontologia Good Relations utilizadas nos modelos .....	41
Tabela 11- Domínios e contradomínios das propriedades da ontologia ORG utilizadas nos modelos.....	41
Tabela 12- Domínios e contradomínios das propriedades da ontologia OWL utilizadas nos modelos.....	41
Tabela 13- Domínios e contradomínios das propriedades da ontologia PROV utilizadas nos modelos.....	42
Tabela 14- Domínios e contradomínios das propriedades das ontologias RDF e RDFS utilizadas nos modelos .....	42
Tabela 15- Domínios e contradomínios das propriedades da ontologia Schema.Org utilizadas nos modelos .....	43
Tabela 16- Domínios e contradomínios das propriedades da ontologia SSN utilizadas nos modelos.....	44
Tabela 17- Domínios e contradomínios das propriedades da ontologia VCARD utilizadas nos modelos.....	45
Tabela 18 - Comparação entre os módulos SOSA, SSN e System Capability .....	50
Tabela 19 - Restrições e suas propriedades na classe <i>?SensorModel</i> .....	71
Tabela 20 - Restrições e suas propriedades (modelo de atuadores).....	86
Tabela 21 - Implantação Web das Coisas (WoT) .....	93
Tabela 22 - Definição dos prefixos.....	93
Tabela 23 - Dados utilizados na descrição do sensor IP68 Smart Sensor .....	148
Tabela 24- Dados utilizados na descrição do sensor PIR 555-28027.....	159

## LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface  
CRUD - Create Read Update Delete  
DB - DBpedia Ontology  
DCAT - Data Catalog Vocabulary  
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  
SKOS - Simple Knowledge Organization System  
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  
UML - Unified Modelling Language  
URI - Uniform Resource Identifier  
VCARD - VCard Ontology  
W3C - World Wide Web Consortium  
WoT - Web of Things  
XML - Extensible Markup Language

## SUMÁRIO

1. INTRODUÇÃO .....	15
1.1. Objetivo geral.....	16
1.2. Objetivos específicos .....	16
1.3. Justificativa .....	17
1.4. Contribuições .....	17
1.5. Metodologia .....	18
1.6. Estrutura do documento .....	18
2. FUNDAMENTAÇÃO TEÓRICA.....	19
2.1. Modelo de dados RDF .....	19
2.2. Ontologias .....	21
2.3. Princípios Linked Data .....	26
2.4. Linguagem SPARQL .....	28
2.5. Da Internet das Coisas à Web Semântica das Coisas .....	29
3. TRAJETÓRIA METODOLÓGICA .....	31
3.1. Revisão bibliográfica .....	31
3.2. Elicitação de requisitos .....	33
3.3. Modelo ontológico .....	36
3.4. Desenvolvimento do catálogo.....	45
3.5. Exemplos de uso do catálogo.....	47
4. ONTOLOGIA SEMANTIC SENSOR NETWORK ONTOLOGY - SSN .....	48
5. MODELO ONTOLÓGICO DO CATÁLOGO .....	54
5.1. Módulo Pessoa (Person) .....	54
5.2. Módulo Organização.....	56
5.3. Módulo Base de Dados (Dataset) .....	58
5.4. Módulo Objeto de Interesse (Feature Of Interest) .....	61
5.5. Módulo Tipo de Propriedade (Property Type).....	62
5.6. Módulo Propriedade de um Objeto de Interesse (Property Of a Feature Of Interest) .....	63
5.7. Módulo Procedimento (Procedure).....	65
5.8. Módulo Propriedade de Sistema (System Property) .....	67
5.9. Módulo Propriedade de operação (Operating Property) .....	68
5.10. Módulo Propriedade de sobrevivência (Survival Property).....	69
5.11. Módulo Modelo Sensor (Sensor Model).....	69
5.12. Módulo Sensor (Sensor) .....	82
5.13. Módulo Modelo de Atuador (Actuator Model).....	84
5.14. Módulo Atuador (Actuator) .....	87
5.15. Módulo Plataforma (Platform).....	88
5.16. Módulo Implantação (Deployment).....	90
6. DESENVOLVIMENTO DO PROTÓTIPO .....	92
6.1. Cenário realista de uso do catálogo .....	92
7. EXEMPLOS DE USO DO CATÁLOGO .....	148
8. TRABALHOS RELACIONADOS .....	164
9. RESULTADOS .....	167
10. CONSIDERAÇÕES FINAIS .....	169
11. TRABALHOS FUTUROS .....	170
REFERÊNCIAS BIBLIOGRÁFICAS.....	171
APÊNDICE.....	175

## 1. INTRODUÇÃO

A sociedade moderna tem demandado cada vez mais o desenvolvimento de sistemas inteligentes que possam tornar tarefas, a princípio, puramente manuais em tarefas automatizadas que tragam resultados rápidos e eficientes. Tais demandas estão atreladas à visão de mundo, na qual computadores minúsculos, com sensores e interfaces de comunicação, são incorporados na infraestrutura das cidades, em carros, estradas, aeroportos, shoppings, roupas, etc. Nesse caso, diversos tipos de dados de sensores são produzidos todos os dias - status de voos, clima, informações sobre chegada e saída de ônibus, trânsito, câmeras de shoppings, etc. Comumente, tais dados são gerados por diversos sensores geograficamente distribuídos e com tecnologias distintas. Com o propósito de facilitar a comunicação entre os diversos dispositivos e aplicações, independente de como estão fisicamente conectados e de suas tecnologias, tais elementos devem ser disponibilizados na Web. Desse modo, o protocolo e os padrões da Web que são altamente disseminados podem ser reutilizados e a interoperabilidade pode ser garantida. Uma vez dispostos na Web, os dados e serviços oferecidos pelos dispositivos tornam-se mais acessíveis a um maior número de desenvolvedores. Esta prática é conhecida como Web das Coisas e já vem mudando no qual a sociedade vive, trabalha e faz negócios (GUINARD & TRIFA, 2016).

Com o surgimento da Web das Coisas, além dos sensores, os atuadores tornam-se presenças marcantes na Web. Comumente, tais dispositivos colaboram uns com os outros para fornecer serviços de controle automático em sistemas geograficamente distribuídos. Nesse cenário, torna-se extremamente conveniente a disponibilização dos dados de tais dispositivos na Web, por meio de diferentes colaboradores, localizados em diferentes partes do mundo (KANSAL *et al.*, 2007). Para tal, é necessário que uma aplicação seja desenvolvida para que usuários possam cadastrar e catalogar sensores e atuadores, além dos serviços oferecidos pelos mesmos, ou ainda buscar por dispositivos de interesse. Com o advento da nova Web Semântica (Web 3.0), o ideal é que esta aplicação seja semântica. Nesse caso, os os dados publicados tornam-se inteligíveis tanto por seres humanos quanto por máquinas. Isso é possível ao estender a Web de simples documentos textuais para dados estruturados, com semântica explícita, definida em modelos formais de representação de conhecimento conhecidos por ontologias (BERNERS-LEE *et al.*, 2006).

Em 2006 Tim Berners Lee, fundador da Web, anunciou um conjunto de boas práticas para publicação, compartilhamento e interconexão de dados, informação e conhecimento na Web Semântica, conhecido como princípios *Linked Data*. O uso dos destes princípios

pressupõe a publicação não só de documentos, mas sim de dados sobre estes documentos, dados que estejam ligados a outros dados e que permitam uma estruturação do conteúdo disponibilizado na Web (FREITAS & JACYNTHO, 2016). Em direção a este nível de maturidade, este trabalho propõe a criação de aplicação para catalogação semântica de sensores e atuadores na Web. Uma aplicação semântica, em consonância com os princípios *Linked Data*, na qual humanos e máquinas compreendam o conteúdo disponibilizado, permitindo assim, que agentes de software autônomos auxiliem na busca, seleção e combinação de sensores e atuadores para os mais diversos tipos de aplicação.

### 1.1. Objetivo geral

Este trabalho tem como objetivo central apoiar a catalogação semântica de sensores e atuadores geograficamente distribuídos, de forma a permitir a descrição das características, especificações e tipos de serviços oferecidos e o compartilhamento de tais informações na Web, fazendo uso das novas tecnologias e padrões da Web Semântica definidos pelo W3C. Para isto, este trabalho propõe um modelo ontológico para descrição de sensores e atuadores e desenvolvimento de aplicação, baseada no modelo proposto, para catalogação semântica de sensores e atuadores. Nesta aplicação, dados são inteligíveis tanto por seres humanos, quanto por máquinas, bem como ligados a dados correlacionados de outras bases de dados semânticas disponíveis na Web, seguindo as diretrizes de publicação conhecidas como princípios *Linked Data* (BERNERS-LEE *et al.*, 2006).

**Delimitação da pesquisa:** A aplicação proposta permite a catalogação das características dos sensores, sem se preocupar com os dados de observação que fica a cargo das aplicações cliente que fazem uso do catálogo. A definição de perfis de usuários para catalogação dados de sensores e atuadores não está prevista nesta fase do trabalho, assim como a validação da aplicação com potenciais usuários. Além disso, a descrição formal semântica de interface de uso (protocolo de comunicação, padrões de interação, tipos de dados, formatos de arquivo, etc.) tanto de sensores quanto de atuadores, não faz parte do escopo deste trabalho, sendo, portanto, o próximo passo, na continuação prevista para este trabalho.

### 1.2. Objetivos específicos

Além do desenvolvimento da aplicação para atender a demanda de sensores, surgem também outros objetivos específicos, a saber: ratificar a eficiência da Web Semântica, especialmente o conceito *Linked Data* no contexto de sensores e atuadores, propor um conjunto de ontologias para descrição de sensores e atuadores, por meio de seleção criteriosa de

ontologias (vocabulários) *Linked Data* de referência, apresentar exemplos de uso que exemplifiquem e corroborem a aplicabilidade da aplicação Web de catalogação de sensores e atuadores.

### 1.3. Justificativa

Segundo Barnaghi *et al.* (2011) as observações e os sensores que as obtêm estão no centro da ciência empírica. Além disso, tais dispositivos são utilizados em diversas aplicações do cotidiano, tais como serviços de meteorologia, monitoramento ambiental, segurança e vigilância, etc. Portanto, não há dúvida de que sensores são ricas fontes de informação. Segundo Patni *et al.* (2010), tornar dados de sensores abertos e acessíveis implica em maior oportunidade de utilização dos dispositivos. Por exemplo, dados de sensores podem ser muito úteis para pesquisa e análises. Além disso, o acesso a esses conjuntos de dados pode facilitar a melhoria dos desempenhos dos processos, padronização e a comparação de tecnologias relacionadas a sensores. Assim sendo, se cuidadosamente combinados, dados de sensores podem ser fundamentais no processo de tomada de decisão.

Para que um sensor seja utilizado, antes de tudo, ele precisa ser encontrado. Em outras palavras, os sensores precisam ser devidamente catalogados e divulgados. Por razões óbvias, a Web se apresenta como uma alternativa para divulgação destes sensores. Nesse caso, Barnaghi & Presser (2010) ressalta a necessidade de construir serviços e aplicações que atuem como intermediários para captura, entrega e apresentação de dados dinâmicos do mundo real para aplicações de consumo e usuários na Web. Com o movimento da nova Web Semântica, é possível catalogar estes sensores, de forma semiestruturada compreensível por máquina, permitindo que agentes de software, de forma autônoma, selecionem e combinem os sensores, inclusive incorporando novos sensores que, por ventura, surjam no futuro. Além disso, *links* semânticos podem servir como um meio para interligar dados do sensor com fontes externas na Web, permitindo a integração de dados de várias fontes e a descoberta de novos dados em contextos históricos, temporais e espaciais (BARNAGHI *et al.*, 2011).

Com o surgimento da Web das Coisas, os atuadores e os dados produzidos por tais dispositivos também se tornam parte da Web. Nesse caso, aplicações para catalogar semanticamente, além dos sensores, os atuadores, atendem de forma mais ampla as novas soluções tecnológicas da sociedade moderna (HALLER *et al.*, 2017).

### 1.4. Contribuições

As principais contribuições deste trabalho incluem:

- Modelo ontológico para descrição de sensores e atuadores (seleção de ontologias *Linked Data*);
- Geração de base de dados RDF contendo descrição de metadados de sensores e atuadores na Web por meio de aplicação semântica desenvolvida;
- Disponibilização de dados de sensores na nuvem LOD permitindo buscas e análises dos dados de sensores e atuadores;
- Extensão da ontologia *Semantic Sensor Network Ontology* - SSN;

### **1.5. Metodologia**

A metodologia deste trabalho está dividida, basicamente, em cinco etapas: revisão bibliográfica (estudo das novas tecnologias e padrões), elicitação de requisitos, modelo ontológico, desenvolvimento da aplicação e exemplos realistas de uso do catálogo. A etapa de desenvolvimento compreende a seleção da plataforma *Linked Data* de desenvolvimento, experimentação da plataforma escolhida e desenvolvimento da aplicação - o desenvolvimento é demonstrado a partir de cenário de uso realista da aplicação. Por fim, exemplos de uso realistas do catálogo são apresentados para corroborar a eficácia da aplicação.

### **1.6. Estrutura do documento**

A estrutura deste documento está organizada como segue: no Capítulo 2, os fundamentos da Web Semântica e *Linked Data* são brevemente apresentados. No Capítulo 3, a trajetória metodológica utilizada para o desenvolvimento do trabalho é apresentada. O Capítulo 4 apresenta uma breve discussão sobre a ontologia *Semantic Sensor Network Ontology* - SSN. O Capítulo 5 inclui descrição detalhada do modelo ontológico proposto para catalogação semântica de sensores e atuadores. O Capítulo 6 discorre sobre o desenvolvimento da aplicação e apresenta cenário realista para demonstrar a aplicabilidade do catálogo. Exemplos de uso realistas são apresentados no Capítulo 7 para corroborar a eficácia da aplicação. O Capítulo 8 apresenta trabalhos relacionados ao tema, como resultado da pesquisa bibliográfica realizada. O Capítulo 9 discorre sobre alguns resultados obtidos e, finalmente, os Capítulos 10 e 11 concluem este documento com as considerações finais e trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

A maior parte do conteúdo publicado na Web convencional tem como foco a compreensão humana. Nesse caso, agentes de softwares processam informações disponíveis na Web sem compreender o significado de tais conteúdos. Para que seres humanos e máquinas possam entender e interpretar conteúdos publicados na Web, uma nova Web está sendo proposta: a Web Semântica ou Web de dados (BERNERS-LEE *et al.*, 2001).

### 2.1. Modelo de dados RDF

Esta nova Web Semântica pode ser entendida como uma extensão da Web atual, onde cada recurso ou entidade (por exemplo, uma pessoa, um lugar, um produto, enfim, qualquer entidade do mundo real) é identificado por um endereço Web, um *Uniform Resource Identifier* (URI), e ao acessar este URI, é retornado um arquivo, contendo uma representação (descrição) deste recurso, numa linguagem estruturada padrão compreensível por máquina (NEVES & JACYNTHO, 2014).

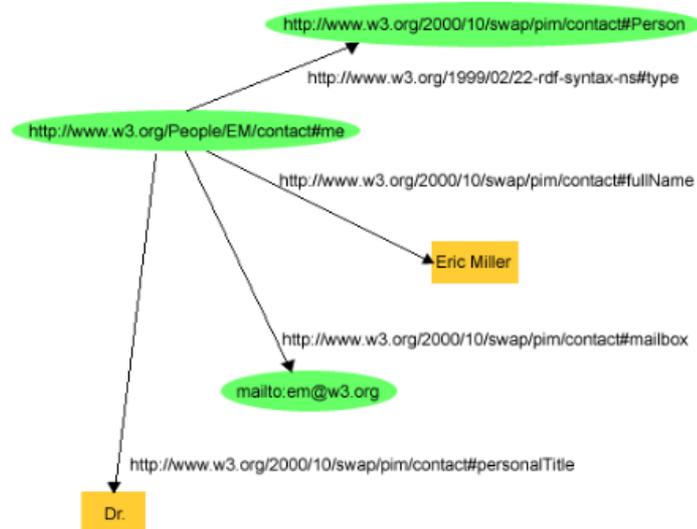
Para que a Web Semântica funcione, padrões específicos para publicação de informações devem ser respeitados – o conteúdo deve ser organizado segundo um modelo de dados padrão (RDF - *Resource Description Framework*), o conteúdo deve ser publicado usando sintaxes padrão (Turtle, RDF XML, etc.) e, por fim, o conteúdo precisa de semântica explícita (ontologias).

O modelo de dados RDF permite que recursos sejam descritos na Web de maneira simples e estruturada. Trata-se de um modelo baseado em grafo, onde os recursos são descritos por meio de triplas contendo três elementos: sujeito, predicado e objeto. O sujeito pode ser identificado como o recurso principal, aquele que se deseja descrever, o predicado consiste em uma propriedade do sujeito e o objeto é o valor resultante da relação do sujeito e sua propriedade (o valor do objeto pode ser um literal ou outro recurso). Neste modelo, cada recurso e propriedade recebe uma identificação única (URI) (MANOLA & MILLER, 2004).

A Figura 1 apresenta um exemplo de grafo RDF<sup>2</sup> que descreve uma pessoa identificada por “<http://www.w3.org/People/EM/contact#me>”, cujo nome é “Eric Miller”, título “Dr.” e o endereço de e-mail “[em@w3.org](mailto:em@w3.org)”.

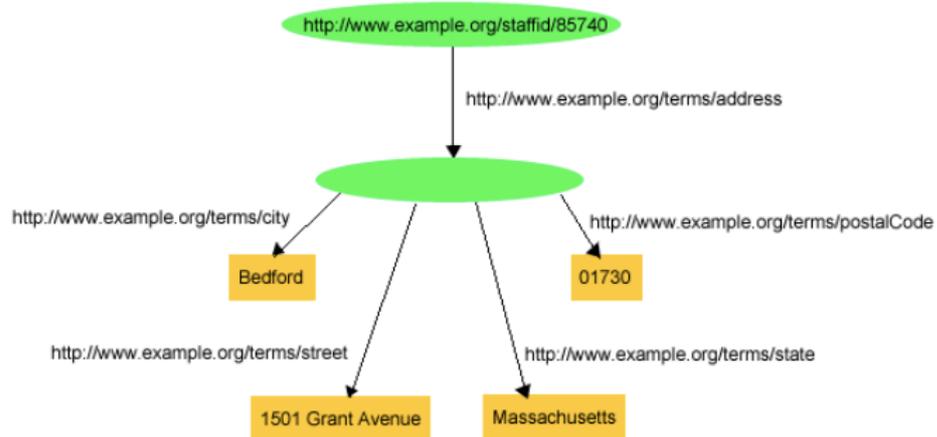
---

<sup>2</sup> Por convenção, na notação gráfica, elipses representam recursos e retângulos representam valores literais.



**Figura 1** - Exemplo de grafo RDF  
Fonte: Manola & Miller (2004)

Na Figura 1, a propriedade *rdf:type* é utilizada para definir a classe a qual o recurso pertence. Por definição, objetos podem conter URIs ou um literal como valor. No entanto, em alguns casos, pode ser útil descrever um recurso sem se preocupar com seu identificador. Recursos sem um URI que os especifica podem ser representados em RDF como nós em branco (blank nodes) ou recursos anônimos. Recursos anônimos podem aparecer como sujeitos ou objetos em uma tripla fazendo referência a recurso sem “nome” explícito por um URI. A Figura 2 apresenta um exemplo de tripla RDF contendo recurso anônimo.



**Figura 2** - Exemplo de grafo RDF contendo nó em branco  
Fonte: Manola & Miller (2004)

Na Figura 2, o recurso identificado pelo URI “*http://www.example.org/staffid/85470*” possui um dado endereço. Para descrever tal informação, relaciona-se o recurso “*http://www.example.org/staffid/85470*” com nó em branco por meio da propriedade “*http://www.example.org/terms/address*”. O nó em branco em questão possui diferentes

atributos que o qualificam, tais como: logradouro, código postal, cidade e estado. Em conjunto, estes atributos descrevem o endereço do recurso “<http://www.example.org/staffid/85470>”.

Em essência, um modelo de dados é apenas uma maneira de ver os dados. O modelo relacional já bem estabelecido visualiza os dados por meio de relações e tuplas. O modelo de grafo RDF, baseado em triplas, é uma representação natural para vários tipos de aplicações (por exemplo, Facebook, Twitter, sistemas de recomendação, etc.), onde as entidades estão fortemente ligadas entre si. Além disso, se a aplicação tem alta heterogeneidade no seu esquema de dados e necessidade frequente de adaptação, o modelo RDF é bem mais conveniente. Os bancos de dados RDF simplificam o desenvolvimento aplicações Web semânticas e também se alinham muito bem com diversos algoritmos e técnicas estatísticas desenvolvidas para grafos (JACYNTHO & SCHWABE, 2016).

Por se tratar de um modelo abstrato de dados em grafo, para que estes grafos sejam efetivamente publicados na Web, é necessário definir sintaxes concretas padrão para publicar arquivos RDF. Para tal propósito, foram criadas várias sintaxes, tais como RDF/XML<sup>3</sup>, Turtle<sup>4</sup>, JSON-LD<sup>5</sup>, etc. (NEVES & JACYNTHO, 2014). Há ainda outras sintaxes, cujo uso tem se destacado na Web, tal como a sintaxe RDFa<sup>6</sup>. RDFa é uma forma padrão de embutir metadados RDF em documentos HTML. As máquinas de busca (Google, Yahoo, etc.) estão, cada vez mais, fazendo uso destes metadados para recuperar as informações de forma mais precisa.

## 2.2. Ontologias

A semântica explícita das triplas RDF é dada por meio das ontologias, que consistem em vocabulários que definem formalmente os termos utilizados para descrever uma área de conhecimento. Estes vocabulários apresentam a relação entre os diversos conceitos e carregam um conjunto de axiomas (regras) para que inferências possam ser realizadas - inúmeros relacionamentos entre diversas entidades podem ser expressos por meio da definição de classes e propriedades. Ao acessar um conteúdo na Web, as máquinas passam a compreender as informações manipuladas ao navegar pelos links das ontologias vinculadas ao conteúdo publicado. Desse modo, o uso de ontologias na Web Semântica é fundamental na integração de dados, uma vez que define um vocabulário comum para uma dada área de conhecimento a ser compartilhado, eliminando assim ambiguidades nos termos usados nas diferentes fontes de

---

<sup>3</sup> <https://www.w3.org/TR/rdf-syntax-grammar/>

<sup>4</sup> <https://www.w3.org/TR/turtle/>

<sup>5</sup> <http://json-ld.org/>

<sup>6</sup> <https://rdfa.info/>

dados, bem como viabilizando a inferência automática de mais conhecimento, por meio dos axiomas (HEFLIN, 2004).

Segundo Morais & Jacyntho (2015), em termos práticos, ontologias representam um determinado domínio de conhecimento e consistem em: classes, que representam os principais conceitos do domínio e agrupam os indivíduos com características comuns; propriedades que representam relacionamentos e atributos das classes; axiomas, que são regras de restrição sobre o domínio e instâncias das classes (instâncias consistem em indivíduos ou exemplares que pertencem a uma classe) (apud NOY & MCGUINNESS, 2001).

Os tópicos a seguir abordam maiores detalhes sobre ontologias importantes para semântica dos dados na Web Semântica, tais como os vocabulários rdf, rdfs e owl.

### 2.2.1. Ontologia Resource Description Framework - RDF

Toda e qualquer informação publicada na Web Semântica deve ser estruturada em consonância com o modelo de dados RDF. O vocabulário ou ontologia RDF define os termos primitivos do modelo RDF essenciais para a descrição de recursos quaisquer. Este vocabulário serve com base para a construção de outros vocabulários.

Como extensão do vocabulário RDF, duas linguagens de criação de ontologias (metaontologias) foram definidas pelo W3C, a saber: RDF Schema (RDFS) (BRICKLEY & GUHA, 2004) e Web Ontology Language (OWL) (DEAN & SCHREIBER, 2004).

### 2.2.2. Metaontologia RDF Schema - RDFS

A metaontologia RDF Schema (RDFS) consiste em vocabulário que contém recursos para descrever outros recursos RDF. Trata-se de uma extensão do vocabulário RDF que prevê mecanismos para descrever grupos (classes) de recursos relacionados e as relações (propriedades) entre estes recursos. Diversas características de recursos podem ser descritas por meio do uso do vocabulário RDFS, tais como domínio (*rdfs:domain*)<sup>7</sup> e contra-domínio (*rdfs:range*)<sup>8</sup> de propriedades (BRICKLEY & GUHA, 2004). A Tabela 1 apresenta as classes e propriedades da metaontologia RDFS.

<sup>7</sup> *P rdfs:domain*  $\{C_1, C_2, \dots, C_n\}$  indica que qualquer recurso que tenha a propriedade *P* é instância de  $\{C_1 \cap C_2 \cap \dots \cap C_n\}$ .

<sup>8</sup> *P rdfs:range*  $\{C_1, C_2, \dots, C_n\}$  indica que os valores da propriedade *P* são instâncias de  $\{C_1 \cap C_2 \cap \dots \cap C_n\}$ .

**Tabela 1** - Classes e propriedades da ontologia RDFS

Classes	<i>rdfs:Class</i>
	<i>rdfs:Resource</i>
	<i>rdfs:Datatype</i>
	<i>rdfs:Container</i>
Propriedades	<i>rdfs:comment</i>
	<i>rdfs:ContainerMembershipProperty</i>
	<i>rdfs:domain</i>
	<i>rdfs:isDefinedBy</i>
	<i>rdfs:label</i>
	<i>rdfs:literal</i>
	<i>rdfs:member</i>
	<i>rdfs:range</i>
	<i>rdfs:seeAlso</i>
	<i>rdfs:subClassOf</i>

Fonte: Elaboração própria

Por definição, qualquer “coisa” pode ser considerada um recurso, desde que possua um URI que o defina. Assim, propriedades e classes das ontologias também são consideradas recursos. A título de exemplo, um trecho da ontologia RDFS é apresentado na Figura 3, com o propósito de demonstrar a descrição de um dos seus recursos.

```

rdfs:seeAlso a rdf:Property ;
  rdfs:label "seeAlso" ;
  rdfs:comment "Further information about the subject resource." ;
  rdfs:domain rdfs:Resource ;
  rdfs:isDefinedBy rdfs: ;
  rdfs:range rdfs:Resource .

```

**Figura 3** - Exemplo de descrição de recurso na ontologia RDFS

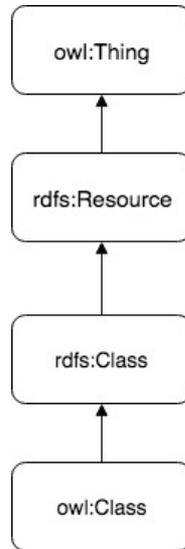
Fonte: Brickley & Guha (2004)

Na Figura 3, o recurso *rdfs:seeAlso* é descrito fazendo uso de termos que têm como origem a própria ontologia RDFS. Maiores informações sobre a ontologia RDFS podem ser encontradas em Brickley & Guha (2004).

### 2.2.3. MetaOntologia Web Ontology Language – OWL

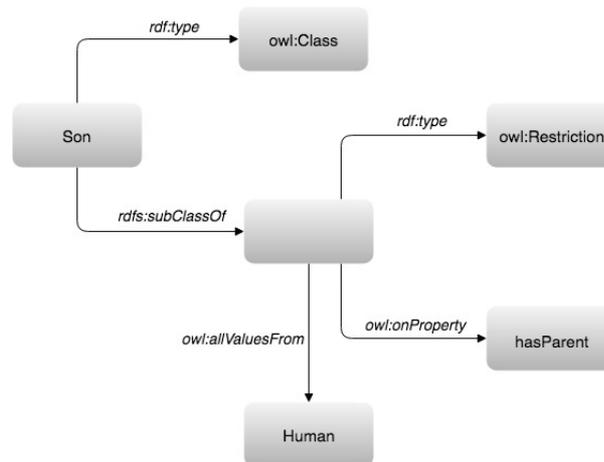
A metaontologia *Web Ontology Language* (OWL) consiste em vocabulário utilizado para representar informações complexas sobre coisas, conjunto de coisas e as relações entre coisas. Assim como a ontologia RDF Schema, trata-se de uma linguagem baseada em lógica computacional. Nesse caso, agentes de software podem processar as informações estruturadas em OWL a fim de verificar a consistência do conhecimento ou tornar explícito um conhecimento implícito (inferência). Por consistir em vocabulário RDF, pode ser usado em combinação com RDF Schema. Nesse caso, todos os termos criados em RDFS continuam

válidos em OWL (DEAN & SCHREIBER, 2004). A Figura 4 apresenta relações entre classes das ontologias RDFS e OWL.



**Figura 4** - Relações entre classes das ontologias RDFS e OWL  
Fonte: Elaboração própria

Em OWL, classes são definidas como *owl:Class*. Tais classes, são subclasses de *rdfs:Class*, *rdfs:Resource* e *owl:Thing*. Portanto, ao definir uma dada classe como *owl:Class* tem-se uma maior abrangência em termos de semântica. Além de *owl:Class*, outra classe importante do vocabulário OWL é a classe *owl:Restriction*. Classes do tipo *owl:Restriction* podem incorporar restrições para diferentes propriedades. A Figura 5 apresenta um exemplo de uso de restrição *owl:Restriction*.



**Figura 5** - Restrição na propriedade *hasParent*  
Fonte: Elaboração própria

Na Figura 5, a classe *owl:Restriction* é utilizada para definir o tipo do contra-domínio (range) da propriedade “*hasParent*”, no contexto da classe “*Son*”. O uso da propriedade *owl:allValuesFrom* (restrição universal) implica que todos os valores da propriedade

“*hasParent*” sejam do mesmo tipo, ou seja, consistam em instâncias da classe “*Human*”. Tal restrição vale para todas as instâncias da classe “*Son*”, tendo em vista que a classe “*Son*” é subclasse da restrição em questão. Além disso, o termo “*owl:allValuesFrom*” é coloquialmente simbolizado pelo advérbio “*only*” (“somente”). A Figura 6 apresenta, envolto em moldura pontilhada, um exemplo de tripla inferida, a partir da restrição universal apresentada na Figura 5.

James	<i>rdf:type</i>	Son
James	<i>hasParent</i>	Nick
Nick	<i>rdf:type</i>	Human

**Figura 6** - Tripla gerada com uso da restrição na propriedade *owl:allValuesFrom*

Fonte: Elaboração própria

Na Figura 6, “James” é uma instância da classe “*Son*”. Além disso, “James” se relaciona com o objeto “Nick” por meio da propriedade “*hasParent*”. Tendo em vista que James é uma instância da classe “*Son*” e tal classe é subclasse de uma dada restrição, onde o tipo de *range* da propriedade “*hasParent*” é definido como “*Human*”, então, pode-se concluir que “Nick” é uma instância da classe “*Human*”. Em outras palavras, é possível inferir que o objeto Nick é um ser humano, ainda que este conhecimento não tenha sido explicitado. O Quadro 1 apresenta o esquema da Figura 5 em formato RDF, na sintaxe Turtle.

**Quadro 1** - Representação de restrição em RDF em Turtle

```

:Son
  a owl:Class ;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:onProperty :hasParent ;
      owl:allValuesFrom <http://www.example.org/Human>
    ].

```

Fonte: Elaboração própria

Além disso, por relacionar recursos, ou seja, uma dada pessoa à outra pessoa, a propriedade *hasParent* consiste em propriedade de objeto (*owl:ObjectProperty*). Tal classificação tem como origem a ontologia OWL. Em OWL, propriedades são divididas em duas categorias: Propriedades de Objetos (*owl:ObjectProperty*) e Propriedades de Dados (*owl:DatatypeProperty*). *Datatype Properties* são utilizadas para relacionar um recurso a um valor literal RDF ou um tipo primitivo, como os predefinido pelo XML Schema, por exemplo, decimal (*xsd:decimal*), string (*xsd:string*), data (*xsd:date*), entre outros. Já as *Object Properties* descrevem relacionamento entre recursos (MORAIS & JACYNTHO, 2015).

A ontologia OWL apresenta outros tipos de restrições, além de *owl:allValuesFrom*, que podem ser utilizadas para enriquecer informações. Tais restrições são classificadas como restrição de valores e de cardinalidade dependendo da propriedade utilizada. A Tabela 2 apresenta os tipos de restrição definidas em OWL.

**Tabela 2 - Tipos de restrição em OWL**

<i>Tipo de Restrição</i>	<i>Propriedade</i>	<i>Descrição</i>
Restrição de valores	<i>owl:allValuesFrom</i>	Todos os valores da propriedade tem que ser de um certo tipo
	<i>owl:someValuesFrom</i>	Pelo menos um valor da propriedade tem que ser de um certo tipo
	<i>owl:hasValue</i>	Pelo menos um dos valores da propriedade é um certo valor
Restrição de cardinalidade	<i>owl:cardinality</i>	A propriedade tem que ter exatamente “n” valores
	<i>owl:minCardinality</i>	A propriedade tem que ter pelo menos “n” valores
	<i>owl:maxCardinality</i>	A propriedade tem que ter no máximo “n” valores

Fonte: Elaboração própria

A Tabela 2 apresenta alguns dos diversos axiomas existentes no vocabulário OWL. Este vocabulário vem passando por revisões desde a publicação de sua primeira versão em 2004. A versão atual de OWL, conhecida como “OWL 2”, foi publicada em 2009 e caracteriza-se como sendo uma extensão de OWL. Em sua nova versão, todos os elementos de OWL são aproveitados e novos elementos foram adicionados. Informações adicionais sobre OWL 2 podem ser encontradas em Bao *et al.* (2012).

### 2.3. Princípios Linked Data

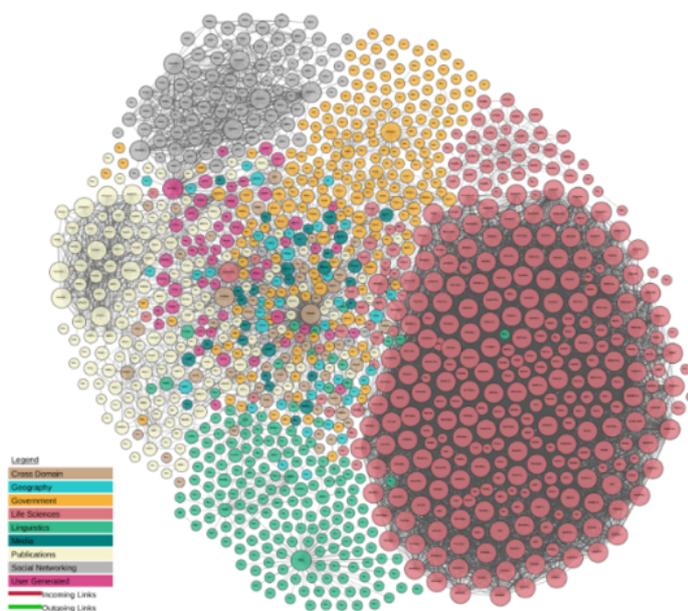
De maneira geral, o propósito da Web semântica ou Web de Dados é transformar a Web atual em um grande banco de dados global, onde qualquer indivíduo ou sistema possa acessar as informações disponíveis nela. Para tal, além de disponibilizar os dados em um formato padrão, acessível e gerenciável por ferramentas da Web Semântica, é necessário, ainda, estabelecer relações (links RDF) entre os recursos, das diferentes fontes de dados geograficamente distribuídas, para criar uma Web de Dados Ligados (caso contrário, teríamos coleções isoladas de conjuntos de dados). Assim, ao reutilizar informação de terceiros na Web de dados, é necessário que o relacionamento entre os diversos recursos seja bem definido. Por exemplo, ao descrever um cantor famoso em uma aplicação Web é possível relacionar o recurso cantor a sua cidade de origem. Caso as informações do recurso cidade se encontrem em outras fontes de dados na Web é necessário que o relacionamento entre tais recursos seja definido por uma propriedade pertencente a uma ontologia que englobe o contexto localização de pessoas.

Esta prática de relacionar e reusar dados na Web por meio de links semânticos é conhecida como *Linked Data* ou Dados ligados (BIZER *et al.*, 2009). Em Berners-Lee *et al.* (2006), foram elencados os quatro princípios *Linked Data* para publicação de dados ligados na Web. São eles:

1. Use URIs para nomear os recursos;
2. Use URIs HTTP de forma que estes recursos possam ser acessados na Web (por pessoas ou agentes de software);
3. Quando um URI for acessado, retorne alguma informação útil, usando os padrões (RDF, SPARQL);
4. Inclua links para outros URIs, de modo que se possa navegar para outros recursos na Web e obter mais informações (*Linked Data Mashup*).

À medida que novas aplicações disponibilizam seus conteúdos seguindo princípios *Linked Data*, esta nova Web se torna cada vez mais interoperável. Além disso, por estarem dentro dos padrões, estas fontes de dados ganham espaço no *Linked Open Data Cloud Diagram* ou *LOD* (ABELE & McCRAE, 2017). Trata-se de um canal na Web onde as fontes de dados *Linked Data* são listadas em uma espécie de diagrama. Quanto mais reutilizada por outras aplicações a fonte de dados for, maior será seu destaque no diagrama.

A Figura 7 apresenta a topologia da Web de Dados Ligados, onde várias fontes de dados (círculos) são interligadas por triplas RDF.



**Figura 7** - LOD Cloud Diagram<sup>9</sup>, Fevereiro de 2017  
Fonte: Abele & McCrae (2017)

<sup>9</sup> <http://lod-cloud.net/>

A DBpedia<sup>10</sup>, que é uma versão *Linked Data* da famigerada Wikipedia, é um exemplo de fonte de dados que segue os padrões da Web de Dados Ligados, ocupando localização centralizada e tamanho considerável na nuvem LOD, o que significa que tem sido amplamente reutilizada por outras aplicações. As informações da DBpedia e de outras aplicações que fazem parte da LOD podem ser facilmente consultadas com o uso da linguagem de consulta SPARQL.

## 2.4. Linguagem SPARQL

SPARQL *Protocol and RDF Query Language* consiste em uma linguagem de consulta e protocolo de acesso a dados estruturados em RDF desenvolvida pelo W3C. A linguagem SPARQL permite que as informações de um grafo RDF sejam manipuladas à medida que pré-requisitos para consultas são definidos. Por exemplo, é possível buscar por triplas RDF de interesse por meio da definição de propriedades específicas, do tipo de recurso, de restrições ou filtros em relação aos padrões dos grafos que se deseja obter como resultado da consulta (PRUD'HOMMEAUX & SEABORNE, 2008).

De forma similar ao modelo de dados RDF, a linguagem SPARQL é baseada no conceito de *triple pattern*, ou seja, é escrita como sujeito, predicado e objeto. No entanto, na linguagem SPARQL cada um destes elementos pode ser caracterizado como uma variável que irá armazenar o resultado de uma dada consulta. A Figura 8 apresenta um exemplo de tripla RDF que pode ser utilizada como base de uma consulta SPARQL.

<code>http://www.example.com/person245</code>	<code>foaf:name</code>	Ana
---	------------------------	-----

**Figura 8** - Exemplo de tripla RDF

Fonte: Elaboração própria

Para que o valor da propriedade *foaf:name* armazenado no grafo RDF apresentado na Figura 8 seja retornado como resultado de uma busca SPARQL, deve-se fazer uso das cláusulas “*Select*” e “*Where*” da linguagem SPARQL. A Figura 9 apresenta a estrutura de uma consulta SPARQL.

```

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

```

**Figura 9** - Exemplo de consulta SPARQL

Fonte: Elaboração própria

<sup>10</sup> <http://wiki.dbpedia.org/>

Na Figura 9, caso a consulta seja aplicada à base de dados que contenha a tripla RDF apresentada na Figura 8, tal consulta retornará como resultado o valor “Ana”, que será armazenado na variável *?name*. A variável *?Person*, por sua vez, armazenará o valor “*http://www.example.com/person245*”. A Figura 10 apresenta exemplo de consulta SPARQL com mais de um resultado.

**Dados:**

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Johnny Lee Outlaw" .
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy" .
_:b foaf:mbox <mailto:peter@example.org> .
_:c foaf:mbox <mailto:carol@example.org> .
```

**Consulta SPARQL:**

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

**Resultado da consulta:**

name	mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

**Figura 10** - Exemplo de consulta SPARQL com mais de um resultado

Fonte: Harris, Seaborne & Prud'hommeaux (2013)

Na Figura 10, a consulta SPARQL tem como objetivo retornar os valores nas propriedades *foaf:name* e *foaf:mbox*. Como a base de dados possui mais de um recurso descritos com tais propriedades, o resultado da consulta retorna mais de um valor. Além das cláusulas “*Select*” e “*Where*” apresentadas na Figura 10, a linguagem SPARQL inclui uma vasta gama de estruturas para construção desde consultas simples a consultas complexas. Maiores detalhes sobre a linguagem SPARQL podem ser encontrados em Prud'Hommeaux & Seaborne (2008).

## 2.5. Da Internet das Coisas à Web Semântica das Coisas

A Internet das Coisas (IoT) consiste em um novo paradigma que tem ganhado espaço no cenário da tecnologia sem fio de telecomunicações. A ideia básica deste conceito é a presença massiva de uma variedade de coisas ou objetos no cotidiano da sociedade moderna, tais como etiquetas de identificação por radiofrequência (RFID), sensores, atuadores, telefones celulares, etc. que, através de esquemas de endereçamento único, são capazes de interagir uns

com os outros e cooperar com seus vizinhos para alcançar objetivos comuns (GIUSTO *et al.*, 2010).

Para que objetos físicos de sistemas IoT possam ser descobertos, monitorados, controlados ou acessados, uma abordagem estruturada e processável por máquina para fornecer serviços do mundo real é necessária para tornar objetos físicos heterogêneos acessíveis em larga escala e integrá-los ao mundo digital. Dentro deste contexto, De *et al.* (2012) ressaltam o uso da Web na criação de infraestruturas dinâmicas para promover a integração dos dados de dispositivos físicos e fornecer acesso unificado a esses dados em níveis de serviço e de aplicação. Assim, a Web torna-se uma ferramenta importante para viabilização dos sistemas IoT, pois permite o desenvolvimento de aplicações interativas e inovadoras que mesclam os mundos físico e digital. Com a introdução da Web, um novo termo para Internet das Coisas é criado: A Web das Coisas.

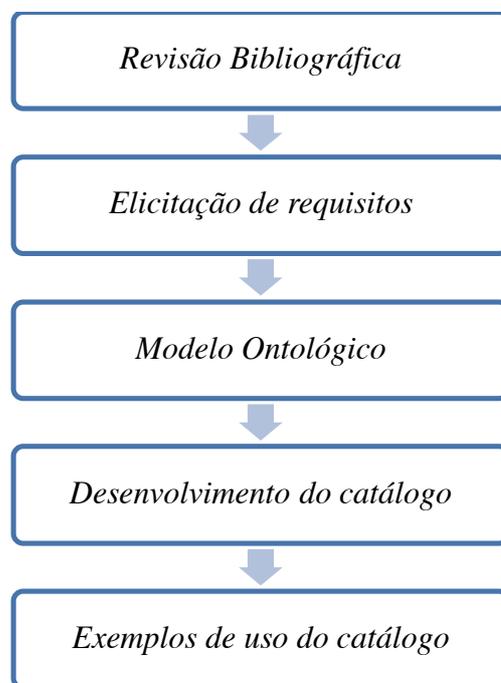
Segundo Guinard & Trifa (2016), a Web das Coisas é uma especialização da Internet das Coisas que usa o que tornou a Web tão bem-sucedida e a aplica a dispositivos físicos para tornar os últimos desenvolvimentos na Internet das Coisas acessíveis a um maior número de desenvolvedores. Ao fazer uso dos padrões da Web para conectar objetos da vida cotidiana à Web, questões relacionadas a como representar, armazenar, interconectar, pesquisar e organizar informações geradas pelos sistemas IoT são facilitadas.

Com os movimentos da nova Web é possível ir além - as tecnologias da Web Semântica podem desempenhar um papel fundamental na integração de dispositivos físicos heterogêneos. Atzori *et al.* (2010) resalta que a partir do uso de tais tecnologias é possível explorar soluções de modelagem apropriadas para descrição das coisas e o desenvolvimento de mecanismos autônomos de raciocínio, permitindo tomadas de decisão a partir dos dados gerados por sistemas IoT. A junção dos conceitos da Web Semântica e os dos sistemas Internet das Coisas resulta na abordagem conhecida como Web Semântica das Coisas.

Pfisterer *et al.* (2011), destaca três pontos importantes no que diz respeito a aplicações Web Semântica das Coisas: os dispositivos físicos devem ser conectados à Internet, as máquinas devem ser capazes de descobrir e entender a semântica dos dados retornados pelos dispositivos. Além disso, uma técnica para encontrar os dispositivos que possam prover dados relevantes é necessária. Assim, na Web Semântica das Coisas, dados de dispositivos físicos heterogêneos do mundo real são anotados (descritos) semanticamente por meio de ontologias, estruturados segundo um modelo de dados padrão e interconectados com dados de diferentes fontes de dados da Web, facilitando a geração autônoma de novos conhecimentos sobre o contexto dos dados ou dos próprios objetos.

### 3. TRAJETÓRIA METODOLÓGICA

A metodologia aplicada na realização do trabalho foi dividida em cinco etapas, enumeradas na Figura 11.



**Figura 11** - Principais etapas do trabalho

Fonte: Elaboração própria

A primeira etapa consistiu em revisão bibliográfica para levantamento do estado da arte do tema, utilizando a plataforma SCOPUS. A busca teve como foco trabalhos que relacionassem as seguintes áreas de interesse: Sensores, Web Semântica e *Linked Data*. Após a familiarização com o tema, a próxima etapa consistiu na elicitação dos requisitos funcionais do catálogo. A partir da análise dos requisitos, pôde-se dar início a etapa de construção dos modelos ontológicos. Com os modelos ontológicos em mãos, a aplicação semântica é desenvolvida, fazendo uso da plataforma *Linked Data Callimachus*<sup>11</sup>. Por fim, exemplos de uso realistas são apresentados para corroborar a eficácia da aplicação.

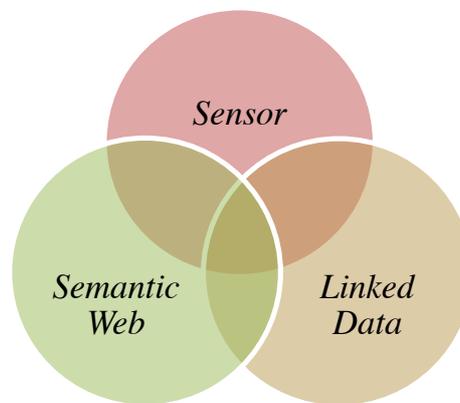
#### 3.1. Revisão bibliográfica

Em 2009, o W3C iniciou um projeto para criação da ontologia *Semantic Sensor Network Ontology* (SSN) para descrição de dados de sensores em termos de suas capacidades técnicas, processo de medição, observações e implantação seguindo os padrões da Web Semântica. O grupo desenvolvedor, que foi nomeado como *W3C Semantic Sensor Network Incubador group*

<sup>11</sup> <http://callimachusproject.org/>

(SSN-XG), iniciou seus trabalhos a partir do estudo e revisão de ontologias pré-existentes que tinham como foco atender a área de sensores. A finalização da ontologia se deu com a disponibilização de um documento final contendo toda descrição desta ontologia (BARNAGHI *et al.*, 2011). A partir desta iniciativa, muitos trabalhos envolvendo a área de sensores e Web semântica começaram a surgir.

Os avanços no desenvolvimento de ontologias e os incentivos para publicação de dados ligados na Web refletiram em aumento do número de aplicações *Linked Data* relacionadas à área de sensores nos últimos anos, de acordo com pesquisa realizada na base dados SCOPUS. As áreas de interesse envolvidas na pesquisa são evidenciadas na Figura 12.



**Figura 12** - Resultado de pesquisa na base de dados SCOPUS

Fonte: Elaboração própria

Com o propósito de identificar trabalhos relacionados ao tema, os termos indicados na Figura 12 serviram como base para pesquisa realizada na base de dados SCOPUS, indicada no Quadro 2.

**Quadro 2** - Termos utilizados na pesquisa

*(TITLE-ABS-KEY (linked AND sensor AND data) OR TITLE-ABS-KEY (linked AND data) OR TITLE-ABS-KEY (linked AND open AND data ) AND TITLE-ABS-KEY ( Web AND semantic ) AND TITLE-ABS-KEY (sensor) OR TITLE-ABS-KEY (sensing))*

Fonte: Elaboração própria

Como resultado da pesquisa, foram retornados 193 documentos. A intenção desta busca foi encontrar trabalhos sobre *Linked Data* aplicado à área de sensores. O termo Web Semântica (Semantic Web) foi utilizado para refinar ainda mais a busca. Embora o termo *Linked Data* (Dados Ligados) por si só já implique em Web Semântica, apenas a utilização dos termos *Linked Data* e *Sensor* (Sensor) gera resultados pouco precisos. A Tabela 3 apresenta a relação dos trabalhos encontrados na pesquisa, organizados de acordo com o ano em que foram desenvolvidos ou publicados.

**Tabela 3** - Número de publicações envolvendo as áreas de interesse entre 2009 e 2017

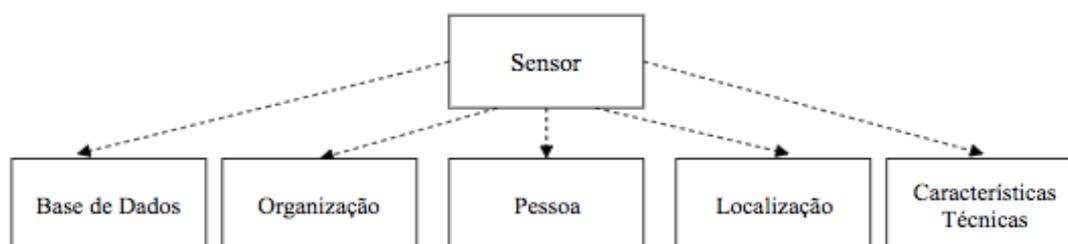
<i>Ano</i>	<i>Número de publicações</i>
2009	4
2010	12
2011	14
2012	23
2013	19
2014	33
2015	30
2016	29
2017	29

Fonte: SCOPUS (2018)

Na Tabela 3, os trabalhos retornados na busca refletem esforços crescentes, nos últimos anos, na criação de aplicações para que dados de sensores, geograficamente distribuídos, possam ser disponibilizados na Web Semântica. Abordagens mais recentes visam incorporar, além de sensores, os atuadores à Web, tendo em vista a nova versão da ontologia SSN publicada em 2017, que inclui termos para descrição de sensores, atuadores e amostradores (HALLER *et al.* 2017). Os trabalhos resultantes desta pesquisa serão discutidos no capítulo 8.

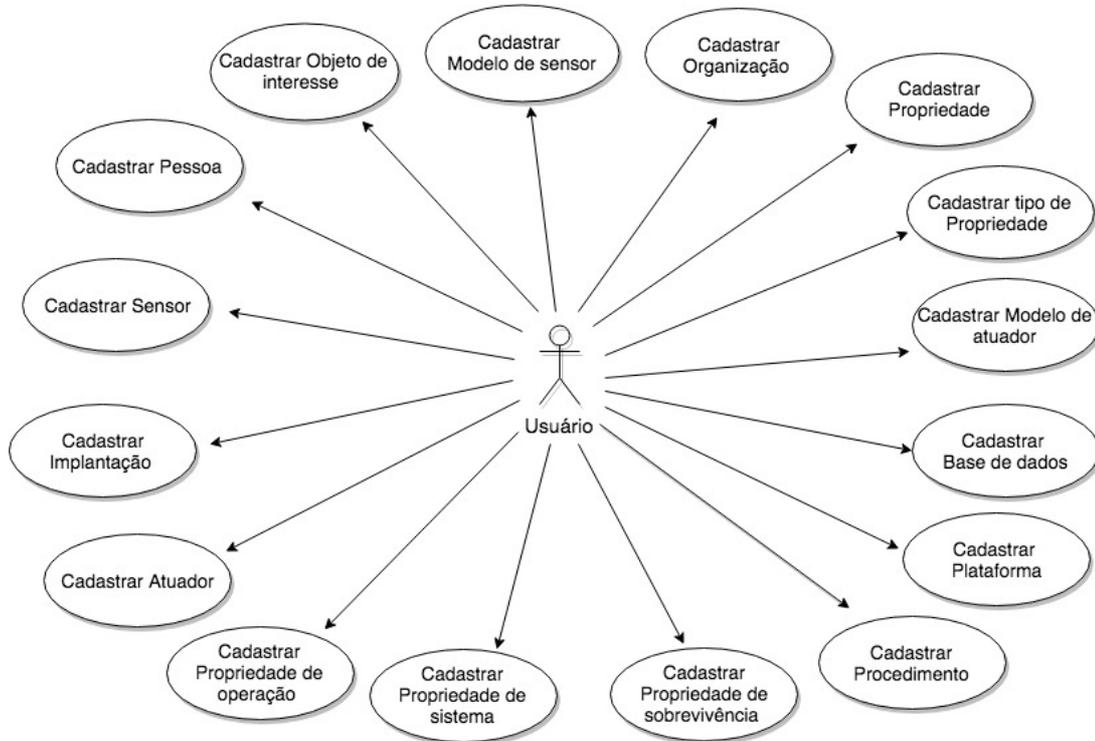
### 3.2. Elicitação de requisitos

Com o resultado da pesquisa bibliográfica, foi possível verificar que os trabalhos consultados descrevem seus sensores por meio de atributos de espaço, tempo e características técnicas. Neste trabalho, propõe-se estender a contextualização de sensores e atuadores explorando outros atributos que os qualificam sob diversos aspectos, por meio do uso de diferentes ontologias. Assim, a análise de artigos constitui-se como o ponto de partida da etapa de elicitação de requisitos, sobretudo a análise do artigo sobre a ontologia SSN, no qual exemplos de uso da ontologia são apresentados (HALLER *et al.*, 2017). Ainda, estudos de manuais técnicos de sensores e atuadores e reuniões periódicas com o orientador foram realizados a fim de elencar os diferentes requisitos. A Figura 13 apresenta os diferentes aspectos adotados para descrever sensores.

**Figura 13** - Aspectos adotados para descrição de sensores

Fonte: Elaboração própria

Tendo como objetivo descrever sensores, por meio dos aspectos apresentados na Figura 13, diferentes funcionalidades foram identificadas. A Figura 14 apresenta todas as funcionalidades da aplicação, ou seja, todos os recursos que podem ser cadastrados na plataforma, por meio de um diagrama de casos de uso UML.



**Figura 14 - Funcionalidades do catálogo**

Fonte: Elaboração própria

Tendo em vista a presença frequente de atuadores em aplicações Web das Coisas e a inclusão de tais dispositivos na nova versão da ontologia SSN, optou-se por possibilitar a catalogação de atuadores na aplicação proposta fazendo uso dos mesmos aspectos utilizados para descrever sensores, exceto pela adoção do aspecto “Base de dados”. Nesse caso, na versão atual deste trabalho, a catalogação de atuadores configura-se como uma derivação da catalogação de sensores. Assim sendo, de maneira geral, tais dispositivos são descritos de forma similar, ou seja, fazendo uso dos recursos definidos na Figura 14. Cada um destes recursos representa uma funcionalidade na aplicação. Para melhor entendimento destas funcionalidades, a Tabela 4 apresenta descrição e/ou exemplos de uso dos recursos que podem ser cadastrados no catálogo.

**Tabela 4** - Descrição dos recursos que podem ser cadastrados na plataforma

<i>Recurso</i>	<i>Descrição/Exemplo</i>
Sensor	Exemplar físico instalado que detecta e mede uma propriedade física (p. ex.: acelerômetro, giroscópio, etc.).
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 <i>do ar</i> , temperatura <i>da sala de estar</i> , vazão <i>do rio Paraíba</i> , nível <i>do tanque da cozinha</i> , 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 uma 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.: um telefone móvel, a parede da sala de estar, etc.)
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)
Modelo de Atuador	Classe que descreve as características de um modelo de atuador. Um atuador físico é uma instância de uma classe (modelo).

Fonte: Elaboração própria

Os recursos indicados na Tabela 4 podem ser cadastrados na aplicação por possuírem alguma relação com o universo de sensores e atuadores. A maior parte destes recursos pode ser associada diretamente ao cadastro do sensor (ou do modelo de sensor) por meio de propriedades específicas. No entanto, no caso de atuadores, tais dispositivos são cadastros de maneira independente, ou seja, a existência de um atuador não demanda, necessariamente, a existência de um sensor (e vice-versa), uma vez que a relação entre tais dispositivos não é explicitada no catálogo semântico proposto. Ainda 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 de tais informações e obtenham como resultado a solução de diversos problemas.

### 3.3. Modelo ontológico

Com base na análise dos requisitos funcionais, ontologias consagradas foram selecionadas. Com as ontologias em mãos, pôde-se dar início a identificação de classes e propriedades das ontologias selecionadas para descrever os diferentes recursos envolvidos. O uso combinado destas ontologias, ou seja, o modelo ontológico propriamente dito é apresentado no capítulo 5.

#### 3.3.1. Seleção de ontologias

Para o desenvolvimento da aplicação foi essencial a análise e seleção das ontologias (vocabulários). Dois critérios básicos foram utilizados para este fim: verificação das ontologias aplicáveis aos tipos de recursos envolvidos no trabalho e o reconhecimento destas ontologias na Web. As ontologias mais reconhecidas ou que se tornaram padrão de *facto* podem ser encontradas no catálogo *Linked Open Vocabularies* (LOV)<sup>12</sup>. Assim, fazendo uso do catálogo LOV foi possível buscar ontologias de interesse, de acordo com cada funcionalidade do catálogo, e obter como resultado apenas ontologias desenvolvidas em consonância com os padrões regidos pelo W3C.

Uma vez que o foco da aplicação é a catalogação de sensores e atuadores, a principal ontologia utilizada para esta finalidade é a *Semantic Sensor Network* (SSN), pois 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

---

<sup>12</sup> <http://lov.okfn.org/dataset/lov>

que qualificam sensores e atuadores. A Tabela 5 reúne todas as ontologias utilizadas na aplicação.

**Tabela 5** - Ontologias adotadas para descrição de recursos

<i>Prefixo</i>	<i>Nome da ontologia</i>	<i>Descrição/Finalidade</i>
Foaf	Friend of a friend (BRICKLEY & MILLER, 2014)	Descreve pessoas e organizações
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 <i>et al.</i> , 2017)	Descreve sensores, atuadores, observações e conceitos relacionados.
Vcard	VCard (IANNELLA & MCKINNEY, 2014)	Descreve pessoas e organizações.
Sosa	Sensor, Observation, Sample and Actuator (HALLER <i>et al.</i> , 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.
Skos	Simple Knowledge Organization System (MILES & BECHHOFFER, 2009)	Oferece padrões para representar sistemas de organização de conhecimento (taxonomias e tesouros).
ssn-system	System Capability module (HALLER <i>et al.</i> , 2017)	Extensão da ontologia SSN para descrição de sistemas em termos de suas capacidades, faixas de operação e de sobrevivência.
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.
Schema	Schema.org <sup>13</sup>	Vocabulário multidomínio, criado pela Google, Yahoo, Microsoft e Yandex para descrever entidades, seus relacionamentos e ações.
dcterms	Dublin Core <sup>14</sup>	Descreve metadados genéricos.
dcat	Data Catalog Vocabulary (MAALI & ERICKSON, 2014)	Vocabulário para descrição de bases de dados que publicam dados em diferente formatos.
Gr	Good Relations (HEPP, 2011)	Vocabulário para descrever comércio eletrônico.
Org	Organizational Ontology (REYNOLDS, 2014)	Descreve organizações e suas estruturas organizacionais.

Fonte: Elaboração própria

<sup>13</sup> <http://schema.org/>

<sup>14</sup> <http://dublincore.org/documents/dcmi-terms/>

Cada funcionalidade identificada na etapa de elicitación de requisitos (Figura 14) 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 a qual pertence, sob o ponto de vista das ontologias e metaontologias empregadas. A Tabela 6 apresenta os recursos que podem ser cadastrados na aplicação e a definição dos seus “tipos”.

**Tabela 6 - Tipos de recursos**

<i>Recurso</i>	<i>Tipos</i>
Sensor ( <i>Sensor</i> )	?SensorModel sosa:Sensor ssn:System
Modelo de Sensor ( <i>Sensor Model</i> )	owl:Class (rdfs:subClassOf sosa:Sensor rdfs:subClassOf sosa:System)
Propriedade de sistema ( <i>System Property</i> )	owl:Class (rdfs:subClassOf ssn-system:SystemProperty)
Propriedade de operação ( <i>Operating Property</i> )	owl:Class (rdfs:subClassOf ssn-system:OperatingProperty)
Propriedade de sobrevivência ( <i>Survival Property</i> )	owl:Class (rdfs:subClassOf ssn-system:SurvivalProperty)
Tipo de Propriedade ( <i>Property Type</i> )	owl:Class (rdfs:subClassOf ssn:Property)
Propriedade ( <i>Property</i> )	?PropertyType ssn:Property
Procedimento ( <i>Procedure</i> )	sosa:Procedure
Objeto de interesse ( <i>Feature Of Interest</i> )	sosa:FeatureOfInterest
Implantação ( <i>Deployment</i> )	ssn:Deployment
Plataforma ( <i>Platform</i> )	sosa:Platform
Pessoa ( <i>Person</i> )	foaf:Person schema:Person
Organização ( <i>Organization</i> )	foaf:Organization vcard:Organization schema:Organization org:Organization
Base de dados ( <i>Dataset</i> )	dcat:Dataset
Atuador ( <i>Actuator</i> )	?ActuatorModel sosa:Actuator ssn:System
Modelo de Atuador ( <i>Actuator Model</i> )	owl:Class (rdfs:subClassOf sosa:Actuator rdfs:subClassOf ssn:System)

Fonte: Elaboração própria

Na Tabela 6, 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 prédefinida 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 6, diversas propriedades são utilizadas. Tais propriedades e seus respectivos domínios e contradomínios, conforme definidos na ontologia de origem, são apresentadas nas Tabelas 7-17, a seguir.

i. Propriedades da ontologia DCAT:

A ontologia DCAT é utilizada para descrever “Bases de dados” no catálogo semântico proposto. As propriedades selecionadas para descrever tais recursos são apresentadas na Tabela 7.

**Tabela 7** - Domínios e contradomínios das propriedades da ontologia DCAT utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>dcat:byteSize</i>	dcat:Distribution	rdfs:Literal
<i>dcat:distribution</i>	dcat:Dataset	dcat:Distribution
<i>dcat:downloadURL</i>	dcat:Distribution	rdfs:Resource
<i>dcat:mediaType</i>	dcat:Distribution	dcat:MediaTypeOrExtent
<i>dcat:byteSize</i>	dcat:Distribution	rdfs:Literal

Fonte: Elaboração Própria

As propriedades da ontologia DCAT apresentadas na Tabela 7 que possuem como domínio *dcat:Dataset* são utilizadas para descrever “Bases de dados” (de sensores). Já as propriedades cujo domínio consiste em *dcat:Distribution* são utilizadas para descrever o arquivo (distribuição) no qual os dados de observação de sensores são armazenados. A relação entre uma dada base de dados e o arquivo contendo os dados do dispositivo é definida pela propriedade *dcat:distribution*.

ii. Propriedades da ontologia Dublin Core:

A ontologia Dublin Core é adotada para descrever diferentes recursos na aplicação. As propriedades da ontologia Dublin Core selecionadas para descrição dos recursos são apresentadas na Tabela 8.

**Tabela 8-** Domínios e contradomínios das propriedades da ontologia Dublin Core utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>dcterms:creator</i>		dcterms:Agent
<i>dcterms:isReferencedBy</i>		
<i>dcterms:issued</i>		rdfs:Literal
<i>dcterms:language</i>		dcterms:LinguisticSystem
<i>dcterms:subject</i>		
<i>dcterms:title</i>		rdfs:Literal

Fonte: Elaboração Própria

Na Tabela 8, domínio (e contradomínio) vazio é o mesmo que *owl:Thing*. Assim, as propriedades da ontologia Dublin Core apresentadas na Tabela 8 são utilizadas para descrever diferentes “coisas” ou recursos na aplicação, tais como “Pessoa”, “Organização”, “Base de dados”, “Sensor”, etc. Por exemplo, no catálogo semântico proposto, todos os recursos cadastrados possuem um “nome” dado pela propriedade *dcterms:title*. Além disso, o valor nesta propriedade é sempre um literal (*rdfs:Literal*), vide Tabela 8.

### iii. Propriedades da ontologia FOAF:

A ontologia FOAF é utilizada para descrever, principalmente, pessoas e organizações na aplicação. As propriedades selecionadas na descrição destes e outros recursos da aplicação são apresentadas na Tabela 9.

**Tabela 9-** Domínios e contradomínios das propriedades da ontologia FOAF utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>foaf:birthday</i>	foaf:Agent	rdfs:Literal
<i>foaf:depiction</i>	owl:Thing	foaf:Image
<i>foaf:homepage</i>	owl:Thing	foaf:document
<i>foaf:maker</i>	owl:Thing	foaf:Agent
<i>foaf:mbox</i>	foaf:Agent	owl:Thing
<i>foaf:name</i>	owl:Thing	rdfs:Literal
<i>foaf:nick</i>		
<i>foaf:phone</i>		xsd:string

Fonte: Elaboração Própria

Na Tabela 9, como o domínio das propriedades indicadas são, em sua maioria, definidos como *owl:Thing*, foi possível fazer uso destas propriedades não apenas na descrição de pessoas e organizações. Por exemplo, um “Sensor” pode ser vinculado a uma “Imagem” por meio da propriedade *foaf:depiction*. As demais propriedades da ontologia FOAF adotadas e indicadas na Tabela 9 serão abordadas de forma detalhada no capítulo 5.

## iv. Propriedades da ontologia Good Relations:

Para descrever o recurso “Organização” faz-se uso da ontologia Good Relations, mais especificamente, da propriedade apresentada na Tabela 10.

**Tabela 10-** Domínios e contradomínios das propriedades da ontologia Good Relations utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>gr:legalName</i>	gr:BusinessEntity	rdfs:Literal

Fonte: Elaboração Própria

A propriedade *gr:legalName* indicada na Tabela 10 é utilizada para definição do nome legal ou formal de uma dada organização. Dado que o domínio desta propriedade é definido como sendo *gr:BusinessEntity*, esta propriedade é utilizada para descrever, apenas, organizações na aplicação proposta.

## v. Propriedades da ontologia ORG:

A ontologia ORG é adotada para descrever organizações no catálogo proposto. As propriedades da ontologia ORG, selecionadas para descrever o recurso em questão, são apresentadas na Tabela 11.

**Tabela 11-** Domínios e contradomínios das propriedades da ontologia ORG utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>org:classification</i>	org:Organization	skos:Concept
<i>org:hasMember</i>	org:Organization	foaf:Agent

Fonte: Elaboração Própria

Na Tabela 11, a propriedade *org:classification* é adotada para descrever os diferentes tipos de organização. Tais tipos são definidos como instâncias da classe *skos:Concept*, ou seja, trata-se de um tesouro de tipos de organização. Já a propriedade *org:hasMember* é utilizada para vincular uma dada organização a um agente (*foaf:Agent*) associado à instituição.

## vi. Propriedades da ontologia OWL:

A ontologia OWL é utilizada para descrever diferentes recursos na aplicação. As propriedades adotadas são apresentadas na Tabela 12.

**Tabela 12-** Domínios e contradomínios das propriedades da ontologia OWL utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>owl:hasValue</i>	owl:Restriction	rdfs:Resource
<i>owl:onProperty</i>	owl:Restriction	rdf:Property
<i>owl:sameAs</i>	owl:Thing	owl:Thing

Fonte: Elaboração Própria

É possível que os recursos cadastrados na aplicação existam em outras bases de dados na Web. Nestes casos, a propriedade *owl:sameAs* é empregada para explicitar que dois ou mais recursos dotados de URIs distintos referem-se a mesma “coisa”. Esta propriedade pôde ser aplicada na descrição de pessoas, organizações e objetos de interesse, tendo em vista o domínio e contradomínio (*owl:Thing*). As demais propriedades da ontologia OWL adotadas e indicadas na Tabela 12 serão abordadas de forma detalhada no capítulo 5.

vii. Propriedades da ontologia PROV:

No catálogo proposto, a ontologia PROV é utilizada para descrever diferentes recursos, tais como sensores, atuadores e implantação. As propriedades selecionadas para descrição destes recursos são apresentadas na Tabela 13.

**Tabela 13-** Domínios e contradomínios das propriedades da ontologia PROV utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>prov:wasAssociatedWith</i>	prov:Activity	prov:Agent
<i>prov:wasAttributedTo</i>	prov:Entity	prov:Agent

Fonte: Elaboração Própria

A ontologia PROV define *prov:Activity* como sendo uma atividade que acontece em um dado período de tempo. Assim, esta propriedade é utilizada para descrever o recurso “Implantação”. No catálogo proposto, implantações estão associadas a certos agentes por meio da propriedade *prov:wasAssociatedWith*, tais como organizações ou pessoas. Além disso, sensores e atuadores são atribuídos a estes mesmos “tipos” de agentes por meio da propriedade *prov:wasAttributedTo*.

viii. Propriedades das ontologias RDF e RDFS:

Propriedades dos vocabulários RDF e RDFS são utilizadas na descrição de diversos recursos na aplicação. Por exemplo: para indicar que um recurso é uma instância de uma classe ou, ainda, que todas as instâncias de uma classe são instâncias de outra classe, vide Tabela 14.

**Tabela 14-** Domínios e contradomínios das propriedades das ontologias RDF e RDFS utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>rdf:type</i>	rdfs:Resource	rdfs:Class
<i>rdfs:comment</i>	rdfs:Resource	rdfs:Literal
<i>rdfs:isDefinedBy</i>	rdfs:Resource	rdfs:Resource
<i>rdfs:label</i>	rdfs:Resource	rdfs:Literal
<i>rdfs:seeAlso</i>	rdfs:Resource	rdfs:Resource
<i>rdfs:subClassOf</i>	rdfs:Class	rdfs:Class

Fonte: Elaboração Própria

As propriedades indicadas na Tabela 14, que possuem domínio definido como sendo *rdfs:Resource*, podem ser aplicadas a todos recursos cadastrados na aplicação. No caso da propriedade *rdfs:subClassOf*, tal propriedade é aplicada na descrição de recursos definidos na aplicação como sendo *owl:Class* (ou *rdfs:Class*). Por exemplo: Ao definir “Modelo de Sensor” (SensorModel) como sendo subclasse (*rdfs:subClassOf*) da classe *sosa:Sensor*, torna-se explícito que todas as instâncias da classe “Modelo de Sensor” são instâncias da classe *sosa:Sensor*. As demais propriedades adotadas e indicadas na Tabela 14 serão abordadas de forma detalhada no capítulo 5.

ix. Propriedades da ontologia Schema.Org:

A ontologia Schema.Org é utilizada para descrever diferentes recursos na aplicação. As propriedades, selecionadas desta ontologia, são apresentadas na Tabela 15.

**Tabela 15-** Domínios e contradomínios das propriedades da ontologia Schema.Org utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>schema:addressCountry</i>	schema:GeoCoordinates, schema:GeoShape, schema:PostalAddress	schema:Country schema:Text
<i>schema:addressLocality</i>	schema:PostalAddress	schema:Text
<i>schema:latitude</i>	schema:GeoCoordinates	schema:Text, schema:Number
<i>schema:location</i>	schema:Organization, schema:Event, schema:Action	schema:PostalAddress, schema:Place, schema:Text
<i>schema:longitude</i>	schema:GeoCoordinates	schema:Text, schema:Number
<i>schema:minValue</i>	schema:MonetaryAmount, schema:PropertyValue, schema:QuantitativeValue, schema:PropertyValueSpecification	schema:Number
<i>schema:postalCode</i>	schema:GeoShape, schema:PostalAddress	schema:Text
<i>schema:unitCode</i>	schema:QuantitativeValue, schema:TypeAndQuantityNode, schema:UnitPriceSpecification	schema:Text schema:URL
<i>schema:value</i>	schema:PropertyValue, schema:QuantitativeValue, schema:MonetaryAmount	schema:Number, schema:Text, schema:Boolean, schema:StructuredValue

Fonte: Elaboração Própria

Na Tabela 15, contradomínios com mais de uma classe indica interseção das classes. Por exemplo, no caso da propriedade *schema:latitude* o contradomínio desta propriedade é definido como sendo *schema:Text* e *schema:Number*. Assim, o valor associado à propriedade *schema:latitude* irá pertencer tanto a classe *schema:Text* quanto a classe *schema:Number*. O uso das propriedades indicadas na Tabela 15 é demonstrado no capítulo 5.

x. Propriedades da ontologia SSN:

A ontologia SSN é utilizada para descrever sensores, atuadores e outros recursos associados a tais dispositivos. As propriedades selecionadas da ontologia SSN podem ser visualizadas na Tabela 16.

**Tabela 16-** Domínios e contradomínios das propriedades da ontologia SSN utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>sosa:observes</i>	<i>sosa:Sensor</i>	<i>sosa:ObservableProperty</i>
<i>sosa:actsOnProperty</i>	<i>sosa:Actuator</i>	<i>sosa:ActuatableProperty</i>
<i>ssn-system:hasOperatingProperty</i>		
<i>ssn-system:hasOperatingRange</i>		
<i>ssn-system:hasSurvivalProperty</i>		
<i>ssn-system:hasSurvivalProperty</i>		
<i>ssn-system:hasSystemCapability</i>		
<i>ssn-system:hasSystemProperty</i>		
<i>ssn-system:inCondition</i>		
<i>ssn:deployedOnPlataform</i>		
<i>ssn:deployedSystem</i>		
<i>ssn:forProperty</i>		
<i>ssn:hasInput</i>		
<i>ssn:hasOutput</i>		
<i>ssn:implements</i>		
<i>ssn:isPropertyOf</i>		

Fonte: Elaboração Própria

Dentre as propriedades apresentadas na Tabela 16 apenas as propriedades *sosa:observes* e *sosa:actsOnProperty* possuem domínio e contradomínio definidos na ontologia SSN. Tal definição faz com que a aplicação destas propriedades seja destinada a “coisas” (recursos) específicas. Por exemplo, a propriedade *sosa:observes* deve ser utilizada, exclusivamente, na descrição de instâncias da classe *sosa:Sensor*. Enquanto a propriedade *sosa:actsOnProperty* deve ser aplicada a instâncias da classe *sosa:Actuator*. As demais propriedades indicadas na Tabela 16 podem ser utilizadas para descrever recursos (*owl:Thing*) de diversas naturezas. Apesar disso, o domínio e contradomínio das demais propriedades, indicadas na Tabela 16, são definidos dentro das classes da ontologia SSN.

xi. Propriedades da ontologia VCARD:

A ontologia VCARD, em conjunto com as ontologias Good Relations e ORG, é utilizada na descrição de organizações. As propriedades da ontologia VCARD, selecionadas para descrever organizações, são apresentadas na Tabela 17.

**Tabela 17-** Domínios e contradomínios das propriedades da ontologia VCARD utilizadas nos modelos

<i>Propriedade</i>	<i>Domínio (Domain)</i>	<i>Contradomínio (Range)</i>
<i>vcard:hasURL</i>		
<i>vcard:adr</i>		
<i>vcard:locality</i>		
<i>vcard:street-address</i>		

Fonte: Elaboração Própria

Embora tenha sido aplicada exclusivamente para descrever organizações, os campos vazios (domínio e contradomínio) na Tabela 17 indicam que as propriedades apresentadas podem ser utilizadas na descrição de recursos de diferentes naturezas. No catálogo proposto, a propriedade *vcard:hasURL* é utilizada para descrever o endereço da página Web de uma dada organização. As demais propriedades são utilizadas para descrever as propriedades de localização de organizações.

Em conjunto, as propriedades apresentadas nas Tabelas 7-17 compõem os modelos ontológicos para descrição dos diferentes recursos ou requisitos funcionais da aplicação. Tendo em vista que sensores e atuadores são os elementos protagonistas da aplicação, a ontologia SSN, pode ser considerada a espinha dorsal deste trabalho e, por isso, é discutida de forma individualizada e detalhada no capítulo 4.

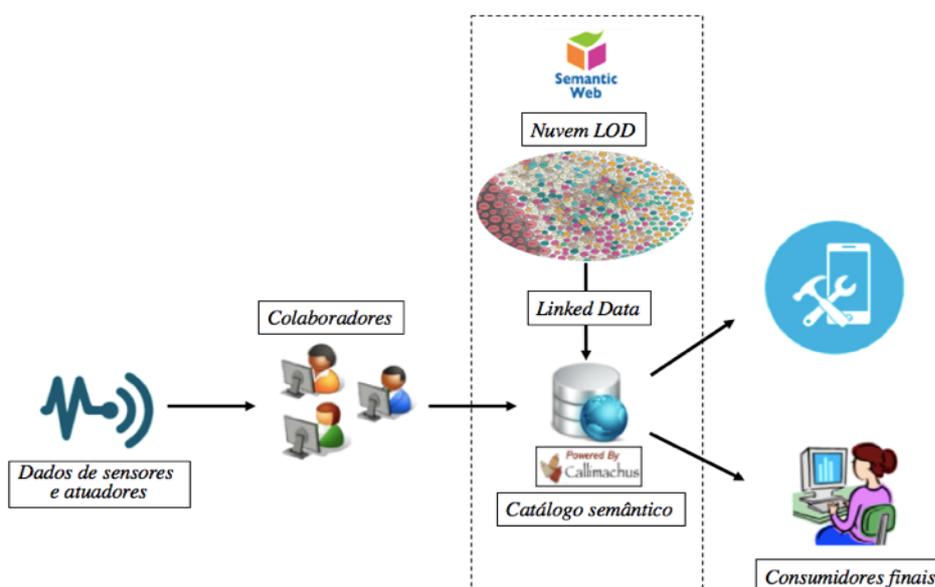
### 3.4. Desenvolvimento do catálogo

Esta etapa da metodologia envolve a seleção do *framework* para desenvolvimento do catálogo semântico proposto, a plataforma *Linked Data Callimachus*<sup>15</sup>, 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. Tendo em vista que o escopo da versão atual deste trabalho não inclui a validação da aplicação com potenciais usuários e a definição de perfis de usuários, optou-se por qualificar o estágio atual do desenvolvimento como sendo um protótipo de aplicação. Os detalhes sobre a construção deste protótipo são apresentados no capítulo 6.

<sup>15</sup> <http://callimachusproject.org/>

### 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 Callimachus facilita o desenvolvimento de aplicações de dados ligados por seguir padrões da Web e os princípios *Linked Data*, oferecendo: armazenamento de grafos RDF, consultas SPARQL, HTML 5 com RDFa, CSS3, Javascript, templates HTML CRUD<sup>16</sup> semi-prontos, 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 15 apresenta uma visão macro da arquitetura do catálogo proposto.

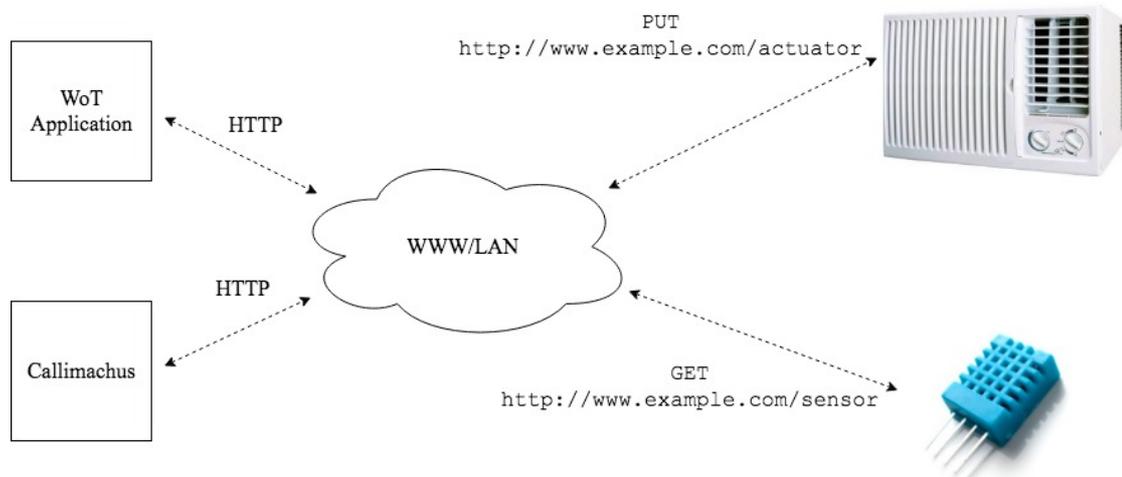


**Figura 15** - Macro-arquitetura da aplicação

Fonte: Elaboração própria

A Figura 15 evidencia o princípio de funcionamento do catálogo proposto. A ideia é que colaboradores de várias partes do mundo possam catalogar metadados de sensores e atuadores seguindo os padrões da Web Semântica e interliguem tais dados a dados de outras bases semânticas na Web. Tem-se como propósito, por exemplo, tornar disponível dados de sensores e atuadores para possibilitar que agentes de software promovam a integração de tais informações e obtenham como resultado a solução de diversos problemas. A Figura 16 apresenta um esquema que representa a utilização do catálogo semântico proposto em aplicações específicas.

<sup>16</sup> CRUD - Acrônimo para cadastro: Create Read Update Delete



**Figura 16** - Compartilhamento de dados do catálogo proposto na Web

Fonte: Elaboração própria

A Figura 16 apresenta exemplo de aplicação Web das Coisas, na qual dispositivos físicos são disponibilizados na Web. Por meio de aplicações como estas, desenvolvedores e agentes de software são capazes de trocar dados com os dispositivos usando solicitações HTTP padrão, independentemente de como o dispositivo está conectado. No entanto, apesar de estarem disponíveis na Web, encontrar tais dispositivos nem sempre é uma tarefa trivial. Nesse caso, a descoberta e uso dos dados dos dispositivos pode ser facilitada ao catalogar tais objetos no catálogo proposto, tendo em vista que, uma vez catalogados, os dados se tornam abertos e, portanto, podem ser acessados por qualquer agente de software ou usuário que queira fazer uso das informações.

No caso das aplicações WoT, além de divulgar seus dispositivos no catálogo, tais aplicações podem, ainda, fazer uso de dados de sensores e atuadores, geograficamente distribuídos e catalogados na aplicação, para diversas finalidades, tais como realizar *mashups* de dados, controlar dispositivos em uma rede local ou acessar dispositivos abertos na Web, vide Figura 16.

### 3.5. Exemplos de uso do catálogo

Nesta etapa da metodologia, exemplos de uso realistas são apresentados com o propósito de corroborar a eficácia do catálogo proposto. Por questões de simplificação, apenas os grafos RDF subjacentes ao cadastro dos recursos catalogados, representados como *graph pattern* instanciados, serão apresentados no capítulo 7.

#### 4. ONTOLOGIA SEMANTIC SENSOR NETWORK ONTOLOGY - SSN

A ontologia *Semantic Sensor Network* (SSN) surgiu em 2012 com o objetivo de alinhar vocabulários para representação de sensores, pré-existentes, com os padrões do W3C. Este trabalho foi realizado pelo grupo “*The W3C Semantic Sensor Network Incubator Group (SSNO)*” e os resultados foram publicados em Barnaghi *et al.* (2011). No entanto, ao decorrer de sua utilização, alguns inconvenientes foram identificados nesta primeira versão da ontologia SSN, tais como alta complexidade, principalmente, devido o uso do vocabulário Dolce-UltraLite (DUL)<sup>17</sup> como ontologia de topo, além de seu elevado número de axiomas. Nesse cenário, em 2017, uma nova versão da ontologia SSN foi publicada, contendo diversas mudanças no escopo e nas suas características de axiomatização (HALLER *et al.*, 2017).

A busca por vocabulários mais leves, principalmente nas aplicações de Internet das Coisas, foi um fator considerado na nova versão da ontologia SSN. Nesse caso, técnicas de modularização foram aplicadas para criação de uma ontologia central e extensível (core) nomeada *Sensor, Observation, Sample, and Actuator* (SOSA) – um vocabulário independente, contendo termos elementares e com baixo nível de axiomatização. Outra mudança importante foi a adição de novas classes para descrever, além de sensores, outros dispositivos, tais como atuadores e amostradores.

Haller *et al.* (2017) define a nova versão da ontologia SSN da seguinte forma:

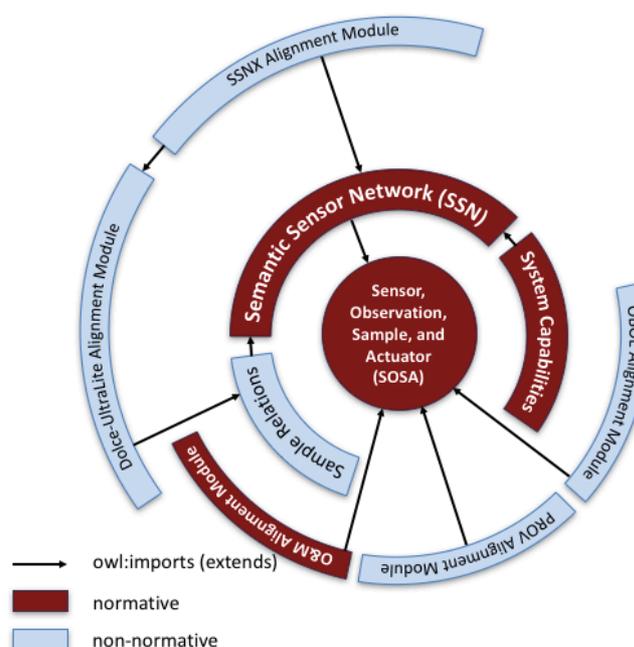
*“A ontologia SSN é uma ontologia para descrever sensores e suas observações, os procedimentos envolvidos, os elementos de interesse estudados, as amostras usadas para realizar as observações, as propriedades observadas, bem como os atuadores. A SSN segue uma arquitetura de modularização horizontal e vertical, incluindo uma ontologia central (extensível) leve e autossuficiente chamada SOSA (Sensor, Observação, Amostrador e Atuador) em relação às suas classes e propriedades elementares. Com um escopo diferenciado e diferentes níveis de axiomatização, SSN e SOSA são capazes de dar suporte para uma ampla gama de aplicações e casos de uso, incluindo imagens de satélite, monitoramento científico em grande escala, infraestruturas industriais e domésticas, sensoriamento social, ciência cidadã, observação orientada à engenharia do conhecimento e à Web das Coisas”.*

---

<sup>17</sup> [http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS\\_Ultralite](http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite)

A ontologia SSN é construída com base no conceito de modularização. Modularizar uma ontologia significa segmentar o conhecimento em partes menores e coesas. Como resultado desta prática, tem-se vocabulários com maior poder de expressividade e outros contendo apenas elementos estritamente necessários para um determinado caso de uso (HALLER *et al.*, 2017).

No caso da modularização da ontologia SSN, a mesma é caracterizada pelo uso da propriedade *owl:import* para relacionar os diferentes módulos. Além disso, dois tipos de segmentações são definidas dependendo do direcionamento da propriedade *owl:import*, são elas: segmentação vertical e horizontal (HALLER *et al.*, 2017). A Figura 17 apresenta os diferentes módulos que compõem a ontologia SSN.



**Figura 17** - Módulos da ontologia Semantic Sensor Network Ontology (SSN)

Fonte: Haller *et al.* (2017)

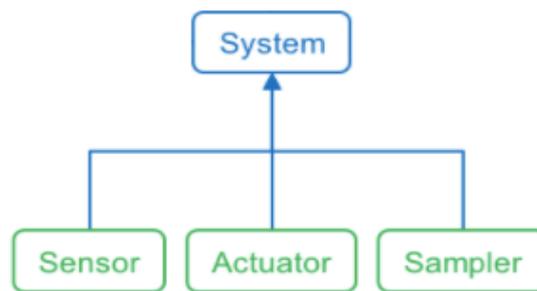
Diferente dos demais módulos, SOSA consiste em vocabulário totalmente independente. A autossuficiência de SOSA está relacionada ao fato de que este módulo não faz uso da propriedade *owl:import* para importar outros módulos. Outra característica marcante deste vocabulário é sua flexibilidade dada a não definição de domínio e range para as suas propriedades. Ao não definir o domínio e range para propriedades no escopo da ontologia, tem-se como resultado maior flexibilidade no uso destas propriedades por diferentes classes. Além disso, o módulo SOSA pode ser considerado um vocabulário simples por conter menor número de axiomas quando comparado aos módulos System Capability e SSN. A Tabela 18 apresenta uma comparação entre os módulos SOSA, SSN e System Capability.

**Tabela 18** - Comparação entre os módulos SOSA, SSN e System Capability

<i>Métricas</i>	<i>SOSA</i>	<i>SSN</i>	<i>System Capability</i>
Classes	16	23	46
Object Properties	21	36	44
Datatype Properties	2	2	2
Axiomas	12	107	163

Fonte: Elaboração própria

O módulo SSN importa o módulo SOSA. Enquanto System Capability importa o módulo SSN. Como resultado destas relações, o módulo System Capability herda os elementos contidos em SSN e em SOSA. Isso explica o maior número de classes, propriedades e axiomas contidos em System Capability (Tabela 18). Nesse caso, o vocabulário System Capability garante maior expressividade na descrição dos sistemas. As classes e propriedades contidas nestes módulos têm como propósito descrever três tipos de sistemas: Sensores, Atuadores e Amostradores, vide Figura 18.

**Figura 18** - Tipos de sistemas na ontologia SSN

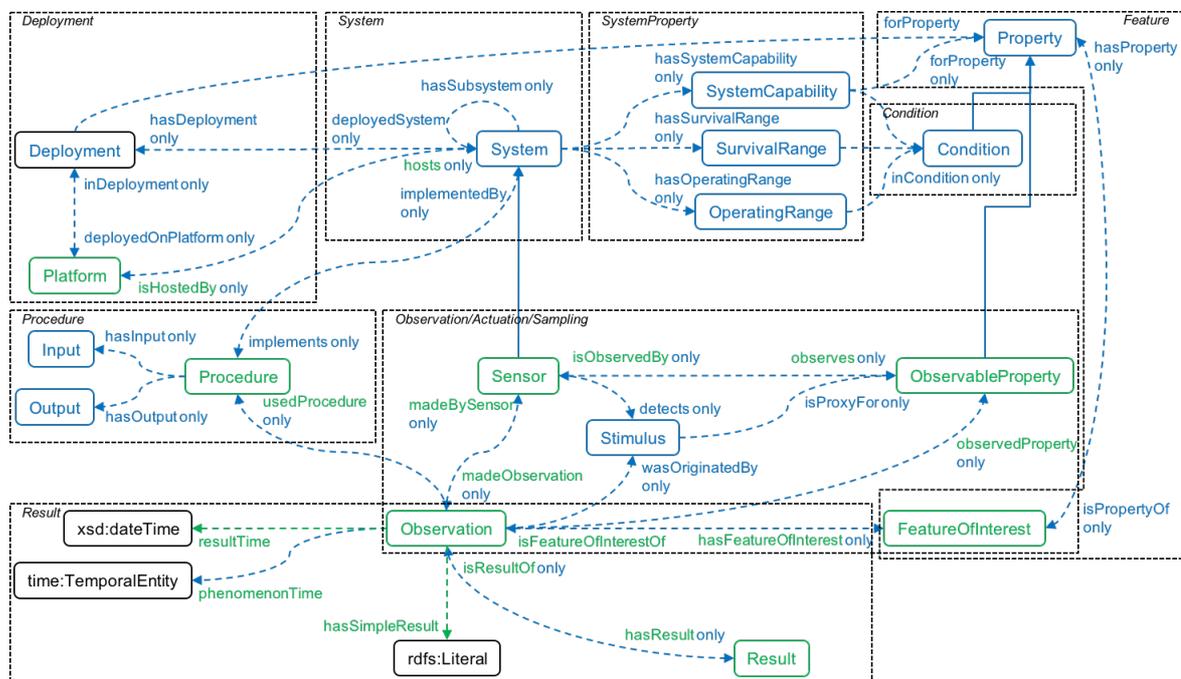
Fonte: Haller *et al.* (2017)

Por definição, Sensores (Sensor), Atuadores (Actuator) e Amostradores (Sampler) são subclasses da classe Sistema (System), vide Figura 18. Nesse caso, além de propriedades específicas, tais classes herdam todas as características da classe Sistema (System). Assim, é possível utilizar as mesmas propriedades que descrevem sistemas (System) para descrever Sensores (Sensor), Atuadores (Actuator) ou Amostradores (Sampler). Haller *et al.* (2017) define os sistemas apresentados na Figura 18 como segue:

- Sensor (Sensor): Dispositivo, agente (incluindo humanos) ou software (simulação) que implementa um dado procedimento (de observação). Sensores respondem a estímulos, por exemplo, uma mudança no ambiente e geram um resultado. Tais dispositivos podem ser hospedados por plataformas e fazer parte de uma implantação.

- Atuador (Actuator): Dispositivo capaz de implementar um procedimento (de atuação) que modifica o estado de um dado sistema. Assim como sensores, podem ser hospedados em plataformas e fazer parte de uma implantação.
- Amostrador (Sampler): Dispositivo utilizado para implementar um procedimento (de amostragem) para criação ou transformação de uma ou mais amostras. Assim como sensores e atuadores, podem ser hospedados em plataformas e fazer parte de uma implantação.

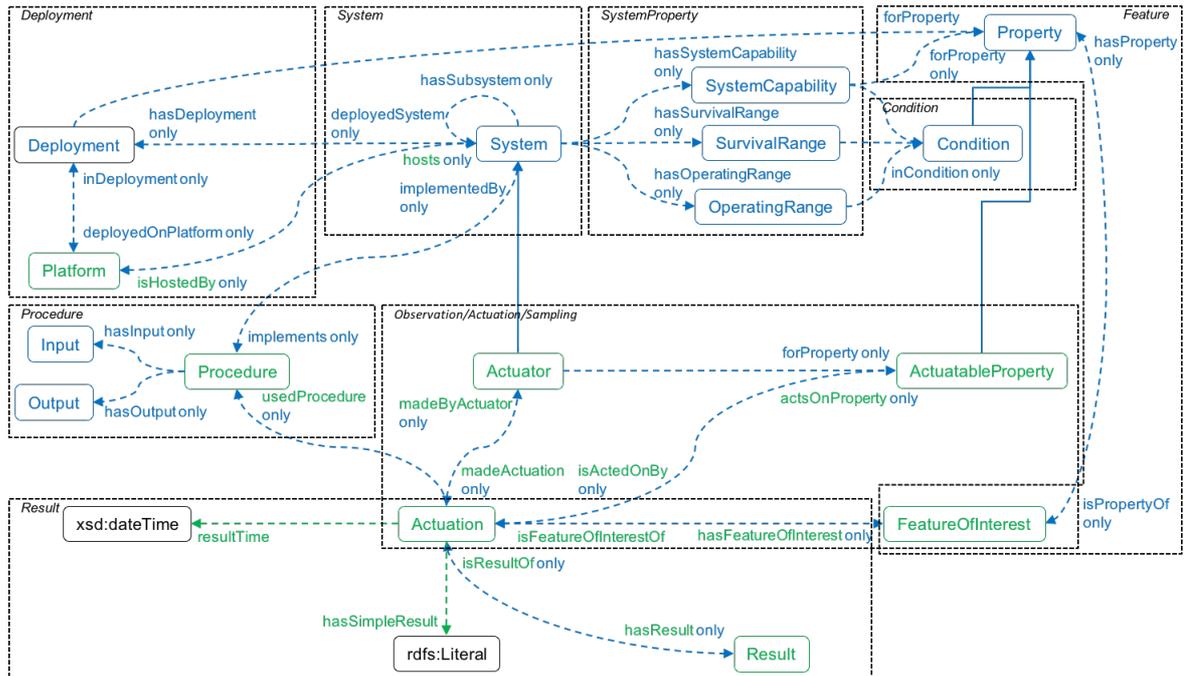
A Figura 19 apresenta classes e propriedades da ontologia SSN na perspectiva da observação realizada por sensores.



**Figura 19** - Classes e propriedades da ontologia SSN na perspectiva de observação

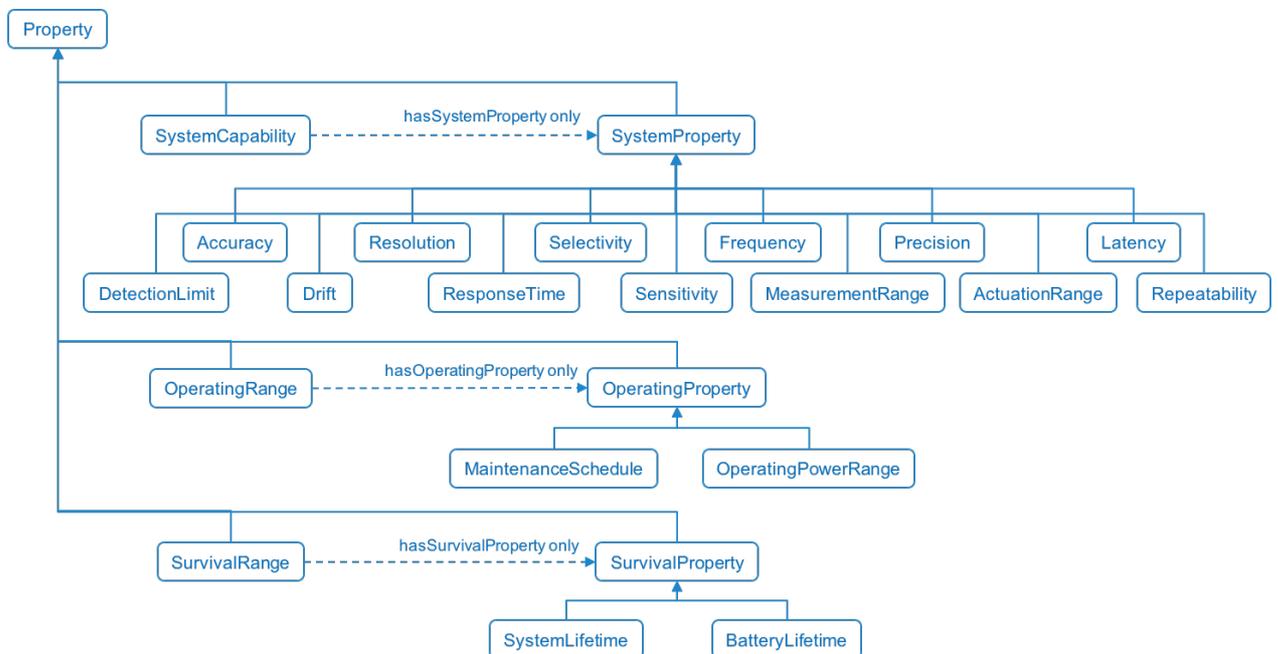
Fonte: Haller *et al.* (2017)

Na Figura 19, o módulo conceitual “Observation/Actuation/Sampling” pode envolver diferentes tipos de sistemas. Nesse caso, para cada tipo de sistema, este módulo irá conter classes e propriedades específicas. Por exemplo, no módulo Observation/Actuation/Sampling da Figura 19, é possível verificar propriedades específicas para descrição de sensores e seus processos de observação que, portanto, não podem ser utilizadas na descrição dos demais sistemas (sensores e amostradores), dada o domínio destas propriedades definidos na ontologia SSN. Além da perspectiva da observação, a ontologia SSN envolve, ainda, a perspectiva da atuação, vide Figura 20.



**Figura 20** - Classes e propriedades da ontologia SSN na perspectiva da atuação  
 Fonte: Haller *et al.* (2017)

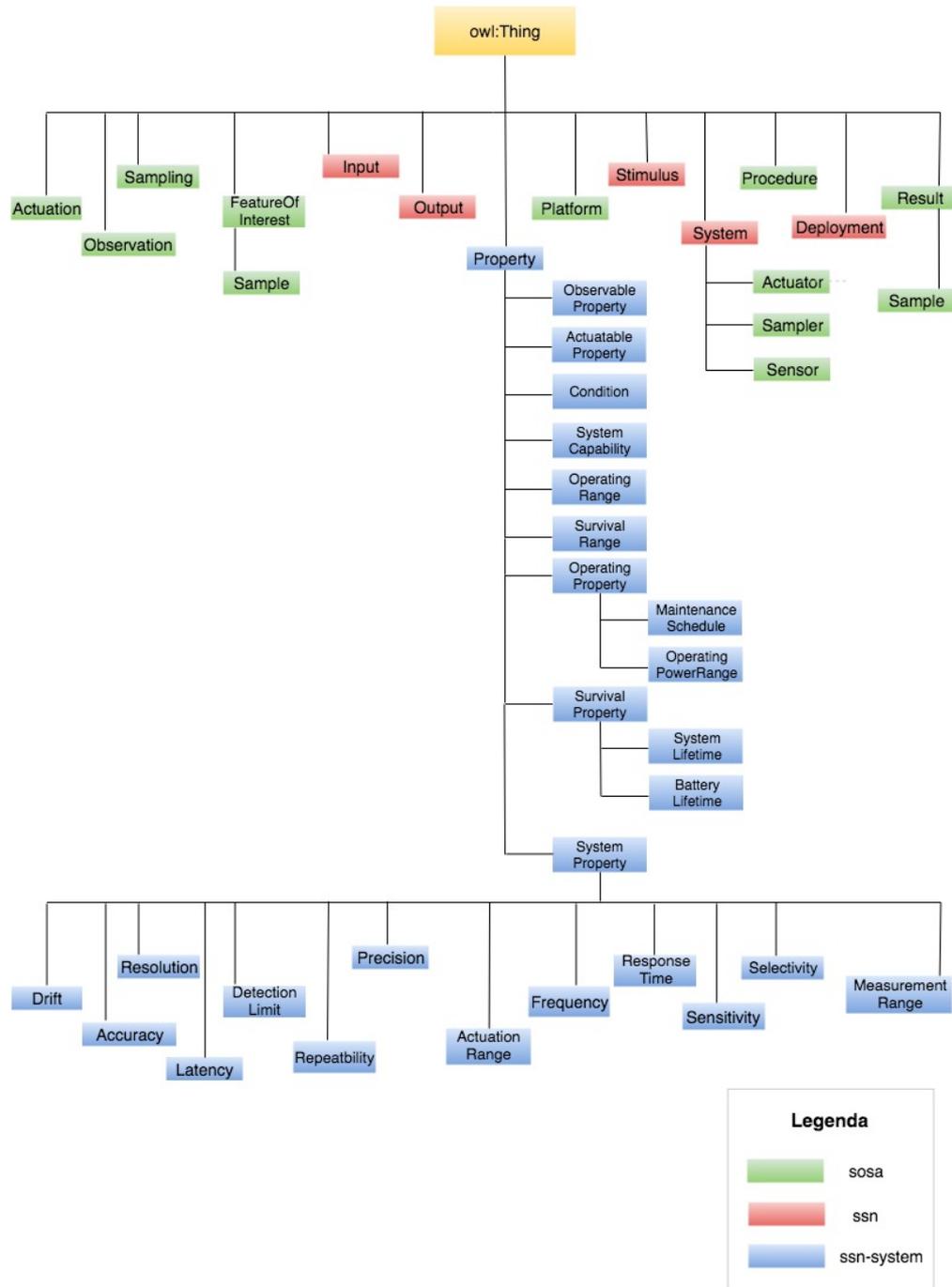
O módulo conceitual “System Property” das Figuras 19 e 20 não apresenta todas as propriedades (subclasses de “Property” ou *ssn:Property*) existentes na ontologia SSN. Tais propriedades são apresentadas na Figura 21.



**Figura 21** - Subclasses de *ssn:Property* na ontologia SSN  
 Fonte: Haller *et al.* (2017)

A Figura 21 evidencia diferentes classes encontradas no módulo System Capability da ontologia SSN utilizadas para descrever as características ou especificações técnicas de

sensores, atuadores e amostradores. Estas e outras classes da ontologia SSN, distribuídas nos diferentes módulos da ontologia, são apresentadas na Figura 22.



**Figura 22-** Hierarquia de classes e subclasses da ontologia SSN

Fonte: Elaboração própria

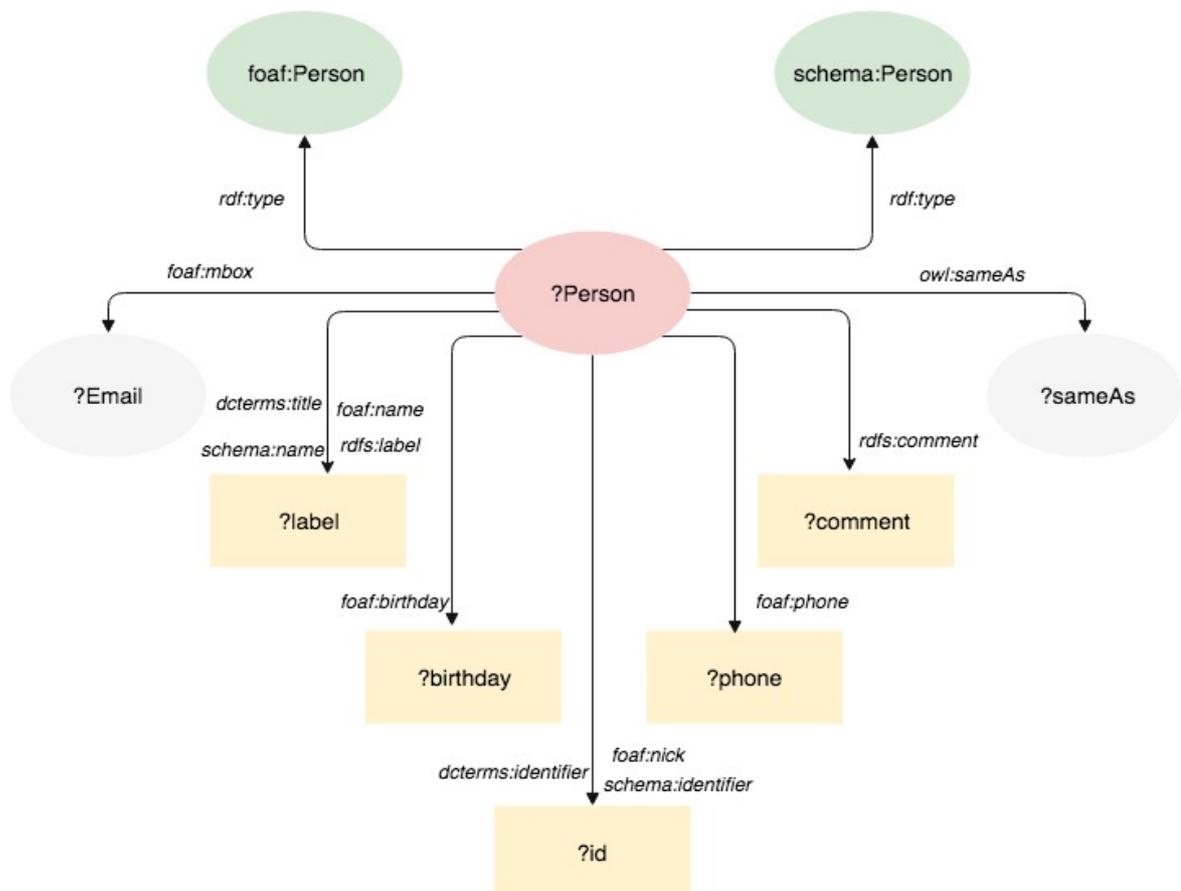
Na Figura 22, as classes representadas em verde pertencem ao módulo SOSA. Tais classes associadas àquelas representadas na cor vermelha compõem o módulo SSN. Enquanto as classes em verde, vermelho e azul estão incorporadas ao módulo System Capability. A modularização da ontologia permite que o desenvolvedor selecione o módulo de interesse de acordo com o grau de expressividade que deseja obter na descrição de seus sistemas.

## 5. MODELO ONTOLÓGICO DO CATÁLOGO

Este capítulo apresenta o uso combinado das ontologias selecionadas na descrição de cada requisito funcional elencado na etapa de elicitação de requisitos da metodologia adotada neste trabalho. Tais 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.

### 5.1. Módulo Pessoa (Person)

Pessoas podem ser vinculadas a diferentes recursos cadastrados no catálogo. Por exemplo: uma dada pessoa pode ser associada a uma implantação, ser responsável pelo registro de informações de sensores ou de atuadores na aplicação, ser o contato principal de uma organização ou, ainda, ser editor responsável de uma dada base de dados de sensores. Dada sua importância para descrição de sensores, atuadores e recursos relacionados, a aplicação prevê o cadastro de pessoas. As ontologias, classes e propriedades utilizadas no modelo para descrição de pessoas estão indicadas na Figura 23.



**Figura 23** - Modelo para descrição de pessoas

Fonte: Elaboração própria

Na Figura 23, o recurso pessoa (*?Person*) é definido pelas classes *foaf:Person* e *schema:Person* por meio da propriedade *rdf:type*. Assim, ao cadastrar uma nova pessoa, o usuário estará cadastrando instâncias das classes *foaf:Person* e *schema:Person*. O uso de classes equivalentes, ou seja, que têm como origem ontologias distintas, mas que possuem a mesma finalidade, implica em maior interoperabilidade no intercâmbio de informações. Esta mesma ideia também pode ser aplicada as propriedades.

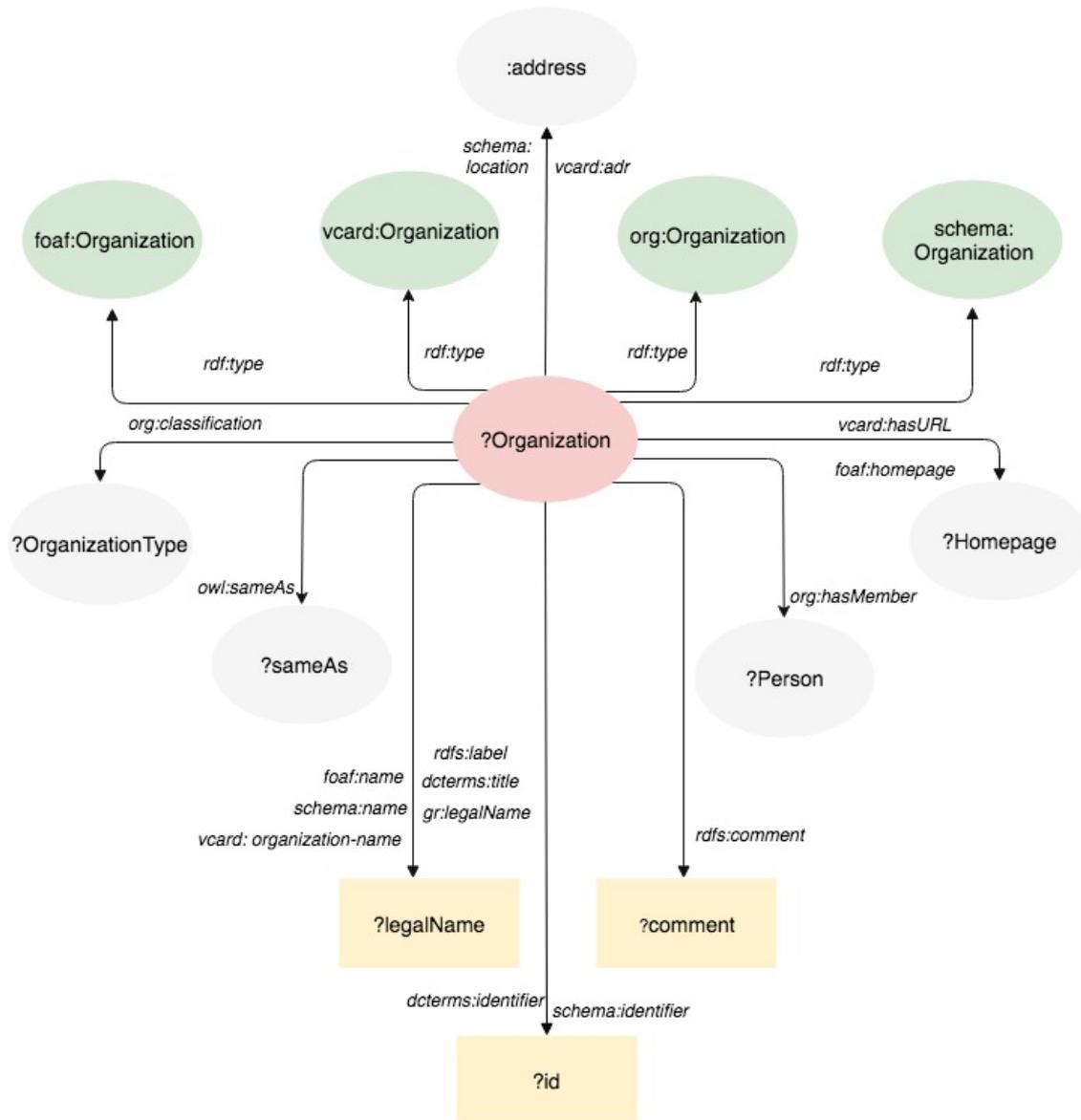
No modelo indicado na Figura 23, as propriedades *foaf:nick*, *schema:identifier* e *dcterms:identifier* são utilizadas para definir o URI do recurso. Além disso, uma pessoa deve possuir um nome formal, descrito por meio das propriedades *foaf:name*, *rdfs:label*, *schema:name* e *dcterms:title*. Outros dados pessoais, tais como data de nascimento, telefone de contato e email podem ser vinculados ao cadastro de pessoas, fazendo uso das propriedades *foaf:birthday*, *foaf:phone* e *foaf:mbox*, respectivamente. No caso do objeto *?Email*, o valor a ser associado deve consistir em endereço de email, ou seja, em recurso identificado por URI (p. ex.: *mailto:example@example.com*). Ainda, um comentário sobre a pessoa a ser catalogada pode ser inserido, na propriedade *rdfs:comment*. A prática de possibilitar a inserção de identificador único, nome e comentário para os recursos é adotada em todos modelos apresentados neste trabalho.

Pessoas podem também ser relacionadas a outros recursos por meio da propriedade *owl:sameAs*, perfazendo o *Linked Data Mashup*. A tripla formada por esta propriedade indica que os recursos associados representam a mesma pessoa, em fontes de dados distintas, seguindo o quarto princípio *Linked Data*. Neste trabalho, especificamente, faz-se uso da base de dados semântica DBpedia para buscar pessoas que possam ser associadas àquelas cadastradas na aplicação. Ao associar recursos da base DBpedia aos recursos cadastrados na aplicação por meio da propriedade *owl:sameAs*, tem-se como resultado o enriquecimento das informações catalogadas.

É importante ressaltar que o uso das propriedades indicadas no modelo da Figura 23, e nos demais modelos que serão apresentados ao longo deste capítulo, levam em consideração as definições de domínio e contradomínio das propriedades adotadas.

## 5.2. Módulo Organização

No catálogo semântico proposto, organizações podem ser responsáveis por produzir determinados modelos de sensores e atuadores ou estar associadas a certas implantações. Nesse caso, a aplicação permite o cadastro de organizações para que tais recursos possam ser reusados ou associados a outros recursos. A Figura 24 apresenta modelo proposto para descrição de organizações.



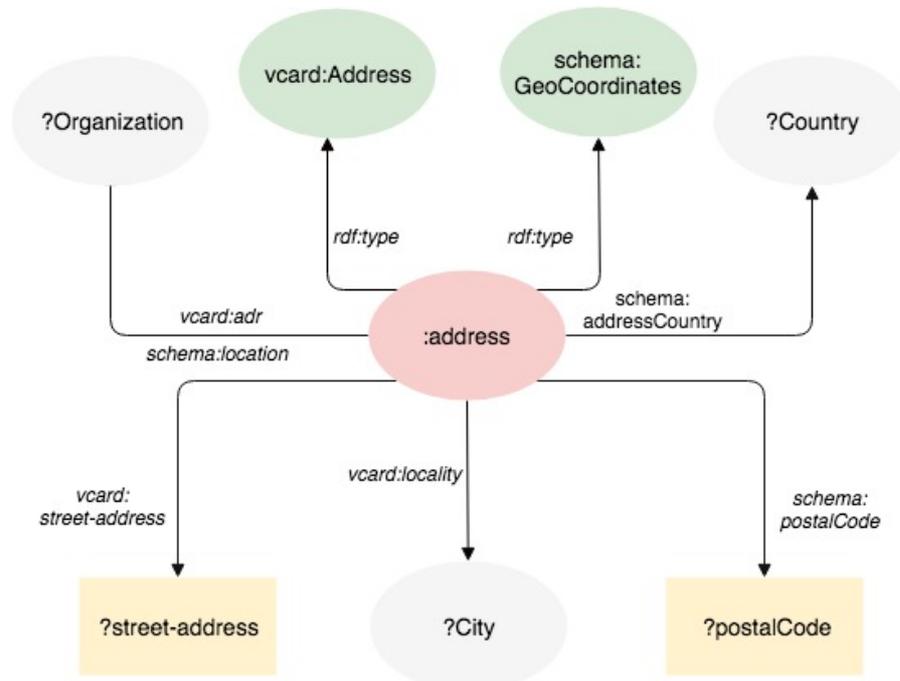
**Figura 24** - Modelo para descrição de organizações

Fonte: Elaboração própria

Novas organizações são representadas pela variável *?Organization* e têm seus URIs gerados a partir das propriedades *schema:identifier* e *dcterms:identifier*. O URI de tais recursos é resultante do valor recebido na variável *?id*. Por questão de interoperabilidade, diferentes propriedades são utilizadas para na descrição do nome (*?label*) de uma organização, tais como *gr:legalName*, *rdfs:label*, *foaf:name*, *dcterms:title* e *vcard:organization-name*. Além de

associar identificador para construção de URI e nome à organização, o usuário pode associar, ainda, informações de pessoa para contato (*?Person*), página Web (*?Homepage*) e URI equivalente (p. ex.: recurso da DBpedia) na variável *?sameAs*. Tudo isso é possível por meio do uso de propriedades *org:hasMember*, *foaf:homepage* e *vcard:hasURL*, *owl:sameAs*, respectivamente.

No modelo proposto, dados de localização de organizações são descritos por meio de propriedades das ontologias Vcard e Schema.Org, fazendo uso de nó em branco (*:address*). A Figura 25 apresenta o modelo adotado para descrição de dados de localização de organizações.



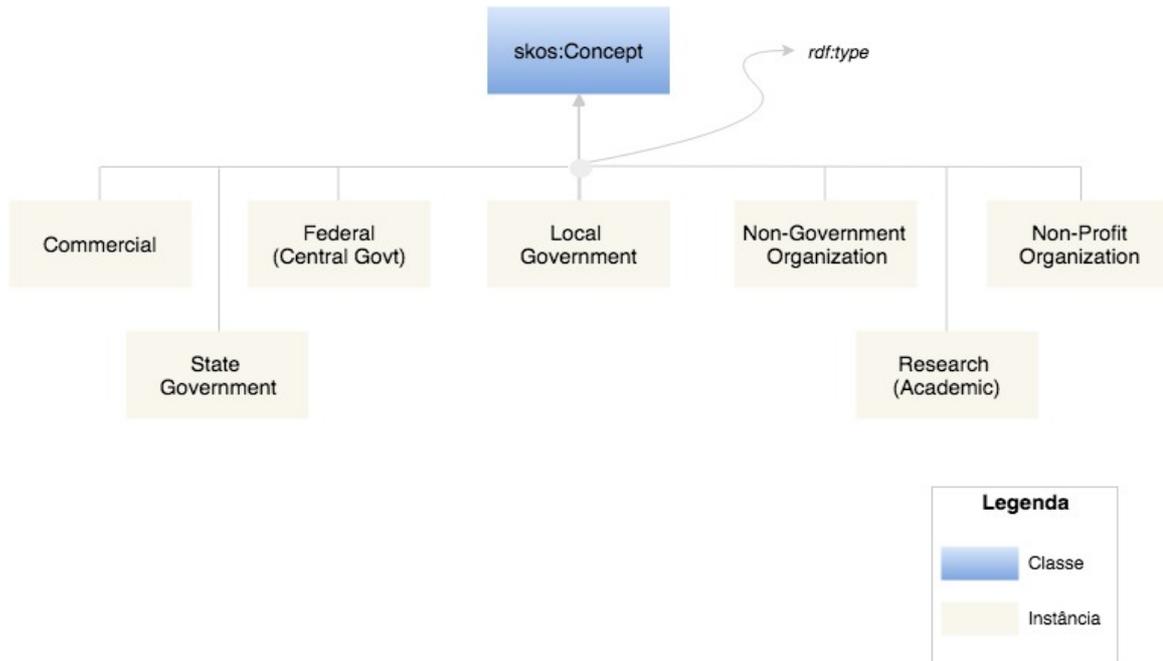
**Figura 25** - Modelo para descrição de dados de localização de organizações

Fonte: Elaboração própria

Na Figura 25, por meio das propriedades *schema:location* e *vcard:adr*, o recurso *?Organization* pode ser vinculado a objeto identificado como *:address*. Tal objeto é definido no modelo com sendo instância das classes *schema:GeoCoordinates* e *vcard:Address* e é composto de diferentes atributos, tais como nome e número da rua (*?street-address*), código postal (*?postalCode*) e cidade (*?City*) e país (*?Country*). Tais atributos são relacionados ao objeto *:address* por meio das propriedades *vcard:street-address*, *schema:postalCode*, *vcard:locality* e *schema:addressCountry*, respectivamente. Como o catálogo proposto não prevê catalogação/descrição de cidades e países, as variáveis *?City* e *?Country* devem receber URIs de recursos de bases de dados de terceiros, tal como os da base de dados DBpedia.

As ontologias utilizadas no modelo para descrição de organizações não compreendem termos para definição dos diferentes tipos de organização. Nesse caso, um tesouro de tipos de

organização é proposto com base na ontologia SKOS. A Figura 26 apresenta os tipos de organização definidos como instâncias (*rdf:type*) da classe *skos:Concept*.

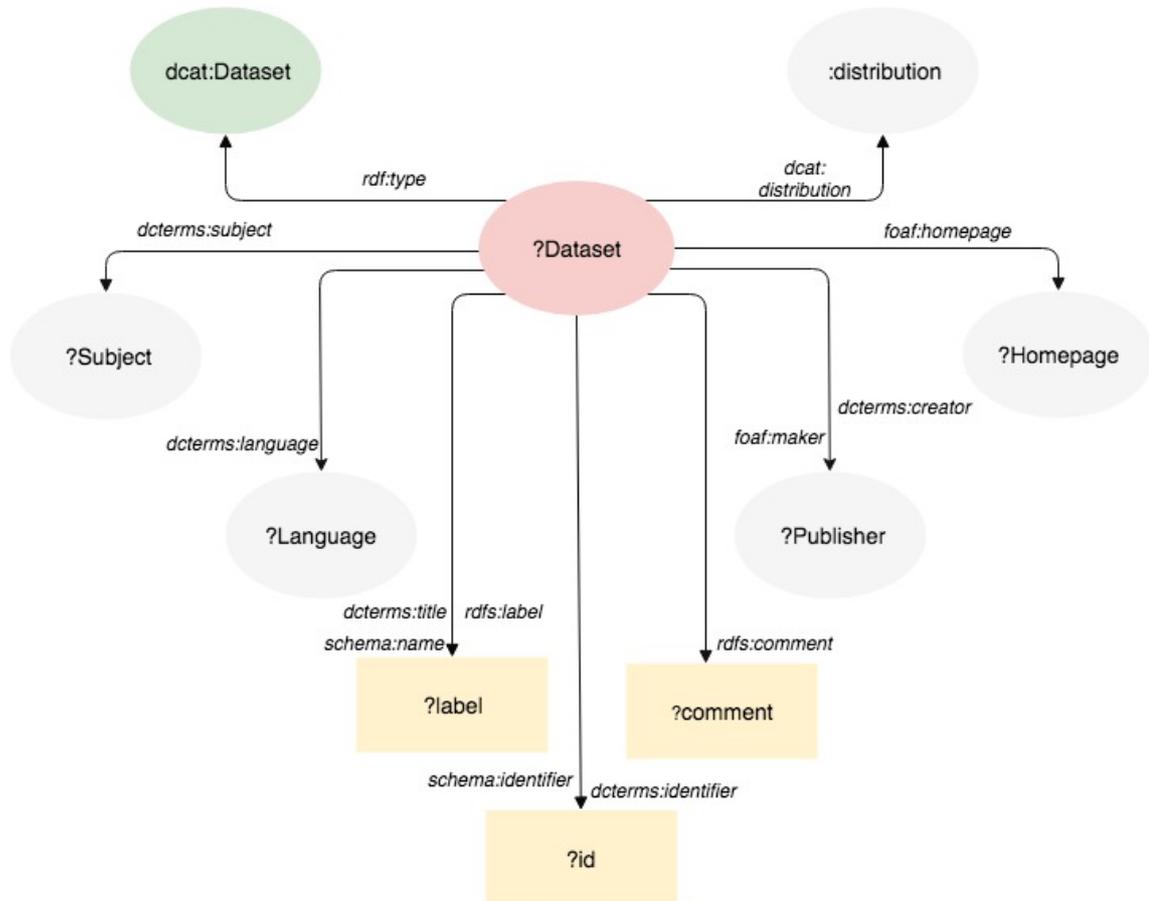


**Figura 26** - Tesouro de tipos de organização  
Fonte: Elaboração própria

No caso do catálogo semântico proposto neste trabalho, instâncias da classe *skos:Concept* são cadastradas pelo desenvolvedor e ficam disponíveis na base de dados da aplicação para serem associadas ao cadastro de uma dada organização. Na Figura 24, a variável *?OrganizationType* representa, portanto, instâncias da classe *skos:Concept* previamente cadastradas na aplicação (Figura 26). Como o desenvolvimento de tesouro de tipos de organização não faz parte do escopo deste trabalho, o tesouro adotado (Figura 26) é inspirado em exemplo apresentado em 3-Round-Stones (2009).

### 5.3. Módulo Base de Dados (Dataset)

Embora o catálogo semântico proposto não preveja o registro de dados de observação de sensores, bases de dados, que armazenam dados de sensores catalogados na aplicação, podem ser cadastradas. Uma vez cadastrados, tais recursos podem ser associados à descrição de sensores. Esta prática facilita o reuso de dados de observação de sensores na Web. A Figura 27 apresenta o modelo proposto para descrição de bases de dados.



**Figura 27** - Modelo para descrição de base de dados  
Fonte: Elaboração própria

Ao cadastrar bases de dados na aplicação, o usuário cadastra instâncias das classe *dcat:Dataset* (Figura 27). A ontologia DCAT define base de dados como sendo um conjunto de dados publicado e disponível para acesso ou *download* em um ou mais formatos. Nesse caso, não é possível garantir que as bases de dados cadastradas na aplicação publiquem seus dados em consonância com os padrões da Web Semântica. Embora existam vocabulários específicos para descrever bases de dados RDF, tal como a ontologia VOID<sup>18</sup>, optou-se pela ontologia DCAT que permite descrever bases de dados que disponibilizam dados em qualquer formato, tendo em vista que dados de observação de sensores são frequentemente encontrados em diversos formatos na Web.

Bases de dados podem estar relacionadas a diversos domínios de conhecimento. Assim, por meio da propriedade *dcterms:subject* é possível esclarecer o assunto ou tópico que uma dada fonte de dados (*?Dataset*) aborda. Nesse caso, a variável *?Subject* representa recursos que têm como origem o famoso tesouro WordNet<sup>19</sup> para termos da língua inglesa. De forma similar,

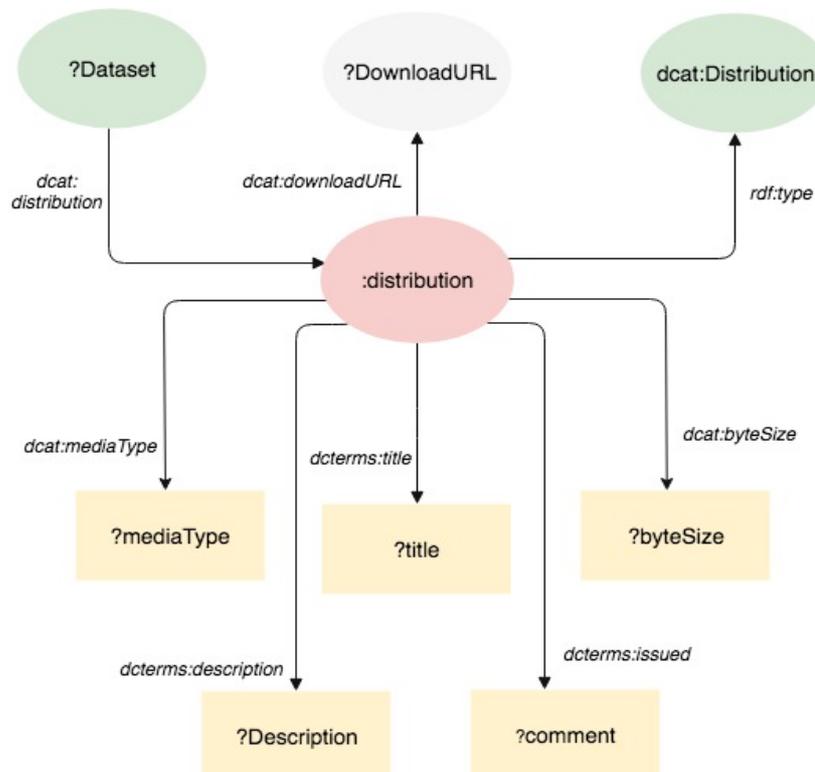
<sup>18</sup> <https://www.w3.org/TR/void/>

<sup>19</sup> <http://wordnet-rdf.princeton.edu/>

recursos da base de dados Library of Congress<sup>20</sup> podem ser associadas ao cadastro de uma dada fonte de dados de sensores. Assim, a variável *?Language* representa as diferentes línguas nas quais uma dada base de dados publica seus dados. Para tornar explícita tal informação faz-se uso da propriedade *dcterms:language*.

Uma fonte de dados pode, ainda, ser associada a uma página Web (*?Homepage*) por meio da propriedade *foaf:homepage* e a um editor (*?Publisher*) responsável pelas publicações fazendo uso das propriedades *foaf:maker* e *dcterms:creator*. No caso de editor (*?Publisher*) de bases de dados, tal recurso pode consistir em pessoas ou organizações.

Comumente, bases de dados distribuem informações em arquivos de diferentes formatos. Nesse caso, cada forma de representação dos dados pode ser definida como instância da classe *dcat:Distribution* e descritas por características específicas, tais como tamanho do arquivo, *link* para *download*, tipo de mídia (formato), etc. No modelo proposto, bases de dados são relacionadas, portanto, a nó em branco do tipo *dcat:Distribution* e identificado como *:distribution*, que por sua vez, inclui diferentes atributos capazes de descrever a forma na qual dados são representados em uma dada fonte de dados. Exemplos de “distribuição” incluem arquivos CSV, RDF, uma API, etc. O modelo proposto para descrever formatos de publicação é apresentado na Figura 28.



**Figura 28** - Modelo para descrição de formatos de publicação

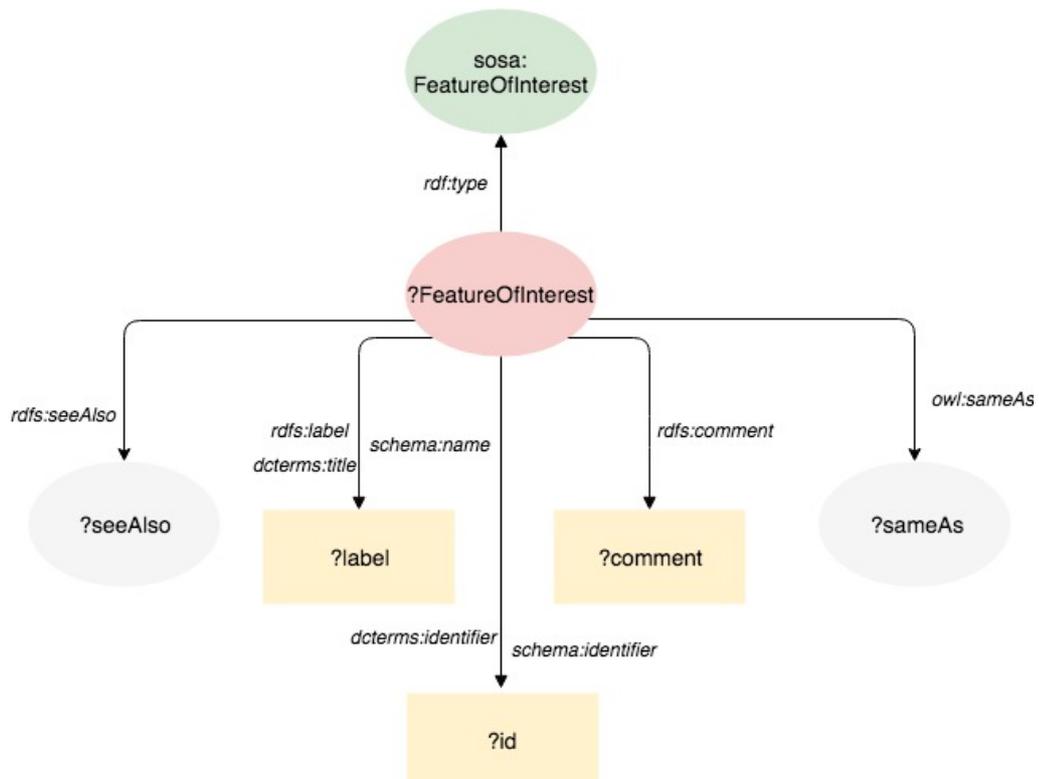
Fonte: Elaboração própria

<sup>20</sup> <http://www.loc.gov/>

Na Figura 28, para descrever um dado formato de publicação, diferentes propriedades são utilizadas - a propriedade *dcat:mediaType* e *dcat:byteSize* descrevem o tipo (*?mediaType*) e o tamanho (*?byteSize*) do arquivo, respectivamente. Já a propriedade *dcat:downloadURL* explicita o URL (*?DownloadURL*) na Web onde o *download* do arquivo pode ser realizado. Além disso, o usuário pode associar título (*?title*), data de criação (*?Issued*) e descrição (*?Description*) do arquivo publicado. Isso é possível, ao fazer uso das propriedades *dcterms:tittle*, *dcterms:issued* e *dcterms:description* no modelo.

#### 5.4. Módulo Objeto de Interesse (Feature Of Interest)

Objetos de interesse podem ser entendidos como entidades observadas por um sensor ou, ainda, modificadas por um atuador. Por fazerem parte do contexto de sensores e atuadores, a ontologia SSN define uma classe específica para caracterizar objetos de interesse: *sosa:FeatureOfInterest*. A título de exemplo, em uma observação do peso de uma dada pessoa, a pessoa pode ser classificada como objeto de interesse e peso como uma propriedade da pessoa. A Figura 29 apresenta o modelo proposto para descrição de objetos de interesse.



**Figura 29** - Modelo para descrição de objetos de interesse

Fonte: Elaboração própria

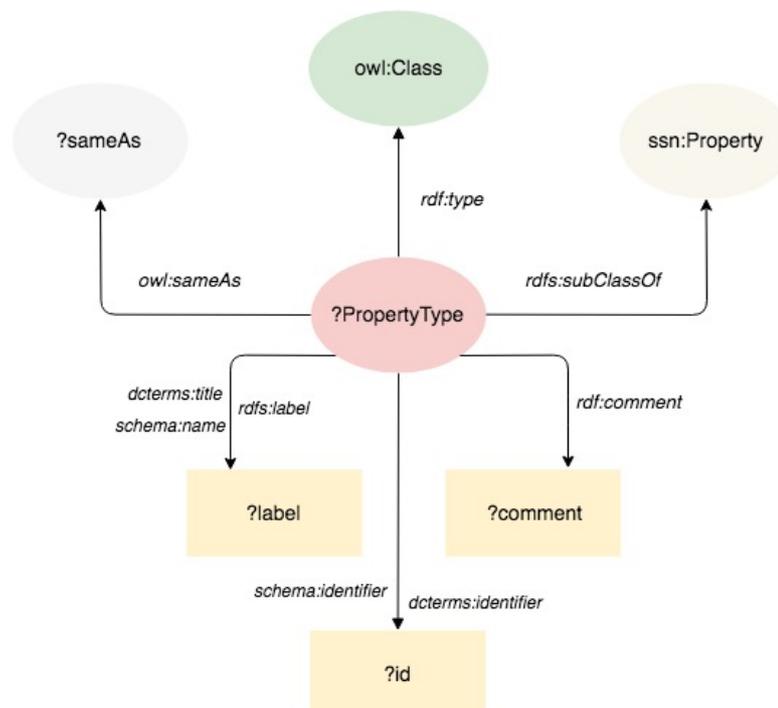
Na Figura 29, o recurso pessoa (*?FeatureOfInterest*) é definido pela classe *sosa:FeatureOfInterest*, por meio da propriedade *rdf:type*. Assim, ao cadastrar um novo objeto de interesse, o usuário estará cadastrando instâncias da classe *sosa:FeatureOfInterest*. As

propriedades *schema:identifier* e *dcterms:identifier* são utilizadas para definir o URI do recurso.

No modelo apresentado na Figura 29, instâncias da classe *sosa:FeatureOfInterest* podem ser associadas a recursos relacionados (p. ex.: *mashup* com a DBpedia) por meio das propriedades *owl:sameAs* e *rdfs:seeAlso*. O uso da propriedade *owl:sameAs* indica forte relação entre o recurso cadastrado (*?FeatureOfInterest*) e o recurso associado (*?sameAs*), a medida que define tais recursos como sendo o mesmo recurso, ainda que sejam identificados por URIs distintos. Já a propriedade *rdfs:seeAlso*, reflete a existência de informação adicional sobre o recurso descrito (*?FeatureOfInterest*).

### 5.5. Módulo Tipo de Propriedade (Property Type)

Este módulo tem como propósito apresentar modelo para descrição de tipos de propriedades, ou seja, propriedades que não são dotadas de objeto de interesse. A ideia é que estes tipos de propriedades sirvam como base para construção de propriedades mais específicas. Por exemplo, diversas propriedades podem ser derivadas do tipo de propriedade “Temperatura”, tais como “Temperatura de ar”, “Temperatura de uma sala”, “Temperatura do corpo humano”, etc. A Figura 30 apresenta detalhes sobre o modelo para descrição de tipos de propriedade.



**Figura 30** - Modelo para descrição de tipos de propriedades

Fonte: Elaboração própria

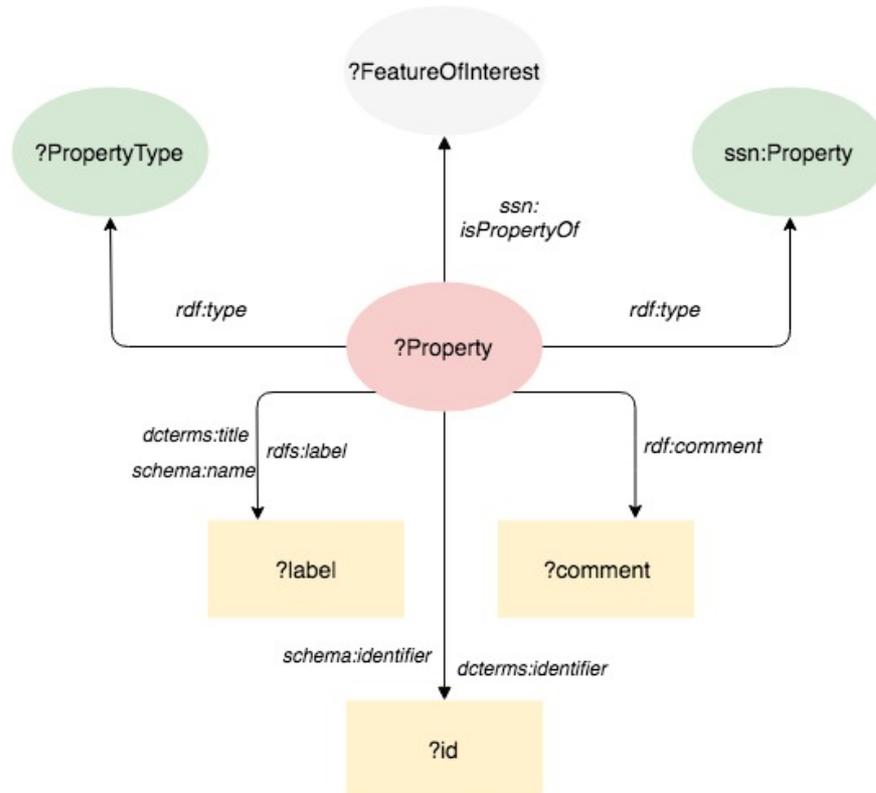
Na Figura 30, os novos tipos de propriedade são representados pela variável *?PropertyType*. Tais recursos têm seus URIs gerados com base no identificador associado, nas propriedades *schema:identifier* e *dcterms:identifier*. Ao cadastrar novos tipos de propriedades, o usuário estará cadastrando uma nova subclasse direta de *ssn:Property*, o que implica na extensão da ontologia SSN. Por exemplo, o usuário pode cadastrar tanto um tipo de propriedade “Temperatura” cujas instâncias serão temperaturas de objetos de interesse observadas por sensores, quanto o tipo de propriedade “Vazão” cujas instâncias serão vazões de objetos de interesse manipuladas por atuadores. Desse modo, no momento do cadastro, não há diferenciação entre as naturezas dos tipos de propriedade (se são de observação ou atuação). No entanto, posteriormente, quando uma propriedade (instância do tipo de propriedade cadastrado) de um objeto de interesse específico for usada e associada a sensores e/ou atuadores, por inferência com base nos axiomas da ontologia SSN, esta propriedade específica também será automaticamente classificada como sendo instância de *ssn-system:ObservableProperty* e/ou *ssn-system:ActuatableProperty*. Por exemplo, se a propriedade de objeto de interesse “Temperatura do ar” (instância do tipo de propriedade “Temperatura”) for vinculada à descrição de um sensor, o sistema irá inferir que a propriedade “Temperatura do ar” é do tipo *system:ObservableProperty*.

No modelo ontológico proposto, os diferentes tipos de propriedade (*?PropertyType*) podem ser associados a recurso equivalente (*?sameAs*) por meio da propriedade *owl:sameAs*, perfazendo o *Linked Data Mashup*. Tais recursos (equivalentes) podem ter como origem, por exemplo, a base de dados DBpedia.

## 5.6. Módulo Propriedade de um Objeto de Interesse (Property Of a Feature Of Interest)

O processo de observação de uma propriedade realizado por um sensor tem sempre um objeto de interesse envolvido, ou seja, a "coisa" cuja propriedade se quer observar. Por exemplo: um dado modelo de sensor pode ser utilizado para observar a "Temperatura do ar". Nesse caso, o recurso "Temperatura do ar" é a propriedade (*?Property*) cadastrada. Além disso, o recurso "Temperatura do ar" é uma instância da classe "Temperatura" que é um tipo de propriedade (*?PropertyType*) cadastrado anteriormente. O recurso "Temperatura do ar" também é uma instância da classe *ssn:Property*, que é superclasse da classe "Temperatura". Por fim, o recurso "Temperatura do ar" está associado ao recurso "ar", que é objeto de interesse (*?FeatureOfInterest*) cadastrado anteriormente. Assim, é fundamental relacionar os diferentes tipos de propriedades a elementos de interesse, definido pois uma propriedade a ser observada por sensores ou manipulada por atuadores. Neste contexto, o modelo proposto possibilita

vincular os tipos de propriedades (*?PropertyType*) a objetos de interesse (*?FeatureOfInterest*) cadastrados previamente na aplicação. A Figura 31 apresenta mais detalhes do modelo em questão.



**Figura 31** - Modelo para descrição de propriedades associadas a objeto de interesse  
Fonte: Elaboração própria

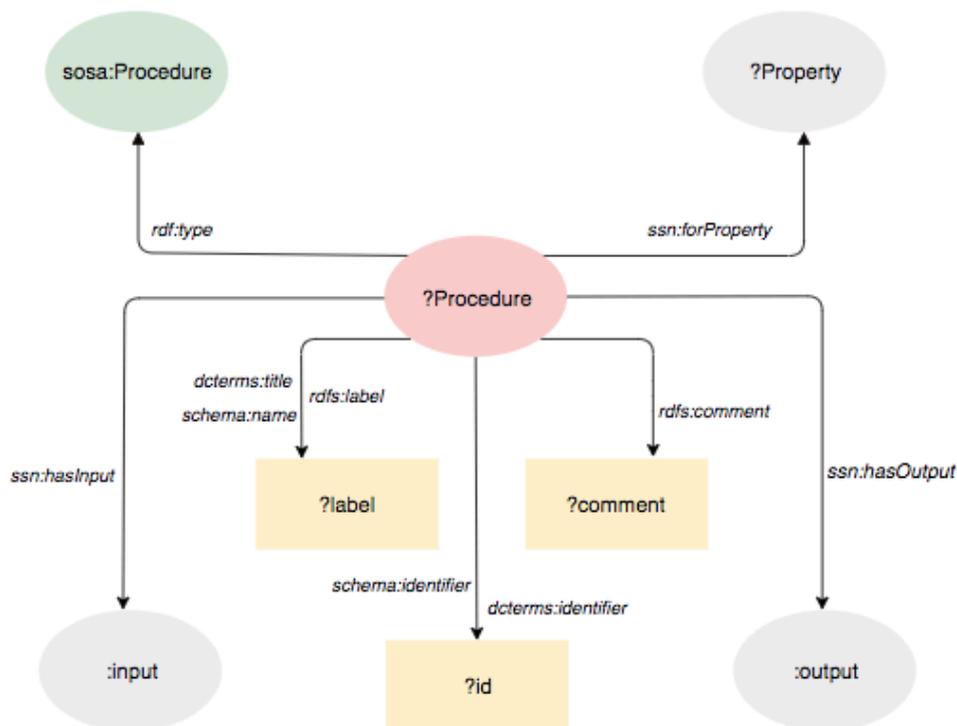
No modelo da Figura 31, o URI de uma nova propriedade de objeto de interesse (*?Property*) é gerado com base no valor recebido na variável *?id*, nas propriedades *schema:identifier* e *dcterms:identifier*. Neste mesmo modelo, é possível verificar que propriedades de objeto de interesse (*?Property*) não são definidas pelas classes *system:ObservableProperty* e/ou *ssn-system:ActuatableProperty*, sendo classificadas, portanto, de forma genérica pelas classes *?PropertyType* e *ssn:Property*. Assim, ao criar uma nova propriedade, representada pela variável *?Property*, o usuário cadastra, além de uma instância da classe *ssn:Property*, uma instância de *?PropertyType*. Ao vincular a propriedade (*?Property*) a objeto de interesse (*?FeatureOfInterest*), fazendo uso propriedade *ssn:isPropertyOf*, garante-se a especificidade da propriedade em questão.

Embora a ontologia SSN apresente as classes *ssn-system:ObservableProperty* e *ssn-system:ActuatableProperty* para classificar e distinguir as propriedades de observação (de sensores) das propriedades de atuação (de atuadores), no momento do cadastro de propriedades de objeto de interesse, não há necessidade de fazer distinção entre estas duas classes, deixando esta distinção a cargo de inferência ontológica. Assim, o modelo apresentado na Figura 31

possibilita a catalogação de diferentes propriedades que serão definidas como sendo instâncias de *ssn-system:ObservableProperty* ou *ssn-system:ActuatableProperty* por inferência. Por exemplo, ao associar uma dada propriedade (*?Property*) a um sensor (*?Sensor*) por meio da propriedade *sosa:observes*, o sistema irá inferir que trata-se de uma propriedade de observação (*ssn-system:ObservableProperty*).

### 5.7. Módulo Procedimento (Procedure)

Observações realizadas por sensores, geograficamente distribuídos, podem ser fruto de um mesmo procedimento (*?Procedure*). Por este motivo, informações sobre procedimentos de observação de sensores são atrelados ao modelo de sensor (*?SensorModel*). Assim, tais dados podem ser reusados por diferentes exemplares (*?Sensor*) que compartilham o mesmo modelo. Por exemplo, a velocidade do vento medida pode variar dependendo da altura do sensor acima da superfície. Consequentemente, os procedimentos para medir a velocidade do vento definem uma altura padrão para anemômetros acima do solo. Tais procedimentos são válidos para todos os exemplares de anemômetros que compartilham o mesmo modelo. A Figura 32 oferece detalhes sobre o modelo proposto para descrição de procedimentos.

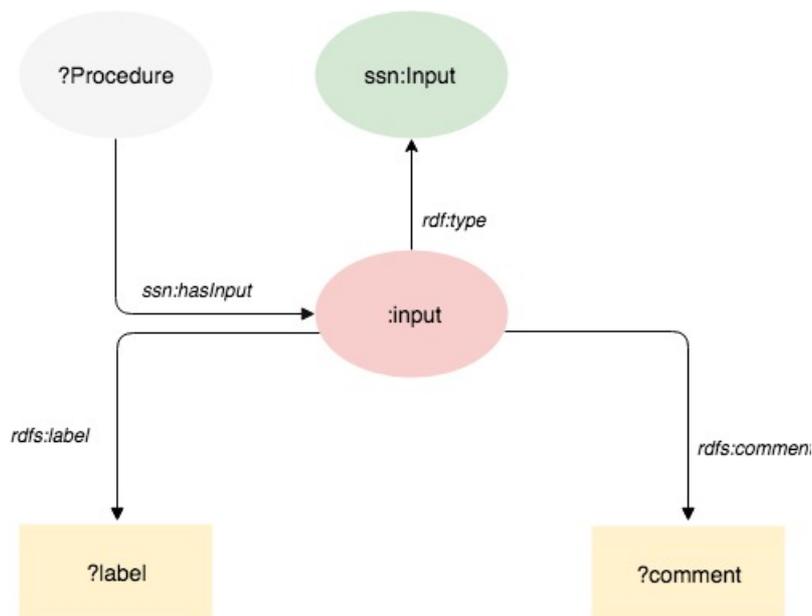


**Figura 32** - Modelo para descrição de procedimentos

Fonte: Elaboração própria

Na Figura 32, procedimentos são definidos como instâncias da classe *sosa:Procedure*. O URI do procedimento (*?Procedure*) é gerado com base no valor recebido na variável *?id*, nas propriedades *schema:identifier* e *dcterms:identifier*.

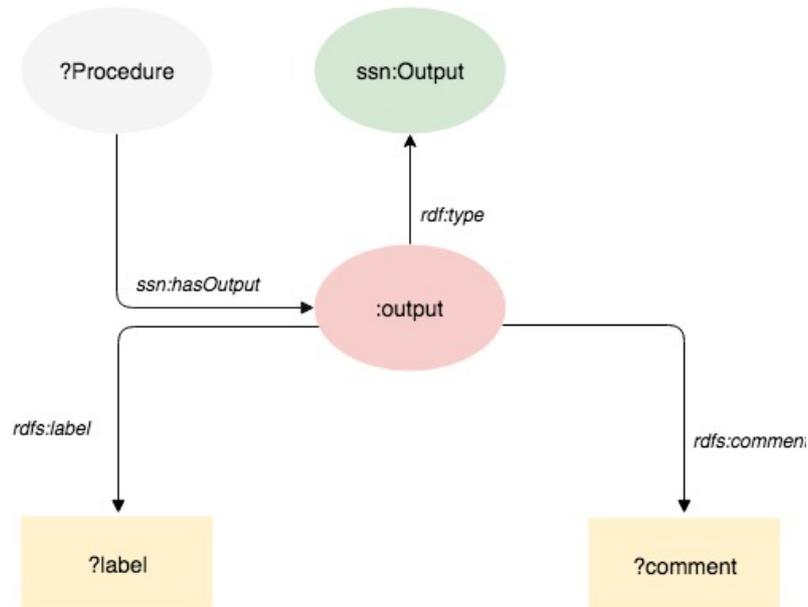
Sensores que observam mais de uma propriedade de objeto de interesse (*?Property*) podem conter procedimentos distintos para cada uma das propriedades observadas. Assim, no modelo apresentado na Figura 32 faz-se uso da propriedade *ssn:forProperty* com o propósito de diferenciar os procedimentos de um dado modelo de sensor. Além disso, procedimentos podem conter, ainda, informações de entrada, ou seja, parâmetros de entrada. Tais parâmetros são definidos como *ssn:Input* e representados como nó em branco (*:input*) no modelo. A relação entre um procedimento e seus parâmetros de entrada é dada pela propriedade *ssn:hasInput*. Além disso, parâmetros de saída (resultados) de um procedimento são definidos como *ssn:Output* e também são representados como nó em branco (*:output*). Já a relação entre um procedimento e seus resultados é definida pela propriedade *ssn:hasOutput*. As Figuras 33 e 34 apresentam o modelo para descrição das informações de entrada (*:input*) e saída (*:output*) de procedimentos.



**Figura 33** - Modelo para descrição de parâmetros de entrada

Fonte: Elaboração própria

A Figura 33 evidencia a propriedade utilizada para relacionar procedimentos aos seus parâmetros de entrada, a saber: *ssn:hasInput*. De forma similar, o modelo de descrição parâmetros de saída de procedimentos evidencia a propriedade *ssn:hasOutput*, vide Figura 34.



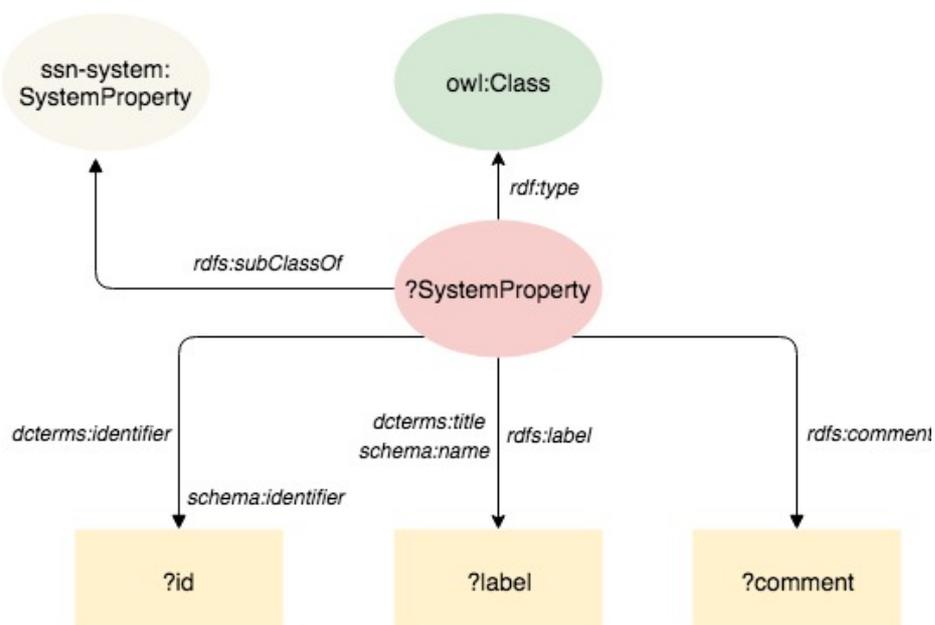
**Figura 34** - Modelo para descrição de parâmetros de saída

Fonte: Elaboração própria

Nas Figuras 33 e 34, como tratam-se de nós em branco, os parâmetros de entrada (*:input*) e saída (*:output*) são intrínsecos à descrição ou a catalogação do procedimento.

### 5.8. Módulo Propriedade de Sistema (System Property)

O sistema para catalogação de sensores proposto não prevê apenas a utilização de propriedades pré-existentes na ontologia SSN. A aplicação permite que a ontologia SSN seja estendida por meio do cadastro de novas propriedades de sistema. A Figura 35 apresenta o modelo proposto para descrição de propriedades de sistema.



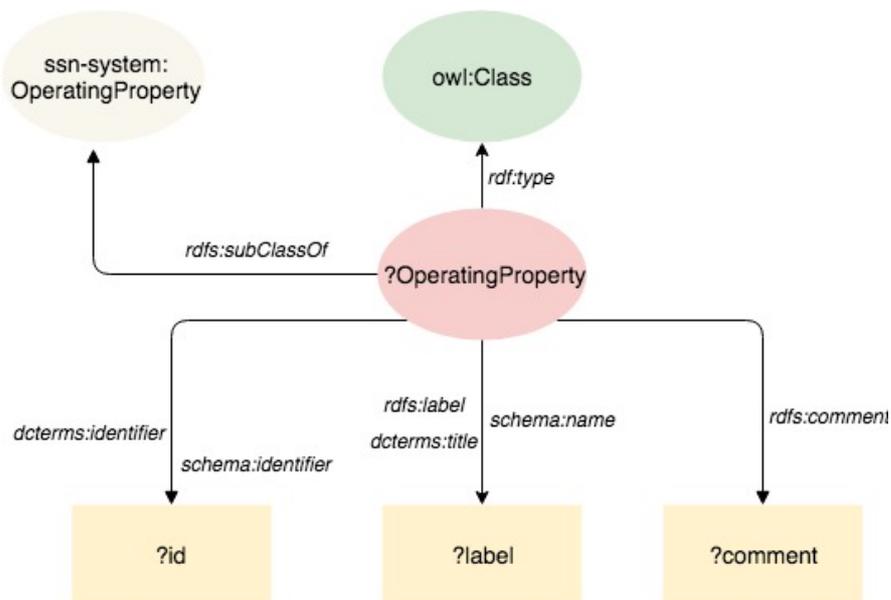
**Figura 35** - Modelo para descrição de propriedades de sistema

Fonte: Elaboração própria

Na Figura 35, ao definir o recurso *?SystemProperty* como sendo *owl:Class* e subclasse de *ssn-system:SystemProperty*, novas subclasses de propriedade de sistema podem ser criadas pelo usuário para serem, posteriormente, associadas ao cadastro de modelos de sensor ou atuador. O URI do recurso *?SystemProperty* é criado por meio das propriedades *schema:identifier* e *dcterms:identifier*. O usuário pode, ainda, associar um nome (*?label*) para tal recurso por meio das propriedades *rdfs:label*, *schema:name* e *dcterms:title* e vincular um comentário sobre a propriedade em questão através da propriedade *rdfs:comment*.

### 5.9. Módulo Propriedade de operação (Operating Property)

Com o intuito de estender a ontologia, além das propriedades de sistema, propriedades de operação também podem ser cadastradas no catálogo. A Figura 36 apresenta o modelo proposto para descrição de propriedade de operação de um modelo de sensor.

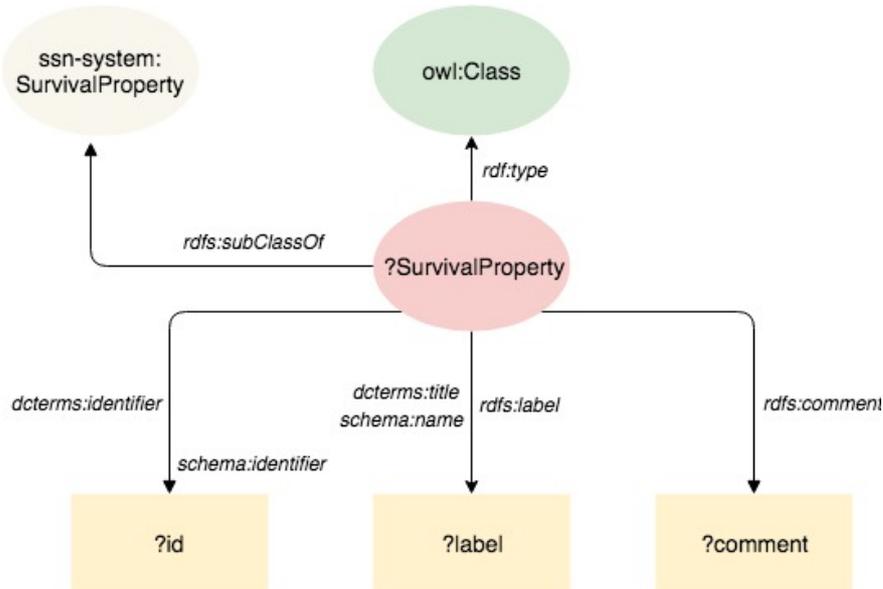


**Figura 36** - Modelo para descrição de propriedades de operação  
Fonte: Elaboração própria

O modelo para descrição de propriedades de operação proposto é idêntico àquele utilizado para descrição de propriedades de sistema (Figura 36), tendo como diferença, apenas, a natureza da propriedade. Nesse caso, o usuário pode cadastrar novas subclasses de *ssn-system:OperatingProperty*. O URI deste recurso é criado por meio das propriedades *schema:identifier* e *dcterms:identifier*. Ainda, um nome pode ser vinculado à propriedade de sistema por meio das propriedades *rdfs:label*, *dcterms:title* e *schema:name*.

### 5.10. Módulo Propriedade de sobrevivência (Survival Property)

O modelo para descrição de propriedades de sobrevivência proposto é idêntico àquele utilizado para descrição de propriedades de sistema (Figura 35) e propriedades de operação (Figura 36), tendo como diferença, apenas, a natureza da propriedade. Assim, propriedades de sobrevivência podem ser catalogadas na plataforma com base no modelo apresentado na Figura 37.

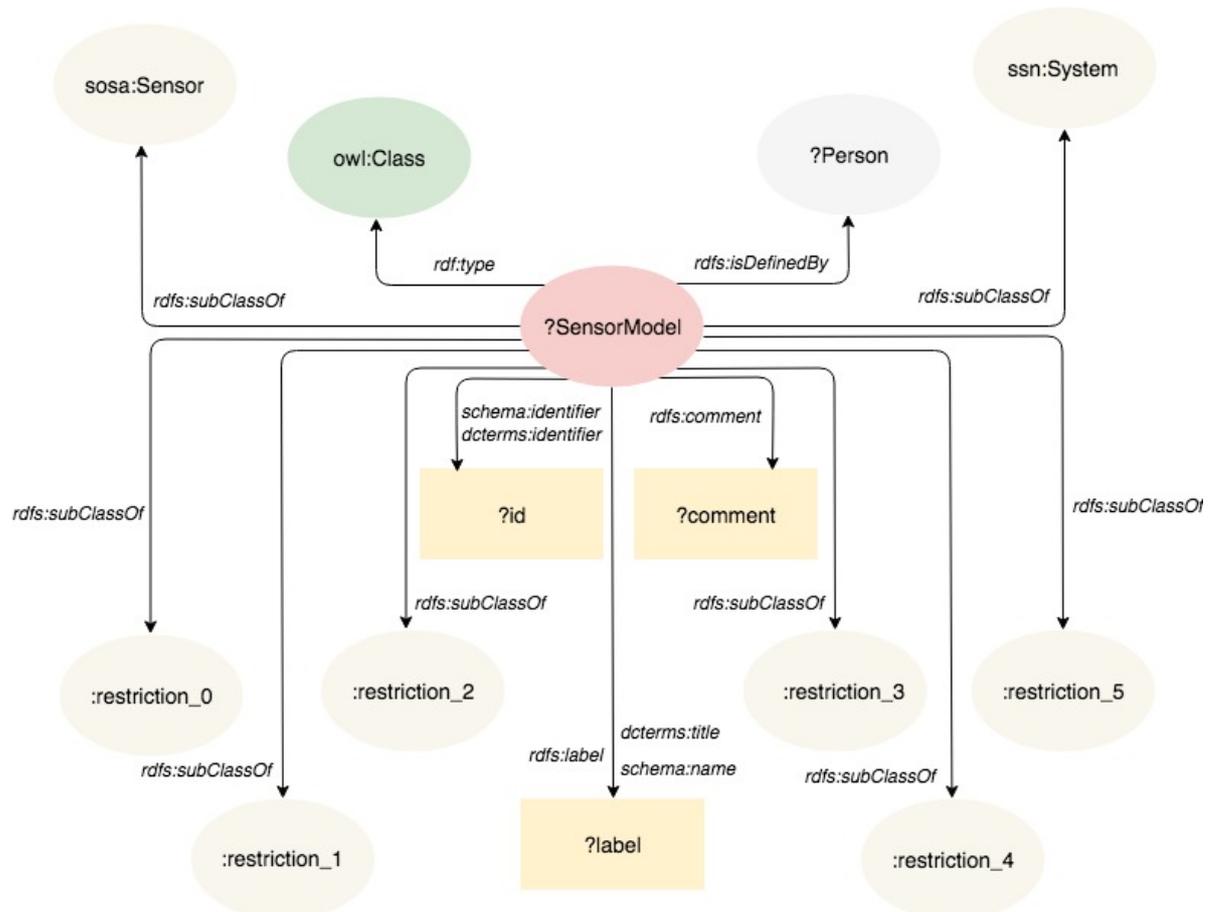


**Figura 37** - Modelo para descrição de propriedades de sobrevivência  
Fonte: Elaboração própria

Na Figura 37, ao cadastrar uma nova propriedade de sobrevivência, o usuário estará cadastrando uma nova subclasse de *ssn-system:SurvivalProperty*. O URI deste novo recurso é criado por meio das propriedades *schema:identifier* e *dcterms:identifier*.

### 5.11. Módulo Modelo Sensor (Sensor Model)

O modelo de um sensor representa o modelo propriamente dito, com a descrição de suas especificações e capacidades técnicas. Sob o ponto de vista formal, este modelo é definido pela classe (conjunto) de todos os exemplares (instâncias) deste modelo. A Figura 38 apresenta o modelo proposto para descrição de modelo de sensores.



**Figura 38** - Modelo para descrição de modelo de sensor

Fonte: Elaboração própria

Na Figura 38, o recurso principal a ser cadastrado, o modelo de sensor, é representado pela variável *?SensorModel*. O modelo de sensor cadastrado tem seu URI gerado a partir do valor nas propriedades *schema:identifier* e *dcterms:identifier*. Um nome pode ser associado a um dado modelo de sensor por meio das propriedades *rdfs:label*, *dcterms:title* e *schema:name*. Além disso, tais recursos podem ser vinculados a pessoas responsáveis por catalogar as informações de tais dispositivos (*?Person*). O relacionamento entre um modelo de sensor (*?SensorModel*) e uma pessoa (*?Person*) é dado pela propriedade *rdfs:isDefinedBy*. O modelo para descrição de modelo de sensores proposto define *?SensorModel* como sendo do tipo *owl:Class* e subclasse de *sosa:Sensor* e *ssn:System*. Nesse caso, ao cadastrar um modelo de sensor no catálogo, o usuário está cadastrando novas classes de modelos de sensores, estendendo a ontologia SSN.

Uma das peculiaridades da ontologia SSN é a não definição do domínio (*domain*) e contradomínio (*range*) de muitas de suas propriedades. Na verdade, restrições para definição de domínio e contradomínio das propriedades são vinculadas diretamente nas classes que fazem uso destas propriedades. Por exemplo, ao vincular um modelo de sensor

(*?SensorModel*) a um dado recurso por meio da propriedade *ssn:implements*, por inferência, o recurso associado será definido como uma instância de *sosa:Procedure*. Isso é possível, uma vez que a definição do contradomínio da propriedade *ssn:implements* está inserida na classe *ssn:System*. Estas e outras restrições atreladas às classes *ssn:System* e *sosa:Sensor* são automaticamente herdadas pela classe *?SensorModel* (por ser subclasse de *ssn:system* e *sosa:Sensor*). No entanto, para que o valor do contradomínio das propriedades utilizadas para descrever modelos de sensores seja compartilhado com as instâncias da classe *?SensorModel*, restrições específicas devem ser criadas dentro desta nova classe de modelos de sensores.

Na Figura 38, é possível verificar seis restrições apresentadas como nós em branco do tipo *owl:Restriction*. À medida que a classe modelo de sensor (*?SensorModel*) é definida como subclasse de cada uma das restrições, tais restrições passam fazer parte desta classe. Cada uma destas restrições tem como propósito definir valor de uma propriedade (restrição de valor - *owl:hasValue*), dentro da classe *?SensorModel*. A Tabela 19 organiza tais restrições com base em suas funções.

**Tabela 19** - Restrições e suas propriedades na classe *?SensorModel*

<b>Restrição</b>	<b>Função</b>
:restriction_0	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn-system:hasSystemCapability</i>
:restriction_1	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn-system:hasOperatingRange</i>
:restriction_2	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn-system:hasSurvivalRange</i>
:restriction_3	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>sosa:observes</i>
:restriction_4	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn:implements</i>
:restriction_5	Restrição de valor ( <i>owl:hasValue</i> ) nas propriedades <i>dcterms:creator</i> e <i>foaf:maker</i>

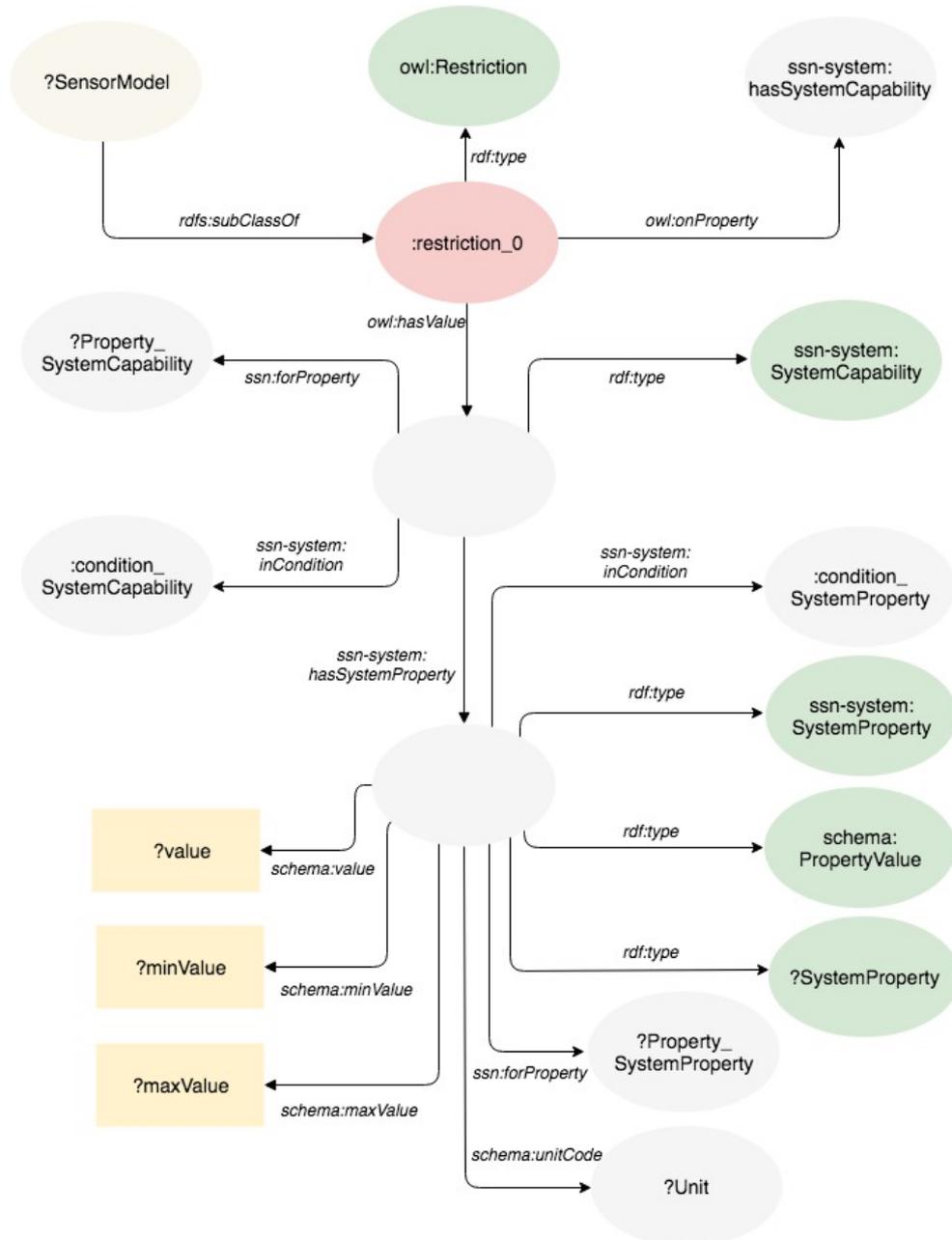
Fonte: Elaboração própria

Os valores das propriedades (indicadas na Tabela 19), uma vez definidos na classe *?SensorModel*, serão atribuídos aos exemplares (instâncias) de *?SensorModel* por inferência. Assim, as propriedades indicadas na Tabela 19 foram selecionadas com base na necessidade de compartilhar seus respectivos valores com os exemplares dos modelos. Por este motivo, a propriedade *rdfs:isDefinedBy* não foi incluída na Tabela 19 – a pessoa responsável por cadastrar as informações do modelo não é, necessariamente, a mesma pessoa que cadastra os metadados dos exemplares desses modelos no catálogo proposto. As Figuras 39 e 40 oferecem maiores detalhes sobre a restrição *:restriction\_0*, indicada na Figura 38.

- Restrição “:restriction\_0”

A ontologia SSN define capacidades de sistema como sendo habilidades que afetam a finalidade principal do sistema (p. ex.: o processo de observação realizado por um sensor). A classe que representa tais capacidades, na ontologia SSN, é a classe *ssn-system:SystemCapability*. Esta classe relaciona-se com as classes *ssn-system:SystemProperty* e *ssn-system:Condition* por meio das propriedades *ssn-system:hasSystemProperty* e *ssn-system:inCondition*, respectivamente. Portanto, a capacidade de um sistema está relacionada com suas propriedades de sistema sob determinadas condições. Por exemplo: a precisão de um sensor é dada sob uma determinada condição. Em conjunto, estas duas propriedades impactam diretamente a habilidade de um sensor em executar uma observação. Portanto, o desempenho esperado de um sensor só pode ser garantido quando submetido a condições préestabelecidas pelo fabricante.

A capacidade de sistema de um modelo de sensor pode conter uma ou mais propriedades de sistema. Para vincular um modelo de sensor à sua capacidade de sistema, que por sua vez deve conter uma ou mais propriedades de sistema, faz-se uso da propriedade *ssn-system:hasSystemCapability*. Nesse caso, é necessário definir o valor da propriedade *ssn-system:hasSystemCapability* na nova classe *?SensorModel*. Para isto, uma restrição, nó em branco do tipo *owl:Restriction*, na propriedade *ssn-system:hasSystemCapability* deve ser criada. O valor desta restrição, dado pela propriedade *owl:hasValue*, consiste no valor da propriedade *ssn-system:hasSystemCapability*. Ao definir a restrição *:restriction\_0* como sendo superclasse da classe *?SensorModel*, tal restrição passa fazer parte classe em questão, por herança. Como resultado, o valor da propriedade será automaticamente associado, por inferência, a todas instâncias da classe *?SensorModel*, sem haver necessidade, portanto, de cadastrar tal informação repetidamente no cadastrado de cada exemplar (instância) do modelo. A Figura 39 apresenta maiores detalhes sobre a restrição *:restriction\_0*.



**Figura 39** - Restrição na propriedade *ssn-system:hasSystemCapability*

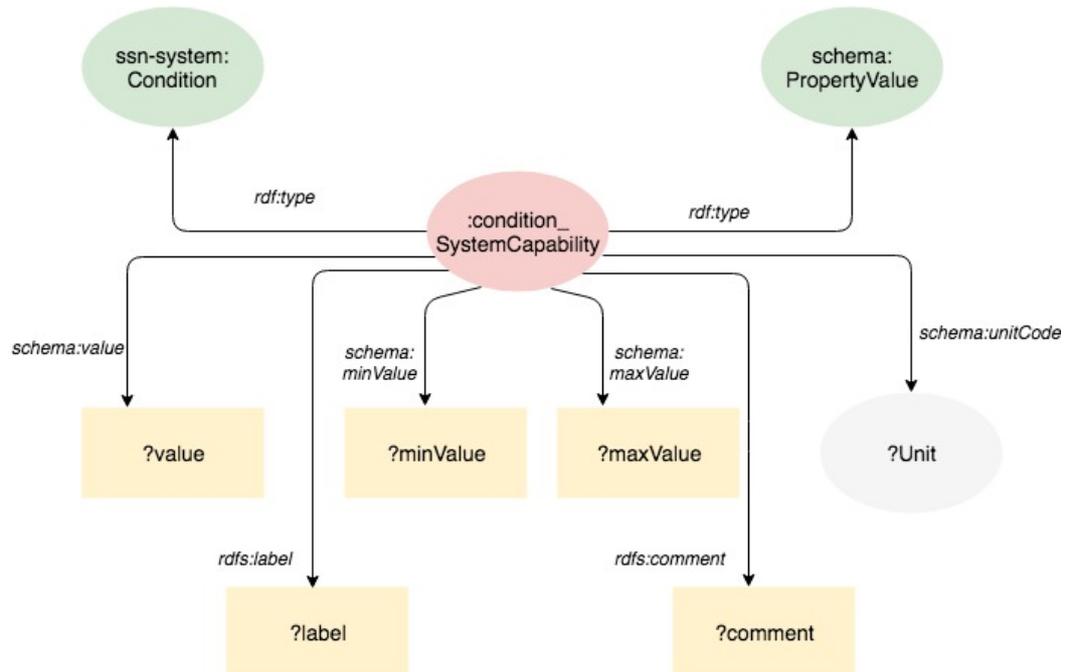
Fonte: Elaboração Própria

Na Figura 39, pode-se verificar que o valor definido para propriedade *ssn-system:hasSystemCapability* é uma instância da classe *ssn-system:SystemCapability*. Tal instância é representada no modelo como nó em branco e pode possuir uma ou mais propriedades de sistema, por meio da propriedade *ssn-system:hasSystemProperty*, cujo valor também é um nó em branco, instância das classes *ssn-system:SystemProperty*, *schema:PropertyValue* e *?SystemProperty*.

A variável *?SystemProperty* deve armazenar propriedades de sistema pré-definidas na ontologia SSN ou aquelas cadastradas previamente no catálogo, tendo em vista que a catalogação de novas propriedades de sistema consiste em uma das funcionalidades da

aplicação. Assim, a definição de objeto (Figura 39) como sendo uma instância da classe *ssn-system:SystemProperty* é puramente enfática, a medida que associá-lo a pelo menos uma propriedade de sistema (*?SystemProperty*) já seria suficiente para torná-lo uma instância da classe *ssn-system:SystemProperty*, por inferência. Já sua definição como sendo uma instância da classe *schema:PropertyValue* possibilita a atribuição de valores numéricos. Assim, por meio das propriedades *schema:value*, *schema:minValue* e *schema:maxValue* é possível relacionar valores específicos às diferentes propriedades de sistema. É importante ressaltar que propriedades de sistema não possuem apenas valores numéricos. Elas possuem, ainda, unidades de medida destes valores. Assim, a propriedade *schema:unitCode* é adotada de modo que unidades de medidas sejam armazenadas na variável *?Unit*. No catálogo semântico proposto, faz-se uso das classes da ontologia QUTD<sup>21</sup> que representam as diferentes unidades de medidas.

No caso das condições de funcionamento das capacidades de sistema, estas são automaticamente compartilhadas com todas as suas propriedades de sistema. A relação entre a capacidade de sistema de um modelo e suas conduções é dada pela propriedade *ssn-system:inCondition* (Figura 39). A Figura 40 apresenta o modelo para descrever a condição das capacidades de sistema de modelos de sensores.



**Figura 40** - Modelo para descrição das condições de funcionamento

Fonte: Elaboração própria

Na Figura 40, condições são definidas como recurso anônimo (*:condition\_SystemCapability*) do tipo *ssn-system:Condition* e *schema:PropertyValue* - o que

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

implica que valores numéricos podem ser associados à descrição das condições de funcionamento das capacidades de sistema de modelos de sensores. Este mesmo modelo apresentado na Figura 40 é utilizado para descrever condições de outros elementos, tais como: faixa de sobrevivência, faixa de operação, propriedades de sistema, propriedades de operação e propriedades de sobrevivência. Por exemplo, uma dada propriedade de sistema (*?SystemProperty*) pode apresentar condições de funcionamento específicas. Nesse caso, tais condições devem ser associadas diretamente à propriedade em questão por meio da propriedade *ssn-system:inCondition*. Além disso, condições não se limitam em quantidade, ou seja, cada propriedade pode possuir várias condições. Por este motivo, tanto as instâncias de *ssn-system:SystemCapability* quanto as instâncias de *ssn-system:SystemProperty* podem ser associadas a um ou mais nós em branco do tipo *ssn-system:Condition*.

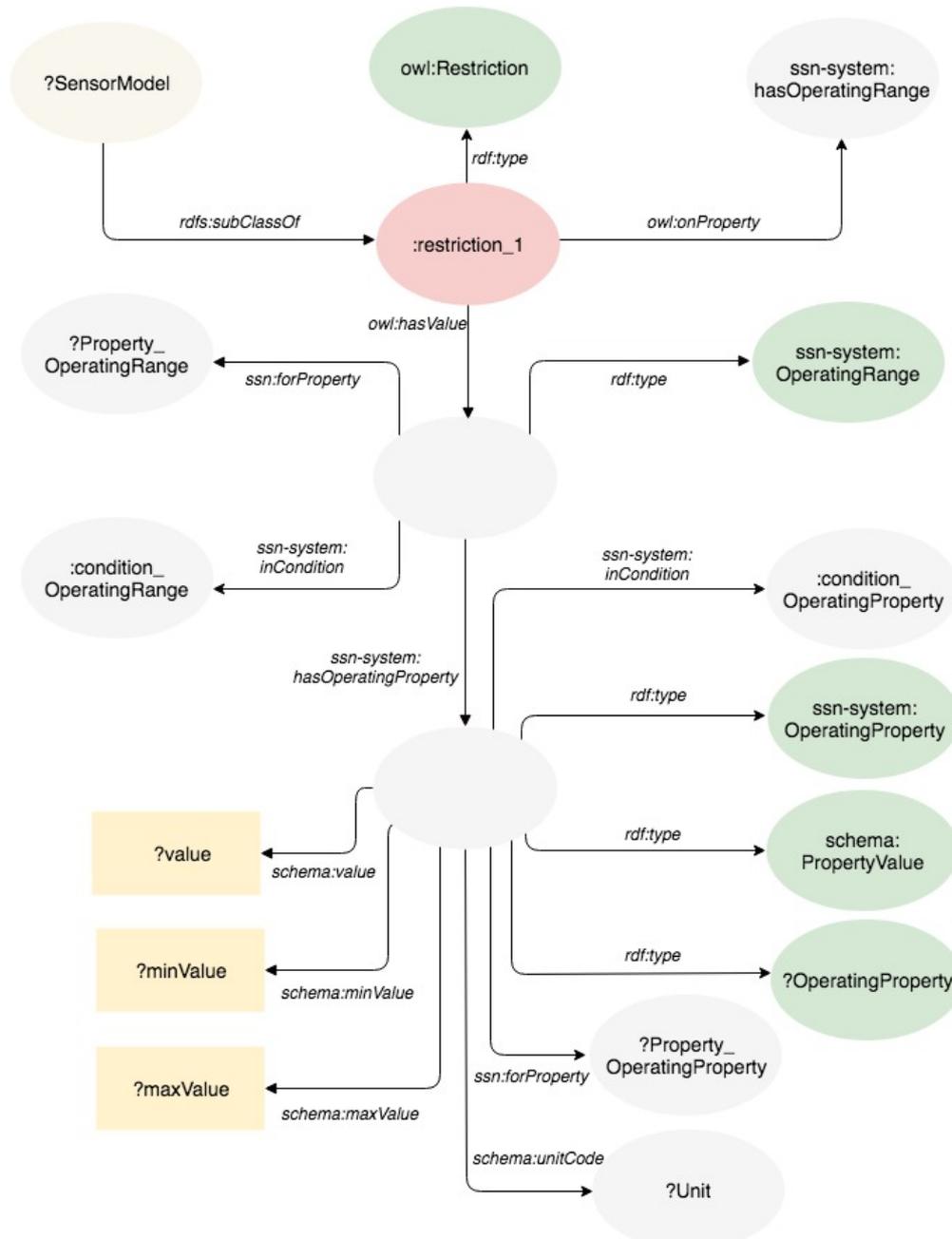
A ontologia SSN permite que a capacidade de sistema de um sensor seja atrelada a sua propriedade de observação (*?Property\_SystemCapability*) por meio da propriedade *ssn:forProperty*. Ao vincular a capacidade de sistema de um sensor a sua propriedade de observação, todas as propriedades de sistema, que compõem a capacidade de sistema de um sensor, são automaticamente relacionadas à propriedade observada pelo sensor. No entanto, é preciso estar atento ao fato de que alguns modelos de sensores observam mais de uma propriedade. Tais tipos de modelos, comumente, apresentam propriedades de sistema de mesma natureza atreladas a propriedades de observação distintas. Por isso, o ideal é que cada propriedade de sistema, de forma particular, seja vinculada a sua propriedade de observação (*?Property\_SystemProperty*), vide Figura 39. Por exemplo: um modelo de sensor pode observar as propriedades temperatura do ar e umidade do ar e, portanto, possuir duas propriedades de sistema de mesma natureza: precisão, cujos valores sejam  $\pm 2$  °C para a propriedade temperatura do ar e  $\pm 5\%$  para umidade do ar.

- Restrição “:restriction\_1”

A ontologia SSN define propriedades de operação como sendo aquelas propriedades que representam o funcionamento normal do sensor sob determinadas condições. Diferente das propriedades de sistema, as propriedades de operação não afetam diretamente o processo de observação de um sensor, embora sejam essenciais para a operação normal do dispositivo.

A faixa de operação de um sensor é definida pela classe *ssn-system:OperatingRange* e engloba as propriedades de operação e suas condições. Desse modo, esta classe relaciona-se

com as classes *ssn-system:OperatingProperty* e *ssn-system:Condition* por meio das propriedades *ssn-system:hasOperatingProperty* e *ssn-system:inCondition*, respectivamente. Tais considerações estão compreendidas no modelo para descrição de modelo de sensores, mais especificamente, na restrição *:restriction\_1*, utilizada para definir o valor da propriedade *ssn-system:hasOperatingRange*, como ilustrado na Figura 41.



**Figura 41** - Restrição na propriedade *ssn-system:hasOperatingRange*

Fonte: Elaboração própria

Na Figura 41, pode-se verificar que o valor definido para propriedade *ssn-system:hasOperatingRange* é uma instância de *ssn-system:OperatingRange*. Tal instância é representada no modelo como um nó em branco e pode possuir uma ou mais propriedades de

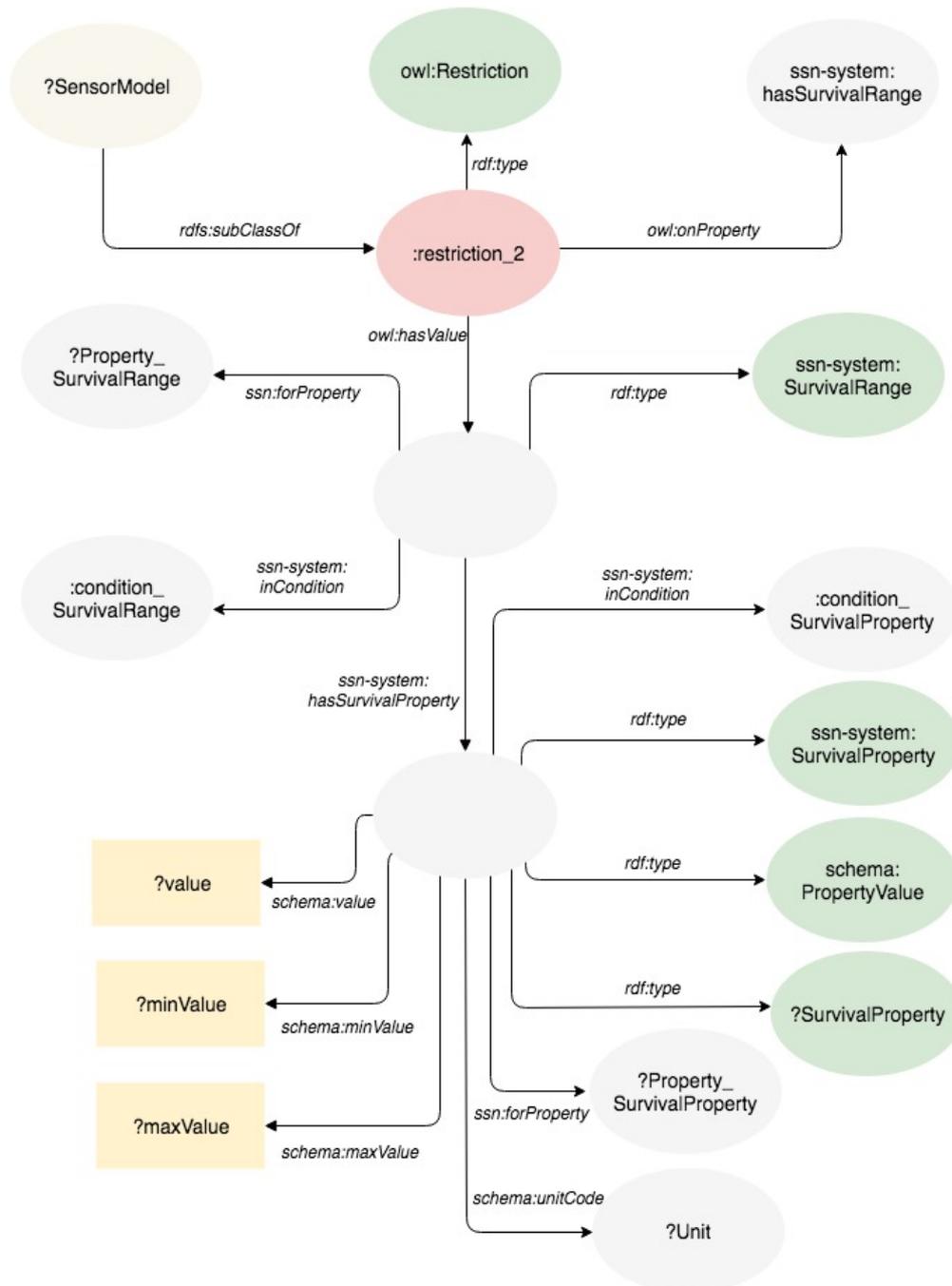
operação, por meio da propriedade *ssn-system:hasOperatingProperty*, cujo valor também é um nó em branco, instância das classes *ssn-system:OperatingProperty*, *schema:PropertyValue* e *?OperatingProperty*. A variável *?OperatingProperty* deve armazenar propriedades de operação - tais propriedades se apresentam como classes (subclasse de *ssn-system:OperatingProperty*) na ontologia SSN ou, ainda, como recursos previamente cadastrados pelo usuário no catálogo, tendo em vista que propriedades de operação consistem em requisito funcional da aplicação.

Uma ou mais condições (*:condition\_OperatingRange*) podem ser associadas à faixa de operação. Além disso, uma dada faixa de operação também pode ser vinculada a uma ou mais propriedades observadas pelo sensor (*?Property\_OperatingRange*) por meio da propriedade *ssn:forProperty*. De forma similar, para cada propriedade de operação (*?OperatingProperty*) condições podem ser associadas (*:condition\_OperatingProperty*), assim como uma dada propriedade observada pelo sensor (*?Property\_OperatingProperty*) por meio da propriedade *ssn:forProperty*. O uso da propriedade *ssn:forProperty* é útil, principalmente, nos casos em que o modelo de sensor, que se deseja catalogar, observa mais de uma propriedade.

- Restrição “:restriction\_2”

Um sistema deve operar de acordo com sua capacidade de sistema. No entanto, se sua faixa de sobrevivência for violada, o sistema será "danificado" e as especificações das capacidades de sistema não poderão ser mantidas. A faixa de sobrevivência, portanto, reúne propriedades de sobrevivência que descrevem as condições nas quais um sistema pode ser exposto sem danos. Por exemplo, o tempo de vida de um sistema (*?SurvivalProperty*) é dado sob uma faixa de temperatura especificada (condição). Nesse caso, se o sistema for exposto à faixa de temperatura diferente daquela estipulada pelo fabricante, o tempo de vida do sistema não pode ser garantido.

A faixa de sobrevivência de um sensor é definida pela classe *ssn-system:SurvivalRange* e relaciona-se com as classes *ssn-system:SurvivalProperty* e *ssn-system:Condition* por meio das propriedades *ssn-system:hasSurvivalProperty* e *ssn-system:inCondition*, respectivamente. Além disso, por meio da propriedade *ssn:forProperty* propriedades de observação podem ser atreladas à faixa de sobrevivência e a cada propriedade de sobrevivência associada. Tais considerações estão compreendidas no modelo para descrição de modelo de sensores, mais especificamente na restrição *:restriction\_2* utilizada para definir o valor da propriedade *ssn-system:hasSurvivalRange*, como ilustrado na Figura 42.



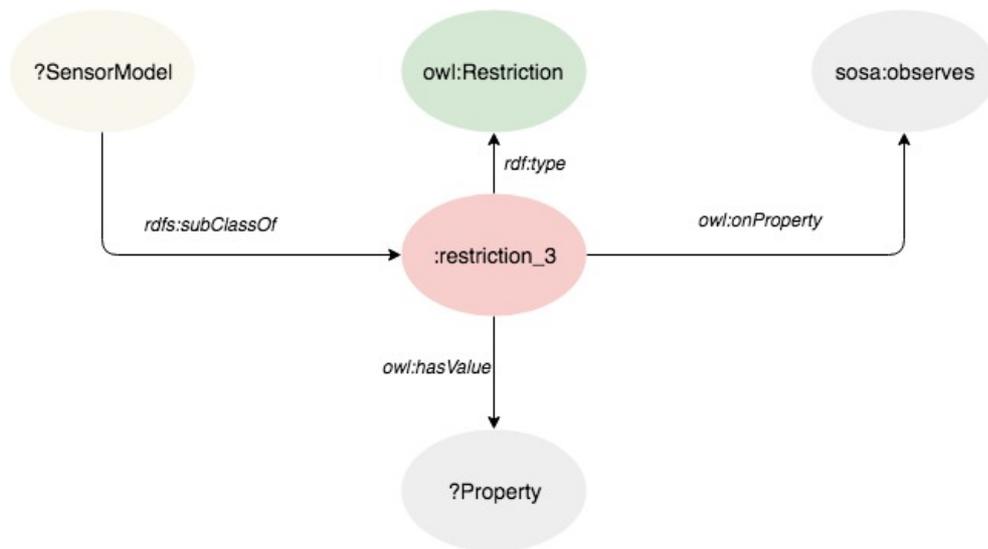
**Figura 42** - Restrição na propriedade *ssn-system:hasSurvivalRange*

Fonte: Elaboração própria

Na Figura 42, pode-se verificar que o valor definido para propriedade *ssn-system:hasSurvivalRange* é uma instância de *ssn-system:SurvivalRange*. As propriedades de sobrevivência se apresentam como classes (subclasse de *ssn-system:SurvivalProperty*) na ontologia SSN (ou recursos previamente cadastrados no catálogo). Além disso, assim como as capacidades de sistemas e a faixa de operação a faixa de sobrevivência pode conter inúmeras condições (*?Condition\_SurvivalRange*) e propriedades de observação (*?Property\_SurvivalRange*) - o mesmo se aplica às propriedades de sobrevivência que compõem a faixa de operação.

- Restrição “:restriction\_3”

Modelos de sensores observam uma ou mais propriedades de um dado objeto de interesse (p. ex.: temperatura do ar, umidade do ar, temperatura da sala 25, etc.). Por meio do uso da propriedade *sosa:observes*, é possível relacionar um dado modelo de sensor (*?SensorModel*) às suas propriedades de observação. Nesse caso, a restrição “:restriction\_3” é utilizada para definir o valor da propriedade *sosa:observes*. A Figura 43 apresenta detalhes da restrição *:restriction\_3*.



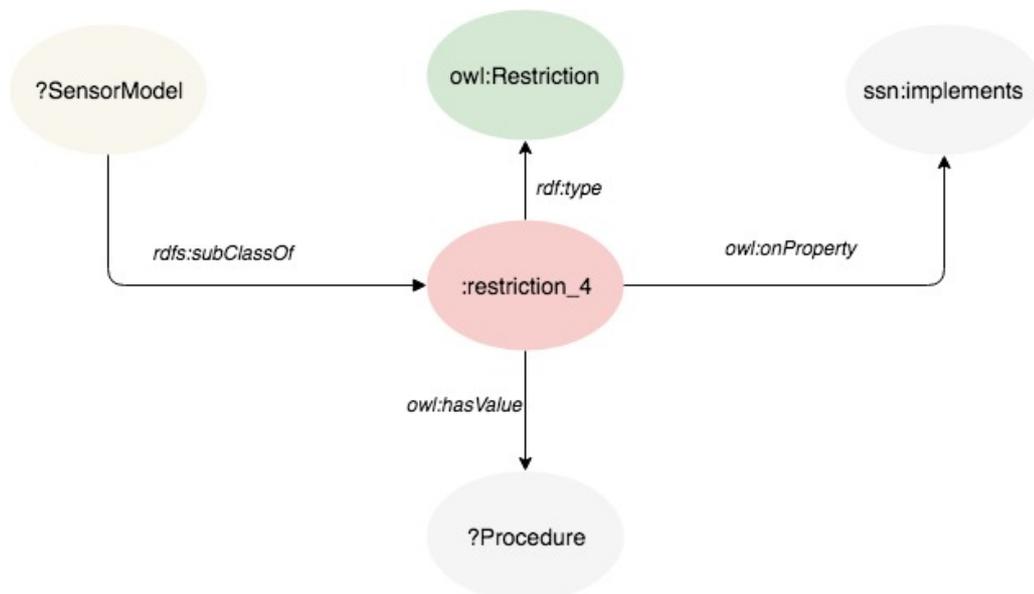
**Figura 43** - Restrição na propriedade *sosa:observes*

Fonte: Elaboração própria

Na Figura 43, o valor na propriedade *sosa:observes* é armazenado na variável *?Property*. Tais valores consistem em instâncias da classe *ssn:Property*. Tendo em vista que a aplicação prevê o cadastro de propriedades de objetos de interesse, a variável *?Property* deve receber propriedades previamente cadastradas na aplicação. O usuário pode cadastrar tanto propriedades de objetos de interesse genéricas quanto específicas. No entanto, na catalogação de modelo de sensores, apenas propriedades genéricas devem ser associadas, uma vez que tais recursos devem ser compartilhados com os diferentes exemplares do modelo. Nesse caso, fica a cargo do usuário fazer a distinção entre propriedades genéricas e específicas durante o processo de catalogação de modelos de sensores, ou seja, o usuário deverá avaliar se o objeto de interesse da propriedade observada é genérico ou específico. Por exemplo, o “ar” é um objeto de interesse genérico, enquanto o “ar na latitude -21.49” é específico.

- Restrição “:restriction\_4”

O processo de observação, de uma ou mais propriedades, implementado por um sensor tem como base um procedimento (*?Procedure*). Procedimentos podem ser entendidos como um protocolo, um plano ou um algoritmo computacional que especifica como a observação é ou deve ser realizada. Para que modelos de sensores (*?SensorModel*) compartilhem seus procedimentos (*?Procedure*) com seus exemplares, faz-se uso de restrição de valor na propriedade *ssn:implements* do modelo. Maiores detalhes sobre a restrição *:restriction\_4* são apresentadas na Figura 44.



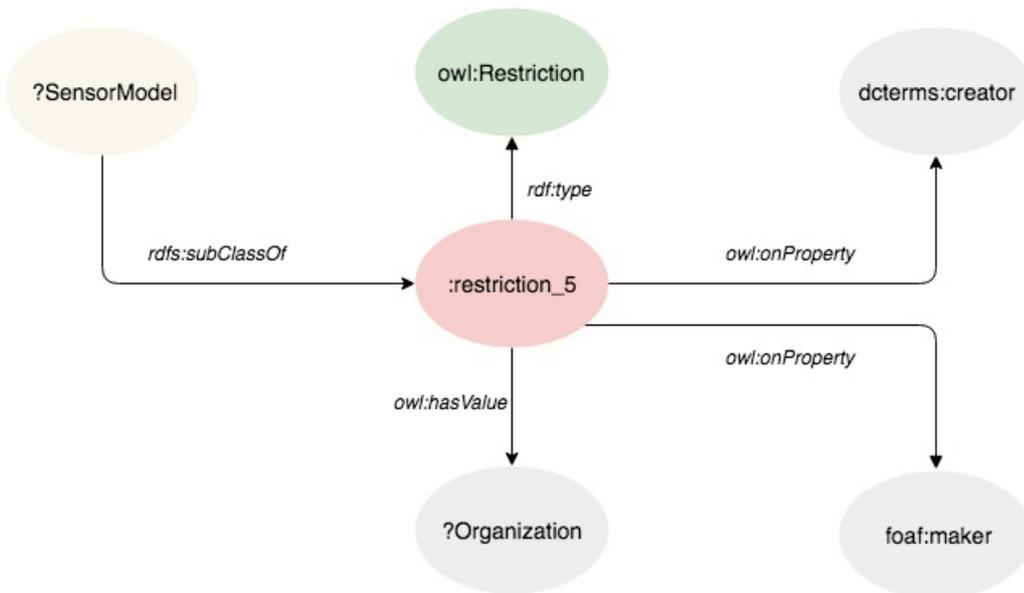
**Figura 44** - Restrição na propriedade *ssn:implements*

Fonte: Elaboração própria

Na Figura 44, o valor da propriedade *ssn:implements* consiste em instâncias da classe *sosa:Procedure*. Tendo em vista que a aplicação prevê o cadastro de procedimentos, a variável *?Procedure* deve receber procedimentos previamente cadastrados na aplicação. Um ou mais procedimentos podem ser associados a modelos de sensores. Uma vez associados, os procedimentos são herdados por todos os exemplares do modelo. Isso é possível ao incluir a restrição de valor, indicada na Figura 38, ao modelo ontológico de modelos de sensores.

- Restrição “:restriction\_5”

Modelos de sensores, normalmente, estão associados à organização que produz sensores (exemplares) com base nas características definidas nos modelos. Para que o fabricante de um dado modelo de seja compartilhado com todas os exemplares do modelo, a restrição *:restriction\_5* é criada na classe *?SensorModel* para que o valor das propriedades *foaf:maker* e *dcterms:creator* seja definido. A Figura 45 apresenta mais detalhes da restrição *:restriction\_5*.



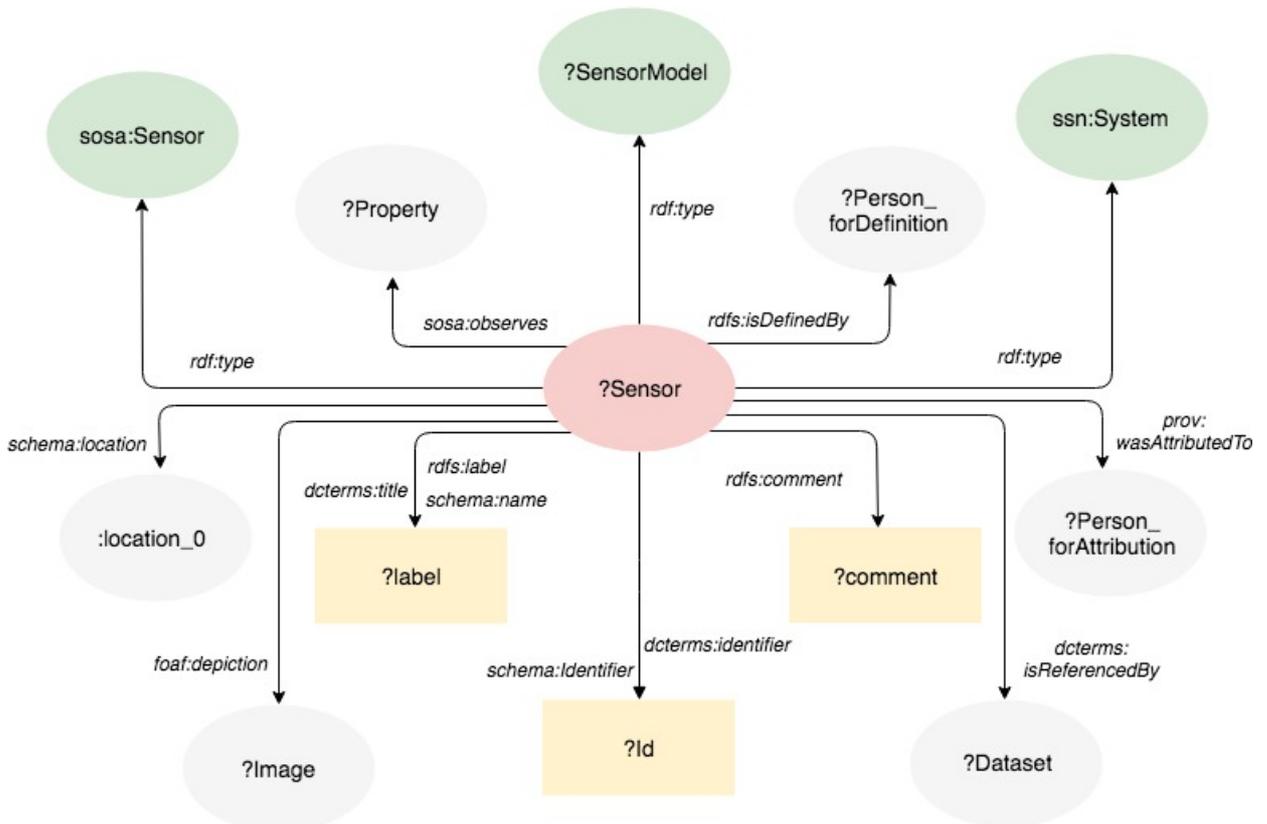
**Figura 45** - Restrição nas propriedades *dcterms:creator* e *foaf:maker*

Fonte: Elaboração própria

Organizações podem ser previamente cadastradas na aplicação e associadas ao modelo de sensor na variável *?Organization*. Na Figura 45, duas propriedades são utilizadas para descrever a relação de modelos de sensores e seus fabricantes, a saber: *dcterms:creator* e *foaf:maker*. Ao catalogar um sensor e associá-lo a um dado modelo que contenha um certo valor nas propriedades *dcterms:creator* e *foaf:maker* (armazenado na variável *?Organization*), o sensor, por ser uma instância do modelo em questão, irá herdar o mesmo valor associado ao modelo nas propriedades *dcterms:creator* e *foaf:maker*. Em outras palavras, a organização associada à fabricação do modelo será a mesma associada à fabricação do exemplar deste modelo (sensor).

### 5.12. Módulo Sensor (Sensor)

Sensores consistem em importante fonte de dados. Embora os dados dos sensores possam ser publicados como meros valores, pesquisar, reutilizar, integrar e interpretar esses dados requer mais do que apenas os resultados da observação. Nesse universo, este trabalho propõe um modelo ontológico para descrição de sensores que facilite a identificação e integração de tais dispositivos, descrevendo-os em termos de sua localização, fabricante, propriedade observada, modelo, etc. A Figura 46 apresenta o modelo ontológico para descrição de sensores.



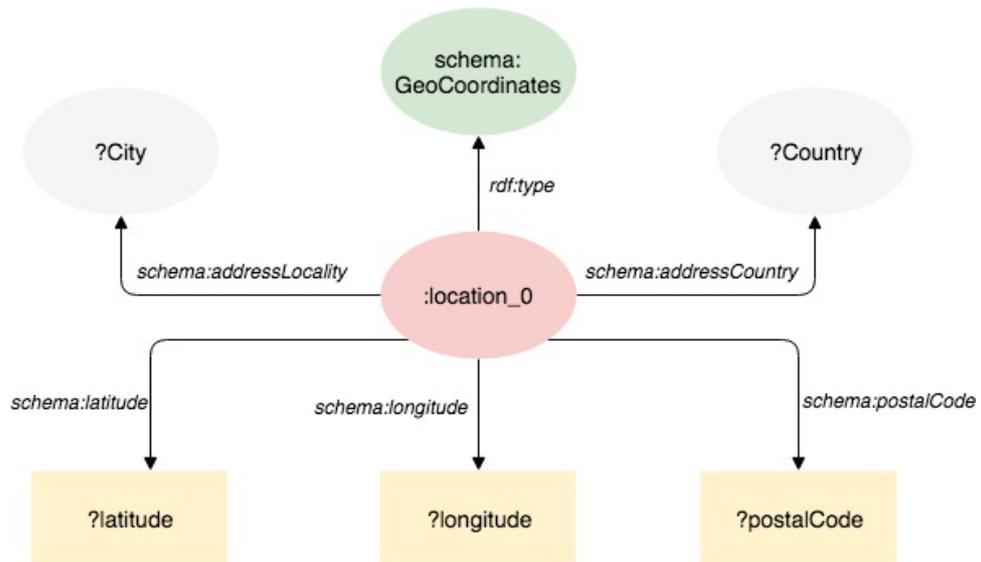
**Figura 46 - Modelo para descrição de sensor**

Fonte: Elaboração própria

No modelo proposto e apresentado na Figura 46, sensores são definidos como recurso do tipo *?SensorModel* (modelo de sensor) e também do tipo *sosa:Sensor* e *ssn:System*. Embora apenas a definição do sensor como recurso pertencente à classe *?SensorModel* seja suficiente para que o sistema infira que este mesmo sensor é uma instância da classe *sosa:Sensor* e *ssn:System*, tendo em vista que a classe *?SensorModel* é previamente definida como sendo subclasse de *sosa:Sensor* e *ssn:System* (Figura 38), para tornar esta informação ainda mais enfática, as classes *sosa:Sensor* e *ssn:System* foram incluídas no modelo de descrição do sensor. Ao catalogar um novo sensor o usuário estará cadastrando, portanto, uma instância das classes *sosa:Sensor*, *ssn:System* e *?SensorModel*.

Na Figura 46, o recurso sensor, é representado pela variável *?Sensor*. O sensor cadastrado tem seu URI gerado a partir das propriedades *schema:identifier* e *dcterms:identifier*. Além disso, a aplicação prevê um segundo identificador para o sensor. No caso deste segundo identificador (nome), as propriedades *rdfs:label*, *dcterms:title* e *schema:name* são utilizadas para relacionar o sensor ao valor recebido na variável *?label*. Ainda, comentário e imagem do sistema podem ser inseridos na catalogação do sensor por meio das propriedades *rdfs:comment* e *foaf:depiction*, respectivamente.

No esquema apresentado na Figura 46, o elemento *:location\_0* consiste em nó em branco (recurso anônimo) que contém as propriedades de localização do sensor. Mais detalhes do modelo para descrição dados de localização de sensores podem ser visto na Figura 47.



**Figura 47** - Dados de localização de sensor

Fonte: Elaboração própria

Na Figura 47, diferentes propriedades da ontologia Schema.org são utilizadas para descrever a localização de sensores. Informações sobre a latitude, longitude e código postal são descritas por meio das propriedades *schema:latitude*, *schema:longitude* e *schema:postalCode*, respectivamente. Além disso, as propriedades *schema:addressLocality* e *schema:addressCountry* são utilizadas para associar sensores à cidade (região) e país onde encontram-se implantados. Diferente das propriedades *schema:latitude*, *schema:longitude* e *schema:postalCode*, os valores dos objetos nas propriedades *schema:addressLocality* e *schema:addressCountry* consistem em recursos, ou seja, elementos dotados de URI. Como cidades e países não configuram requisitos funcionais do catálogo, tais recursos devem ter como origem bases de dados de terceiros (p. ex.: base de dados DBpedia).

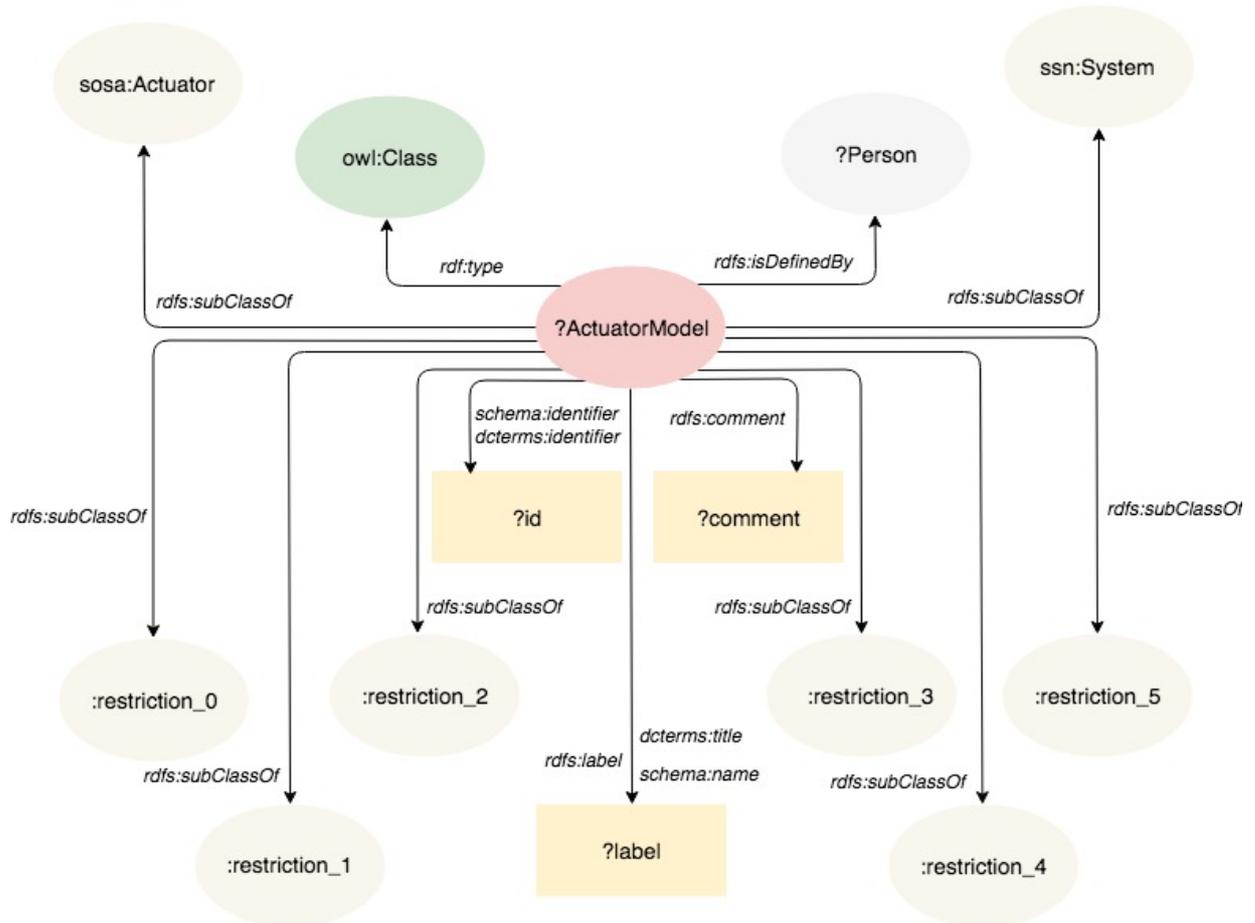
Sensores observam certas propriedades (de objetos de interesse genéricos ou específicos), tais como: temperatura do ar, temperatura da sala 245, nível de um tanque, umidade da sala 245, pressão de um fluido em um duto, etc. Assim, é possível associar tais propriedades (*?Property* - instâncias de *ssn:Property*), previamente cadastradas na plataforma, ao cadastro do sensor. O relacionamento de propriedades (*?Property*) com o sensor é definido pela propriedade *sosa:observes*. Diferente dos modelos de sensores, sensores podem ser associados a propriedades de objeto de interesse genéricas ou específicas.

Sensores podem, ainda, ser associados a pessoas – sejam aquelas responsáveis por tais dispositivos, ou aquelas que fazem o seu cadastro na plataforma. O relacionamento de um sensor (*?Sensor*) com a pessoa responsável pelo seu registro (*?Person\_forDefinition*) é dado pela propriedade *rdfs:isDefinedBy*. Enquanto a propriedade *prov:wasAttributedTo* descreve a relação de um sensor com a pessoa responsável pelo dispositivo (*?Person\_forAttribution*).

O catálogo semântico proposto não prevê a descrição de dados de observação de sensores, apenas de metadados de sensores. Embora seja possível encontrar dados de observação de sensores em diferentes bases de dados na Web, em muitos casos, tais bases de dados não apresentam dados de sensores em formato RDF. Nesse cenário, o modelo proposto prevê o cadastro de bases de dados de sensores (*?Dataset*), sejam elas semânticas ou não (definidas como instâncias da classe *dcat:Dataset*), e sua associação a sensores, cujos dados são utilizados - o relacionamento entre tais recursos é definido pela propriedade *dcterms:isReferencedBy*.

### **5.13. Módulo Modelo de Atuador (Actuator Model)**

De acordo com ontologia SSN, propriedades de sistema, operação e sobrevivência podem ser vinculadas a sensores e atuadores. Isso ocorre porque tais dispositivos são definidos como sendo subclasses de *ssn:System* na própria ontologia SSN. Nesse caso, o conjunto propriedades disponível para descrever ou qualificar sensores e atuadores é praticamente o mesmo. Assim, o modelo proposto para descrição de modelos de atuadores possui muitas semelhanças com aquele apresentado para descrição de modelos de sensores. Outra similaridade dos modelos em questão é a adoção de restrições para que o valor do range das diferentes propriedades, utilizadas para descrever tais modelos, sejam definidos durante a catalogação destes modelos para serem compartilhados com todas suas instâncias. A Figura 48 apresenta maiores detalhes do modelo proposto.



**Figura 48** - Modelo para descrição de modelos de atuadores

Fonte: Elaboração própria

Na Figura 48, a variável *?ActuatorModel* representa as novas classes de modelos de atuadores criadas na aplicação. Tais modelos são definidos como subclasses das classes *sosa:Actuator* e *ssn:System*. Ao definir modelos de atuadores como *ssn:System*, todas as propriedades (p. ex.: propriedades de sistema, operação, sobrevivência) podem ser aplicadas na descrição de tais elementos. Como no caso dos modelos de sensores, para que as informações contidas nos modelos de atuadores sejam compartilhadas com as instâncias da classe em questão (*?ActuatorModel*) restrições para definir o valor do range das diferentes propriedades utilizadas devem ser criadas na própria classe *?ActuatorModel*. Tais restrições são apresentadas na Figura 48 e identificadas como *:restriction\_0*, *:restriction\_1*, *:restriction\_2*, *:restriction\_3*, *:restriction\_4* e *:restriction\_5*. Cada uma destas restrições define o valor de uma propriedade. A Tabela 20 esclarece as propriedades nas quais o valor é definido.

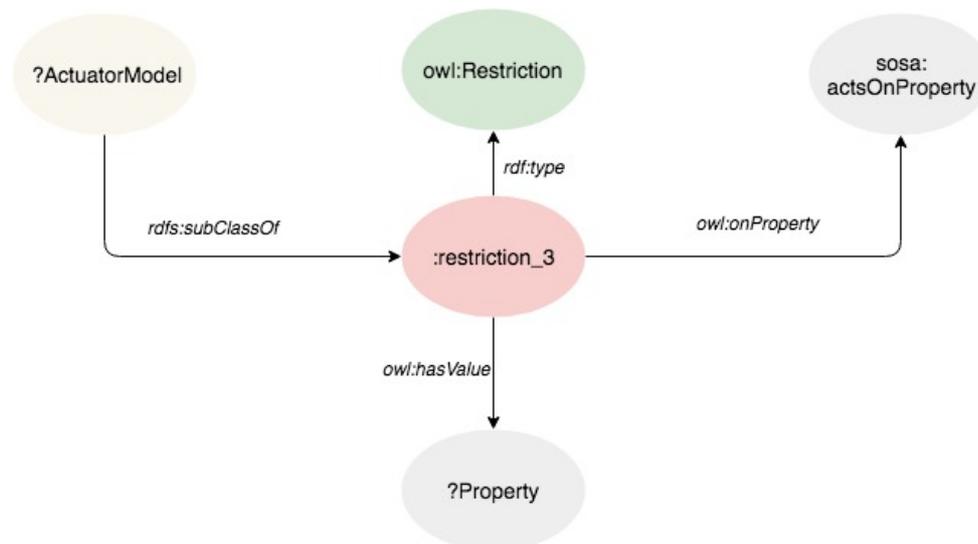
**Tabela 20** - Restrições e suas propriedades (modelo de atuadores)

<b>Restrição</b>	<b>Função</b>
:restriction_0	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn-system:hasSystemCapability</i>
:restriction_1	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn-system:hasOperatingRange</i>
:restriction_2	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn-system:hasSurvivalRange</i>
:restriction_3	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>sosa:actsOnProperty</i>
:restriction_4	Restrição de valor ( <i>owl:hasValue</i> ) na propriedade <i>ssn:implements</i>
:restriction_5	Restrição de valor ( <i>owl:hasValue</i> ) nas propriedades <i>dcterms:creator</i> e <i>foaf:maker</i>

Fonte: Elaboração própria

As restrições indicadas na Tabela 20 são descritas exatamente como apresentado no “Módulo Modelo de Sensores” (Figura 38), com exceção à restrição *:restriction\_3*. Assim, tais restrições foram reaproveitadas na construção do modelo de descrição de modelos de atuadores proposto. Nesse caso, para atender o contexto dos atuadores, os modelos das Figuras 39, 40, 41, 42 e 43 foram utilizados com a seguinte ressalva: a classe *?SensorModel* foi modificada para *?ActuatorModel*.

Tendo em vista que a propriedade *sosa:actsOnProperty* deve ser utilizada exclusivamente na descrição de atuadores, ela não se faz presente na descrição de modelos de sensores, podendo ser identificada, apenas, no modelo para descrição de atuadores. A restrição *:restriction\_3* tem como propósito definir o valor da propriedade *sosa:actsOnProperty*. Maiores detalhes sobre a restrição *:restriction\_3* são apresentados na Figura 49.

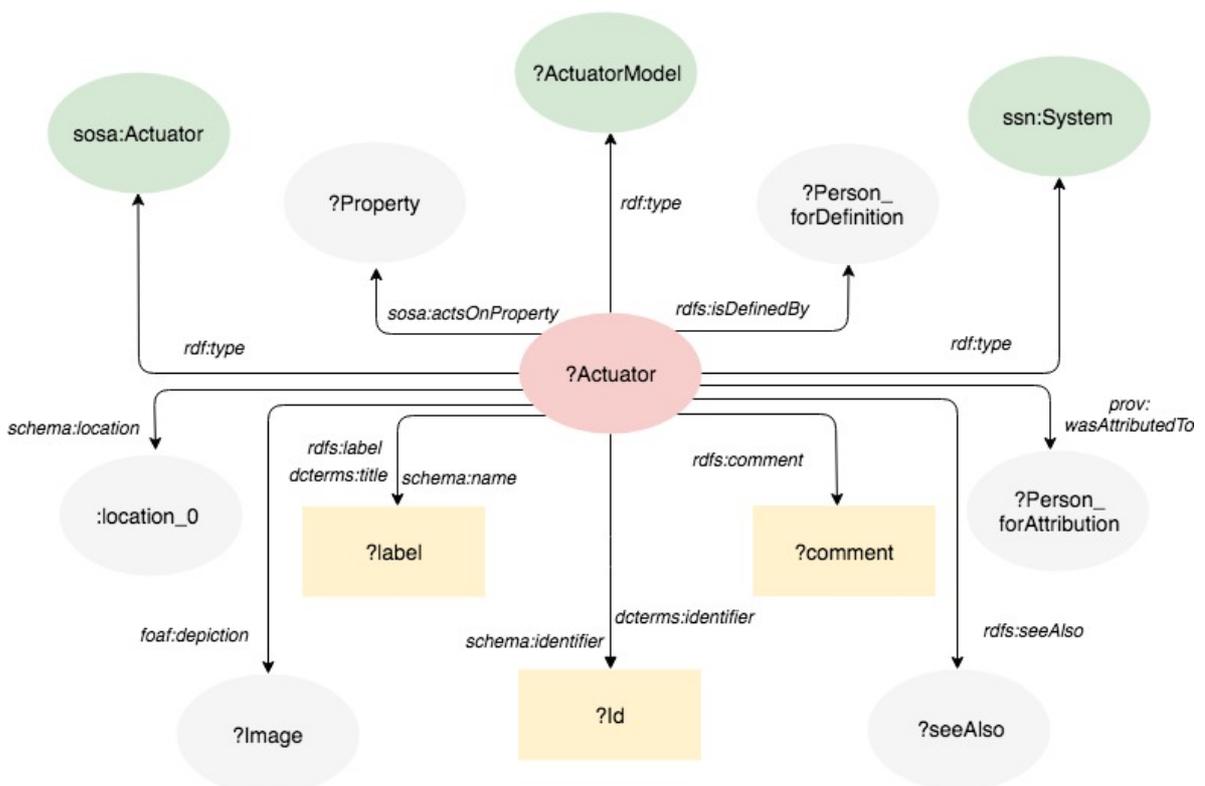
**Figura 49** - Restrição na propriedade *sosa:actsOnProperty*

Fonte: Elaboração própria

O valor da propriedade *sosa:actsOnProperty* consiste em instâncias de *ssn:Property* que são representadas pela variável *?Property*. Ao associar uma dada propriedade ao cadastro de um modelo de atuador na propriedade *sosa:actsOnProperty*, uma tripla que classifica a propriedade selecionada como *sosa:ActuableProperty* será gerada por inferência. Isso ocorre porque na ontologia SSN, na classe *sosa:Actuator*, o contradomínio (range) da propriedade *sosa:actsOnProperty* é definido como *sosa:ActuableProperty*, como descrito na Tabela 16.

#### 5.14. Módulo Atuador (Actuator)

A ontologia SSN define atuadores como sendo dispositivos capazes de mudar o estado de um ambiente. Por exemplo, em um sistema de controle automático de uma janela, a abertura e fechamento da janela consistem em ações (atuações) realizadas por atuadores. Como sensores e atuadores são definidos como tipos de sistema (instâncias de *ssn:System*) no catálogo semântico proposto, tanto sensores quanto atuadores herdam todas as propriedades definidas para descrição de sistemas. Assim, o modelo proposto para descrição de atuadores é muito similar àquele utilizado para descrever sensores. O modelo proposto para descrição de atuadores é apresentado na Figura 50.



**Figura 50** - Modelo para descrição de atuadores

Fonte: Elaboração própria

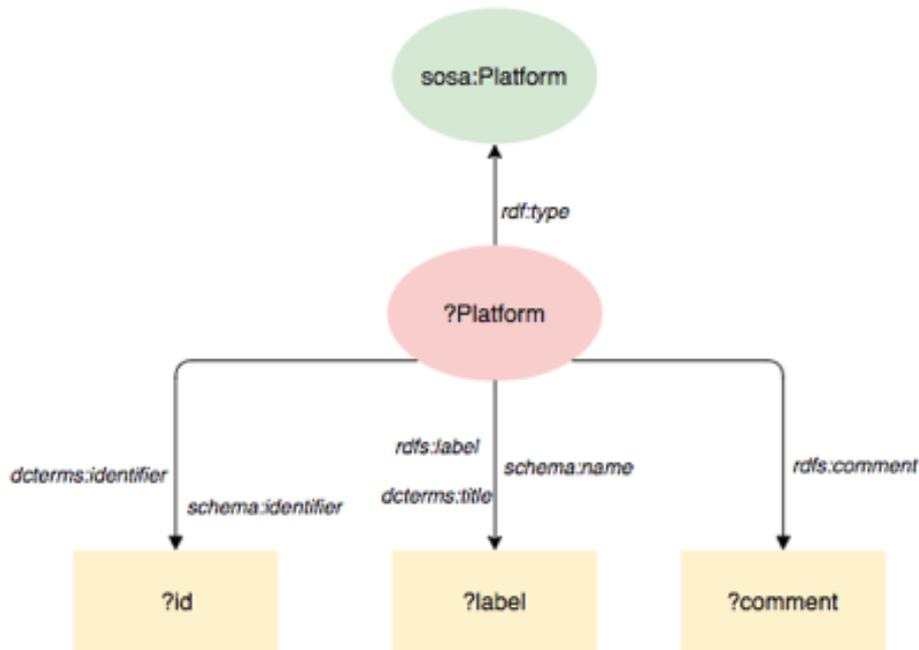
Devido às similaridades dos modelos para descrição de sensores e atuadores, apenas os elementos que não estão inclusos no módulo “Módulo Sensor” serão abordados a seguir.

Atuadores são definidos pelas classes *ssn:System*, *sosa:Actuador* e pela classe *?ActuadorModel* criada previamente pelo usuário na aplicação. 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 por meio da propriedade *sosa:actsOnProperty*. Ao associar um valor à propriedade *sosa:actsOnProperty*, tal valor é definido como *sosa:ActuatableProperty* por inferência, vide Tabela 16. Diferente dos modelos de atuadores e de forma análoga aos sensores, atuadores podem ser associados a propriedades (de atuação) de objeto de interesse genéricas ou específicas.

Sensores podem ser associados a bases de dados que armazenam seus dados de observação, no caso de atuadores tais dispositivos estão comumente associados a Web services ou APIs. No conceito de Internet das Coisas e da Web Semântica das Coisas, solicitações usando o protocolo HTTP para interagir com uma API podem ser realizadas com o objetivo de acessar um atuador. Assim, a propriedade *rdfs:seeAlso* foi utilizada no modelo com propósito de descrever, por exemplo, *links* que permitam acessar os atuadores cadastrados em tempo real. Vale reiterar que a descrição formal semântica da interface de uso (interação) tanto de sensores quanto de atuadores, não faz parte do escopo deste trabalho, sendo previsto como próximo passo a ser realizado.

### **5.15.Módulo Plataforma (Platform)**

Plataforma consiste em entidade que hospeda outras entidades, tais como sensores, atuadores ou outras plataformas. Por exemplo, veículos, navios, aeronaves, satélites, celulares, seres humanos ou animais podem atuar como plataformas para sensores e atuadores. Dada sua relação com o universo de sensores e atuadores, o cadastro de plataformas configura um requisito funcional do catálogo proposto. O modelo para descrição de plataformas é apresentado na Figura 51.



**Figura 51** - Modelo para descrição de plataformas

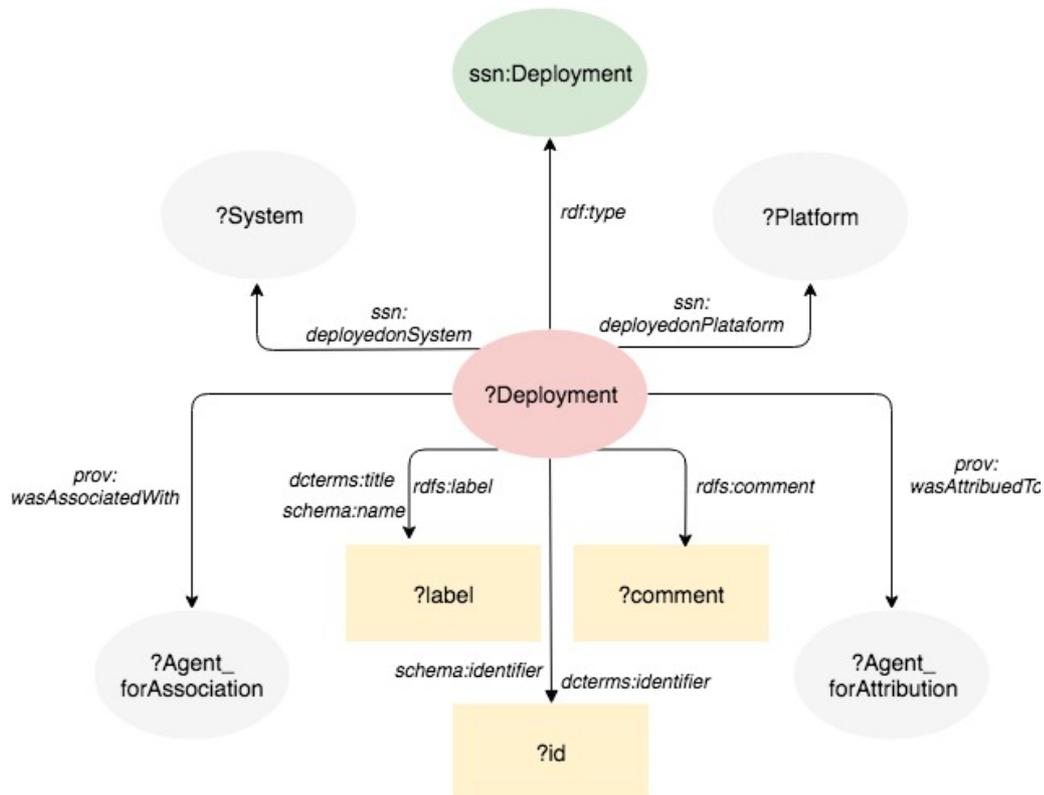
Fonte: Elaboração própria

Na ontologia SSN (e no catálogo proposto), plataformas são definidas pela classe *sosa:Platform*. Assim, ao cadastrar uma dada plataforma (*?Platform*), o usuário cadastra instâncias da classe *sosa:Platform*. Para descrever plataformas, optou-se por um modelo simples, contendo identificador único, nome e comentário. O URI do recurso plataforma é gerado com base no valor nas propriedades *dcterms:identifier* e *schema:identifier*. Já o nome da plataforma é associado na variável *?label* fazendo uso das propriedades *rdfs:label*, *dcterms:title* e *schema:name*. Além disso, comentários sobre o recurso em questão são armazenados na variável *?comment* por meio da propriedade *rdfs:comment*.

A ontologia SSN apresenta propriedades que permitem relacionar plataformas (*?Platform*) a sensores (*?Sensor*) e atuadores (*?Actuator*) fazendo uso da propriedade *sosa:isHostedBy* ou, inversamente, sensores e atuadores podem ser vinculados a plataformas fazendo uso da propriedade *sosa:hosts*. Ainda, plataformas podem ser relacionadas a implantações (*?Deployment*) por meio da propriedade *ssn:inDeployment* e nesse caso a relação inversa também é possível fazendo uso da propriedade da *ssn:deployedOnPlatform*. No catálogo proposto, optou-se por vincular plataformas à implantação por meio da propriedade *ssn:deployedOnPlatform*. Tais considerações estão presentes no modelo ontológico para descrição de implantações (Módulo Implantação).

### 5.16. Módulo Implantação (Deployment)

Diversos elementos (recursos) podem compor a implantação de um sistema (sensor ou atuador), tais como: plataforma (?Platform), propriedades (?Property), organizações (?Organization), pessoas (?Person) ou até mesmo outros sistemas (?System). A Figura 52 apresenta o modelo para descrição de implantação proposto.



**Figura 52** - Modelo para descrição de implantação

Fonte: Elaboração própria

Na ontologia SSN, implantações são definidas pela classe *ssn:Deployment*. Assim, ao cadastrar uma dada implantação (?Deployment), o usuário cadastra instâncias da classe *ssn:Deployment*. O URI do recurso implantação é gerado com base no valor nas propriedades *dcterms:identifier* e *schema:identifier*, vide Figura 52.

É possível relacionar um sistema (?System) a uma implantação (?Deployment) por meio da propriedade *ssn:hasDeployment*, ou, ainda, definir uma relação inversa entre estes recursos, por meio do uso da propriedade inversa *ssn:deployedSystem*. Além disso, a ontologia SSN prevê três tipos de sistemas: Sensor (Sensor), Atuador (Actuator) e Amostrador (Sampler). Assim, qualquer um destes tipos de sistemas pode ser associado a uma implantação. De forma similar aos sistemas, plataformas também podem ser associadas a uma implantação. Nesse caso, pode-se fazer uso da propriedade *ssn:inDeployment*, ou, ainda, da relação inversa fazendo uso da propriedade *ssn:deployedOnPlatform* para relacionar os recursos implantação e plataforma.

Embora haja três tipos de sistemas definidos na ontologia SSN, apenas instâncias da classe *sosa:Sensor* (*?Sensor*) e *sosa:Actuator* (*?Actuator*) podem ser catalogados na aplicação. Desse modo, apenas tais tipos de sistemas podem ser associados na variável *?System* (Figura 52), cuja relação com a implantação é dada pela propriedade *ssn:deployedSystem*. Ao definir a relação no sentido da implantação para o sistema, por meio da propriedade *ssn:deployedSystem*, não é necessário descrever a relação inversa, ou seja, do sistema para a implantação, tendo em vista que esta última relação será inferida. O mesmo ocorre entre os recursos implantação e a plataforma (*?Platform*) – no modelo proposto, a relação entre a implantação e o recurso plataforma é definida por *ssn:deployedOnPlatform*.

Com propósito de contextualizar ainda mais implantações de sensores e/ou atuadores, o modelo proposto prevê a vinculação de um agente (pessoa ou organização) que está associado à implantação por meio da propriedade *prov:wasAssociatedWith* (*?Agent\_forAssociation*). Além disso, pessoas ou organizações responsáveis pela implantação também podem ser associados a uma dada implantação por meio da propriedade *prov:wasAttribuedTo* (*?Agent\_forAttribution*).

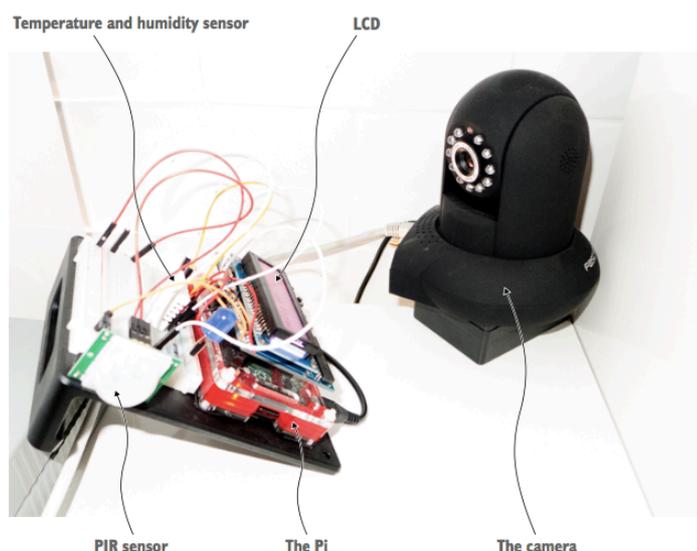
## 6. DESENVOLVIMENTO DO PROTÓTIPO

Para cada um dos requisitos e correspondentes modelos ontológicos serão apresentadas telas com base em cenário de uso realista que descreve uma aplicação WoT. Por razões de simplificação, o grafo RDF gerado como resultado da catalogação dos recursos envolvidos no exemplo realista serão apresentados graficamente por meio de um *graph pattern* (instanciação do *graph pattern*). A versão em sintaxe Turtle dos grafos RDF subjacentes aos cadastros podem ser encontrados no apêndice deste trabalho. Além disso, os cadastros serão apresentados na ordem cronológica em que foram executados para evidenciar a sequência lógica de cadastro.

Os códigos utilizados na construção do protótipo podem ser encontrados em: <http://github.com/app-catalog/sensors-actuators>.

### 6.1. Cenário realista de uso do catálogo

Com o advento da Internet das Coisas, dispositivos físicos passam acessar a infraestrutura da Internet com o propósito de compartilhar informações. Para que tais informações sejam mais facilmente integradas e consumidas, o ecossistema da Web é utilizado na construção de aplicações que permitem a integração de dispositivos heterogêneos de forma interoperável. Nesse universo, Guinard & Trifa (2016) propõem implantação envolvendo sensores e atuadores, cujos dados são disponibilizados na Web. Maiores detalhes sobre a implantação são apresentados na Figura 53.



**Figura 53** - Implantação Web das Coisas

Fonte: Guinard & Trifa (2016)

A Figura 53 apresenta diversos elementos que consistem em requisitos funcionais do catálogo e, por isso, podem ser catalogados na aplicação proposta, configurando um possível

cenário de uso do catálogo. A Tabela 21 apresenta os diferentes dispositivos físicos encontrados na implantação WoT da Figura 53.

**Tabela 21** - Implantação Web das Coisas (WoT)

<i>Item</i>	<i>Descrição</i>
DHT22	Sensor que observa temperatura e umidade
PIR	Sensor que observa variação de sinal infravermelho
Câmera	Sensor que observa imagem
Raspberry Pi	Plataforma que hospeda os sensores e atuadores
Display LCD	Atuador que exibe informação no display
LED	Atuador que transforma eletricidade em energia luminosa

Fonte: Elaboração própria

Para demonstrar o uso da aplicação e as telas desenvolvidas para o cadastro de cada um dos requisitos funcionais, optou-se por apresentar os cadastros do sensor DHT22, da plataforma Raspberry Pi e do atuador LCD, dentre os dispositivos físicos apresentados na Tabela 21 (Figura 53). Além disso, também serão apresentados os cadastros dos recursos associados a tais dispositivos, tais como, pessoa, organização, base de dados, propriedade de objeto de interesse, procedimento, etc. Para facilitar a demonstração dos grafos RDF resultantes dos exemplos realistas, prefixos são adotados para simplificar os URIs dos recursos cadastrados na aplicação e de terceiros (p. ex.: recursos da base de dados DBpedia), vide Tabela 22.

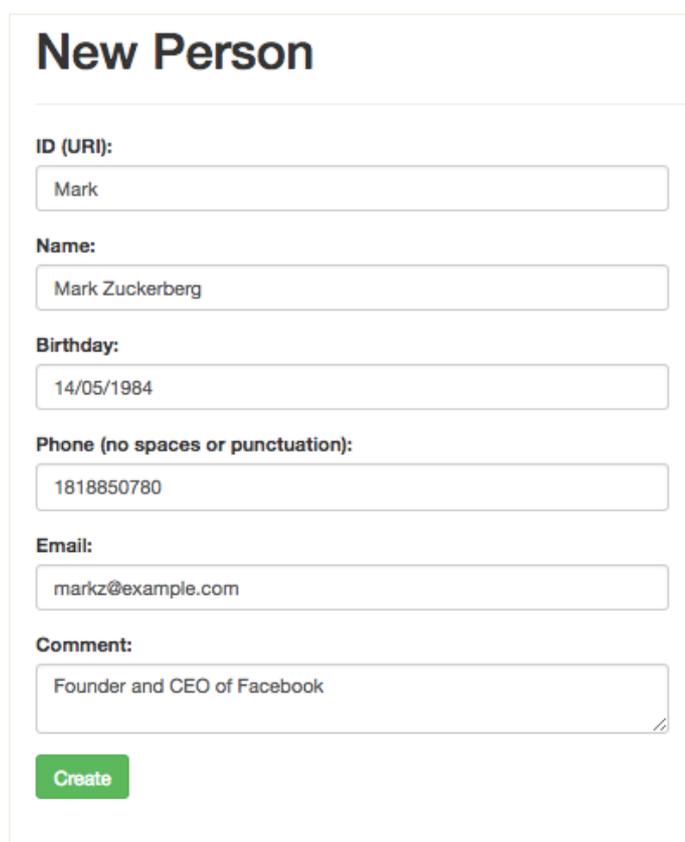
**Tabela 22** - Definição dos prefixos

<b>Prefixo</b>	<b>URI</b>
act	<a href="http://localhost:8080/Resource/Actuator/">http://localhost:8080/Resource/Actuator/</a>
act-model	<a href="http://localhost:8080/Resource/ActuatorModel/">http://localhost:8080/Resource/ActuatorModel/</a>
agent	<a href="http://localhost:8080/Resource/Person/">http://localhost:8080/Resource/Person/</a>
board	<a href="http://localhost:8080/Resource/Platform/">http://localhost:8080/Resource/Platform/</a>
company	<a href="http://localhost:8080/Resource/Organization/">http://localhost:8080/Resource/Organization/</a>
dataset	<a href="http://localhost:8080/Resource/Dataset/">http://localhost:8080/Resource/Dataset/</a>
db	<a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a>
loc	<a href="http://id.loc.gov/vocabulary/iso639-1/">http://id.loc.gov/vocabulary/iso639-1/</a>
method	<a href="http://localhost:8080/Resource/Procedure/">http://localhost:8080/Resource/Procedure/</a>
model	<a href="http://localhost:8080/Resource/SensorModel/">http://localhost:8080/Resource/SensorModel/</a>
object	<a href="http://localhost:8080/Resource/FeatureOfInterest/">http://localhost:8080/Resource/FeatureOfInterest/</a>
pattern	<a href="http://localhost:8080/Resource/Sensor/">http://localhost:8080/Resource/Sensor/</a>
placement	<a href="http://localhost:8080/Resource/Deployment/">http://localhost:8080/Resource/Deployment/</a>
quality	<a href="http://localhost:8080/Resource/Property/">http://localhost:8080/Resource/Property/</a>
quality-type	<a href="http://localhost:8080/Resource/PropertyType/">http://localhost:8080/Resource/PropertyType/</a>
system-prop	<a href="http://localhost:8080/Resource/SystemProperty/">http://localhost:8080/Resource/SystemProperty/</a>
unit	<a href="http://qudt.org/vocab/unit/#">http://qudt.org/vocab/unit/#</a>
wordnet	<a href="http://wordnet-rdf.princeton.edu/ontology#">http://wordnet-rdf.princeton.edu/ontology#</a>
wot	<a href="http://devices.webofthings.io/pi/actuators/">http://devices.webofthings.io/pi/actuators/</a>

Fonte: Elaboração própria

### 6.1.1. Cadastro de Pessoa (Person)

Com base no modelo ontológico para descrição de pessoas apresentado no capítulo 5, telas de criação e edição de uma nova pessoa foram desenvolvidas. Cada uma destas telas, portanto, apresenta diferentes propriedades das ontologias adotadas. Sem que o usuário perceba, triplas RDF são geradas com base nos valores inseridos nestas propriedades durante o processo de catalogação. A Figura 54 ilustra a inserção de uma nova pessoa no catálogo.



The image shows a web form titled "New Person". It contains several input fields with the following labels and values:

- ID (URI):** Mark
- Name:** Mark Zuckerberg
- Birthday:** 14/05/1984
- Phone (no spaces or punctuation):** 1818850780
- Email:** markz@example.com
- Comment:** Founder and CEO of Facebook

At the bottom of the form is a green button labeled "Create".

**Figura 54** - Exemplo de criação de cadastro de pessoa  
Fonte: Elaboração própria

Na Figura 54, o valor “Mark” inserido no campo “ID (URI)” é utilizado para construir o URI do recurso. Na tela de criação do novo recurso, além de identificador único, o usuário pode definir, ainda, o nome (Name), data de nascimento (Birthday), telefone de contato (Phone), email (Email) e inserir comentário (Comment) sobre a pessoa cadastrada. Tais informações podem ser modificadas na tela de edição do recurso, exceto o valor inserido no campo “ID (URI)”. A tela de edição do recurso catalogado é apresentada na Figura 55.

**Figura 55** - Exemplo de edição de cadastro de pessoa

Fonte: Elaboração própria

A tela de edição apresentada na Figura 55 evidencia o campo “Same as at DBpedia.org”, no qual a pessoa catalogada pode ser relacionada a recurso da base de dados DBpedia (por meio da propriedade *owl:sameAs*). Nesse caso, o recurso catalogado e o recurso associado representam a mesma pessoa, em bases de dados distintas, seguindo o quarto princípio *Linked Data*. Para associar “pessoas” da DBpedia, faz-se uso da consulta SPARQL da Figura 56.

```
#
# @Cache-Control:
# @infer true
# @view
#
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX schema: <http://schema.org/>

SELECT ?resource ?label
{
  ?resource rdfs:label ?label .
  ?resource a foaf:Person .

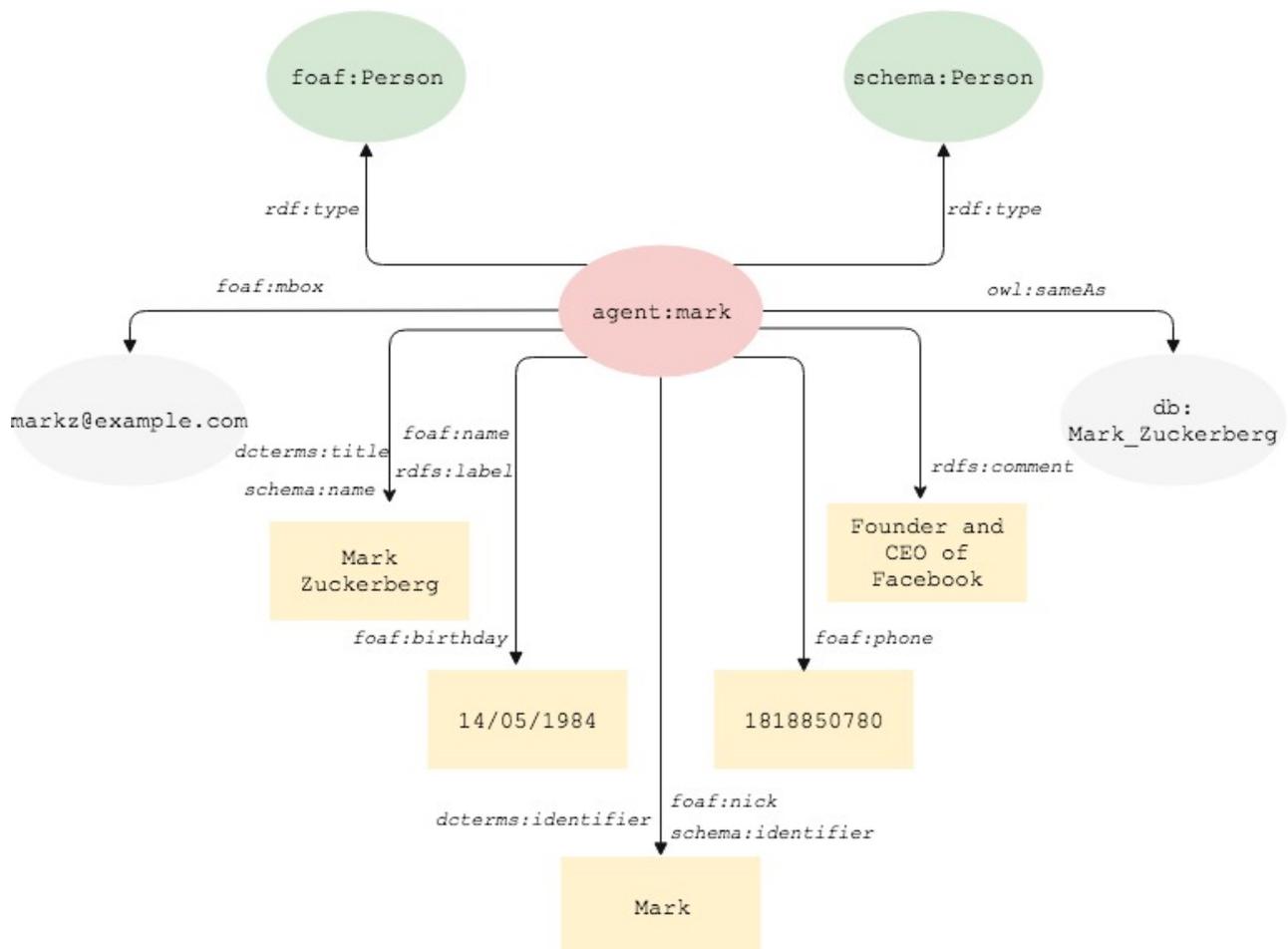
  FILTER (regex(str(?resource), "dbpedia.org", "i") &&
    (lang(?label) = '' || lang(?label) = 'pt' || lang(?label) = 'en') &&
    regex(str(?label), "$keyword", "i")) .
}
ORDER BY ?label
```

**Figura 56** - Consulta SPARQL de pessoas

Fonte: Elaboração própria

Utilizando como base o exemplo em questão (Figura 55), a consulta da Figura 56, basicamente, retorna os recursos da DBpedia, que consistem em instâncias da classe *foaf:Person* e que possuam na sua propriedade *rdfs:label* o valor "Mark Zuckerberg". Os recursos retornados são então exibidos para o usuário selecionar no campo "Same as at DBpedia.org" do formulário de edição apresentado na Figura 55.

A catalogação do recurso, cujo nome é "Mark Zuckerberg" apresentada nas Figuras 54 e 55 gera um grafo RDF subjacente que fica armazenado na base de dados da aplicação. Este grafo é apresentado graficamente na Figura 57 por meio de um *graph pattern* instanciado.



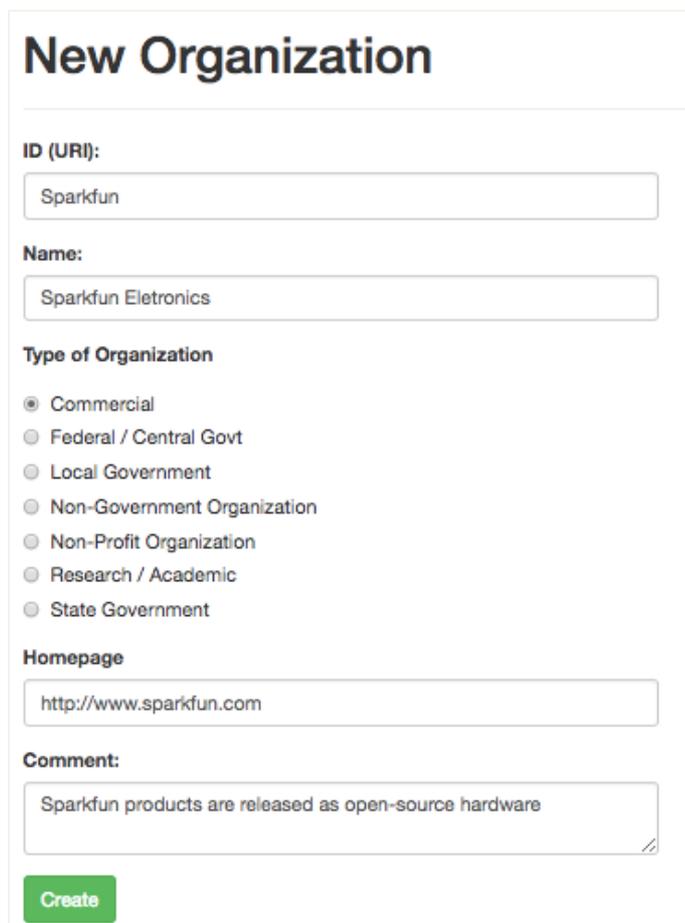
**Figura 57** - Grafo RDF subjacente ao cadastro de uma nova pessoa

Fonte: Elaboração própria

A Figura 57 explicita os valores de cada uma das propriedades, utilizadas para descrever pessoas, associados durante o processo de catalogação. Por meio do uso da tela de edição é possível modificar ou alterar os valores apresentados no grafo em questão, exceto aquele associado às propriedades *foaf:nick*, *schema:identifier* e *dcterms:identifier*.

### 6.1.2. Cadastro de Organização (Organization)

Com base no modelo ontológico para descrição de organizações apresentado no capítulo 5, telas de criação e edição de organizações foram desenvolvidas. A Figura 58 ilustra a inserção de uma nova organização.



**New Organization**

**ID (URI):**  
Sparkfun

**Name:**  
Sparkfun Eletronics

**Type of Organization**

- Commercial
- Federal / Central Govt
- Local Government
- Non-Government Organization
- Non-Profit Organization
- Research / Academic
- State Government

**Homepage**  
http://www.sparkfun.com

**Comment:**  
Sparkfun products are released as open-source hardware

**Create**

**Figura 58** - Tela de criação de organização

Fonte: Elaboração própria

Na Figura 58, o URI do recurso organização é criado com base no valor “Sparkfun” inserido no campo “ID (URI)”. Além disso, diferentes tipos de organização são apresentados na tela de criação por meio de consulta SPARQL. Dentre as opções apresentadas no tesauro de tipos de organização, o usuário pode selecionar uma única opção e alterá-la na tela de edição do recurso em questão, vide Figura 59.

**Sparkfun Eletronics**

**Name:**  
Sparkfun Eletronics

**Address +**  
6333 Dry Creek Parkway  
80503  
Nova Iorque  
Sydney  
Austrália  
Brasil  
Estados Unidos  
Remove

**Type of Organization:**

- Commercial
- Federal / Central Govt
- Local Government
- Non-Government Organization
- Non-Profit Organization
- Research / Academic
- State Government

**Homepage:**  
http://www.sparkfun.com

**Contact Person:**  
Adam Swetlix

**Same as at DBpedia.org:**

**Comment:**  
Sparkfun products are released as open-source hardware

Save Cancel Delete

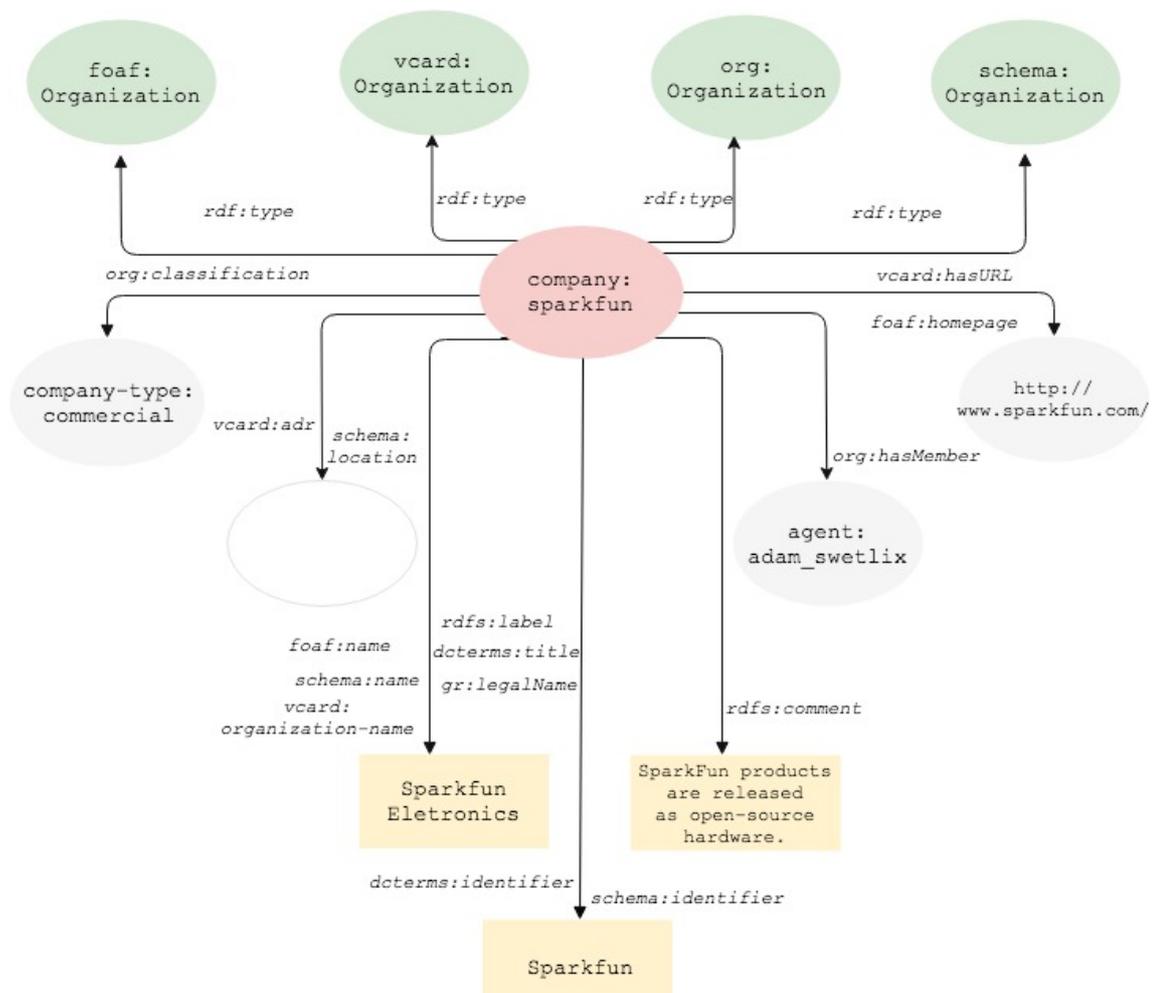
**Figura 59** - Tela de edição de organização

Fonte: Elaboração própria

Na tela de edição apresentada na Figura 59, o usuário pode alterar o nome da organização, o tipo de organização, a *homepage* da organização catalogada e o comentário inserido previamente. Além disso, é possível inserir uma pessoa para contato no campo “Contact Person” - neste campo podem ser associadas, por meio de consulta SPARQL, pessoas previamente cadastradas na aplicação. De forma similar, ou seja, também por meio de consulta SPARQL, recurso da fonte de dados DBpedia equivalente à organização cadastrada pode ser associado ao campo “Same as at DBpedia.org”. Como não foi possível encontrar informações sobre a organização catalogada na base de dados DBpedia, o preenchimento deste campo não foi realizado.

A tela de edição compreende, ainda, os dados de localização da organização. Nesse caso, a aplicação permite que um ou mais endereços sejam associados. No exemplo ilustrado da Figura 59, os recursos que caracterizam cidades e países, associados por meio de consulta SPARQL no campo “Address” (Endereço), têm como origem a base de dados DBpedia.

As informações catalogadas apresentadas nas Figuras 58 e 59 ficam armazenadas na base de dados da aplicação. O RDF subjacente ao cadastro do recurso organização é apresentado por meio de um *graph pattern* instanciado na Figura 60.



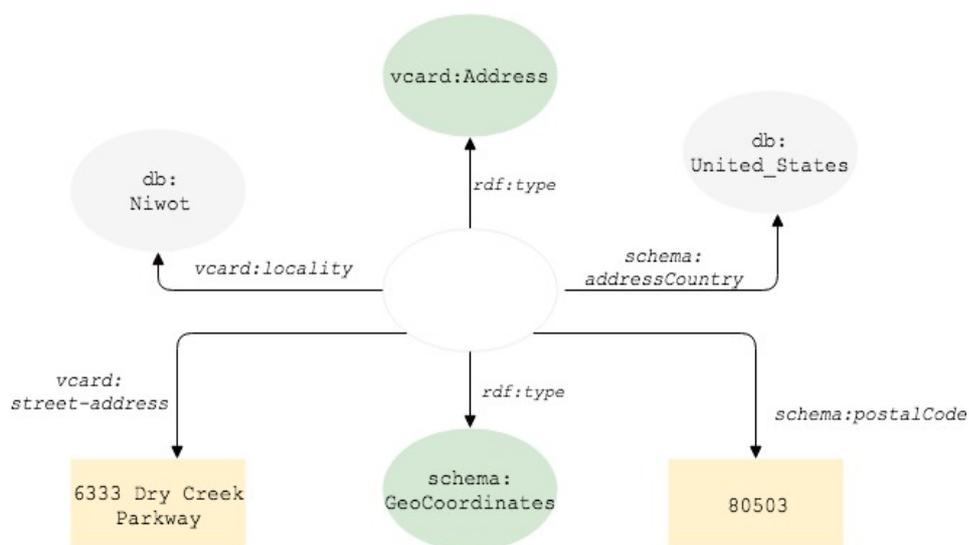
**Figura 60-** Grafo RDF subjacente ao cadastro de organização

Fonte: Elaboração própria

A Figura 60 apresenta os valores associados nas diferentes propriedades utilizadas para descrever organizações. É possível verificar que ao cadastrar uma nova organização, sem perceber, o usuário cria uma nova instância das classes `foaf:Organization`, `vcard:Organization`, `org:Organization` e `schema:Organization`.

O grafo RDF apresentado na Figura 60 é composto por diferentes triplas RDF formadas à medida que valores são inseridos nos campos da tela de criação e edição da organização cadastrada (Figuras 58 e 59). Por exemplo, a tripla RDF formada ao associar um dado recurso no campo “Contact Person” da tela de edição (Figura 59) possui os seguintes elementos: sujeito (`company:sparkfun`), propriedade (`org:hasMember`) e valor (`agent:adam_swetlix`).

Por questão de simplificação, a parte do grafo RDF subjacente ao cadastrado da organização, apresentado na Figura 60, não contempla as propriedades de localização. Maiores detalhes sobre as propriedades de localização da organização cadastrada podem ser vistos na Figura 61.



**Figura 61** - Propriedades de localização da organização catalogada

Fonte: Elaboração própria

Na Figura 61, as propriedades das ontologias Schema.Org e Vcard são evidenciadas. Tais propriedades são vinculadas aos diferentes campos que compõem as informações de endereço, na tela de edição (Figura 59). Por exemplo, um destes campos refere-se ao “País” no qual a organização reside, cujo valor associado consiste em recurso “Estados Unidos” (*db:United\_States*) da DBpedia.

### 6.1.3. Cadastro de Fonte de dados (Dataset)

O catálogo proposto não inclui a descrição dos dados de observação de sensores. Com o propósito de minimizar tal limitação, Bases de dados, que armazenam dados de observação de tais dispositivos, podem ser catalogadas. A Figura 62 ilustra o cadastro de uma Base de dados.

## New Dataset

---

**ID (URI):**

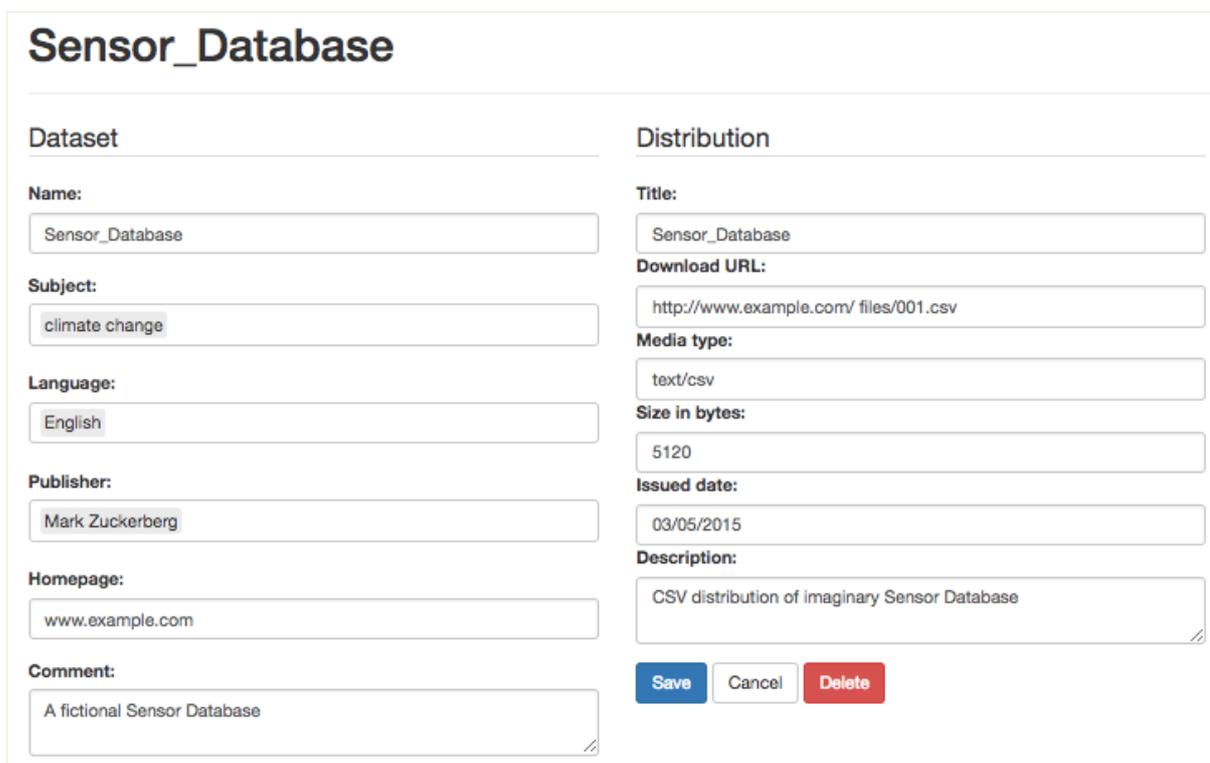
**Name:**

**Comment:**

**Figura 62** - Tela de criação de Base de dados de sensores

Fonte: Elaboração própria

Na Figura 62, o valor inserido no campo “ID (URI)” é utilizado para construir o URI do novo recurso. Nesse caso, uma vez inserido este valor não poderá ser modificado. A tela de edição de Bases de dados não apresenta, portanto, o campo em questão, vide Figura 63.



**Figura 63** - Tela de edição de Bases de dados

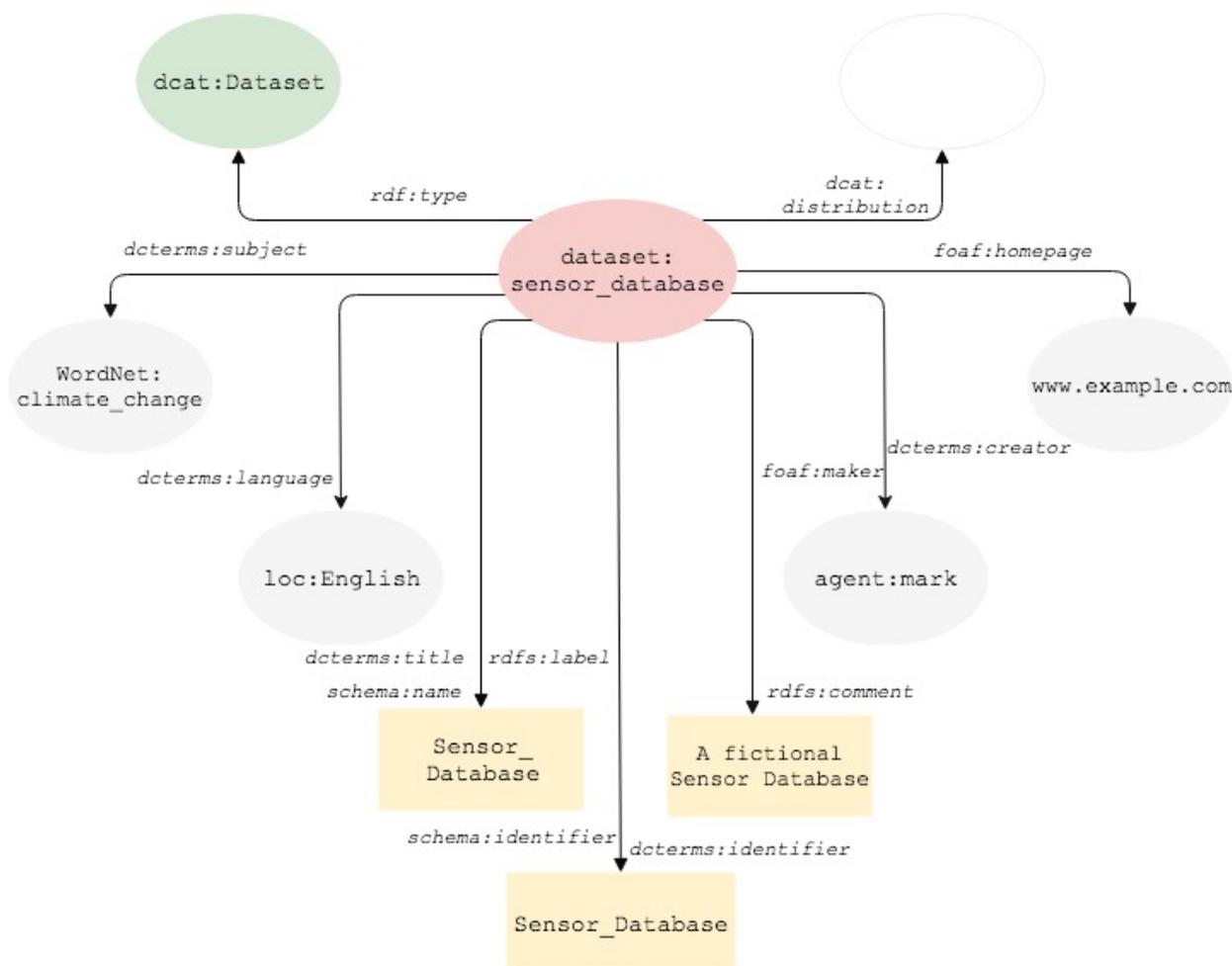
Fonte: Elaboração própria

A Figura 63 evidencia campos do formulário de edição de bases de dados de sensores que são dotados de diferentes propriedades. Os campos indicados na coluna “Dataset” são utilizados para descrever a base de dados em si, por meio de diferentes atributos, tais como “Name” (nome), “Subject” (assunto), “Language” (língua), “Publisher” (editor responsável), “Homepage” (homepage) e “Comment” (comentário). Já os campos indicados na coluna “Distribution” são utilizados para descrever o arquivo (armazenado na Fonte de dados catalogada) que contém os dados de observação de sensores.

No exemplo de cadastro apresentado na Figura 63, nos campos “Subject” e “Language” os valores associados consistem em recursos que têm como origem as bases de dados WordNet e Library Of Congress, respectivamente. Enquanto no campo “Publisher” pessoa, previamente cadastrada na aplicação, é associada (a descrição da pessoa associada encontra-se nas Figuras 54 e 55). Nestes campos, faz-se uso de consultas SPARQL para associação dos recursos.

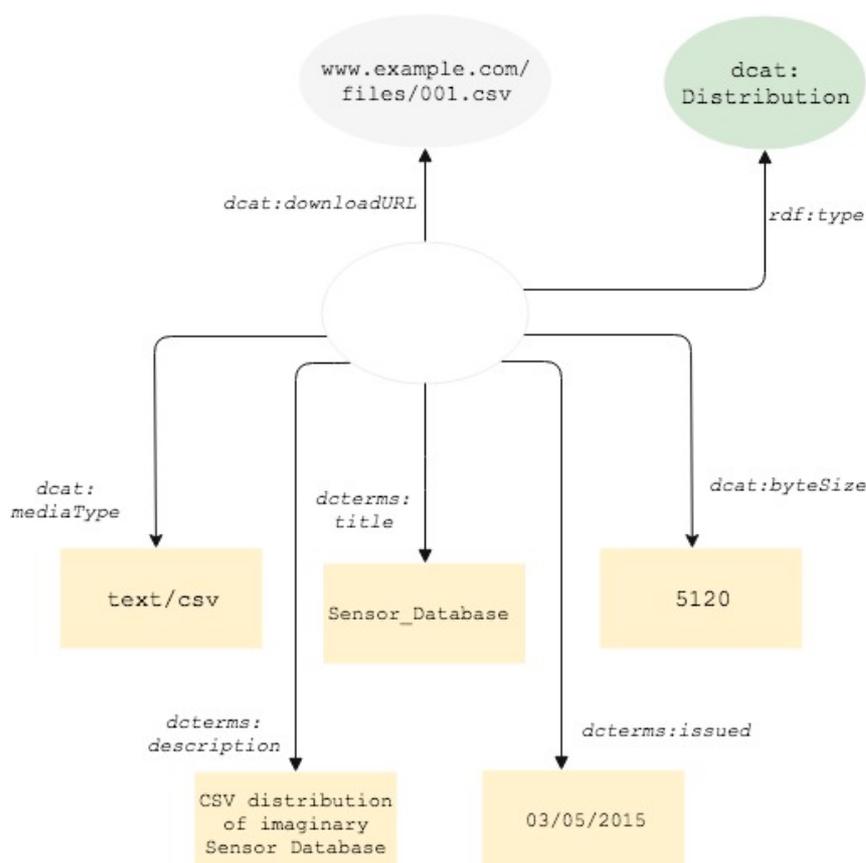
O arquivo que apresenta os dados de observação de sensores é descrito com sendo um arquivo de formato CSV (tipo de mídia ou “Media Type”), ou seja, trata-se de exemplo de

cadastro de fonte de dados que não publica dados em formato RDF. O RDF subjacente ao cadastro do recurso “Fonte de dados” é apresentado por meio de um *graph pattern* instanciado na Figura 64.



**Figura 64** - Grafo RDF subjacente ao cadastro de Bases de dados  
Fonte: Elaboração própria

Na Figura 64, é possível verificar todos os valores associados nos diferentes campos da tela de criação e edição da Base de dados catalogada. Por exemplo, nas propriedades *dcterms:subject* e *dcterms:language*, recursos das bases de dados semânticas “WordNet” e “Library Of Congress” são associados por meio consultas SPARQL. Nesse caso, há um enriquecimento da base de conhecimento, sobretudo, da organização catalogada. Por motivo de simplificação, a parte do grafo RDF subjacente ao cadastro de da Base de dados, que concentra os detalhes do arquivo que contém os dados de observação de sensores é apresentado separadamente, na Figura 65.



**Figura 65** - Grafo RDF subjacente ao cadastro de Bases de dados com foco na descrição arquivo que contém os dados de observação de sensores

Fonte: Elaboração própria

A Figura 65 evidencia as diferentes propriedades para descrever o arquivo que contém os dados de observação de sensores da base de dados cadastrada. Maior parte destas propriedades possuem contradomínio definido como *rdfs:Literal*, ou seja, os valores associados nestas propriedades são literais. Em contraste, na propriedade *dcat:downloadURL* o valor associado é um URI e, portanto, consiste em um recurso.

As informações contidas na Figura 65 são inerentes ao cadastro da base de dados catalogada *dataset:sensor\_database*. Ao apagar tal recurso, as informações sobre o arquivo, que contém os dados de observação de sensores, também são removidos, tendo em vista que se trata de recurso anônimo.

#### 6.1.4. Cadastro de Objeto de Interesse (Feature Of Interest)

Objetos de Interesse são utilizados na descrição de propriedades de observação e/ou atuação e por isso devem ser cadastrados previamente ao cadastro de tais propriedades no catálogo proposto. Com base no modelo ontológico proposto para descrição de objetos de interesse apresentado no Capítulo 5, telas de criação e edição para a catalogação de tais recursos foram desenvolvidas e são apresentadas nas Figuras 66 e 67.

**New Feature of Interest**

**ID (URI):**

**Name:**

**Comment:**

**Create**

**Figura 66** - Tela de criação de um novo Objeto de Interesse  
 Fonte: Elaboração própria

A Figura 66 ilustra o cadastro de um novo Objeto de Interesse: o ar atmosférico (air). Fazendo uso da tela de edição, é possível enriquecer a catalogação deste recurso associando-o à base de dados DBpedia, vide Figura 67.

**Air**

**Name:**

**Same as at DBpedia.org:**

**See Also:**

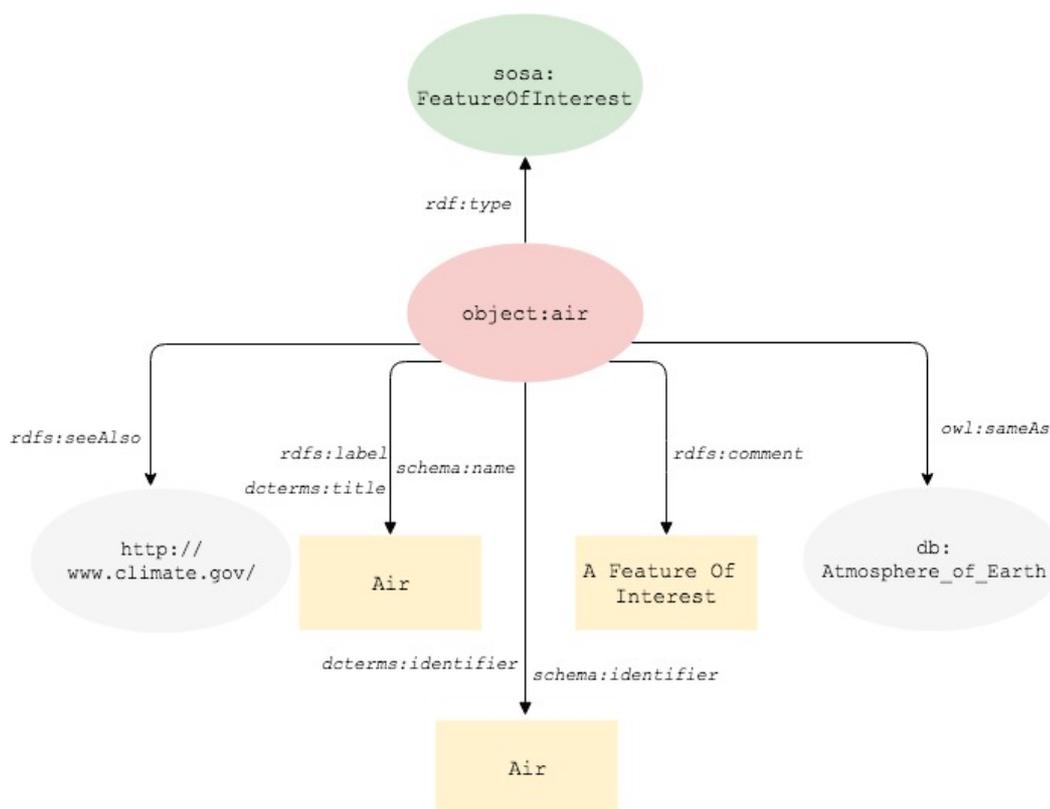
**Comment:**

**Save** **Cancel** **Delete**

**Figura 67** -Tela de edição de um Objeto de Interesse  
 Fonte: Elaboração própria

Na Figura 67, o nome do Objeto de Interesse cadastrado pode ser modificado no campo “Name”. Além disso, no campo “Same as at DBpedia.org” recurso da DBpedia é associado por meio de consulta SPARQL. Mais informações sobre o recurso podem ser encontradas no *link*

apresentado no campo “See Also” (<http://www.climate.gov/>). O Grafo RDF subjacente ao cadastro em questão pode ser verificado na Figura 68.



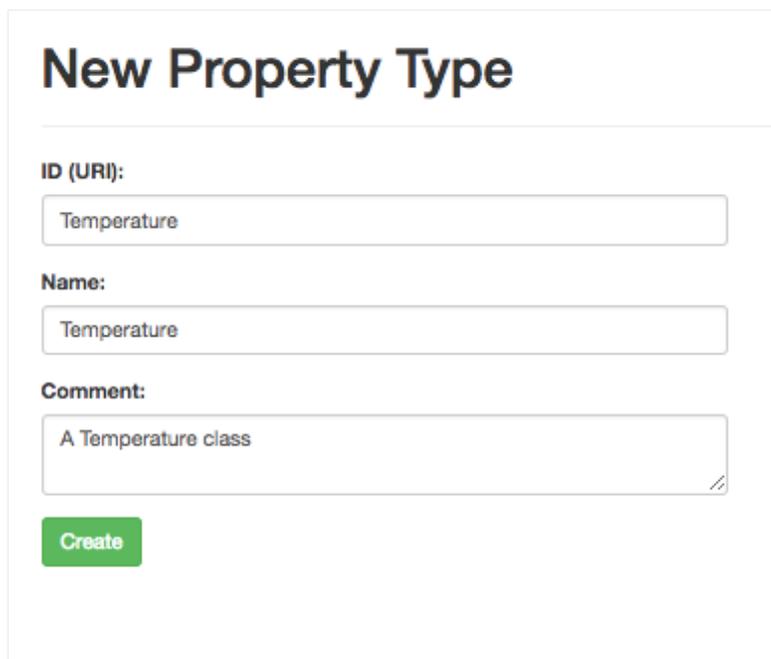
**Figura 68** - Grafo RDF do objeto de interesse “air” (ar)

Fonte: Elaboração própria

Na Figura 68, é possível verificar que todas as informações contidas nas telas de criação e edição de Objetos de Interesse fazem parte do grafo RDF resultante do cadastro deste recurso. Tais informações ficam disponíveis na base de dados da aplicação para serem utilizadas quando o usuário desejar, por exemplo, na descrição de outros recursos.

### 6.1.5. Cadastro de Tipo de propriedade (Property Type)

Antes de criar novas propriedades de observação e/ou atuação (de objetos de interesse), é preciso criar as classes nas quais tais propriedades irão pertencer. No catálogo proposto, tais classes são chamadas de “Tipos de Propriedades”. Assim, ao criar uma nova propriedade, o seu tipo deve, necessariamente, ter sido criado previamente para que possa ser associado à descrição da propriedade. A Figura 69 ilustra a criação de um novo Tipo de propriedade.



**New Property Type**

**ID (URI):**

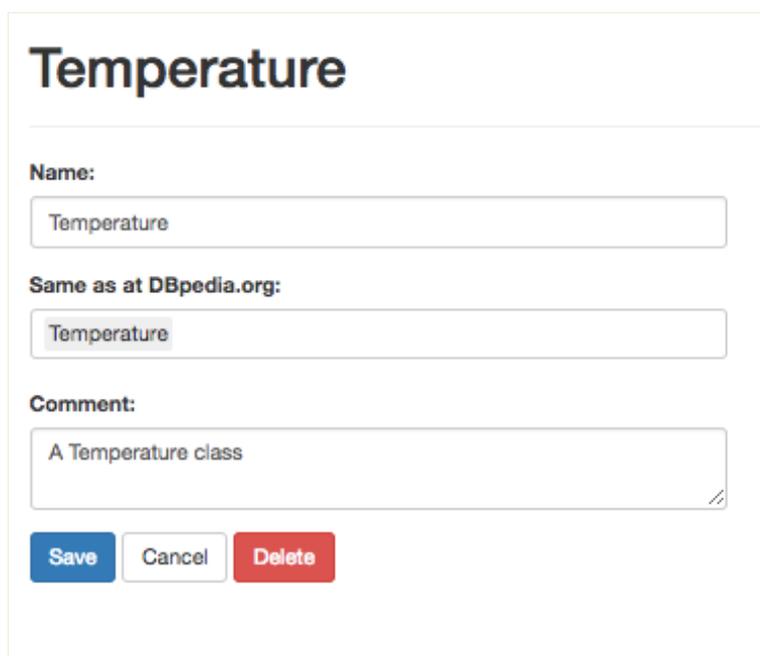
**Name:**

**Comment:**

**Create**

**Figura 69** - Tela de criação de um novo Tipo de propriedade  
Fonte: Elaboração própria

A Figura 69 ilustra a criação do novo Tipo de propriedade “Temperature” (Temperatura). Trata-se de uma nova classe “Temperature” e sua criação permite que instâncias desta classe possam ser criadas pelo usuário. A Figura 70 apresenta a tela na qual as informações do recurso em questão podem ser editadas.



**Temperature**

**Name:**

**Same as at DBpedia.org:**

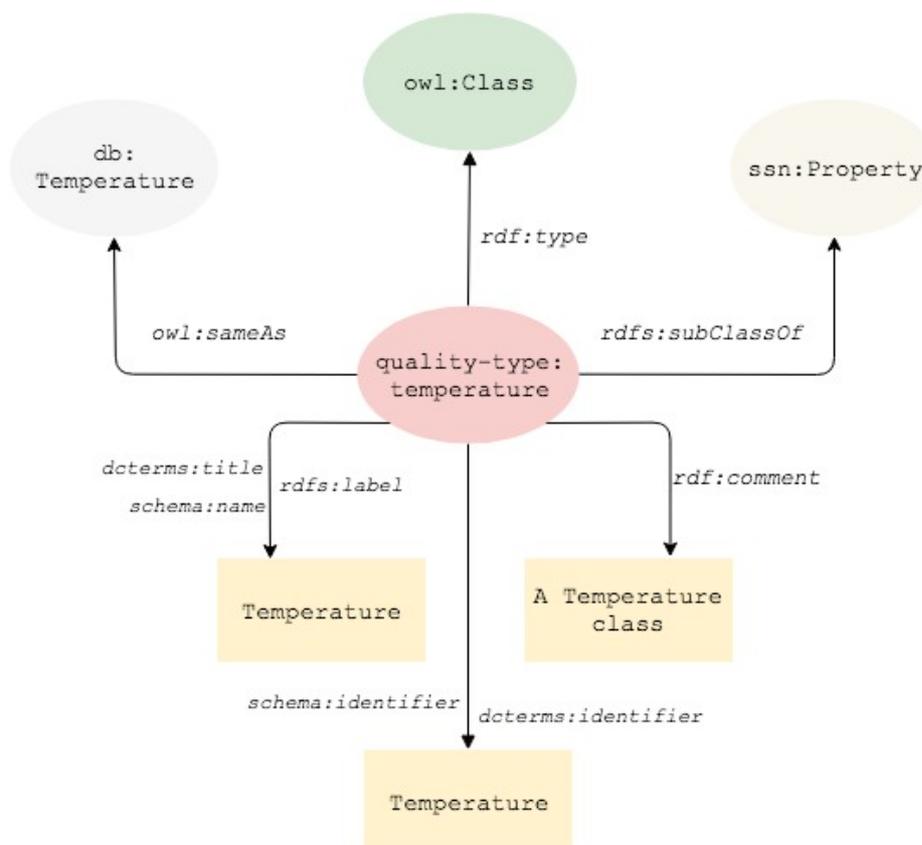
**Comment:**

**Save** **Cancel** **Delete**

**Figura 70** - Tela de edição de um Tipo de propriedade  
Fonte: Elaboração própria

Na Figura 70, assim como Objetos de Interesse, Tipos de propriedade podem ser associados a recurso da base de dados DBpedia no campo “Same as at DBpedia.org”. As

informações apresentadas nas telas de criação e edição, referentes ao cadastro do Tipo de propriedade “Temperature”, são apresentadas em *graph pattern* instanciado, na Figura 71.



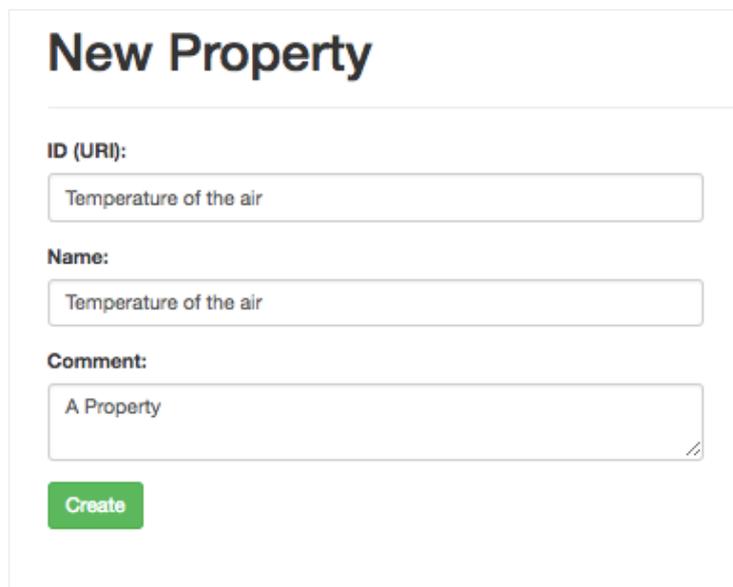
**Figura 71**-RDF subjacente ao cadastro de um novo Tipo de propriedade  
Fonte: Elaboração própria

A Figura 71 evidencia a descrição do Tipo de propriedade “Temperature” (Temperatura). Nesse caso, as propriedades das ontologias selecionadas e associadas em cada campo da tela de criação e edição podem ser visualizadas. Por exemplo, ao associar o recurso “*db:Temperature*” no campo “Same as at DBpedia.org” na tela de edição, uma relação entre o recurso da base de dados DBpedia “*db:Temperature*” e o Tipo de propriedade “Temperature” é estabelecida e definida pela propriedade *owl:sameAs*, ou seja, torna-se explícito que o recurso, que tem como origem à base de dados DBpedia, é equivalente (mesmo que) ao recurso catalogado.

#### 6.1.6. Cadastro de Propriedade de Objeto de Interesse (Property)

A criação de uma nova Propriedade de Objeto de Interesse demanda a criação ou a existência de outros recursos para que sua descrição/catalogação seja realizada com maior completude. Por exemplo, Tipos de propriedades e Objetos de Interesse podem ser associados

à catalogação de Propriedades de Objeto de Interesse. A Figura 72 ilustra a inserção de uma nova Propriedade de Objeto de Interesse.



**New Property**

**ID (URI):**  
Temperature of the air

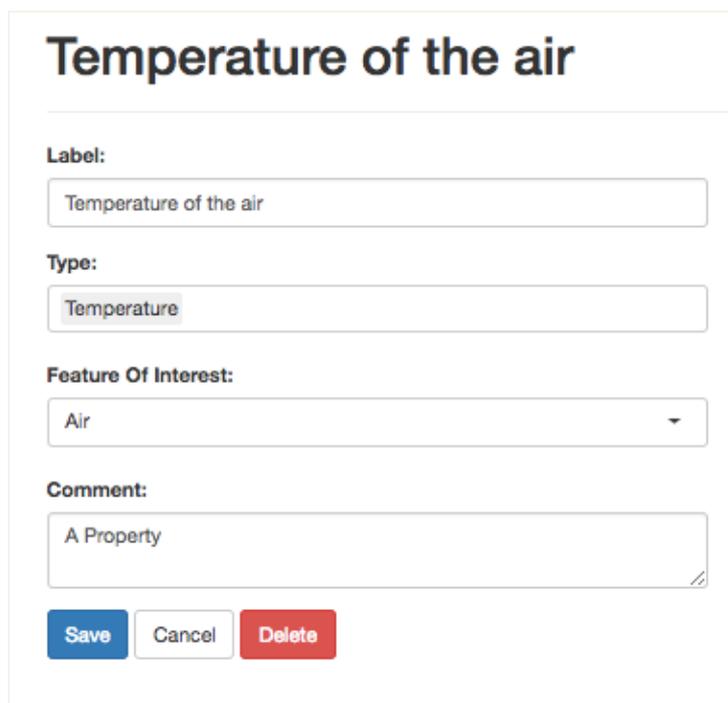
**Name:**  
Temperature of the air

**Comment:**  
A Property

**Create**

**Figura 72-** Tela de criação de uma nova Propriedade de Objeto de Interesse  
Fonte: Elaboração própria

A Figura 72 apresenta a tela de criação da nova Propriedade de Objeto de Interesse “Temperature of the air” (Temperatura do ar). O URI do recurso é construído com base no valor inserido no campo ID (URI). Maiores detalhes sobre a propriedade em questão são inseridos na tela de edição, vide Figura 73.



**Temperature of the air**

**Label:**  
Temperature of the air

**Type:**  
Temperature

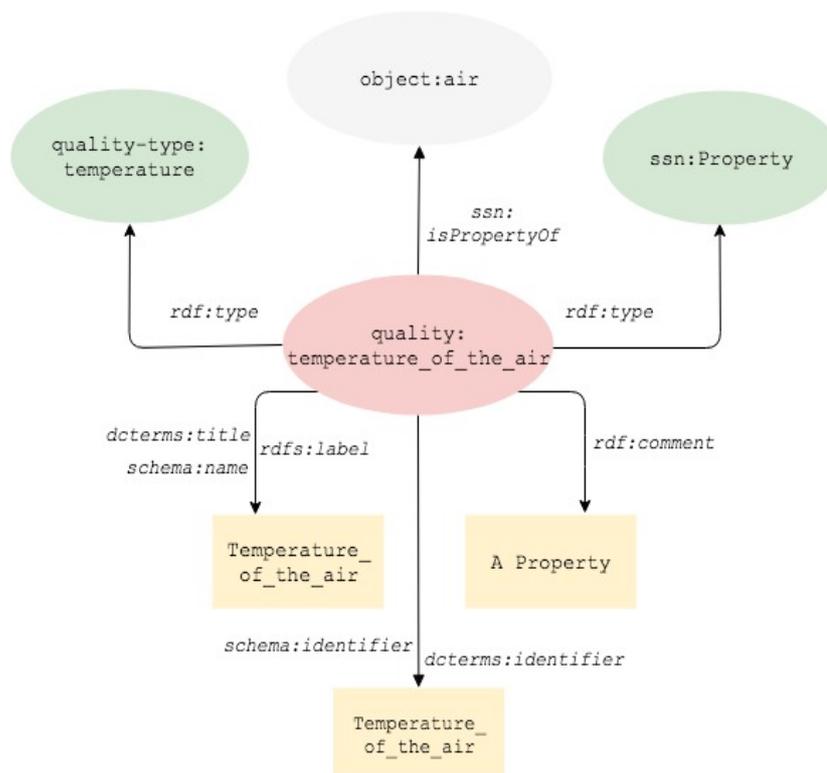
**Feature Of Interest:**  
Air

**Comment:**  
A Property

**Save** **Cancel** **Delete**

**Figura 73-** Tela de edição de uma nova Propriedade de Objeto de Interesse  
Fonte: Elaboração própria

Na Figura 73, no campo “Type”, a classe na qual a propriedade “Temperature of the air” pertence é selecionada, a saber: “Temperature”. A classe “Temperature” consiste em um “Tipo de propriedade” previamente cadastrado no catálogo. Assim, o recurso “Temperature of the air” é uma instância da classe “Temperature”. Além disso, no campo “Feature Of Interest” a propriedade catalogada é associada ao Objeto de Interesse “Air” (ar), também previamente cadastrado na aplicação. As informações referentes à Propriedade “Temperature of the air” são apresentadas em grafo RDF na Figura 74.



**Figura 74** - Grafo RDF subjacente ao cadastro de propriedade de objeto de interesse “Temperature of the air”

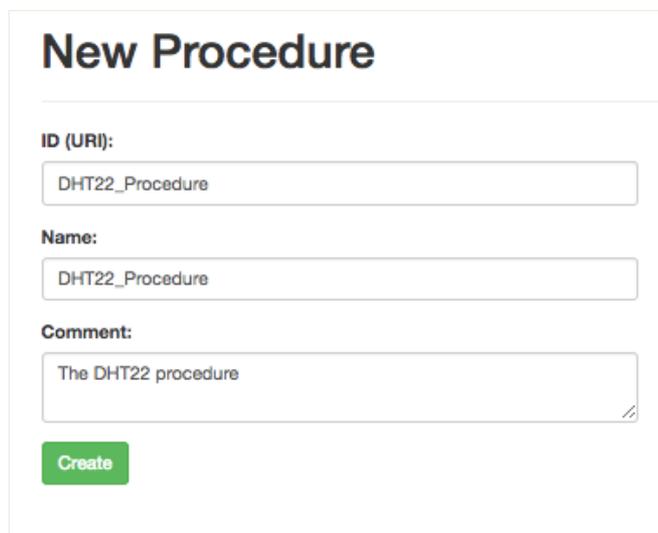
Fonte: Elaboração própria

A Figura 74 evidencia tripla que define o recurso em questão como sendo uma instância da classe `ssn:Property`. Além disso, uma segunda tripla RDF descreve o recurso como sendo instância da classe “Temperature” por meio do uso da propriedade `rdf:type`. A definição da propriedade catalogada como sendo, por exemplo, instância de `sosa:ObservableProperty` fica a cargo da inferência.

### 6.1.7. Cadastro de Procedimento

No catálogo semântico proposto, é possível associar Procedimentos a Propriedades de Objetos de Interesse (de observação ou atuação). Em alguns casos, um mesmo procedimento pode estar associado a uma única propriedade de objeto de interesse. Em outros, um mesmo

Procedimento pode estar associado a mais de uma Propriedade de Objeto de Interesse. A Figura 75 ilustra a criação de um novo Procedimento no catálogo.



**New Procedure**

**ID (URI):**  
DHT22\_Procedure

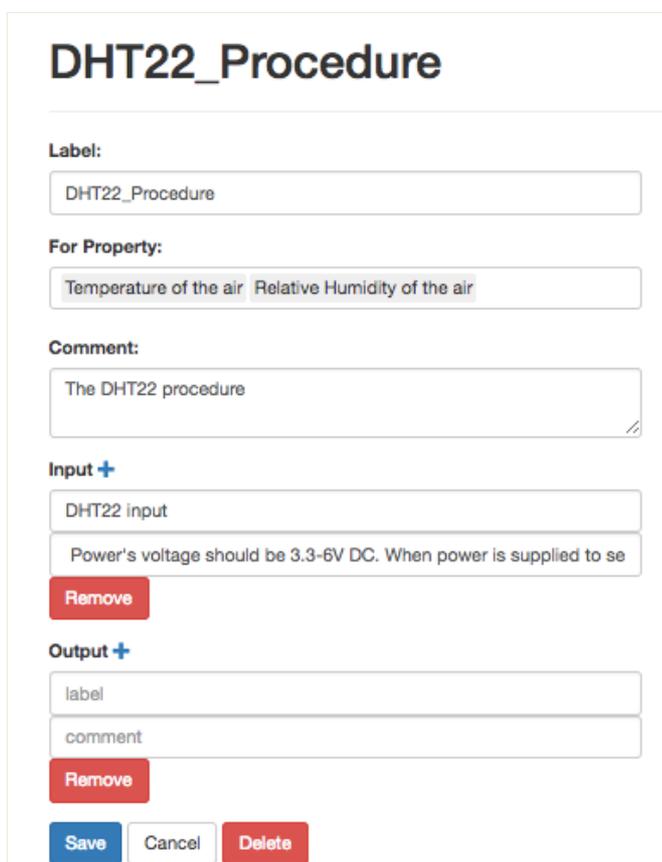
**Name:**  
DHT22\_Procedure

**Comment:**  
The DHT22 procedure

**Create**

**Figura 75-** Tela de criação de um novo Procedimento  
Fonte: Elaboração própria

Na Figura 75, o valor “DHT22\_Procedure” inserido no campo “ID (URI)” irá compor o URI do recurso catalogado. Maiores detalhes sobre o procedimento em questão podem ser inseridos na tela de edição do recurso, vide Figura 76.



**DHT22\_Procedure**

**Label:**  
DHT22\_Procedure

**For Property:**  
Temperature of the air Relative Humidity of the air

**Comment:**  
The DHT22 procedure

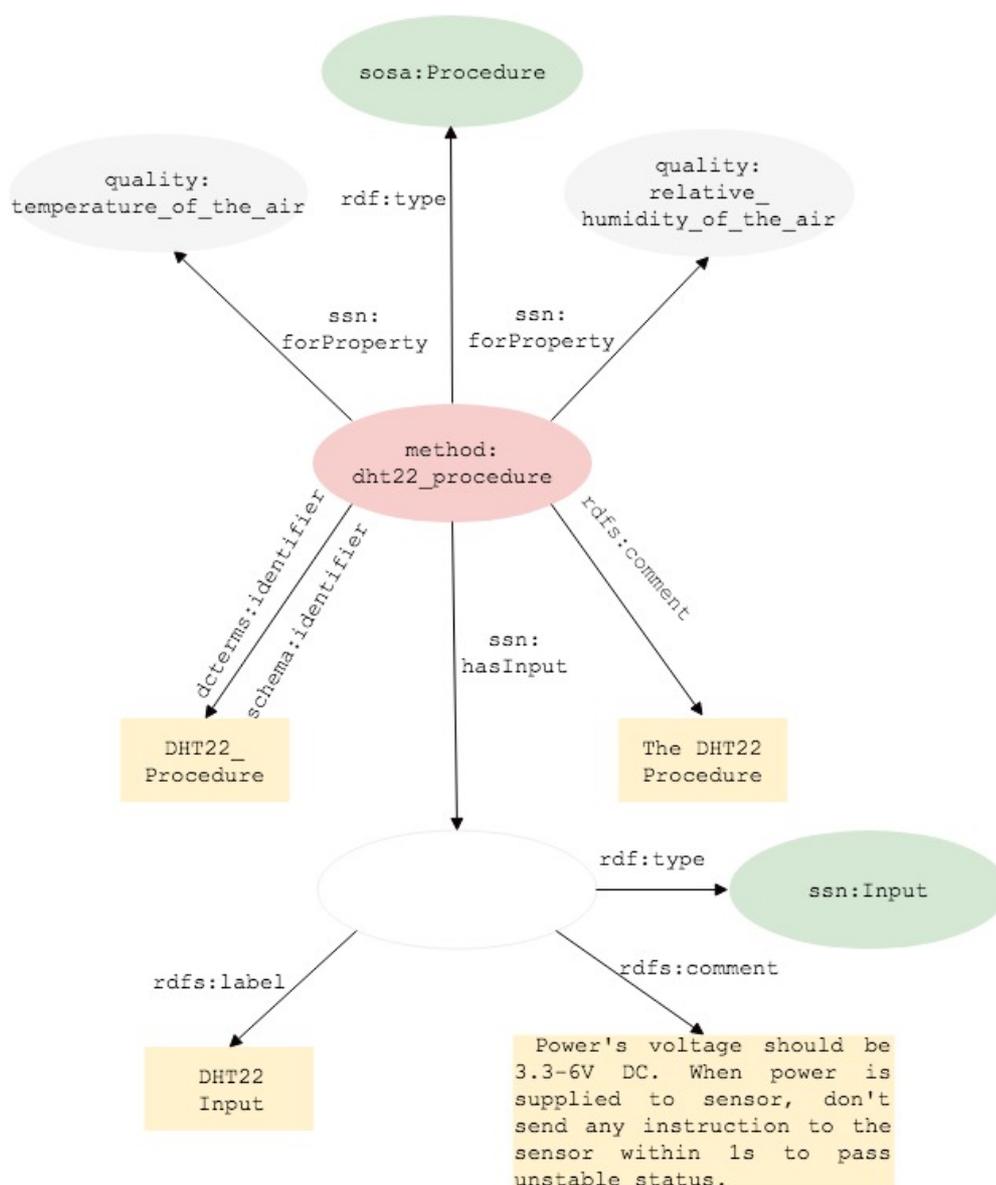
**Input +**  
DHT22 input  
Power's voltage should be 3.3-6V DC. When power is supplied to se  
**Remove**

**Output +**  
label  
comment  
**Remove**

**Save** **Cancel** **Delete**

**Figura 76-** Tela de edição de um novo Procedimento  
Fonte: Elaboração própria

Na Figura 76, no campo “For Property”, as propriedades de objeto de interesse “Temperature of the air” e “Relative Humidity of the air” são associadas à descrição do procedimento do modelo de sensor DHT22. Isso ocorre porque este modelo observa tanto a propriedade “Temperature of the air” quanto a propriedade “Relative Humidity of the air” (Umidade relativa do ar). Diferente de muitos modelos de sensores, o DHT22 apresenta um único procedimento. O parâmetro de entrada do procedimento em questão pôde ser identificado no manual do modelo e pode ser visualizado no campo “Input” da tela de edição (Figura 76). Parâmetros de saída (“Output”) do modelo DHT22 não foram encontrados no manual do sensor e por isso não foram associados ao campo “Output”. As informações apresentadas nas telas de criação e edição referentes ao cadastro do Procedimento “DHT22\_Procedure” são apresentadas em *graph pattern* instanciado, na Figura 77.



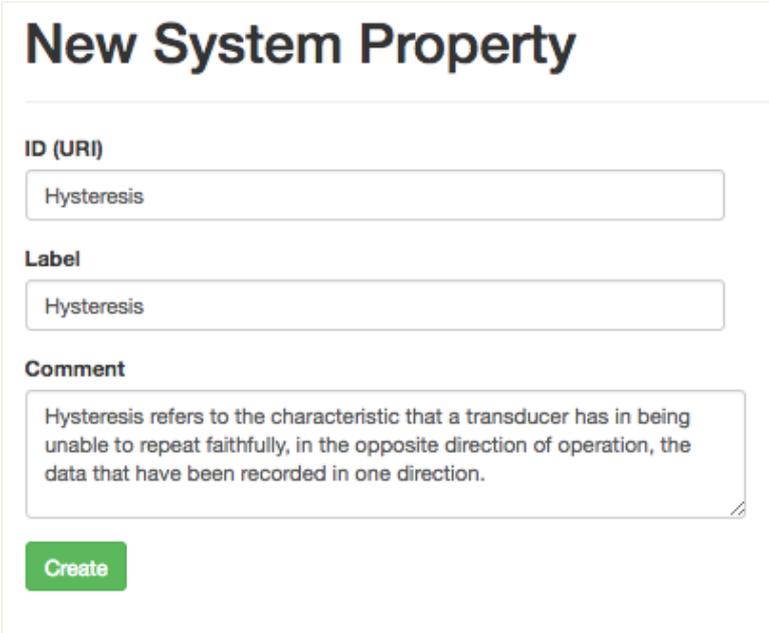
**Figura 77** - Grafo RDF de procedimento do modelo de sensor DHT22

Fonte: Elaboração própria

Na Figura 77, é possível verificar o nó em branco associado ao procedimento catalogado. Tal nó em branco apresenta, apenas, detalhes sobre o parâmetro de entrada (Input) do procedimento catalogado. Nesse caso, o parâmetro de entrada é descrito fazendo uso das propriedades *rdfs:label* e *rdfs:comment*. Assim, os valores nestas propriedades definem o parâmetro de entrada. Por se tratar de nó em branco, ao apagar o procedimento catalogado, as informações de entrada deste procedimento também são apagadas do banco de dados da aplicação.

#### 6.1.8. Cadastro de Propriedade de Sistema (System Property)

Alguns modelos de sensores podem apresentar propriedades de sistema, além daquelas previstas na ontologia SSN. Por exemplo, a propriedade de sistema histerese (“Hysteresis”), muito comum em sensores, não foi prevista na versão atual da ontologia SSN. Nesse caso, é possível cadastrar a propriedade em questão no catálogo proposto. A Figura 78 ilustra a criação da nova propriedade de sistema em questão.



**New System Property**

**ID (URI)**  
Hysteresis

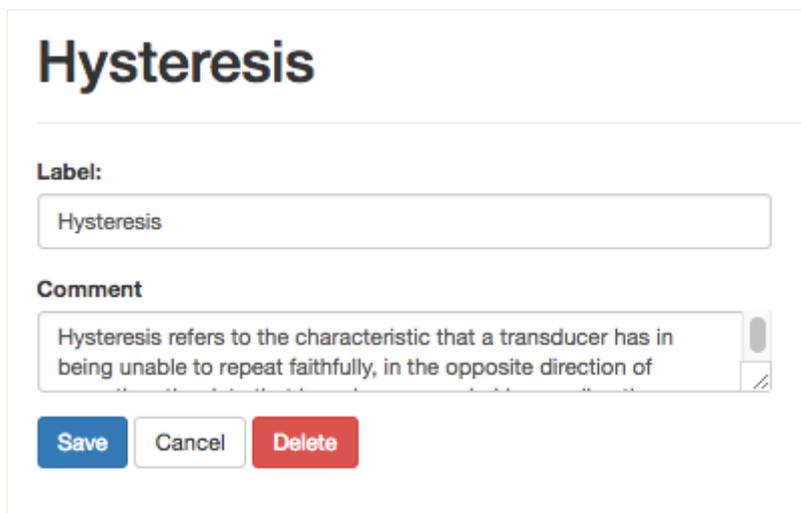
**Label**  
Hysteresis

**Comment**  
Hysteresis refers to the characteristic that a transducer has in being unable to repeat faithfully, in the opposite direction of operation, the data that have been recorded in one direction.

**Create**

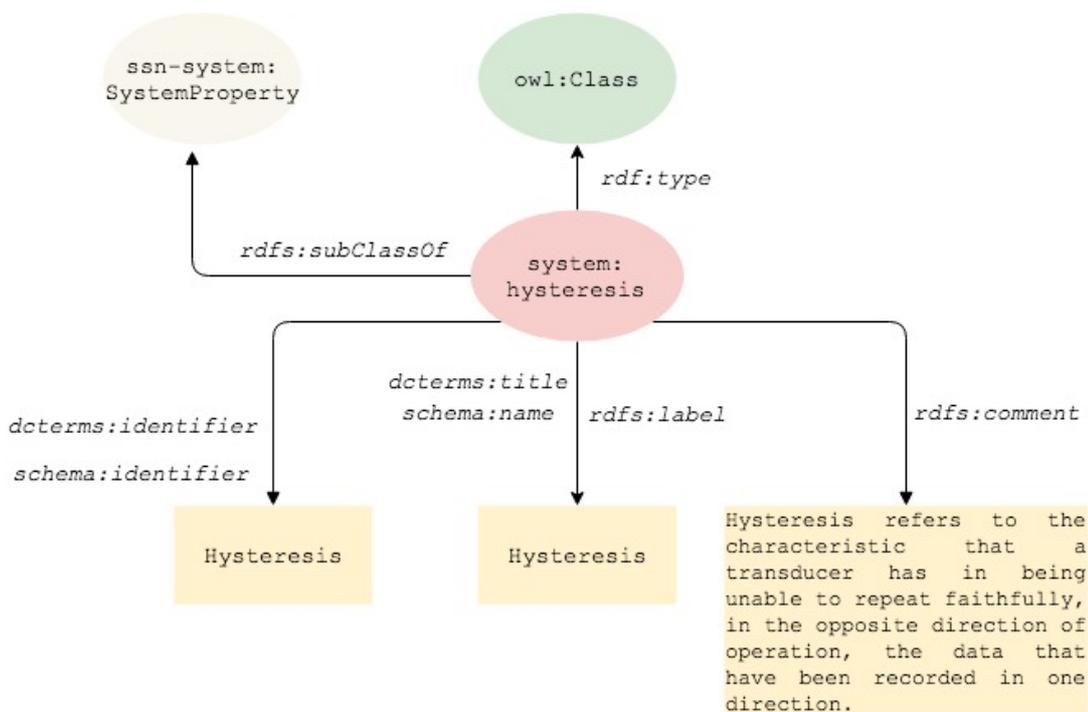
**Figura 78-** Tela de criação de nova propriedade de sistema  
Fonte: Elaboração própria

Na Figura 78, o URI do recurso será gerado com base no valor indicado no campo “ID (URI)”. Assim, parte do URI do recurso catalogado deve conter o termo “Hysteresis”. A Figura 79 apresenta a tela de edição da propriedade de sistema catalogada, na qual as informações inseridas na tela de criação podem ser modificadas.



**Figura 79-**Tela de edição de propriedade de sistema  
Fonte: Elaboração própria

As informações inseridas na tela de criação da Figura 78 podem ser editadas, exceto o valor inserido no campo “ID (URI)”. Assim, apenas os valores inseridos nos campos “Label” e “Comment” podem ser modificados pelo usuário na tela de edição (Figura 79). O grafo RDF adjacente ao cadastro apresentado nas telas 78 e 79 pode ser visualizado na Figura 80.



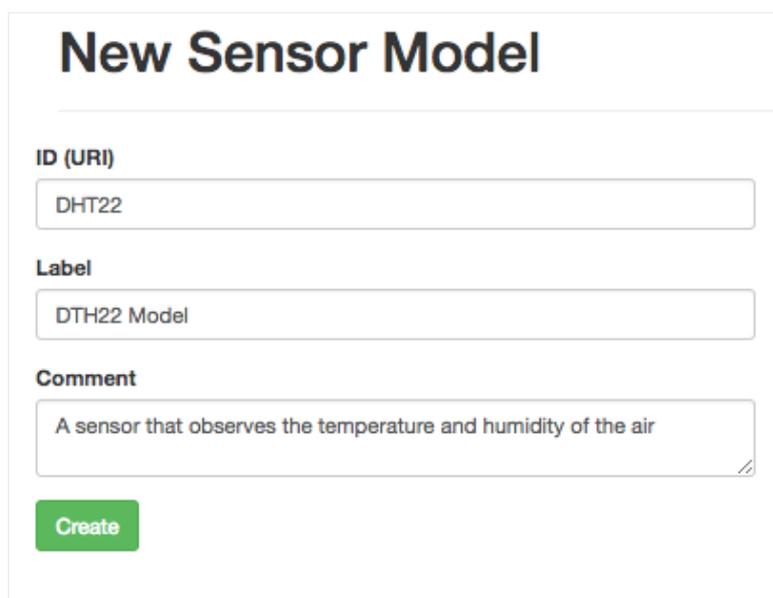
**Figura 80-** Grafo RDF subjacente ao cadastro de propriedade de sistema  
Fonte: Elaboração própria

Na Figura 80, a nova propriedade de sistema “Hysteresis” consiste em subclasse da classe *ssn-system:SystemProperty*. Tendo em vista que as propriedades que descrevem propriedades de sistema são as mesmas utilizadas para descrever propriedades de operação e sobrevivência,

as telas de criação e edição destas propriedades são idênticas. Assim, as telas desenvolvidas para catalogar propriedades de operação e sobrevivência não serão apresentadas neste trabalho.

### 6.1.9. Cadastro de Modelo de Sensor (Sensor Model)

O modelo ontológico para descrição de Modelos de sensores prevê a associação das características técnicas dos dispositivos. Assim, a catalogação dos modelos demanda diferentes recursos, tais como procedimentos, propriedades de observação, capacidade de sistema, faixa de operação, etc. Para ilustrar a descrição de modelos de sensores, optou-se por apresentar o cadastro do modelo DHT22. Este modelo de sensor é capaz de observar a temperatura e a umidade do ar. A Figura 81 apresenta a tela de criação do modelo de sensor DHT22.



The image shows a web form titled "New Sensor Model". It contains three input fields: "ID (URI)" with the text "DHT22", "Label" with the text "DTH22 Model", and "Comment" with the text "A sensor that observes the temperature and humidity of the air". Below the fields is a green button labeled "Create".

**Figura 81-** Tela de criação do modelo de sensor DHT22

Fonte: Elaboração própria

Na Figura 81, o URI do modelo cadastrado será construído com base no valor inserido no campo "ID (URI)". Outras informações podem ser incluídas à descrição de modelos de sensores na tela de edição. Por motivo de simplificação, a tela de edição será apresentada de forma segmentada. Além disso, para facilitar a compreensão, o grafo RDF subjacente ao cadastro do modelo DHT22 também será apresentado de forma segmentada.

A Figura 82 apresenta parte da tela de edição do modelo de sensor na qual o nome, a propriedade observada, o procedimento, o fabricante do modelo e a pessoa responsável pelo cadastro são evidenciados.

### Sensor Model

**Label:**

DHT22 Model

**Observes (Property):**

Infrared signal  
 Infrared signal of WOT deployment room  
**Relative Humidity of the air**  
 Temperature of WOT deployment room  
**Temperature of the air**

**Procedure:**

**DHT22\_Procedure**  
 IP68\_Outdoor\_Temperature\_Sensor#P  
 IP68\_Outdoor\_Temperature\_Sensor#b  
 IP68\_Outdoor\_Temperature\_Sensor#te

**Producer/creator:**

Organization/1  
**Sparkfun Eletronics**

**Comment:**

A sensor that observes the  
 temperature and humidity of the air

**Registered by:**

**Mark Zuckerberg**

**Figura 82-** Parte da tela de edição do modelo de sensor DHT22

Fonte: Elaboração própria

Na Figura 82, é possível verificar algumas informações que descrevem o modelo de sensor DHT22. Por exemplo, no campo “Observes (Property)” as duas propriedades observadas pelo modelo em questão são selecionadas, a saber: “Temperature of the air” e “Relative humidity of the air” (“Temperatura do ar” e “Umidade relativa do ar”). Além disso, o procedimento executado por este modelo é associado no campo “Procedure”. Já os campos “Producer/creator” e “Registered by” fazem referência ao fabricante do modelo e a pessoa responsável pelo cadastro, respectivamente. Todos estes recursos, utilizados para descrever o modelo do sensor, foram previamente cadastrados na aplicação e associados por meio de consultas SPARQL.

Ainda na tela de edição do modelo (Figura 82), é possível descrever o recurso cadastrado sob o ponto de vista de suas capacidades de sistema, faixa de operação e de sobrevivência. A

Figura 83 apresenta parte da tela de edição com foco na capacidade de sistema do modelo de sensor DHT22.

**System Capability**

**For property:**

- Infrared signal of WOT deployment roc
- Relative Humidity of the air
- Temperature of WOT deployment room
- Temperature of the air
- Temperature of the room 125

**Condition +**

**Label:**

TemperatureNormalCondition

**Value**

-40

80

**Unit of measurement:**

- Decibel Referred to 1mw
- Decimeter
- Degree Angle
- Degree Celsius
- Degree Celsius Centimeter

**Comment**

Remove

**Figura 83-** Tela de edição do modelo de sensor DHT22 com foco na capacidade de sistema

Fonte: Elaboração própria

Descrever a capacidade de sistema de um dado modelo de sensor significa, em termos práticos, associá-lo a propriedades de sistema sob determinadas condições, especificando as propriedades observadas. Assim, ao catalogar o modelo DHT22 propriedades (de observação) previamente cadastradas na aplicação puderam ser selecionadas no campo “For Property”, a saber: “Temperature of the air” e “Relative Humidity of the air” (Figura 83). Neste mesmo exemplo, a capacidade de sistema também é descrita sob o ponto de vista de suas condições de funcionamento, são elas: condições normais de temperatura e de umidade. A Figura 84 apresenta maiores detalhes das condições normais de umidade do modelo DHT22.

**Label:**

HumidityNormalCondition

**Value**

5

85

**Unit of measurement:**

Per Mole Unit  
Per Square Giga Electron Volt Unit  
Per Tesla Meter Unit  
Per Tesla Second Unit  
Percent

**Comment**

Remove

**Figura 84-** Parte da tela de edição do modelo de sensor DHT22 com foco na capacidade de sistema e condições normais de funcionamento

Fonte: Elaboração própria

Com base no manual do modelo do sensor DHT22, as condições de temperatura e umidade variam entre  $-40$  e  $80$  °C e 5 a 85%, respectivamente. Nas Figuras 83 e 84, as unidades de medida destas condições, associadas no campo “Unit of measurement”, têm como origem a ontologia QUTD. Por meio de consulta SPARQL diferentes classes desta ontologia, que representa diferentes unidades de medida, podem ser selecionadas pelo usuário (p. ex.: “Degree Celsius” e “Percent”).

A descrição da capacidade de sistema de um dado modelo de sensor envolve uma ou mais propriedades de sistema. No caso do modelo DHT22, precisão, acurácia, sensibilidade, histerese são algumas propriedades especificadas pelo fabricante no manual do modelo. A título de exemplo, a Figura 85 ilustra a associação da propriedade precisão (“Precision”) à catalogação do modelo DHT22 no catálogo.

**System Property**

Select System Property +

- Latency
- Measurement Range
- Precision**
- Resolution
- Response Time

Value

-0.2

0.2

**Unit of measurement:**

- Decibel Referred to 1mW
- Decimeter
- Degree Angle
- Degree Celsius**
- Degree Celsius Centimeter

**For Property:**

- Relative Humidity of the air
- Temperature of WOT deployment room
- Temperature of the air**
- Temperature of the room 125

**Condition +**

**Remove**

**Figura 85-** Parte da tela de edição do modelo de sensor DHT22 com foco na propriedade de sistema “Precision”

Fonte: Elaboração própria

A Figura 85 evidencia as diferentes propriedades de sistemas que podem ser encontradas na ontologia SSN, tais como precisão (Precision), resolução (Resolution), tempo de resposta (Response Time), etc. A disponibilização de tais propriedades é resultado de consulta SPARQL que retorna todos os recursos armazenados no banco de dados da aplicação que consistem em subclasses da classe *ssn-system:SystemProperty*. Nesse caso, tanto as propriedades de sistema preexistentes na ontologia SSN e aquelas catalogadas pelos usuários são retornadas nesta consulta. Como se trata da catalogação do modelo DHT22, o valor da precisão para este modelo varia entre -0.2 e 0.2 °C. Além disso, esta propriedade e valor referem-se à propriedade observada “Temperature of the air”, como evidenciado no campo “For Property” (Figura 85). Assim, torna-se explícito que tais informações valem apenas para a propriedade indicada, tendo em vista que este modelo observa também a propriedade “Relative Humidity of the air”.

Além da propriedade “Precisão”, o manual do modelo DHT22 apresenta informações sobre a propriedade de sistema “Histerese” (Hysteresis). Segundo o manual, a propriedade em questão refere-se ao processo de observação da “Umidade relativa do ar” (“Relative Humidity of the air”). O cadastro da propriedade “Hysteresis” foi previamente realizado, tendo em vista que tal propriedade não pôde ser encontrada na ontologia SSN. A Figura 86 apresenta parte da tela edição do cadastro do modelo de sensor DHT22 com foco na associação da propriedade “Hysteresis”.

The screenshot displays the 'System Property' configuration window. It features a title bar 'System Property' and a section 'Select System Property +' with a list of properties: Response Time, Selectivity, Sensitivity, and Hysteresis (highlighted). Below this, there are two input fields for 'Value' containing '-0.3' and '0.3'. The 'Unit of measurement:' section includes a list with 'Percent' selected. The 'For Property:' section has a list with 'Relative Humidity of the air' selected. A 'Condition +' button is visible at the bottom.

**Figura 86**-Tela de edição do modelo DHT22 com foco na propriedade de sistema “Hysteresis”

Fonte: Elaboração própria

Na Figura 86, a propriedade “Hysteresis” (“Histerese”) selecionada tem seu valor definido como sendo  $\pm 0.3\%$  para a propriedade “Relative Humidity of the air” (“Umidade relativa do ar”). Isso significa dizer a “Histerese” descrita refere-se, apenas, ao processo de observação da “Umidade relativa do ar”. As condições não foram descritas nesta parte da tela de edição, tendo em vista que tais informações já foram inseridas na descrição da capacidade de sistema do modelo DHT22, vide Figuras 83 e 84.

No manual do modelo DHT22 é possível encontrar informações sobre a faixa de operação do modelo em questão. Assim como no caso da capacidade de sistema, a faixa de

operação pode ser descrita de acordo com a propriedade observada e condições de funcionamento. A Figura 87 apresenta parte da tela de edição de modelo de sensor com foco na descrição da faixa de operação do modelo DHT22.

**Operating Range**

**For property:**

- Infrared signal of WOT deployment room
- Relative Humidity of the air
- Temperature of WOT deployment room
- Temperature of the air

**Condition +**

**Label:**

TemperatureNormalCondition

**Value**

-40

80

**Unit of measurement:**

- Decibel Carrier Unit
- Decibel Referred to 1mw
- Decimeter
- Degree Angle
- Degree Celsius

**Comment**

Remove

**Figura 87-** Tela de edição do modelo DHT22 com foco na faixa de operação

Fonte: Elaboração própria

Na Figura 87, as propriedades relacionadas à faixa de operação do modelo DHT22 são “Temperature of the air” e “Relative Humidity of the air”. Além disso, a Figura 87 evidencia a condição na qual o sensor deve ser submetido para que a faixa de operação seja aquela prevista pelo fabricante, a saber: -40 a 80 °C. Ainda, uma segunda condição, que influencia no funcionamento normal do sensor, é indicada nesta parte do cadastro e pode ser visualizada na Figura 88.

The image shows a web-based form for editing a model. It has the following sections:

- Label:** A text input field containing "HumidityNormalCondition".
- Value:** Two stacked text input fields. The top one contains "5" and the bottom one contains "85".
- Unit of measurement:** A dropdown menu with the following options: "Per mole Unit", "Per Square Giga Electron Volt Unit", "Per Tesla Meter Unit", "Per Tesla Second Unit", and "Percent". The "Percent" option is currently selected and highlighted.
- Comment:** A large, empty text area for adding a comment.
- Remove:** A red button with the text "Remove" in white.

**Figura 88-** Tela de edição do modelo DHT22 com foco na condição de funcionamento para faixa de operação

Fonte: Elaboração própria

Na Figura 88, a condição descrita refere-se à condição de umidade na qual modelos de sensores DHT22 devem ser submetidos para que a faixa de operação esteja de acordo com o esperado, a saber: 5 a 85%.

No manual do modelo DHT22, foi possível encontrar informações sobre a tensão de alimentação necessária para que sensores deste modelo operem normalmente quando submetidos às condições previstas (indicadas nas Figuras 87 e 88). A ontologia SSN prevê a descrição da propriedade de operação “Tensão de alimentação” (“Operating Power Range”). Assim, não foi necessário cadastrar tal propriedade no catálogo. Por meio de consulta SPARQL a propriedade em questão pôde ser associada ao cadastro do modelo DHT22, vide Figura 89.

**Operating Property**

Select Operating Property +

- Maintenance Schedule
- Operating Power Range**

Value

3.3

6

Unit of measurement:

- Volt**
- Volt Meter
- Volt per Meter
- Volt per Square Meter
- Volt per second

For property:

- Infrared signal of WOT deployment room
- Relative Humidity of the air
- Temperature of WOT deployment room
- Temperature of the air
- Temperature of the room 125**

Condition +

Remove

**Figura 89-**Tela de edição do modelo DHT22 com foco na propriedade de operação “Operating Power Range”

Fonte: Elaboração própria

Na Figura 89, a propriedade de operação “Faixa de tensão de alimentação” (“Operating Power Range”) é associada à descrição do modelo DHT22. O valor da “Faixa de tensão de alimentação” varia entre 3.3 e 6 V para que o sensor opere corretamente, considerando as condições normais de temperatura e umidade. Tal faixa de operação é aplicável tanto na observação da propriedade temperatura quanto da propriedade umidade. Tendo em vista que tais propriedades observadas já foram associadas à “Faixa de operação” e que a propriedade de operação “Faixa de tensão de alimentação” faz parte da faixa de operação, não foi preciso especificar qualquer propriedade no campo “For Property”.

Informações sobre a faixa de sobrevivência e propriedades de sobrevivência do modelo DHT22 não puderam ser encontradas no manual do modelo. Assim, a título de exemplo, a parte da tela de edição para descrição da “Faixa de sobrevivência” e “Propriedades de sobrevivência” será apresentada fazendo uso de valores fictícios, vide Figuras 90 e 91.

### Survival Range

**For property:**

- Relative Humidity of the air
- Relative Humidity of the room 125
- Temperature of WOT deployment room
- Temperature of the air
- Temperature of the room 125

**Condition +**

**Label:**

TemperatureNormalCondition

Value

-40

80

**Unit of measurement:**

- Decibel Heterred to 1mw
- Decimeter
- Degree Angle
- Degree Celsius
- Degree Celsius Centimeter

**Comment**

**Remove**

**Figura 90-**Tela de edição do modelo de sensor DHT22 com foco na faixa de sobrevivência  
 Fonte: Elaboração própria

A Figura 90 apresenta a tela para descrição da faixa de sobrevivência (Survival Range) do modelo DHT22 para as propriedades observadas “Temperature of the air” e “Relative Humidity of the air” e condição normal de funcionamento de -40 a 80 °C. A Figura 91 apresenta a tela na qual a propriedade de sobrevivência pode ser associada ao cadastro do modelo DHT22.

**Survival Property**

**Select Survival Property +**

- Battery Lifetime
- System Lifetime**

5

min Value

max Value

**Unit of measurement:**

- wattnour
- Weber
- Yard
- Year (365 Day)**
- Year Sidereal

**For Property:**

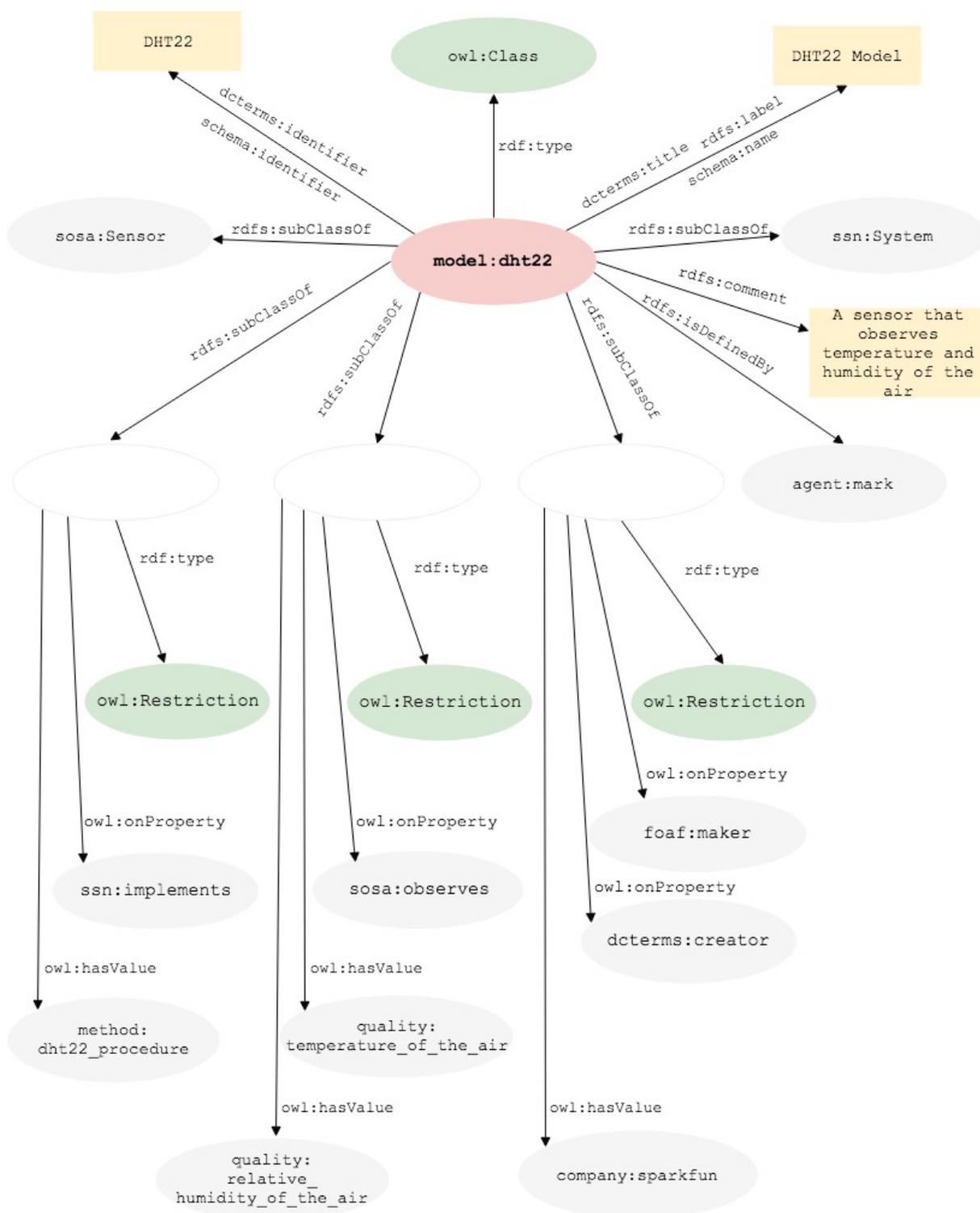
- Air?lat=45.75&long=4.85#temperature
- Battery state of the IP68 Outdoor Temp
- Content of LCD G240641C Actuator
- Content\_of\_LCD\_panel

**Condition +**

**Figura 91**-Tela de edição de modelo de sensores com foco na propriedade de sobrevivência  
Fonte: Elaboração própria

A Figura 91 evidencia as propriedades de sobrevivência previstas na ontologia SSN, a saber: “Battery Lifetime” e “System Lifetime”. A propriedade “System Lifetime” selecionada diz respeito ao tempo de vida útil de modelos de sensores DHT22. Ao fazer uso de sensor deste modelo para observação de temperatura e/ou umidade e submetê-lo às condições de temperatura ambiente descritas na Figura 90 (-40 a 80 °C), tal dispositivo apresentará tempo de vida previsto pelo fabricante, cujo valor é evidenciado na Figura 91 (5 anos).

Não foi necessário inserir as condições de funcionamento na descrição da propriedade de sobrevivência, tendo em vista que tais informações são as mesmas associadas à faixa de operação, vide Figura 91. As informações apresentadas nas telas de criação e edição referentes ao cadastro do modelo DHT22 são apresentadas em *graph pattern* instanciado na Figura 92.



**Figura 92** - Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22

Fonte: Elaboração própria

A Figura 92 apresenta parte do grafo RDF subjacente ao cadastro do modelo DHT22. Nesta parte do grafo, recursos previamente cadastrados no catálogo, tais como procedimento, propriedades observadas, fabricante e pessoa responsável pelo cadastro são evidenciados. Ainda, é possível verificar as propriedades utilizadas para relacionar o modelo de sensor cadastrado e tais recursos. No entanto, a parte do grafo apresentada na Figura 92 não evidencia



apresenta parte da descrição da capacidade de sistema do modelo DHT22 com foco na condição de umidade para funcionamento normal do dispositivo.

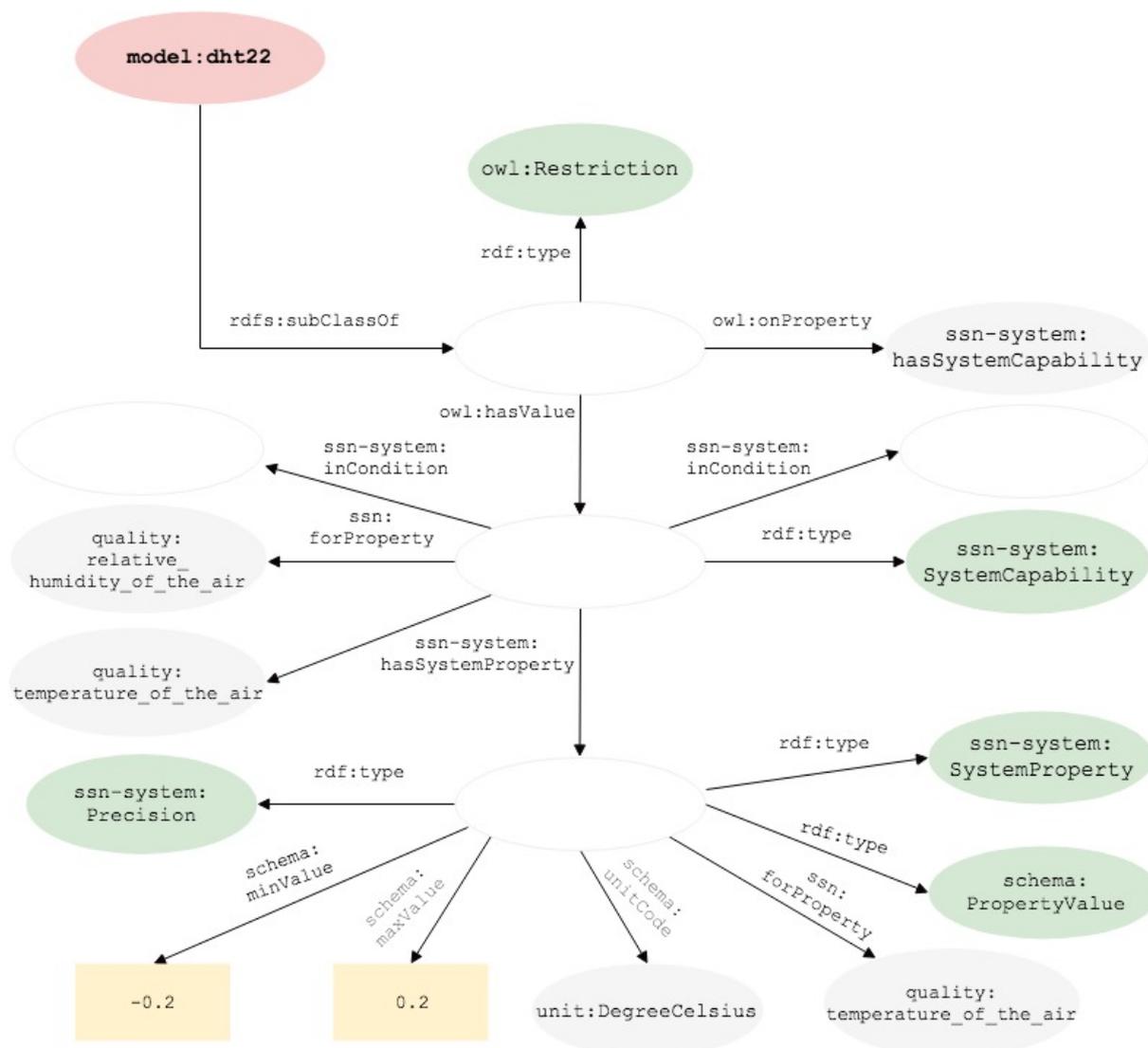


**Figura 94-** Parte do grafo RDF subjacente ao cadastro do modelo DHT22 com foco na capacidade de sistema e sua condição de umidade

Fonte: Elaboração própria

No grafo apresentado na Figura 94, o valor de umidade no qual sensores (modelo DHT22) devem ser submetidos para funcionarem corretamente está entre 5 e 85 %. Tal condição vale tanto para o processo de observação da temperatura quanto da umidade do ar. Por este motivo, a capacidade de sistema foi associada as propriedades de objeto de interesse “*quality:temperature\_of\_the\_air*” (Temperatura do ar) e “*quality:relative\_humidity\_of\_the\_air*” (Umidade relativa do ar), como evidenciado nas Figuras 93 e 94.

Além disso, capacidade de sistema deste modelo de sensor é composta de diferentes propriedades de sistema. A Figura 95 apresenta parte do grafo RDF da capacidade de sistema do modelo DHT22 com foco na propriedade de sistema “Precisão”.

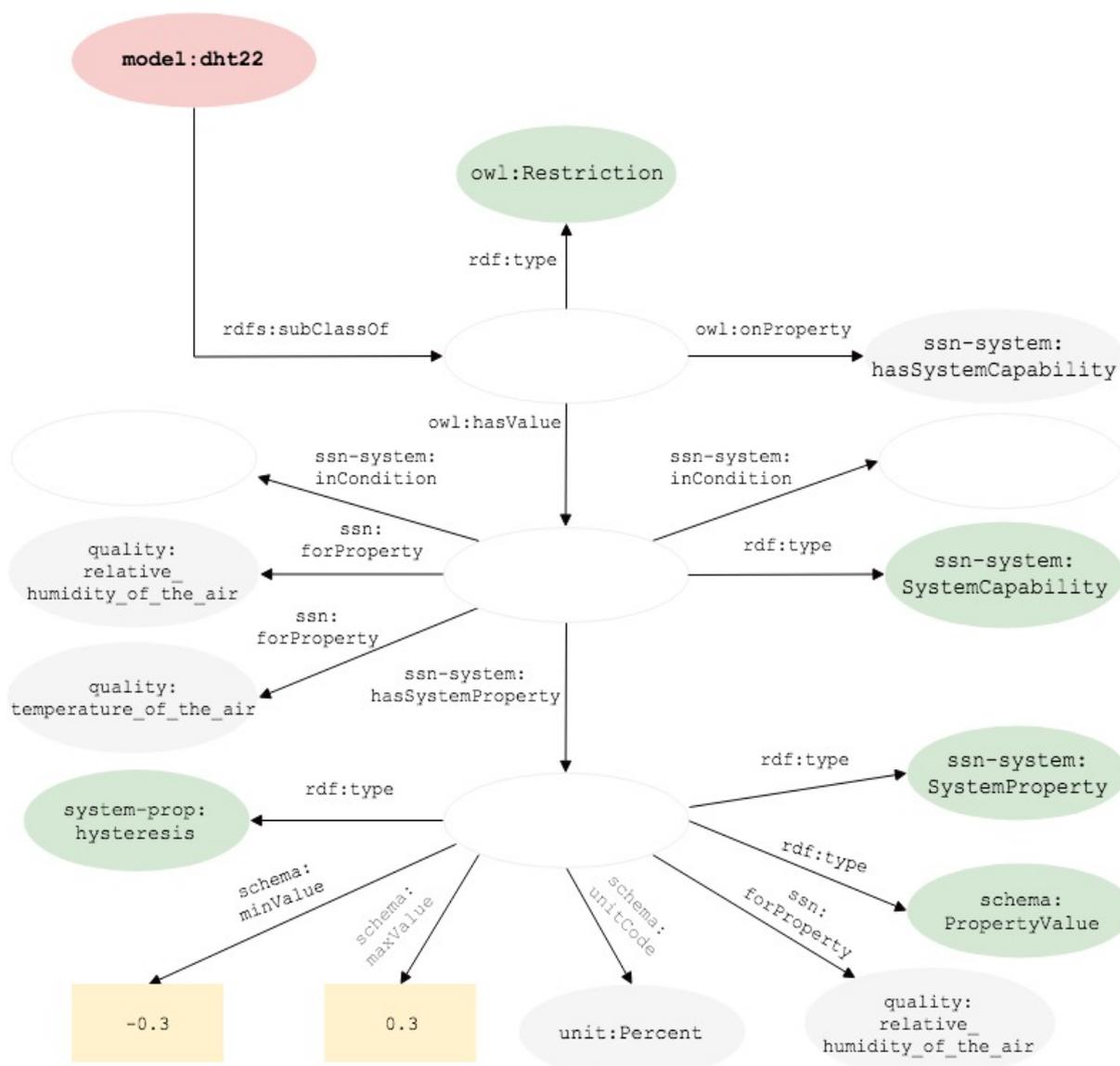


**Figura 95**-Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de sistema “Precisão”

Fonte: Elaboração própria

A Figura 95 evidencia um exemplo de propriedade de sistema do modelo DHT22 “Precision” (Precisão). Como este modelo é capaz de observar duas propriedades distintas, optou-se por vincular a propriedade de objeto de interesse “Temperatura do ar” à propriedade de sistema “Precisão”. Desse modo fica explícito que a propriedade de sistema em questão é válida, apenas, para o processo de observação da “Temperatura do ar”.

Além da propriedade de sistema precisão, o modelo DHT22 apresenta, ainda, a propriedade “Hysteresis” (Histerese). A Figura 96 evidencia mais detalhes sobre esta propriedade.

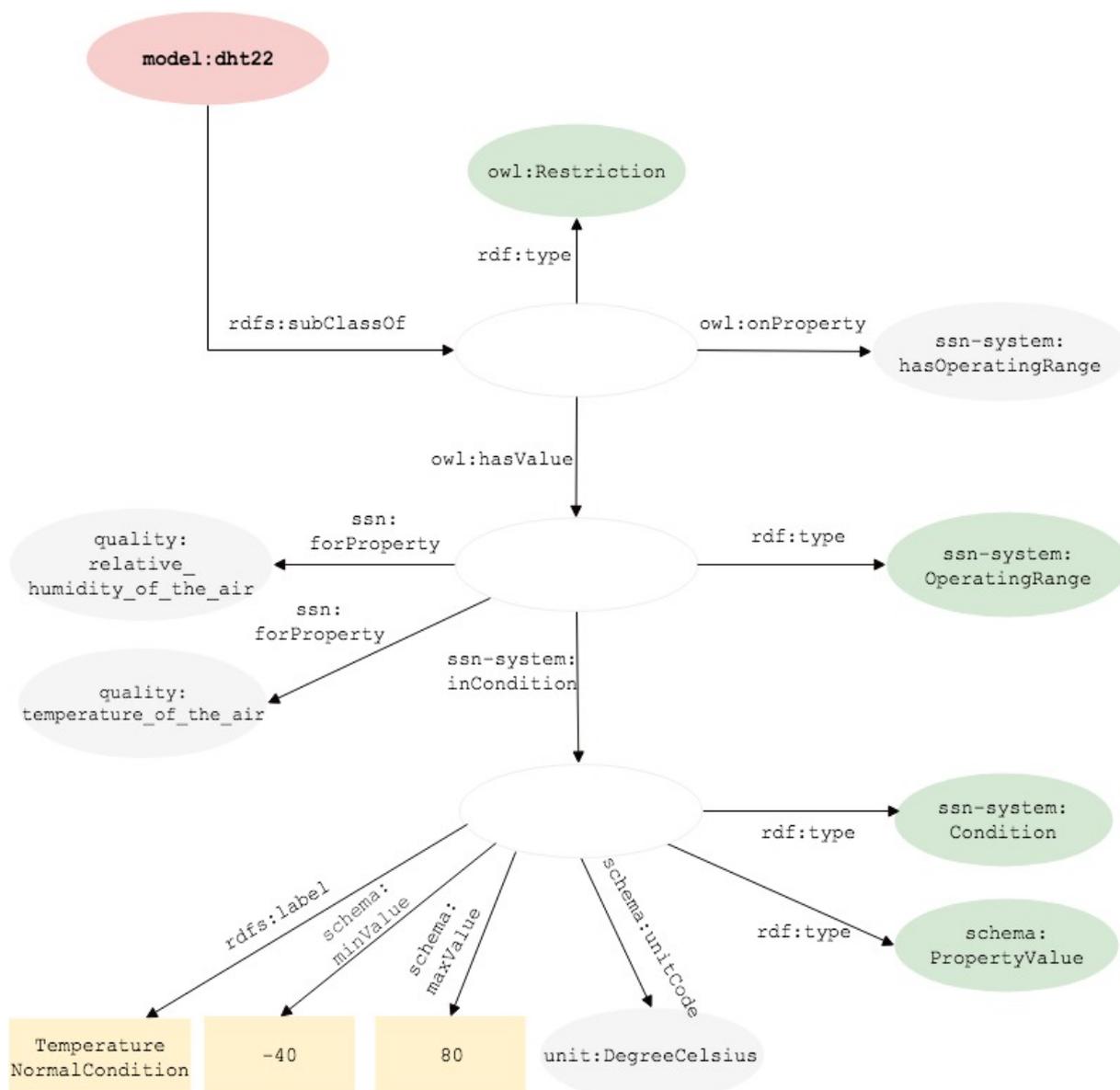


**Figura 96-** Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de sistema “Hysteresis”

Fonte: Elaboração própria

Na Figura 96, a propriedade de sistema “Hysteresis” (Histerese) é evidenciada. Tal propriedade foi previamente cadastrada pelo usuário e associada à catalogação do modelo DHT22. Ao associar a propriedade “Umidade Relativa do ar” à propriedade de sistema Histerese (por meio de *ssn:forProperty*), fica explícito que a propriedade de sistema em questão é válida, apenas, para o processo de observação da “Umidade Relativa do ar”.

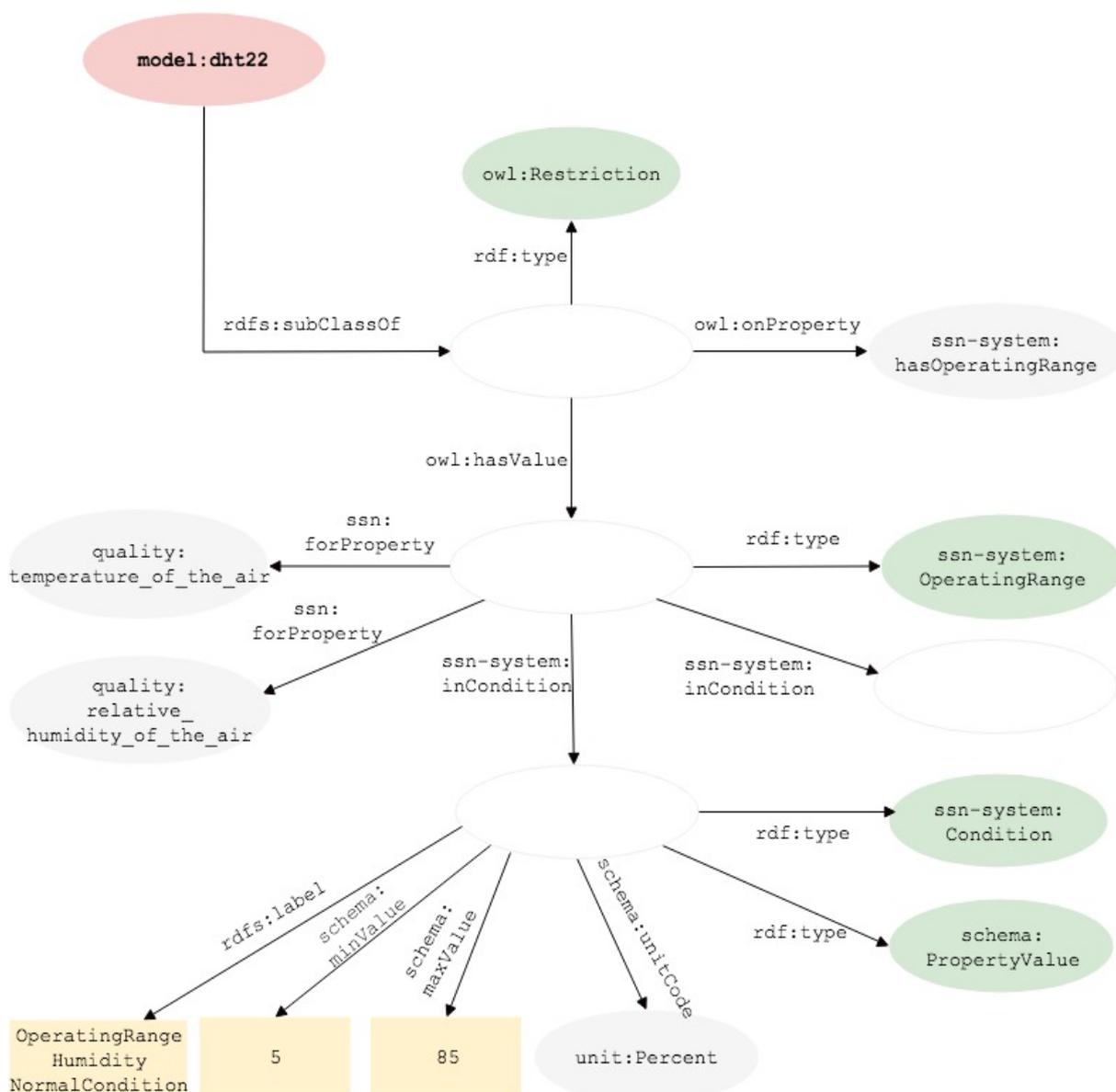
Na catalogação de modelos de sensores, além da capacidade de sistema, é possível descrever também a faixa de operação dos modelos cadastrados. A Figura 97 apresenta parte a descrição da faixa de operação do modelo DHT22.



**Figura 97-** Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na faixa de operação e condição de temperatura

Fonte: Elaboração própria

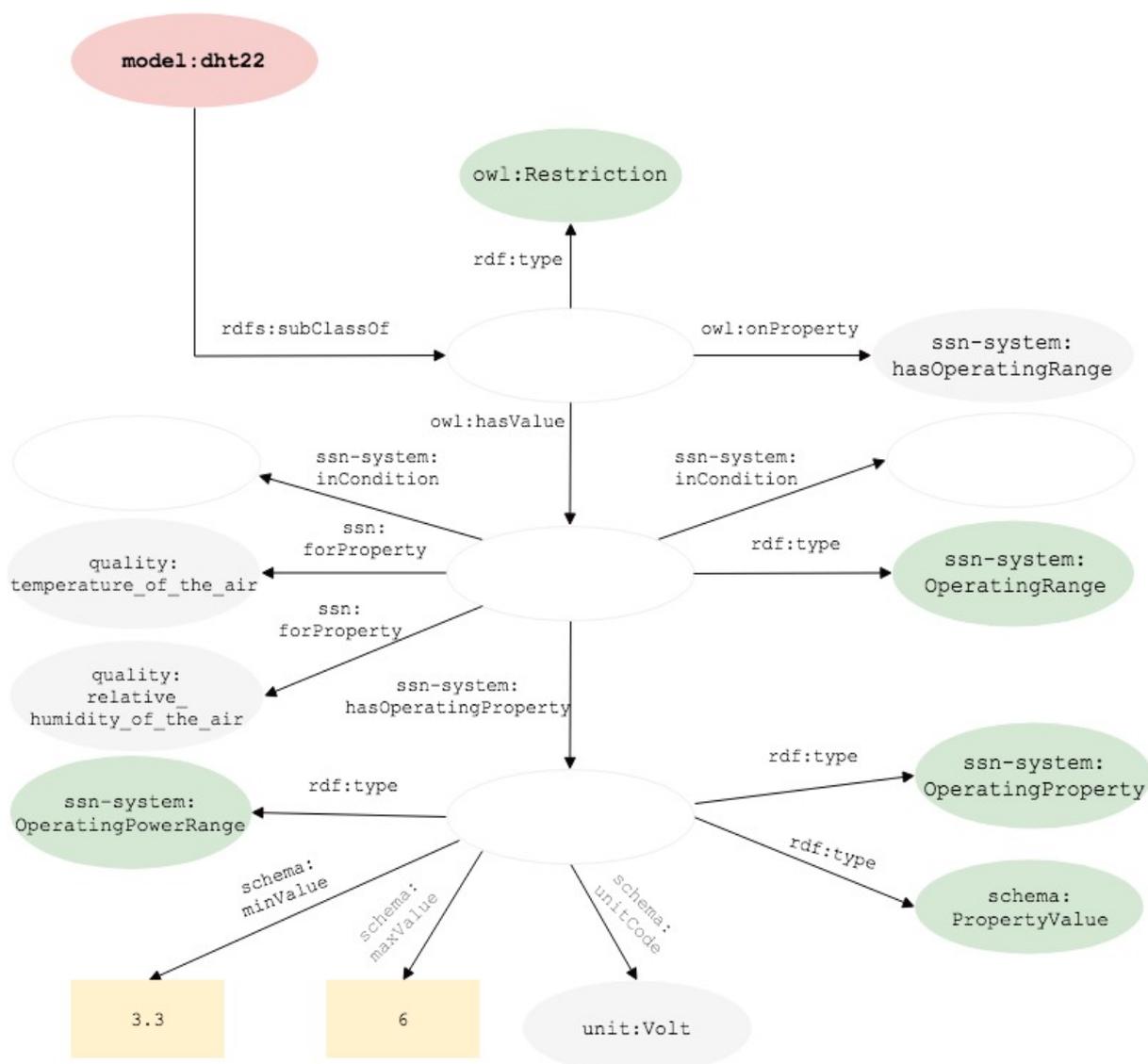
A faixa de operação foi descrita e associada às propriedades de observação “Temperatura do ar” e “Umidade do ar” por meio da propriedade `ssn:forProperty`, vide Figura 97. Além disso, a faixa de operação do modelo DHT22 possui mais de uma condição de funcionamento. Por razões de espaço, a Figura 97 evidencia apenas a condição de temperatura (-40 a 80 °C). A Figura 98 apresenta, portanto, parte do grafo RDF subjacente ao cadastro do modelo com foco na condição de umidade da faixa de operação.



**Figura 98-** Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na faixa de operação e condição normal de umidade

Fonte: Elaboração própria

A Figura 98 evidencia a condição de umidade para que o modelo de sensor DHT22 opere de acordo com o previsto pelo fabricante. Nas condições descritas nas Figuras 97 e 98 o modelo em questão apresenta a propriedade de operação “Faixa de Tensão de alimentação” com valores entre 3,3 e 6V. Tal informação está contemplada na parte do grafo RDF, subjacente ao cadastrado do modelo DHT22, que contém a descrição da propriedade de operação “Faixa de Tensão de alimentação”, vide Figura 99.

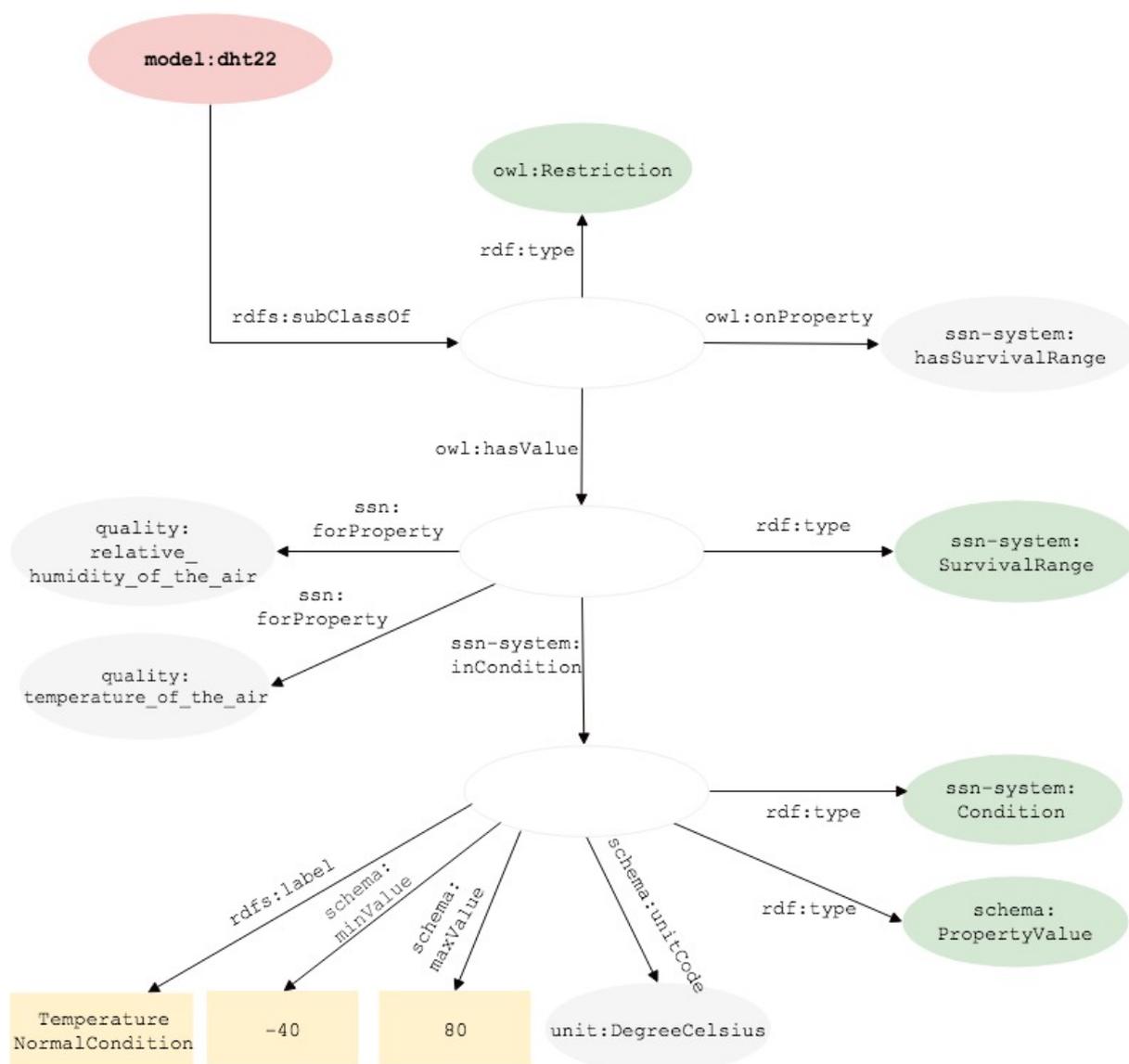


**Figura 99**-Grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de operação “OperatingPowerRange”

Fonte: Elaboração própria

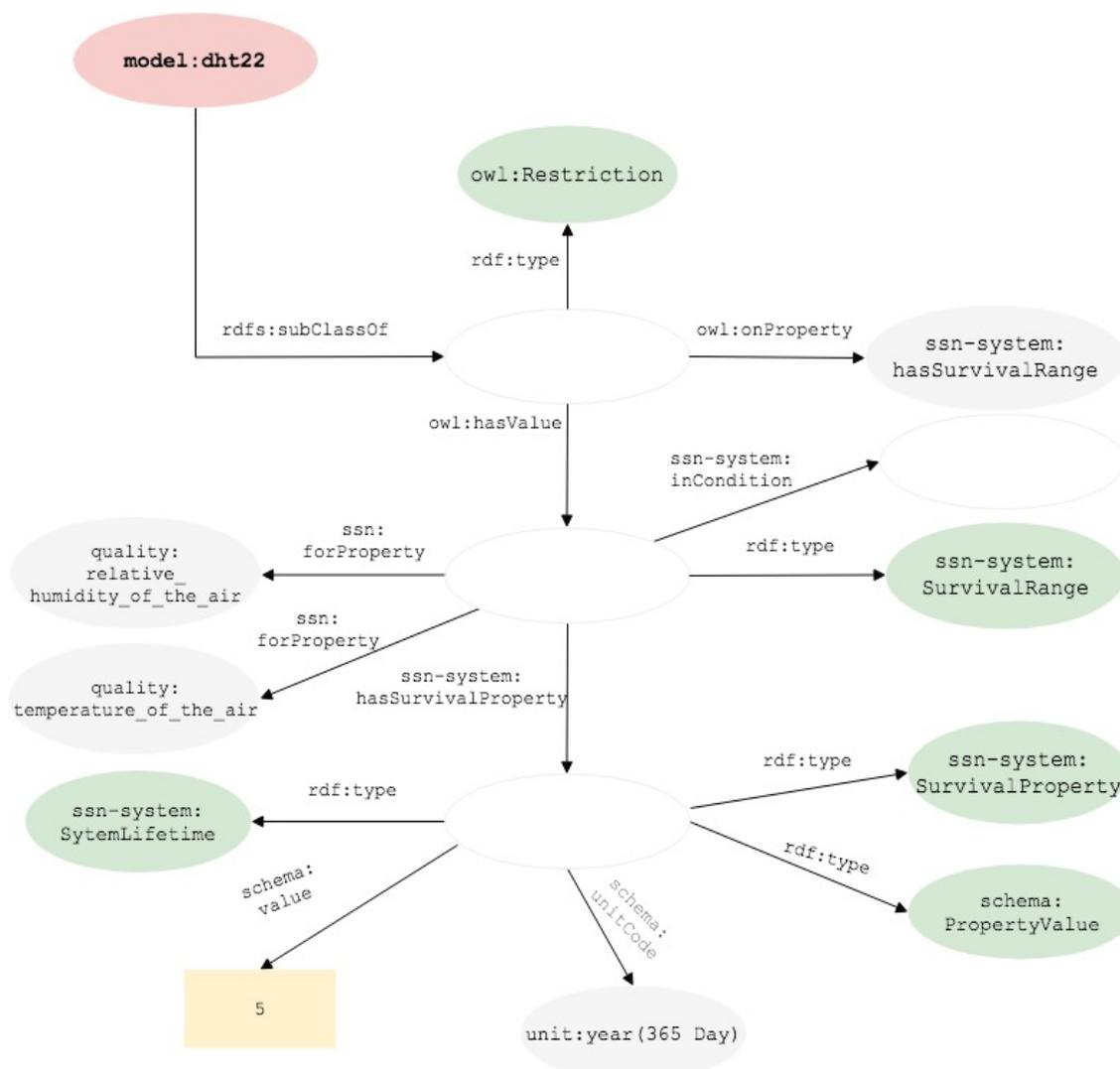
A Figura 99 evidencia os valores da propriedade de operação “Faixa de Tensão de Alimentação” (OperatingPowerRange). Como tal propriedade de operação é aplicável aos processos de observação das propriedades “Temperatura do ar” e “Umidade relativa do ar” foi necessário associar as propriedades observadas, apenas, à descrição da “Faixa de Operação”.

Além das propriedades de sistema e operação, modelos de sensores podem apresentar, ainda, propriedades de sobrevivência. As Figuras 100 e 101 apresentam detalhes sobre a faixa e propriedade de sobrevivência do modelo DHT22.



**Figura 100-** Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na faixa de sobrevivência

A faixa de sobrevivência do modelo DHT22 é descrita em função das propriedades observadas (Temperatura e umidade do ar), condição de funcionamento (-40 a 80 °C) e da propriedade de sobrevivência envolvida (Tempo de vida do sistema). Por questão de simplificação, o grafo RDF da Figura 100 não evidencia a propriedade de sobrevivência que compõe a faixa de sobrevivência do modelo catalogado. Tal propriedade pode ser visualizada no grafo da Figura 101.



**Figura 101-** Parte do grafo RDF subjacente ao cadastro do modelo de sensor DHT22 com foco na propriedade de sobrevivência “System Lifetime”

A Figura 101 evidencia a propriedade de sobrevivência “Tempo de vida do Sistema” (SystemLifetime). Neste grafo, é possível verificar que o modelo de sensor cadastrado tem tempo de vida igual a 5 anos (*unit:year*). A partir desta descrição, torna-se explícito que modelos de sensores DHT22 possuem tempo vida igual a 5 anos, quando submetidos a temperatura entre -40 e 80 °C e enquanto observam as propriedades temperatura e umidade do ar.

#### 6.1.10. Cadastro de Sensor

A catalogação de sensores demanda a existência de diversos outros recursos, tais como pessoas, organizações, bases de dados, propriedades de observação, etc. A Figura 102 ilustra a inserção de um novo exemplar de sensor.

## New Sensor

**ID (URI)**

**Label**

**Comment**

**Create**

**Figura 102-** Tela de criação de exemplar de sensor DHT22  
 Fonte: Elaboração própria

Na Figura 102, o valor “DHT22\_Sensor” inserido no campo ID (URI) irá compor o URI do exemplar cadastrado. Tal dispositivo, cujo modelo é o DHT22, é capaz de observar as propriedades “Temperatura do ar” e “Umidade do ar”. Maiores detalhes sobre o dispositivo são inseridos na tela de edição, vide Figura 103.

## DHT22 Sensor

**Label:**

**Model:**

**Observes (Property):**



**Dataset:**

**Attributed to:**

**Registered by:**

**Comment**

**Postal code:**

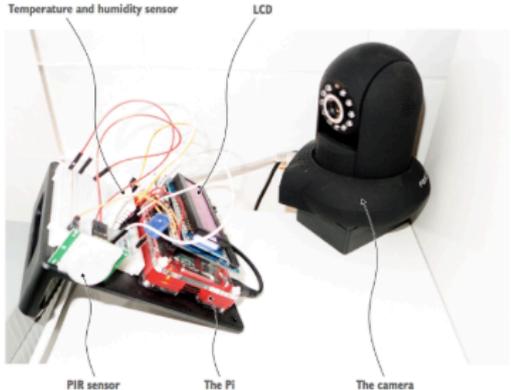
**Latitude:**

**Longitude:**

**City:**

**Country:**

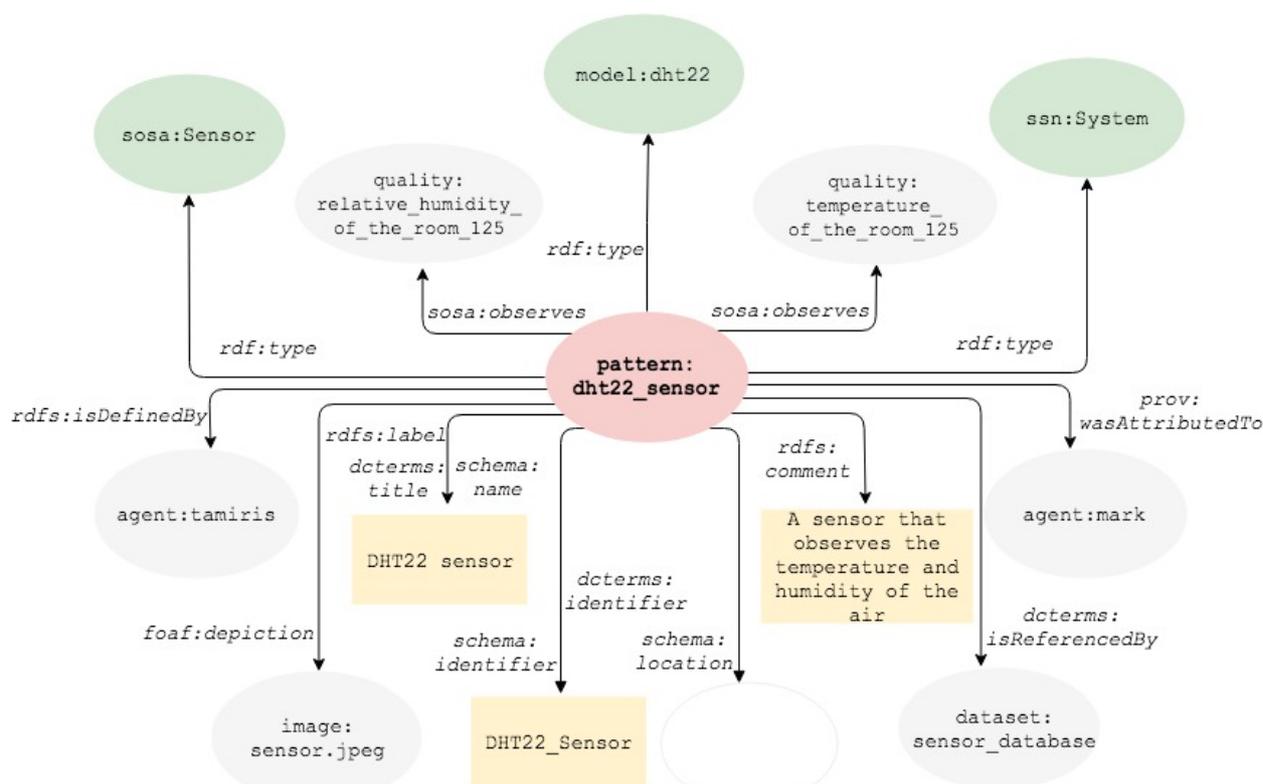
**Select an image of the system**



**Figura 103-** Tela de edição de exemplar de sensor DHT22  
 Fonte: Elaboração própria

Na Figura 103, no campo “Model” o modelo do exemplar catalogado é associado (DHT22). Já no campo “Observes (Property)” as propriedades de observação do exemplar são diferentes daquelas associadas ao modelo deste exemplar. No caso do exemplar apresentado na Figura 103, o objeto de interesse das propriedades observadas associadas é específico, a saber: “Room 125” (Sala 125). Em outras palavras, o exemplar em questão observa a temperatura e umidade de uma parcela específica do ar, ou seja, da sala 125.

Os dados de observação deste sensor podem ser encontrados na base de dados “Sensor\_Database”, indicada no campo “Dataset”. Além disso, tal dispositivo tem como responsável o agente “Mark Zuckerberg”, associado no campo “Attributed to”. O sensor catalogado reside na cidade “Campos dos Goytacazes” e país “Brasil”. Tais recursos têm como origem a base de dados DBpedia e são associados por meio de consultas SPARQL. O registro das informações é realizado pelo agente (pessoa) atribuído no campo “Registered By”. A Figura 104 apresenta o grafo RDF subjacente ao cadastro do sensor DHT22 por meio de um graph pattern instanciado.

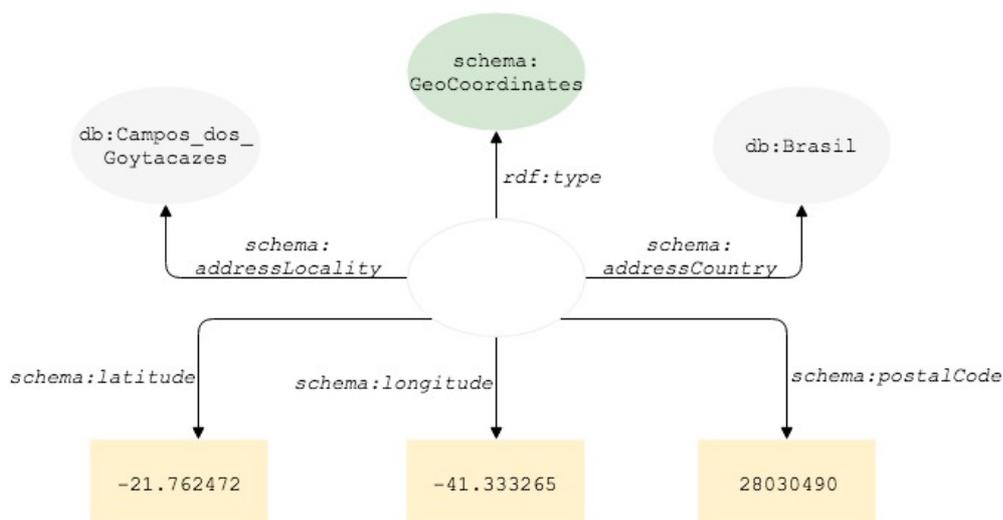


**Figura 104** - Grafo RDF do exemplar do modelo de sensor DHT22

Fonte: Elaboração própria

Na Figura 104, é possível verificar as propriedades das ontologias utilizadas para descrever sensores. A Figura 104 evidencia, ainda, os diferentes recursos utilizados para contextualizar o sensor cadastrado. Por exemplo, as propriedades de observação: “*quality:relative\_humidity\_of\_the\_room\_125*” e “*quality:temperature\_of\_the\_room\_125*” são

alguns destes recursos. Os dados de localização do exemplar catalogado podem ser visualizados na Figura 105.



**Figura 105**-Propriedades de localização de exemplar do modelo DHT22

Fonte: Elaboração própria

Na Figura 105, os recursos “*db:Campos\_dos\_Goytacazes*” e “*db:Brasil*”, que têm como origem a base de dados DBpedia podem ser visualizados. Tais recursos são associados a nó em branco, cuja relação com o exemplar é dada pela propriedade *schema:location*, vide Figura 105. As demais informações associadas ao nó em branco da Figura 105 referem-se a “Latitude”, “Longitude” e “Código postal” do exemplar, cujos valores são literais.

### 6.1.11. Cadastro de Modelo de Atuador

O desenvolvimento das telas de criação e edição de modelos de sensores serviu como base para o desenvolvimento das telas para catalogação de modelos de atuadores. A Figura 106 ilustra a tela de criação e a inserção de um novo modelo de atuador.

A tela de criação de um novo modelo de atuador (New Actuator Model) apresenta os seguintes campos de entrada:

- ID (URI)**: LCD G240641C
- Label**: LCD G240641C
- Comment**: Toshiba CMOS Digital Integrated circuit Silicon Monolithic T6963CFG

Um botão verde "Create" está localizado na base da tela.

**Figura 106**- Tela de criação de um novo modelo de atuador

Fonte: Elaboração própria

O novo modelo de atuador apresentado na Figura 106 consiste em display LCD G240641C. O valor “LCD G240641C” inserido no campo “ID (URI)” irá compor o URI do recurso cadastrado. Mais detalhes sobre o modelo de atuador “LCD G240641C” podem ser vistos na Figura 107.

The image shows a web form titled "Actuator Model" with several sections:

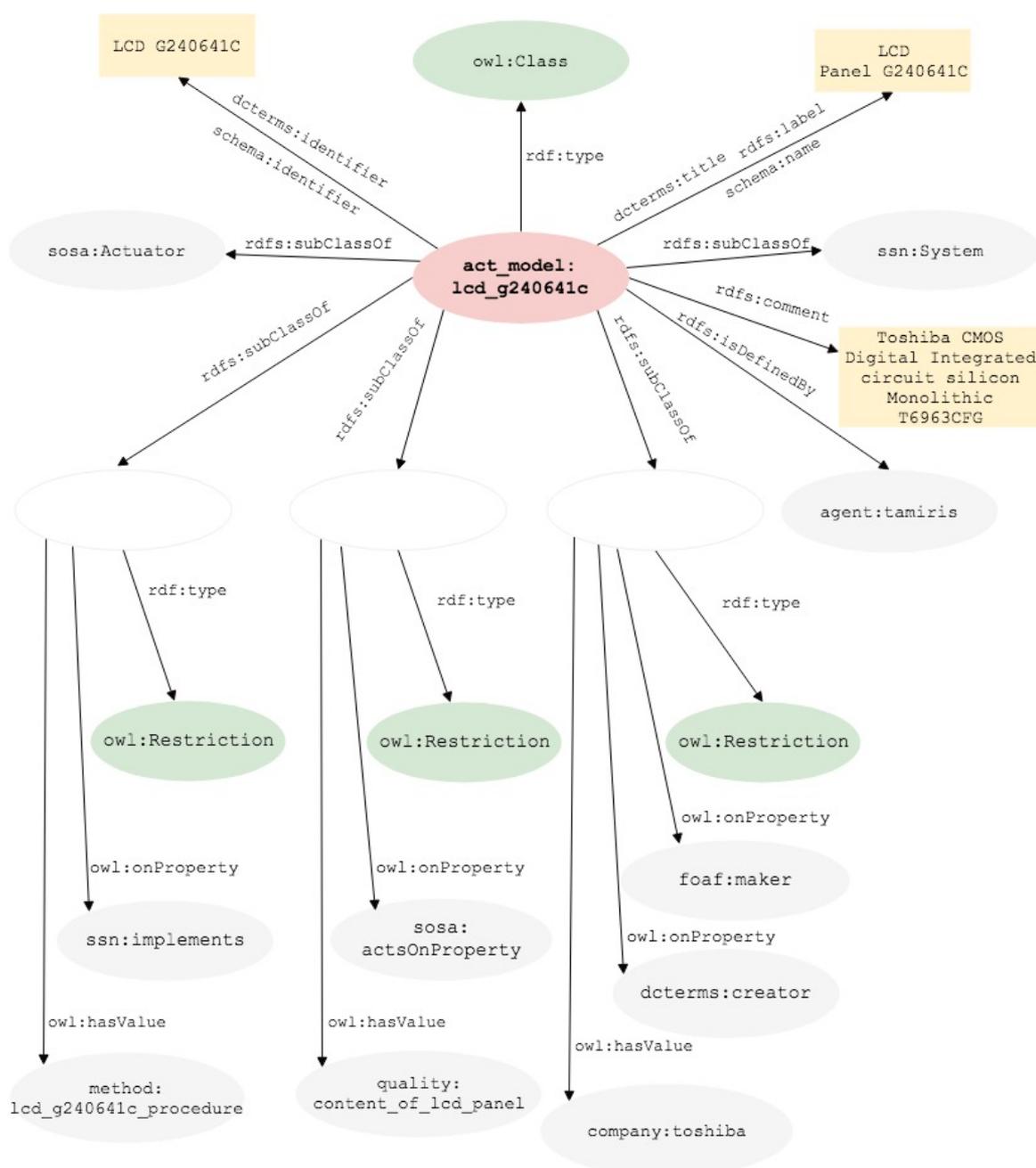
- Label:** A text input field containing "LCD G240641C".
- Acts on Property:** A list box containing several options: "Content\_of\_LCD\_panel" (which is selected), "Humidity of WOT deployment room", "Image", "Image of WOT deployment room", and "Infrared signal".
- Procedure:** A list box containing several options: "IP68\_Outdoor\_Temperature\_Sensor#Pi", "IP68\_Outdoor\_Temperature\_Sensor#b", "IP68\_Outdoor\_Temperature\_Sensor#te", and "LCD\_G240641C\_Procedure" (which is selected).
- Producer/creator:** A list box containing several options: "Organization/1", "Sparkfun Eletronics", "Toshiba" (which is selected), and "teste".
- Comment:** A text area containing the text "Toshiba CMOS Digital Integrated circuit Silicon Monolithic T6963CFG".
- Registered by:** A text input field containing "Tamiris Sousa".

**Figura 107**-Parte da tela de edição de modelo de atuador

Fonte: Elaboração própria

A Figura 107 evidencia a tela de edição do modelo de atuador catalogado. No campo “Acts on Property”, propriedades de atuação são selecionadas. É possível perceber que este campo do formulário retorna todas as “Propriedades de Objeto de Interesse” (“Property”) pré-existentes na base de dados da aplicação. Ao selecionar a propriedade desejada, o sistema irá inferir se a propriedade é de atuação ou observação. Assim, a propriedade “Content\_of\_lcd\_panel” (Informação exibida na tela do display) selecionada (Figura 107) será inferida como sendo uma propriedade de atuação do modelo de atuador “LCD G240641C”.

Além disso, a tela de edição apresenta, ainda, outros recursos associados ao modelo de atuador, tais como: o procedimento executado pelo atuador (“LCD\_G240641C\_Procedure”), o fabricante (“Toshiba”) e a pessoa responsável pelo cadastro das informações (“Tamiris Sousa”). A Figura 107 representa apenas parte da tela de edição de modelos de atuadores. As demais partes, onde as propriedades de sistema, operação e sobrevivência podem ser definidas, não serão apresentadas. Por serem idênticas àquelas apresentadas para descrição de sensores, podem ser visualizadas nas Figuras 85-91. O grafo RDF subjacente ao cadastro do modelo de atuador “LCD G240641C” é apresentado na Figura 108, por meio de um *graph pattern* instanciado.

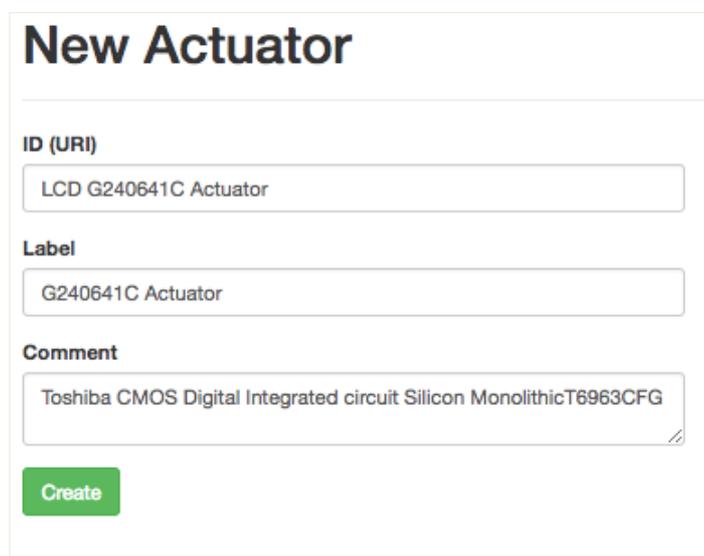


**Figura 108**-Grafo RDF subjacente ao cadastro de modelo de atuador  
Fonte: Elaboração própria

Na Figura 108, é possível verificar a única propriedade que diferencia o cadastro de modelos de sensores do cadastro de modelos de atuadores: a propriedade *sosa:actsOnProperty*. Tal propriedade é utilizada para associar a propriedade de objeto de interesse “Content\_of\_lcd\_panel” ao modelo de atuador catalogado. Nesse caso, trata-se de propriedade de objeto de interesse genérica, tendo em vista que o objeto de interesse não é um atuador específico e sim o próprio modelo “LCD\_G240641C” catalogado. Além disso, como as propriedades de sistema, operação e sobrevivência do modelo de atuador catalogado não foram inseridas, o grafo RDF da Figura 108 apresenta, apenas, as restrições para definição do valor do range nas propriedades relacionadas aos campos “Acts On Property” “Procedure” e “Producer/creator” da tela de edição (Figura 107). Como resultado, os exemplares do modelo catalogado serão atrelados aos recursos associados nos campos em questão por inferência,

#### 6.1.12. Cadastro de Atuador

O modelo para catalogação de atuadores apresentado neste trabalho consiste em uma derivação do modelo para descrição de sensores. Assim, as telas para criação e edição de atuadores e sensores são muito similares. A Figura 109 ilustra a criação de um novo atuador.



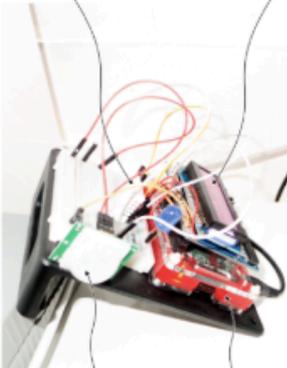
The image shows a web form titled "New Actuator". It contains three text input fields and a "Create" button. The first field, labeled "ID (URI)", contains the text "LCD G240641C Actuator". The second field, labeled "Label", contains "G240641C Actuator". The third field, labeled "Comment", contains "Toshiba CMOS Digital Integrated circuit Silicon MonolithicT6963CFG". The "Create" button is green and located at the bottom left of the form area.

**Figura 109**-Tela de criação de um novo atuador

Fonte: Elaboração própria

O novo atuador apresentado na Figura 109 consiste em display LCD, modelo “LCD G240641C”. O valor “LCD G240641C Actuator” inserido no campo “ID (URI)” irá compor o URI do recurso cadastrado. Mais detalhes sobre o atuador “LCD G240641C Actuator” podem ser vistos na Figura 110.

## G240641C Actuator

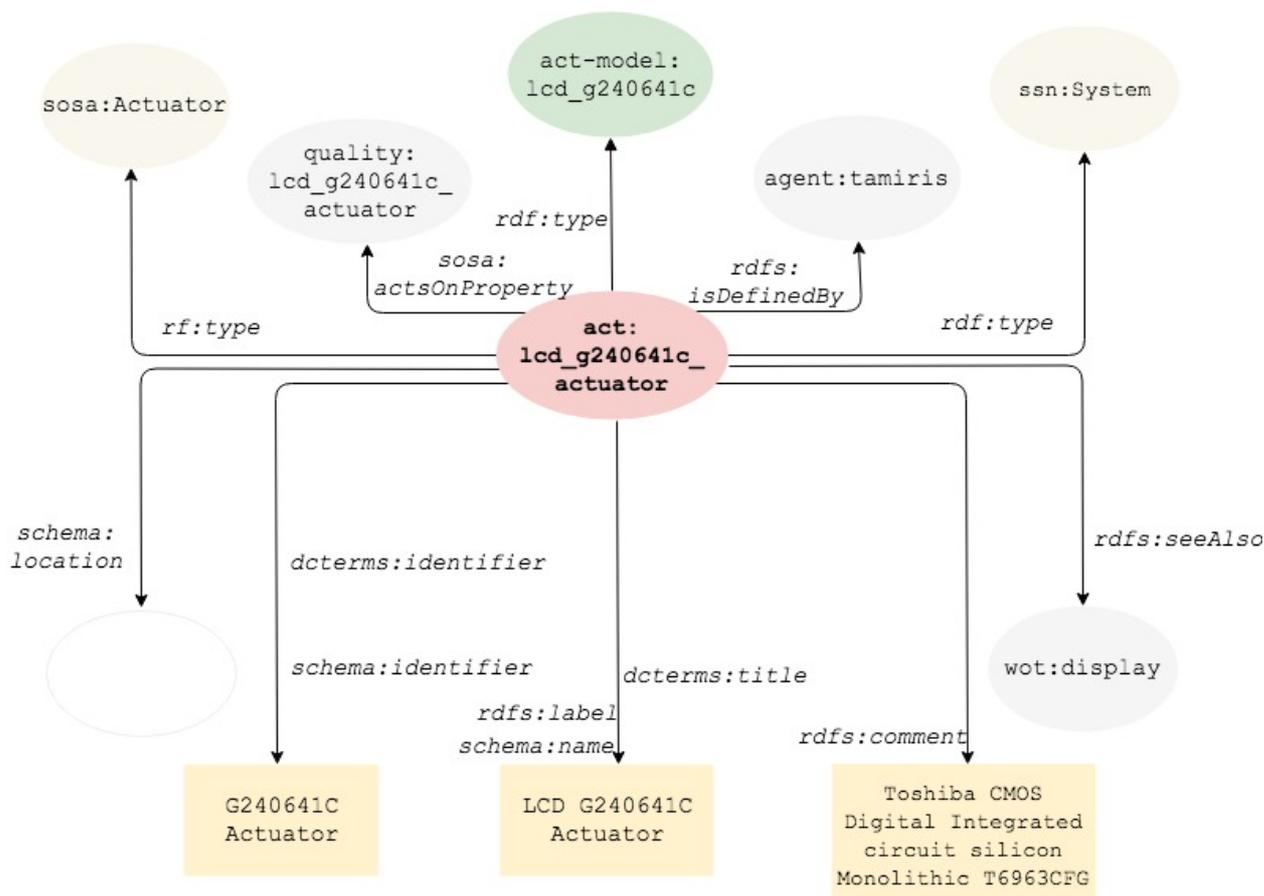
<p><b>Label:</b>  <input type="text" value="G240641C Actuator"/></p> <p><b>Acts on Property:</b>  <input type="text" value="Content of LCD G240641C Actuator"/></p> <p><b>Model:</b>  <input type="text" value="LCD G240641C"/></p> <p><b>See Also:</b>  <input type="text" value="http://devices.webofthings.io/pi/actuators/display/"/></p>	<p><b>Postal code:</b>  <input type="text" value="28030490"/></p> <p><b>Latitude:</b>  <input type="text" value="-21.762472"/></p> <p><b>Longitude:</b>  <input type="text" value="-41.333265"/></p> <p><b>City:</b>  <input type="text" value="Campos dos Goytacazes"/></p> <p><b>Country:</b>  <input type="text" value="Brasil"/></p> <p><b>Comment</b>  <input type="text" value="Toshiba CMOS Digital Integrated circuit Silicon MonolithicT6963CFG"/></p> <p><b>Registered by:</b>  <input type="text" value="Tamiris Sousa"/></p> <p><b>Select an image of the system</b> </p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>Temperature and humidity sensor</p>  <p>PIR sensor</p> </div> <div style="text-align: center;"> <p>LCD</p>  <p>The Pi</p> </div> <div style="text-align: center;"> <p>The camera</p>  </div> </div> <p style="text-align: center;">✕</p> <p style="text-align: center;"> <input type="button" value="Save"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/> </p>
---	---

**Figura 110-** Tela de edição de atuador

Fonte: Elaboração própria

A Figura 110 apresenta a tela de edição do exemplar “LCD G240641C Actuator”. Diferente da tela de edição de sensores, a tela de edição de modelos de atuadores possui o campo “See Also”. Neste campo, o usuário pode inserir recursos que complementem as informações sobre o dispositivo catalogado. Por exemplo, na Figura 110, um *link* para acessar o atuador em tempo real é associado ao campo “See Also”. Além disso, no caso do campo “Acts on Property”, a propriedade de objeto de interesse “Content of LCD G240641C Actuator” associada é específica, tendo em vista que o objeto de interesse desta propriedade é o próprio atuador “LCD G240641C Actuator”. Em outras palavras, o processo de atuação consiste na

modificação da informação exibida no próprio exemplar catalogado. As propriedades envolvidas na descrição do atuador em questão podem ser visualizadas no grafo RDF da Figura 111.

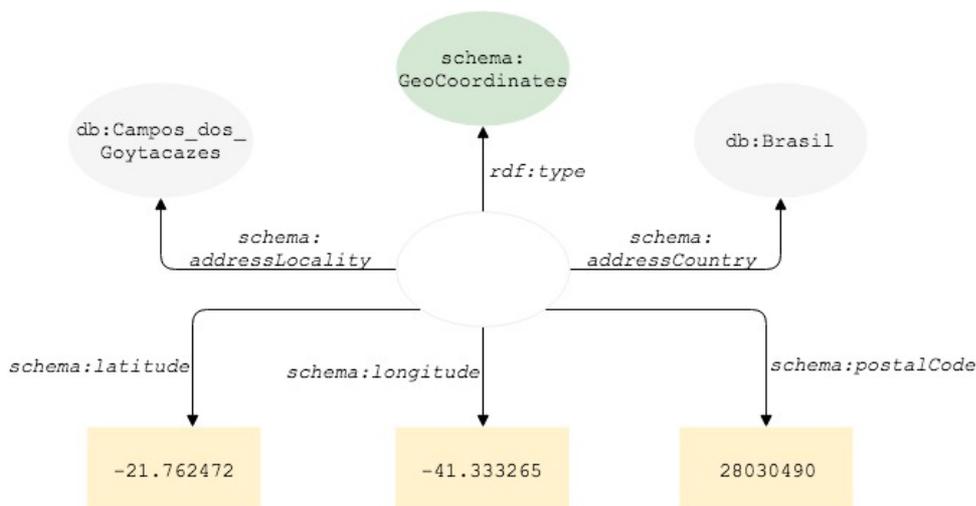


**Figura 111**-Grafo RDF subjacente ao cadastro de atuador

Fonte: Elaboração própria

Na Figura 111, as classes nas quais o atuador catalogado pertence são evidenciadas. Por exemplo, além de ser uma instância de *sosa:Actuator* e *ssn:System*, o atuador “LCD\_G240641C\_Actuator” consiste, ainda, em uma instância da classe “LCD\_G240641C”. Por ser uma instância do modelo “LCD\_G240641C”, tal exemplar herda todas as características associadas a este modelo, tais como informações sobre procedimentos, fabricante e propriedades de objeto de interesse.

No caso das propriedades de localização de atuadores, o modelo adotado para descrever tais informações é o mesmo adotado para sensores. Além disso, neste exemplo em particular, os valores das propriedades de localização envolvidas no cadastro do atuador catalogado são os mesmos utilizados no cadastro do sensor “DHT22\_Sensor”, tendo em vista que tais dispositivos fazem parte da mesma aplicação Web das Coisas apresentada na Figura 53. A Figura 112 apresenta parte do grafo RDF subjacente ao cadastro de atuador “LCD\_G240641C\_Actuator” com foco nas propriedades de localização.



**Figura 112-** Grafo RDF subjacente ao cadastro de atuador com foco nas propriedades de localização

Fonte: Elaboração própria

Na Figura 112, os recursos *db:Brasil* e *db:Campos\_dos\_Goytacazes* são resultados de consultas SPARQL que filtram recursos definidos como *dbo:Location* - *dbo* é o prefixo da ontologia da base de dados DBpedia e *dbo:Location* é uma classe desta ontologia para diferentes localizações do mundo (países e cidades).

### 6.1.13. Cadastro de Plataforma (Platform)

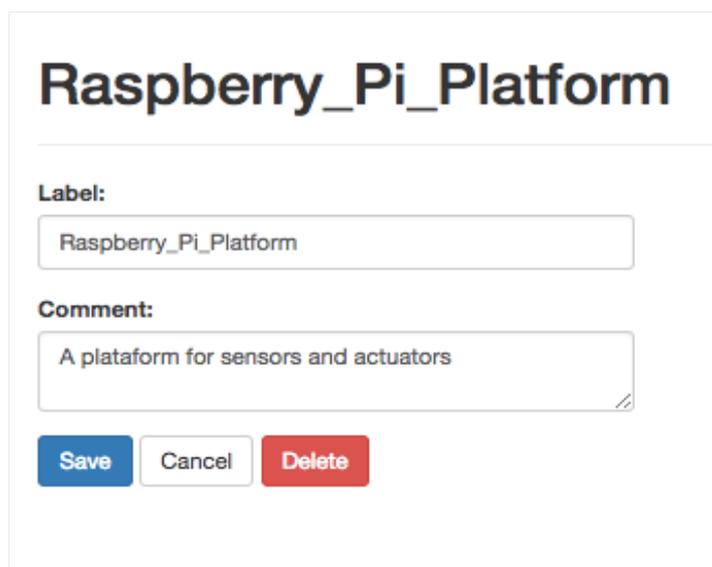
O modelo ontológico para descrição de plataformas consiste em modelo simples que contém, basicamente, identificador, nome e comentário. A Figura 113 ilustra o cadastro da plataforma “Raspberry Pi” que hospeda sensores e atuadores na aplicação WoT proposta por Guinard & Trifa (2016).

The screenshot shows a web form titled "New Platform". It has three input fields: "ID (URI):" with the value "Raspberry\_Pi", "Label:" with the value "Raspberry\_Pi\_Platform", and "Comment:" with the value "A plataform for sensors and actuators". Below the fields is a green "Create" button.

**Figura 113-** Tela de criação de plataforma

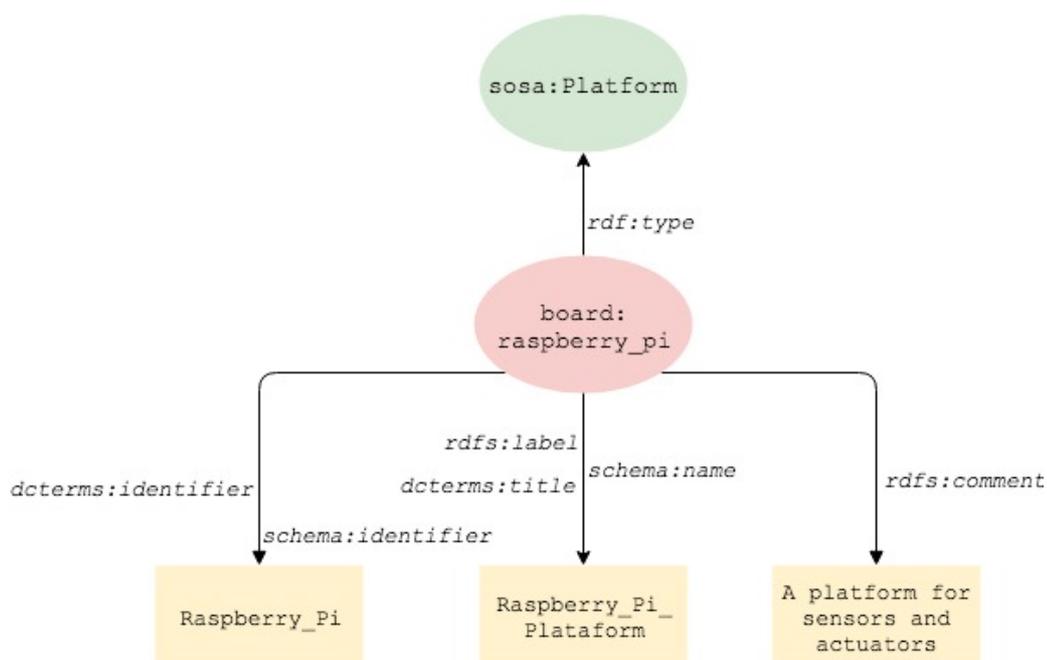
Fonte: Elaboração própria

Assim como no cadastro dos demais recursos no catálogo, o URI do recurso é definido no campo “ID(URI)”. Assim, o valor “Rasberry\_Pi” será utilizado pelo sistema para criar o URI do novo recurso (Figura 113). As informações inseridas nos demais campos da tela de criação podem ser modificadas na tela de edição, vide Figura 114.



**Figura 114-** Tela de edição de plataforma  
Fonte: Elaboração própria

A Figura 114 apresenta os campos disponíveis para edição. Nesse caso, é possível editar o nome e o comentário inserido no momento de criação do novo recurso. O grafo RDF subjacente ao cadastro da plataforma em questão é apresentado na Figura 115.

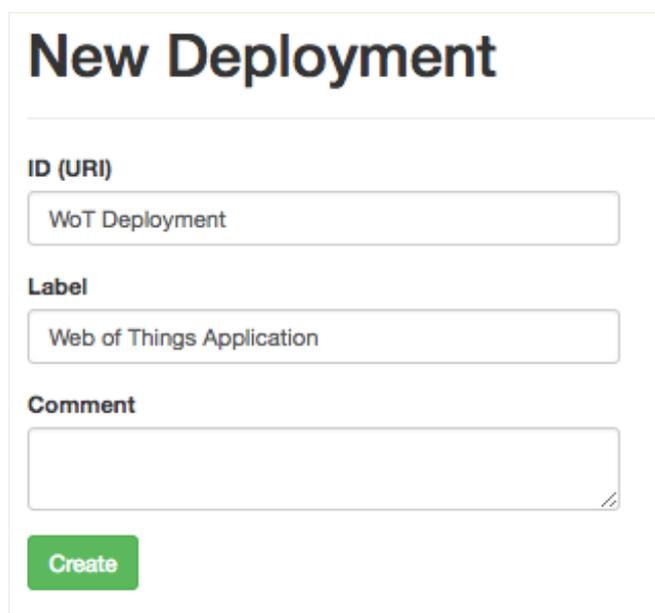


**Figura 115-** Grafo RDF subjacente ao cadastro da Plataforma “Raspberry Pi”  
Fonte: Elaboração própria

A Figura 115 evidencia os valores inseridos nos campos das telas de criação e edição da plataforma “Raspebrry Pi” (Figuras 113 e 114) e as propriedades utilizadas na descrição do recurso. Por exemplo, com o uso da propriedade *rdf:type*, a nova plataforma cadastrada “Raspberry Pi” é definida como sendo uma instância da classe *sosa:Plataform* da ontologia SSN.

#### 6.1.14. Cadastro de Implantação

A catalogação de implantações demanda a existência de diversos recursos, tais como sensores, atuadores, plataformas, pessoas, organizações, etc. Para ilustrar a catalogação de implantações, a aplicação WoT apresentada em Guinard & Triffa (2016) será catalogada de forma realística. Os dispositivos físicos envolvidos nesta aplicação WoT podem ser visualizados na Figura 53 e Tabela 21. A Figura 116 ilustra a criação da aplicação referida no catálogo semântico proposto.



The image shows a web form titled "New Deployment". It contains three input fields: "ID (URI)" with the text "WoT Deployment", "Label" with the text "Web of Things Application", and "Comment" which is empty. Below the fields is a green button labeled "Create".

**Figura 116-** Tela de criação de implantação WoT

Fonte: Elaboração própria

A Figura 116 evidencia a criação da nova implantação no catálogo. O valor “Wot Deployment”, inserido no campo ID (URI), irá compor o URI deste novo recurso. Por se tratar de uma aplicação WoT, é possível agregar sensores, atuadores, plataformas e outros recursos ao cadastro de implantações na tela de edição, vide Figura 117.

## Web of Things Application

**Label:**

**Deployed on Platform:**

**Deployed System:**

**Associated with:**

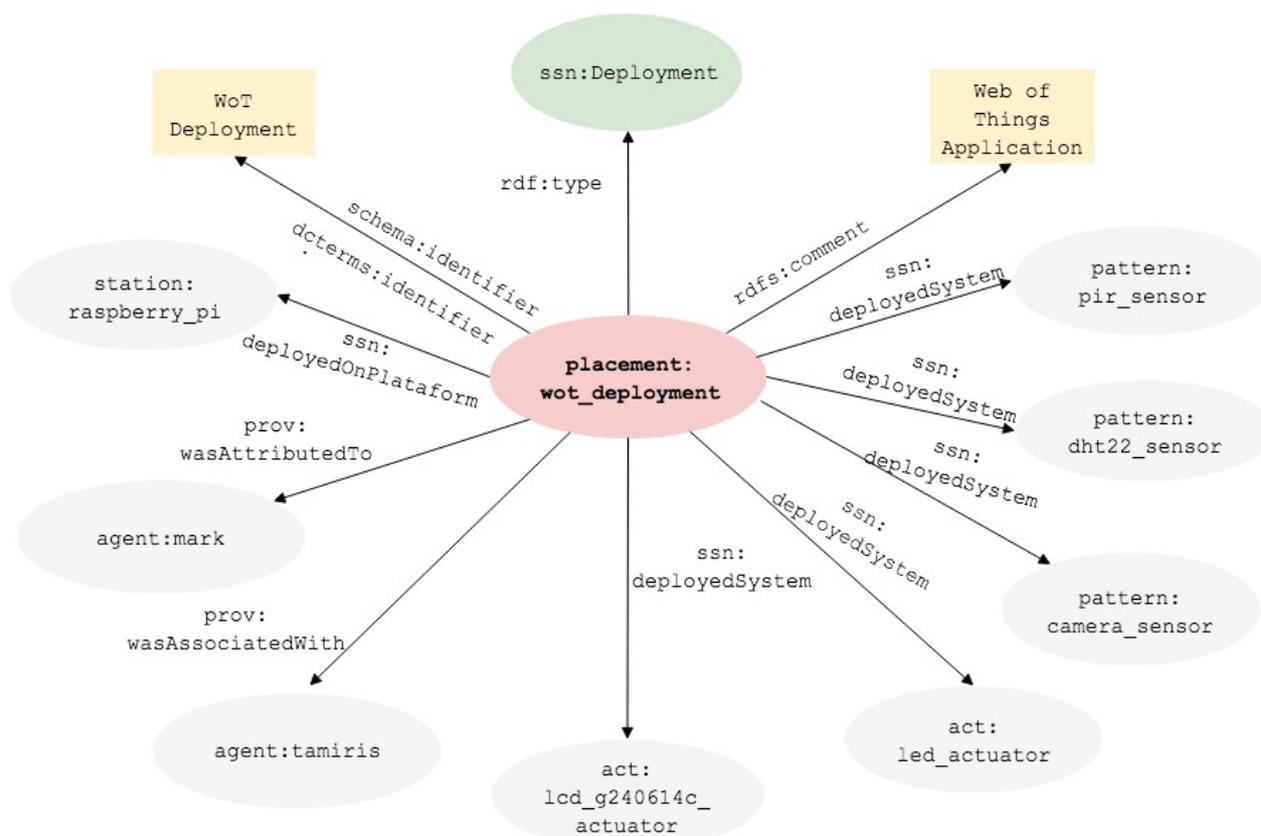
**Attributed to:**

**Comment**

**Figura 117-** Tela de edição de implantação WoT

Fonte: Elaboração própria

A implantação WoT catalogada é dotada de diferentes sensores e atuadores. Tais objetos são selecionados no campo “Deployed System”. Por inferência, todos estes objetos são vinculados à plataforma evidenciada no campo “Deployed on Platform” (Raspberry Pi). No caso do campo “Associated With” e “Attributed to” agentes do tipo pessoa ou organização podem ser associados por meio de consultas SPARQL. Ao associar uma pessoa ao campo “Associated With”, torna explícito que esta pessoa possui algum relacionamento (não definido especificamente) com a implantação. Já o campo “Attributed to” faz referência ao principal responsável pela implantação ou aquele que responde por esta implantação. No exemplo ilustrado na Figura 117, a pessoa associada neste campo, pode ser o idealizador, o técnico responsável ou ainda o gestor desta implantação. O grafo RDF subjacente ao cadastro da implantação em questão é apresentado na Figura 118.



**Figura 118-** Grafo RDF subjacente ao cadastro de Implantação WoT  
 Fonte: Elaboração própria

Na Figura 118, os dispositivos físicos apresentados na Figura 53 e listados na Tabela 21 são evidenciados. Como a implantação possui mais de uma instância da classes *sosa:Sensor* e *sosa:Actuator*, ou seja, é dotada de mais de um sensor e atuador, diferentes triplas são geradas com a mesma propriedade. Por exemplo, a propriedade *ssn:deployedSystem* é utilizada para associar tanto sensores quanto atuadores a implantações. Assim, na Figura 118, o uso de tal propriedade se repete a cada novo sensor ou atuador vinculado.

## 7. EXEMPLOS DE USO DO CATÁLOGO

Este capítulo tem como objetivo apresentar exemplos realistas de uso do catálogo para corroborar a eficácia da versão atual aplicação. Tais exemplos envolvem a descrição de sensores e recursos relacionados. Por questão de simplificação, apenas os grafos RDF subjacentes aos cadastros dos recursos serão apresentados por meio de *graph pattern* instanciados. Para facilitar a demonstração dos grafos RDF resultantes dos exemplos realistas, prefixos são adotados para simplificar os URIs dos recursos cadastrados na aplicação (e de terceiros), de acordo com a Tabela 22. A versão em Turtle destes grafos pode ser encontrada no apêndice deste trabalho.

### A. IP68 Smart Sensor

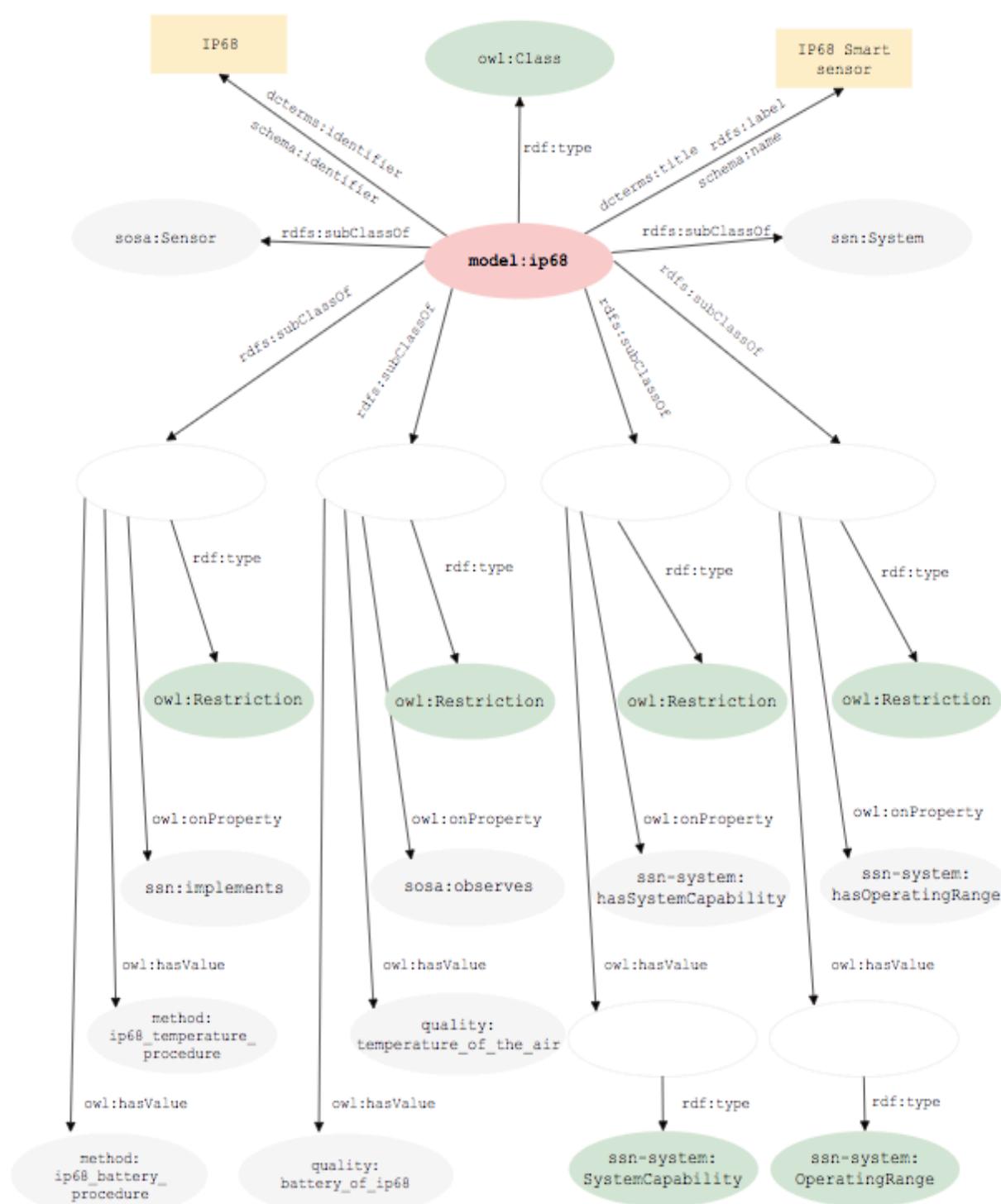
O sensor IP68 Smart Sensor é capaz de observar propriedades distintas, tais como: temperatura do ar e estado de sua bateria. A descrição deste dispositivo no catálogo proposto é inspirada nos exemplos de uso da ontologia SSN apresentados em Haller *et al.* (2017). A Tabela 23 apresenta os dados utilizados para descrever o sensor IP68 Smart Sensor.

**Tabela 23** - Dados utilizados na descrição do sensor IP68 Smart Sensor

Item	Especificação
Modelo de Sensor	IP68
Propriedades observadas	Temperatura e estado da bateria
Capacidade de sistema	Dada em condição normal de temperatura (-20 a 70°C)
Acurácia (sensor de temperatura)	+/- 0.2°C
Sensibilidade	-137 dBmW
Resolução (sensor de temperatura )	0.065°C
Resolução (sensor de estado de bateria)	$3.937e^{-3}\%$
Objetos de interesse	Ar e sensor IP68

Fonte: Elaboração própria

Além dos dados indicados na Tabela 23, o sensor cadastrado também é associado a uma organização, implantação, procedimento e modelo. No catálogo proposto, é possível descrever o modelo do sensor IP68 associando seus atributos técnicos. A partir desta prática, ao descrever o sensor (IP68 Smart Sensor) e vinculá-lo ao seu modelo, tem-se como resultado o aproveitamento de todas as informações do modelo em questão. A Figura 119 parte do grafo RDF subjacente cadastro do modelo do sensor IP68 no catálogo semântico.

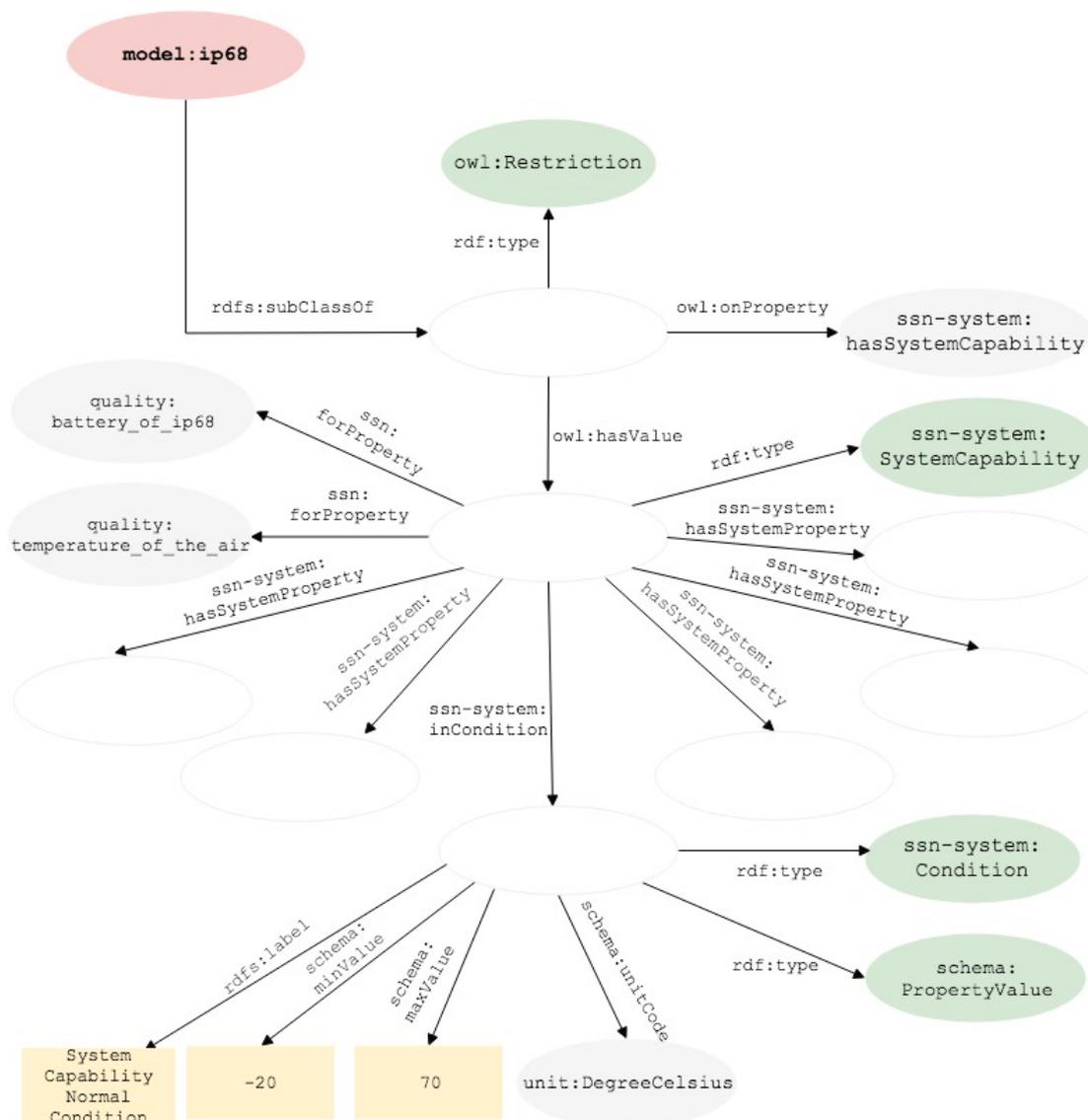


**Figura 119** - Parte do grafo RDF subjacente ao cadastro do modelo de sensor IP68

Fonte: Elaboração própria

A Figura 119 evidencia os diferentes recursos associados ao modelo catalogado, de acordo com a Tabela 23. Por exemplo, é possível verificar as propriedades de observação associadas ao modelo IP68, na propriedade `sosa:observes`. Optou-se por vincular apenas propriedades genéricas ao modelo de sensor, ou seja, que são dotadas de objetos de interesse genéricos, são elas: Temperatura do ar (“Temperatura of the air”) e “Estado da Bateria do

modelo IP68” (“Battery of IP68”). Além disso, é possível verificar mais de um procedimento para o modelo de sensor catalogado, um para cada propriedade observada (*method:ip68\_battery\_procedure* e *method:ip68\_temperature\_procedure*). Por questão de espaçamento, as capacidades de sistema e faixa de operação não são apresentadas nesta parte do grafo da Figura 119. A Figura 120 apresenta maiores detalhes sobre as capacidades de sistema do modelo IP68.

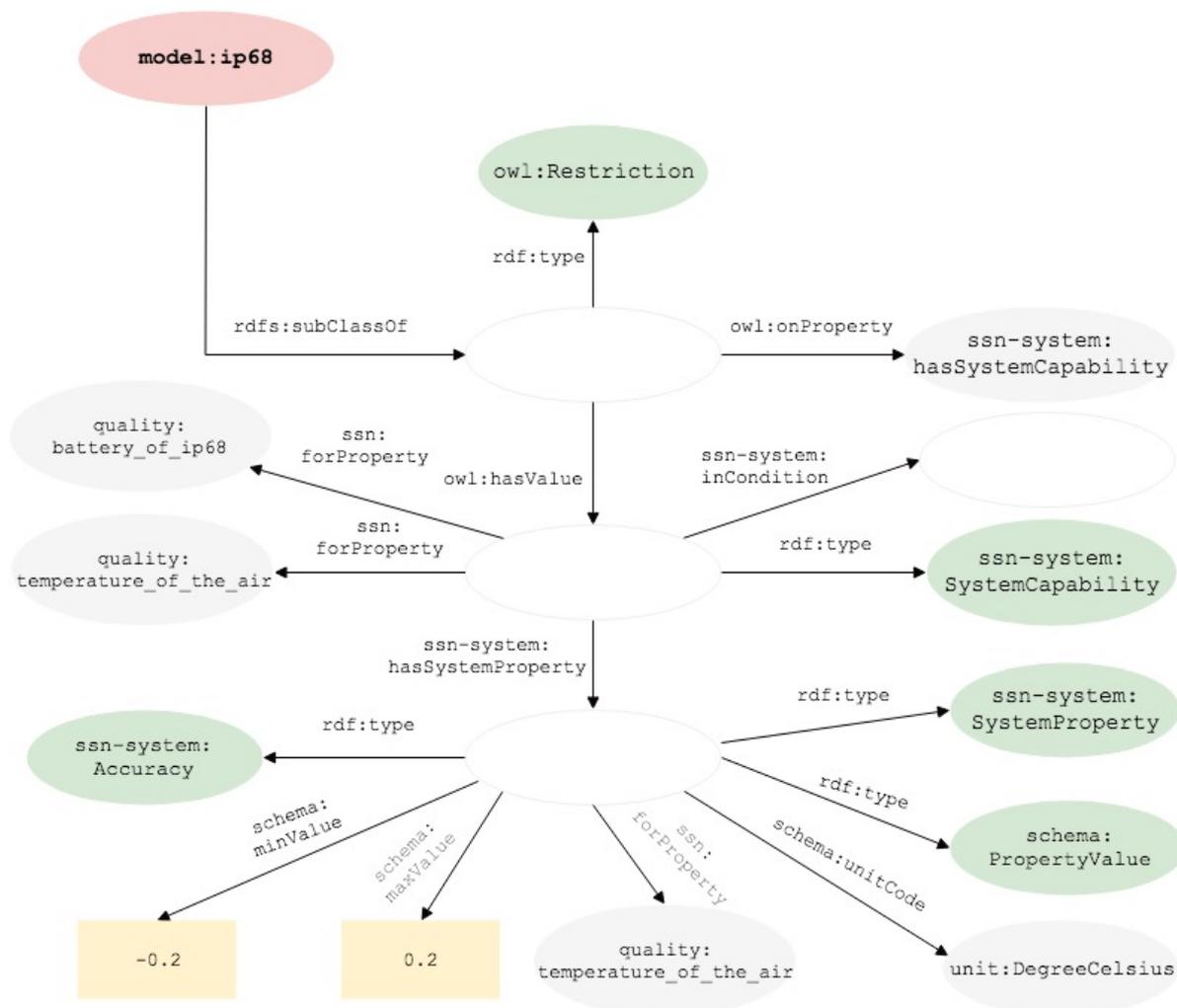


**Figura 120** - Parte da descrição da capacidade de sistema do modelo IP68

Fonte: Elaboração própria

A Figura 120 evidencia as condições da capacidade de sistema e as propriedades de objeto de interesse envolvidas, a saber: “Bateria do modelo IP68” (*quality:battery\_of\_ip68*) e “Temperatura do ar” (*quality:temperature\_of\_the\_air*). Além das informações apresentadas na Figura 120, a capacidade de sistema do modelo IP68 envolve diferentes propriedades de

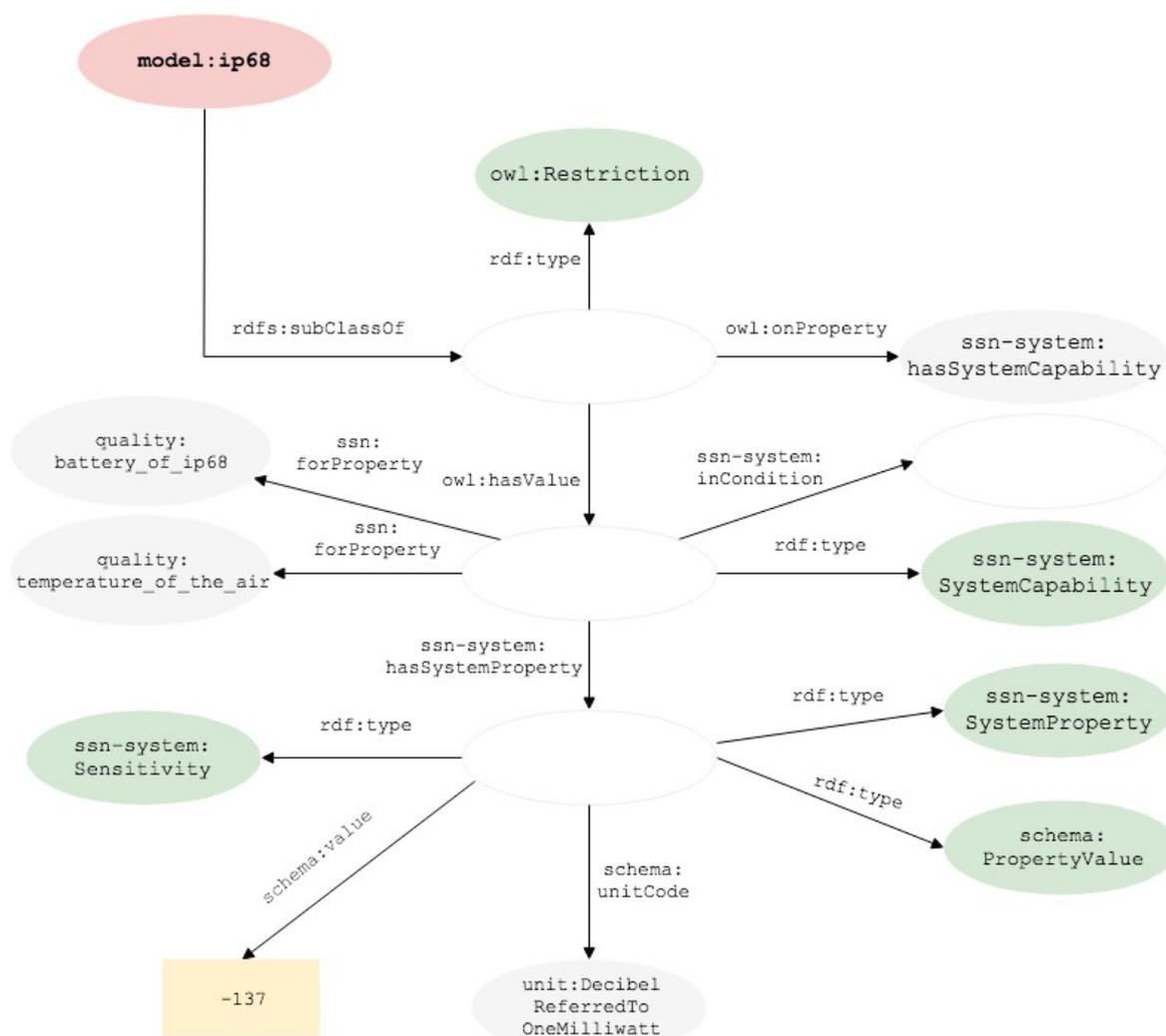
sistemas, a exemplo da propriedade “Acurácia” (Accuracy), cuja descrição é apresentada na Figura 121.



**Figura 121** - Parte do grafo RDF da capacidade de sistema do modelo IP68 com foco na propriedade de sistema “Accuracy”

Fonte: Elaboração própria

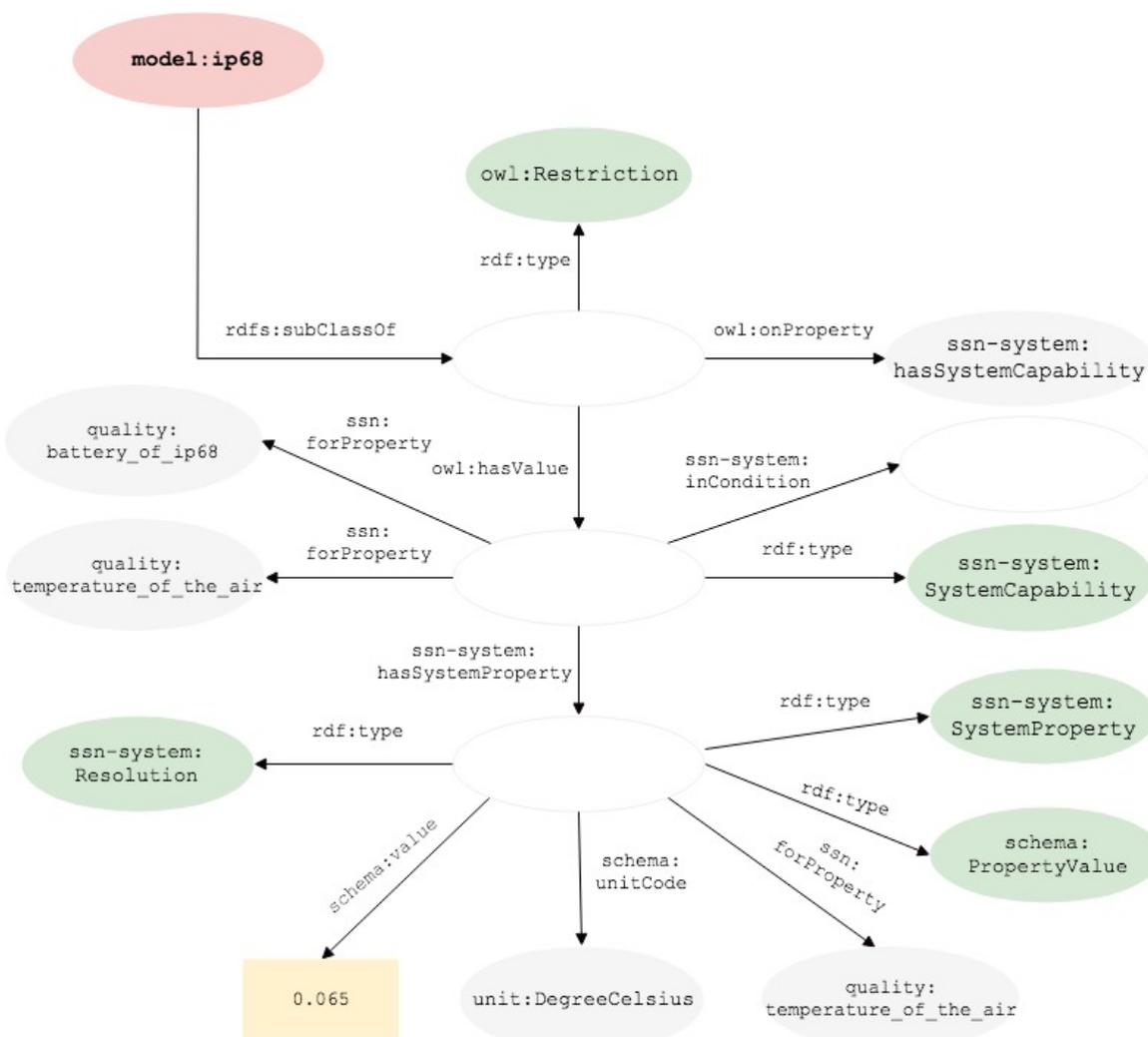
A Figura 121 evidencia parte da descrição da capacidade de sistema com foco na propriedade de sistema “Accuracy” (acurácia). Por razões de espaço, as condições das capacidades de sistema não foram apresentadas nesta parte do grafo RDF e podem ser visualizadas na Figura 120. A propriedade “acurácia” refere-se ao processo de observação da propriedade “Temperatura do ar” (*quality:temperature\_of\_the\_air*). Tal informação pôde ser explicitada por meio da propriedade *ssn:forProperty* e torna-se necessária, uma vez que o modelo de sensor catalogado observa mais de uma propriedade. Além da propriedade acurácia, a descrição do sensor inclui também a propriedade de sistema “Sensitivity” (Sensibilidade). A Figura 122 apresenta a descrição da capacidade de sistema com foco na propriedade de sistema “Sensitivity”.



**Figura 122** - Parte do grafo RDF da capacidade de sistema como foco na propriedade “Sensitivity”

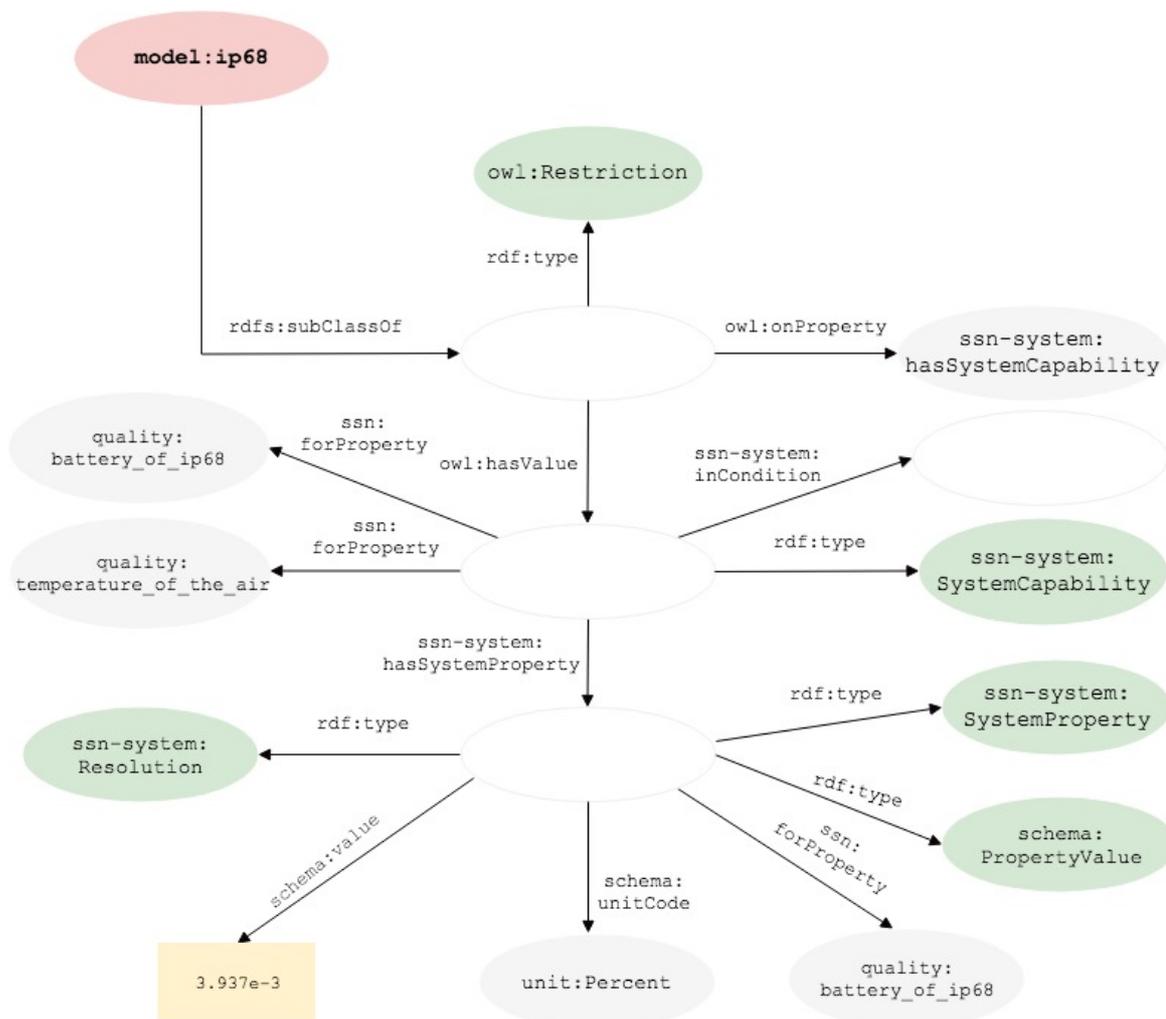
Fonte: Elaboração própria

Modelos de sensores IP68 possuem sensibilidade (“Sensitivity”) no valor de -137 dBmW, vide Tabela 23. Como esta propriedade é aplicada tanto ao processo de observação da temperatura do ar quanto da bateria do sensor, optou-se por vincular estas propriedades de observação, apenas à descrição da capacidade de sistema, fazendo uso da propriedade *ssn:forProperty*. Além da sensibilidade este modelo de sensor apresenta, ainda, a propriedade de sistema resolução (“Resolution”). A Figura 123 apresenta maiores detalhes sobre a resolução do modelo IP68.



**Figura 123** - Parte do grafo RDF da capacidade de sistema como foco na propriedade “Resolution” para a propriedade “Temperatura do ar”  
Fonte: Elaboração própria

Na Figura 123, diferente da propriedade “Sensitivity” (Sensibilidade), a propriedade “Resolution” (Resolução) é descrita em termos de valor e propriedade de objeto de interesse observada. Como o modelo de sensor catalogado observa propriedades distintas e apresenta resoluções diferentes para cada uma destas propriedades, o uso da propriedade *ssn:forProperty* torna-se crucial. Assim, os valores de resolução apresentados na Figura 123, referem-se ao processo de observação da propriedade de objeto de interesse “Temperatura do ar” (*quality:temperature\_of\_the\_air*) ou para o sensor de temperatura, como descrito na Tabela 23. Os valores da resolução (“Resolution”) para o processo de observação do estado da bateria do modelo IP68 são apresentados na Figura 124.

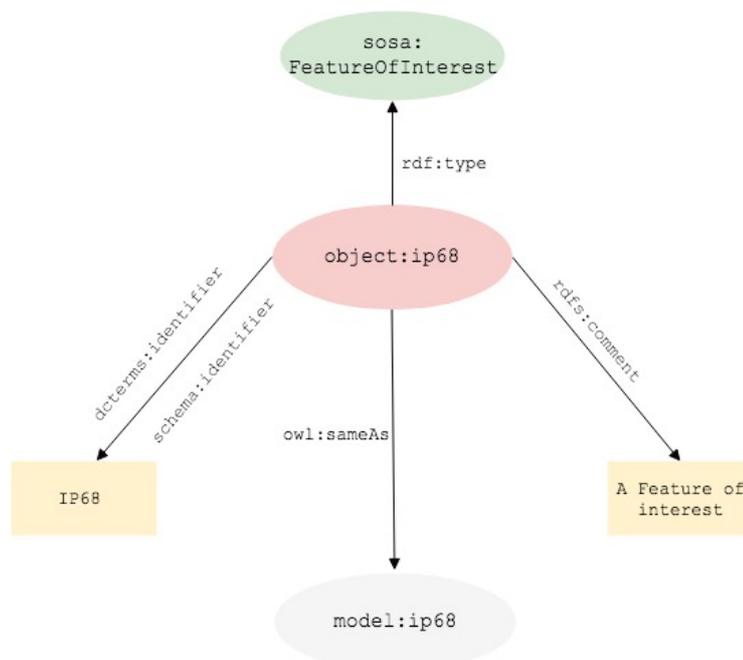


**Figura 124** - Parte do grafo RDF da capacidade de sistema como foco na propriedade “Resolution” para a propriedade “Bateria do modelo IP68”

Fonte: Elaboração própria

Com o uso da propriedade *ssn:forProperty*, vinculada às diferentes resoluções do modelo IP68 (Figuras 123 e 124), é possível esclarecer o processo de observação no qual cada propriedade de sistema pertence. No caso do grafo apresentado na Figura 124, a propriedade de objeto de interesse relacionada é a “Bateria do modelo IP68” (*quality:battery\_of\_ip68*), ou seja trata-se da resolução para o sensor de estado de bateria, como descrito na Tabela 23.

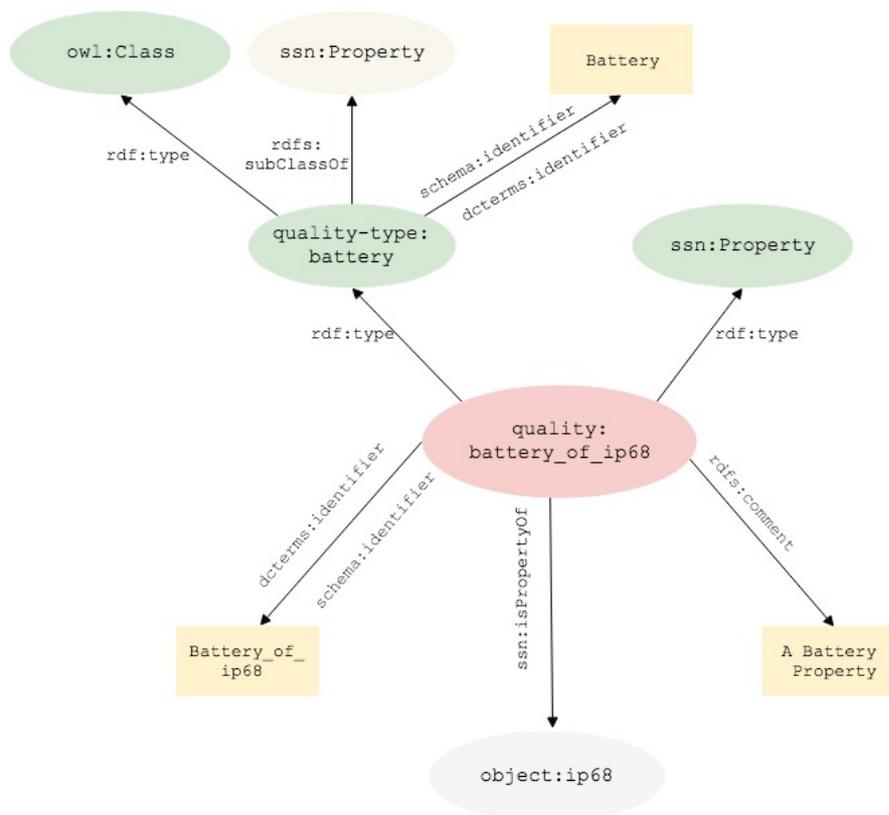
Qualquer coisa pode consistir em objeto de interesse. No caso do grafo RDF da Figura 124, é possível verificar que a propriedade “Bateria do modelo IP68” é dotada de objeto de interesse “IP68”. Tal objeto de interesse consiste no próprio modelo IP68, configurando-se, portanto, como um objeto de interesse genérico. Antes mesmo desta propriedade existir, foi necessário catalogar o objeto de interesse em questão. A Figura 125 apresenta o grafo RDF do objeto de interesse IP68 (modelo de sensor IP68).



**Figura 125** - Grafo RDF do objeto de interesse “IP68”

Fonte: Elaboração própria

Na Figura 125, faz-se uso da propriedade *owl:sameAs* para explicitar que o modelo IP68 (*model:ip68*) é exatamente o objeto de interesse catalogado (*object:ip68*). A Figura 126 apresenta o uso do objeto de interesse *object:ip68* na descrição de propriedade.

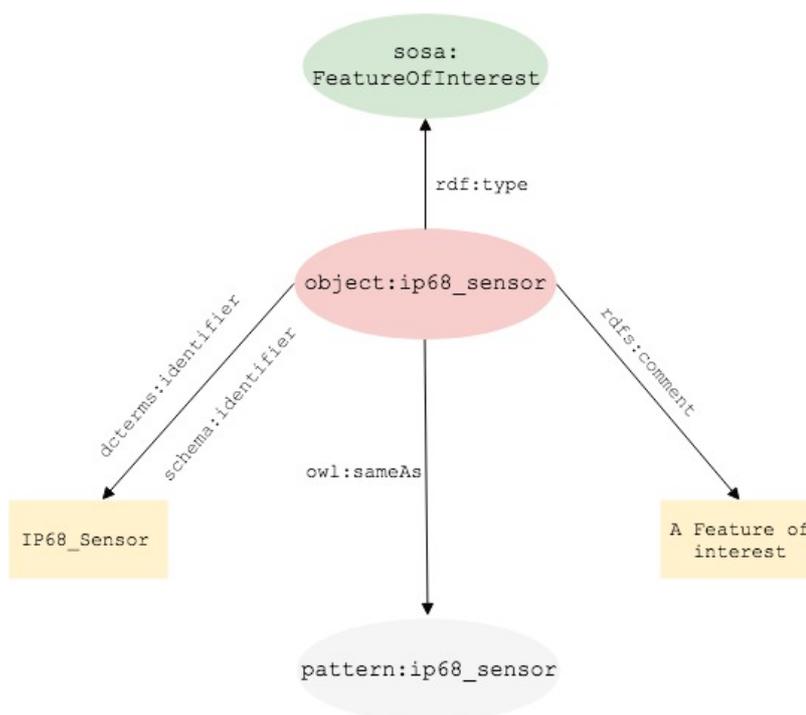


**Figura 126** - Grafo RDF contendo a descrição das propriedades *quality:battery\_of\_ip68* e *quality-type:battery*

Fonte: Elaboração própria

A Figura 126 apresenta o grafo RDF da descrição da propriedade de objeto de interesse “Bateria do modelo IP68” (*quality:battery\_of\_ip68*) e do tipo de propriedade “Bateria” (*quality-type:battery*). Para descrever a propriedade “Bateria do modelo IP68”, o objeto de interesse “IP68” (*object:ip68*) foi associado na propriedade *ssn:isPropertyOf*. De forma similar, o tipo de propriedade “Bateria” foi vinculado à propriedade “Bateria do modelo IP68” por meio da propriedade *rdf:type*, tendo como resultado a definição da propriedade catalogada como sendo uma instância da classe “Bateria”.

Por razões óbvias, modelos de sensores devem ser associados a propriedades de objetos de interesse genéricas, enquanto seus exemplares podem ser associados a propriedades de objetos de interesse genéricas ou específicas. Assim, optou-se por associar o modelo de sensor IP68 à propriedade de objeto de interesse genérica (*quality:battery\_of\_ip68*) e o exemplar do sensor IP68 à propriedade de objeto de interesse específica (*quality:battery\_of\_ip68\_sensor*). A Figura 127 apresenta o grafo RDF do objeto de interesse (específico) utilizado na descrição da propriedade de objeto de interesse *quality:battery\_of\_ip68\_sensor* (específica).

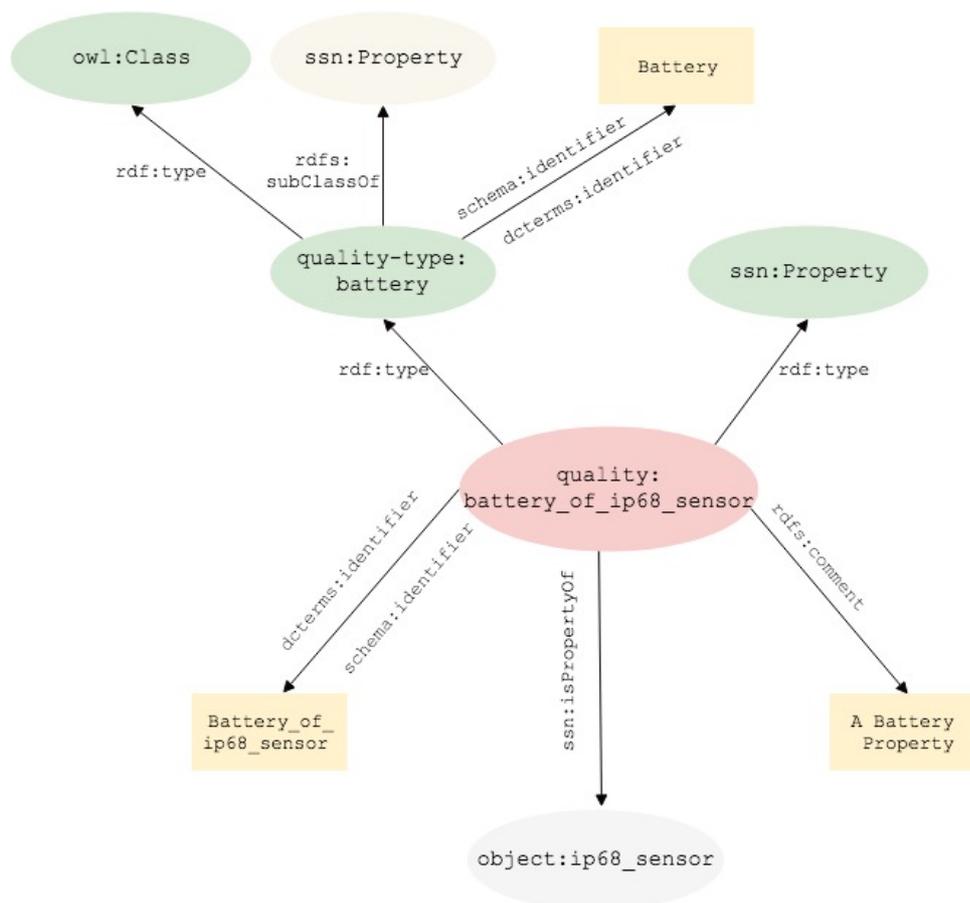


**Figura 127** - Grafo RDF do objeto de interesse “IP68 sensor”

Fonte: Elaboração própria

O objeto de interesse apresentado na Figura 127 é utilizado na descrição da propriedade de objeto de interesse “Estado da Bateria do Sensor IP68”. Trata-se de propriedade específica, a medida que o exemplar “Sensor IP68” é o objeto de interesse desta propriedade. Tal fato fica

evidente na Figura 127, à medida que é possível verificar o exemplar o URI do sensor IP68, previamente catalogado, como sendo o valor associado na propriedade *owl:sameAs*. O grafo RDF da propriedade de interesse “Estado da Bateria do Sensor IP68” (*quality:battery\_of\_ip68\_sensor*) pode ser visualizado na Figura 128.

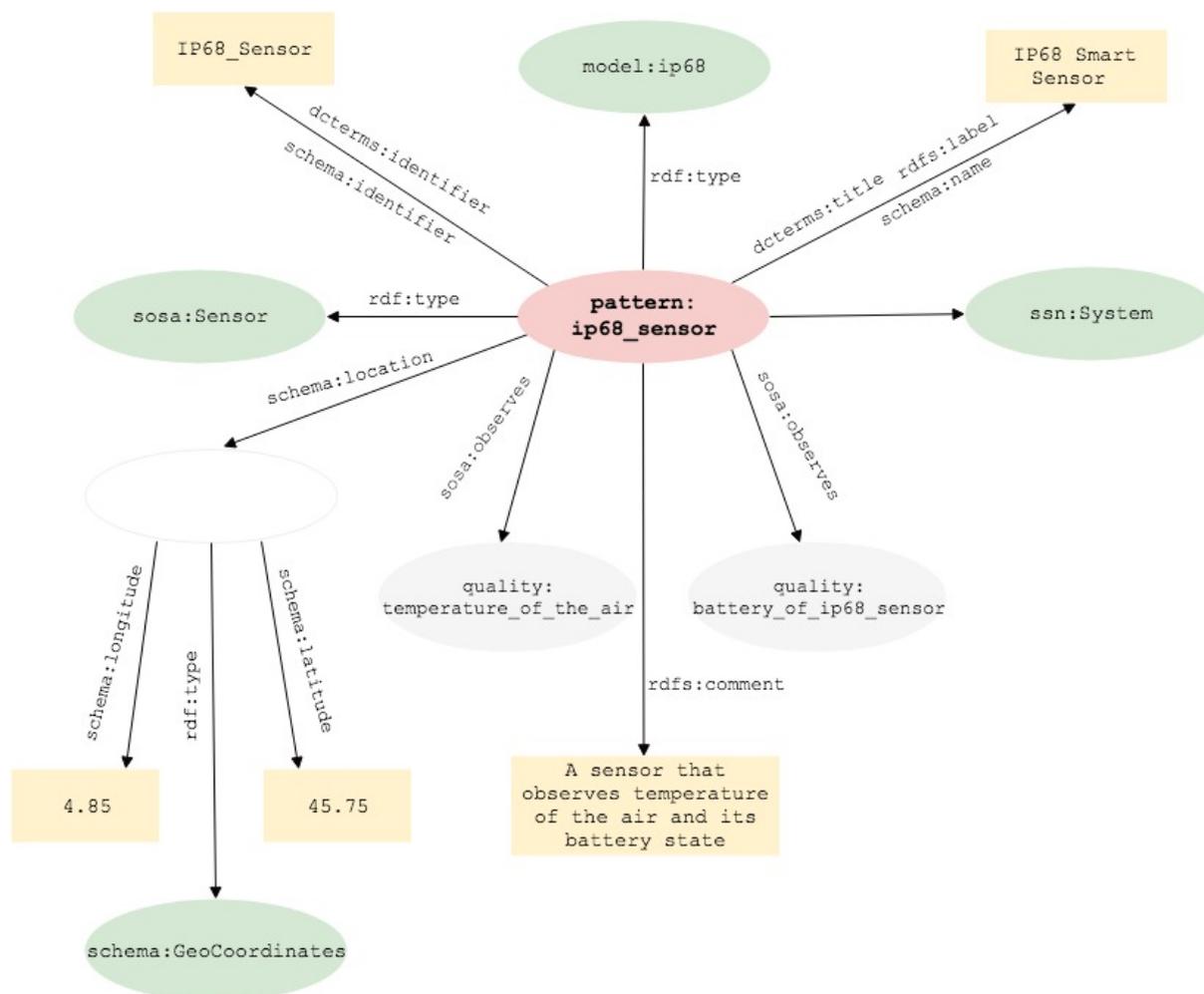


**Figura 128** - Grafo RDF contendo a descrição das propriedades *quality:battery\_of\_ip68\_sensor* e *quality-type:battery*

Fonte: Elaboração própria

O uso da propriedade de objeto de interesse apresentada na Figura 128 na descrição de do sensor IP68, explicita que este sensor observa o estado de sua própria bateria. Nesse caso, fica claro que tal exemplar é tanto um sensor, quanto um objeto de interesse. Além disso, é possível verificar que a propriedade de objeto de interesse “Estado da Bateria do Sensor IP68” não é definida com sendo *sosa:ObservableProperty* - por decisão de projeto, optou-se por não definir propriedades de objeto de interesse como propriedades de observação ou atuação separadamente, por meio do uso das classes *sosa:ObservableProperty* e *sosa:ActuatableProperty*, pois este refinamento pode ser automaticamente obtido por inferência. Assim, propriedades de objeto de interesse são definidas como *ssn:Property*, a exemplo da propriedade *quality:battery\_of\_ip68\_sensor* indicada na Figura 128.

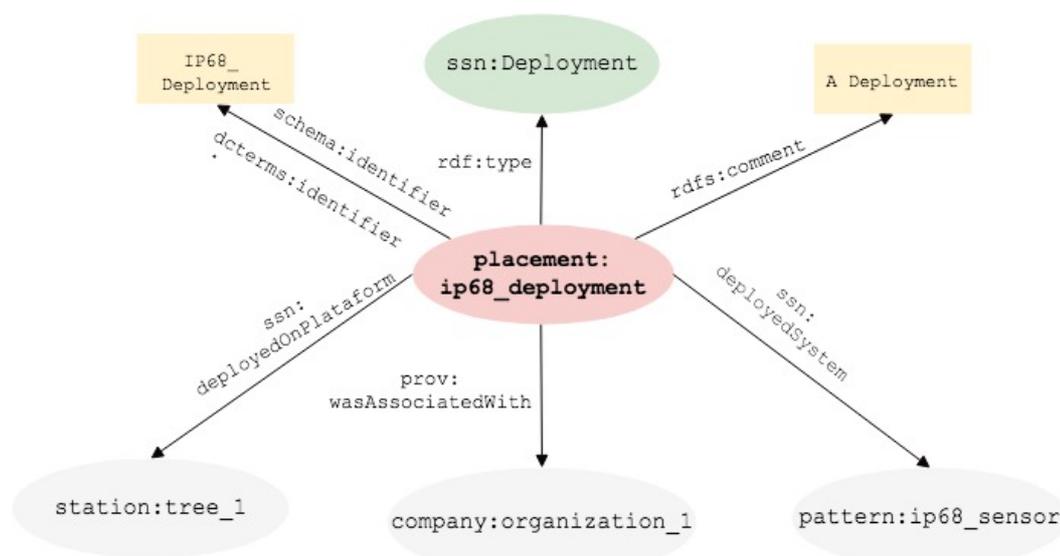
Após a catalogação do modelo IP68, dos objetos de interesse envolvidos e propriedades observadas, foi possível realizar a catalogação do exemplar do modelo IP68, cuja descrição é apresentada na Figura 129.



**Figura 129** - Grafo RDF do exemplar do modelo IP68

Fonte: Elaboração própria

A Figura 129 evidencia a descrição do exemplar do modelo IP68 no catálogo semântico proposto. Tal exemplar é vinculado a propriedade de objeto de interesse genérica “Temperatura do ar” (*quality:temperature\_of\_the\_air*) e a propriedade de objeto de interesse específica “Bateria do sensor IP68” (*quality:battery\_of\_ip68\_sensor*). Ainda, ao associar o valor *model:ip68* à propriedade *rdf:type*, torna-se explícito que o sensor catalogado é uma instância da classe modelo IP68 (*model:ip68*). Além disso, a Figura 129 apresenta a descrição das propriedades de localização de sensores - o sensor catalogado está situado na longitude 4.58 e latitude 45.75. Este sensor faz parte também de uma implantação. Detalhes sobre tal implantação podem ser visualizados na Figura 130.



**Figura 130** - Grafo RDF da implantação do sensor IP68

Fonte: Elaboração própria

A Figura 130 evidencia os diferentes recursos envolvidos na implantação do sensor IP68. Além do próprio sensor IP68, esta implantação envolve, ainda, uma plataforma (*station:tree\_1*) e organização (*company:organization\_1*).

#### B. Passive Infra-Red Sensor (555-28027)

O sensor *Passive Infra-Red Sensor* (PIR) é capaz de observar variação de sinal infravermelho, sendo, portanto, utilizado na detecção de movimento. O modelo do sensor PIR a ser descrito é o 555-28027<sup>22</sup>, do fabricante “Parallax”. A Tabela 24 apresenta os dados utilizados para descrever o sensor PIR 555-28027.

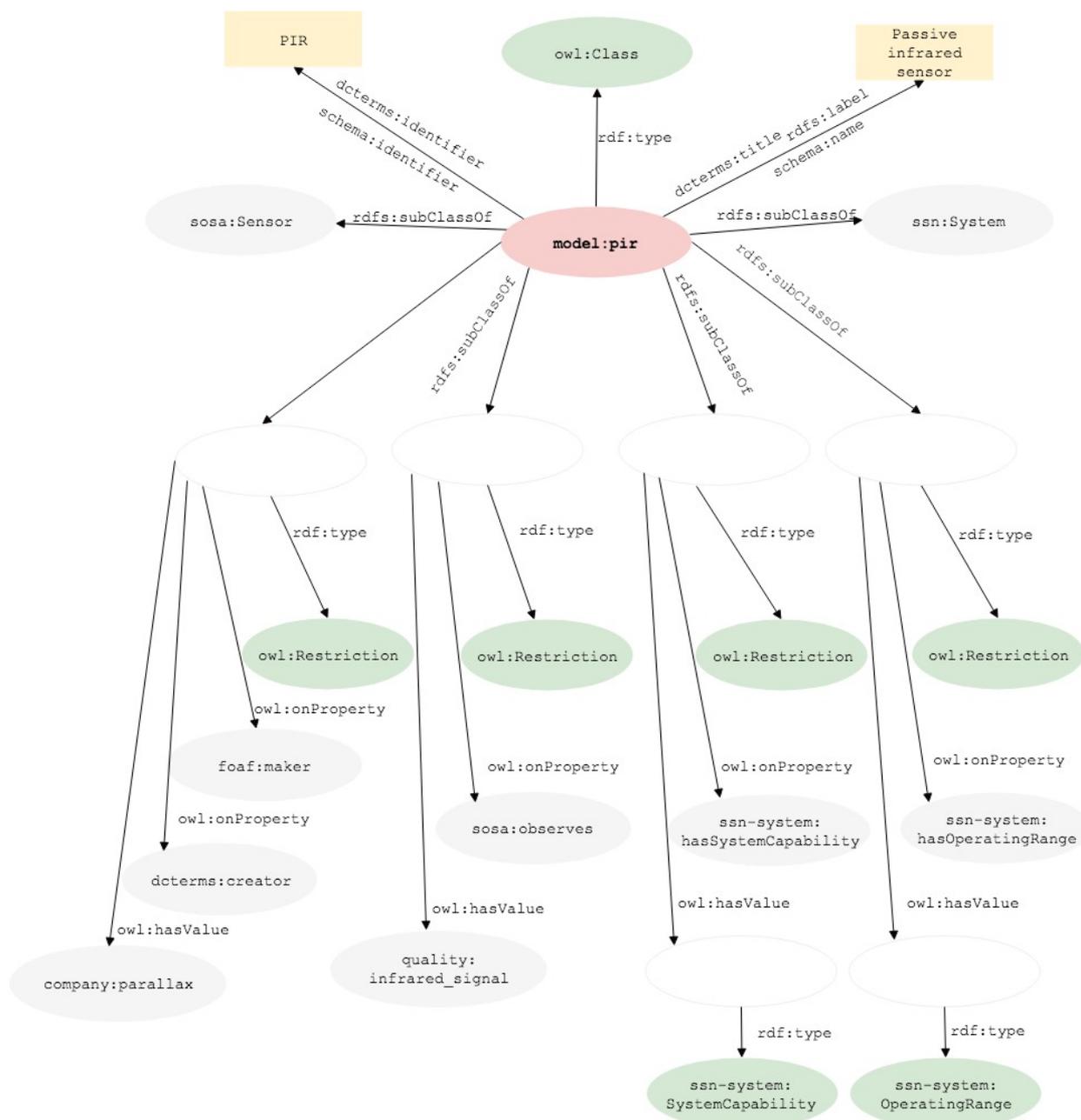
**Tabela 24-** Dados utilizados na descrição do sensor PIR 555-28027.

Item	Especificação
Modelo do sensor	PIR 555-28027
Propriedade observada	Variação de Sinal infravermelho
Capacidade de sistema	Dada em condição normal de temperatura (0 a 50°C)
Faixa de operação	Dada em condição normal de temperatura (0 a 50°C)
Sensibilidade	15 a 30 pés
Tensão de alimentação	3 a 6 V
Fabricante	Parallax

Fonte: Elaboração própria

Com base nos dados técnicos indicados na Tabela 24, foi possível catalogar o modelo de sensor PIR 555-28027. A Figura 131 apresenta o grafo RDF subjacente ao cadastro deste modelo.

<sup>22</sup> <https://www.parallax.com/sites/default/files/downloads/555-28027-PIR-Sensor-Prodcut-Doc-v2.2.pdf>

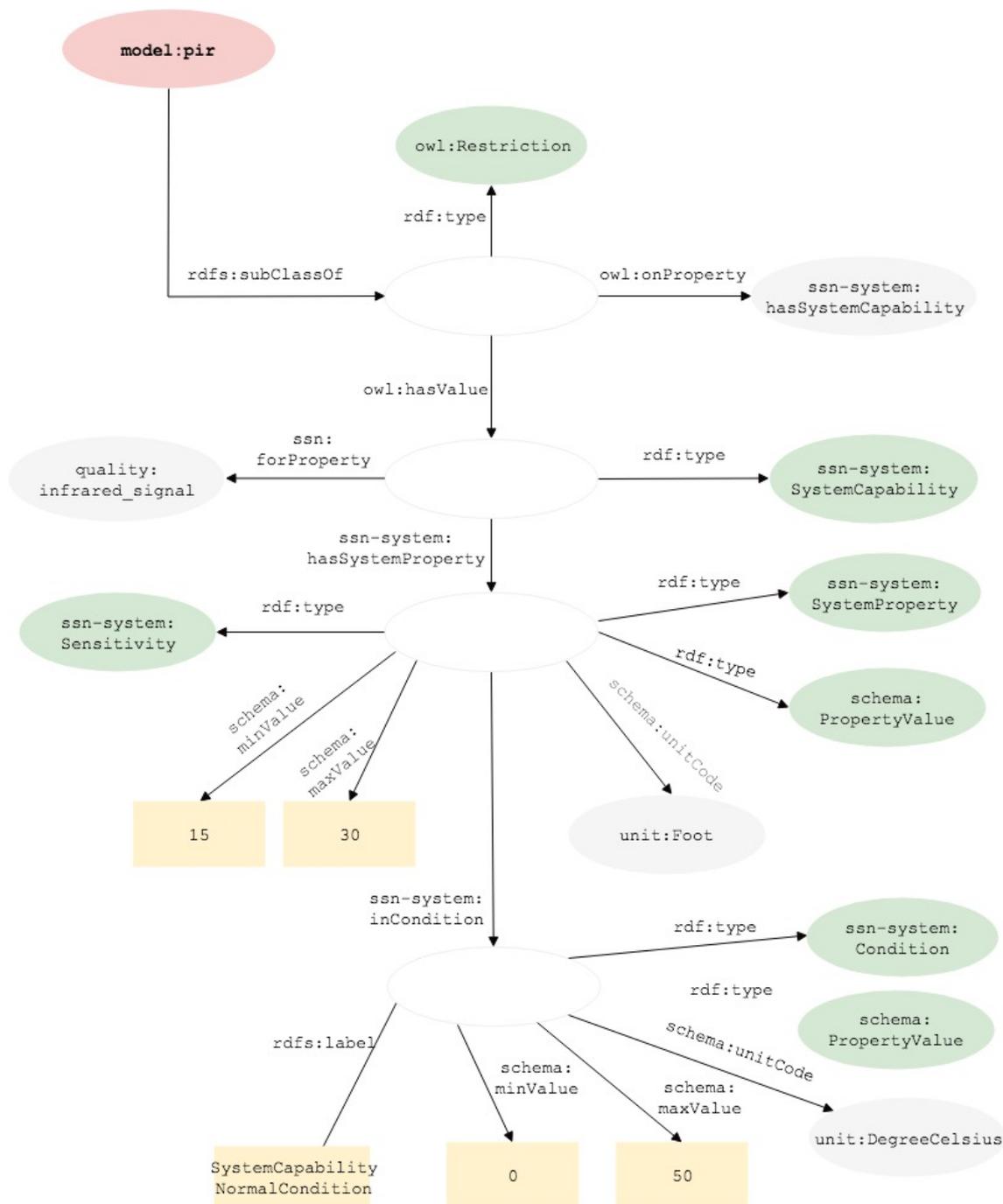


**Figura 131** - Visão geral do grafo RDF do modelo de sensor PIR 555-28027

Fonte: Elaboração própria

A Figura 131 evidencia os diferentes recursos utilizados na descrição do modelo de sensor “PIR 555-28027”. Por exemplo, é possível verificar o fabricante “Parallax” nas propriedades *dcterms:creator* e *foaf:maker*. Além disso, a propriedade observada “Sinal Infravermelho” (*quality:infrared-signal*) é associada na propriedade *sosa:observes*. Vale ressaltar que tais informações são herdadas por todas as instâncias deste modelo, tendo em vista as restrições de valor apresentadas na Figura 131. O mesmo princípio se aplica à capacidade de sistema e faixa de operação. No entanto, por questão de espaço, a descrição completa de tais elementos não é apresentada na Figura 131. A Figura 132 apresenta a parte do grafo RDF

subjacente ao cadastro do modelo com foco na capacidade e propriedade de sistema do modelo catalogado.

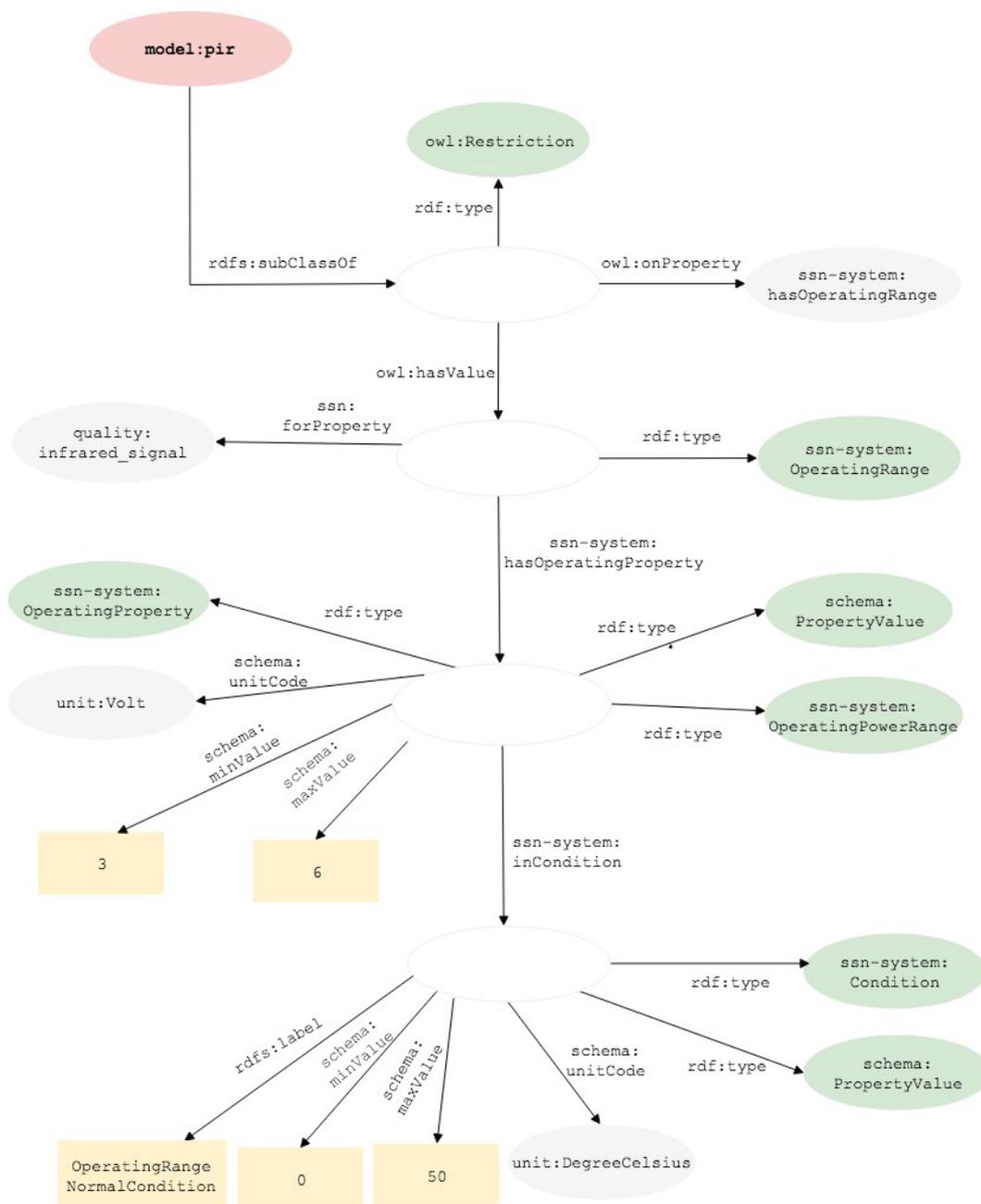


**Figura 132** - Grafo RDF do modelo de sensor PIR (555-28027) com foco na propriedade de sistema “Sensitivity”

Fonte: Elaboração própria

O modelo de sensor descrito na Figura 132 observa apenas a propriedade “Sinal infravermelho” (*quality:infrared\_signal*) e possui condição de funcionamento única (0 a 50 °C), vide Tabela 24. No exemplo apresentado na Figura 132, optou-se por vincular à propriedade observada (Sinal infravermelho) à descrição da capacidade de sistema e a condição

de funcionamento à descrição da propriedade de sistema “Sensitivity” (Sensibilidade). Desse modo, fica explícito que o valor da sensibilidade foi dimensionado pelo fabricante levando em consideração que o sensor é submetido à temperatura entre 0 e 50 °C. Além da propriedade de sistema “Sensitivity”, to modelo catalogado modelo apresenta, ainda, a propriedade de operação “OperatingPowerRange”. A Figura 133 apresenta mais detalhes da faixa e propriedade de operação do modelo em questão.



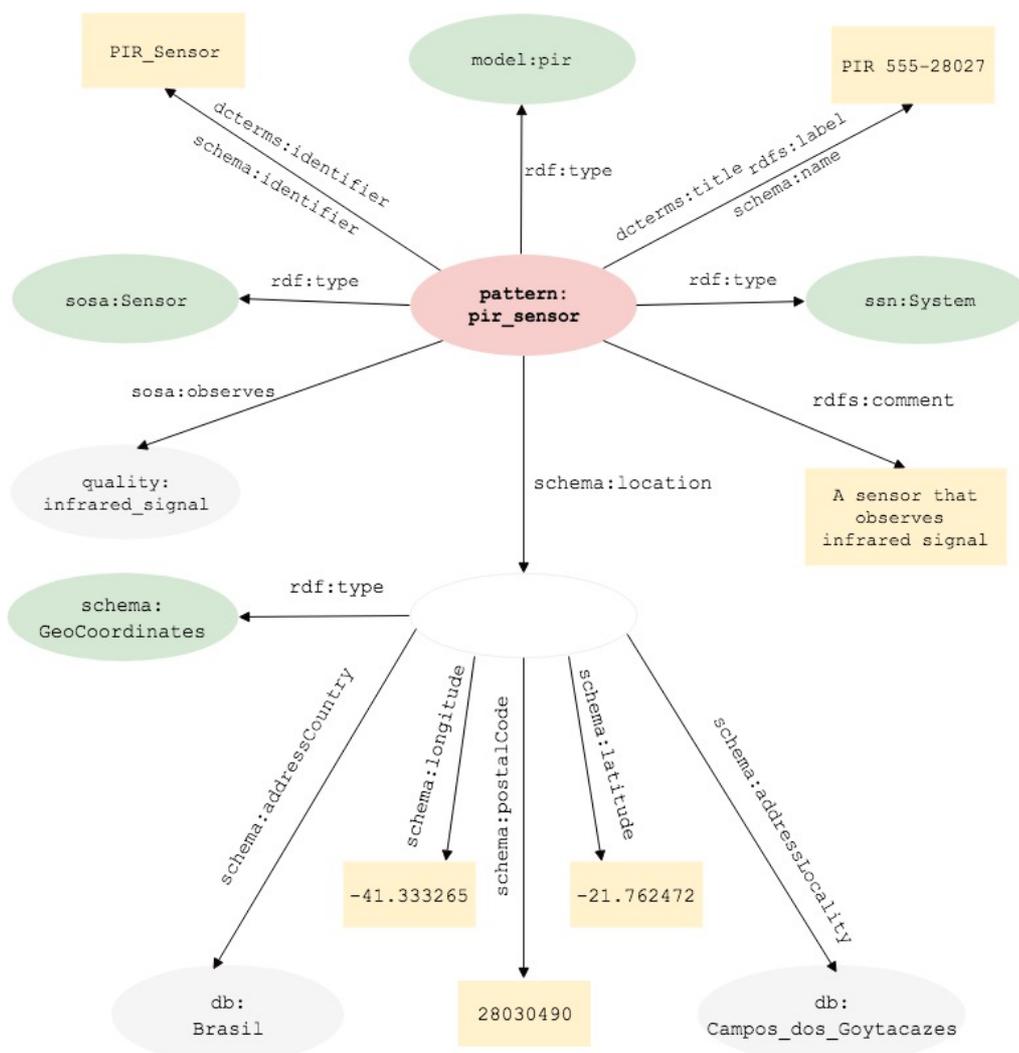
**Figura 133** - Grafo RDF do modelo de sensor PIR (555-28027) com foco na propriedade de operação “OperatingPowerRange”

Fonte: Elaboração própria

A Figura 133 evidencia a faixa de operação do modelo de sensor PIR (555-28027), cuja propriedade de operação “OperatingPowerRange” consiste na faixa de tensão de alimentação

na qual o modelo em questão deve ser submetido para funcionar corretamente ou de acordo com o esperado (3 a 6 V). A determinação deste range de tensão foi dimensionado pelo fabricante levando em consideração condições de temperatura específicas (0 a 50 °C).

Uma vez catalogado, o grafo RDF do modelo PIR (555-28027) fica disponível na base de dados da aplicação e pode ser associado aos seus exemplares. A Figura 134 apresenta a catalogação de exemplar deste modelo.



**Figura 134 - Grafo RDF do exemplar do modelo PIR (555-28027)**

Fonte: Elaboração própria

Na Figura 134, pode-se verificar que não foi necessário descrever as características técnicas do sensor, tendo em vista que tal dispositivo é associado a modelo previamente cadastrado na aplicação que contém os dados técnicos deste sensor. A descrição deste sensor inclui, ainda, propriedades de localização e de observação. Tais informações ficam disponíveis na base de dados da aplicação e podem ser utilizadas como base para consultas de sensores de interesse.

## 8. TRABALHOS RELACIONADOS

Aplicações e serviços em diversas áreas de conhecimento consomem massivas quantidades de dados de sensores todos os dias: sistemas de cidades inteligentes, de monitoramento de tráfego, de controle aéreo e de carros autônomos são alguns exemplos. Tais aplicações, normalmente, estão interessadas apenas em buscar os dispositivos de interesse e requisitar o acesso aos seus dados. Com o intuito de facilitar esta busca, esforços para interligar dados de dispositivos físicos à Web têm crescido nos últimos anos (GUINARD & TRIFA, 2016).

Em Barnaghi & Presser (2010), os autores ressaltam a necessidade de desenvolver aplicações para busca, entrega e apresentação de dados do mundo real para consumo de outras aplicações e usuários no campo de sensores. Com base nesta ideia, Barnaghi & Presser (2010) apresentam o desenvolvimento da plataforma “Sense2Web” que faz uso dos princípios do *Linked Data* e da ontologia SSN. Nesta plataforma, os usuários são capazes de registrar seus dados, buscar aplicações e requisitar dados de dispositivos. Os links com URIs de terceiros foram realizados principalmente com as bases de dados DBpedia e Geonames<sup>23</sup>.

Outro exemplo de trabalho envolvendo *Linked Data* aplicado à área de sensores é apresentado em Patni *et al.* (2010). Este trabalho consiste no desenvolvimento de um *framework* para publicação na nuvem LOD que atende a demanda de sensores. Diferente do trabalho Barnaghi & Presser (2010), não se trata de uma aplicação colaborativa, onde qualquer usuário pode registrar seus dados. O usuário final ou aplicações em geral podem apenas buscar pelos sensores registrados na plataforma e requerer informações dos dispositivos. Cerca de 20.000 dados de sensores (e suas observações) de diferentes estações meteorológicas dos Estados Unidos foram descritos e disponibilizados na aplicação. Os dados foram ligados à fonte de dados Geonames resultando em mais um bilhão de triplas RDF.

Em Barnaghi *et al.* (2013) um *framework* para anotação semântica e distribuição de dados de redes de sensores é proposto. Nesse caso, este trabalho se diferencia dos demais aqui abordados à medida que concentra esforços nas redes de sensores e suas observações e não apenas em sensores isolados. Para atender esta demanda, um vocabulário foi construído para descrever detalhes sobre os dados de observação das redes de sensores, tendo em vista que a ontologia SSN se mostrou insuficiente para esta aplicação. Foram atribuídos aos dados

---

<sup>23</sup> <http://www.geonames.org/>

informações de tempo, localização, valor, etc. Os links semânticos foram realizados com recursos da DBpedia e Geonames.

Por meio de plataformas como Callimachus, a união das ideias dos trabalhos discutidos nesta seção seria possível e facilitada, tendo como resultado uma aplicação para anotação semântica de sensores como apresentado em Patni *et al.* (2010) e colaborativa, onde usuários de qualquer parte do mundo poderiam registrar os metadados de seus sensores e de suas observações, assim como proposto em Barnaghi & Presser (2010). Sofisticando ainda mais, poder-se-ia utilizar a ontologia SSN atrelada a um vocabulário que pudesse oferecer mais detalhes aos dados de observação, como apresentado em Barnaghi *et al.* (2013). Tudo isso em uma aplicação em consonância com os princípios *Linked Data*. Esta possibilidade de fundir várias destas ideias em uma única aplicação foi uma das motivações para o desenvolvimento do protótipo proposto neste trabalho.

Abordagens mais recentes discutem anotação e publicação de dados de sensores e suas observações sob o ponto de vista da Internet das Coisas (IoT), na qual a interoperabilidade entre diferentes plataformas se tornou uma questão importante e que, portanto, deve ser considerada no processo de criação de *frameworks* para publicação de dados de sistemas IoT, em larga escala. Em aplicações desta natureza, o uso de modelos que envolvam muitos detalhes aumenta a complexidade e o tempo de processamento das informações, sendo, portanto, modelos inadequados para sistemas dinâmicos e responsivos tais como as aplicações IoT. Assim, mediante o desafio de gerar expressividade com baixa complexidade em sistemas IoT, Bermudez-edo *et al.* (2017) propõem um modelo para descrição de aspectos chaves de sistemas IoT, utilizando como base a ontologia SSN. O IoT-Lite, nome dado ao *framework* em questão, é o resultado de pesquisas, cujo foco consiste na utilização mínima de conceitos e relacionamentos que ofereçam respostas para a maioria das consultas de usuários finais. Neste modelo, as dimensões temporal, espacial e alguns atributos específicos que qualificam o sensor quanto ao tipo de variável medida, suas características técnicas, tais como resolução e unidade de medição são definidas. Ainda, há especificação quanto ao serviço sob o qual o elemento sensor é exposto.

Nos últimos anos, diferentes vocabulários foram desenvolvidos com propósito de atender as diferentes demandas do campo de sensores. Em Wang *et al.* (2015), as principais ontologias para descrição de sensores, com destaque para a *Semantic Sensor Network Ontology* (SSN), são apresentadas. Comumente, extensões deste modelo são realizadas de modo que demandas específicas de aplicações possam ser atendidas. Por exemplo, em alguns casos a SSN

é utilizada como base para construir modelos ainda mais genéricos, a exemplo dos modelos para aplicações IoT. Em outros casos, existe o interesse em unir ou explorar diferentes ontologias em um mesmo modelo para que os dados sejam descritos com maior abrangência, sob os diversos aspectos e atributos, tal como o modelo para catalogação de metadados de sensores proposto neste trabalho. Com propósito de acompanhar estas novas demandas, o grupo desenvolvedor da ontologia SSN, *The W3C Semantic Sensor Network Incubator group*, publicou uma nova versão da ontologia SSN em 2017. Esta nova versão ontologia SSN alinha-se com as demandas de aplicações IoT, à medida que um vocabulário mais leve e flexível é proposto (HALLER *et al.*, 2017).

Os trabalhos apresentados nesta seção apresentam alguma interseção com a proposta apresentada neste trabalho, sobretudo, pelo uso da ontologia SSN que se tornou um padrão *de facto* para este domínio de conhecimento. Todavia, a aplicação proposta neste trabalho é voltada exclusivamente para catalogação das características dos sensores, sem se preocupar com os dados de observação que ficam a cargo das aplicações cliente que fazem uso do catálogo proposto. Tendo como foco a descrição de sensores e atuadores, a aplicação proposta se diferencia por agregar outras ontologias além da SSN, descrevendo tais dispositivos sob outros aspectos (pessoas e organizações, localização, base de dados, etc.) aumentando, pois, a completude do processo de catalogação.

Em 2005, uma pesquisa realizada pela empresa de tecnologia Gartner previu que até 2015 as redes de sensores sem fio estariam em todos os lugares e formariam uma nova Web. Mas, que isso só teria valor se os dados gerados pudessem ser coletados, analisados e interpretados (RASKINO *et al.*, 2015). A pesquisa bibliográfica realizada neste trabalho evidencia que a empresa Gartner estava certa – os esforços para disponibilizar os dados de dispositivos físicos na Web de forma interoperável é uma realidade e se apresenta como uma tendência. Além disso, o desenvolvimento de plataformas que facilitam esta prática, a exemplo do Callimachus, contribui ainda mais para que essa tendência não pare de crescer.

Graças às tecnologias da Web Semântica, desenvolvedores podem criar aplicações para armazenamento de dados na Web, construir vocabulários e escrever regras para o tratamento de dados. Embora tais tecnologias para dar suporte à Web de Dados estejam em constante desenvolvimento, inúmeras aplicações já fazem seu uso. Os usuários de tais aplicações, por sua vez, podem sentir os resultados da nova Web quando experimentam serviços que apresentam soluções cada vez mais sofisticadas, rápidas e inteligentes para seus problemas.

## 9. RESULTADOS

O presente trabalho atingiu o objetivo proposto que consistiu no desenvolvimento de aplicação (protótipo) *Linked Data* para catalogação de sensores e atuadores, tendo em vista que os quatro princípios fundamentais para publicação de dados ligados explanados por Berners-Lee *et al.* (2006) foram atendidos: URIs são utilizadas para nomear os recursos cadastrados na aplicação, as URIs possuem acesso HTTP, retornam arquivo RDF e são ligadas a outros recursos (ou URIs) na Web. Os links semânticos (*mashup*) foram realizados, principalmente, com a fonte de dados DBpedia, que é o coração da nuvem LOD. No entanto, além da DBpedia, buscou-se explorar outras bases de dados semânticas, tais como WordNet e Library of Congress. Com o uso das bases WordNet e Library of Congress pôde-se suprir necessidades que a DBpedia não atende ou se mostra insuficiente. O reuso de informações de terceiros enriqueceu a aplicação, uma vez que aumentou o universo da base de conhecimento.

A seleção das ontologias configurou-se como uma etapa importante do trabalho. Buscou-se utilizar ontologias consagradas que fazem parte do catálogo LOV. Esta prática faz toda diferença em projetos de aplicações de dados ligados, tendo em vista a necessidade de produzir informação interoperável. Além disso, a adoção de conjunto diversificado de ontologias *Linked Data* propiciou a descrição de sensores sob diversos aspectos e atributos, diferenciando a aplicação desenvolvida de outras identificadas na revisão bibliográfica.

O uso da ontologia SSN permitiu que, além de sensores, atuadores pudessem ser catalogados. Embora a aplicação tenha como foco inicial o cadastro de sensores e de recursos que agregam valor à descrição de tais dispositivos, com os modelos para descrição de sensores em mãos, optou-se por aproveitar tal oportunidade para adaptá-los de modo a atender o contexto dos atuadores. Ao propiciar o cadastro de sensores e atuadores na aplicação aumentam-se as chances de atender, com maior completude, os sistemas Internet das Coisas ou Web das Coisas. Tal fato ficou evidente no cenário realista apresentado envolvendo sensores e atuadores, a medida que todos os elementos identificados na implantação Web das Coisas puderam ser catalogados fazendo uso da aplicação semântica desenvolvida.

Para fins de experimentação do modelo de descrição de sensores e atuadores, exemplos de uso foram reproduzidos neste trabalho. A partir desta prática foi possível demonstrar o nível de expressividade alcançado ao descrever uma dada situação, fazendo uso do catálogo semântico proposto. Na reprodução de tais exemplos, ficou evidente que a aplicação é capaz de descrever os problemas apresentados contemplando os aspectos propostos neste trabalho, tais

como localização e características técnicas. Além disso, foi possível verificar a catalogação de recursos que resultam na extensão da ontologia SSN, por consistirem em subclasses diretas de classes da ontologia SSN, a exemplo dos modelos de sensores e atuadores.

As concepções iniciais deste trabalho foram publicadas em artigo científico e podem ser consultadas em Rangel & Jacyntho (2017). Desde sua publicação, novos resultados foram obtidos e refletem a evolução do desenvolvimento deste trabalho. Em outras palavras, maiores esforços foram concentrados na análise e estudo de ontologias, tendo como resultado a inclusão de novos termos para descrição dos recursos. Além disso, novos atributos para qualificar sensores foram identificados, resultando no aumento de funcionalidades da aplicação. Por meio destes e outros avanços futuros, uma vez disponível na Web de Dados, recursos poderão ser cadastrados na aplicação e dados estruturados, inteligíveis por máquina e enriquecidos pelas bases de dados da nuvem LOD, serão gerados. Isso permitirá que máquinas auxiliem na busca precisa por informações, bem como em tomadas de decisão.

## 10. CONSIDERAÇÕES FINAIS

Para que sensores e atuadores sejam apropriadamente utilizados, eles precisam ser facilmente encontrados. Para serem encontrados, nada melhor que sejam catalogados e publicados na Web. O ideal é que as informações catalogadas sejam inteligíveis não apenas por humanos, mas, sobretudo, por agentes de software que possam, de forma autônoma, encontrar, selecionar, combinar e usar tais dispositivos para auxiliarem na resolução de um problema específico. O catálogo semântico proposto vem ao encontro deste ideal, a medida que permite que usuários ao redor do mundo cataloguem seus dispositivos de acordo com os novos padrões da Web.

A aplicação proposta possibilita que dados de sensores e atuadores sejam catalogados de forma estruturada, inteligível por seres humanos e máquinas e enriquecidos por outras bases de dados da nuvem LOD. Esta prática contribui com o reuso de informação na Web, torna as buscas mais precisas e possibilita análises mais profundas sobre os dados. Além disso, por apresentar o uso das tecnologias da Web Semântica de maneira prática, o fornecimento de diretrizes para o desenvolvimento de aplicações *Linked Data*, envolvendo a área de sensores e atuadores configura mais uma contribuição importante deste trabalho - o cadastro de um único sensor envolve conceitos fundamentais da nova Web: modelo de dados RDF, modelos de representação do conhecimento (ontologias) e linguagem de consulta SPARQL.

Antes mesmo do desenvolvimento do catálogo, foi necessário analisar e selecionar um conjunto de ontologias para construção de modelos ontológicos para descrição de sensores, atuadores e recursos qualificam tais dispositivos. A seleção de ontologias levou em consideração o reconhecimento e a consonância dos vocabulários com os padrões definidos pelo W3C. O uso combinado dos termos dos diferentes vocabulários adotados resultou na elaboração de modelos ontológicos. Tais modelos serviram como base para o desenvolvimento do catálogo e podem ser reutilizados no desenvolvimento de aplicações similares.

Dentre as ontologias utilizadas, a *Semantic Sensor Network Ontology - SSN* pode ser considerada a fonte principal de conhecimento para o desenvolvimento deste trabalho, pois é resultado de diversos esforços aplicados, ao longo dos anos, para a inserção de dispositivos físicos, principalmente sensores, na Web. O catálogo proposto, contribui para que esta ontologia se desenvolva ainda mais, a medida que permite a catalogação de recursos, que representam novos termos para descrição de sensores e atuadores e ficam disponíveis na base de dados da aplicação para serem reutilizados por qualquer usuário ou agente de software.

## 11. TRABALHOS FUTUROS

Ao longo do desenvolvimento do trabalho, oportunidades de avanços significativos foram identificadas. Assim, trabalhos futuros devem, certamente, compreender a criação ou seleção de uma ontologia para descrever como usar o serviço oferecido pelo sensor catalogado, ou seja, a descrição da interação (protocolo de comunicação, URL onde o sensor está fisicamente acessível, formato de dados de entrada, formato de dados de saída, etc.). Além disso, percebeu-se a necessidade analisar mais a fundo as ontologias adotadas a fim de verificar a necessidade de explorar ainda mais tais ontologias, a exemplo da SSN. Embora diversos termos da ontologia SSN tenham sido utilizados na descrição dos sistemas envolvidos no trabalho, é nítido que alguns elementos, que não foram utilizados, podem enriquecer ainda mais o processo de catalogação de sensores e atuadores. Por exemplo, o registro semântico dos dados observados. No caso destas observações, é indispensável pensar em uma arquitetura *Big Data* para os dados obtidos em tempo real ou quase real. Para este fim, a tecnologia de *semantic blockchain*<sup>24</sup> (RUTA *et al.*, 2017) surge como importante opção a ser investigada.

Ainda no decorrer do desenvolvimento do trabalho, percebeu-se também a necessidade de estender ou de criar vocabulários novos de modo que as demandas atuais e futuras de projeto sejam atendidas. Avanços futuros devem contemplar, ainda, a definição de perfis de usuários para catalogação dados de sensores e atuadores, assim como a validação da aplicação com potenciais usuários. Além disso, por ser resultado de adaptação, o modelo para descrição de atuadores, certamente, deverá ser revisto em trabalhos futuros a fim de identificar atributos mais específicos que possam melhor qualificar atuadores. Espera-se que as melhorias futuras possam levar a um produto final totalmente funcional, um portal semântico de sensores e atuadores que seja usado em larga escala, por diversos usuários ao redor do mundo, quer seja cadastrando sensores e atuadores, quer seja buscando tais dispositivos para usar em suas aplicações.

---

<sup>24</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

Abele, A., Mccrae, J. LOD Cloud Diagram. Disponível em: <<http://lod-cloud.net>>. Acesso em: 16 mar. 2018.

Atzori et al., The Internet of Things: A survey, *Comput. Netw.* (2010), doi:10.1016/j.comnet.2010.05.010

Bao, J., Kendall, E. F., McGuinness, D. L., Patel-Schneider, P. F., (editors). OWL 2 Web Ontology Language: Quick Reference Guide (Second Edition). W3C Recommendation, 11 December 2012, <http://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/>. Latest version available at <http://www.w3.org/TR/owl2-quick-reference/>.

Barnaghi, P., Compton, M., Corcho, O., Castro, R.G., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., Page, K. Semantic Sensor Network XG Final Report. W3C Recommendation 28 June 2011. Disponível em: <<https://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/>>. Acesso em: 18 maio. 2017.

Barnaghi, P., Presser, M. Publishing Linked Sensor Data, 2010. In The 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010). Shanghai, China.

Barnaghi, P., Wang, W., Dong, L., Wang, C., 2013. A linked-data model for semantic sensor streams. Presented at the Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCoM 2013, pp. 468– 475.

Berners-Lee, T., Hall, W., Hendler, J., Shadbolt, N., Weitzner, D.J., 2006. Creating a Science of the Web. *Science*. 313, 769–771.

Berners-Lee, T., Hendler, J., Lassila, O., 2001. The semantic web. *Sci. Am.* 284, 34–43.

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K., 2017. IoT-Lite: A lightweight semantic model for the Internet of things and its use with dynamic semantics. *Pers. Ubiquitous Comput.* 21, 475–487.

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.

Brickley, D., Guha, R. V. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>. Latest version available at <http://www.w3.org/TR/rdf-schema/>

Brickley, D., Miller, L. FOAF Vocabulary Specification 0.99. W3C Recommendation, 14 January 2014. Latest version available at <http://xmlns.com/foaf/spec/>.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K., 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semant. Sci. Serv. Agents World Wide Web.* 17, 25–32.

Cyganiak, R., Wood, D., Lanthaler, M. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, 25 February 2014. Latest version available at <https://www.w3.org/TR/rdf11-concepts/>

De, S, Elsaleh, T, Barnaghi, P and Meissner, S, 2012. An Internet of Things Platform for Real-World and Digital Objects Scalable. *Computing: Practice and Experience*, 13 (1). pp. 45-57.

Dean, M., Schreiber, G., (editors). OWL Web Ontology Language Reference. W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-owl-ref-20040210/> . Latest version available at <http://www.w3.org/TR/owl-ref/> .

Freitas, N. J., and Jacyntho, M. D. Um protótipo Linked data para catalogação semântica de publicações. *Perspect. ciênc. inf.* 2016, vol.21, n.4, pp.48-65. Epub Jan 16, 2017. ISSN 1981-5344. <http://dx.doi.org/10.1590/1981-5344/2664>.

Giusto, D.; Iera, A.; Morabito, G.; Atzori, L. (Orgs), 2010. *The Internet of Things: 20<sup>th</sup> Tyrrhenian Workshop on Digital Communication*. Springer: Nova Iorque, Dordrecht, Heidelberg, Londres.

Guinard, D.D., Trifa, V.M., 2016. *Building the Web of Things with examples in Node.js and Raspberry Pi*. Shelter Island, NY. June 2016 . ISBN 9781617292682

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/>

Harris, S.; Seaborne, A.; Prud'hommeaux, E. SPARQL 1.1 Query Language. W3C Recommendation, 2013. Disponível em: <<http://www.w3.org/TR/sparql11-query/>>. Acesso em: 16 mar. 2018.

Hausenblas, M. *Linked Data Applications - The genesis and the challenges of using Linked Data on the Web: DERI Technical Report 2009-07-26*.

Heflin, J., 2004. OWL Web Ontology Language Use Cases and Requirements. W3C Recommendation 10 February 2004. Disponível em: <<https://www.w3.org/TR/2004/REC-webont-req-20040210/>> Acessado em 19.07.17).

Hepp, M. Good Relations- The Web Vocabulary for E-commerce. 2011. Disponível em: <http://www.heppnetz.de/projects/goodrelations/>

Iannella, R., McKinnney, J., vCard Ontology - for describing People and Organizations. W3C Interest Group Note 22 May 2014. Lastest version available at <http://www.w3.org/TR/vcard-rdf/>

Jacyntho, M. D.; Schwabe, D. A multigranularity locking model for RDF. Web Semantics: Science, Services and Agents on the World Wide Web, v. 39, n. 0, 4 ago. 2016.

Kansal, A., Nath S., Liu J., Zhao, F. SenseWeb: An Infrastructure for Shared Sensing. 2007. IEEE MultiMedia (Volume: 14, Issue: 4, Oct.-Dec. 2007 )

Lebo, T., Sahoo, S., McGuinness, D. PROV-O: The Prov Ontology. W3C Recommendation 30 April 2013. Lastest version available at <https://www.w3.org/TR/prov-o/>

Maali, F., Erickson, J. Data Catalog Vocabulary (DCAT). W3C Recommendation 16 January 2014. Lastest version available at <https://www.w3.org/TR/vocab-dcat/>

Manola, F.; Miller, E. RDF Primer. Disponível em: <<https://www.w3.org/TR/2004/REC-rdf-primer-20040210/>>. Acesso em: 18 maio. 2017.

Miles, A., Bechhofer, S. SKOS Simple Knowledge Organization System Reference. W3C Recommendation 18 August 2009. Lastest version available at <http://www.w3.org/TR/skos-reference>

Morais, M. D.; Jacyntho, M.D., 2015. Issue Procedure Ontology (IPO): Uma ontologia extensível para o domínio de sintomas, problemas e soluções. Tese de mestrado - Programa de Pós-Graduação Universidade Cândido Mendes, Campos dos Goytacazes, RJ. 2015.

Neves, R., Jacyntho, M.D., 2014. Um modelo baseado em ontologias Linked Data para catalogação de projetos de Software. 12ª Conferência Ibero-Americana WWW/Internet 2014. Porto, Portugal, Anais da Conferência Ibero-Americana WWW/INTERNET 2014.

Noy, N. F.; McGuinness, D. L. Ontology development 101: A guide to creating your first ontology. Stanford Medical Informatics Technical Report, SMI-2001- 0880, 2001. Disponível em: <[https://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](https://protege.stanford.edu/publications/ontology_development/ontology101.pdf)>. Acesso em: 30 mar. 2018.

Patni, H., Henson, C., Sheth, A., 2010. Linked sensor data. Presented at the 2010 International Symposium on Collaborative Technologies and Systems, CTS 2010, Dayton, Ohio, pp. 362–370.

Pfisterer, D., Römer, K. U., Bimschas, D., Hasemann, H., Hauswirth, M., Karnstedt, M., Truong, C. (2011). SPITFIRE: Toward a Semantic Web of Things. *IEEE communications magazine*, 49(11), 40-48. DOI: 10.1109/MCOM.2011.6069708

Phuoc, L., Nguyen-Mau, Parreira, J.X., Hauswirth, M., 2011. The Linked Sensor Middleware - Connecting the real world and the Semantic Web. 9th Semantic Web Challenge co-located with 10th International Semantic Web Conference–ISWC 2011. Boon, Germany.

Prud'hommeaux, E.; Seaborne, A. SPARQL Query Language for RDF. Disponível em: <<https://www.w3.org/TR/rdf-sparql-query/>>. Acesso em: 18 maio. 2017.

Rangel, T., Jacyntho, M.D., 2017. Catalogação semântica de sensores com base nos princípios Linked Data. 16ª Conferência Ibero-Americana WWW/Internet 2017. Algarve, Portugal, Anais da Conferência Ibero-Americana WWW/INTERNET 2017.

Raskino, M., Fenn, J., Liden, A., 2005. Extracting Value From the Massively Connected World of 2015. Gartner Research., Stanford, CT, USA, Tech. Rep. G00125949, Apr. 2005 Disponível em: <<https://www.gartner.com/doc/476440/extracting-value-massively-connected-world>>. Acesso em: 6 abr. 2017.

Reynolds, D. The Organization Ontology. W3C Recommendation 16 January 2014. Lastest version available at <https://www.w3.org/TR/vocab-org/>

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.

Wang, X., Zhang, X., Li, M., 2015. A Survey on Semantic Sensor Web: Sensor Ontology, Mapping and Query. *International Journal of u- and e- Service, Science and Technology*. v8, pp. 325–342.

3 Round Stones: The Callimachus Project, 2009. Sample Organization Directory Webapp. Disponível em: <https://github.com/3-Round-Stones/directory>. Acessado em 01/08/2017.

## APÊNDICE

Este apêndice tem como objetivo apresentar os grafos RDF (em sintaxe Turtle) gerados como resultado dos cadastros dos recursos apresentados nos capítulos 6 e 7.

### A. Descrição de pessoa

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Person/mark?describe> a <http://www.w3.org/ns/ldp#RDFSSource> ;
  foaf:primaryTopic <http://localhost:8080/Resource/Person/mark> .
<http://localhost:8080/Resource/Person/mark>
  a <http://localhost:8080/app-sensor-
catalog/Person/Person>,
  schema:Person,
  foaf:Agent,
  foaf:Person ;
  rdfs:label "Mark Zuckerberg"^^xsd:string ;
  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> ;
  dcterms:created "2018-06-29T12:15:18.980000+00:00"^^xsd:dateTime ;
  dcterms:identifier "Mark"^^xsd:string ;
  dcterms:modified "2018-06-29T12:16:14.519000+00:00"^^xsd:dateTime ;
  dcterms:title "Mark Zuckerberg"^^xsd:string ;
  schema:identifier "Mark"^^xsd:string ;
  schema:name "Mark Zuckerberg"^^xsd:string ;
  rdfs:comment "Founder and CEO of Facebook"^^xsd:string ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/08/31/t16590d39853x926#provenance> ;
  foaf:birthday "1984-05-14"^^xsd:string ;
  foaf:mbox <mailto:markz@example.com> ;
  foaf:name "Mark Zuckerberg"^^xsd:string ;
  foaf:nick "Mark"^^xsd:string ;
  foaf:phone "1818850780"^^xsd:string ;
  owl:sameAs <http://dbpedia.org/resource/Mark_Zuckerberg> .

```

**Figura 135** - Grafo RDF subjacente ao cadastro de pessoa no catálogo semântico em Turtle

Fonte: Elaboração própria

## B. Descrição de organização

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix gr: <http://purl.org/goodrelations/v1#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Organization/sparkfun a <http://localhost:8080/app-sensor-
catalog/Organization/Organization>,
    schema:Organization,
    vcard:Organization,
    org:Organization,
    foaf:Organization ;
rdfs:label "Sparkfun Eletronics"^^xsd:string ;
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> ;
dcterms:created "2018-07-13T22:45:37.970000+00:00"^^xsd:dateTime ;
dcterms:identifier "Sparkfun"^^xsd:string ;
dcterms:modified "2018-08-31T16:41:58.636000+00:00"^^xsd:dateTime ;
dcterms:title "Sparkfun Eletronics"^^xsd:string ;
gr:legalName "Sparkfun Eletronics"^^xsd:string ;
schema:identifier "Sparkfun"^^xsd:string ;
schema:location _:N6a8f8226e7f0419aa547e5501fd2a758 ;
schema:name "Sparkfun Eletronics"^^xsd:string ;
rdfs:comment "Sparkfun products are released as open-source hardware"^^xsd:string ;
vcard:adr _:N6a8f8226e7f0419aa547e5501fd2a758,
    [ a vcard:Address ] ;
vcard:hasURL <http://www.sparkfun.com> ;
vcard:organization-name "Sparkfun Eletronics"^^xsd:string ;
org:classification <http://localhost:8080/app-sensor-catalog/Organization/type-of-
organization/commercial> ;
org:hasMember <http://localhost:8080/Resource/Person/adam_swetlix> ;
prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/08/31/t16590d39853x1652#provenance> ;
foaf:homepage <http://www.sparkfun.com> ;
foaf:name "Sparkfun Eletronics"^^xsd:string .

_:N6a8f8226e7f0419aa547e5501fd2a758 a schema:GeoCoordinates,
    vcard:Address ;
schema:addressCountry <http://dbpedia.org/resource/United_States> ;
schema:postalCode "80503"^^xsd:integer ;
vcard:locality <http://dbpedia.org/resource/Sydney> ;
vcard:street-address "6333 Dry Creek Parkway"^^xsd:string .

```

**Figura 136** - Grafo RDF subjacente ao cadastro de organização no catálogo semântico em Turtle

Fonte: Elaboração própria

### C. Descrição de Base de dados

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Dataset/sensor_database?describe>
<http://www.w3.org/ns/ldp#RDFSsource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Dataset/sensor_database> .
<http://localhost:8080/Resource/Dataset/sensor_database> a <http://localhost:8080/app-sensor-
catalog/Dataset/Dataset>, dcat:Dataset ;
    rdfs:label "Sensor_Database"^^xsd:string ;
    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> ;
    dcterms:created "2018-07-02T01:31:17.538000+00:00"^^xsd:dateTime ;
    dcterms:creator <http://localhost:8080/Resource/Person/mark> ;
    dcterms:description "A fictional Sensor Database"^^xsd:string ;
    dcterms:identifier "Sensor_Database"^^xsd:string ;
    dcterms:language <http://id.loc.gov/vocabulary/iso639-1/en> ;
    dcterms:modified "2018-07-02T01:34:48.149000+00:00"^^xsd:dateTime,
        "2018-07-02T01:45:29.167000+00:00"^^xsd:dateTime,
        "2018-08-31T17:04:25.786000+00:00"^^xsd:dateTime ;
    dcterms:subject <http://wordnet-rdf.princeton.edu/ontology#climate_change-n> ;
    dcterms:title "Sensor_Database"^^xsd:string ;
    schema:identifier "Sensor_Database"^^xsd:string ;
    schema:name "Sensor_Database"^^xsd:string ;
    rdfs:comment "A fictional Sensor Database"^^xsd:string ;
    dcat:distribution [ a dcat:Distribution ;
        dcterms:description "CSV distribution of imaginary Sensor Database"^^xsd:string ;
        dcterms:issued "2015-05-03"^^xsd:date ;
        dcterms:title "Sensor_Database"^^xsd:string ;
        dcat:byteSize "5120"^^xsd:integer ;
        dcat:downloadURL <http://www.example.com/files/001.csv> ;
        dcat:mediaType "text/csv"^^xsd:string ] ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/08/31/t16590d39853x2987#provenance> ;
    foaf:homepage <http://example.com>;
    foaf:maker <http://localhost:8080/Resource/Person/mark> .

```

**Figura 137** - Grafo RDF subjacente ao cadastro de base de dados no catálogo semântico em Turtle

Fonte: Elaboração própria

### D. Descrição do objeto de interesse “ar”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/FeatureOfInterest/air?describe> a ldp:RDFSource ;
  foaf:primaryTopic <http://localhost:8080/Resource/FeatureOfInterest/air> .
<http://localhost:8080/Resource/FeatureOfInterest/air> a <http://localhost:8080/app-sensor-
catalog/Feature-of-Interest/FeatureOfInterest>,
  sosa:FeatureOfInterest ;
  schema:identifier " " "Air"^^xsd:string ;
  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> ;
  dcterms:created "2018-01-24T02:58:42.616000+00:00"^^xsd:dateTime ;
  dcterms:modified "2018-05-13T23:46:21.583000+00:00"^^xsd:dateTime,
    "2018-05-14T00:09:31.233000+00:00"^^xsd:dateTime,
    "2018-05-14T00:09:46.340000+00:00"^^xsd:dateTime,
    "2018-05-14T00:10:42.989000+00:00"^^xsd:dateTime ;
  dcterms:identifier "Air"^^xsd:string ;
  rdfs:comment "The air from atmosphere"^^xsd:string ;
  rdfs:label "The air from atmosphere"^^xsd:string ;
  schema:name "The air from atmosphere"^^xsd:string ;
  dcterms:title "The air from atmosphere"^^xsd:string ;
  owl:sameAs <http://dbpedia.org/resource/Atmosphere_of_Earth> ;
  prov:wasGeneratedBy
    <http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx4169#provenance> .

```

**Figura 138** - Grafo RDF subjacente ao cadastro do objeto de interesse “ar” no catálogo semântico em Turtle

Fonte: Elaboração própria

### E. Descrição do tipo de propriedade “Temperatura”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Property/temperature?describe> a
<http://www.w3.org/ns/ldp#RDFSource> ;
  foaf:primaryTopic <http://localhost:8080/Resource/Property/temperature> .
<http://localhost:8080/Resource/Property/temperature> a
  <http://localhost:8080/app-sensor-catalog/PropertyType/PropertyType>,
  owl:Class ;
  rdfs:subClassOf ssn:Property ;
  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> ;
  dcterms:created "2018-05-13T23:50:32.759000+00:00"^^xsd:dateTime ;
  dcterms:identifier "Temperature"^^xsd:string ;
  dcterms:title "Temperature"^^xsd:string ;
  dcterms:modified "2018-05-13T23:50:53.166000+00:00"^^xsd:dateTime ;
  schema:identifier "Temperature"^^xsd:string ;
  rdfs:label "Temperature"^^xsd:string ;
  schema:name "Temperature"^^xsd:string ;
  rdfs:comment "A Temperature class"^^xsd:string ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx3207#provenance> ;
  owl:sameAs <http://dbpedia.org/resource/Temperature> .

```

**Figura 139** - Grafo RDF subjacente ao cadastro de tipo de propriedade “Temperatura” no catálogo semântico em Turtle

Fonte: Elaboração própria

### F. Descrição da propriedade “Temperatura do ar”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Property/temperature_of_the_air?describe> a
<http://www.w3.org/ns/ldp#RDFSource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Property/temperature_of_the_air> .
<http://localhost:8080/Resource/Property/temperature_of_the_air> a
<http://localhost:8080/Resource/PropertyType/temperature>,
    <http://localhost:8080/app-sensor-catalog/Property/Property>,
    ssn:Property ;
    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> ;
    dcterms:created "2018-05-13T23:50:32.759000+00:00"^^xsd:dateTime ;
    dcterms:identifier "Temperature_of_the_air"^^xsd:string ;
    dcterms:modified "2018-05-13T23:50:53.166000+00:00"^^xsd:dateTime ;
    schema:identifier "Temperature_of_the_air"^^xsd:string ;
    rdfs:comment "Temperature Property"^^xsd:string ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx3207#provenance> ;
    ssn:isPropertyOf <http://localhost:8080/Resource/FeatureOfInterest/air> .

```

**Figura 140** - Grafo RDF subjacente ao cadastro de propriedade de objeto de interesse “Temperatura do ar” em Turtle

Fonte: Elaboração própria

### G. Descrição da propriedade “Umidade do ar”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Property/humidity_of_the_air?describe> a
<http://www.w3.org/ns/ldp#RDFSource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Property/humidity_of_the_air> .
<http://localhost:8080/Resource/Property/humidity_of_the_air> a
<http://localhost:8080/Resource/PropertyType/humidity>,
    <http://localhost:8080/app-sensor-catalog/Property/Property>,
    ssn:Property ;
    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> ;
    dcterms:created "2018-05-13T23:50:32.759000+00:00"^^xsd:dateTime ;
    dcterms:identifier "Humidity_of_the_air"^^xsd:string ;
    dcterms:modified "2018-05-13T23:50:53.166000+00:00"^^xsd:dateTime ;
    schema:identifier "Humidity_of_the_air"^^xsd:string ;
    rdfs:comment "Humidity Property"^^xsd:string ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx3207#provenance> ;
    ssn:isPropertyOf <http://localhost:8080/Resource/FeatureOfInterest/air> .

```

**Figura 141** - Grafo RDF subjacente ao cadastro de propriedade de objeto de interesse “Umidade do ar” em Turtle

Fonte: Elaboração própria

### H. Descrição de procedimento do modelo DHT22

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Procedure/dht22_procedure?describe> a ldp:RDFSResource ;
    foaf:primaryTopic <http://localhost:8080/Resource/Procedure/dht22_procedure> .
<http://localhost:8080/Resource/Procedure/dht22_procedure> a <http://localhost:8080/app-
sensor-catalog/Procedure/Procedure>, sosa:Procedure ;
    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> ;
    dcterms:created "2018-05-14T00:24:23.791000+00:00"^^xsd:dateTime ;
    dcterms:identifier "DHT22_Procedure"^^xsd:string ;
    schema:identifier "DHT22_Procedure"^^xsd:string ;
    ssn:hasInput [ a ssn:Input ;
        rdfs:label "DHT22 input"^^xsd:string ;
        rdfs:comment "Power's voltage should be 3.3-6V DC. When power is
supplied to sensor, don't send any instruction to the sensor within 1s to pass unstable
status." ],
    ssn:forProperty <http://localhost:8080/Resource/Property/relative_humidity_of_the_air>,
        <http://localhost:8080/Resource/Property/temperature_of_the_air> ;
    rdfs:comment "The DHT22 Procedure"^^xsd:string ;
    prov:wasGeneratedBy<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx4931#pro
venance> ;

```

**Figura 142** - Grafo RDF subjacente ao cadastro de procedimento do modelo de sensor DHT22 em Turtle

Fonte: Elaboração própria

### I. Descrição de propriedade de sistema (Histerese)

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn-system: <http://www.w3.org/ns/ssn/systems> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/SystemProperty/hysteresis?describe> a
<http://www.w3.org/ns/ldp#RDFSResource> ;
  foaf:primaryTopic <http://localhost:8080/Resource/SystemProperty/hysteresis> .
<http://localhost:8080/Resource/SystemProperty/hysteresis> a
  <http://localhost:8080/app-sensor-catalog/SystemProperty/SystemProperty>,
  owl:Class;
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> ;
dcterms:created "2018-05-13T23:50:32.759000+00:00"^^xsd:dateTime ;
dcterms:identifier "Hysteresis"^^xsd:string ;
dcterms:title "Hysteresis"^^xsd:string ;
dcterms:modified "2018-05-13T23:50:53.166000+00:00"^^xsd:dateTime ;
schema:identifier "Hysteresis"^^xsd:string ;
schema:name "Hysteresis"^^xsd:string ;
rdfs:label "Hysteresis"^^xsd:string ;
rdfs:comment "Hysteresis refers to the characteristic that a transducer has in being
unable to repeat faithfully, in the opposite direction of operation, the data that have been
recorded in one direction"^^xsd:string ;
rdfs:subClassOf
  ssn-system:SystemProperty ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx3207#provenance> .

```

**Figura 143** - Grafo RDF subjacente ao cadastro de propriedade de sistema “Histerese” em Turtle

Fonte: Elaboração própria

## J. Descrição do modelo DHT22

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix ssn-system: <http://www.w3.org/ns/ssn/systems/> .
@prefix unit: <http://qudt.org/1.1/vocab/unit#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/dht22?describe> a ldap:RDFSSource ;
  foaf:primaryTopic <http://localhost:8080/Resource/SensorModel/dht22> .
<http://localhost:8080/Resource/SensorModel/dht22> a <http://localhost:8080/app-sensor-
catalog/Sensor-Model/SensorModel>,
  owl:Class ;
  rdfs:label "DHT22 model"^^xsd:string ;
  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> ;
  dcterms:created "2018-05-13T20:43:04.550000+00:00"^^xsd:dateTime ;
  dcterms:identifier "DHT22"^^xsd:string ;
  dcterms:modified "2018-05-13T21:07:57.598000+00:00"^^xsd:dateTime,
    "2018-05-13T21:13:28.728000+00:00"^^xsd:dateTime,
    "2018-05-13T21:19:58.558000+00:00"^^xsd:dateTime,
    "2018-05-13T21:24:42.552000+00:00"^^xsd:dateTime ;
  dcterms:title "DHT22 model"^^xsd:string ;
  schema:identifier "DHT22"^^xsd:string ;
  schema:name "DHT22 model"^^xsd:string ;
  rdfs:isDefinedBy <http://localhost:8080/Resource/Person/mark>;
  rdfs:comment "A sensor that observes the temperature a humidity of the air";
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:hasValue
<http://localhost:8080/Resource/Property/relative_humidity_of_the_air>,
  <http://localhost:8080/Resource/Property/temperature_of_the_air> ;
      owl:onProperty sosa:observes ],
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Organization/sparkfun> ;
      owl:onProperty foaf:maker,
      owl:onProperty dcterms:creator ],
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Procedure/dht22_procedure>;
      owl:onProperty ssn:implements ],
    [ a owl:Restriction ;
      owl:hasValue [ a ssn-system:OperatingRange ;
        ssn:forProperty
<http://localhost:8080/Resource/Property/relative_humidity_of_the_air>,
<http://localhost:8080/Resource/Property/temperature_of_the_air> ;
        ssn-system:hasOperatingProperty [ a schema:PropertyValue,
          ssn-system:OperatingPowerRange,
          ssn-system:OperatingProperty ;
          schema:unitCode unit:Volt ;
          schema:maxValue "3"^^xsd:decimal;
          schema:minValue "6"^^xsd:decimal ] ,
        ssn-system:inCondition [ a schema:PropertyValue,
          ssn-system:Condition ;
          rdfs:label "HumidityNormalCondition"^^xsd:string ;
          schema:maxValue "85"^^xsd:integer ;
          schema:minValue "5"^^xsd:integer ;
          schema:unitCode unit:Percent ] ,
        [ a schema:PropertyValue,
          ssn-system:Condition ;
          rdfs:label "TemperatureNormalCondition"^^xsd:string ;

```

Figura 144 - Grafo RDF subjacente ao cadastro do modelo de sensor DHT22 em Turtle (continua)

```

schema:maxValue "80"^^xsd:integer ;
                schema:minValue "-40"^^xsd:integer ;
                schema:unitCode unit:DegreeCelsius ] ] ;
owl:onProperty ssn-system:hasOperatingRange ],

    [ a owl:Restriction ;
      owl:hasValue [ a ssn-system:SystemCapability ;
        ssn:forProperty
          <http://localhost:8080/Resource/Property/relative_humidity_of_the_air>,
          <http://localhost:8080/Resource/Property/temperature_of_the_air>;
        ssn-system:hasSystemProperty [ a schema:PropertyValue,
          ssn-system:Sensitivity,
          ssn-system:SystemProperty ;
          schema:unitCode unit:DegreeCelsius ;
          schema:value "0.1"^^xsd:decimal;
          ssn:forProperty
            <http://localhost:8080/Resource/Property/temperature_of_the_air> ] ,
          [ a schema:PropertyValue,
            <http://localhost:8080/Resource/SystemProperty/hysteresis>,
            ssn-system:SystemProperty ;
            schema:maxValue "0.3"^^xsd:decimal ;
            schema:minValue "-0.3"^^xsd:decimal ;
            schema:unitCode unit:Percent ;
            ssn:forProperty
              <http://localhost:8080/Resource/Property/humidity_relative_of_the_air> ] ,
          [ a schema:PropertyValue,
            ssn-system:Precision,
            ssn-system:SystemProperty ;
            schema:maxValue "0.2"^^xsd:decimal ;
            schema:minValue "-0.2"^^xsd:decimal ;
            schema:unitCode unit:DegreeCelsius ;
            ssn:forProperty
              <http://localhost:8080/Resource/Property/temperature_of_the_air> ] ,
          [ a schema:PropertyValue,
            ssn-system:Frequency,
            ssn-system:SystemProperty ;
            schema:unitCode unit:Hertz ;
            schema:value "0.5"^^xsd:decimal ;
            ssn:forProperty
              <http://localhost:8080/Resource/Property/temperature_of_the_air> ] ;
          ssn-system:inCondition [ a schema:PropertyValue,
            ssn-system:Condition ;
            rdfs:label "HumidityNormalCondition"^^xsd:string ;
            schema:maxValue "85"^^xsd:integer ;
            schema:minValue "5"^^xsd:integer ;
            schema:unitCode unit:Percent ],
          [ a schema:PropertyValue,
            ssn-system:Condition ;
            rdfs:label "TemperatureNormalCondition"^^xsd:string ;
            schema:maxValue "80"^^xsd:integer ;
            schema:minValue "-40"^^xsd:integer ;
            schema:unitCode unit:DegreeCelsius ] ] ;
        owl:onProperty ssn-system:hasSystemCapability ] ,

    [ a owl:Restriction ;
      owl:hasValue [ a ssn-system:SurvivalRange ;
        ssn:forProperty
          <http://localhost:8080/Resource/Property/relative_humidity_of_the_air>,
          <http://localhost:8080/Resource/Property/temperature_of_the_air>;
        ssn-system:hasSurvivalProperty [ a schema:PropertyValue,
          ssn-system:SystemLifetime,
          ssn-system:SurvivalProperty ;
          schema:unitCode unit:year ;
          schema:value "5"^^xsd:decimal ] ] ;
      owl:onProperty ssn-system:hasSurvivalRange ] ,
    sosa:Sensor,
    ssn:System ;
prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b51b026x4001#provenance> .

```

**Figura 144 - Continuação**

Fonte: Elaboração própria

### K. Descrição do exemplar do modelo DHT22

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Sensor/dht22_sensor?describe> a
<http://www.w3.org/ns/ldp#RDFSource> ;
  foaf:primaryTopic <http://localhost:8080/Resource/Sensor/dht22_sensor> .
<http://localhost:8080/Resource/Sensor/dht22_sensor> a
<http://localhost:8080/Resource/SensorModel/dht22>, <http://localhost:8080/app-sensor-
catalog/Sensor/Sensor>,
  sosa:Sensor,
  ssn:System ;
  rdfs:label "DHT22 Sensor"^^xsd:string ;
  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> ;
  dcterms:created "2018-05-13T23:29:45.877000+00:00"^^xsd:dateTime ;
  dcterms:identifier "DHT22_Sensor"^^xsd:string ;
  dcterms:modified "2018-05-13T23:30:02.494000+00:00"^^xsd:dateTime ;
  dcterms:title "DHT22 Sensor"^^xsd:string ;
  foaf:depiction <http://localhost:8080/Resource/Image/wot.jpeg> ;
  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:decimal ;
    schema:longitude "-41.333265"^^xsd:decimal ;
    schema:postalCode "28030490"^^xsd:integer ] ;
  schema:identifier "DHT22_Sensor"^^xsd:string ;
  schema:name "DHT22 Sensor"^^xsd:string ;
  rdfs:comment "A sensor that observes the temperature and humidity of the air"^^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> ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx858#provenance> .

```

**Figura 145** - Grafo RDF subjacente ao cadastro do exemplar do modelo DHT22 em Turtle

Fonte: Elaboração própria

### L. Descrição do modelo de atuador LCD

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix ssn-system: <http://www.w3.org/ns/ssn/systems/> .
@prefix unit: <http://qudt.org/1.1/vocab/unit#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/lcd_g240641c?describe> a ldp:RDFSource ;
  foaf:primaryTopic <http://localhost:8080/Resource/ActuatorModel/> .
<http://localhost:8080/Resource/SensorModel/dht22> a <http://localhost:8080/app-sensor-catalog/Actuator-Model/ActuatorModel>,
  owl:Class ;
  rdfs:label "DHT22 model"^^xsd:string ;
  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> ;
  dcterms:created "2018-05-13T20:43:04.550000+00:00"^^xsd:dateTime ;
  dcterms:identifier "LCD_G240641C"^^xsd:string ;
  dcterms:modified "2018-05-13T21:07:57.598000+00:00"^^xsd:dateTime,
    "2018-05-13T21:13:28.728000+00:00"^^xsd:dateTime,
    "2018-05-13T21:19:58.558000+00:00"^^xsd:dateTime,
    "2018-05-13T21:24:42.552000+00:00"^^xsd:dateTime ;
  dcterms:title " LCD G240641C "^^xsd:string ;
  schema:identifier " LCD_G240641C "^^xsd:string ;
  schema:name " LCD G240641C "^^xsd:string ;
  rdfs:isDefinedBy <http://localhost:8080/Resource/Person/tamiris>;
  rdfs:comment "Toshiba CMOS Digital Integrated circuit Silicon Monolithic T6963CFG";
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Property/content_of_lcd_panel> ;
      owl:onProperty sosa:actsOnProperty ],
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Organization/toshiba> ;
      owl:onProperty foaf:maker,
        owl:onProperty dcterms:creator ],
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Procedure/lcd_g240641c_procedure>;
      owl:onProperty ssn:implements ],
    sosa:Actuator,
    ssn:System ;
  prov:wasGeneratedBy
    <http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx6537#provenance> .

```

**Figura 146** - Grafo RDF subjacente ao cadastro do modelo de atuador em Turtle

Fonte: Elaboração própria

*M. Descrição de atuador LCD*

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Actuator/g240641c_actuator?describe> a
<http://www.w3.org/ns/ldp#RDFSsource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Actuator/g240641c_actuator> .
<http://localhost:8080/Resource/Actuator/g240641c_actuator> a
<http://localhost:8080/Resource/ActuatorModel/lcd_g240641c>,
    <http://localhost:8080/app-sensor-catalog/Actuator/Actuator>,
    sosa:Actuator,
    ssn:System ;
    rdfs:label "G240641C Actuator"^^xsd:string ;
    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> ;
    dcterms:created "2018-05-14T10:29:36.130000+00:00"^^xsd:dateTime ;
    dcterms:identifier "LCD_G240641C_Actuator"^^xsd:string ;
    dcterms:modified "2018-05-13T10:57:24.933000+00:00"^^xsd:dateTime ;
    schema:identifier "LCD_G240641C_Actuator"^^xsd:string ;
    schema:location [ a schema:GeoCoordinates ;
        schema:addressCountry <http://dbpedia.org/resource/Campos_dos_Goytacazes> ;
        schema:addressLocality <http://dbpedia.org/resource/Brasil> ;
        schema:latitude "-21.762472"^^xsd:decimal ;
        schema:longitude "-41.333265"^^xsd:decimal ;
        schema:postalCode "28030490"^^xsd:integer ] ;
    rdfs:comment "Toshiba CMOS Digital Integrated circuit Silicon Monolithic
T6963CFG"^^xsd:string ;
    rdfs:seeAlso <http://devices.webofthings.io/pi/actuators/display/> ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/14/t1635c986068x2072#provenance> ;
    sosa:actsOnProperty
<http://localhost:8080/Resource/Property/content_of_lcd_g240641c_actuator> .

```

**Figura 147** - Grafo RDF subjacente ao cadastro de atuador em Turtle

Fonte: Elaboração própria

### N. Descrição de implantação WoT

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Deployment/wot_deployment?describe> a
<http://www.w3.org/ns/ldp#RDFSsource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Deployment/wot_deployment> .
<http://localhost:8080/Resource/Deployment/ip68_deployment> a <http://localhost:8080/app-
sensor-catalog/Deployment/Deployment>,
    ssn:Deployment ;
    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> ;
    dcterms:created "2018-05-14T03:25:23.566000+00:00"^^xsd:dateTime ;
    dcterms:identifier "WoT_Deployment"^^xsd:string ;
    dcterms:title "Web of Things App"^^xsd:string ;
    dcterms:modified "2018-05-14T03:26:00.232000+00:00"^^xsd:dateTime ;
    schema:identifier "WoT_Deployment"^^xsd:string ;
    schema:name "Web of Things App"^^xsd:string ;
    rdfs:label "Web of Things App"^^xsd:string ;
    prov:wasAssociatedWith <http://localhost:8080/Resource/Person/tamiris> ;
    prov:wasAttributedTo <http://localhost:8080/Resource/Person/mark> ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/14/t1635c986068x756#provenance> ;
    ssn:deployedOnPlatform <http://localhost:8080/Resource/FeatureOfInterest/raspberry_pi> ;
    ssn:deployedSystem <http://localhost:8080/Resource/Sensor/pir_sensor> ,
<http://localhost:8080/Resource/Sensor/dht22_sensor> ,
<http://localhost:8080/Resource/Sensor/camera_sensor> ,
<http://localhost:8080/Resource/Actuator/lcd_g240614c_actuator> ,
<http://localhost:8080/Resource/Actuator/led_actuator> .

```

**Figura 148** - Grafo RDF subjacente ao cadastro de implantação WoT em Turtle

Fonte: Elaboração própria

### O. Descrição da plataforma “Raspberry Pi”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Platform/raspberry_pi?describe> a
<http://www.w3.org/ns/ldp#RDFSResource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Platform/raspberry_pi> .
<http://localhost:8080/Resource/Platform/raspberry_pi> a
    <http://localhost:8080/app-sensor-catalog/Platform/Platform>,
    sosa:Platform ;
    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> ;
    dcterms:created "2018-05-13T23:50:32.759000+00:00"^^xsd:dateTime ;
    dcterms:title "Raspberry_Pi_Platform"^^xsd:string ;
    dcterms:identifier "Raspberry_Pi"^^xsd:string ;
    dcterms:modified "2018-05-13T23:50:53.166000+00:00"^^xsd:dateTime ;
    schema:identifier "Raspberry_Pi"^^xsd:string ;
    schema:name "Raspberry_Pi_Platform"^^xsd:string ;
    rdfs:label "Raspberry_Pi_Platform"^^xsd:string ;
    rdfs:comment "A Platform for sensors and actuators"^^xsd:string ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx3207#provenance> .

```

**Figura 149-** Grafo RDF subjacente ao cadastro de plataforma em Turtle

Fonte: Elaboração própria

*P. Descrição do modelo de sensor “IP68”*

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix ssn-system: <http://www.w3.org/ns/ssn/systems/> .
@prefix unit: <http://qudt.org/1.1/vocab/unit#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/SensorModel/ip68?describe> a ldp:RDFSResource ;
  foaf:primaryTopic <http://localhost:8080/Resource/SensorModel/ip68> .
<http://localhost:8080/Resource/SensorModel/ip68> a <http://localhost:8080/app-sensor-
catalog/Sensor-Model/SensorModel>,
  owl:Class ;
  rdfs:label "IP68 Smart Sensor"^^xsd:string ;
  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> ;
  dcterms:created "2018-05-14T01:40:55.799000+00:00"^^xsd:dateTime ;
  dcterms:identifier "IP68"^^xsd:string ;
  dcterms:modified "2018-05-14T01:52:32.414000+00:00"^^xsd:dateTime ;
  dcterms:title "IP68 Smart Sensor"^^xsd:string ;
  schema:identifier "IP68"^^xsd:string ;
  schema:name "IP68 Smart Sensor"^^xsd:string ;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Property/temperature_of_the_air>,
        <http://localhost:8080/Resource/Property/battery_of_ip68> ;
      owl:onProperty sosa:observes ],
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Procedure/ip68_battery_procedure>,
        <http://localhost:8080/Resource/Procedure/ip68_temperature_procedure>;
      owl:onProperty ssn:implements ],
    [ a owl:Restriction ;
      owl:hasValue [ a ssn-system:SystemCapability ;
        ssn:forProperty <http://localhost:8080/Resource/Property/battery_of_ip68>,
          <http://localhost:8080/Resource/Property/temperature_of_the_air> ;
        ssn-system:hasSystemProperty [ a schema:PropertyValue,
          ssn-system:Resolution,
          ssn-system:SystemProperty ;
          schema:unitCode unit:Percent ;
          schema:value "3.937e-3"^^xsd:decimal ;
          ssn:forProperty
            <http://localhost:8080/Resource/Property/battery_of_ip68> ] ,

```

**Figura 150** - Grafo RDF subjacente ao cadastro do modelo de sensor IP68 em Turtle  
(continua)

```

    [ a schema:PropertyValue,
      ssn-system:Accuracy,
      ssn-system:SystemProperty ;
      schema:maxValue "0.2"^^xsd:integer ;
      schema:minValue "-0.2"^^xsd:integer ;
      schema:unitCode unit:DegreeCelsius ;
      ssn:forProperty
<http://localhost:8080/Resource/Property/temperature_of_the_air> ] ,

    [ a schema:PropertyValue,
      ssn-system:Resolution,
      ssn-system:SystemProperty ;
      schema:unitCode unit:DegreeCelsius ;
      schema:value "0.065"^^xsd:decimal ;
      ssn:forProperty
<http://localhost:8080/Resource/Property/temperature_of_the_air> ] ,
    [ a schema:PropertyValue,
      ssn-system:Sensitivity,
      ssn-system:SystemProperty ;
      schema:unitCode unit:DecibelReferredToOneMilliwatt ;
      schema:value "-137"^^xsd:integer ] ;
      ssn-system:inCondition [ a schema:PropertyValue,
        ssn-system:Condition ;
        rdfs:label "SystemCapability_NormalCondition"^^xsd:string ;
        schema:maxValue "70"^^xsd:integer ;
        schema:minValue "-20"^^xsd:integer ;
        schema:unitCode unit:DegreeCelsius ] ] ;
      owl:onProperty ssn-system:hasSystemCapability ] ,
      sosa:Sensor,
      ssn:System ;
      prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx6537#provenance> .

```

**Figura 150 - Continuação**

Fonte: Elaboração própria

### Q. Descrição do objeto de interesse “Modelo de sensor IP68”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://localhost:8080/Resource/FeatureOfInterest/ip68?describe> a
<http://www.w3.org/ns/ldp#RDFSsource> ;
  foaf:primaryTopic <http://localhost:8080/Resource/FeatureOfInterest/ip68> .
<http://localhost:8080/Resource/FeatureOfInterest/ip68> a <http://localhost:8080/app-sensor-
catalog/Feature-of-Interest/FeatureOfInterest>, sosa:FeatureOfInterest ;
  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> ;
  dcterms:created "2018-05-14T02:33:29.948000+00:00"^^xsd:dateTime ;
  dcterms:identifier "IP68"^^xsd:string ;
  dcterms:modified "2018-05-14T02:33:46.156000+00:00"^^xsd:dateTime ;
  schema:identifier "IP68"^^xsd:string ;
  rdfs:comment "A Feature of Interest"^^xsd:string ;
  owl:sameAs <http://localhost:8080/Resource/SensorModel/ip68> ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx7032#provenance> .

```

**Figura 151 - Grafo RDF subjacente ao cadastro do objeto de interesse “Modelo IP68” em**

Turtle

Fonte: Elaboração própria

### R. Descrição da propriedade de objeto de interesse “Battery of IP68”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Property/battery_of_ip68?describe> a
<http://www.w3.org/ns/ldp#RDFSResource> ;
  foaf:primaryTopic <http://localhost:8080/Resource/Property/battery_of_ip68> .
<http://localhost:8080/Resource/Property/battery_of_ip68> a
<http://localhost:8080/Resource/PropertyType/battery>,
  <http://localhost:8080/app-sensor-catalog/Property/Property>,
  ssn:Property ;
  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> ;
  dcterms:created "2018-05-13T23:50:32.759000+00:00"^^xsd:dateTime ;
  dcterms:identifiier "Battery_of_ip68"^^xsd:string ;
  dcterms:modified "2018-05-13T23:50:53.166000+00:00"^^xsd:dateTime ;
  schema:identifiier "Battery_of_ip68"^^xsd:string ;
  rdfs:comment "A Battery property"^^xsd:string ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx3207#provenance> ;
  ssn:isPropertyOf <http://localhost:8080/Resource/FeatureOfInterest/ip68> .

```

Figura 152 - Grafo RDF subjacente ao cadastro da propriedade “Bateria do modelo IP68” em

Turtle

Fonte: Elaboração própria

### S. Descrição do objeto de interesse “Sensor IP68”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/FeatureOfInterest/ip68_sensor?describe> a
<http://www.w3.org/ns/ldp#RDFSResource> ;
  foaf:primaryTopic <http://localhost:8080/Resource/FeatureOfInterest/ip68_sensor> .
<http://localhost:8080/Resource/FeatureOfInterest/ip68_sensor> a <http://localhost:8080/app-
sensor-catalog/Feature-of-Interest/FeatureOfInterest>, sosa:FeatureOfInterest ;
  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> ;
  dcterms:created "2018-05-14T02:33:29.948000+00:00"^^xsd:dateTime ;
  dcterms:identifiier "IP68_sensor"^^xsd:string ;
  dcterms:modified "2018-05-14T02:33:46.156000+00:00"^^xsd:dateTime ;
  schema:identifiier "IP68_sensor"^^xsd:string ;
  rdfs:comment "A Feature of Interest"^^xsd:string ;
  owl:sameAs <http://localhost:8080/Resource/Sensor/ip68_sensor> ;
  prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx7032#provenance> .

```

Figura 153 - Grafo RDF subjacente ao cadastro do objeto de interesse “Sensor IP68” em

Turtle

Fonte: Elaboração própria

*T. Descrição da propriedade de objeto de interesse “Battery of IP68 sensor”*

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Property/battery_of_ip68_sensor?describe> a
<http://www.w3.org/ns/ldp#RDFSSource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Property/battery_of_ip68_sensor> .
<http://localhost:8080/Resource/Property/battery_of_ip68_sensor> a
<http://localhost:8080/Resource/Property/PropertyType/battery>,
    <http://localhost:8080/app-sensor-catalog/Property/Property>,
    ssn:Property ;
    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> ;
    dcterms:created "2018-05-13T23:50:32.759000+00:00"^^xsd:dateTime ;
    dcterms:identifier "Battery_of_ip68_sensor"^^xsd:string ;
    dcterms:modified "2018-05-13T23:50:53.166000+00:00"^^xsd:dateTime ;
    schema:identifier "Battery_of_ip68_sensor"^^xsd:string ;
    rdfs:comment "A Battery property"^^xsd:string ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx3207#provenance> ;
    ssn:isPropertyOf <http://localhost:8080/Resource/FeatureOfInterest/ip68_sensor> .

```

**Figura 154** - Grafo RDF subjacente ao cadastro da propriedade “Bateria do sensor IP68” em

Turtle

Fonte: Elaboração própria

*U. Descrição do sensor “IP68 Smart Sensor”*

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Sensor/ip68_sensor?describe> a
<http://www.w3.org/ns/ldp#RDFSsource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Sensor/ip68_sensor> .

<http://localhost:8080/Resource/Sensor/ip68_sensor> a
<http://localhost:8080/Resource/SensorModel/ip68>,
    <http://localhost:8080/app-sensor-catalog/Sensor/Sensor>,
    sosa:Sensor,
    ssn:System ;
    rdfs:label "IP68 Smart Sensor"^^xsd:string ;
    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> ;
    dcterms:created "2018-05-14T03:01:18.759000+00:00"^^xsd:dateTime ;
    dcterms:identifier "IP68_Sensor"^^xsd:string ;
    dcterms:modified "2018-05-14T03:02:56.040000+00:00"^^xsd:dateTime ;
    dcterms:title "IP68 Smart Sensor"^^xsd:string ;
    schema:identifier "IP68_Sensor"^^xsd:string ;
    schema:location [ a schema:GeoCoordinates ;
        schema:latitude "45.75"^^xsd:decimal ;
        schema:longitude "4.85"^^xsd:decimal ] ;
    schema:name "IP68 Smart Sensor"^^xsd:string ;
    rdfs:comment "A sensor that observes temperature of the air and its battery
state"^^xsd:string ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/14/t1635c986068x197#provenance> ;
    sosa:observes <http://localhost:8080/Resource/Property/battery_of_ip68_sensor>,
        <http://localhost:8080/Resource/Property/temperature_of_the_air> .

```

**Figura 155** - Grafo RDF subjacente ao cadastro do exemplar IP68 em Turtle  
 Fonte: Elaboração própria

## V. Descrição da implantação do sensor IP68 Smart Sensor

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://localhost:8080/Resource/Deployment/ip68_deployment?describe> a
<http://www.w3.org/ns/ldp#RDFSResource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Deployment/ip68_deployment> .

<http://localhost:8080/Resource/Deployment/ip68_deployment> a <http://localhost:8080/app-
sensor-catalog/Deployment/Deployment>,
    ssn:Deployment ;
    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> ;
    dcterms:created "2018-05-14T03:25:23.566000+00:00"^^xsd:dateTime ;
    dcterms:identifier "IP68_Deployment"^^xsd:string ;
    dcterms:modified "2018-05-14T03:26:00.232000+00:00"^^xsd:dateTime ;
    schema:identifier "IP68_Deployment"^^xsd:string ;
    rdfs:comment "A Deployment"^^xsd:string ;
    prov:wasAssociatedWith <http://localhost:8080/Resource/Organization/organization_1> ;
    prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/14/t1635c986068x756#provenance> ;
    ssn:deployedOnPlataform <http://localhost:8080/Resource/FeatureOfInterest/tree_1> ;
    ssn:deployedSystem <http://localhost:8080/Resource/Sensor/ip68_sensor> .

```

**Figura 156** - Grafo RDF subjacente ao cadastro da implantação do sensor IP68 em Turtle

Fonte: Elaboração própria

*W. Descrição do modelo de sensor “Passive Infra-Red Sensor (555-28027)”*

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix ssn-system: <http://www.w3.org/ns/ssn/systems/> .
@prefix unit: <http://qudt.org/1.1/vocab/unit#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/SensorModel/pir?describe> a ldp:RDFSResource ;
  foaf:primaryTopic <http://localhost:8080/Resource/SensorModel/pir> .
<http://localhost:8080/Resource/SensorModel/pir> a <http://localhost:8080/app-sensor-
catalog/Sensor-Model/SensorModel>,
  owl:Class ;
  rdfs:label "Passive infrared sensor"^^xsd:string ;
  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> ;
  dcterms:created "2018-05-14T01:40:55.799000+00:00"^^xsd:dateTime ;
  dcterms:identifier "PIR"^^xsd:string ;
  dcterms:modified "2018-05-14T01:52:32.414000+00:00"^^xsd:dateTime ;
  dcterms:title "Passive infrared sensor"^^xsd:string ;
  schema:identifier "PIR"^^xsd:string ;
  schema:name "Passive infrared sensor"^^xsd:string ;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Property/infrared_signal> ;
      owl:onProperty sosa:observes ],
    [ a owl:Restriction ;
      owl:hasValue <http://localhost:8080/Resource/Organization/parallax> ;
      owl:onProperty dcterms:creator,
        owl:onProperty foaf:maker ],
    [ a owl:Restriction ;
      owl:hasValue [ a ssn-system:OperatingRange ;
        ssn:forProperty
          <http://localhost:8080/Resource/Property/infrared_signal> ;
          ssn-system:hasOperatingProperty [ a schema:PropertyValue,
            ssn-system:OperatingPowerRange,
            ssn-system:OperatingProperty ;
            schema:maxValue "3"^^xsd:integer ;
            schema:minValue "6"^^xsd:integer ;
            schema:unitCode unit:Volt ] ;
          ssn-system:inCondition [ a schema:PropertyValue,
            ssn-system:Condition ;
            rdfs:label "OperatingRangeCondition"^^xsd:string ;
            schema:maxValue "50"^^xsd:integer ;
            schema:minValue "0"^^xsd:integer ;
            schema:unitCode unit:DegreeCelsius ] ] ;
      owl:onProperty ssn-system:hasOperatingRange ] ,
    [ a owl:Restriction ;
      owl:hasValue [ a ssn-system:SystemCapability ;
        ssn:forProperty
          <http://localhost:8080/Resource/Property/infrared_signal> ;
          ssn-system:hasSystemProperty [ a schema:PropertyValue,
            ssn-system:Sensitivity,
            ssn-system:SystemProperty ;
            schema:maxValue "30"^^xsd:integer ;
            schema:minValue "15"^^xsd:integer ;
            schema:unitCode unit:Foot ] ;
          ssn-system:inCondition [ a schema:PropertyValue,
            ssn-system:Condition ;
            rdfs:label "SystemCapabilityCondition"^^xsd:string ;
            schema:maxValue "50"^^xsd:integer ;
            schema:minValue "0"^^xsd:integer ;
            schema:unitCode unit:DegreeCelsius ] ] ;
      owl:onProperty ssn-system:hasSystemCapability ] ,
    sosa:Sensor ,
    ssn:System ;
  prov:wasGeneratedBy
    <http://localhost:8080/callimachus/changes/2018/05/13/t1635b727d0dx6537#provenance> .

```

**Figura 157** - Grafo RDF subjacente ao cadastro do modelo de sensor PIR em Turtle

Fonte: Elaboração própria

X. Descrição do sensor “Passive Infra-Red Sensor (555-28027)”

```

@prefix calli: <http://callimachusproject.org/rdf/2009/framework#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://localhost:8080/Resource/Sensor/pir_sensor?describe> a
<http://www.w3.org/ns/ldp#RDFSResource> ;
    foaf:primaryTopic <http://localhost:8080/Resource/Sensor/pir_sensor> .
<http://localhost:8080/Resource/Sensor/pir_sensor> a
<http://localhost:8080/Resource/SensorModel/pir>, <http://localhost:8080/app-sensor-
catalog/Sensor/Sensor>,
    sosa:Sensor,
    ssn:System ;
rdfs:label "PIR 555-28027"^^xsd:string ;
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> ;
dcterms:created "2018-05-14T03:01:18.759000+00:00"^^xsd:dateTime ;
dcterms:identifier "PIR_Sensor"^^xsd:string ;
dcterms:modified "2018-05-14T03:02:56.040000+00:00"^^xsd:dateTime ;
dcterms:title "PIR 555-28027"^^xsd:string ;
schema:identifier "PIR_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:decimal ;
    schema:longitude "-41.333265"^^xsd:decimal ] ;
schema:name "PIR 555-28027"^^xsd:string ;
rdfs:comment "A sensor that observes infrared signal ;
sosa:observes <http://localhost:8080/Resource/Property/infrared_signal> ;
prov:wasGeneratedBy
<http://localhost:8080/callimachus/changes/2018/05/14/t1635c986068x197#provenance> .

```

**Figura 158** - Grafo RDF subjacente ao cadastro do sensor PIR em Turtle  
 Fonte: Elaboração própria