



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

SAYMON CASTRO DE SOUZA

**UMA ARQUITETURA PARA DESENVOLVIMENTO DE CENÁRIOS EM
INTERNET DAS COISAS**

TESE DE DOUTORADO

VITÓRIA, BRASIL
ABRIL 2021
SAYMON CASTRO DE SOUZA

UMA ARQUITETURA PARA DESENVOLVIMENTO DE CENÁRIOS EM INTERNET DAS COISAS

Proposta de Tese de Doutorado submetida ao Programa de Pós-graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação, sob orientação do Prof. Dr. José Gonçalves Pereira Filho.

Vitória-ES, BRASIL
Abril, 2021

Saymon Castro de Souza

Uma arquitetura para desenvolvimento de cenários em internet
das coisas

Tese submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Em 30 de Abril de 2021.

Prof. Dr. José Gonçalves Pereira Filho
Departamento de Informática - UFES
Orientador

Prof. Dr. Magnos Martinello
Membro Interno

Prof. Dr. Vinícius Fernandes Soares Mota
Membro Interno

Prof. Dr. Pedro Frosi Rosa
Membro Externo

Prof^a. Dr^a. Silvana Rossetto
Membro Externo

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

DEDICATÓRIA

Dedico este trabalho à minha família, aos amigos, ao orientador e a minha esposa por todo apoio necessário para que eu chegasse aqui.

AGRADECIMENTOS

À minha família, que me forneceu a base para que eu conseguisse alcançar meus objetivos. Também gostaria de agradecer aos meus amigos, seja pelo apoio, pelas discussões e pelos momentos de descontração, durante o doutorado.

A todos do Centro de Referência em Formação e em Educação a Distância (Cefor), do Instituto Federal do Espírito Santo (Ifes), que me deram apoio total, sempre incentivando durante essa caminhada.

Aos professores do Departamento de Informática da UFES, pelos ensinamentos. Uma homenagem *in memoriam* ao professor Ricardo Almeida Falbo, que me inspirou no desenvolvimento do primeiro artigo desta pesquisa.

Ao Lucas, Rafael e Nathânia pela importante colaboração nas atividades de pesquisa.

Ao meu orientador, José Gonçalves, pela grande experiência e enorme dedicação que me serviram de exemplo profissional e pessoal.

Especialmente, à Jésica, pela paciência e sua importância em minha vida, desde a graduação, estando junto a mim, compreendendo e apoiando meus estudos.

“A persistência é o caminho do êxito.”

(Charles Chaplin)

ABSTRACT

In recent years, the massive Internet of Things has emerged as an important field of research. Research in this area aims to deal with the challenge of achieving the full convergence of technologies and IT services, in order to facilitate connectivity, accessibility and control of devices in an environment with a huge volume of heterogeneous IoT devices, in order to enable and expanding generic Internet of Things solutions on a global scale. In this new scenario, applications may require a particular arrangement of Things (“Things”) as an important part of their solution. However, the selection of devices that meet the criteria of functionality, capacity and state demanded by the applications, composing a higher level and dynamic structure, capable of defining a particular scenario or even a domain of interest, still demonstrate to be a problem of open search. In this sense, this work proposes the development of an infrastructure for building scenarios in a massive Internet of Things environment. The research includes a systematic study of the literature, the result of which pointed to the use of ontologies in IoT and the design and implementation of a conceptual architecture that supports the generation of a simulation environment, the selection of devices and their composition in scenarios of highest level based on semantic descriptions. The experiments carried out showed that the proposed architecture facilitates the interaction with a massive IoT environment, simplifying the construction of applications and reducing the fragmentation existing in this environment.

Keywords: Internet of Things, Interoperability, Ontology, Device Description Approaches.

RESUMO

Nos anos recentes, a Internet das Coisas massivas (do inglês, “massive IoT”) vem emergindo como um importante campo de pesquisa. As pesquisas nessa área visam lidar com o desafio de atingir a plena convergência de tecnologias e serviços de TI, a fim de facilitar a conectividade, acessibilidade e controle de dispositivos em um ambiente com um enorme volume de dispositivos IoT heterogêneos, com o intuito de viabilizar e expandir soluções genéricas de Internet das Coisas a uma escala global. Neste novo cenário, as aplicações podem demandar um arranjo particular de Coisas (“Things”) como parte importante da sua solução. No entanto, a seleção de dispositivos que atendam aos critérios de funcionalidade, capacidade e estado demandados pelas aplicações, compondo uma estrutura de mais alto nível e dinâmica, capaz de definir um particular cenário ou mesmo um domínio de interesse, ainda demonstram ser um problema de pesquisa em aberto. Nesse sentido, este trabalho propõe o desenvolvimento de uma infraestrutura para construção de cenários em um ambiente de Internet das Coisas massivos. A pesquisa inclui um estudo sistemático da literatura, cujo resultado apontou para o uso de ontologias em IoT e o projeto e a implementação de uma arquitetura conceitual que suporta a geração de um ambiente de simulação, a seleção de dispositivos e a sua composição em cenários de mais alto nível baseada em descrições semânticas. Os experimentos realizados mostraram que a arquitetura proposta facilita a interação com um ambiente de IoT massivos, simplificando a construção de aplicações e diminuindo a fragmentação existente neste ambiente.

Palavras-chave: Internet das Coisas, Interoperabilidade, Ontologia, Abordagens de Descrição de Dispositivos.

Lista de Tabelas

Tabela 1 - Dimensões da integração.....	29
Tabela 2 - String de busca.	51
Tabela 3 - Resultado do processo de seleção (por estágio).	52
Tabela 4 - Etapas do experimento síncrono.....	94
Tabela 5 - Tempo médio de resposta.	95
Tabela 6 - Falhas de requisições ao dispositivo.	96
Tabela 7 - Tempo médio de resposta.....	98
Tabela 8 - Falhas de requisições ao dispositivo.	98
Tabela 9 - Tempo para geração de dispositivos virtuais.	108
Tabela 10 - Consultas realizadas no catálogo implementado.....	114
Tabela 11 - interações registradas no gerador de dispositivos IoT virtuais.....	130

Lista de Figuras

Figura 1 - Design Science Research (HEVNER; CHATTERJEE, 2010).	24
Figura 2 - Dimensões da integração (Fonte: (IZZA, 2009)).....	28
Figura 3 - Modelo de dispositivo DDL (Fonte (CHEN; HELAL, 2009)).....	34
Figura 4 - Um fragmento do esquema DDL que mostra o tipo de dispositivo para consistir em dois elementos: descrição e interface (Fonte (CHEN; HELAL, 2009)). ...	35
Figura 5 - Sensor de temperatura (Fonte: (CHEN; HELAL, 2009)).....	36
Figura 6 - Descrição de coisa baseada em JSON-LD para um LED (Fonte: (DATTA; BONNET, 2016)).	37
Figura 7 - W3C Web of Things - Thing Description (Fonte: (KAEBISCH S., 2019))..	39
Figura 8 - Redes de Petri - elementos básicos.	43
Figura 9 - Rede de Petri.....	43
Figura 10 - Rede de Petri (Fonte: (CARDOSO; VALETTE, 1997)).	44
Figura 11 - Arquitetura genérica para anotação semântica (Fonte: (SHI et al., 2018))..	48
Figura 12 - Apresenta uma distribuição anual dos artigos selecionados.	53
Figura 13 - Distribuição dos estudos selecionados com foco em ontologias de fundamentação.....	54
Figura 14 - Linguagens de representação de ontologias em abordagens em IoT.....	55
Figura 15 - Tipos de ontologias conforme seu nível de generalização.	56
Figura 16 - Linguagens e sua adoção por nível de generalização das ontologias.	56
Figura 17 – Abordagens metodológicas usadas no desenvolvimento de ontologias em IoT.	57
Figura 18 - Estudos que reutilizam ontologias existentes em suas propostas.	58
Figura 19 – Ontologias reutilizadas em abordagens IoT.....	59
Figura 20 - Infraestrutura SI4-IoT.....	68
Figura 21 - Modelo de informação.....	70
Figura 22 - Visão geral da arquitetura proposta.	72
Figura 23 - Catálogo.....	75
Figura 24 - Componentes da Camada Scenario.....	76
Figura 25 - Rede de Petri representando um cenário.....	79
Figura 26 - Arquitetura de implementação.....	83
Figura 27 - Dispositivo utilizado na implementação da especificação do W3C <i>Thing Description</i>	85
Figura 28 - W3C WoT - Thing Description.	86
Figura 29 - Fragmento do WoT Thing Description - Action.	87
Figura 30 - Interaction Affordance - Action.....	87
Figura 31 - Fragmento do Thing Description – property.	88
Figura 32 - Interaction Affordance - Properties	89
Figura 33 - Interaction Affordance – Properties with two sensors.....	90
Figura 34 - Fragmento do WoT Thing Description - Event.....	90
Figura 35 - Diagrama de sequência para descrição de um evento.....	92
Figura 36 - Avaliação do protótipo.....	93
Figura 37 – Implementação do Virtual Thing Generator.	100
Figura 38 - Gerador de dispositivos de Internet das Coisas	100
Figura 39 - Lista de dispositivos gerados hospedados na Amazon AWS.	103
Figura 40 - Descrição de um dispositivo e suas interações.	104
Figura 41 - Propriedades do dispositivo IoT virtual gerado.....	105
Figura 42 - Detalhes da propriedade oxigênio do dispositivo IoT virtual gerado.....	106

Figura 43 - Diagrama de sequência da interação com a propriedade oxigênio do dispositivo IoT virtual gerado.....	106
Figura 44 - Detalhes do atuador <i>buzzer</i> do dispositivo IoT virtual gerado.	107
Figura 45 - Interação do tipo ação - <i>buzzer</i>	108
Figura 46 - Teste de requisições da propriedade oxigênio de um dispositivo virtual. .	110
Figura 47 - Catálogo armazenando WoT Thing Description.	111
Figura 48 - Catálogo.	111
Figura 49 - Persistência dos <i>Thing Descriptions</i> no Catálogo.....	113
Figura 50 - Implementação da camada Scenario.	116
Figura 51 - Cenário C ₁	121
Figura 52 - Cenário C ₂	124
Figura 53 - Cenário C ₃	127
Figura 54 - Infraestrutura para provisão de um cenário massivo de dispositivos IoT virtuais	132
Figura 55 - Dispositivos IoT Virtuais hospedados na infraestrutura do Ifes.	133

Sumário

1. INTRODUÇÃO.....	15
1.1. MOTIVAÇÃO.....	15
1.2. DESAFIOS DO DOMÍNIO.....	20
1.3. PRINCÍPIOS E HIPÓTESES DE PESQUISA	22
1.4. OBJETIVOS DO TRABALHO	23
1.5. ASPECTOS METODOLÓGICOS	24
1.6. ORGANIZAÇÃO DA TESE.....	25
2. REFERENCIAL TEÓRICO.....	27
2.1. INTEROPERABILIDADE.....	27
2.2. ONTOLOGIA.....	30
2.3. DESCRIÇÃO DE DISPOSITIVOS IoT.....	32
2.3.1. ABORDAGEM DDL - DEVICE DESCRIPTION LANGUAGE	33
2.3.2. ABORDAGEM CORE LINK FORMAT	36
2.3.3. WEB OF THINGS – THING DESCRIPTION	37
2.4. ONTOLOGIAS NA DESCRIÇÃO SEMÂNTICA DE DISPOSITIVOS IoT.....	40
2.5. CONTEXTO E SITUAÇÃO	41
2.6. REDES DE PETRI.....	43
2.7. CONSIDERAÇÕES SOBRE O CAPÍTULO	46
3. INTEROPERABILIDADE EM INTERNET DAS COISAS.....	47
3.1. METODOLOGIA.....	47
3.2. TRABALHOS SELECIONADOS	51
3.3. SÍNTESE DE DADOS E RESULTADOS	53
3.4. DISCUSSÃO.....	59
4. A ARQUITETURA PROPOSTA.....	65
4.1. PRINCÍPIOS E REQUISITOS	65
4.2. ARQUITETURA CONCEITUAL.....	67
4.2.1. CAMADA <i>THING</i>	68
4.2.2. <i>VIRTUAL THING GENERATOR</i>	70
4.2.3. CAMADA CATÁLOGO	74
4.2.4. CAMADA SCENARIO.....	76
4.2.4.1. DEFININDO CENÁRIOS.....	77
4.3. CONSIDERAÇÕES DO CAPÍTULO	81
5. IMPLEMENTAÇÃO DA ARQUITETURA SI4-IOT	82
5.1. ARQUITETURA DE IMPLEMENTAÇÃO	82
5.2. REALIZAÇÃO DA CAMADA <i>THING</i>	84
5.2.1. EXPERIMENTOS DE AVALIAÇÃO.....	92
5.2.1.1. EXPERIMENTO 1: APLICAÇÕES SÍNCRONAS	94
5.2.1.2. EXPERIMENTO 2: APLICAÇÕES ASSÍNCRONAS	97
5.2.1.3. CONSIDERAÇÕES SOBRE OS EXPERIMENTOS 1 E 2	99
5.3. REALIZAÇÃO DA CAMADA DE GERAÇÃO MASSIVA DE DISPOSITIVOS IoT VIRTUAIS	99
5.3.1. EXPERIMENTOS DE AVALIAÇÃO.....	101
5.3.1.1. EXPERIMENTO 1: IMPLANTAÇÃO MASSIVA DE DISPOSITIVOS IoT EM UMA INFRAESTRUTURA DE NUVEM COMPUTACIONAL	102
5.3.1.2. EXPERIMENTO 2: MEDIÇÃO DE DESEMPENHO DA GERAÇÃO DE DISPOSITIVOS IoT VIRTUAIS MASSIVOS	108
5.3.1.3. EXPERIMENTO 3: MEDIÇÃO DE DESEMPENHO DE UM DISPOSITIVO IoT VIRTUAL	109
5.3.2. CONSIDERAÇÕES DOS EXPERIMENTOS 1, 2 E 3	110
5.4. REALIZAÇÃO DA CAMADA CATÁLOGO	111
5.4.1. EXPERIMENTOS DE AVALIAÇÃO	112
5.4.1.1. EXPERIMENTO 1: INCLUSÃO MASSIVA DE <i>THING DESCRIPTIONS</i>	113

5.4.1.2.	EXPERIMENTO 2: BUSCA POR DISPOSITIVOS IoT	114
5.4.2.	CONSIDERAÇÕES DOS EXPERIMENTOS 1 E 2.....	115
5.5.	REALIZAÇÃO DA CAMADA COMPOSIÇÃO DE CENÁRIOS	115
5.5.1.	DEFININDO ESTADOS DE INTERESSE	117
6.	ESTUDO DE CASO	119
6.1.	ESTUDO DE CASO – CONTRAMEDIDAS DA POLUIÇÃO DO AR – NA PERSPECTIVA DE UMA CIDADE 119	
6.2.	FORMALIZAÇÃO DO CENÁRIO - CONTRAMEDIDAS DA POLUIÇÃO DO AR	121
6.2.1.	C ₁ - CARROS EM TRÂNSITO.....	121
6.2.2.	C ₂ - ÁREAS COM CONCENTRAÇÃO DE POLUENTES.....	124
6.2.3.	C ₃ - VAGAS DE ESTACIONAMENTO	127
6.3.	INFRAESTRUTURA MASSIVA DE DISPOSITIVOS IoT	130
6.4.	DISCUSSÃO.....	134
7.	CONCLUSÕES	136
7.1.	CONSIDERAÇÕES FINAIS.....	136
7.2.	RESULTADOS ACADÊMICOS	138
7.3.	TRABALHOS FUTUROS	139

Capítulo 1

1. Introdução

Este capítulo apresenta uma visão geral desta tese de doutorado. São apresentados o contexto em que o trabalho está imerso, os problemas e justificativas que o motivaram, as hipóteses de pesquisa, os objetivos do trabalho e as atividades necessárias para alcançá-los, bem como o escopo e a estrutura do documento.

1.1. Motivação

A iniciativa de embarcar inteligência nos objetos do dia-a-dia e posteriormente integrá-los à Internet, inaugura uma nova fase do desenvolvimento da Web, em que a principal promessa é a realização da Internet das Coisas (Internet of Things – IoT). Conceito originado em 2000 no Auto-ID Center do MIT (SARMA; BROCK; ASHTON, 2000) a visão de objetos inteligentes permeando o dia-a-dia das pessoas já era vislumbrado por Mark Weiser, considerado o pai da Computação Ubíqua, em seu famoso artigo “*The Computer of 21th Century*”(WEISER, 1991).

A IoT é um novo conceito de Internet que caracteriza a visão de uma rede global formada por objetos inteligentes (*smart objects*) interconectados e endereçáveis, a partir do uso e adaptação de protocolos e padrões de comunicação bem definidos e conhecidos da Internet atual. O Centro de Pesquisa Europeu sobre Internet das Coisas (*Internet European Research Center – IERC*), uma organização de destaque na área de IoT, definiu Internet das Coisas como uma rede ou associações entre objetos e dispositivos inteligentes ligados à Internet, que são capazes de trocar informações usando um método e esquema de dados acordados (M. SERRANO, P. BARNAGHI, F. CARREZ, P. COUSIN; FRIESS, 2015).

Neste novo cenário de Internet do Futuro, objetos do mundo real – ou seja, coisas (*things*) do mundo físico, tais como ferramentas, eletrodomésticos, itens pessoais e de vestuário etc – são embarcados com capacidades de processamento, armazenamento, sensoriamento e comunicação sem fio e integrados à Internet, sendo vistos pelas

aplicações como entidades autônomas, com comportamento pró-ativo e conhecimento sobre o contexto circundante, além de capacidade de colaboração para alcançar uma meta comum. A ideia básica é disponibilizar o estado das coisas que formam o mundo físico para os aplicativos, tornando-os cientes, em tempo real, das mudanças observadas no mundo físico e, conseqüentemente, permitindo promover adaptações e respostas mais rápidas a esses aplicativos.

Na IoT, existe uma clara intenção de se diminuir a distância entre o objeto e a sua representação no mundo digital, com o intuito de tornar mais simples a interação com o mundo físico, da mesma forma que atualmente é experimentada a interação com objetos do mundo virtual. A ideia básica é integrar (mover) o estado das coisas que formam o mundo concreto para dentro das aplicações, tornando-as cientes, em tempo real, das alterações de estado observadas no mundo físico e podendo, com isso, promover uma adaptação e uma resposta mais rápida às mudanças. Desta maneira, quando objetos do mundo real passam a ser “percebidos” de forma ubíqua, a “sentir” a presença de outros objetos e a se comunicar entre si, com os usuários e outros sistemas computacionais, altera-se a maneira como, onde e quem toma as decisões sobre o mundo físico.

Esta visão de IoT desempenha um papel central na arquitetura da Internet do Futuro, uma vez que acarreta uma expansão extraordinária da Internet atual devido ao número imenso de dispositivos não convencionais, de alto grau de heterogeneidade, que ingressarão na rede. Novos serviços e aplicações se aproveitarão desta massa de dados e informações de contexto originadas destes objetos inteligentes para oferecer aos usuários finais diferentes formas de diálogo com o mundo físico, em distintos domínios, tais como transporte e logística, comércio, saúde, setor automotivo, automação industrial, cidades inteligentes, entre outros (BARNAGHI et al., 2012; KRANENBURG et al., 2011)

Pesquisas na área de Internet das Coisas vem sendo apoiada por esforços de colaboração da academia, indústria e organismos de normatização em várias comunidades. A iniciativa em curso na construção da Internet das Coisas é suportada, principalmente, pelo progresso contínuo dos circuitos de hardware e plataformas de sensores, que permitem acuradamente sensorear, processar e transmitir dados inteligentes, bem como pela criação de infraestruturas de software voltados para a captura, comunicação, armazenamento, acesso e compartilhamento dos dados obtidos do mundo físico (BARNAGHI et al., 2012). Entretanto, a Internet das Coisas está longe de atingir a plena convergência de tecnologias e serviços de TI. Desafios tecnológicos relacionados à heterogeneidade de hardware de IoT, segurança dos dados, conectividade

e modelos de comunicação, ausência de padrões, processamento e análise de volumes imensos de dados em tempo real, dentre outros, constituem um desafio à expansão de soluções genéricas de Internet das Coisas a uma escala global (WANG et al., 2015).

Nos anos recentes, a fragmentação, a heterogeneidade e a dinamicidade do domínio da IoT têm atraído mais atenção da comunidade de IoT da academia e da indústria, que estão direcionando esforços de pesquisa para uma questão fundamental: interoperabilidade semântica em IoT. De acordo com o IERC Report “*IoT Semantic Interoperability: Research Challenges, Best Practices, Recommendations and Next Steps*” (SERRANO et al., 2015), a interoperabilidade semântica é o próximo passo na consolidação da Internet das Coisas e ela já vem acontecendo. Segundo (THULUVA; ANICIC; RUDOLPH, 2018), a promessa da IoT em termos de interoperabilidade ainda não foi entregue, sendo os esforços atuais de pesquisa direcionados a conectar dispositivos heterogêneos a uma plataforma comum. Uma vez conectados, os dispositivos são capazes de trocar dados; porém, a semântica dos dados e as características desses dispositivos não são descritos em uma forma interpretável por máquina. Esse conjunto de fatores dificulta o potencial da IoT a fim de permitir o desenvolvimento de serviços de valor agregado em larga escala.

Uma abordagem promissora em contextos onde a interoperabilidade é um requisito fundamental é adotar de *Ontologias* (GRUBER, 1995), (CALHAU; DE ALMEIDA FALBO, 2010) como tecnologia base de descrição semântica. Ontologias vem sendo usadas na área de Engenharia de Software para enriquecer o modelo de informação e fornecer expressividade semântica à informação, podendo apoiar o intercâmbio entre aplicações, serviços e sistemas em diferentes níveis de abstração (NARDI; DE ALMEIDA FALBO; ALMEIDA, 2013). Do mesmo modo, no domínio da Internet das Coisas, também tem crescido o interesse na adoção de ontologias como um importante aliado para abordar problemas relacionados à interoperabilidade semântica, permitindo o compartilhamento de vocabulários comuns e a representação interoperável de dados, possibilitando a criação de estruturas de IoT em larga escala (BERMUDEZ-EDO et al., 2016a).

Outra questão relevante está relacionada aos esforços recentes de pesquisa com foco em conectar dispositivos heterogêneos a uma plataforma comum, com objetivo de atenuar a conhecida fragmentação existente neste ambiente. Iniciativas como (CHARPENAY; KÄBISCH; KOSCH, 2016; CHEN; HELAL, 2009; KHALED et al., 2018) tem se dedicado à proposição de formas de expor dispositivos, de maneira

padronizada, por meio de abordagens de descrição de dispositivos com intuito de facilitar o reconhecimento das capacidades de um dispositivo. Associado a essas abordagens estão os mecanismos padronizados de interação, nos quais são definidos um conjunto de protocolos de comunicação e apresentação de como interagir com as funcionalidades disponíveis em um dispositivo, sendo crucial que tal compreensão seja interpretável por máquinas. Neste contexto, a iniciativa do W3C *Web of Things* (KAEBISCH S., 2019) tem se destacado como uma abordagem promissora, embora ainda pouco explorada na literatura em termos de implementação e estudos de caso. Por exemplo, em (DATTA; BONNET, 2018; KÄBISCH; ANICIC, [s.d.]) é proposta uma linguagem para descrição de dispositivos – *Thing Description*, que é capaz de explicitar as características de uma “coisa” incluindo: seu nome; descrição; contexto (associação à conceitos ontológicos); suas interações (propriedades, ações e eventos) e a forma pelo qual uma máquina possa interagir, incluindo os respectivos protocolos; o formato de dados que o dispositivo pode retornar; aspectos de segurança, entre outros.

Recentemente, (LE et al., 2019) inaugurou uma discussão a respeito de interações em um cenário de Internet das Coisas massivos onde foi proposta uma linguagem para descrição de grupos de dispositivos, na qual é possível criar um nível de abstração mais elevado, denominado *asset*. No entanto, tal abordagem apresenta limitações quanto à maneira de como lidar com um conjunto massivo de dispositivos, dentre as quais, é possível destacar: (i) não há uma infraestrutura capaz de suportar a seleção de dispositivos que atendem aos critérios de requisitos funcionais em relação a um determinado domínio de aplicação, tornando os dispositivos acessíveis aos usuários, ou seja, um ambiente que permita o registro, compartilhamento e possibilite processar operações de pesquisa e (ii) não é possível definir regras de negócio associados ao domínio em questão que considerem o estado dos dispositivos.

Em relação a (i), considerando a existência de um cenário de disponibilidade massiva de dispositivos de Internet das Coisas, se torna crucial prover um ambiente capaz de permitir a seleção de dispositivos que atendam aos requisitos do domínio. Neste aspecto, é importante diferenciar as características físicas do dispositivo (sensores, atuadores, etc) e seu estado – os valores providos pelos elementos presentes no dispositivo. O uso de tecnologias de descrição semântica surge, portanto, como uma ferramenta fundamental para a de anotação semântica de dispositivos IoT, visto que podem facilitar o processo de seleção em um universo abrangente. A seleção destes dispositivos também é um fator essencial, dado o cenário dinâmico de provisão de novas

“coisas”. Em outras palavras, a todo instante novos dispositivos são conectados à Internet e, eventualmente, podem surgir elementos que atendam aos requisitos de um determinado domínio de aplicação.

Em relação a (ii) uma das estratégias para abordar o problema é promover o que na comunidade de engenharia de software é conhecido por “separação das preocupações” (“*separation of concerns*”) (MITCHELL, 1990), considerando, no presente caso, duas visões: a primeira se refere ao processo de seleção de dispositivos conforme suas características físicas, tendo entendimento dos detalhes referentes aos sensores, atuadores e eventos providos; a segunda diz respeito especificamente às regras e eventos do domínio. Essa separação de “*concerns*” contribui para diferenciar claramente os papéis do usuário programador e do usuário analista do domínio, o que poderia potencializar a participação e a melhoria da interatividade do usuário do domínio na concepção final do cenário. Nesta abordagem, profissionais experientes atuam na seleção de dispositivos, considerando conceitos ontológicos associados à composição física, que devem auxiliar num processo de seleção preliminar e, em seguida, analisar as especificidades da arquitetura de hardware do dispositivo, seus sensores e/ou atuadores. A definição das regras e eventos do domínio de aplicação, por outro lado, passaria a ser uma preocupação do usuário do domínio, que tem um conhecimento muito mais especializado das regras de negócio. Evidentemente, é importante que as regras possam ser facilmente descritas e compreendidas pelos analistas de domínio. Em função disso, a linguagem usada para a descrição de regras deve possuir flexibilidade para alteração e construção de novas regras e eventos, pois essas características do domínio geralmente são dinâmicas. A relevância de possibilitar a descrição de regras do domínio que sejam capazes de avaliar o estado do dispositivo, se tornam um nível complementar ao processo de seleção de dispositivos e viabilizam a concepção de um cenário.

Considerando o suporte atual ao desenvolvimento de aplicativos em ambiente de Internet das Coisas massivos (HEJAZI et al., 2018), tendo em vista a importância da interoperabilidade semântica e ancorado nos resultados apresentados em (DE SOUZA; PEREIRA FILHO, 2019), pode-se constatar que existe uma lacuna na literatura científica para soluções que suportem a seleção de dispositivos de IoT massivos e que permitam viabilizar o desenvolvimento de cenários que possam ser utilizados em diferentes contextos. Especificamente, nota-se uma carência de soluções que permitam - de uma maneira simples e flexível - incluir novos requisitos ou fazer alterações em tempo real

para lidar com novas situações contextuais ou eventos que podem ocorrer a qualquer momento.

1.2. Desafios do domínio

As questões colocadas na Seção 1.1 indicam que a realização plena da Internet das Coisas e a sua adoção como uma infraestrutura de massa, acessível a todos os setores da sociedade, vão demandar a superação de uma série de desafios tecnológicos e de pesquisa. Gestão e armazenamento de um volume gigantesco de dados, segurança dos dados, garantia de privacidade da informação, implantação de infraestruturas adequadas de Data Centers, além de questões legais e regulatórias ligadas à construção e uso de equipamentos IoT, e.g., equipamentos e dispositivos médico-hospitalares, bem como dificuldades de padronização, são exemplos de obstáculos a serem superados. A escala apresentada pelos ambientes IoT massivos eleva, portanto, estes desafios a um outro patamar, exigindo preocupações adicionais com eficiência operacional, confiabilidade e disponibilidade. Todavia, como destacado, reproduzir o cenário real com vistas aos testes das aplicações IoT massivas, pode ser uma tarefa árdua e custosa, dada a diversidade e complexidade do ambiente concreto de operação.

Projetos de implantação de cidades inteligentes são exemplos de iniciativas que dão a dimensão do enorme esforço e investimento que são necessários para a provisão massiva de dispositivos físicos inteligentes e a implantação da infraestrutura requerida para a sua operação plena. Uma amostra nesta direção é o projeto SmartSantander (SANCHEZ et al., 2014), um projeto de pesquisa que propõe uma plataforma de experimentação de aplicações e serviços típicos para uma cidade inteligente, incluindo experimentos sobre novas tecnologias IoT e avaliação e testes de aceitabilidade dos usuários. Neste projeto foram implantados mais de 20.000 nós sensores, com um orçamento total de mais de oito milhões de euros.

Um outro exemplo de destaque no mesmo domínio é o projeto CityPulse (PRESSER; VESTERGAARD; GANEA, 2014). O projeto foi realizado com a meta de coletar cenários de potenciais aplicações IoT visando obter a maior diversidade possível de ideias para discussão com as partes interessadas da cidade e com a comunidade em geral. O levantamento realizado no projeto revelou a dimensão dos eventuais novos negócios que podem emergir do usufruto de uma infraestrutura existente de Internet das Coisas massiva. Foi também constatado pelo projeto que o desenvolvimento de potenciais novos

negócios está limitado pela disponibilidade de infraestrutura massiva de dispositivos IoT. Os resultados colhidos demonstraram a necessidade de ambientes de simulação e avaliação experimental como elementos estratégicos no processo de concepção das novas aplicações IoT massivas, por possibilitarem que as soluções possam ser testadas em uma escala estendida. Por exemplo, considere neste domínio um cenário de gerenciamento de mobilidade urbana em que seja necessário desenvolver uma aplicação IoT massiva que consiga definir rotas de um ponto A até um ponto B analisando aspectos de tempo, custo, disponibilidade, meio ambiente, entre outros aspectos, considerando todos os modais de transporte (aéreo, ferroviário, aquaviário, rodoviário) e, adicionalmente, incluindo os diversos fornecedores de serviço de transporte de última milha, como aluguel de bicicletas e patinetes. Em tal cenário, um olhar dedicado a um certo dispositivo, como um particular indivíduo de uma grande infraestrutura de IoT, pode se fazer necessário com o objetivo de extrair toda a potencialidade do mesmo para uso em inúmeros contextos, propiciando tomadas de decisão complexas envolvendo diversas variáveis. No entanto, especificidades individuais podem impor restrições quanto ao desenvolvimento de novos serviços que buscam utilizar um conjunto heterogêneo de dispositivos como parte de sua solução. Nesse sentido, é necessário existir mecanismos padronizados de descrição e acesso que facilitem a interação com os componentes da infraestrutura de IoT, bem como promovam a compreensão semântica das características individuais de cada elemento.

Neste ambiente dinâmico e complexo a proposta ora apresentada nesta tese busca suprir três importantes demandas do cenário de IoT massivo, que podem ser traduzidas nos seguintes desafios de domínio (DD):

- DD₁ - implantação de infraestruturas de testes que permitam simular, de maneira simples e rápida, cenários diversos compostos por uma quantidade expressiva de fontes contextuais, possivelmente heterogêneas, gerando dispositivos adaptáveis a cenários particulares idealizados pelos desenvolvedores de soluções IoT. Tais infraestruturas devem viabilizar a provisão de uma extensa lista de dispositivo que representem o domínio de aplicação, de forma simplificada e com baixo custo.
- DD₂ - interação individual e de forma homogênea com dispositivos IoT das mais diversas tecnologias, de modo a abstrair as especificidades de implementação de cada dispositivo em termos de seus componentes (sensores e atuadores), tecnologias de hardware e software básico embarcado. No caso de cidades inteligentes, por exemplo, onde a

integração de dados é fundamental, pode haver a necessidade de interagir com inúmeros dispositivos IoT, com características de hardware e funcionalidades completamente diversas, a fim de atender aos propósitos das aplicações. Implantar, portanto, serviços e aplicações que explorem as potencialidades que se abrem a partir da composição de múltiplas e diferentes fontes de dados pode ser desafiador, haja vista a necessidade de compreender a especificidade de cada dispositivo.

- DD₃ - Suporte a seleção de dispositivos IoT em um ambiente com disponibilidade extensiva e que permitam analisar as propriedades físicas e contextuais, a fim de viabilizar o desenvolvimento de cenários que possam ser utilizados em diferentes contextos. No caso de cidades inteligentes, como visto, onde a diversidade de especificações de dispositivos IoT é inerentemente ampla e heterogênea, há um enorme potencial retido pela dificuldade de escolher um subconjunto de dispositivos de interesse que atendem aos diversos cenários que podem ser explorados por novas aplicações e negócios.

A arquitetura proposta nesta tese procura enfrentar esses três (3) desafios, promovendo a flexibilidade necessária para atender às demandas de aplicações de IoT do mundo real. De fato, o aumento do interesse em aplicativos nos cenários mencionados de IoT exigirá soluções capazes de lidar com novos cenários que podem mudar a qualquer momento. Essas soluções devem fornecer recursos como portabilidade, escalabilidade e capacidade de desacoplamento, a fim de oferecer suporte a um ambiente de negócios de TI complexo e heterogêneo, composto por uma ampla variedade de tecnologias integradas aos sistemas corporativos. Isso torna as soluções de suporte, como a proposta nesta tese, um componente estratégico para a implantação de aplicações de IoT.

1.3. Princípios e Hipóteses de Pesquisa

Baseado nos resultados da seção anterior, alguns princípios básicos foram definidos e categorizados em três dimensões, conforme descrito abaixo:

Dimensão teórica: A descrição conceitual e funcional de uma arquitetura facilita o entendimento e, especialmente, a dissociação entre a função de cada camada com seus respectivos módulos, elementos e as escolhas tecnológicas que podem ser feitas no momento da implementação, de acordo com as necessidades específicas;

Dimensão da prática: A definição de uma arquitetura de implementação e o desenvolvimento de estudos de caso que representem cenários de IoT, através de experimentos, permite que a arquitetura conceitual seja validada e atua como uma prova de conceito que pode ser implementada de acordo com escolhas tecnológicas ou interesses específicos.

Dimensão da relevância: O resultado final do trabalho proposto deve ser útil, relevante, coerente e compatível com os mais variados ambientes de Internet das Coisas. Assim, a arquitetura proposta deve apoiar o desenvolvimento de novas aplicações cujo modelo de negócio se beneficiam do uso de informações oriundas de dispositivos IoT.

Baseado no contexto, motivação, desafios do domínio e nos princípios aqui definidos, a hipótese de pesquisa desta tese é assim estabelecida:

Uma arquitetura de IoT de código aberto, equipada com recursos selecionados, como os propostos neste trabalho de pesquisa, pode contribuir para superar os desafios da criação de aplicações finais em um contexto de disponibilidade massiva de dispositivos IoT heterogêneos.

1.4. Objetivos do Trabalho

Existem três pilares que norteiam os objetivos desta pesquisa de doutorado: o suporte conceitual (teoria), a implantação (tecnologia) e os experimentos (prática). O Objetivo Geral (OG) do trabalho é conceber uma infraestrutura de software que suporte a seleção de dispositivos no contexto de Internet das Coisas massivos e a composição de cenários para prover um maior nível de abstração para aplicações finais.

Com base neste OG, os seguintes Objetivos Específicos (OE) são estabelecidos:

OE1: Especificar uma arquitetura de referência cuja organização conceitual (conjunto de elementos funcionais e suas interfaces de comunicação) operem de maneira integrada para prover a composição de cenários de maior nível de abstração e seleção de dispositivos IoT;

OE2: Especificar uma arquitetura de implementação com base na arquitetura conceitual definida em OE1 que seja apoiada em escolhas tecnológicas que privilegiem o uso de padrões em IoT, necessários para lidar com conjuntos heterogêneos de dispositivos;

OE3: Validar a arquitetura proposta através do desenvolvimento de: (i) experimentos reais e simuladas que permitam avaliar a capacidade da arquitetura de suportar cenários diversos de dispositivos massivos e heterogêneos; e (ii) elaboração de estudo de caso que demonstre a construção de cenários em diferentes domínios no contexto de IoT.

1.5. Aspectos metodológicos

O suporte metodológico para a construção desta tese foi baseado na teoria e prática da *Design Science Research* (DSR) apresentada em (HEVNER; CHATTERJEE, 2010). O DSR é um paradigma de pesquisa que visa compreender e explicar um problema relevante de design e sua solução, a fim de criar um artefato de Tecnologia da Informação e Comunicação inovador e útil que atendam aos requisitos funcionais de um domínio. A compreensão de um *problema de design* está relacionado ao conhecimento que permite prever como um fenômeno se comporta. Três ciclos DSR são definidos para conduzir qualquer projeto relacionado a um *problema de design*. A Figura 1 apresenta uma visão geral dos três ciclos de DSR com a indicação das atividades de trabalho da tese planejadas para cada ciclo.

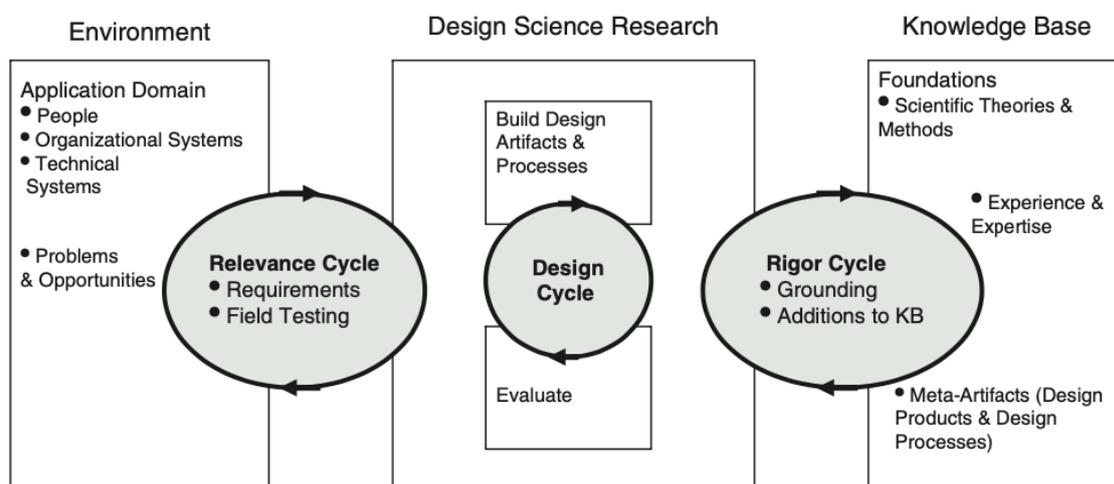


Figura 1 - Design Science Research (HEVNER; CHATTERJEE, 2010).

O *Ciclo de Relevância* conecta o ambiente contextual às atividades da ciência do design e define os critérios para a avaliação dos resultados. O aspecto principal deste ciclo é a identificação e caracterização de problemas e oportunidades no ambiente de problemas do domínio. O *Ciclo Rigor* é fundamentado na literatura do domínio e deve

adicionar novos conhecimentos por meio dos resultados obtidos. O rigor está relacionado à escolha e aplicação de métodos e teorias apropriados para a criação e avaliação dos artefatos produzidos. O *Ciclo de Design* é o elemento central de qualquer projeto DSR executando o trabalho mais fundamental e difícil. Um trabalho consistente neste ciclo requer bons argumentos, com uma boa avaliação deles com base nos requisitos (entradas do ciclo de relevância) até atingir o resultado desejado. Os ciclos DSR definidos para a realização desta tese serão apresentados a seguir.

O *Ciclo de Relevância* - o ambiente, o problema do domínio e os requisitos foram apresentados nas seções anteriores. Essas seções ajudaram a definir desafios de pesquisa e contribuições desta tese.

O *Ciclo Rigor* é coberto pela análise e os resultados apresentados em (DE SOUZA; PEREIRA FILHO, 2019) e também no capítulo 3. Os resultados contribuíram para apoiar a especificação das funcionalidades da arquitetura proposta de acordo com os objetivos da tese.

O *Ciclo de Design* foi realizado e a análise e os resultados desse ciclo serão apresentados nos capítulos 4, 5 e 6.

1.6. Organização da Tese

O restante desta Tese está assim organizada:

- **Capítulo 2. – Fundamentação teórica:** apresenta conceitos de interoperabilidade, ontologias e linguagens de descrição de dispositivos e escolhas tecnológicas importantes para o entendimento da infraestrutura proposta.
- **Capítulo 3. – Interoperabilidade semântica em Internet das Coisas.** apresenta os resultados do estudo de mapeamento sistemático que investiga o uso de ontologias na descrição semântica de dispositivos em aplicações no contexto de Internet das Coisas.
- **Capítulo 4. – A Arquitetura conceitual:** apresenta a arquitetura de referência cuja organização conceitual e conjunto de elementos funcionais operem de maneira integrada para prover a composição de cenários em Internet das Coisas massivos.
- **Capítulo 5. – Implementação da arquitetura:** apresenta detalhes da implementação, tendo por base a arquitetura conceitual, indicando todas as

escolhas tecnológicas e incluindo os ambientes e padrões necessários para lidar com conjunto massivo de dispositivos no contexto de Internet das Coisas.

- **Capítulo 6. – Estudo de Caso:** apresenta um estudo de caso demonstrando o desenvolvimento de cenários em domínios que evidenciem a capacidade da arquitetura proposta.
- **Capítulo 7. – Considerações finais:** apresenta as considerações finais, as contribuições da pesquisa, as implicações e conexões com outros trabalhos e as perspectivas futuras que apontam onde novos estudos podem ser desenvolvidos.

Capítulo 2

2. Referencial Teórico

Este trabalho explora diversas áreas de pesquisa atuais, como interoperabilidade semântica e estratégias para descrição de dispositivos IoT. Para um melhor entendimento das contribuições do trabalho, este capítulo realiza uma revisão dos conceitos básicos dessas áreas de pesquisa. O capítulo está organizado da seguinte forma: a Seção 2.1 introduz o conceito de “Interoperabilidade”, apresentando as principais definições da literatura e suas diferentes dimensões; a Seção 2.2 discute conceitos relacionados à “Ontologias”, incluindo classificações, linguagens de representação, metodologias e sua adoção como forma de promover a Interoperabilidade; a Seção 2.3 descreve as abordagens para descrição de dispositivos IoT, apresentando as diferentes propostas para lidar com a maneira de expor as características e funcionalidades de um dispositivos IoT; a Seção 2.4 traz uma discussão sobre o uso de Ontologias na descrição semântica de dispositivos IoT; a Seção 2.5 apresenta uma breve discussão acerca dos conceitos de “contexto” e “situação” sendo importantes para a arquitetura proposta; a Seção 2.6 introduz Redes de Petri, seus principais elementos e classificações; e, finalmente, a Seção 2.7 apresenta as considerações finais do capítulo.

2.1. Interoperabilidade

A interoperabilidade tem sido um importante tema de pesquisa na área de computação há anos. Em (SHETH, 1999) é apresentado o contexto histórico de pesquisas desta área, classificando-a em três gerações:

1. Nos anos 80, quando o foco das pesquisas era promover a interoperabilidade de banco de dados departamentais de uma organização;

2. Na década de 90, que trouxe oportunidades de pesquisa em decorrência de dois motivos: (i) a proliferação da variedade de dados estruturados e não estruturados; e (ii) a disseminação da Internet e surgimento da Web;
3. A terceira geração, que lidou com o progresso da interconectividade global e informações mais heterogêneas, consistindo não apenas de uma variedade mais ampla de dados digitais, mas também de operações, tais como, raciocínio, inferência, predição que podem criar novos dados e informações.

Interoperabilidade, em uma visão mais ampla, pode ser definida como a habilidade de operar em conjunto ou, mais especificamente no contexto de sistemas computacionais, como a capacidade de trocar dados e serviços entre aplicativos ou componentes de aplicativos (WEGNER, 1996). Segundo (IZZA, 2009), há diversas dimensões de integração que contribuem para caracterizar a noção de interoperabilidade, a saber, nível, escopo, camada e ponto de vista. A figura abaixo ilustra a dimensões defendidas por (IZZA, 2009).

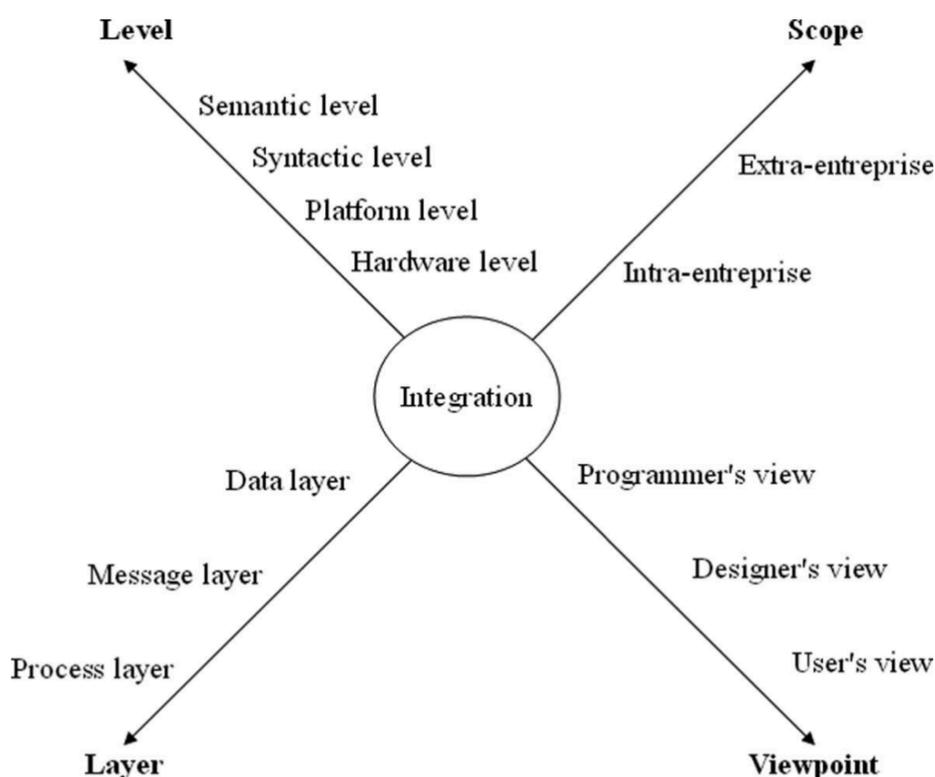


Figura 2 - Dimensões da integração (Fonte: (IZZA, 2009))

A tabela 1 apresenta a visão do autor a respeito das dimensões que contribuem para a concepção da interoperabilidade em uma organização.

Tabela 1 - Dimensões da integração

Dimensões da integração	
<i>Scope</i>	A integração pode ocorrer dentro da organização, envolvendo apenas aplicações internas da mesma, ou entre organizações, conectando aplicações de diferentes parceiros.
<i>Viewpoint</i>	A integração é distinguida pelo o ponto de vista do usuário, do projetista ou do programador. O ponto de vista do usuário tem foco nas diferentes visões que especialistas de domínio e usuários de negócio possuem a respeito do domínio. O ponto de vista do projetista enfoca os diferentes modelos usados durante o projeto de sistemas de informação (visão conceitual). Finalmente, o ponto de vista do programador refere-se à implementação dos sistemas.
<i>Layer</i>	A integração de aplicações pode ocorrer em três camadas: camada de dados, camada de mensagem ou serviço e camada de processo. A camada de dados lida com a troca de dados entre vários repositórios. Neste ponto, uma aplicação manipula os dados de outra aplicação diretamente no banco de dados, por meio de sua interface nativa, ignorando as regras de negócio da aplicação. A camada de mensagens ou serviços ocorre quando duas ou mais aplicações trocam mensagens entre si. Por fim, a última camada, a camada de processo constitui a integração mais complexa, pois ela enxerga uma organização como uma série de processos relacionados e os sistemas responsáveis pela execução desse processo não podem ser vistos como ilhas de informações. Dessa forma a integração nessa camada é responsável por tratar o fluxo de mensagens, regras de execução e definir o processo de execução global.
<i>Level</i>	Por fim, o autor esclarece que a integração pode acontecer em diferentes níveis, a saber: nível de hardware, de plataforma, sintático e semântico. O nível de hardware envolve a integração de diferentes computadores, redes de computadores etc. Neste nível é utilizado, por exemplo, protocolos de redes para que duas ou mais redes possam se comunicar. O nível de plataforma trata de sistemas operacionais, sistemas de bancos de dados etc. O nível sintático refere-se ao modo como modelos de dados são escritos. Por fim, o nível semântico abrange o significado pretendido dos conceitos no esquema de dados.

No domínio de Internet das Coisas – esforços têm sido realizados no sentido de construir padrões para atenuar os problemas relacionados à interoperabilidade. O *Sensor Web Enablement* (SWE) (BOTTS et al., 2006), por exemplo, é uma iniciativa do *Open Geospatial Consortium* (OGC) que definiu padrões de codificações de dados e serviços Web para acessar e armazenar dados de sensores. Esses padrões, tais como o SensorML (BOTTS; ROBIN, 2007) e O&M (COX, 2007), oferecem interoperabilidade sintática (SHETH; HENSON; SAHOO, 2008); entretanto, uma camada adicional é necessária para lidar com a compatibilidade semântica (JANOWICZ et al., 2010). Tecnologias da Web Semântica têm também sido propostas como um meio para permitir a interoperabilidade entre sensores e sistemas de detecção, podendo ser utilizadas isoladamente ou como maneira de ampliar os padrões SWE na forma da *Semantic Sensor Web* (SHETH; HENSON; SAHOO, 2008).

2.2. Ontologia

O termo Ontologia tem sua origem na filosofia e é usado neste meio para se referir a uma disciplina e também como um sistema de categorias independente de domínio (GUIZZARDI, 2005). Uma definição amplamente aceita na comunidade acadêmica descreve ontologia como sendo “uma especificação formal de uma conceituação compartilhada” (BORST, 1997).

Existem diferentes formas de classificar ontologias, sendo uma delas de acordo com o seu nível de generalidade (GUARINO, 1998): ontologias de alto nível, ontologias de domínio, ontologias de tarefas e ontologias de aplicação. Ontologias de alto nível ou ontologias de fundamentação descrevem conceitos muito gerais, independentemente de um problema ou domínio particular, como objeto, evento, ação entre outros. Ontologias de domínio descrevem a conceituação relacionada a um domínio genérico (por exemplo, direito ou biologia). Ontologias de tarefas descrevem a conceituação relacionada a uma tarefa genérica (como vendas ou serviço). Ontologias de aplicação descrevem conceitos dependentes de um domínio e uma tarefa específica.

Ontologias de nível superior são úteis no projeto de ontologias de nível inferior (por exemplo, ontologias de domínio e de aplicação), por fornecer importantes distinções

ontológicas. Entre as bem conhecidas ontologias de alto nível, é possível citar: DOLCE (GANGEMI et al., 2002), SUMO (NILES; PEASE, 2001) e UFO (GUIZZARDI, 2005).

Outra importante forma de classificar ontologias se refere à visão proposta por (GUIZZARDI, 2007), também discutida em (FALBO et al., 2013), em que é defendida uma abordagem que distingue ontologias de referência e ontologias operacionais. Uma ontologia de referência é desenvolvida com o objetivo de fazer a melhor descrição possível do domínio, sendo um tipo especial de modelo conceitual, um artefato de engenharia com o requisito adicional de representar um modelo de consenso dentro de uma comunidade. Por outro lado, uma vez que os usuários já tenham concordado com uma conceituação comum, versões operacionais de uma ontologia de referência podem ser criadas. Ao contrário das ontologias de referência, as ontologias operacionais são projetadas com o foco em garantir propriedades computacionais desejáveis.

Aspectos importantes devem ser considerados em relação à adoção de linguagens para representação de ontologias de referência e operacionais. Em (GUARINO, 2009), por exemplo, é feita uma discussão a respeito de linguagens de nível ontológico e epistemológico. O autor defende que linguagens de nível ontológico são mais apropriadas para representação clara, precisa e não ambígua do conhecimento sobre o domínio mas, por outro lado, tendem a não ser computacionalmente tratáveis, sendo este tipo de linguagem especialmente mais adequada para representar ontologias de referência. Uma vez que se deseja garantir propriedades computacionais conforme a necessidade da aplicação, típico cenário de ontologias operacionais, torna-se então necessário utilizar linguagens de nível lógico/epistemológico que tenham tal objetivo.

Pesquisas tem sido realizadas no sentido de definir metodologias sistemáticas para o desenvolvimento de ontologias. Esta atividade, conhecida como Engenharia de Ontologias, engloba um conjunto de atividades realizadas durante a conceituação, projeto, implementação e implantação de ontologias. Exemplos representativos de estudos sobre engenharia de ontologias são: Methontology (FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ; JURISTO, 1997), NeOn (SUÁREZ-FIGUEROA; GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ, 2012), Ontology Development 101 (NOY; MCGUINNESS, 2001).

Devido ao potencial das ontologias como meio de abordar aspectos da interoperabilidade, na última década, muitos formalismos de representação do conhecimento foram desenvolvidos e também adotados nas iniciativas da Internet das Coisas. Existem diversas pesquisas em IoT que têm se concentrado no uso de ontologias para lidar com problemas de interoperabilidade. Exemplos incluem os trabalhos de (M. SERRANO, P. BARNAGHI, F. CARREZ, P. COUSIN; FRIESS, 2015), (GYRARD et al., 2015), (JAYARAMAN; CALBIMONTE; MAU, 2015), (QU; LIU; TAO, 2015), (HASAN; CURRY, 2014), (BONINO; CORNO; DE RUSSIS, 2014), (SONG; CÁRDENAS; MASUOKA, 2010), (CALBIMONTE et al., 2011), e (COMPTON et al., 2012).

Em relação aos esforços de padronização, o W3C *Semantic Sensor Network Group* (SSN-XG) (COMPTON et al., 2012), definiu a ontologia *Semantic Sensor Network* (SSN) em OWL 2 (WORKING, 2009), conformidade com a ontologia de alto nível *DOLCE-UltraLite*, para descrever os recursos, as propriedades dos sensores, o ato de detecção e as observações resultantes. Desde então, vários trabalhos foram desenvolvidos adotando a ontologia SSN como base para seus modelos de descrição semântica da IoT, evidenciando sua importância para a comunidade da IoT: (COMPTON et al., 2012); (COX, 2013); (AHMEDI; JAJAGA; AHMEDI, 2013); (CALBIMONTE et al., 2014); (ZHANG; ZHAO; LIU, 2015); (STOCKER et al., 2015); (BERMUDEZ-EDO et al., 2016b); (JANOWICZ et al., 2018).

Como visto, ontologias tem sido a principal ferramenta utilizada em direção à interoperabilidade semântica a fim de atenuar a fragmentação existente no domínio de IoT. Nesse sentido, a realização de um estudo mais amplo sobre o uso de ontologias em abordagens de IoT, à luz da extensa literatura em torno deste tema, se faz necessário pois pode contribuir para sua caracterização, identificação de tendências e oportunidades de pesquisa. O Capítulo 3 apresenta a metodologia, os resultados obtidos e uma discussão geral sobre o estudo desenvolvido.

2.3. Descrição de dispositivos IoT

Como visto, a Internet das Coisas está transformando a visão padrão da Web de um conjunto de documentos e links digitais em uma Internet totalmente integrada, que

inclui dispositivos físicos, bem como elementos cibernéticos. Essa nova Web, combinando os mundos cibernético e físico, cria um novo ecossistema com novas oportunidades de programação por meio das interações e interconexões dos dois domínios.

Também foi destacado que o potencial da Internet das Coisas (IoT) é amplamente reconhecido, mas que o seu aspecto comercial está sendo retido pela fragmentação. A integração do ecossistema dispositivos inteligentes é, portanto, uma questão de pesquisa urgente, requer um esforço considerável e a sua ausência limita as oportunidades de programação para espaços inteligentes. Um sensor por si só tem valor limitado ao domínio de aplicação ao qual ele foi projetado; entretanto, existem enormes oportunidades para mercados abertos de serviços que combinam sensores, atuadores e múltiplas fontes de informação. Segundo (KHALED et al., 2018), este desafio introduz, como primeiro passo, um requisito para uma maneira uniforme de descrever as coisas em espaços inteligentes em termos do que é uma coisa (por exemplo, seus componentes), o que faz (por exemplo, os serviços oferecidos) e como ela se comunica (por exemplo, quais protocolos são suportados). Essa descrição abre o caminho para resolver os desafios de integração, configuração e gerenciamento de coisas, além de permitir interações de forma intuitiva.

Para fazer isso, as interfaces lógicas das Coisas são usadas para apresentar os recursos e serviços do dispositivo usando linguagens e anotações da Web Semântica, como DDL (*Device Description Language*) (CHEN; HELAL, 2009), IoT-DDL (KHALED et al., 2018), CoRE-TD (DATTA; BONNET, 2016) e WoT-TD. Nas próximas seções são apresentados detalhes das abordagens citadas, explorando as estratégias e características da maneira de descrever um dispositivo de Internet das Coisas.

2.3.1. Abordagem DDL - Device Description Language

Em (CHEN; HELAL, 2009), um dispositivo é caracterizado como uma entidade consistindo em um conjunto de propriedades, alguns mecanismos internos e uma interface. As propriedades provêm informações sobre o dispositivo, tais como a sua

proposta, a sua capacidade, o fabricante e requisitos operacionais. Estas são informações críticas tanto para o programador quanto para a integração de sistemas. Mecanismos internos são responsáveis pela operação do dispositivo e é desconhecido para o mundo externo. A interface do dispositivo é a ponte entre os mecanismos internos e o mundo externo, especificando a entrada e a saída do dispositivo e fornecendo orientação para as aplicações e outros serviços de como interagir com o dispositivo. A Figura 3 abaixo ilustra o modelo proposto.

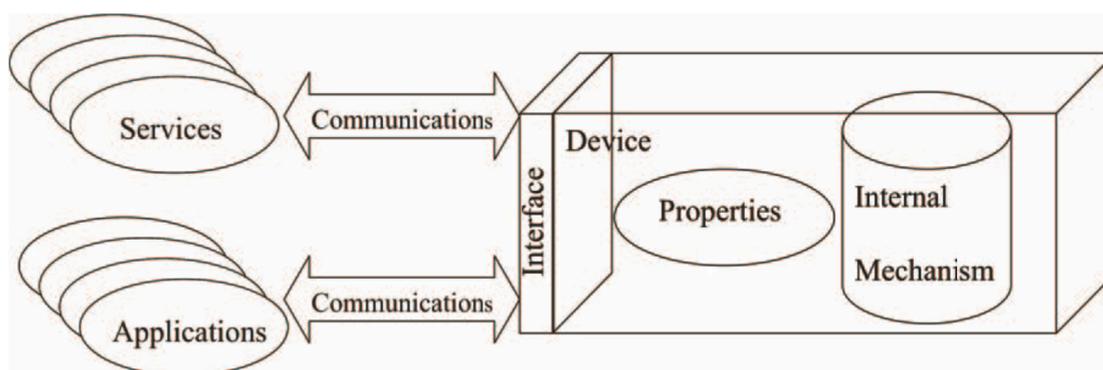


Figura 3 - Modelo de dispositivo DDL (Fonte (CHEN; HELAL, 2009)).

Os autores classificam os dispositivos em três categorias: (i) sensores, em que apenas provê uma entrada para o usuário externo; (ii) atuadores, somente reproduz uma saída oriunda de um usuário externo; e (iii) dispositivos complexos, cuja funcionalidade possibilita a capacidade tanto de prover a entrada quanto reproduzir a saída para o usuário externo.

A implementação da *Device Description Language* (DDL) consiste em duas partes: um esquema que especifica a sintaxe e um processador da linguagem Atlas-DDL, que trabalha em conjunto com o *middleware* Atlas, gerando pacotes de software de acordo com a descrição do dispositivo.

A abordagem proposta para descrever dispositivos consiste em uma linguagem de marcação que adota o XML como codificação, fornecendo alta legibilidade para humanos e máquinas. Além das restrições sintáticas básicas impostas pelo próprio XML, é fornecido um esquema DDL para definir as restrições na estrutura e no conteúdo de um documento DDL. Um trecho do esquema é mostrado na Figura 4. O processador de linguagem possui um módulo de verificação que avalia cada instância do DDL para garantir que esteja em conformidade com o esquema.

```

<xsd:complexType name="Device">
  <xsd:sequence>
    <xsd:element name="Description" type="DescriptionType"
      minOccurs = "1" maxOccurs="1" />
    <xsd:element name="Interface" type="InterfaceType"
      minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

```

Figura 4 - Um fragmento do esquema DDL que mostra o tipo de dispositivo para consistir em dois elementos: descrição e interface (Fonte (CHEN; HELAL, 2009)).

Nesta proposta, cada arquivo descritor *DDL* descreve um único tipo de dispositivos. Ele contém as informações necessárias para o registro e descoberta de serviços, como o nome e o modelo do dispositivo e a descrição de suas funções. Também descreve as principais operações do dispositivo. O *DDL* modela um dispositivo de uma perspectiva orientada a dados, em que cada operação de um dispositivo é uma coleção de cadeias de funções de entrada, processamento e saída. Essa visão é consistente com a arquitetura orientada a serviços, na qual cada dispositivo é representado por um serviço que fornece uma interface para suas propriedades, criando uma camada de abstração a respeito de seu mecanismo interno. Portanto, o foco da *DDL* é descrever a interface dos dispositivos.

O *DDL* descreve as comunicações entre um dispositivo e sua entidade de serviço correspondente usando “sinais”. Por exemplo, um sinal pode ser uma saída de tensão de um pino, um número de um conversor analógico para digital (ADC) ou um fluxo de bytes de uma porta serial, em conformidade com um determinado protocolo bem definido. Dependendo do tipo de dispositivo, os sinais podem ser unidirecionais ou bidirecionais. As “leituras” são as informações significativas que a entidade de serviço recebe do dispositivo associado, geralmente, especifica uma conversão ou um método de análise que converte sinais brutos em dados semanticamente significativos. Por exemplo, o sensor de temperatura analógico na Figura 5 emite um sinal '0010000000', que é um valor de 10 bits da porta ADC. É então convertido para 325 graus, uma leitura na unidade graus célsius centígrados.

```

<Sensor>
<Description>...</Description>
<Interface>
  <Signal id="ADC1">...</Signal>
  <Reading id="Temp1">
    <Type>Physical</Type>
    <Measurement>Temperature
  </Measurement>
  <Unit>Centigrade</Unit>
  <Computation>
    <Type>Formula</Type>
    <Expression> Temp1 =
((ADC1/1023) * 3.3)-0.5)*
(1000/10)</Expression>
  </Computation>
</Reading>
</Interface>
</Sensor>

```

Figura 5 - Sensor de temperatura (Fonte: (CHEN; HELAL, 2009))

2.3.2. Abordagem CoRE Link Format

Em (DATTA; BONNET, 2016), é apresentada uma visão sobre dispositivos de Internet das Coisas, com um olhar sobre o aspecto funcional do dispositivo e como essas questões se refletem na maneira de prover uma descrição de uma Coisa. Nessa direção, foi proposto um modelo de descrição de dispositivos, em que uma Coisa é vista em termos de suas propriedades, ações e eventos, adotando uma nomenclatura uniforme para descrever os dispositivos e um modelo de serialização baseado em JSON-LD, sob a justificativa de ser adequado a dispositivos com capacidade computacional restrita, de fácil entendimento por humanos e interpretável por máquina.

Como mencionado, as Coisas são representadas em termos de eventos, propriedades e ações, sendo esse modelo independente de protocolo. Um exemplo de LED é mostrado na Figura 6. Os valores de brilho, temperatura de cor e RGB são codificados como propriedades do LED enquanto a função liga / desliga é uma ação e a mudança de temperatura de cor é um evento. O modelo revela que o LED está conectado a um interruptor de luz (ação liga - desliga), sua temperatura de cor pode ser configurada (evento) e os valores RGB também são acessíveis.

```

{
  "interactions": [
    {
      "@type": "Property",
      "name": "brightness",
      "outputData": "xsd:unsignedByte",
      "writable": true
    }, {
      "@type": "Property",
      "name": "colorTemperature",
      "outputData": "xsd:unsignedShort",
      "writable": true
    }, {
      "@type": "Property",
      "name": "rgbValueRed",
      "outputData": "xsd:unsignedByte",
      "writable": true
    }, {
      "@type": "Property",
      "name": "rgbValueGreen",
      "outputData": "xsd:unsignedByte",
      "writable": true
    }, {
      "@type": "Property",
      "name": "rgbValueBlue",
      "outputData": "xsd:unsignedByte",
      "writable": true
    }, {
      "@type": "Action",
      "name": "ledonoff",
      "inputData": "xsd:boolean",
      "outputData": ""
    }, {
      "@type": "Event",
      "outputData": "xsd:unsignedShort",
      "name": "colorTemperatureChanged"
    }
  ]
}

```

Figura 6 - Descrição de coisa baseada em JSON-LD para um LED (Fonte: (DATTA; BONNET, 2016)).

2.3.3. Web of Things – Thing Description

Em (CHARPENAY; KÄBISCH; KOSCH, 2016), reforçado em (KAEBISCH S., 2019), é apresentada uma visão onde a descrição de um dispositivo (*Thing Description – TD*) de Internet das Coisas consiste em um Recurso semântico que descreve formalmente uma coisa única, com a qual um agente de software pode interagir, tendo como objetivo descrever os metadados e interfaces das Coisas, onde uma “Coisa” é definida como uma abstração de uma entidade física ou virtual que fornece interações e participa da Web das Coisas. As descrições de uma Coisa fornecem um conjunto de interações com base em um pequeno vocabulário que possibilita a integração de diversos dispositivos, potencializando a interoperabilidade de diversas aplicações. As *Thing Descriptions*, por padrão, são codificadas em um formato JSON que também permite o processamento

JSON-LD. Este último fornece uma base poderosa para representar o conhecimento sobre as Coisas de uma maneira compreensível por máquina. Uma instância de uma *Thing Description* pode ser hospedada pelo próprio dispositivo ou hospedada externamente quando um dispositivo tem restrições de recursos computacionais.

O *Thing Description* WoT (TD) é um elemento central do *Web of Things* (WoT) (KAEBISCH S., 2019) e pode ser considerado como o ponto de entrada de uma Coisa, sendo que, por meio da obtenção do *Thing Description*, é possível que aplicações possam conhecer, em mais detalhes, as características e capacidades de uma coisa. Uma instância de TD possui quatro componentes principais: metadados textuais sobre o dispositivo (*Thing*), um conjunto de formas de acesso à interação, que indicam como a Coisa pode ser usada, esquemas para os dados a ser trocados com o dispositivo, com intuito de facilitar a legibilidade por máquina e, finalmente, links da Web para expressar qualquer relação formal ou informal com outras coisas ou documentos na Web.

O Modelo de Interação do *W3C WoT* define três tipos de recursos de Interação: (i) Propriedades - podem ser usadas para detectar e controlar parâmetros, como obter o valor atual ou definir um estado de uma operação; (ii) Ações - modelam a invocação de processos físicos, mas também podem ser usadas para abstrair chamadas remotas; e (iii) Eventos - usados para o modelo de comunicação por *push* em que notificações, eventos discretos ou fluxos de valores são enviados de forma assíncrona para o receptor. A Figura 7 apresenta um exemplo de *Thing Description* que descreve os metadados e o modelo de interação de uma lâmpada.

```

{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "id": "urn:dev:ops:32473-WoTLamp-1234",
  "title": "MyLampThing",
  "securityDefinitions": {
    "basic_sc": {"scheme": "basic", "in":"header"}
  },
  "security": ["basic_sc"],
  "properties": {
    "status" : {
      "type": "string",
      "forms": [{"href": "https://mylamp.example.com/status"}]
    }
  },
  "actions": {
    "toggle" : {
      "forms": [{"href": "https://mylamp.example.com/toggle"}]
    }
  },
  "events":{
    "overheating":{
      "data": {"type": "string"},
      "forms": [{
        "href": "https://mylamp.example.com/oh",
        "subprotocol": "longpoll"
      }]
    }
  }
}

```

Figura 7 - W3C Web of Things - Thing Description (Fonte: (KAEBISCH S., 2019)).

No exemplo supracitado, os metadados são representados pelo *context*, *id* e o *title*; já o modelo de interação é exemplificado pela *properties*, *actions* e *events*. O elemento “*context*” representa o vocabulário utilizado na descrição do dispositivo, adicionalmente, oferece a possibilidade de adicionar conhecimento de contexto a partir de vocabulários adicionais. Esse mecanismo pode ser usado para enriquecer as instâncias de *Thing Description* com semântica adicional. O “*id*” representa a identificação única da Coisa (*Uniform Resource Name – URN*). Por fim, o “*title*” se refere ao nome designado ao dispositivo.

Como pode ser visto no exemplo de *TD* em questão, referente ao modelo de interação, existe uma propriedade denominada “*status*”, em que são fornecidas informações para indicar que esta propriedade está acessível via protocolo *HTTP* com um método *GET* no *URI* <https://mylamp.example.com/status> (indicado na estrutura *forms*) e deve retornar um valor de *status* baseado em *string*. Neste caso, o uso do método *GET* não é declarado explicitamente, mas é uma das suposições padrão definida por esta especificação. De maneira semelhante, uma ação é especificada com objetivo de alternar

o status do computador, usando o método *POST* no recurso <https://mylamp.example.com/toggle>, em que *POST* é novamente uma presunção padrão sobre como invocar uma ação. Por fim, o evento possibilita a implementação de mecanismos para envio de mensagens assíncronas por um dispositivo. Neste caso, o dispositivo permite que interessados possam realizar uma assinatura do mesmo, a fim de a ser notificado sobre um possível evento de superaquecimento da lâmpada sendo obtido por meio do protocolo HTTP (*long polling*) em <https://mylamp.example.com/oh>.

2.4. Ontologias na descrição semântica de dispositivos IoT

Ontologias tem sido investigadas em diferentes áreas da ciência e sua adoção como ferramenta vem obtendo êxito nos mais variados segmentos científicos. Nesse sentido, revisitar a literatura em busca de definições (BORST, 1997), classificações (FALBO et al., 2013),(GUARINO, 1998), metodologias (FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ; JURISTO, 1997),(SUÁREZ-FIGUEROA; GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ, 2012),(NOY; MCGUINNESS, 2001), entre outros, se mostram relevantes, contribuindo na análise de trabalhos que propõe o uso de ontologias no contexto de IoT (o Capítulo 3 expande uma investigação nessa direção).

As abordagens de descrição de dispositivos em conjunto com ontologias que fornecem semântica formal apresentam grande importância no cenário de Internet das Coisas a fim de viabilizar a compreensão das capacidades de um dispositivo por humanos e máquinas, fomentando a integração desses dispositivos físicos com artefatos de software, compondo um contexto extenso de possibilidades de desenvolvimento de novas aplicações. Em (CHEN; HELAL, 2009), iniciou-se a discussão sobre a relevância deste tema, apresentando uma visão a cerca de um dispositivo (interface, propriedades e mecanismos internos) e propondo uma abordagem de descrição em que foi adotado o XML como formato de serialização e um *schema XML* para restringir sua sintaxe. Observa-se que há um maior foco em aspectos de hardware e em explicitar o processo de conversão de sinais brutos de um sensor em formato de dados “semanticamente” significativos. Em (DATTA; BONNET, 2016), a perspectiva sobre um dispositivo evoluiu na direção de compreender o mesmo em torno de suas interações, a saber: propriedades, ações e eventos, também tendo a preocupação na uniformização da sintaxe e adotando o JSON-LD como padrão de serialização. Percebe-se que foi elevado o nível

de abstração quanto aos detalhes de hardware e o padrão de serialização escolhido potencializa sua utilização em dispositivos com capacidade computacional restrita e possibilita a inclusão de novos vocabulários, estendendo sua sintaxe. Por fim, recentemente em (KAEBISCH S., 2019) está sendo concebida uma visão sobre Coisas no contexto de IoT em que advoga a necessidade de se existir uma especificação de metadados precisa a respeito do dispositivo e a representação do seu modelo de interação, que parece ter sido inspirado no modelo apresentado em (DATTA; BONNET, 2016). Observa-se uma evolução sobre as abordagens citadas, na direção de obter um maior detalhamento sobre uma Coisa, permitindo uma descrição mais precisa do dispositivo, com uma extensa lista de campos suportados. O modelo de interação tem como destaque a estratégia adotada para expor as propriedades, ações e eventos, com padronização dos protocolos suportados e a forma de invocar com cada interação.

Atualmente, existe um esforço da academia e da indústria da direção de atenuar o problema da fragmentação existente no domínio de Internet das Coisas e a abordagem proposta pelo grupo de trabalho do *W3C Web of Things* parece ser uma grande promessa nessa direção. Os objetivos da *W3C Web of Things* (WoT) são de melhorar a interoperabilidade e a usabilidade da Internet das Coisas, por meio de uma colaboração envolvendo muitas partes interessadas ao longo de muitos anos. Entretanto, ainda são necessárias investigações em dispositivos reais e simulados a fim de avaliar essa abordagem nos mais variados cenários.

2.5. Contexto e Situação

Nesta seção são introduzidos os conceitos necessários para o entendimento da abstração “Cenário”, um importante elemento da arquitetura conceitual proposta no Capítulo 4. A definição de tal abstração requer uma breve discussão acerca dos conceitos de “contexto” e “situação”.

Segundo (ABOWD et al., 1999), *contexto* é definido como qualquer informação que pode ser usada para caracterizar uma situação pertencente a uma entidade, sendo entidade uma pessoa, um local ou um objeto de um computador, ou seja, contexto é o que pode ser dito sobre uma entidade, ou seja, o contexto não existe por si. São exemplos de entidades: pessoas, dispositivos de computação, edificações. O contexto pode ser decomposto em constituintes ou elementos de contexto. Juntos, esses constituintes

formam o contexto da entidade. Sendo assim, o contexto da entidade é o conjunto de constituintes de contexto da mesma. Exemplos de constituintes de contexto são: localização de uma pessoa, memória disponível em um dispositivo, temperatura de um edifício.

Em (COSTA et al., 2006), contexto é classificado em contexto intrínseco (*Intrinsic Context*) e contexto relacional (*Relational Context*). O contexto intrínseco define um tipo de contexto que pertence à natureza essencial de uma única entidade e não depende do relacionamento com outras entidades. Um exemplo de contexto intrínseco é a localização de uma entidade espacial, como uma pessoa ou um edifício. Contexto relacional define um tipo de contexto que depende da relação entre entidades distintas. Um exemplo de contexto relacional é Contenção, que define um relacionamento de contenção entre entidades, como um edifício de entidade que contém várias pessoas da entidade.

Em (GUIZZARDI, 2005), contexto intrínseco é classificado como uma noção ontológica de Qualidade, podendo esta ser mapeada para um valor (*quale*) em uma dimensão de qualidade. Uma dimensão de qualidade define o possível conjunto de valores aos quais um tipo de qualidade pode estar associado. A localização geográfica de uma entidade é um exemplo de qualidade, cuja dimensão da qualidade é definida por todos os valores possíveis em um sistema de coordenadas geográficas.

Situações são entidades complexas constituídas possivelmente por vários objetos (incluindo outras situações), sendo tratadas aqui como um sinônimo para o que é chamado na literatura de estado de coisas (*state of affairs*), ou seja, uma porção da realidade que pode ser compreendida como um todo (GUIZZARDI; FALBO; GUIZZARDI, 2008). Uma situação é, portanto, entendida como uma interpretação de um estado da realidade, definida pelos seus respectivos dados pertinentes ao domínio de interesse (ou seja, o contexto), e que contribui com algum significado para o usuário pessoa ou sistema que os consome ou executa (COSTA, 2007). De forma similar, (PEREIRA; COSTA; ALMEIDA, 2013) define uma situação como sendo uma abstração de eventos do mundo real derivados do contexto e hipóteses sobre como um contexto observado se relaciona com os interesses de usuários e aplicações. São exemplos de situações: “João estar gripado e com febre”, e “Maria estar casada com João que trabalha na UFES”.

2.6. Redes de Petri

Nesta seção são apresentados conceitos gerais sobre Redes de Petri, sendo esta a abordagem usada nesta tese para formalizar o conceito de cenário e, com isso, utilizada para capturar a características de dinamicidade intrínseca dos ambientes em IoT.

Redes de Petri foram propostas por Carl Adam Petri em 1962, em sua tese de doutorado (PETRI, 1962), consistindo em uma técnica de modelagem que permite a representação de sistemas, utilizando como alicerce uma forte base matemática (MACIEL; LINS; CUNHA, 1996). Essa técnica possui a particularidade de possibilitar modelar sistemas dinâmicos, paralelos, concorrentes, assíncronos e não-determinísticos.

A representação gráfica de uma rede de Petri básica é formada por quatro elementos: (a) lugares – círculo, (b) ficha ou *token* - ponto, (c) transição - barra e por fim (d) arcos. A Figura 8 mostra os elementos básicos de um grafo associado às redes de Petri.



Figura 8 - Redes de Petri - elementos básicos.

Um conjunto de regras governa a operação da rede como um todo. A operação das Redes de Petri é baseada na visão das fichas movendo-se em uma rede abstrata. A Figura 9 ilustra um exemplo de uma Rede de Petri.

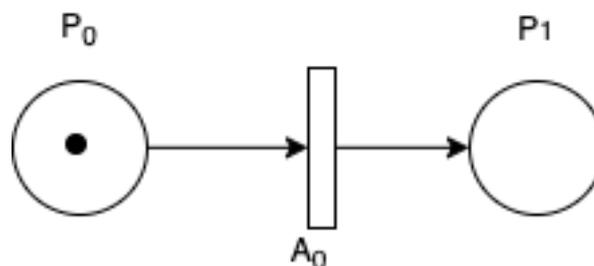


Figura 9 - Rede de Petri.

Os elementos básicos que permitem a definição de uma Rede de Petri, são polivalentes e, em grande medida, podem ser interpretados livremente. A seguir são

apresentados possíveis interpretações sobre os elementos da Rede de Petri (CARDOSO; VALETTE, 1997):

- *Lugar* (representado por um círculo): pode ser interpretado como uma condição, um estado parcial, uma espera, um procedimento, um conjunto de recursos, um estoque, uma posição geográfica em um sistema de transporte, etc. Em geral, todo lugar tem um predicado associado, por exemplo, máquina livre, peça em espera.
- *Transição* (representada por barra ou retângulo): é associada a um evento que ocorre no sistema, como o evento iniciar a operação. Cada transição em uma Rede de Petri tem um número de lugares de entrada e de saída; esses lugares podem representar uma pré-condição ou uma pós-condição de um evento, ou recursos requeridos e liberados por um evento.
- *Ficha* (representado por um ponto em um lugar): são entidades conceituais, que modelam os objetos que se movem por uma rede real. Pode representar um objeto (recurso ou peça) em uma certa posição geográfica (em um determinado estado), ou ainda, uma estrutura de dados que se manipula. Por exemplo, na Figura 10 uma ficha no lugar máquina livre indicada que a máquina está livre (predicado verdadeiro). Se não tem fichas neste lugar, o predicado é falso, por conseguinte a máquina não está livre. Se no lugar peças em espera houvesse três fichas, indicaria que existem três peças em espera.
- *Arcos*: representam os caminhos dos objetos através do sistema; arcos conectam lugares às transições e transições a lugares (a direção do caminho é indicada por uma seta no final do arco).

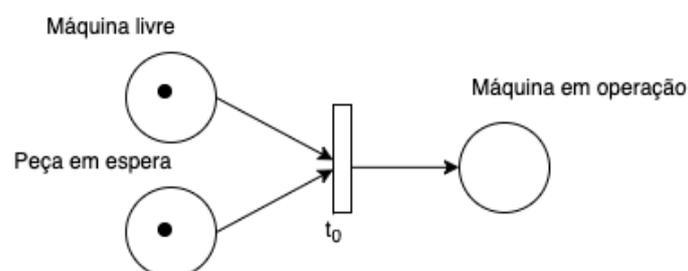


Figura 10 - Rede de Petri (Fonte: (CARDOSO; VALETTE, 1997)).

A primeira observação a fazer é que estas diversas interpretações dos lugares e fichas são bastante variadas (CARDOSO; VALETTE, 1997). Podem ser utilizadas para

descrever entidades abstratas como condições ou estados, mas também, entidades físicas como peças ou depósitos. Pode-se também chegar à banalização completa, no nível descritivo, entre as peças, ferramentas e, de modo geral, entre todos os recursos utilizados na fábrica. É esta grande banalização que permite uma visão sintética do sistema a ser modelado e que autoriza certos procedimentos de análise.

Matematicamente, uma Rede de Petri é um grafo bipartido direcionado, com uma marca inicial (MOORE; BRENNAN, 1995). Um grafo bipartido é aquele no qual existem dois tipos de nós (lugares e transições) e os arcos vão de um tipo de nó para outro tipo, isto é, um arco não pode conectar dois nós do mesmo tipo. Os arcos tem direção, indicando que o fluxo ocorre em uma única direção. A marca inicial corresponde à distribuição inicial das fichas nos lugares da Rede de Petri. Uma ficha muda quando uma transição dispara, isto é, quando um evento ou atividade acontece, e o disparo é equivalente à mudança do estado, descrevendo o comportamento dinâmico de uma Rede de Petri.

Existem diversas maneiras de se classificar as Redes de Petri, sendo uma forma amplamente utilizada, consistindo em agrupá-las quanto ao seu grau de abstração. Neste sentido, podendo separá-las em Redes de Petri de baixo nível e de alto nível (MARRANGHELLO, 2005).

As Redes de Petri de baixo nível são aquelas cujo significado de suas marcas não são diferenciáveis a não ser pela estrutura da rede à qual estão associadas. Elas ainda podem ser subdivididas em elementares e lugar/transição. As redes elementares são aquelas que conservam as características das redes propostas por Petri, em sua tese de doutorado (PETRI, 1962). Elas são extremamente restritivas do ponto de vista de modelagem, pois permitem a existência de apenas uma marca em cada elemento da rede forçando, desta forma, a representação das nuances de um sistema modelado totalmente por meio dos elementos estruturais da rede. As redes lugar/transição buscam minimizar este esforço de modelagem relaxando um pouco as restrições impostas nas redes elementares, fundamentalmente, permitindo a utilização de mais de uma marca em cada elemento da rede, permitindo uma compactação razoável dos modelos.

As redes de alto nível são aquelas cujas marcas incorporam alguma semântica, viabilizando sua diferenciação. Esta semântica pode ir desde a atribuição de valores ou cores às marcas, até a adoção de noções de tipos de dados abstratos, conferindo-lhes um grande poder de expressão. Um exemplo de Redes de Petri de alto nível inclui as Redes de Petri Coloridas, Redes de Petri Orientadas a Objetos, Redes de Petri Estocásticas e

Redes de Petri Temporizadas constituem outros exemplos de extensões às Redes de Petri que vem sendo usados na modelagem de sistemas dinâmicos e adaptativos.

2.7. Considerações sobre o capítulo

Este capítulo buscou explorar diversas áreas de pesquisa atuais, com intuito de obter uma melhor compreensão das contribuições deste trabalho, realizando uma revisão dos conceitos gerais dessas áreas de pesquisa que fundamentam esta tese.

O capítulo 3 a seguir apresenta um estudo sistemático sobre o uso de Ontologias em abordagens IoT. Ontologias constituem a tecnologia base de descrição semântica adotada nesta tese como forma de promover a interoperabilidade no contexto de IoT.

Capítulo 3

3. Interoperabilidade em Internet das Coisas

Este capítulo apresenta uma revisão sistemática da literatura que tratam de questões de pesquisa relacionadas aos temas centrais desta tese, com foco no uso de Ontologias em IoT. São discutidas questões como caracterização histórica do interesse da comunidade acadêmica neste tema, linguagens de representação de modelos, metodologias, uso de Ontologias de fundamentação entre outros. O capítulo está estruturado como segue. A Seção 3.1 descreve o protocolo definido para a realização desta revisão; a Seção 3.2 apresenta os trabalhos selecionados conforme o método definido na seção anterior; a Seção 3.3 traz a síntese dos dados e resultados obtidos após análise dos trabalhos selecionados; a Seção 3.4 discute os resultados da seção anterior, posicionando e indicando considerações em relação a literatura da área.

3.1. Metodologia

Com a popularidade e potencial da Internet das Coisas, a interoperabilidade se torna uma questão central, a fim de atenuar a fragmentação presente neste domínio. Um aspecto particular das pesquisas em interoperabilidade no contexto da Internet das Coisas discute a descrição semântica de dados, cujo objetivo central é potencializar a reutilização de dados oriundos deste cenário, se tornando um importante desafio de pesquisa. Neste sentido, pesquisadores se dedicam ao desenvolvimento de arquiteturas que possibilitem transformar automaticamente os dados oriundos de sensores inteligentes em formatos enriquecidos semanticamente. Geralmente, tais dados possuem um conjunto extenso de metadados contextuais que precisam ser descritos em um cenário de reutilização desses dados por aplicações distintas, por exemplo, qual é a acurácia, a precisão e o intervalo de abrangência do sensor? Qual foi o processo utilizado para obtenção do dado? Qual é a característica de interesse? Essas e outras questões são relevantes para tomada de decisão

em um processo de reutilização de dados por aplicações distintas. Nessa direção, o enriquecimento semântico de dados permite que os usuários decidam mais precisamente a respeito do reuso de dados, conforme os requisitos da aplicação.

Em (SHI et al., 2018) é apresentada uma arquitetura genérica, com base em uma revisão da literatura, que define um conjunto de atividades necessárias para o enriquecimento de dados em ambientes de Internet das Coisas. Conforme ilustrado na Figura 11, a arquitetura genérica para enriquecimento semântico é composta por duas partes: (i) uma camada composta por vários dispositivos heterogêneos responsáveis pela detecção e coleta de dados do ambiente. (ii) é o componente principal, denominado “*enriquecimento semântico de dados*”, composto por três atividades, a saber: coleta de dados, pré-processamento de dados e anotação semântica. A coleta de dados lida com questões relacionadas a interação dos dispositivos com a arquitetura. O pré-processamento visa detectar discrepâncias e anomalias oriundas da coleta de dados com objetivo de aprimorar a precisão dos dados sensoreados. Por fim, a anotação semântica é responsável por incluir semântica aos dados pré-processados. Geralmente, a anotação semântica é composta de duas atividades: (i) a modelagem semântica, que consiste em analisar o domínio de aplicação e construir ontologias que representem o domínio em questão; e (ii) o processo de anotação semântica, que ocorre a partir da vinculação dos conceitos presentes na ontologia desenvolvida aos dados pré-processados.

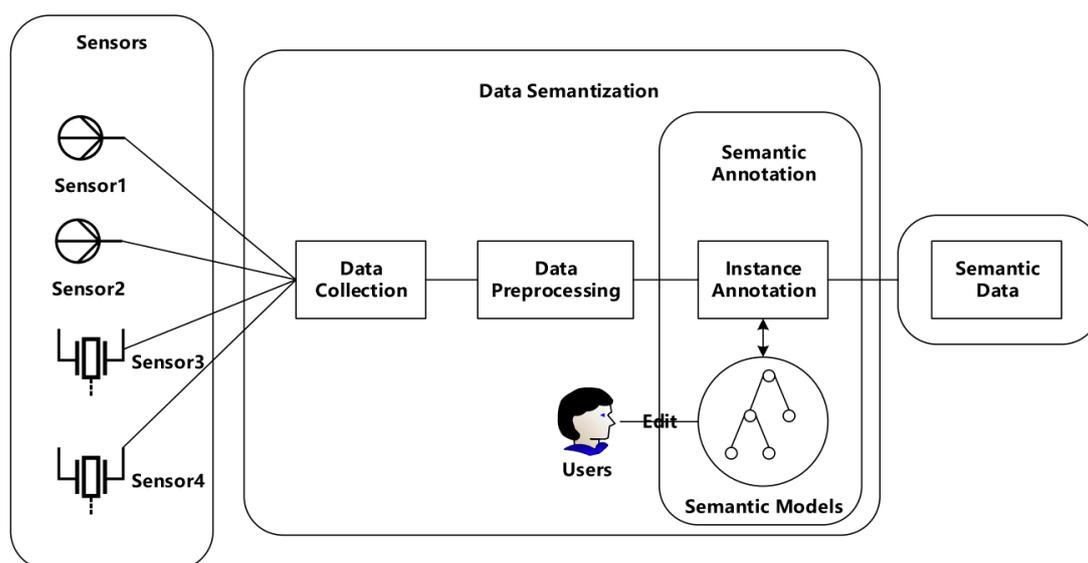


Figura 11 - Arquitetura genérica para anotação semântica (Fonte: (SHI et al., 2018))

Este mapeamento sistemático teve como objetivo investigar abordagens no contexto de Internet das Coisas que adotaram ontologias como forma de promover a interoperabilidade por meio da descrição semântica de dados, tentando entender a evolução histórica, as características e métodos utilizados nas abordagens disponíveis na literatura científica, identificando lacunas de pesquisa e possíveis abordagens que possam ser melhor exploradas, indicando novas possibilidades de pesquisa.

O método de pesquisa aplicado a esta seção foi baseado em (KITCHENHAM, 2004) e (KITCHENHAM; BUDGEN; BRERETON, 2011), e inspirado em outros trabalhos (NARDI; DE ALMEIDA FALBO; ALMEIDA, 2013; TEIXEIRA et al., 2017). Uma revisão sistemática da literatura é um estudo secundário que adota um método bem definido para identificar, analisar e interpretar as evidências disponíveis de maneira imparcial e repetível. Um estudo primário é um estudo empírico que investiga uma questão de pesquisa específica a partir da qual é proposto uma contribuição bem definida. Um estudo secundário é um estudo que revisa estudos primários relacionado a questões específicas de pesquisa com o objetivo de integrar/sintetizar evidências relacionadas as questões de pesquisa.

Uma revisão sistemática da literatura envolve três fases: planejamento, realização e relato da revisão. O planejamento envolve atividades de pré-revisão e abrange a definição dos seguintes itens: questões de pesquisa, critérios de inclusão e exclusão, fontes de estudo, *string* de pesquisa e procedimentos de mapeamento. Estes itens compõem o protocolo de revisão. A fase de realização (“*conducting*”) é preocupada com a busca e seleção de estudos, extração de dados e síntese. A última fase, relato da revisão, envolve redigir os resultados, dando destaque para as partes potencialmente interessadas.

Questões de pesquisa. Este mapeamento visa responder às seguintes questões de pesquisa, considerando o contexto das iniciativas de IoT:

- (RQ1) Quão ampla tem sido a adoção de ontologias em iniciativa IoT, ao longo dos anos?
- (RQ2) Quais estudos usam ou propõem o uso de ontologias de alto nível? Quais são estas ontologias?

- (RQ3) Quais são as formas de representação utilizadas na construção de ontologias no contexto de IoT?
- (RQ4) Quais são os tipos de ontologias (considerando o seu nível de generalização) tem sido empregadas? Quais linguagens de representação foram utilizadas em cada nível?
- (RQ5) Os estudos seguem uma abordagem metodológica para desenvolvimento de ontologias em iniciativas da área de IoT? Quais são essas metodologias?
- (RQ6) Quantos estudos reutilizam ontologias existentes em suas abordagens? Quais são essas ontologias?

As questões de pesquisa supracitadas visam caracterizar as pesquisas que vem sendo realizadas sobre o uso de ontologias em IoT, com propósito de promover a interoperabilidade semântica. Nesse sentido, tais questões visam investigar: (i) o interesse, influências e tendências da comunidade científica, ao longo do tempo, a respeito do uso de ontologias para lidar com o problema de interoperabilidade em iniciativas IoT; (ii) se e como ontologias de alto nível vem sendo usadas para promover a interoperabilidade semântica no contexto de IoT, considerando que elas descrevem conceitos muito gerais, independentemente de domínio particular; (iii) se a forma de representação de uma ontologia pode influenciar na qualidade do modelo construído, conforme os requisitos computacionais ou de expressividade; (iv) se a adoção de um processo sistemático para construção de ontologias tende a gerar modelos com melhor qualidade (SIMPERL; TEMPICH, 2006); (v) se o reuso de ontologias tem como benefícios o aumento da interoperabilidade semântica entre sistemas e pode reduzir o tempo de desenvolvimento de aplicações (GYRARD; ZIMMERMANN; SHETH, 2018).

Critérios de inclusão e exclusão: a seleção dos estudos primários foi baseada nos seguintes critérios, que foram organizados em 1 critério de inclusão (IC) e 6 critérios de exclusão (EC). O critério de inclusão é: (IC1) “O estudo aborda uma proposta para Internet das Coisas que propõe ou adota uma ontologia ou modelo conceitual”. Os critérios de exclusão são: (EC1) “O artigo não possui resumo; (EC2) O artigo está publicado como resumo”; (EC3) O estudo não está escrito em inglês; (EC4) O estudo é uma versão antiga de um trabalho já considerado; (EC5) O artigo não é um estudo primário (o que exclui *short papers*, *editorials*, *summaries of keynotes*, *workshops*, e *tutorials*); (EC6) Não foi possível acessar o texto completo do artigo.

Fontes: foram usadas buscas automáticas para coletar os estudos. A busca foi aplicada em cinco banco de dados eletrônicos representativos, frequentemente utilizados em revisões sistemáticas. As fontes são: IEEE Xplore (<http://ieeexplore.ieee.org>), ACM Digital Library (<http://dl.acm.org>), Scopus (<http://www.scopus.com>), Science Direct (<http://www.sciencedirect.com>), Compendex (<http://www.engineeringvillage2.org>).

String de busca. Para a *string* de busca, foram utilizados dois grupos de termos:

Tabela 2 - String de busca.

Parte 1 – Termos de busca	Parte 2- Termos de busca
((<i>"iot"</i> OR <i>"internet of things"</i> OR <i>"sensor"</i> OR <i>"wsn"</i> OR <i>"wireless sensor network"</i> OR <i>"sensor network"</i> OR <i>"web of things"</i>)	(<i>"ontology"</i> OR <i>"reference model"</i> OR <i>"semantic model"</i> OR <i>"information model"</i>)

3.2. Trabalhos selecionados

O processo de busca foi conduzido em setembro de 2017 cobrindo os cinco banco de dados eletrônicos usados. Como tal, os estudos publicados até essa data foram o foco do nosso processo de recuperação. Um total de 2006 registros foram recuperados: 91 ACM, 957 Compendex, 443 IEEE Xplorer, 107 Science Direct, 408 Scopus.

Após o processo de busca, o processo de seleção foi conduzido progressivamente em três etapas. Na primeira etapa, eliminamos estudos duplicados (768 artigos). Na segunda etapa, foi aplicado os critérios de inclusão e exclusão considerando apenas o título e o resumo (resultando em uma redução de 81,74%). O conjunto resultante de estudos foi refinado em uma terceira etapa, que considerou todo o texto (resultando em redução de 69.02%).

Todo o processo de busca e seleção de artigos foi repetido em novembro de 2018 para atualizar nossos estudos e tendências. Desconsiderando todos os artigos encontrados no primeiro processo de busca, um total de 472 registros foram recuperados: 21 ACM,

204 Compendex, 78 IEEE Xplorer, 53 Science Direct, 116 Scopus. Na primeira etapa, eliminamos os estudos duplicados (143 artigos). Na segunda etapa, foi aplicado os critérios de inclusão e exclusão considerando apenas o título e o resumo (resultando em uma redução de 66.22%). Na terceira etapa, a partir da leitura completa dos estudos, foram selecionados 32 artigos.

A Tabela 3 resume os estágios e seus resultados, mostrando a redução progressiva do número de estudos ao longo do processo de seleção (de 2478 para 103 estudos, com uma taxa de redução de cerca de 95,84%).

Tabela 3 - Resultado do processo de seleção (por estágio).

Stage	Criteria	Analyzed Content	Initial N° of Studies	Final N° of Studies	Rate of Reduction
<i>September 2017</i>					
1st Stage	Eliminating duplications, and wrongly retrieved studies by the engines	Title and abstract	2006	1238	38.28%
2nd Stage	IC1, EC1, EC2, EC3, and EC4	Title and abstract	1238	226	81.74%
3rd Stage	IC1, EC2, EC3, EC4, EC5 and EC6	Whole text	226	70	69.02%
<i>November 2018</i>					
1st Stage	Eliminating duplications, and wrongly retrieved studies by the engines	Title and abstract	472	329	30.30%
2nd Stage	IC1, EC1, EC2, EC3, and EC4	Title and abstract	329	121	66.22%
3rd Stage	IC1, EC2, EC3, EC4, EC5 and EC6	Whole text	121	32	85.84%

Todas as referências obtidas no processo de consulta foram disponibilizadas no Mendeley Dataset.¹

¹ doi:10.17632/pghwxhnrmd.3

3.3. Síntese de Dados e Resultados

Nesta seção é apresentada a síntese de dados e os resultados conduzidos pelas questões de pesquisa. A Seção 3.3 conduz uma discussão realizada à luz dos resultados apresentados nesta seção.

Abordagens em IoT que adotam ontologias ao longo dos anos (RQ1). A fim de apresentar uma visão geral sobre esforços no uso de ontologias na área de IoT, por ano de publicação, é apresentado na Figura 12 uma distribuição dos estudos selecionados (103). Um trabalho em 2007 indicou a relevância da representação de dados e propôs uma abordagem inicial nesse sentido. A partir de 2010 é possível notar um crescimento no número de estudos publicados até 2015, havendo uma pequena diminuição em 2016, se mantendo estável no ano seguinte. Em 2018, houve um aumento expressivo de estudos, indicando um forte interesse da comunidade científica sobre o uso de ontologias em Internet das Coisas.

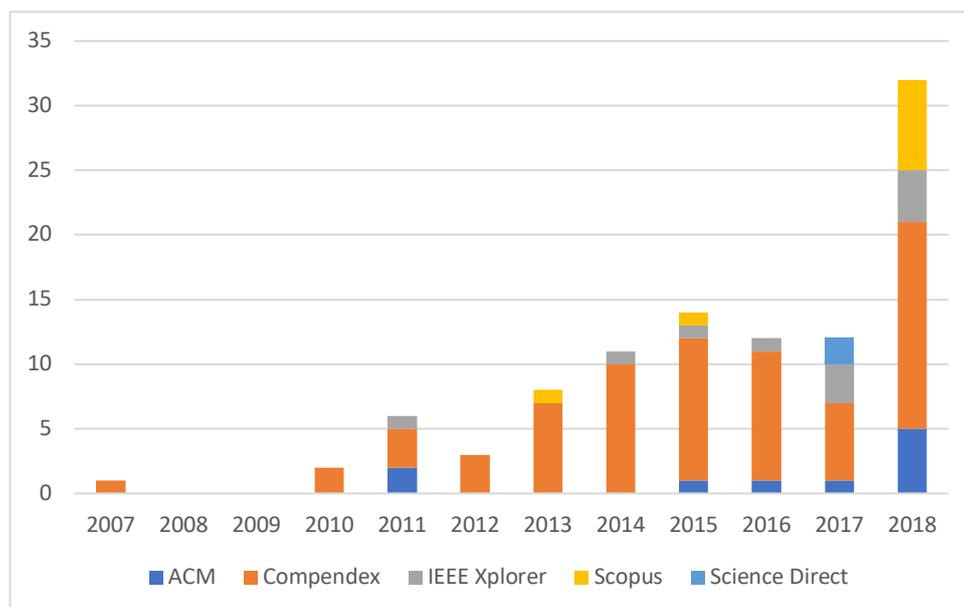


Figura 12 - Apresenta uma distribuição anual dos artigos selecionados.

Uso de ontologias fundamentais (RQ2). Os estudos foram classificados conforme sua menção explícita à utilização de ontologias de fundamentação em abordagens no cenário de IoT. A Figura 13 apresenta as ontologias de fundamentação e o percentual de estudos que adotam.

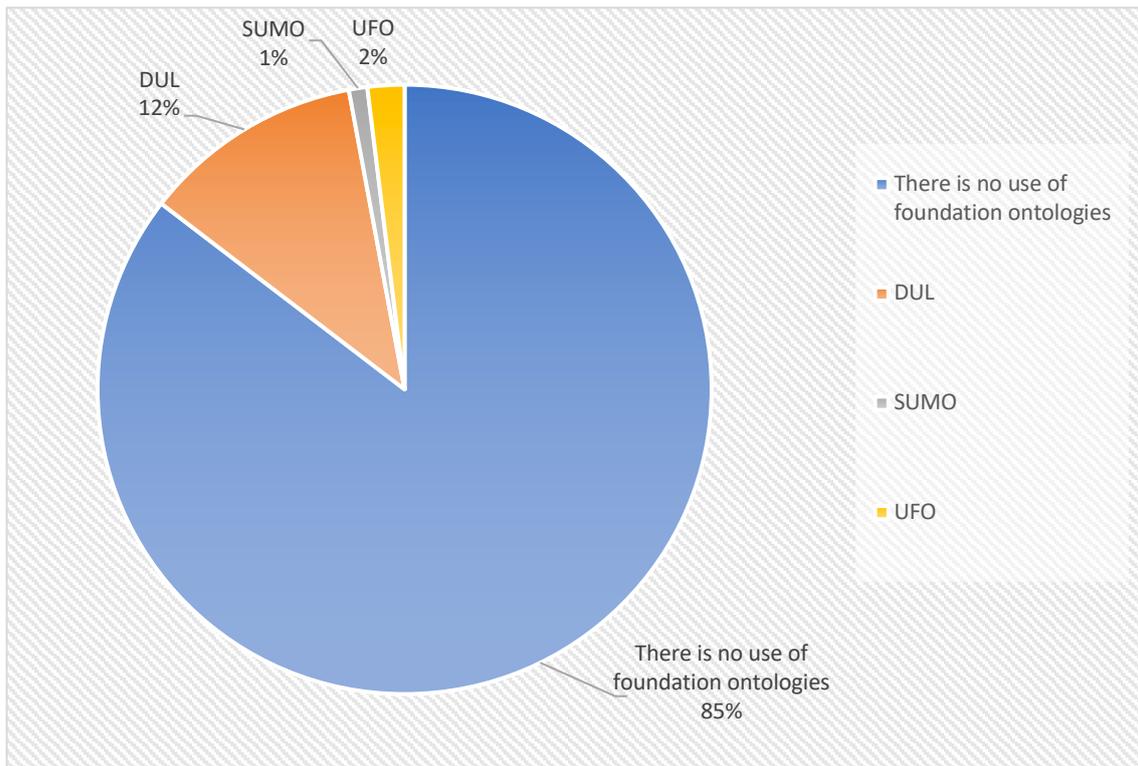


Figura 13 - Distribuição dos estudos selecionados com foco em ontologias de fundamentação.

A maior parte dos estudos (85%) não adota explicitamente ontologias fundamentais em suas abordagens. Os demais estudos (15%), por sua vez, indicam o uso de ontologias de fundamentação, destacando-se a ontologia DUL (DOLCE Ultra Lite), adotada por 12 estudos, UFO (Unified Foundation Ontology), utilizada por 2 artigos, e SUMO (Suggested Upper Merged Ontology) por um estudo.

Linguagens de representação utilizadas na construção de ontologias no contexto de IoT (RQ3). Considerando a análise dos estudos, estes afirmam representar ontologias usando uma variedade de formalismos e técnicas, variando de linguagens da Web Semântica às técnicas de representação de dados mais simplistas. Com base nesse aspecto, identificamos seis categorias: OWL, RDF, XML, UML, JSON e Representação do conhecimento.

As primeiras cinco categorias referem-se diretamente a uma tecnologia específica. A categoria “representação do conhecimento” constitui representações gráficas como taxonomias, mapas conceituais e grafos. A Figura 14 apresenta o percentual de estudos por categoria (um estudo pode se enquadrar em mais de uma categoria).

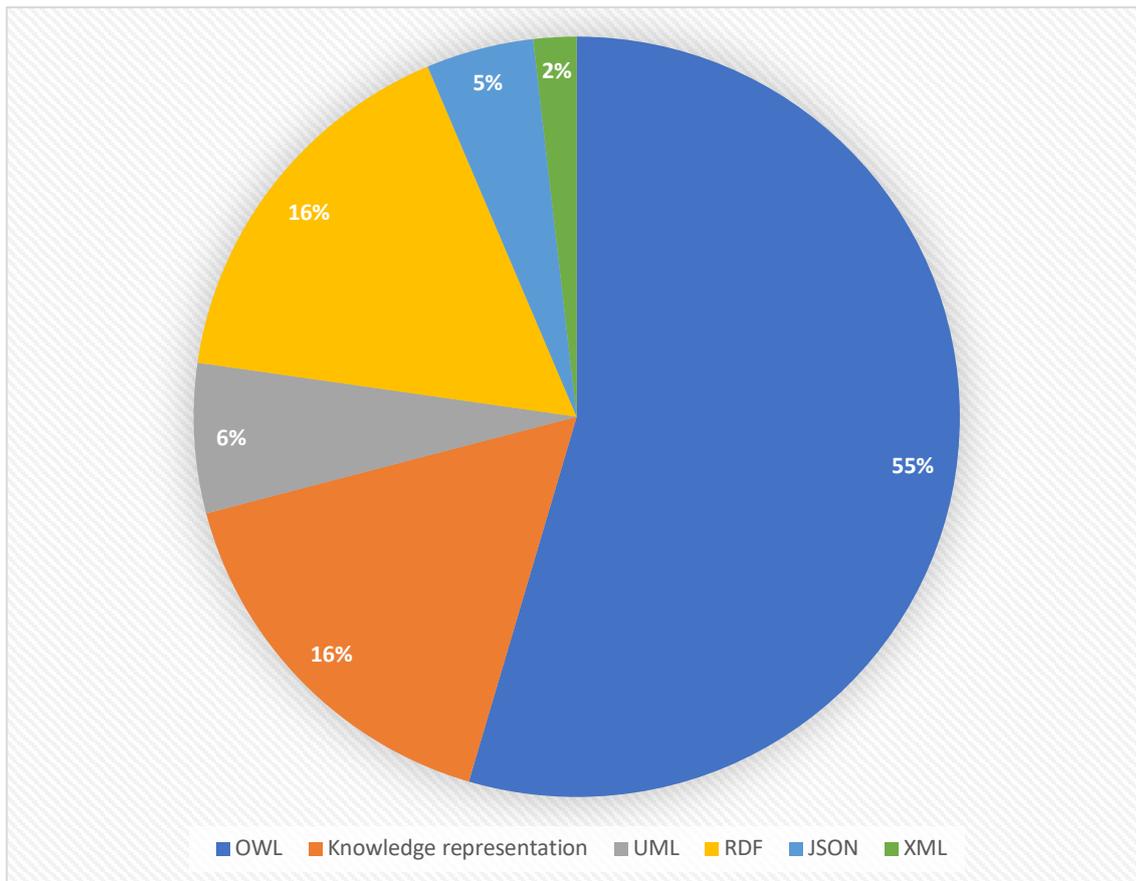


Figura 14 - Linguagens de representação de ontologias em abordagens em IoT.

Tecnologias da Web Semântica representam 73% dos artigos analisados, distribuídos em OWL (60 estudos), RDF (18 estudos), and XML (dois estudos). Com relação às linguagens de modelagem, UML é adotado por 7 estudos. Em contraponto, 18 dos trabalhos utilizam representação do conhecimento como forma de descrever ontologias em abordagem em IoT, dentro desta categoria incluem taxonomias, mapas conceituais e grafos. Por fim, JSON foi utilizado em cinco dos estudos como forma de implementação de ontologias.

Tipos de ontologias (considerando o seu nível de generalização) e suas linguagens de representação (RQ4). A Figura 15 ilustra os resultados referentes ao nível de generalização das ontologias propostas. Observa-se que 44% dos trabalhos analisados desenvolveram ontologias no nível da aplicação e 56% no nível do domínio.

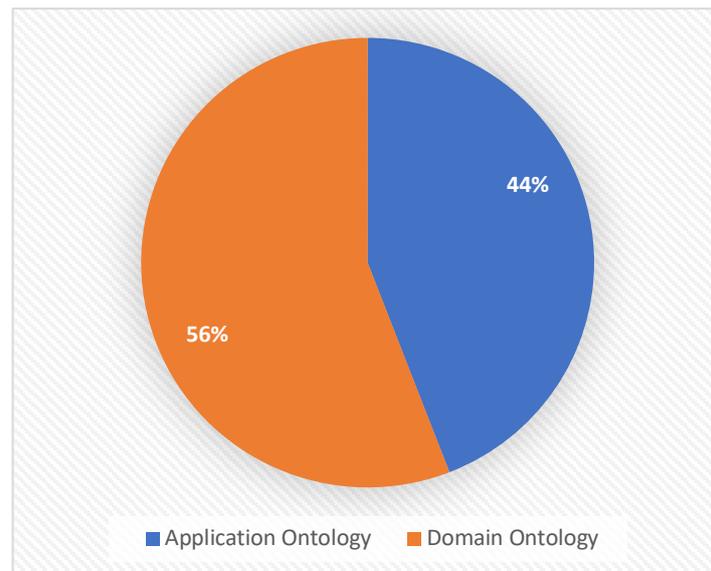


Figura 15 - Tipos de ontologias conforme seu nível de generalização.

A Figura 16 ilustra a relação entre o tipo de ontologias e a linguagem de representação correspondente, indicando a quantidade de estudos que adotam uma determinada linguagem, organizados por nível de generalização.

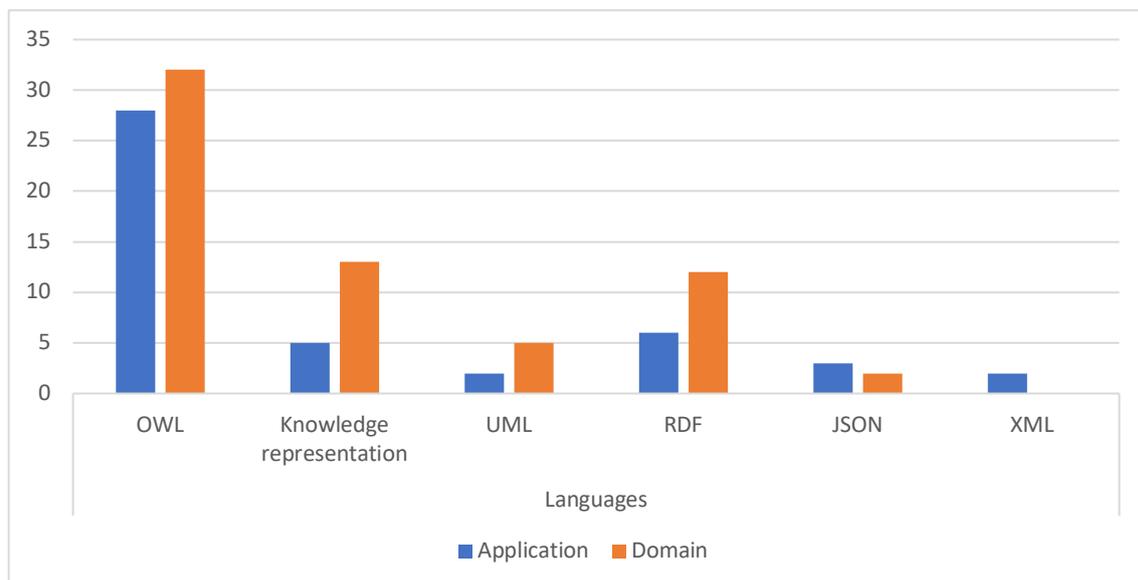


Figura 16 - Linguagens e sua adoção por nível de generalização das ontologias.

Quando organizados por nível de generalização, observamos que 28 estudos adotaram o OWL como a linguagem de implementação no nível da aplicação e 32 estudos no nível do domínio. A categoria Representação do Conhecimento foi adotada por 16%

dos estudos, destacando seu uso para representar ontologias em nível de domínio (13 estudos) e em nível de aplicação (cinco estudos). A UML foi usada por dois estudos para representar ontologias no nível da aplicação e cinco no nível do domínio. O JSON foi adotado por três estudos no nível da aplicação e dois estudos no nível do domínio. Por fim, o XML foi usado por dois estudos, apenas no nível de aplicação.

Abordagem metodológica para desenvolvimento de ontologias (RQ5). A Figura 17 apresenta o quantitativo de estudos que adotam explicitamente uma abordagem metodológica para o desenvolvimento de ontologias em iniciativas de IoT.

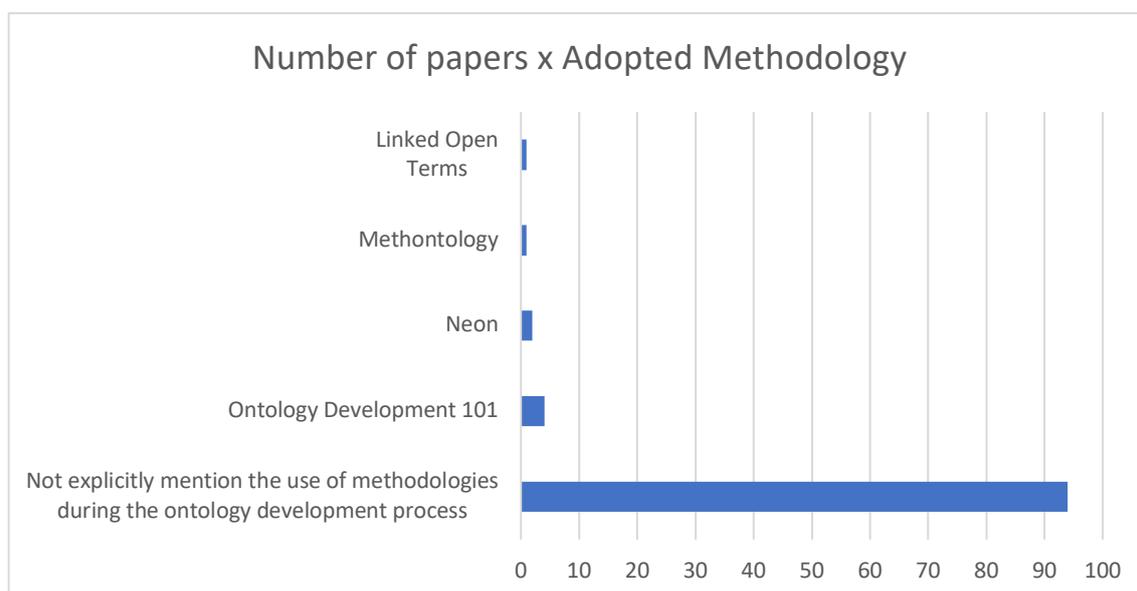


Figura 17 – Abordagens metodológicas usadas no desenvolvimento de ontologias em IoT.

A maioria dos estudos (94) não mencionou explicitamente o uso de metodologias durante o processo de desenvolvimento da ontologia. Entre os oito estudos que adotaram metodologias, quatro deles adotaram o Ontology Development 101 (NOY; MCGUINNESS, 2001) e dois utilizaram o NeOn (SUÁREZ-FIGUEROA; GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ, 2012). Methontology (FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ; JURISTO, 1997) e Linked Open Terms foram adotados por apenas um estudo cada.

Reuso de ontologias (RQ6). Considerando os estudos analisados, ontologias foram reutilizadas em 41% dos trabalhos, enquanto 59% dos estudos não fazem o reuso

de ontologias existentes em seu processo de desenvolvimento, conforme apresentado na Figura 18.

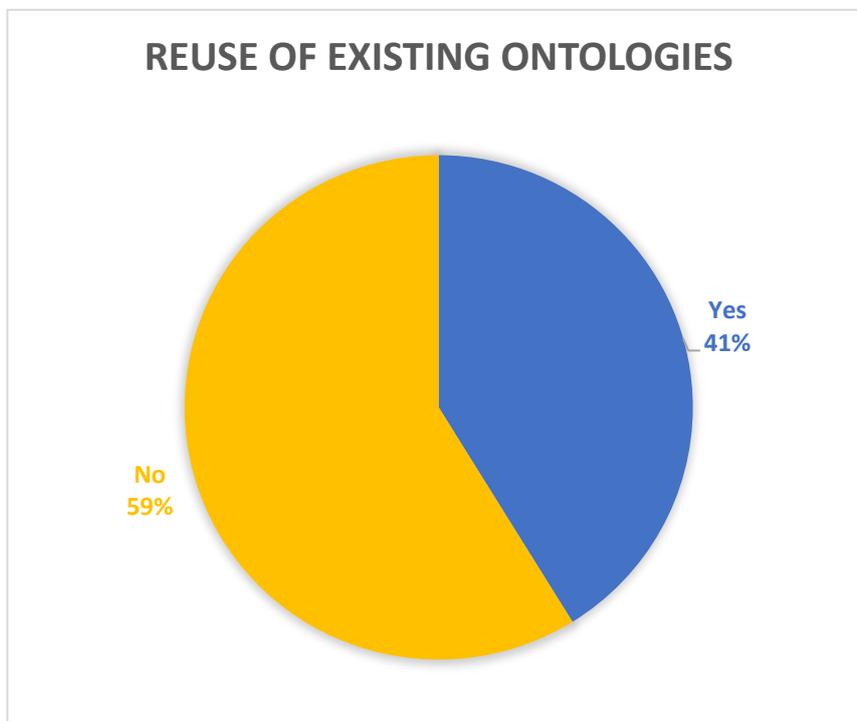


Figura 18 - Estudos que reutilizam ontologias existentes em suas propostas.

No universo dos estudos que reutilizam ontologias, destaca-se a ontologia SSN, sendo adotada por 50 estudos, como pode ser visto na Figura 19.

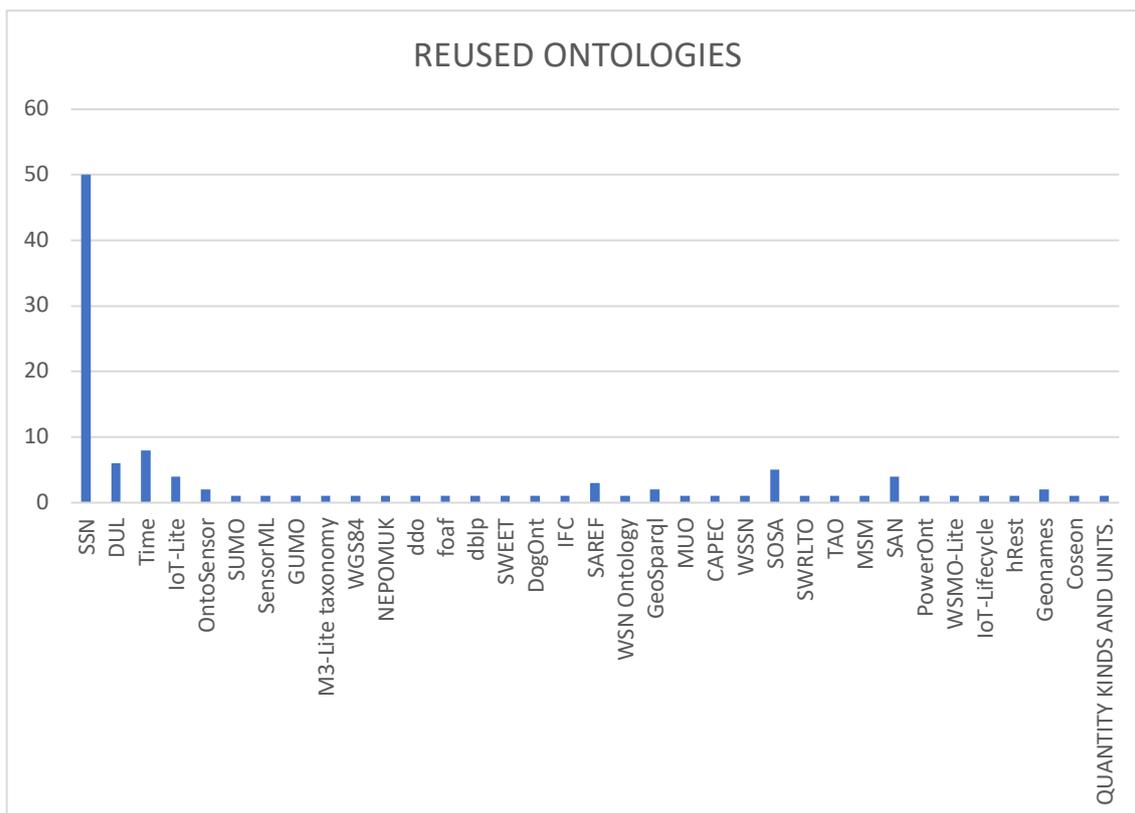


Figura 19 – Ontologias reutilizadas em abordagens IoT.

3.4. Discussão

Com base nos resultados apresentados na seção anterior, algumas questões importantes são discutidas a seguir.

A discussão se inicia analisando a distribuição de estudos relevantes ao longo de 12 anos, conforme mostrado na Figura 12. Os resultados indicam claramente uma tendência ao crescimento de soluções de IoT que empregam modelos para descrições sintáticas e semânticas. Em 2007, o estudo (ABERER; HAUSWIRTH; SALEHI, 2007) apresentou a preocupação com a interoperabilidade de dados em redes de sensores heterogêneas. Os autores propuseram uma arquitetura e uma primitiva descrição sintática dos dados, adotando o XML como uma linguagem de troca de dados. Nesse mesmo ano, o Open Geospatial Consortium (OGC) apresentou o Observations and Measures model O&M (COX, 2007) e o Sensor Model Language sensorML (BOTTS; ROBIN, 2007), um padrão de descrição sintática para redes de sensores, que refletia o tipo de abordagem usada para enfrentar os desafios de interoperabilidade da época. Esses trabalhos inauguram o que ficou mais evidente no final de 2010, quando observamos as primeiras

publicações que abordaram efetivamente a questão da interoperabilidade semântica usando ontologias como base para modelagem conceitual no domínio da IoT. De fato, a partir de 2010, tem havido um crescente interesse de pesquisa em direção à semântica, uma vez que a descrição sintática impõe limitações para alcançar uma interoperabilidade eficaz dos dados (JANOWICZ et al., 2010).

O *W3C Semantic Sensor Network Incubator Group* (W3C-XG) iniciou um trabalho com intuito de integrar e alinhar as pesquisas relacionadas a interoperabilidade de dados em redes de sensores às tecnologias da Web Semântica. O resultado deste grupo de trabalho foi o desenvolvimento da ontologia *Semantic Sensor Network*, SSN (COMPTON et al., 2012). A partir deste trabalho, muitas pesquisas foram realizadas no sentido de estender, combinar, e adicionar novos conceitos à SSN, potencializando sua utilização nos mais variados domínios. Como pode ser observado na seção anterior, considerando o protocolo adotado nesta revisão, 50 estudos reutilizaram de alguma maneira a ontologia SSN, demonstrando a importância desta ontologia e o esforço de padronização da comunidade de IoT.

As extensões da ontologia SSN tomaram três direções (WANG; ZHANG; LI, 2015): (i) estender a SSN para atender uma aplicação específica; (ii) combinar a SSN com outra ontologia para resolver um problema real e (iii) adicionar novos conceitos à SSN de maneira a torná-la de uso mais geral, potencializando sua utilização em domínios distintos.

Em contraponto com as direções supracitadas, o trabalho (BERMUDEZ-EDO et al., 2016b) propõe uma simplificação dos conceitos existentes na ontologia SSN, com intuito de potencializar seu uso em aplicações do mundo real, especialmente no que se refere aos requisitos de desempenho. Na mesma direção, SSN-XG repensou a ontologia SSN com base nas lições aprendidas nos últimos anos. Este esforço resultou na ontologia *Sensor, Observation, Sample and Actuator* (SOSA) (JANOWICZ et al., 2018). SOSA fornece uma especificação formal, leve e de uso geral para modelar a interação entre as entidades envolvidas na observação, atuação e amostragem em cenários de IoT.

Em resumo, podemos observar que o uso de modelos baseados em ontologia para lidar com problemas de interoperabilidade semântica na IoT gerou grande interesse dos

pesquisadores. A análise dos resultados revela claramente uma tendência de crescimento ao longo dos anos estudados e demonstra um movimento incipiente na direção da adoção de ontologias leves.

Outra questão importante nesta revisão sistemática diz respeito ao uso de ontologias de fundamentação em IoT. Referindo-se à Figura 13 e considerando a amostra de estudos analisada, observa-se que apenas 15 estudos mencionaram explicitamente o uso de ontologias de fundamentação, em contraste com os 88 estudos que não mencionaram o uso dessas ontologias de alto nível. No entanto, esse tipo de ontologia desempenha um papel importante no estabelecimento de bases conceituais para abordar conceitos fundamentais, como tempo, evento, papel, propriedade etc. Tais distinções fornecidas pelas ontologias de fundamentação são usadas para melhorar a qualidade dos modelos conceituais (GUIZZARDI, 2007). Além disso, enquanto a comunidade da Internet das Coisas parece começar a advogar o uso de ontologias leves e "fáceis de usar", a fim de favorecer a ampla adoção e reutilização, como no estudo mencionado acima (BERMUDEZ-EDO et al., 2016b), aspectos relacionados à concepção de ontologias de domínio, conforme defendido por (ZAMBORLINI; GONÇALVES; GUIZZARDI, 2008), apoiam o uso de ontologias de fundamentação a fim de reproduzir de forma clara e precisa a descrição dos elementos de domínio. Sendo assim, o trade-off de base ontológica, expressividade e tratabilidade computacional, na concepção de ontologias para a IoT, parecem surgir como importantes e atuais desafios desta área.

No que diz respeito às linguagens de representação de conhecimento usadas para a construção de ontologias em IoT, os resultados desse mapeamento mostraram uma predominância do uso de tecnologias da Web Semântica, especialmente OWL (55%), RDF (16%), e XML (2%). A análise revelou uma tendência ao uso de linguagens interpretáveis por máquina, em detrimento de outras abordagens de representação de nível superior (19%), como gráficos, taxonomias e mapas conceituais. Acredita-se que esta tendência esteja relacionada à forte influência dos padrões propostos pelo grupo OGC Sensor Web Enablement (SWE) (BOTTS et al., 2006) (SHETH; HENSON; SAHOO, 2008), que incentivou o uso de padrões de codificação e serviços alinhados às tecnologias semânticas da web para acessar dados de sensores na Internet. SSN-XG, o grupo W3C responsável pelo desenvolvimento da ontologia SSN foi um dos trabalhos influenciados pelas contribuições do grupo SWE da OGC (COMPTON et al., 2012). Como previamente

notado, (JANOWICZ et al., 2018) recentemente apresentou uma revisão completa da ontologia SSN, dando origem à ontologia leve SOSA. Esses trabalhos têm um forte impacto na comunidade científica, e ambos defendem o uso de linguagens interpretáveis por máquina alinhadas aos fundamentos da Web Semântica para lidar com mais eficiência com problemas de interoperabilidade semântica. De fato, as linguagens adotadas nos trabalhos mencionados (OWL e RDF) representam 71% dos trabalhos selecionados nesta revisão.

Particularmente, a ampla adoção de OWL (55%) pela comunidade da Internet das Coisas evidencia, até agora, a escolha por linguagens de nível epistemológico. No entanto, embora seja interessante facilitar o raciocínio lógico automático, esse tipo de linguagem não é adequado para a representação no mundo real, na etapa de modelagem (GUARINO, 2009). O uso de OWL, bem como a UML e a categoria de representação do conhecimento, podem resultar na criação de modelos ambíguos. Pelo contrário, as linguagens de nível ontológico restringem o significado dos elementos do domínio, promovendo a criação de modelos não ambíguos que representam fielmente os elementos do mundo real. No entanto, como esse mapeamento mostrou, o uso de linguagens de nível ontológico ainda é extremamente tímida em IoT, que, por sua vez, se apresenta como uma interessante lacuna de pesquisa.

Independentemente do nível de ontologia, esse mapeamento também revelou que o uso de abordagens metodológicas para construir ontologias ainda é muito tímido e pouco explorado em IoT, mesmo considerando as dificuldades para a construção e manutenção de ontologias. O uso de estruturas metodológicas, preconizado pela disciplina de Engenharia de Ontologia, poderia proporcionar aos desenvolvedores de sistemas de IoT importantes benefícios, como estruturar o processo; reduzindo sua complexidade a tarefas gerenciáveis; esclarecer as responsabilidades dos participantes no processo; aumentando sua rastreabilidade; e possibilitando procedimentos sistemáticos de garantia da qualidade (SIMPERL; TEMPICH, 2006). A Engenharia de Ontologia lida com um conjunto de atividades focadas no processo de desenvolvimento de ontologias, que podem ser analisadas pelo ciclo de vida, metodologias, ferramentas e linguagens para construir ontologias (CORCHO; FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2003).

Foi analisado quais estudos fizeram referência explícita ao uso de uma metodologia para o desenvolvimento de ontologias. Foi possível concluir que a maioria dos estudos (94) não mencionaram explicitamente o uso de metodologias no processo de desenvolvimento da ontologia. Entre os oito estudos que adotaram metodologias, temos o seguinte resultado: *Ontology Development 101* (NOY; MCGUINNESS, 2001); *NeOn* (SUÁREZ-FIGUEROA; GÓMEZ-PÉREZ; FERNÁNDEZ-LÓPEZ, 2012); *Methontology* (FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ; JURISTO, 1997); e *Linked Open Terms*.

Recentemente, (FRUHWIRTH; KASTNER; KRAMMER, 2018) enfatizou a importância de abordagens metodológicas no desenvolvimento de ontologias na IoT. Os autores também enfatizaram a reutilização limitada das ontologias existentes, como observamos neste mapeamento. Na mesma direção, (GYRARD; ZIMMERMANN; SHETH, 2018) destacou a deficiência de abordagens metodológicas na IoT. Conforme mencionado na discussão de linguagens de nível ontológico, essa situação indica a necessidade de novos esforços de pesquisa para abordar a lacuna de pesquisa.

Particularmente no que se refere à reutilização de ontologias, este capítulo mostrou que a maioria dos trabalhos analisados (59%) não indicava explicitamente a reutilização de ontologias. De acordo com (GYRARD; ZIMMERMANN; SHETH, 2018), a reutilização de ontologias tem os benefícios de aumentar a interoperabilidade semântica entre sistemas, pois permite o compartilhamento de conhecimento do domínio usando um vocabulário comum. Isso poderia reduzir o tempo de desenvolvimento de aplicações e, adicionalmente, de acordo com (SIMPERL, 2009), potencialmente melhorar a qualidade das ontologias reutilizadas porque elas são revisadas e avaliadas continuamente por várias partes através da reutilização. Ainda em, (GYRARD; ZIMMERMANN; SHETH, 2018), é proposto o desenvolvimento de um ambiente de catálogo de ontologias e uma metodologia com intuito de incentivar o reuso.

Entre os trabalhos que relataram o reuso de ontologias existentes, a maioria deles (50) adotou a ontologia SSN, reforçando a importância dessa ontologia para a modelagem de domínio de IoT de última geração. Esse resultado, adicionalmente, confirma a tendência apresentada no (WANG; ZHANG; LI, 2015), que discute as maneiras pelas quais as soluções de IoT reutilizaram a ontologia do SSN, a saber: (i) estender o SSN; (ii)

combinar SSN com outra ontologia; e (iii) adicionar novos conceitos ao SSN de maneira a torná-lo mais útil em diferentes domínios.

Os resultados deste capítulo comprovam o interesse crescente e os esforços da comunidade de IoT em direção à interoperabilidade semântica. A análise e as observações realizadas durante a execução deste trabalho também evidenciam alguns aspectos que podem direcionar pesquisas futuras. Em particular, destaca-se ainda o pouco reuso de ontologias existentes com intuito de potencializar a interoperabilidade nas soluções integradas. Mais importante, o conceito de interoperabilidade ainda é restrito a uma única dimensão e deveria ser discutido de uma maneira mais ampla pela comunidade de IoT, considerando diferentes dimensões de interoperabilidade, tal como apresentado em (IZZA, 2009).

Considerando um ambiente de disponibilidade massiva de dispositivos IoT, a interoperabilidade semântica, embora fundamental, não é suficiente para lidar com a interação com os mais variados dispositivos e a maneira de obter, dentro de um universo extenso, os dispositivos em conformidade com os requisitos de aplicações que fazem uso de IoT como parte de sua solução.

Capítulo 4

4. A Arquitetura Proposta

Este capítulo apresenta o projeto da infraestrutura proposta no contexto de Internet das Coisas Massivos. O capítulo se inicia com um breve resumo dos princípios que nortearam a definição da arquitetura conceitual da solução, seguida de uma descrição dos requisitos que guiaram a definição da infraestrutura de implementação concebida (Seção 4.1). A Seção 4.2 apresenta os componentes da arquitetura conceitual proposta. A Seção 4.3 conclui o capítulo.

4.1. Princípios e Requisitos

Como já discutido, a dimensão apresentada pelos ambientes IoT massivos eleva os desafios de pesquisa a um outro patamar, exigindo preocupações adicionais, especialmente relacionadas à ausência de ambientes de suporte à simulação e seleção de dispositivos IoT em um ambiente com disponibilidade extensiva e heterogênea, que permitam analisar as propriedades físicas e contextuais, a fim de viabilizar o desenvolvimento de cenários que possam ser utilizados em diferentes contextos.

Observa-se que há um enorme potencial retido pela dificuldade de escolher um subconjunto de dispositivos de interesse que atendem aos diversos cenários que podem ser explorados por novas aplicações e negócios. Especificamente, nota-se uma carência de soluções que permitam - de uma maneira simples e flexível - incluir novos requisitos ou fazer alterações em tempo real para lidar com novas situações contextuais que podem ocorrer a qualquer momento.

Nesse sentido, propomos uma arquitetura conceitual inserida numa abordagem de desenvolvimento centrada nos seguintes princípios:

- **Uso de uma abordagem de alto nível para representação de cenários um ambiente de Internet das Coisas Massivos.** A abordagem de alto nível deve possibilitar ao especialista do domínio, a definição de regras flexíveis que considere

o estado dos dispositivos de IoT, definindo a estrutura geral de um cenário a ser utilizado pelas aplicações interessadas.

- **Seleção de dispositivos de Internet das Coisas.** A arquitetura deve prover uma infraestrutura de armazenamento de informações sobre os dispositivos, associado a mecanismos de seleção de componentes de Internet das Coisas que possibilite ao usuário escolher os dispositivos que atendem aos requisitos funcionais de um determinado cenário.
- **Interação com dispositivos de Internet das Coisas de maneira facilitada e padronizada, por meio da exposição de suas características e funcionalidades.** A arquitetura conceitual deve prover mecanismos que permitam que dispositivos de Internet das Coisas possam compartilhar uma descrição completa sobre si mesmo, legível por máquina, adotando uma maneira padronizada de exposição de tais informações. Além disso, as funcionalidades de um dispositivo devem estar acessíveis de forma facilitada, com intuito de potencializar sua utilização nos mais variados cenários.

Além desses princípios, a arquitetura proposta deve atender aos seguintes requisitos:

- *R1. Gestão de regras de cenários:* A construção de regras do cenário deve ser realizada por meio de um paradigma de programação flexível, que viabilize a sua descrição em alto nível de abstração. Deve ser possível a definição de um repositório de regras e eventos do domínio. O processamento das regras do cenário deve ser capaz de considerar o estado dos dispositivos de Internet das Coisas e possuir um mecanismo capaz de selecionar os dispositivos que atendem tanto os requisitos funcionais quanto aos seus estados. O ambiente deve, ainda, suportar uma forma de subscrição de cenários, permitindo que aplicações interessadas possam ter acesso a um subconjunto dinâmico de dispositivos que atendem aos critérios que definem um cenário.
- *R2. Gestão de dispositivos de Internet das Coisas:* Deve ser possível ao especialista de domínio gerenciar a descrição de dispositivos de Internet das Coisas, sendo desejável que haja um repositório onde as descrições são registradas e disponibilizadas para consultas. As descrições devem possuir vinculação com conceitos de uma ou mais ontologias a fim de prover uma descrição semântica mais precisa dos componentes e funcionalidades de um determinado dispositivo.

- *R3. Comunicação entre dispositivos de IoT e aplicações:* Deve existir uma interface capaz de estabelecer e manter conexões entre dispositivos de Internet das Coisas e as aplicações. Esta interface deve, ainda, suportar os protocolos da rede à qual está vinculada, ser capaz interpretar a descrição de um dispositivo e possibilitar que aplicações invoquem funcionalidades expostas pelos dispositivos.
- *R4. Linguagem de descrição de dispositivos.* A arquitetura deve suportar uma forma padronizada, flexível, legível por humanos e interpretável por máquina capaz de descrever as características e funcionalidades de um dispositivo, permitindo que seja associado conceitos de ontologias com intuito de enriquecer semanticamente, tornando mais precisa a definição dos elementos que compõe um dispositivo.

4.2. Arquitetura Conceitual

A Figura 20 mostra a arquitetura conceitual da infraestrutura proposta. São identificados os seus componentes, os quais estão distribuídos em três camadas, e os atores do processo, representados nos papéis do especialista de domínio, gestor do catálogo e do desenvolvedor de aplicações no contexto de Internet das Coisas.

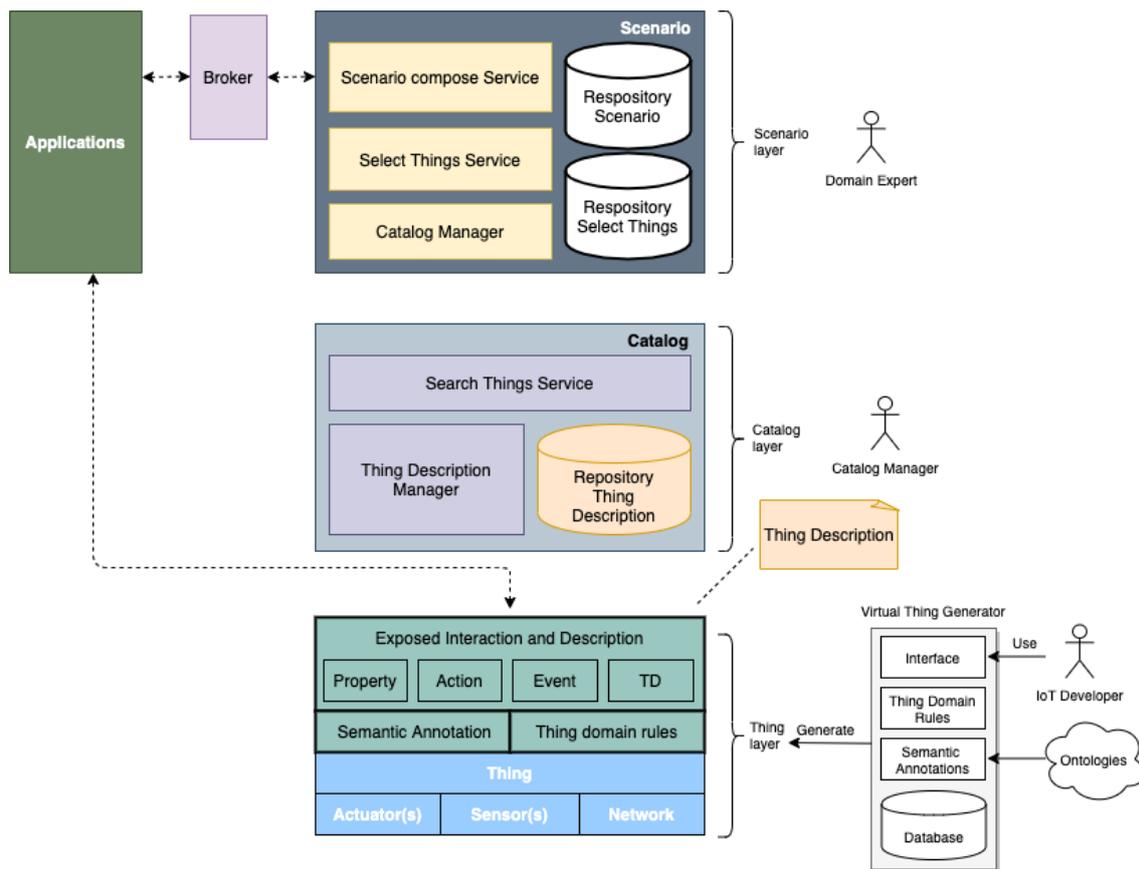


Figura 20 - Infraestrutura SI4-IoT.

A arquitetura conceitual é dividida em três camadas: (i) o dispositivo (*thing*); (ii) um *Catálogo* (*Catalog*); e a (iii) *Camada Cenários* (*Scenario*). Esta divisão visa à separação de responsabilidades entre três importantes papéis: o desenvolvedor IoT que detém conhecimento técnico de desenvolvimento de software e do projeto do hardware para concepção do dispositivo de Internet das Coisas; o gestor do catálogo possui conhecimentos referentes ao armazenamento, busca e seleção de dispositivos; e por fim, o especialista do domínio detém conhecimentos do domínio e das regras de negócio do cenário. Esta abordagem permite a cada um deles usar os seus conhecimentos específicos sem a necessidade de lidar com exigências que não pertencem ao seu campo de especialização (RODRIGUES et al., 2012). As próximas seções discutem em mais detalhes as três camadas supracitadas.

4.2.1. Camada *Thing*

A camada *Thing* tem como foco potencializar o reuso desses dispositivos nos mais diversos contextos. Esta camada é vista sob duas perspectivas, a primeira diz respeito aos componentes de hardware (azul), sendo compostos tipicamente por elementos

computacionais programáveis, com capacidade de processamento, armazenamento, comunicação, sensoriamento e/ou elementos atuadores. A segunda perspectiva são os artefatos de software (verde), cujo objetivo geral é prover um elevado nível de abstração às aplicações, no que se refere à maneira pela qual ocorrem as interações com um dispositivo de Internet das Coisas. Tal abstração é crucial no sentido de facilitar a seleção de dispositivos e o processo de desenvolvimento de aplicações que necessitem obter dados de sensores, eventos e acionar atuadores presentes em dispositivos em um cenário de Internet das Coisas massivos.

Semantic Annotation tem como foco o enriquecimento da maneira de descrever um dispositivo de IoT, por meio da adoção de uma linguagem padrão para descrição de dispositivos que uniformize a sintaxe e possibilite a definição semântica de diversos aspectos de um dispositivo. Na arquitetura proposta, a semântica é concretizada por meio da associação de conceitos de ontologias aos elementos que compõe a Coisa, tornando a definição de tais elementos mais precisa. Essa associação é um processo realizado pelo desenvolvedor de IoT, em que é feita uma análise sobre o conceito mais adequado a ser vinculado, onde é preconizado o reuso de ontologias existentes.

Thing domain rules tem como objetivo a implementação das regras de domínio referentes ao propósito do dispositivo, cujo responsável é o desenvolvedor de IoT. Em outras palavras, a concepção de um dispositivo é comumente o reflexo dos requisitos elencados, a fim de atender as necessidades de um domínio. Nesse sentido, especificidades do domínio influenciam as decisões de implementação das funcionalidades do dispositivo. Um típico exemplo de categoria de funcionalidade revelada pelas características do domínio é o evento. Por exemplo, um dispositivo que possui um sensor de temperatura pode ter a implementação de um evento que notifica aos interessados caso o valor apurado seja maior que um determinado número. É possível observar que os detalhes do evento estão fortemente relacionados ao domínio de aplicação. Sendo assim, invariavelmente, dispositivos de IoT tendem a possuir reflexos do domínio em sua implementação.

Exposed Interaction and Description busca revelar às aplicações interessadas a descrição completa de uma Coisa (*Thing Description - TD*), descrita por meio de uma linguagem de descrição de dispositivos (DDL), explicitando os conceitos associados e a forma pelo qual é possível interagir com suas propriedades, ações e eventos. Essas

informações favorecem o processo de seleção de dispositivos de IoT e promovem um elevado nível de abstração, pois as aplicações precisam apenas conhecer a interface de interação, ou seja, a forma de interagir e os protocolos de comunicação suportados.

4.2.2. *Virtual Thing Generator*

Como frisado, a camada *Thing* explora a abordagem proposta na recomendação *W3C Web of Things* como forma de expor os metadados e o modelo de interação para um protótipo real. A arquitetura proposta nesta tese, entretanto, é projetada para lidar em um contexto de dispositivos IoT massivos. Considerando a dificuldade de reproduzir um ambiente massivo real, a arquitetura propõe um componente arquitetural adicional, um ambiente de simulação denominado *Virtual Thing Generator*, com intuito de simular ambientes IoT massivos. Este componente é capaz de gerar um conjunto extenso de dispositivos virtuais em conformidade com o padrão *W3C Web of Things*, com diversidade de interações conforme a necessidade do usuário.

O modelo de informação proposto para este componente, apresentado na Figura 21, constitui o elemento básico sobre o qual todo o funcionamento do ambiente de simulação se fundamenta. O modelo proposto foi inspirado na recomendação *W3C Web of Things* (KAEBISCH S., 2019).

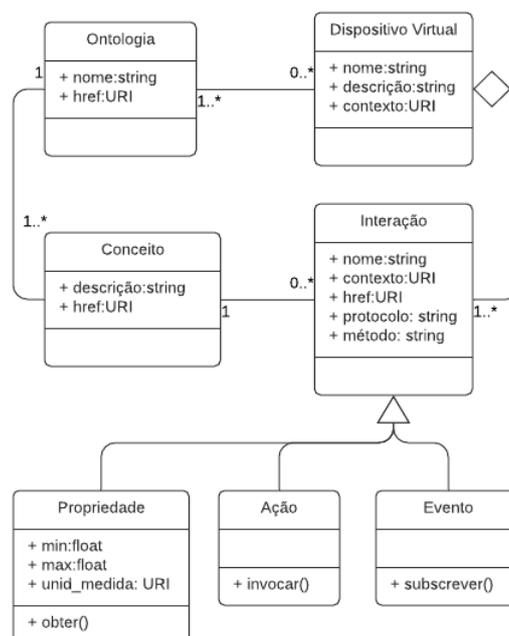


Figura 21 - Modelo de informação.

Neste modelo, a abstração ou conceito de modelagem central é o *dispositivo virtual*. Um dispositivo virtual consiste na representação em software de um elemento do domínio IoT com as mesmas funcionalidades de um dispositivo físico real. Um dispositivo virtual é composto por *interações*, que podem ser entendidas como uma camada de abstração que fornece uma interface e protocolos padronizados a fim de acessar as funcionalidades providas pelo dispositivo IoT. As funcionalidades podem ser dos seguintes tipos: (i) *propriedade* - podem ser usadas para detectar e controlar parâmetros, como obter o valor atual ou definir um estado de uma operação; (ii) *ação* - modelam a invocação de processos físicos, mas também podem ser usadas para abstrair chamadas remotas; ou (iii) *evento* - usados para o modelo de comunicação por *push* em que notificações, eventos discretos ou fluxos de valores são enviados de forma assíncrona para o receptor.

Propriedades fornecem uma interface de abstração ao processo de interação com sensores físicos de um dispositivo. Por exemplo, a obtenção de dados de um determinado sensor requer, muitas vezes, a implementação de funções capazes de interpretar o sinal elétrico oriundo do sensor, obtido através da detecção de modificações de alguma variável de um ambiente, convertendo esses dados em um valor de uma escala conhecida. Neste sentido, essa abstração habilita às aplicações o acesso aos dados de interesse oriundos de um sensor presente em um dispositivo IoT.

Ações fornecem uma abstração quanto ao processo de interação com atuadores físicos de um dispositivo. Um atuador requer um conjunto de instruções específicas a fim de converter sinais elétricos fornecidos como entrada, em ações mecânicas. Por exemplo, um *buzzer* requer sinais elétricos que representem a frequência e o tempo que este atuador deve propagar ondas mecânicas.

Um evento representa um modelo *publish/subscribe* em que aplicações podem se subscrever a determinados eventos ou sequência de episódios de interesse. Por exemplo, a ocorrência de uma arritmia cardíaca detectada a partir dos dados coletados dos dispositivos.

No modelo de informação proposto, um dispositivo virtual deve estar sempre associado a *ontologias*, que constituem a abordagem semântica que dá o suporte necessário para a definição formal e precisa dos elementos que compõem o dispositivo virtual.

Por sua vez, uma *interação* é entendida como uma abstração que fornece uma interface e protocolos de comunicação padronizados para de acessar as funcionalidades providas pelo dispositivo IoT, tais como seus sensores, atuadores e eventos. Cada

interação também deve estar associada a um conceito de uma ontologia. Neste trabalho, ontologias são representadas por vocabulários OWL - *Web Ontology Language* (MCGUINNESS; VAN HARMELEN; OTHERS, 2004).

A arquitetura proposta para o ambiente de geração massiva de dispositivos IoT virtuais é ilustrada na Figura 22, sendo constituída de dois elementos principais: o *Virtual Thing Generator* e o *Virtual Thing*. O *Virtual Thing Generator* é o elemento da arquitetura responsável por todo processo de geração de dispositivos virtuais. Por meio do componente *Interface* o usuário fornece os parâmetros requeridos pelo seu algoritmo de geração, tais como o local onde serão armazenados os arquivos gerados referentes aos dispositivos virtuais, a quantidade desejada de dispositivos e a quantidade máxima de interações (propriedades, ações e eventos) que um dispositivo virtual deve possuir.

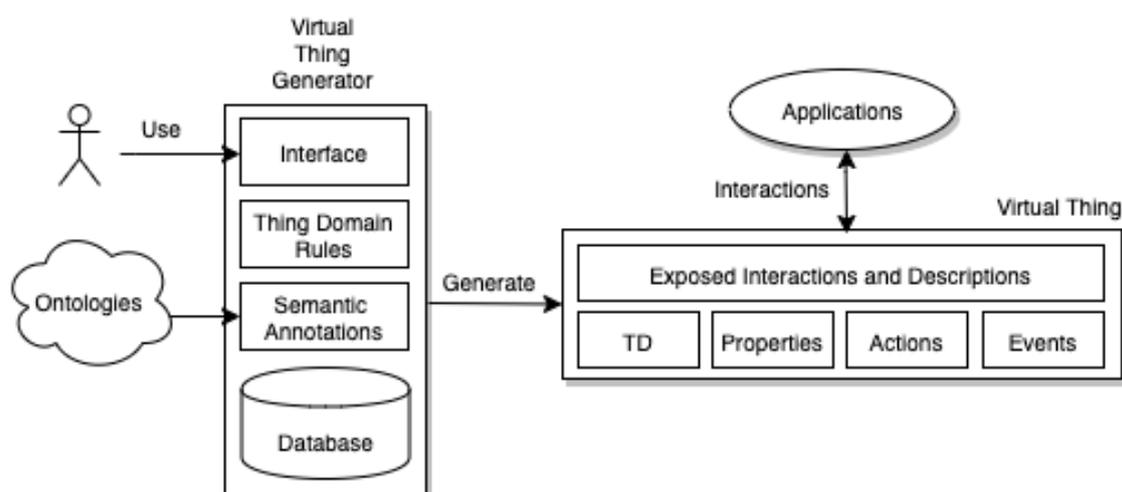


Figura 22 - Visão geral da arquitetura proposta.

Conjuntos de *Interações* são previamente registradas no *Database* do *Virtual Thing Generator*, sendo possível adaptá-los conforme os cenários idealizados. Por exemplo, o usuário é capaz de definir quais são as características internas dos *Virtual Things* a serem gerados pelo ambiente, como o universo de sensores, atuadores e eventos que podem compor o dispositivo gerado. A escolha dessas características deve favorecer a capacidade de definir um cenário IoT mais fidedigno ou até ampliar o universo de possibilidades, propiciando o desenvolvimento de novas soluções.

Thing domain rules diz respeito às regras do domínio referente ao propósito do dispositivo. A concepção de um dispositivo é comumente o reflexo dos requisitos

elencados a fim de atender às necessidades de um domínio. Nesse sentido, especificidades do domínio influenciam as decisões de implementação das funcionalidades do dispositivo. Uma típica interação revelada pelas características do domínio é o evento. Por exemplo, uma arritmia cardíaca, pode ser definida por um conjunto de fatores materializados ou descritos por regras do domínio, e envolve o uso de um dispositivo com sensores capazes de monitorar as variáveis de interesse. A implementação do evento notifica os interessados caso o episódio monitorado seja identificado. É possível observar neste exemplo como os detalhes do evento estão fortemente relacionados ao domínio de aplicação. Sendo assim, invariavelmente, dispositivos de IoT tendem a possuir reflexos do domínio em sua implementação.

O componente *Semantic Annotation*, tem como foco o enriquecimento semântico da descrição de um dispositivo IoT. Para tal, ele faz uso de uma linguagem padrão de descrição de dispositivos, que uniformiza a sintaxe e possibilita a definição semântica dos diversos aspectos de interesse do dispositivo. Na arquitetura proposta, a semântica é concretizada por meio da associação de conceitos de ontologias aos elementos que compõe a Coisa (*Thing*), tornando a definição de tais elementos mais precisa. Essa associação é um processo realizado pelo desenvolvedor de IoT, que efetua uma análise sobre o conceito mais adequado a ser vinculado. Nesta etapa, é preconizado o reuso de ontologias existentes, tais como SAREF (DANIELE; DEN HARTOG; ROES, 2015), M3-Lite (AGARWAL et al., 2016) e OM (RIJGERSBERG; VAN ASSEM; TOP, 2013). Por exemplo, a ontologia SAREF define o conceito de Sensor (*saref:Sensor*²) e atuador (*saref:Actuator*³), que podem utilizados para definir semanticamente as propriedades e ações, respectivamente. Cada sensor pode ter um valor de retorno dentro de uma escala de valores pré-determinada e, neste sentido, o uso de ontologias como OM⁴ podem contribuir para a definição precisa do tipo de escala a ser representada.

O *Virtual Thing* é o produto final do ambiente proposto, sendo gerado a partir dos parâmetros definidos pelo usuário no componente *Virtual Thing Generator*. Ele consiste na representação virtual de um dispositivo IoT, sendo composto tipicamente por elementos computacionais que simulam e capacidade de processamento, comunicação, sensoriamento e/ou elementos atuadores presentes em um dispositivo IoT. Ou seja, o *Virtual Thing* é um artefato de software que possibilita às aplicações a manipulação de

² https://ontology.tno.nl/saref/saref_Sensor.html

³ https://ontology.tno.nl/saref/saref_Actuator.html

⁴ <https://enterpriseintegrationlab.github.io/icity/OM/doc/index-en.html>

dispositivos reais e heterogêneos. Como visto, tal abstração é crucial em um cenário de Internet das Coisas massivos.

O *Virtual Thing* é exposto para as aplicações por meio do componente *Exposed Interaction and Description*, que busca revelar às aplicações interessadas a descrição completa de uma Coisa (*Thing*). Esta descrição (*Thing Description - TD*) é realizada através de uma linguagem de descrição de dispositivos (DDL), explicitando os conceitos associados e a forma pela qual é possível interagir com as suas propriedades, ações e eventos. Estas informações favorecem o processo de seleção de dispositivos de IoT e promovem um elevado nível de abstração, pois as aplicações precisam apenas conhecer a interface de diálogo, isto é, a forma de interagir e os protocolos de comunicação suportados.

Finalmente, a arquitetura proposta provê o acesso às interações e descrições disponibilizadas pelo dispositivo IoT por meio dos protocolos HTTP, CoAP e MQTT. A adoção de cada protocolo é uma decisão particular de cada desenvolvedor IoT, conforme os requisitos elencados e o propósito do dispositivo IoT.

4.2.3. Camada Catálogo

O cenário de dispositivos massivos impõe uma questão particular que diz respeito à maneira pelo qual é possível tornar os dispositivos acessíveis aos usuários. Tal cenário demanda um ambiente que permita o registro, o compartilhamento e possibilite processar operações de pesquisa de descrições das Coisas. Este ambiente, com a funcionalidade de um catálogo, permite que os usuários conheçam os atributos, funcionalidades e formas de acesso antes mesmo da interação com o dispositivo propriamente. Tal característica é fundamental, pois auxilia no processo de seleção de dispositivos que atendem aos requisitos funcionais de interesse.

Na arquitetura proposta, uma camada intermediária, denominada Catálogo tem como função prover uma estrutura capaz de fornecer um maior nível de abstração para as aplicações, no que se refere à seleção de dispositivos de IoT que atendam aos critérios do domínio de interesse, especificamente relacionado aos aspectos físicos (sensores e atuadores). Essa abstração é concretizada por meio da oferta uma infraestrutura que

suporte o registro de descrição de dispositivos de Internet das Coisas e um mecanismo que viabilize o processo de seleção de dispositivos.

A arquitetura do componente Catálogo é mostrada na Figura 23.

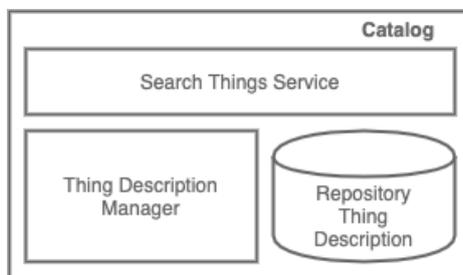


Figura 23 - Catálogo.

O *Thing Description Repository* define uma estrutura capaz de armazenar informações dos dispositivos de IoT, considerando sua enorme diversidade. Tal característica implica na necessidade de suportar uma estrutura flexível de armazenamento, permitindo que diferentes dispositivos com sensores e atuadores distintos sejam descritos, armazenados e descobertos.

O *Thing Description Manager* é responsável por realizar o registro, exclusão e atualização das descrições fornecidas pelos dispositivos. Este componente é capaz de interagir com os dispositivos a fim de obter os metadados necessários e persisti-los no componente *Thing Description Repository*.

O *Search Thing Service*, por sua vez, tem como objetivo fornecer uma maneira de selecionar dispositivos existentes no *Thing Description Repository*, por meio de uma linguagem de consultas que se utilize dos conceitos de ontologias associados aos elementos que compõe uma Coisa.

Apesar desta camada ter um conjunto pequeno de componentes, esta é uma camada central, tendo uma importância crucial na solução proposta, visto que ela mantém um extenso repositório de metadados de dispositivos IoT, permitindo que aplicações possam selecionar quais dispositivos atendam os requisitos funcionais (características de interesse) dentro de um universo massivo de disponibilidade de dispositivos IoT.

4.2.4. Camada Scenario

Na arquitetura proposta, o conceito de Cenário foi introduzido com o objetivo de se definir formalmente as entidades e contextos em um universo extenso e dinâmico, facilitando explorar situações particulares de interesse. Os cenários focam, portanto, na dinamicidade das situações dentro domínio IoT, onde existe uma abundância de Coisas e uma infinidade de possíveis estados de interesse da realidade por parte das aplicações. De fato, de acordo com o dicionário Cambridge⁵, o conceito de cenário é definido como “uma descrição de possíveis ações ou eventos no futuro”. De acordo com (GUIZZARDI; FALBO; GUIZZARDI, 2008), eventos podem alterar o estado de coisas da realidade de um (pré)estado para outro (pós-estado), implicando, assim, em possíveis transformações de uma situação para outra na realidade de interesse.

A camada Scenario materializa, portanto, o conceito abstrato de cenário e permite a sua especificação e o seu gerenciamento. A camada tem como finalidade disponibilizar às aplicações uma infraestrutura que suporta todo o processo de concepção de um cenário, incluindo as etapas de seleção de catálogos, seleção de dispositivos e composição de regras do domínio.

A organização desta camada e suas interações é mostrada na Figura 24.

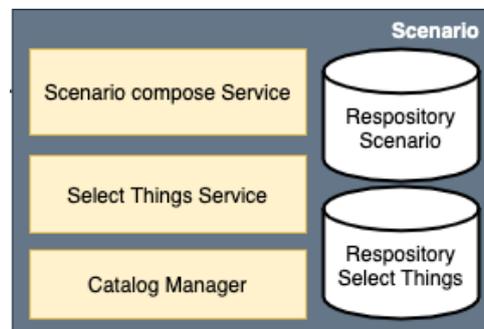


Figura 24 - Componentes da Camada Scenario.

O *Catalog Manager* disponibiliza ao usuário o gerenciamento de catálogos que possuem registros de descrição de dispositivos de IoT, com intuito de prover para a camada superior (*Select Things Service*) uma abstração no que se refere a consulta de dispositivos.

⁵ <https://dictionary.cambridge.org/pt/dicionario/ingles/scenario>

O *Select Things Service* tem como propósito o registro de requisitos físicos de dispositivos de IoT quem atendam a um determinado cenário de interesse, por meio do processo de busca e seleção de dispositivos em um conjunto de repositórios, sendo responsável por prover estratégias de consulta que potencialize a seleção de dispositivos que atendam aos requisitos funcionais do cenário a ser construído.

O *Scenario Compose* provê um ambiente dinâmico de descrição de regras de um cenário, adotando uma linguagem de alto nível capaz de viabilizar a participação do especialista do domínio. Essas regras têm como intuito de analisar o estado atual das capacidades de um dispositivo de IoT, complementando o processo de seleção de dispositivos que atendem ao cenário idealizado. Os cenários são armazenados em no *Scenario Repository*, consistindo na relação entre a seleção de dispositivos que atendam aos requisitos físicos e o conjunto de regras capaz de analisar o estado dos dispositivos. O resultado é um subconjunto de dispositivos que compõe um determinado cenário de interesse, sendo disponibilizados às aplicações.

4.2.4.1. Definindo Cenários

A natureza dinâmica dos cenários num extenso universo de Coisas revela uma necessidade de investigar maneiras de descrevê-los formalmente e de definir uma estratégia para implementá-los computacionalmente. Em uma boa modelagem, a relação entre os objetos físicos e as entidades matemáticas correspondentes é a mais clara e a mais direta possível (CARDOSO; VALETTE, 1997). Nesta tese, as Redes de Petri (PETRI, 1962) foram escolhidas como abordagem de modelagem de cenários.

Redes de Petri possibilitam definir a semântica dos lugares, transições e fichas. Interpretar uma rede de Petri implica, antes de tudo, dar um sentido concreto a um modelo matemático, associando, aos lugares, transições e fichas, os elementos existentes no sistema. Neste trabalho, lugares são interpretados como: (i) uma lista extensa de dispositivos IoT (catálogo); (ii) um subconjunto de dispositivos que atendam aos requisitos funcionais do domínio e por fim; (iii) um subconjunto de dispositivos cujos estados de interesse atendam ao cenário idealizado.

As transições, por sua vez, são associadas aos eventos de inclusão de novos dispositivos na base do catálogo ou em um novo estado do dispositivo pré-selecionado.

As fichas são interpretadas como indivíduos cujo valor é variável, sendo cada ficha associada a uma estrutura de dados, representada por um objeto. Cada objeto pertence a uma classe de objetos à qual são definidas propriedades (atributos).

Rede de Petri a Objetos foi adotada como instrumento de formalização do conceito de cenário, sendo definida pela 9-upla:

$$N_0 = \langle P, T, C_{\text{class}}, V, P_{\text{re}}, P_{\text{ost}}, A_{\text{tc}}, A_{\text{ta}}, M_0 \rangle,$$

Onde:

- C_{class} é um conjunto finito de objetos, eventualmente organizado em uma hierarquia e definindo para cada classe um conjunto de atributos;
- P é um conjunto finito de lugares cujos tipos são dados por C_{class} ;
- T é um conjunto finito de transições;
- V é um conjunto de variáveis cujos tipos são dados por C_{class} ;
- P_{re} é a função lugar precedente que a cada arco de entrada de uma transição faz corresponder uma soma formal de n-uplas de elementos de V ;
- P_{ost} é a função lugar seguinte que a cada arco de saída de uma transição faz corresponder uma soma formal de n-uplas de elementos de V ;
- A_{tc} é uma aplicação que a cada transição associa uma condição fazendo intervir as variáveis formais associadas aos arcos de entrada aos atributos das classes correspondentes;
- A_{ta} é uma aplicação que cada transição associa uma ação fazendo intervir as variáveis formais associadas aos arcos de entrada aos atributos das classes correspondentes;
- M_0 é a marcação inicial que associa a cada lugar uma soma formal de n-uplas de instâncias de objetos.

Com intuito de exemplificar a formalização do conceito cenário, vamos considerar o caso a seguir:

Uma aplicação deseja selecionar dispositivos que possuam um sensor de temperatura e que este retorne valores na escala célsius, cujo valor de interesse seja superior a 22°.

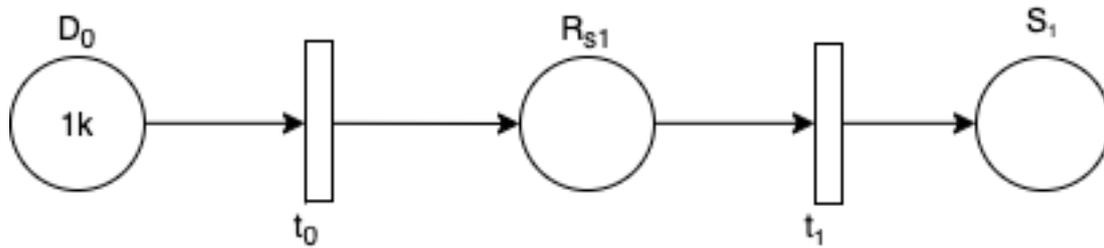


Figura 25 - Rede de Petri representando um cenário.

A Rede de Petri ilustrada pela Figura 25 é formalmente descrita como:

Um conjunto de lugares, $P = \langle D_0, R_{s1}, S_1 \rangle$, onde:

- D_0 representa a lista de dispositivos IoT disponíveis em um catálogo.
- R_{s1} representa a lista de dispositivos IoT que atendem aos critérios de seleção de características físicas, descrito em t_0 .
- S_1 ilustra o subconjunto de estados de interesse dos dispositivos IoT presentes no lugar R_{s1} , sendo os estados de interesse definidos em t_1 .

Um conjunto de transições, $T = \langle t_0, t_1 \rangle$, onde:

- t_0 define as regras de sensibilização que é mapeado para uma *string* de busca em conformidade com o *Thing Description Repository*.
- t_1 define as regras de sensibilização que é mapeado para um conjunto de regras a ser utilizados a fim de definir os estados de interesse dos dispositivos presentes em R_{s1} .

Um conjunto de variáveis V , cujos tipos são dados por Classes $C_{class} = \{\text{Dispositivo}, \text{Interação}\}$ com:

- O conjunto de classes $C_{class} = \{\text{Dispositivo}, \text{Interação}\}$ com:

$$\text{Dispositivo} = \begin{cases} \text{nome: identificador do dispositivo} \\ \text{descrição: descreve o dispositivo} \\ \text{contexto: lista de ontologias} \\ \text{interação: lista de interações} \end{cases}$$

$$\text{Interação} = \begin{cases} \text{tipo: conceito ontológico que o define} \\ \text{descrição: descreve a interação} \\ \text{unidade de medida: conceito que define o valor retornado} \end{cases}$$

- As variáveis x do tipo Dispositivo: $\text{tipo}(x) = \text{Dispositivo}$.

A matriz Pre é a função precedente a cada arco de entrada de uma transição.

$$Pre = \begin{bmatrix} \langle x \rangle & 0 \\ 0 & \langle x \rangle \\ 0 & 0 \end{bmatrix}$$

A matriz Post é a função seguinte a cada arco de saída de uma transição.

$$Post = \begin{bmatrix} 0 & \langle x \rangle \\ \langle x \rangle & 0 \\ 0 & 0 \end{bmatrix}$$

Uma etapa crucial é definir as regras que governam a Rede de Petri, em outras palavras, como ocorrerá a sensibilização das transições, no que diz respeito à seleção das: (a) características de interesse; e (b) os estados de interesse.

a) Requisitos funcionais dos dispositivos:

- $A_{tc}(t_0)$
 - $O_1 = \{\text{https://w3id.org/saref}, \text{http://www.wurvoc.org/vocabularies/om-1.8}\}$
 - $I_1 = \{\text{saref:Temperature}\}$
 - $U_m = \{\text{om:Celsius_temperature_scale}\}$
 - $A_{tc}(t_0) = (x.\text{contexto} \in O_1) \wedge (x.\text{interação.tipo} \in I_1) \wedge (x.\text{interação.unidade de medida} \in U_m)$

Nesse sentido, a transição t_0 será disparada se o dispositivo contido em D_0 possuir, dentro do contexto que os definem as ontologias *Saref*⁶ e *Om*⁷ e possua uma interação do tipo *saref:Temperature* e sua unidade de medida seja *om:Celsius_temperature_scale*.

b) Estados de interesse:

- $A_{tc}(t_1)$
 - $A_{tc}(t_1) = x.\text{interação.temperatura} > 22$

⁶ <https://w3id.org/saref>

⁷ <http://www.wurvoc.org/vocabularies/om-1.8>

Por fim, a transição t_1 será sensibilizada caso o dispositivo possua uma temperatura superior a 22°C.

4.3. Considerações do capítulo

Este capítulo apresentou uma visão geral dos elementos que compõe a arquitetura conceitual proposta nesta tese. Como forma de validar funcionalmente os componentes da arquitetura, uma primeira implementação foi realizada e conduzidos uma série de experimentos práticos para avaliação da sua performance em diferentes condições de uso.

O Capítulo 5 a seguir discute as escolhas tecnológicas utilizadas na implementação da arquitetura e detalha os experimentos realizados. Um estudo de caso em um cenário de Contramedidas da Poluição do Ar, usando a implementação da arquitetura aqui descrita, é apresentada no Capítulo 6.

Capítulo 5

5. Implementação da arquitetura SI4-IoT

Este capítulo apresenta os detalhes de implementação da arquitetura conceitual SI4-IoT, cujo principal objetivo é superar os desafios indicados nas seções anteriores. A seção 5.1 apresenta a visão geral da arquitetura de implementação. A seção 5.2 trata da implementação da Camada Thing, demonstrando as tecnologias utilizadas por meio da prototipação de um dispositivo físico e adoção de uma abordagem de descrição de dispositivos. A seção 5.3 apresenta uma implementação da camada catálogo e introduz um ambiente de simulação de dispositivos massivos. Por fim, a seção 5.4 mostra a implementação da camada Cenário, demonstrando o suporte à composição de cenários em Internet das Coisas e discute os resultados obtidos.

5.1. Arquitetura de implementação

A arquitetura de implementação foi desenvolvida com base na arquitetura conceitual, empregando opções tecnológicas adequadas ao propósito da infraestrutura proposta, e considerando o objetivo de simplificar o trabalho do desenvolvedor de aplicações finais.

As tecnologias de implementação usadas na arquitetura aplicada são representadas na Figura 26. O detalhamento das camadas e as escolhas tecnológicas realizadas em cada uma delas são apresentadas a seguir.

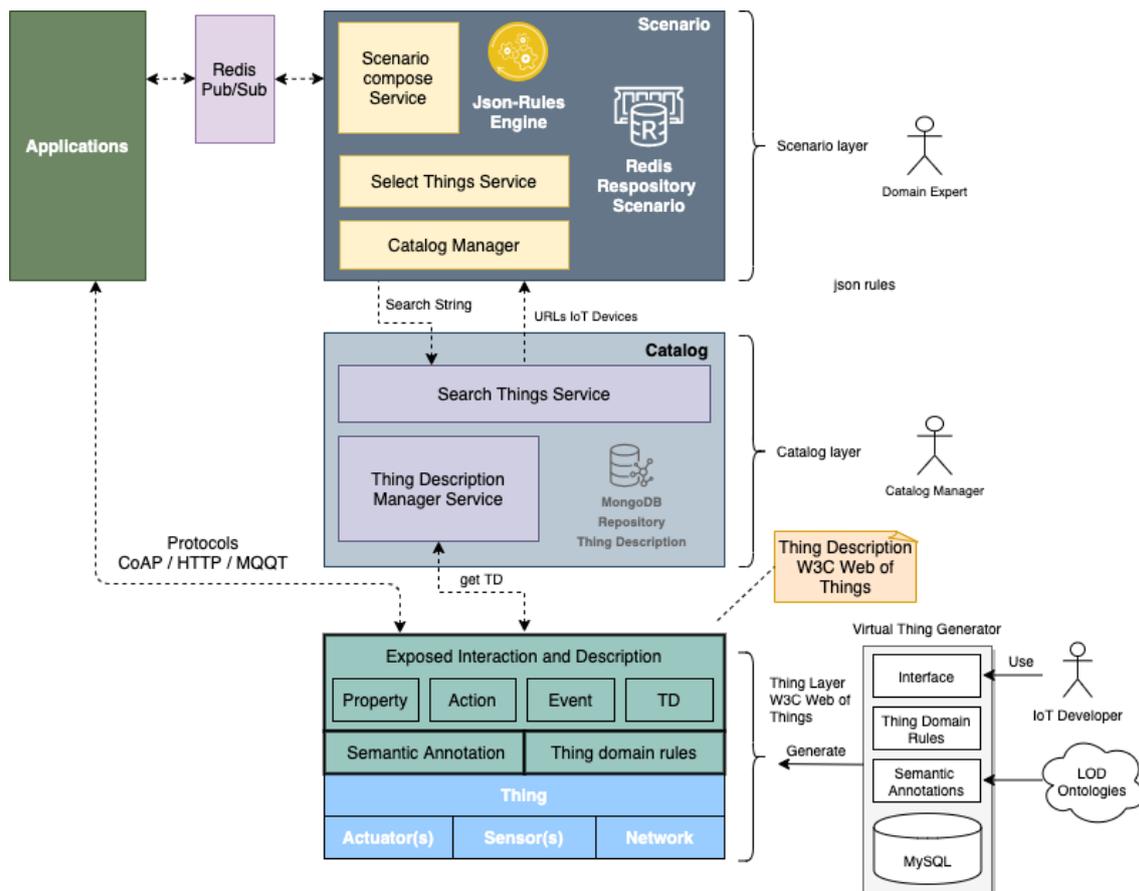


Figura 26 - Arquitetura de implementação.

A arquitetura de implementação adota a recomendação W3C *Web of Things* na sua camada mais fundamental, descrita no Capítulo 2. Esta escolha justifica-se pela plena aderência deste padrão aos princípios e requisitos estabelecidos para a descrição e a manipulação de Coisas no universo IoT massivo, em especial dois deles:

- Existência de um modelo para descrever as capacidades e funcionalidades dos dispositivos IoT. O *Thing Description* (TD) é um elemento central da recomendação *Web of Things* e pode ser considerado como o ponto de entrada de uma Coisa, sendo que, por meio da obtenção do TD, é possível que aplicações possam conhecer, em mais detalhes, as características e capacidades de um dispositivo IoT, ou Coisa. Uma instância de TD possui quatro componentes principais: metadados textuais sobre o dispositivo (*Thing*), um conjunto de formas de acesso à interação, que indicam como ele pode ser usado, esquemas para os dados a serem trocados com o dispositivo, com intuito de facilitar a legibilidade por máquina e, finalmente, links da Web para expressar qualquer relação formal ou informal com outras Coisas ou documentos na Web.

- Existência de um aspecto homogêneo no que se refere a maneira de interagir com os mais variados dispositivos IoT; O Modelo de Interação do *W3C WoT* define três tipos de recursos de interação: (i) Propriedades - podem ser usadas para detectar e controlar parâmetros, como obter o valor atual de um sensor ou definir um estado de uma operação; (ii) Ações - modelam a invocação de processos físicos, como acionar atuadores; e (iii) Eventos - usados para o modelo de comunicação por *push* em que notificações, eventos discretos ou fluxos de valores são enviados de forma assíncrona para o receptor.

5.2. Realização da Camada Thing

Foi realizada uma implementação da especificação *W3C Web of Things* em um dispositivo real, uma atividade de importância fundamental para a avaliação prática do padrão, mas ainda pouco explorada na literatura (DATTA; BONNET, 2018; KÄBISCH; ANICIC, [s.d.]). A implementação teve a finalidade de investigar e demonstrar as principais características da arquitetura proposta e da linguagem *Thing Description* (TD).

O dispositivo (“*Thing*”) construído para este teste é composto de um ESP8266 Node Mcu LoLin v3; um *passive buzzer*; um LED; um DHT11 - um sensor de temperatura e umidade com faixa de leitura de 0 a 50°C (com precisão de $\pm 2,0^{\circ}\text{C}$) e 20 a 90% umidade relativa (com precisão de $\pm 5,0\%$); e um DHT22 - um sensor de temperatura e umidade com faixa de leitura de -40 a 80°C (com precisão de $\pm 0,5^{\circ}\text{C}$) e 0 a 100% umidade relativa (com precisão $\pm 2,0\%$). A Figura 27 apresenta um esquema do circuito construído.

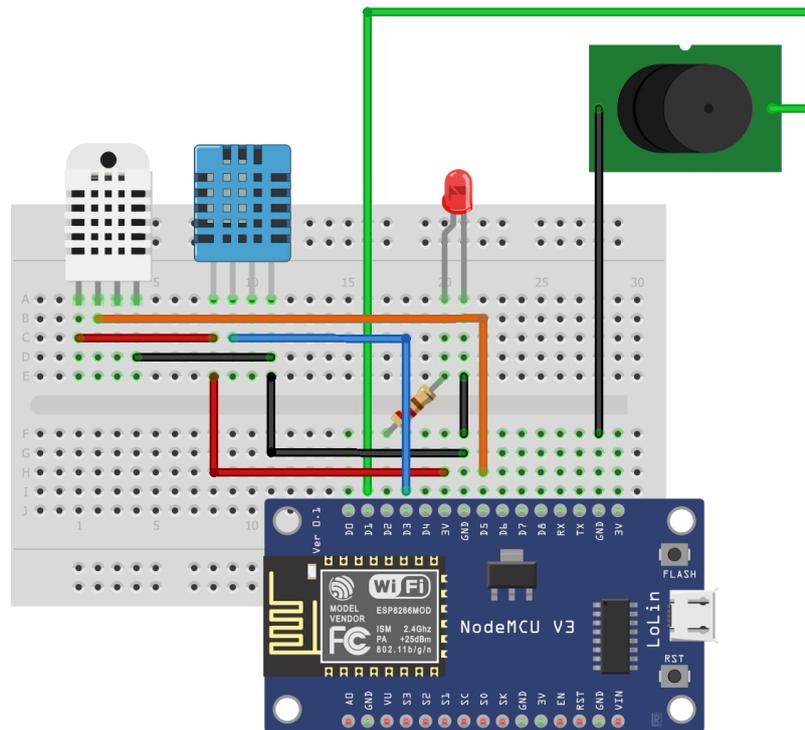


Figura 27 - Dispositivo utilizado na implementação da especificação do W3C *Thing Description*.

Conforme já mencionado, o W3C *Thing Description* (TD) tem como objetivo expor as características e funcionalidades de um dispositivo de forma que possa ser interpretado por máquina, adotando como formato o padrão de serialização JSON-LD 1.1. Nesta implementação de um estudo de caso, a especificação do TD foi adotada para prover a descrição geral do dispositivo, incluindo as ações ou atuadores (Led e o Buzzer), as propriedades (sensores de temperatura e umidade) e seus eventos.

O algoritmo presente no NodeMCU, além da implementação da lógica necessária para prover o funcionamento de seus componentes (ações, propriedades e eventos), tornando-os acessíveis por meio de um protocolo conhecido, também possui a capacidade de compor e expor o TD. A exposição do *Thing Description* ocorre por meio do protocolo HTTP; assim, quando o usuário desejar conhecer todas as características do dispositivo, basta realizar uma requisição utilizando o método GET do protocolo HTTP no endereço do mesmo. A Figura 28 ilustra a composição simplificada do TD.

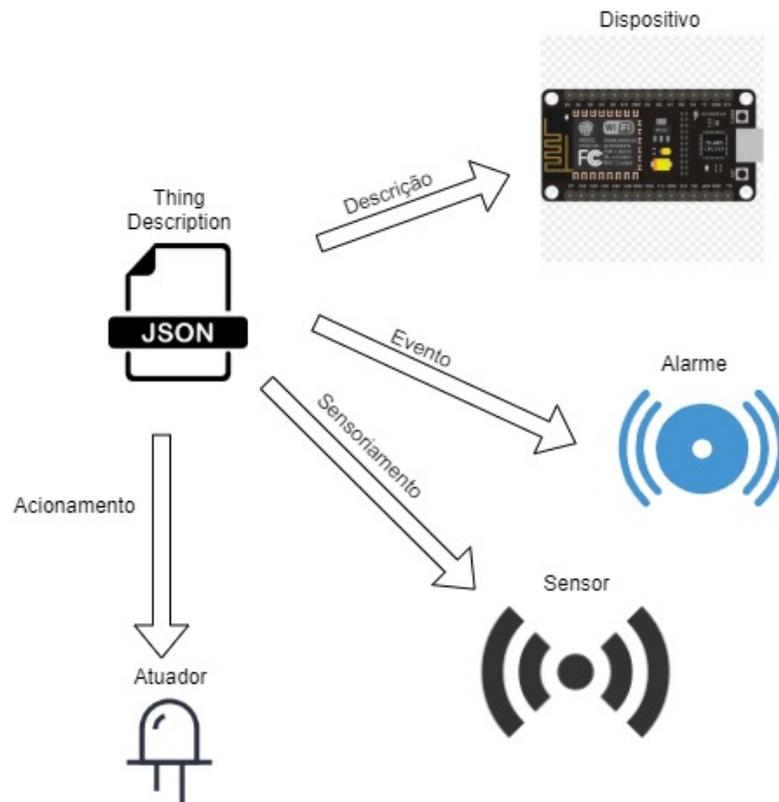


Figura 28 - W3C WoT - Thing Description.

Ao receber o *WoT Thing Description*, o usuário é capaz de compreender a maneira pelo qual será possível interagir com as funcionalidades presentes no dispositivo. As ações descritas no TD possuem suas características e métodos de interação descritas, além de informar qual componente é controlado e o quais são as ações disponíveis. A Figura 29 apresenta um trecho do *WoT Thing Description*, exposto pelo dispositivo construído, que ilustra uma ação que permite ao usuário interagir e realizar a mudança de estado do LED. Este trecho explicita a descrição da ação, os estados possíveis, o protocolo, método e o endereço onde é possível invocar a funcionalidade disponibilizada.

```

"toggle": {
  "description": "Turn on or off the led",
  "contentType": "string",
  "enum": [
    "on",
    "off"
  ],
  "forms": [
    {
      "href": "coap://IP:5683/lightled",
      "cov:methodName": "POST"
    }
  ]
}

```

Figura 29 - Fragmento do WoT Thing Description - Action.

A Figura 30 apresenta um diagrama de sequência que descreve, em mais detalhes, o fluxo de atividades realizadas do usuário para o dispositivo, conforme a implementação executada. O controle de estado do LED, no estudo de caso, é utilizado para demonstrar o conceito de ação: o usuário pode controlar o estado de um LED utilizando o método POST via protocolo CoAP, passando os parâmetros “*on*” e “*off*”, sendo possível controlar o LED de acordo com sua necessidade (1). O dispositivo recebe a requisição e executa a lógica interna para realizar a ação determinada pelo usuário (2). Após a execução, um *feedback* com o novo estado do LED é gerado pelo dispositivo para informar o usuário o novo estado do LED (3). O *feedback* é estruturado em formato Json e enviado para o usuário requisitante por meio do protocolo CoAP (4).

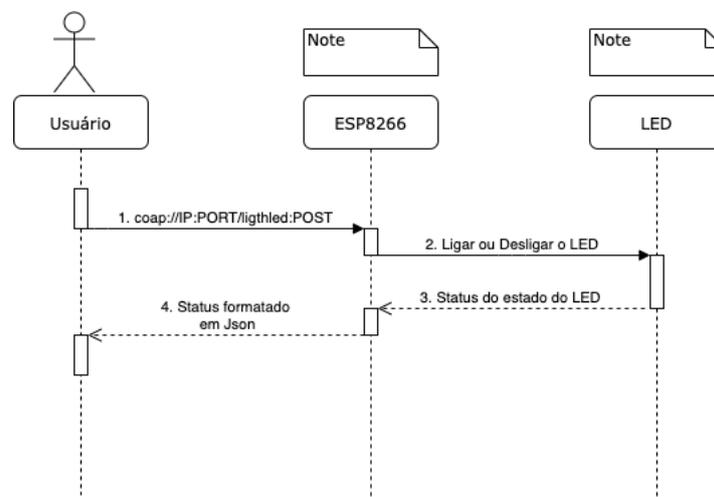


Figura 30 - Interaction Affordance - Action

Sensores são uma parte fundamental dos dispositivos que compõe a Internet das Coisas e a facilitação no processo de obtenção de dados e o entendimento de sua origem potencializam a sua utilização em diferentes contextos, viabilizando o desenvolvimento de novas soluções. Nessa direção, os sensores presentes no dispositivo deste estudo de caso foram expostos em conformidade preconizado na especificação W3C *Web of Things*. A Figura 31 apresenta um fragmento do *Thing Description* da implementação realizada.

```

"status-dht11": {
  "description": "Shows the current temperature and humidity of the sensor DHT11",
  "type": "application/json",
  "forms": [
    {
      "op": "readproperty",
      "href": "coap://IP/dht11",
      "cov:methodName": "GET",
      "unit": "°C and % of humidity of the air"
    }
  ]
},

```

Figura 31 - Fragmento do Thing Description – property.

Como pode ser visto na Figura 31, a propriedade *status-dht11* retorna a temperatura e umidade, referente ao sensor DHT11, em formato *json*, sendo acessível por meio do protocolo *CoAP* através do método *GET* à URL *coap://IP/dht11*.

Deste modo, tal implementação possibilita um maior nível de abstração, pois o sensor fica acessível como um serviço, presente no dispositivo, descrito pelo TD. O diagrama de sequência que demonstra este cenário está representado na Figura 32. Uma requisição de temperatura é usada para demonstrar o acesso às propriedades de um dispositivo no estudo de caso: o usuário inicia o processo, utilizando método *GET* do protocolo *CoAP* para fazer a requisição ao dispositivo conectado (1). O dispositivo conectado à rede recebe a solicitação e executa a lógica necessária para a obtenção da temperatura e umidade (2) fornecidos pelo sensor *dht11*, o qual retorna a temperatura em graus célsius e a porcentagem de umidade relativa do ar (3) lidos do ambiente ao qual o dispositivo está localizado. Ao receber as leituras, o dispositivo processa os valores recebidos em formato *Json* e os envia ao usuário que o requisitou usando o protocolo *CoAP* (4).

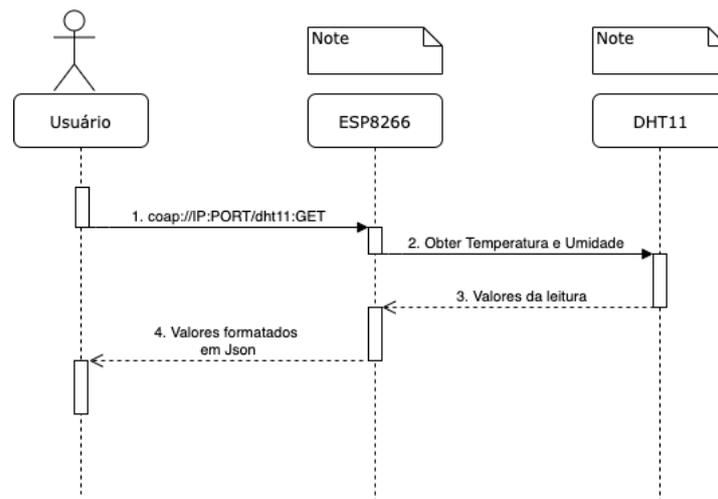


Figura 32 - Interaction Affordance - Properties

Expandindo o exemplo acima, o diagrama de sequência representado pela Figura 33 ilustra o processo pelo qual acontece o múltiplo sensoriamento, demonstrando as diferentes propriedades implementadas, acessíveis por meio do protocolo CoAP. Como já visto, o dispositivo é composto por dois sensores de temperatura e umidade (dht11 e dht22). Tais sensores possuem diferentes faixas de leituras e sensibilidades, o que pode retornar diferentes resultados em um mesmo ambiente.

O usuário deverá escolher a URL que representa o sensor ao qual tem interesse em obter as leituras e executar uma requisição utilizando o método GET do protocolo CoAP. O dispositivo executa a lógica necessária para realizar a leitura do sensor escolhido e faz a adequação do resultado em formato Json. Ao final o dispositivo envia o resultado ao usuário via protocolo CoAP.

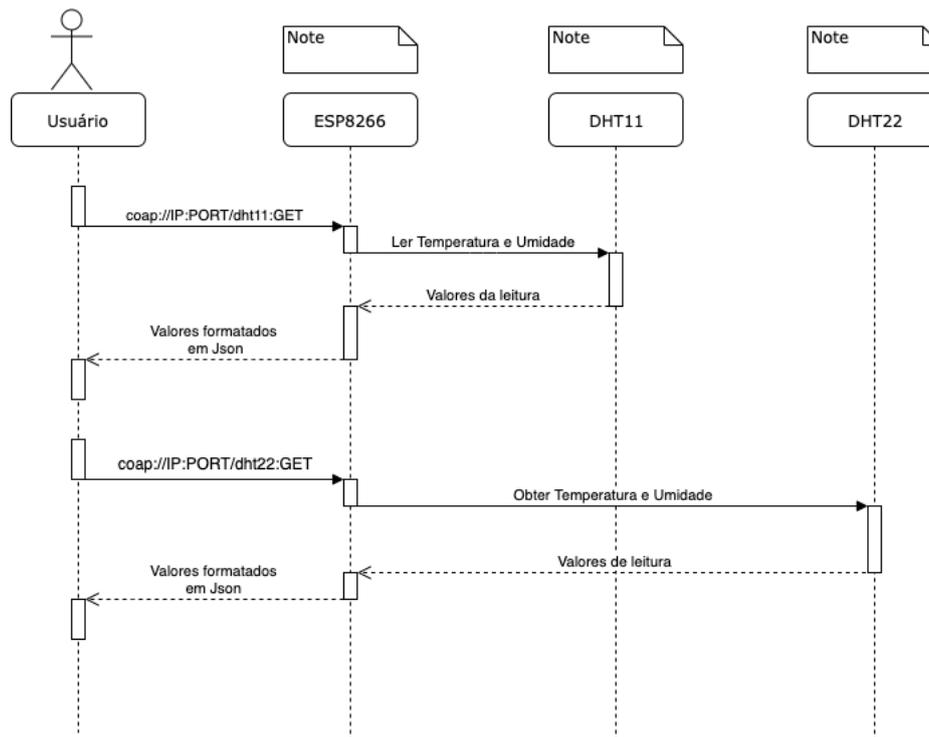


Figura 33 - Interaction Affordance – Properties with two sensors

Por fim, para exemplificar o conceito *evento* – presente na especificação W3C *Web of Things*, foi implementado um evento cujo objetivo é enviar uma notificação ao usuário caso a temperatura aferida seja superior a um determinado limite, acionando o *Buzzer*. A Figura 34 apresenta um fragmento do *Thing Description* fornecido pelo dispositivo deste estudo de caso, sendo composto de uma descrição, o tipo de dados esperado como retorno, o protocolo utilizado e o local onde o usuário pode subscrever ao evento.

```

"overheating" : {
  "description" : "System detects critical temperature (overheating)",
  "data" : {
    "type": "application/json"
  },
  "forms": [
    {
      "href": "mqtt://mqtt.eclipse.org/Alarme",
      "contentType": "application/json",
      "op": "subscribeevent"
    }
  ]
}

```

Figura 34 - Fragmento do WoT Thing Description - Event.

O diagrama de sequência da Figura 35 descreve, em mais detalhes, a implementação do conceito *evento*, consistindo em um processo cíclico de monitoramento de temperatura em que, periodicamente, o dispositivo ESP8266 requisita a leitura do sensor de temperatura, presente nos dispositivos DHT11 e DHT22. A regra de negócio implementada neste evento consiste em verificar se a temperatura obtida excede o valor limite definido. Caso verdadeiro, ocorre o acionamento do *passive buzzer* (dispositivo emissor de ondas mecânicas audíveis) com a finalidade de alertar a ocorrência do evento e uma notificação é enviada ao tópico especificado usando o protocolo MQTT. Após as leituras atingirem valores abaixo ou igual ao limite, o dispositivo desativará o *buzzer* e publicará uma mensagem indicando a mudança de estado. Caso o usuário tenha interesse em receber notificações deste evento, o mesmo deve realizar uma subscrição no tópico do *broker* descrito no *Thing Description*. Neste evento, o dispositivo somente envia notificações caso haja mudança de estado na situação do alarme.

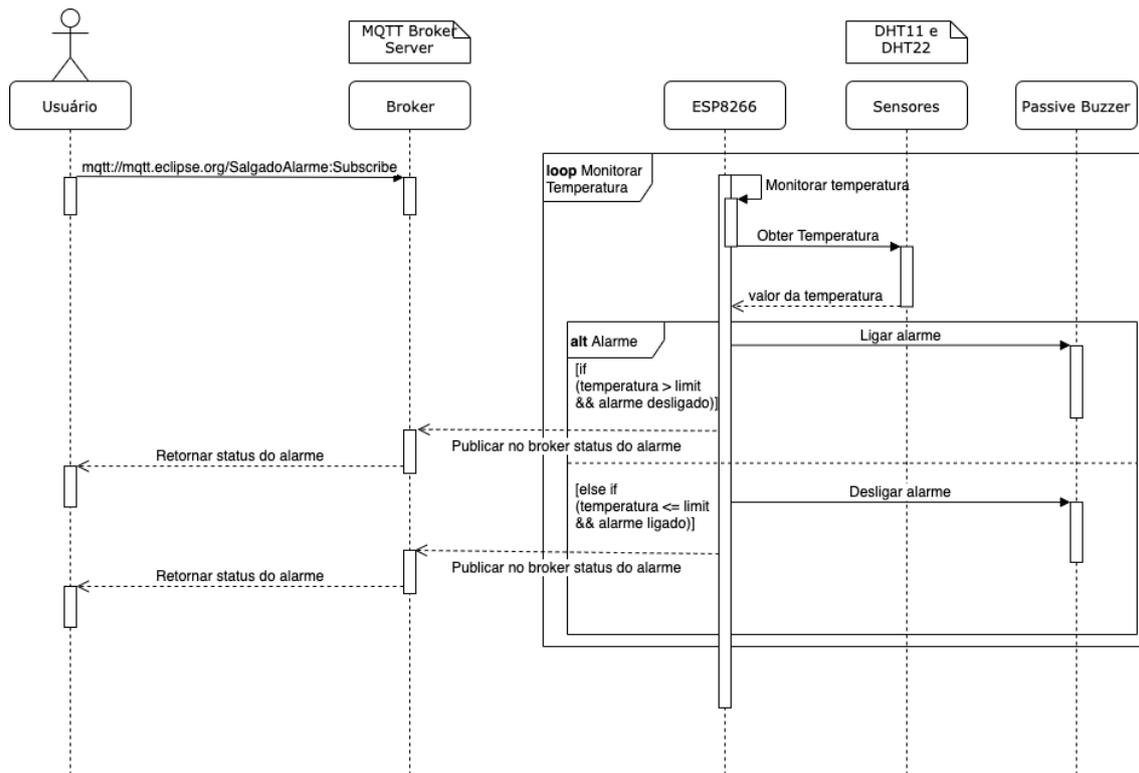


Figura 35 - Diagrama de seqüência para descrição de um evento.

Este caso exemplifica um aspecto da relevância de uma linguagem de descrição de dispositivos, pois a explícita descrição dos elementos que compõe um dispositivo, propicia ao usuário o poder de seleção de acordo com os requisitos do domínio da aplicação. O código fonte da implementação está disponível no repositório do projeto⁸.

5.2.1. Experimentos de Avaliação

Esta seção apresenta uma avaliação do dispositivo IoT projetado com suporte a recomendação W3C *Web of Things*. A metodologia de avaliação empregada foi baseada em um conjunto de experimentos práticos, descritos adiante. O objetivo central desta avaliação é expor este dispositivo a um ambiente que representa a demanda típica de um sistema de informação que adota dispositivos IoT como parte de sua solução, medindo indicadores que explicitem o comportamento do dispositivo em relação a demanda requerida. Para isto, foi construída uma aplicação cujo objetivo é requisitar dados de temperatura e umidade oriundos do sensor DHT11, conforme ilustrado pela Figura 36. Para isto, a aplicação construída solicita o *Thing Description* do dispositivo, ao receber

⁸ <https://github.com/si4-iot/si4-iot>

tais informações, instancia um objeto cujos métodos são as interações disponíveis do dispositivo. A partir disto, uma iteração é iniciada a fim de requisitar informações de temperatura e umidade do sensor DHT11, considerando o intervalo de tempo entre as requisições.

Foram planejados dois experimentos em que: (i) representa um contexto de aplicação que requer um acesso síncrono ao dispositivo, ou seja, requer uma interação periódica às funcionalidades presentes no mesmo e; (ii) ilustra um cenário de aplicação com acesso assíncrono, onde o intervalo entre requisições é aleatório.

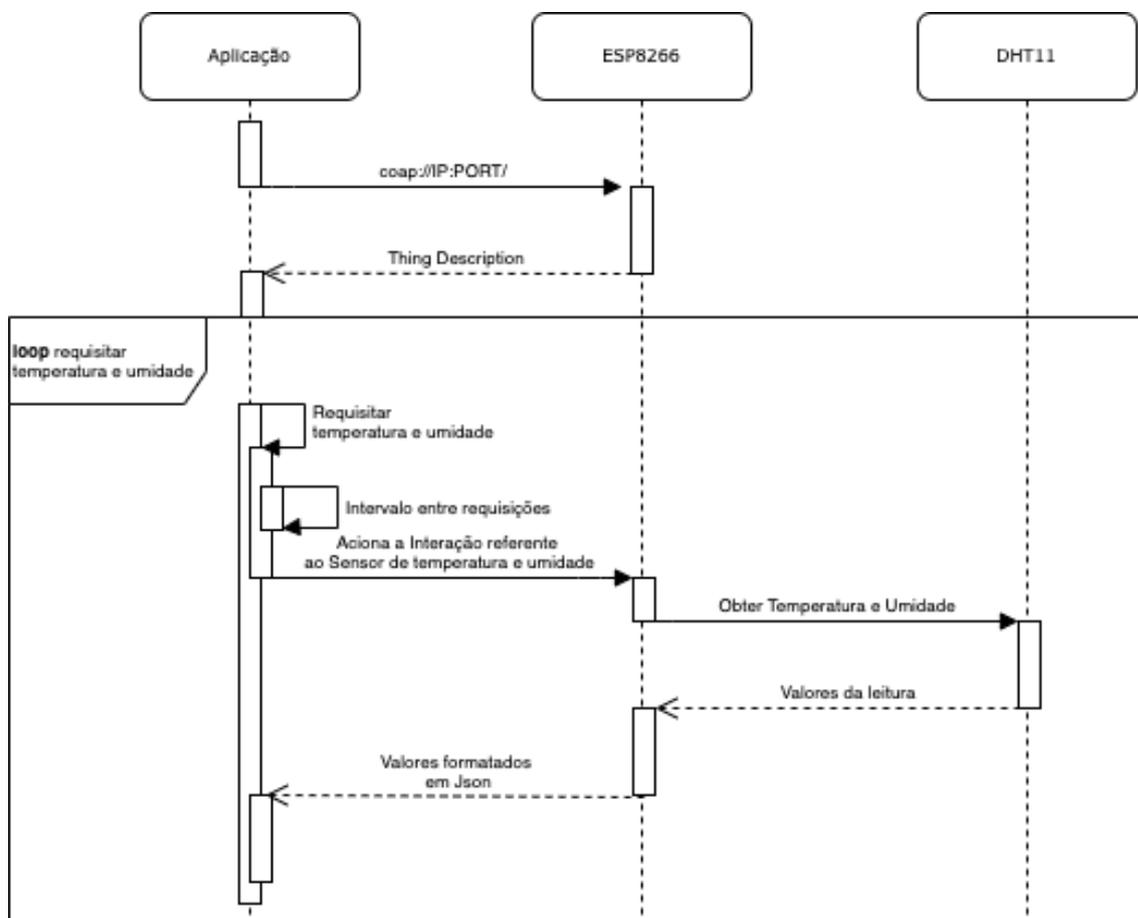


Figura 36 - Avaliação do protótipo.

As seções 5.2.1.1 e 5.2.1.2 apresentam, em mais detalhes, os experimentos (i) e (ii), e discute os resultados obtidos, A seção 5.2.1.3 apresenta as considerações finais.

5.2.1.1. Experimento 1: Aplicações síncronas

Conforme introduzido na seção anterior, este experimento foi realizado com intuito de expor o protótipo de hardware e a implementação da recomendação *W3C Web of Things* a um ambiente cujo requisito da aplicação construída requer interações com um dispositivo IoT em intervalos de tempo pré-determinados. O objetivo é compreender o comportamento da implementação realizada, expondo diversas variáveis, tais como tempo médio de resposta, tempo médio de leitura do sensor, o percentual de falhas e sua origem. Para isto, foi construída uma aplicação capaz de requisitar, por meio de um intervalo de tempo pré-estabelecido, informações de temperatura e umidade do sensor DHT11.

O experimento foi planejado para ser executado em 5 etapas, ilustrando diferentes intensidades de uso do dispositivo, para cada intervalo de tempo entre as requisições, conforme a Tabela 4.

Tabela 4 - Etapas do experimento síncrono.

Etapa	Intervalo de tempo (ms)	Quantidade de requisições
1	10	100
		500
		1000
		5000
		10000
2	50	100
		500
		1000
		5000
		10000
3	100	100
		500
		1000
		5000
		10000
4	300	100
		500
		1000
		5000

		10000
5	500	100
		500
		1000
		5000
		10000

A realização das etapas supracitadas possibilitou avaliar o comportamento do dispositivo, permitindo mensurar o tempo médio de resposta e o tempo médio para leitura dos dados pelo sensor. A Tabela 5 apresenta os resultados obtidos no experimento.

Tabela 5 - Tempo médio de resposta.

Etapa	Intervalo(ms)	Qtd. Req.	Tempo Médio de Resposta (ms)	Tempo Médio de leitura (ms)
1	10	100	162,48	1,83
	10	500	212,39	2,03
	10	1000	236,94	1,95
	10	5000	166,54	2,12
	10	10000	125,52	2,32
2	50	100	78,07	1,59
	50	500	167,31	1,89
	50	1000	222,59	2,00
	50	5000	200,72	2,22
	50	10000	107,48	2,36
3	100	100	78,82	1,93
	100	500	199,83	2,08
	100	1000	244,04	2,22
	100	5000	166,54	2,12
	100	10000	93,82	2,37
4	300	100	68,34	2,08
	300	500	164,23	2,22
	300	1000	167,87	2,39
	300	5000	90,81	2,31
	300	10000	55,84	2,69
5	500	100	42,67	2,54
	500	500	59,79	2,52
	500	1000	57,92	2,58
	500	5000	63,10	2,73
	500	10000	51,57	3,17

É possível perceber que o tempo médio de resposta tende a cair consideravelmente com o aumento do intervalo entre cada requisição. Possivelmente, trata-se de um indicativo de limitações do hardware utilizado no dispositivo; entretanto, o tempo médio para leitura do sensor (DHT11) demonstra ser estável independente do volume de requisições e do intervalo entre as mesmas.

Neste experimento, também foi possível observar um percentual relativamente pequeno de falhas. Nesse sentido, foi realizada uma investigação das falhas apresentadas em cada etapa e sua origem. Neste trabalho, a falha é entendida como uma requisição, realizada pela aplicação, que não obteve resposta do dispositivo. Foram identificados três pontos de falha, a saber: Em (i) é caracterizado pelas requisições que chegam ao dispositivo, porém, houve uma falha no processo da leitura do sensor (DHT11); Em (ii) se refere as interrupções súbitas no funcionamento do dispositivo (ESP8266); e por fim, (iii) indicam o não estabelecimento da conexão entre a aplicação e o dispositivo, isso pode ocorrer em decorrência de um problema de comunicação de rede ou mesmo de uma falha do dispositivo (ii), dependendo do intervalo entre as requisições. A Tabela 6 apresenta os resultados referente as falhas ocorridas em cada etapa do experimento.

Tabela 6 - Falhas de requisições ao dispositivo.

Etapa	Intervalo(ms)	Qtd. Req.	Falhas			Total de Falhas	% de Falhas
			(i)	(ii)	(iii)		
1	10	100	0	0	0	0	0,0%
	10	500	37	18	30	85	17,0%
	10	1000	4	9	13	26	2,6%
	10	5000	7	8	9	24	0,5%
	10	10000	14	47	57	118	1,2%
2	50	100	0	0	0	0	0,0%
	50	500	6	9	12	27	5,4%
	50	1000	0	1	1	2	0,2%
	50	5000	15	47	59	121	2,4%
	50	10000	9	88	105	202	2,0%
3	100	100	2	1	2	5	5,0%
	100	500	4	4	7	15	3,0%
	100	1000	3	8	11	22	2,2%
	100	5000	7	8	9	24	0,5%
	100	10000	5	112	122	239	2,4%
4	300	100	0	1	1	2	2,0%
	300	500	0	0	0	0	0,0%

	300	1000	1	10	12	23	2,3%
	300	5000	1	25	27	53	1,1%
	300	10000	9	69	72	150	1,5%
5	500	100	0	0	0	0	0,0%
	500	500	0	3	3	6	1,2%
	500	1000	0	8	8	16	1,6%
	500	5000	0	34	44	78	1,6%
	500	10000	0	66	68	134	1,3%

O dispositivo utilizado no experimento foi implementado utilizando hardware de baixo custo e, embora apresente uma baixa taxa de erros, se faz necessária uma análise da origem das falhas.

Como pode ser notado, analisando isoladamente cada etapa, em termos percentuais, a taxa de falhas não está relacionada ao volume de requisições, no entanto, observa-se que o maior intervalo entre as requisições (etapa 5) tende a estabilizar a taxa de falhas. Por fim, percebe-se que a falha do dispositivo tende a gerar falha de conexão, visto que as requisições realizadas no intervalo de tempo da indisponibilidade momentânea do dispositivo são medidas como falha do tipo conexão.

5.2.1.2. Experimento 2: Aplicações assíncronas

Esta seção tem como objetivo compreender o comportamento do dispositivo implementado em um ambiente onde uma aplicação pode requerer dados dos sensores de forma não periódica, analisando diversas variáveis, tais como tempo médio de resposta, tempo médio de leitura do sensor, o percentual de falhas e sua origem. Para isto, foi construída uma aplicação capaz de requisitar, por meio de um intervalo de tempo aleatório, informações de temperatura e umidade do sensor DHT11.

Este experimento foi realizado em 5 etapas, em que a aplicação realizou 100, 500, 1000, 5000 e 10000 requisições de dados de temperatura e umidade, em que o intervalo de tempo entre requisições foi um valor aleatório entre 10ms e 500ms. A Tabela 7 apresenta os resultados referentes aos tempos médios de resposta.

Tabela 7 - Tempo médio de resposta.

Etapa	Qtd. Req.	Tempo Médio de Resposta (ms)	Tempo Médio de Leitura (ms)
1	100	70,29	2,13
2	500	95,73	2,06
3	1000	103,16	2,11
4	5000	93,29	2,23
5	10000	62,94	2,56

Observa-se que o tempo médio de resposta, embora tenha uma certa variação, se mostrou relativamente estável se comparado com o experimento anterior. Percebe-se que o tempo médio para leitura do sensor permaneceu estável independente do volume de requisições da aplicação. A Tabela 8 apresenta uma análise das falhas ocorridas durante este experimento. De maneira análoga ao experimento anterior, as falhas foram categorizadas em três tipos: Em (i) é caracterizado pelas requisições que chegam ao dispositivo, porém, houve uma falha no processo da leitura do sensor (DHT11); Em (ii) se refere as interrupções súbitas no funcionamento do dispositivo (ESP8266); e por fim, (iii) indicam o não estabelecimento da conexão entre a aplicação e o dispositivo, isso pode ocorrer em decorrência de um problema de comunicação de rede ou mesmo de uma falha do dispositivo (ii), dependendo do intervalo entre as requisições.

Tabela 8 - Falhas de requisições ao dispositivo.

Etapa	Qtd. Req.	Falhas			Total de falhas	% de Falhas
		(i)	(ii)	(iii)		
1	100	1	1	2	4	4,00%
2	500	0	10	12	22	4,40%
3	1000	0	10	13	23	2,30%
4	5000	4	39	46	89	1,78%
5	10000	1	117	120	238	2,38%

Observou-se que a taxa de falhas nas etapas 1 e 2 tiveram um aumento relevante, em termos percentuais, se comparado com o experimento anterior, já as etapas 3, 4 e 5 manteve taxas de falhas equivalentes ao visto no experimento simulando aplicações de requisições periódicas.

5.2.1.3. Considerações sobre os Experimentos 1 e 2

A implementação da especificação *W3C Web of Things*, em um dispositivo real, ainda pouco explorada na literatura, possibilitou demonstrar suas principais características e a avaliação realizada – simulando aplicações com requisitos distintos em termos de periodicidade de acesso às informações cedidas pelo dispositivo IoT, demonstrou o comportamento do mesmo em situações distintas de volume de requisições e intervalo entre as mesmas. Foi possível observar que o intervalo superior a 500ms entre requisições demonstrou apresentar estabilidade em termos de tempo médio de resposta. Uma descoberta deste experimento foram as falhas ocorridas durante as requisições, embora baixas em termos percentuais, são presentes. A origem destas falhas foi identificada e categorizada, conforme discutido nos experimentos. Por fim, as informações obtidas nos experimentos explicitam alguns importantes fatores que devem ser levados em consideração em um projeto de uma aplicação que adota dados oriundos de um dispositivo IoT como parte de sua solução.

O padrão W3C Web of Things se mostrou adequado no seu propósito central – atuar no problema de interoperabilidade existente no ambiente de Internet das Coisas, promovendo uma padronização em termos de interação com os componentes do dispositivo IoT, da maneira de descrevê-lo, incluindo uma preocupação quanto à descrição semântica. Dispositivos com severas restrições de hardware conseguem estar em conformidade com esta recomendação, por meio de *digital twins*, ou seja, um elemento de software externo ao dispositivo, que é capaz de interagir com o dispositivo físico e fornecer uma camada de abstração em conformidade com o W3C WoT.

5.3. Realização da Camada de Geração Massiva de Dispositivos IoT Virtuais

Esta seção trata da implementação da infraestrutura de geração massiva de dispositivos IoT virtuais. De modo análogo ao procedimento metodológico para a realização e experimentação da Camada *Thing*, foi desenvolvido um experimento de implantação do ambiente proposto em uma infraestrutura de nuvem computacional, bem

como realizada uma avaliação da arquitetura em termos de capacidade de geração de dispositivos IoT virtuais e de desempenho de um dispositivo gerado.

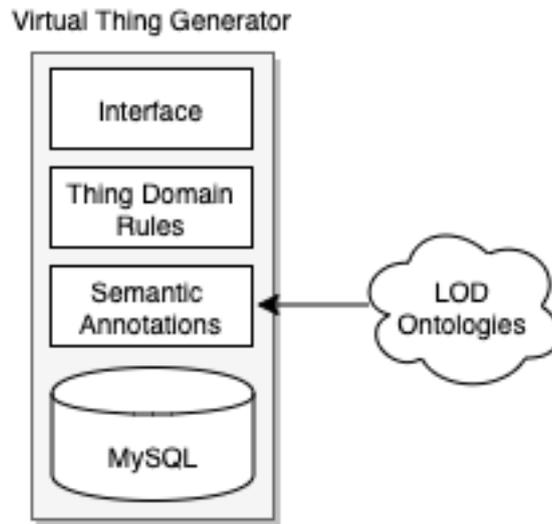


Figura 37 – Implementação do Virtual Thing Generator.

Como visto na arquitetura proposta, o *Virtual Thing Generator* é o componente responsável pelo processo de geração de dispositivos virtuais. É através do componente *Interface* que o usuário informa os parâmetros requeridos pelo seu algoritmo de geração, tais como o local de armazenamento dos arquivos gerados, a quantidade desejada de dispositivos e o número máxima de interações (propriedades, ações e eventos) que um dispositivo virtual deve possuir. Esta interface, bastante simples e intuitiva, é ilustrada na Figura 38.

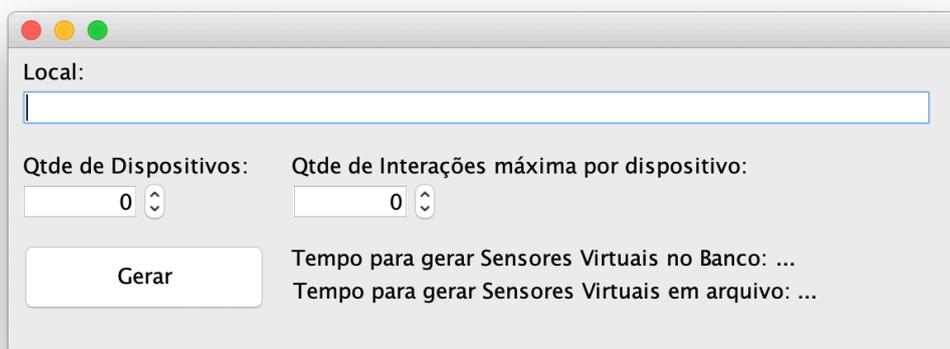


Figura 38 - Gerador de dispositivos de Internet das Coisas

Inicialmente, a quantidade de interações de um dispositivo é especificada, sendo esta um valor inteiro, aleatório, entre 1 e a quantidade máxima de interações definida pelo

usuário. Em seguida, cada interação é selecionada por meio de um sorteio entre as interações previamente registradas e, então, é associada ao novo dispositivo. Cada dispositivo consiste em um arquivo *javascript* gerado pela aplicação, baseado na implementação do projeto *Eclipse Thingweb*⁹ Cada dispositivo virtual gerado possui um comportamento equivalente a um dispositivo real no que se refere à composição e exposição do *Thing Description* e a disponibilização das interações por meio dos protocolos previstos na especificação W3C *Web of Things*. Interações são previamente registradas no banco de dados *MySQL*, em que é definido seu nome, valor mínimo e valor máximo retornado e a associação a um conceito de uma ontologia existente. Por exemplo, a interação denominada *temperature*, retorna valores reais entre 0 e 100, está associado ao conceito *om:Celsius_temperature_scale*. Este ambiente possibilita a inserção de novas interações conforme a necessidade do domínio.

A implementação do *Virtual Thing Generator* foi realizada utilizando a linguagem Java, utilizando o banco de dados relacional *MySQL* para persistir as interações e os dispositivos gerados.

5.3.1. Experimentos de Avaliação

Esta seção apresenta uma avaliação do componente da arquitetura denominado *Virtual Thing Generator*. A metodologia de avaliação empregada foi baseada em um conjunto de experimentos práticos, descritos adiante. O objetivo central desta avaliação é demonstrar a capacidade deste componente em gerar dispositivos IoT virtuais com diversidade de interações. Foram planejados três experimentos em que: (i) relata o processo de implantação massiva de dispositivos IoT virtuais em uma plataforma de nuvem computacional; (ii) realiza a medição de desempenho referente ao processo de geração de dispositivos IoT virtuais massivos e (iii) Medição de desempenho de um Dispositivo IoT Virtual particular gerado pelo *Virtual Thing Generator*.

As seções 5.3.1.1, 5.3.1.2 e 5.3.1.3 apresentam, em mais detalhes, os experimentos (i), (ii) e (iii), e discute os resultados obtidos. A seção 5.3.1.4 apresenta as considerações finais.

⁹ <https://projects.eclipse.org/proposals/eclipse-thingweb>

5.3.1.1. Experimento 1: Implantação Massiva de Dispositivos IoT em uma Infraestrutura de Nuvem Computacional

Como forma de validar o ambiente proposto, ou seja, demonstrar a capacidade efetiva de geração massiva de dispositivos IoT, heterogêneos, interoperáveis e funcionais – em conformidade com a recomendação *W3C Web of Things*, foi realizada uma geração de dispositivos IoT virtuais, que foram hospedados em uma infraestrutura de nuvem computacional. A AWS foi adotada neste experimento, onde os dispositivos foram executados em uma instância EC2 da AWS (t2.micro).

Para isto, o *Virtual Thing Generator* foi configurado para gerar 1 mil dispositivos IoT virtuais, com até 5 interações, sorteadas dentre 30 interações previamente registradas. Como resultado desta operação, foram gerados 1 mil arquivos *Javascript* - baseado na implementação do projeto *Eclipse Thingweb*, que corretamente hospedados, possuem um comportamento em termos de capacidade de responder a requisições equivalente a um dispositivo físico real.

Este experimento foi definido em três fases: (i) a geração de dispositivos IoT virtuais e hospedagem dos arquivos gerados na plataforma de nuvem computacional; (ii) explorar o *Thing Description* de um dispositivo particular e por fim, (iii) apresentar o funcionamento de duas interações: sensor e um atuador. Essas três fases ilustram a capacidade do *Virtual Thing Generator* de gerar dispositivos IoT funcionais e em conformidade com a recomendação *W3C WoT*.

O resultado da fase (i) do experimento pode ser visto na Figura 39. Os parâmetros foram definidos na interface do *Virtual Thing Generator*, conforme discutido anteriormente e os 1 mil arquivos gerados foram copiados para uma instância EC2 da AWS (t2.micro) executando o sistema operacional *Ubuntu Server 18.04*, com os requisitos necessários¹⁰ para o devido funcionamento dos dispositivos IoT Virtuais. Na Figura 39, cada linha representa um dispositivo IoT virtual gerado e funcional.

¹⁰ <https://github.com/si4-iot/si4-iot/tree/VirtualThingGenerator>

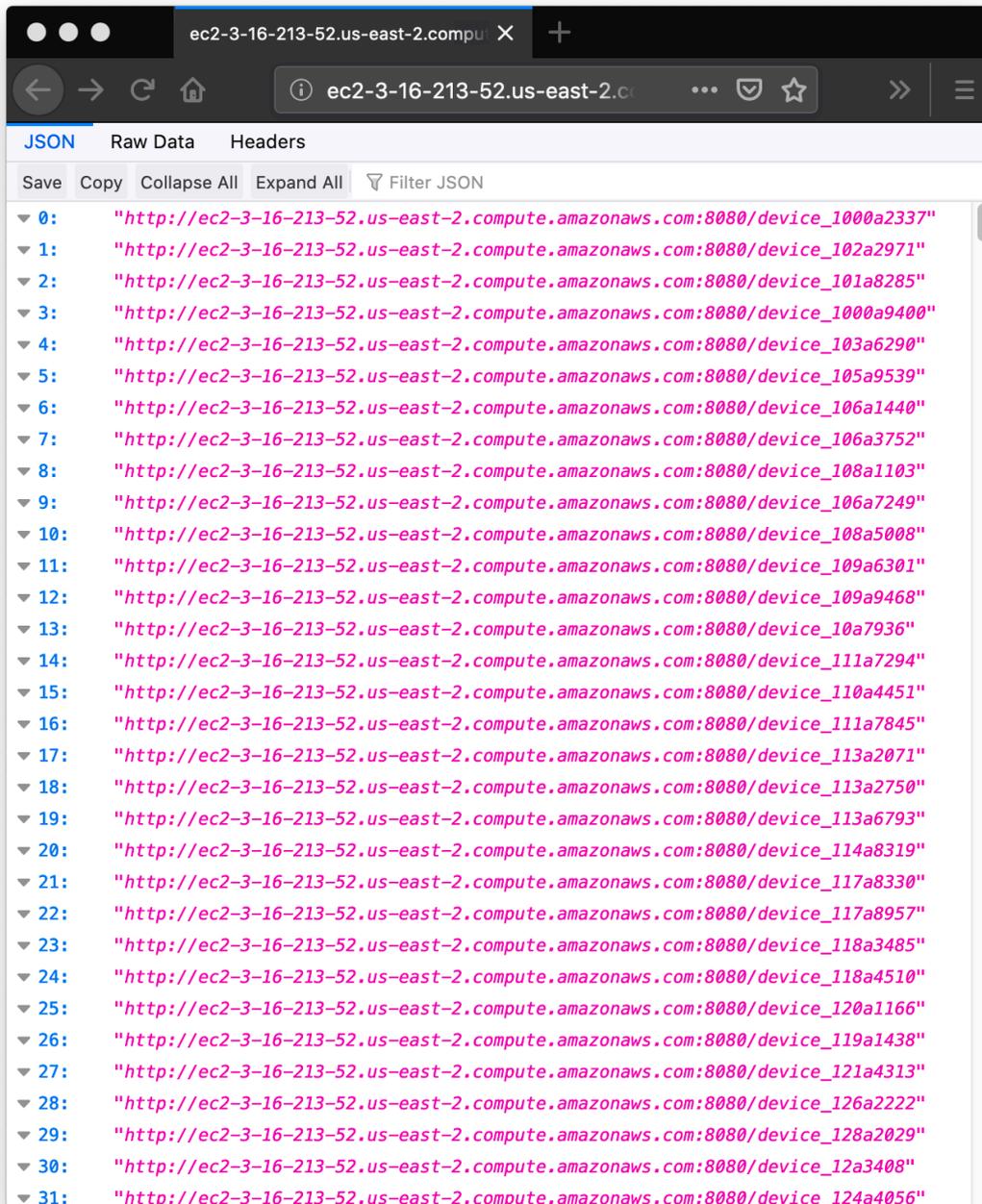


Figura 39 - Lista de dispositivos gerados hospedados na Amazon AWS.

Além da implementação da lógica necessária para prover o funcionamento de seus componentes (ações, propriedades e eventos), tornando-os acessíveis por meio de um particular protocolo, o algoritmo presente em cada dispositivo virtual gerado também possui a capacidade de compor e expor o TD, como preconizado pela arquitetura conceitual. A exposição do *Thing Description* ocorre por meio do protocolo HTTP; assim, quando o usuário quiser conhecer todas as características do dispositivo, basta

realizar uma requisição utilizando o método GET do protocolo HTTP no endereço exposto do dispositivo.

A Figura 40 apresenta o resultado da fase (ii) do experimento. O trecho inicial do *Thing Description* de um dispositivo gerado pela aplicação, onde é possível observar informações como título, descrição, contexto e tipo. O contexto é parte fundamental do processo de descrição semântica do dispositivo e seus componentes, sendo o local onde são referenciadas as ontologias que são utilizadas para distinguir conceitualmente as características e capacidades de um dispositivo IoT Virtual.

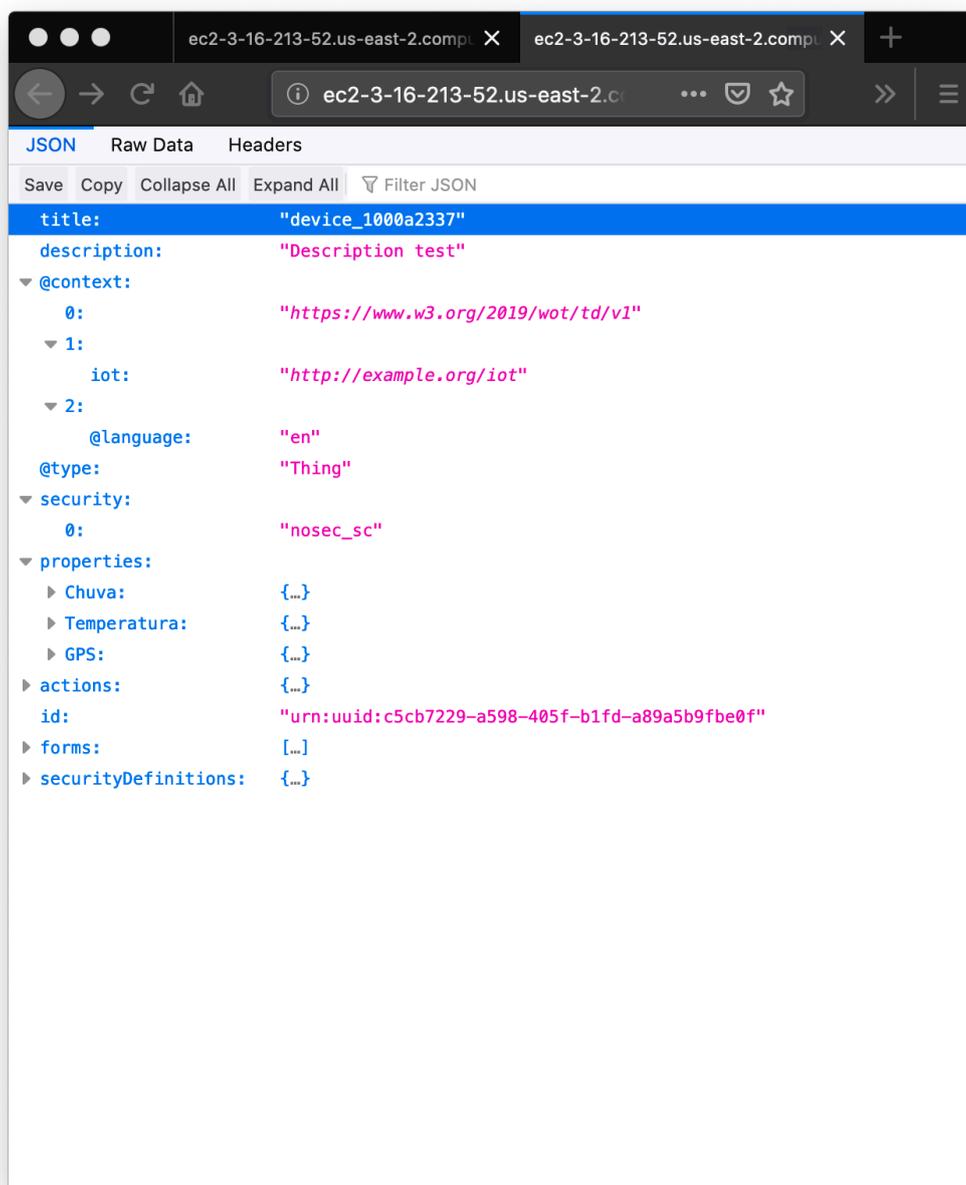


Figura 40 - Descrição de um dispositivo e suas interações.

Como mencionado anteriormente, as interações são previamente registradas no *Database* presente no *Virtual Thing Generator*. Para cada dispositivo a ser gerado, o simulador realiza uma escolha aleatória das interações disponíveis no banco de dados, considerando a quantidade máxima de interações definidas pelo utilizador do sistema. Como resultado, são gerados dispositivos IoT virtuais com extensa composição de interações, fornecendo um ambiente com ampla diversidade de dispositivos. A fase (iii) do experimento se inicia com a análise das interações de um dispositivo particular oriundo do processo de geração do ambiente proposto, conforme apresentado adiante.

A Figura 41 apresenta as propriedades disponíveis em um determinado dispositivo gerado. Neste exemplo, há quatro sensores disponíveis - oxigênio, dióxido de carbono, som e luminosidade. O algoritmo presente no dispositivo gera, para cada propriedade, um valor aleatório entre um valor mínimo e máximo que é definido no momento do registro da mesma, sendo os valores gerados acessíveis por meio dos protocolos HTTP e CoAP.

```
▼ properties:
  ▶ oxygen:           {...}
  ▶ carbon_dioxide:  {...}
  ▶ sound:           {...}
  ▶ light:           {...}
```

Figura 41 - Propriedades do dispositivo IoT virtual gerado.

A Figura 42 apresenta a propriedade oxigênio em mais detalhes, onde é possível observar a associação com conceitos de ontologias existentes com objetivo de descrever semanticamente a propriedade. Neste exemplo, o tipo é associado com o conceito *Sensor* da ontologia SAREF (DANIELE; DEN HARTOG; ROES, 2015) e a unidade de medida retornada pela propriedade é concentração em partes por milhão, definida pelo conceito PPM da ontologia m3-lite (AGARWAL et al., 2016). O valor gerado pela propriedade pode ser obtido por meio dos protocolos HTTP e CoAP e o conteúdo retornado é em formato JSON. Essa abordagem propicia a compreensão precisa e a maneira pelo qual será possível interagir com as funcionalidades presentes no dispositivo.

```

▼ properties:
  ▼ oxygen:
    type: "saref:Sensor"
    description: "null"
    unit of measurement: "m3-lite:PPM"
    observable: false
    readOnly: true
    writeOnly: false
  ▼ forms:
    ▼ 0:
      ▼ href: "http://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:8080/device_114a2276/properties/oxygen"
      contentType: "application/json"
      ▼ op:
        0: "readproperty"
        htv:methodName: "GET"
    ▼ 1:
      ▼ href: "coap://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:5683/device_114a2276/properties/oxygen"
      contentType: "application/json"
      ▼ op:
        0: "readproperty"

```

Figura 42 - Detalhes da propriedade oxigênio do dispositivo IoT virtual gerado.

O diagrama de sequência que demonstra este cenário está representado na Figura 43. Uma requisição de valores do sensor de oxigênio é usada para demonstrar o acesso às propriedades de um dispositivo IoT virtual no estudo de caso. O usuário inicia o processo, utilizando método GET do protocolo CoAP para fazer a requisição ao dispositivo conectado (1); o dispositivo conectado à rede recebe a solicitação e executa a lógica necessária para a obtenção dos valores de oxigênio fornecidos pelo algoritmo (2), e retorna um valor aleatório entre o mínimo e máximo previamente estabelecido no registro desta interação (3). Ao receber as leituras, o dispositivo processa os valores recebidos em formato JSON e os envia ao usuário que requisitou usando o protocolo CoAP (4).

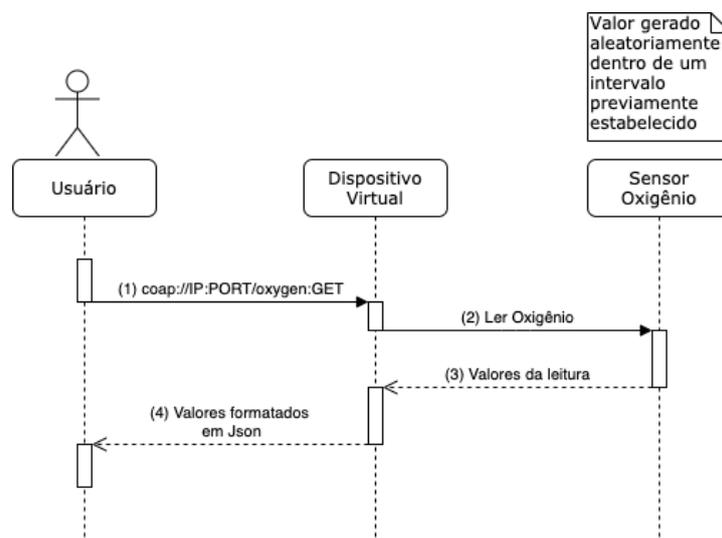


Figura 43 - Diagrama de sequência da interação com a propriedade oxigênio do dispositivo IoT virtual gerado.

Como visto, atuadores são elementos importantes no ambiente de IoT. A Figura 44 apresenta um exemplo de representação das propriedades de um *buzzer*. A ontologia SAREF (DANIELE; DEN HARTOG; ROES, 2015) foi utilizada para caracterizar semanticamente o tipo (*Actuator*) e a unidade de medida retornada (*On/Off*). O valor do estado atual do *buzzer* pode ser obtido por meio dos protocolos HTTP e CoAP, acessado via URL. Os valores devem estar em formato JSON.

```

▼ properties:
  ▼ buzzer:
    type: "saref:Actuator"
    description: "null"
    unit of measurement: "saref:OnOffFunction"
    observable: false
    readOnly: true
    writeOnly: false
  ▼ forms:
    ▼ 0:
      ▼ href: "http://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:8080/device_102a1759/properties/buzzer"
      contentType: "application/json"
      ▼ op:
        0: "readproperty"
        htv:methodName: "GET"
    ▼ 1:
      ▼ href: "coap://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:5683/device_102a1759/properties/buzzer"
      contentType: "application/json"
      ▼ op:
        0: "readproperty"

```

Figura 44 - Detalhes do atuador *buzzer* do dispositivo IoT virtual gerado.

Como apresentado, a forma como o usuário é capaz de invocar uma mudança de estado de um atuador em um dispositivo IoT é representada como uma interação do tipo ação. Isto pode ser observado na Figura 45. As informações contidas na figura revelam o suporte aos protocolos HTTP e CoAP, o método e a URL a ser utilizada para acionar a mudança de estado do *buzzer*. O algoritmo presente no dispositivo virtual gerado possui a lógica necessária para registrar o estado e as eventuais mudanças requeridas.

```

▼ actions:
  ▼ change_buzzer:
    ▼ forms:
      ▼ 0:
        ▼ href:           "http://ec2-3-18-220-42.us-
                           east-2.compute.amazonaws.com:8080
                           /device_102a1759/actions/change_buzzer"
        contentType:     "application/json"
        ▼ op:
          0:              "invokeaction"
          htv:methodName: "POST"
        ▼ 1:
          ▼ href:         "coap://ec2-3-18-220-42.us-
                           east-2.compute.amazonaws.com:5683
                           /device_102a1759/actions/change_buzzer"
          contentType:   "application/json"
          op:             "invokeaction"
      idempotent:        false
      safe:              false

```

Figura 45 - Interação do tipo ação - buzzer.

5.3.1.2. Experimento 2: Medição de Desempenho da Geração de Dispositivos IoT Virtuais Massivos

Com intuito de avaliar o desempenho na geração de dispositivos virtuais, foi realizado um experimento em que foi medido do tempo do processo de geração de tais dispositivos (T_{total}), considerando a quantidade de dispositivos, o tempo para persistência em banco de dados (T_{bd}) e o tempo para gerar os arquivos (T_{arq}) referentes aos dispositivos virtuais. O resultado desta avaliação pode ser visto na Tabela 9.

$$T_{total} = T_{bd} + T_{arq}$$

Tabela 9 - Tempo para geração de dispositivos virtuais.

Quantidade de dispositivos	Tempo DB	Tempo Arquivo
10	00:00:01.456	00:00:00.071
50	00:00:04.087	00:00:00.238
100	00:00:06.811	00:00:00.574
500	00:00:29.419	00:00:02.285
1000	00:00:56.512	00:00:04.588
5000	00:05:02.169	00:00:33.955

Neste experimento, havia uma lista de 35 interações previamente registradas no banco de dados do *Virtual Thing Generator*. Sendo que para cada dispositivo, foi sorteado um número entre 1 e 5 interações a compor o dispositivo IoT virtual gerado.

O objetivo deste experimento foi quantificar o tempo de geração de dispositivos. Observa-se que o tempo total tende a um comportamento linear, demonstrando viabilidade da geração massiva de dispositivos IoT.

5.3.1.3. Experimento 3: Medição de desempenho de um Dispositivo IoT Virtual

Outro experimento realizado foi a implementação de uma aplicação que faz uso de um *Virtual Thing* gerado pela arquitetura proposta. O *Virtual Thing* foi hospedado em uma plataforma de nuvem computacional e o experimento consistiu na requisição de informações de uma determinada propriedade (no caso, sensor de oxigênio), conforme representado na Figura 46.

Esta interação do tipo propriedade foi previamente registrada no banco de dados para fornecer um intervalo de valores entre zero e 200 PPM (Partículas por Milhão) m³-lite:PPM. Foram executadas cinco etapas, nas quais a aplicação realizou 100, 500, 1000, 5000 e 10000 requisições com intervalo de 10ms entre cada requisição, demonstrando um exemplo uso intensivo deste dispositivo IoT virtual. Todas as requisições estão representadas na Figura 46. Esta figura apresenta um gráfico de dispersão que ilustra todas as requisições realizadas, indicando o tempo de resposta e o valor de sensoriamento obtido.

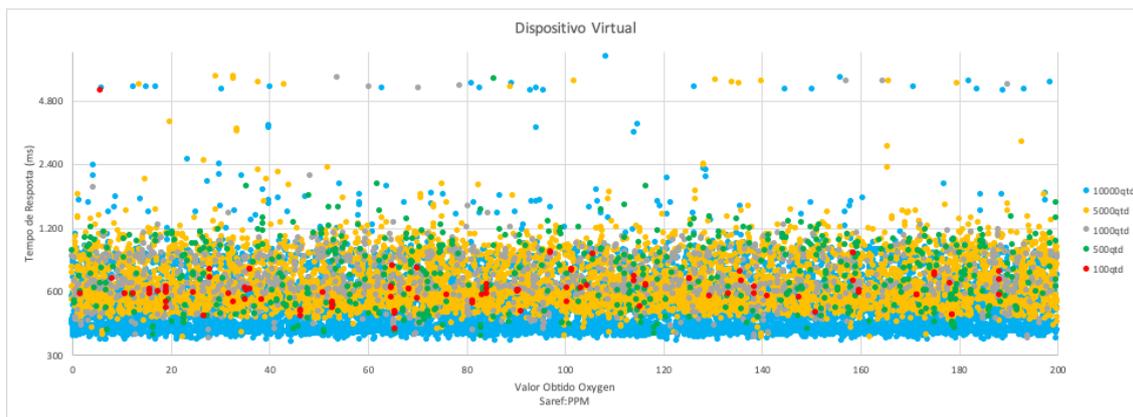


Figura 46 - Teste de requisições da propriedade oxigênio de um dispositivo virtual.

Como pode ser observado, o dispositivo IoT gerado pela arquitetura proposta demonstrou ser capaz de suportar um uso intensivo por aplicações, com a maior parte tempo de resposta entre 300 ms e 1200 ms, tendo um conjunto minoritário de requisições com tempo de resposta superior a 4800ms.

5.3.2. Considerações dos Experimentos 1, 2 e 3

A implementação de um ambiente capaz de gerar um enorme quantitativo de dispositivos IoT virtuais, com extensa diversidade de sensores e atuadores, possibilitou evidenciar as dificuldades de lidar em um contexto de disponibilidade massiva de dispositivos IoT. A característica principal deste ambiente é a adoção da recomendação W3C *Web of Things* como estratégia para uniformizar a maneira de expor as funcionalidades de um dispositivo e a forma de interagir com as funcionalidades disponíveis no mesmo. O objetivo é fornecer uma infraestrutura simples, porém útil, dotada de funcionalidades importantes para apoio às pesquisas que exploram cenários com um conjunto extenso de dispositivos IoT, com diversidade de interações (propriedades, ações e eventos).

Os experimentos realizados evidenciaram o potencial do ambiente na provisão massiva de dispositivos virtuais, e permitiram comprovar a sua flexibilidade na definição dos elementos que compõem uma Coisa (*Thing*) e a eficiência na maneira de expor as

características de dispositivos IoT. Desta forma, foi possível explorar funcionalidades de dispositivos em um contexto próximo de um cenário real.

5.4. Realização da Camada Catálogo

Como visto, a arquitetura dispõe de um ambiente que permite o registro e o compartilhamento de *Things*, e possibilita processar operações de pesquisa de *WoT Thing Descriptions*: o Catálogo. Tal componente é fundamental dentro de uma infraestrutura IoT massiva, pois auxilia no processo de seleção de dispositivos que atendem aos requisitos funcionais de interesse das aplicações.

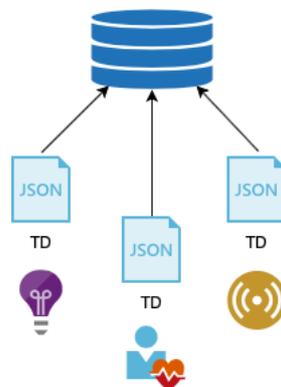


Figura 47 - Catálogo armazenando WoT Thing Description.

Foi realizada uma implementação de um catálogo que possui como elementos: *Thing Descriptions Manager*, *Search Things Service* e *Repository Thing Descriptions*.

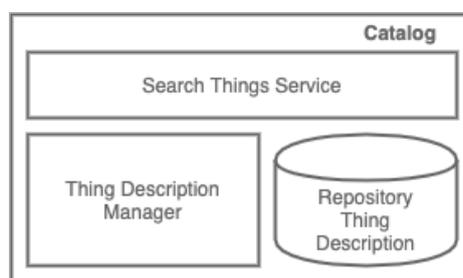


Figura 48 - Catálogo.

O *Thing Description Repository*, que armazena informações dos dispositivos de IoT, considerando sua extensa variedade, foi implementado adotando como solução de armazenamento o banco de dados NoSQL MongoDB. Para a realização do *Thing*

Description Manager, responsável pelo o registro das descrições fornecidas pelos dispositivos, foi implementado um serviço, utilizando a linguagem de programação NodeJS, que recebe a URL do dispositivo como parâmetro e realiza a requisição do *Thing Description* do dispositivo informado e armazena tais informações no *Thing Description Repository*.

Da mesma forma, para a realização do *Search Thing Service*, componente que fornece um meio de buscar dispositivos existentes no *Thing Description Repository*, usando uma linguagem que possibilita utilizar dos conceitos de ontologias associados aos elementos que compõe uma Coisa, foi implementado um outro serviço. Este serviço, que utiliza a linguagem NodeJS, recebe como parâmetro uma *string* de busca, em conformidade com o banco de dados MongoDB, realiza a busca informada e retorna uma lista de URLs referentes aos dispositivos selecionados.

A infraestrutura de serviços implementada referente ao Catálogo foi hospedada na plataforma em nuvem AWS da Amazon, utilizando uma instância EC2 Micro – Ubuntu 18.04 LTS Server.

5.4.1. Experimentos de avaliação

Esta seção apresenta uma avaliação do componente Catálogo. A metodologia de avaliação empregada foi baseada em um conjunto de experimentos práticos, descritos adiante. O objetivo central desta avaliação é demonstrar a capacidade deste componente em obter *Thing Descriptions* dos dispositivos IoT, compor uma base de dados com tais informações e possibilitar um mecanismo de busca, conforme as características requeridas pelo domínio de interesse. Foram planejados dois experimentos em que: (i) relata o processo de obtenção de *Thing Descriptions* em um contexto de disponibilidade massiva de dispositivos IoT virtuais em uma plataforma de nuvem computacional; (ii) realiza a medição de desempenho referente ao processo de busca de dispositivos IoT virtuais que atendam a determinados critérios no repositório implementado.

As seções 5.4.1.1 e 5.4.1.2 apresentam, em mais detalhes, os experimentos (i), (ii) e (iii), e discute os resultados obtidos. A seção 5.4.2 apresenta as considerações finais.

5.4.1.1. Experimento 1: Inclusão massiva de *Thing Descriptions*

Com intuito de validar a realização da camada catálogo, foi realizado um experimento reduzido, porém representativo de inclusão de *Thing Descriptions* na estrutura de armazenamento de dados do catálogo. Para isto, uma amostra de 100 dispositivos IoT Virtuais foi selecionada, dentre os dispositivos previamente gerados pelo *Virtual Thing Generator*, sendo hospedados na plataforma de nuvem computacional AWS. O *Thing Description Manager* – componente presente no Catálogo, foi utilizado para obter os metadados dos 100 dispositivos IoT Virtuais e persistir tais informações no *Thing Description Repository*. O tempo da operação de obtenção dos *Thing Descriptions* (TDs) de cada dispositivo e sua respectiva persistência no repositório foi medido e é apresentado na Figura 49.

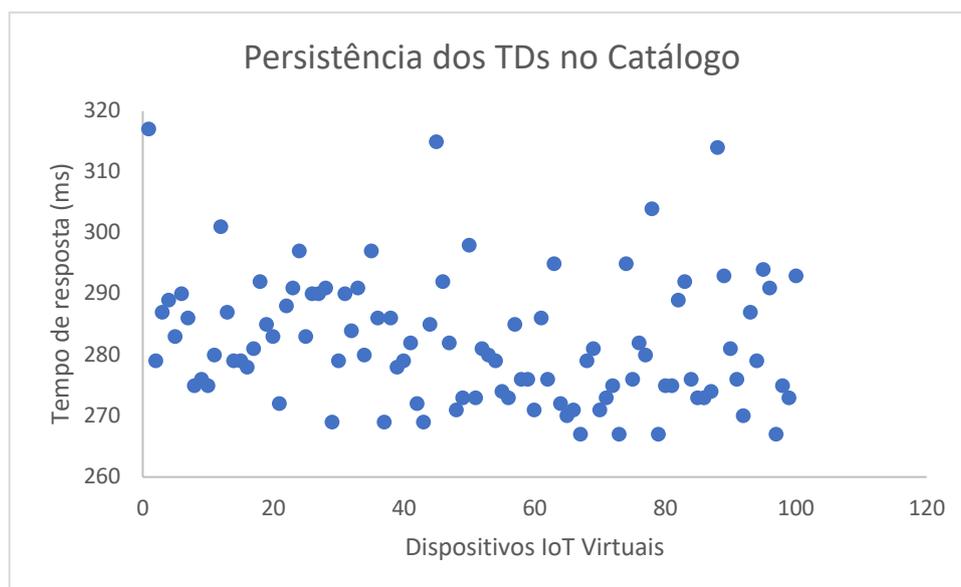


Figura 49 - Persistência dos *Thing Descriptions* no Catálogo.

Como pode ser observado, o catálogo implementado foi capaz de requisitar os metadados dos dispositivos e armazenar tal informação, com a maior parte tempo de operação entre 270 ms e 300 ms, tendo um conjunto minoritário de requisições com tempo de resposta superior a 300ms. A implementação realizada se mostrou consistente em sua proposta, realizando a obtenção e persistência dos *Thing Descriptions* dos dispositivos IoT dentro de um intervalo relativamente baixo e com pequena variabilidade em termos do tempo total da operação.

5.4.1.2. Experimento 2: Busca por dispositivos IoT

Com objetivo de validar a realização da camada catálogo no que se refere ao mecanismo de busca, foi realizado um experimento representativo de busca de *Thing Descriptions* na estrutura de armazenamento de dados do catálogo. Para isto, 5 consultas foram selecionadas a serem executadas pelo *Search Things Service* em um universo de cerca de 50 mil *Thing Descriptions* armazenados no *Repository Thing Description*. O tempo da operação de obtenção dos dispositivos IoT selecionados por cada consulta – realizada em conformidade com a *string* de busca do banco de dados *noSQL MongoDB*, foi medido e é apresentado a seguir:

- a. Busca por dispositivos que possuem o sensor do tipo umidade.
- b. Busca por dispositivos que possuem um GPS.
- c. Busca por dispositivos que possuem um GPS cuja unidade de medida retornada é do tipo *geo:Point*.
- d. Busca por dispositivos que possuem GPS que retornam o tipo *geo:Point* ou dispositivos que possuam o sensor de fumaça que retornam o tipo *m3-lite:PPM* (Partículas por milhão).
- e. Busca por dispositivos capazes de obter informações a respeito de poluição e que possuem GPS, sensor de Monóxido de Carbono ou Dióxido de Carbono, motor, odômetro e medidor de velocidade.

Tabela 10 - Consultas realizadas no catálogo implementado.

Id	String de busca	Tempo
a	"properties.humidity.type": "m3-lite:Humidity"	5ms
b	properties.gps.type:"m3-lite:GPSSensor"	1ms
c	"properties.gps.unit of measurement": "geo:Point"	5ms
d	\$or:[{"properties.gps.unit of measurement": "geo:Point"}, {"properties.smoke.unit of measurement": "m3-lite:PPM"}]	9ms
e	"properties.gps.type":"m3-lite:GPSSensor", \$or:[{"properties.carbon_dioxide.type":"saref:Sensor"}, {"properties.carbon_monoxide.type":"saref:Sensor"}], "properties.engine.type":"saref:Actuator", "properties.odometer.type":"m3-lite:Odometer", "properties.speed.type":"saref:Sensor",	1ms

Como pode ser observado, o catálogo implementado foi capaz de realizar a seleção dos dispositivos, conforme suas características físicas, com a maior parte tempo de operação variando entre 1 ms e 9 ms. O resultado gerado por cada consulta é uma lista de URLs que representam os dispositivos que atendem aos critérios definidos na *string* de busca.

5.4.2. Considerações dos Experimentos 1 e 2

A implementação de um ambiente capaz de armazenar metadados de dispositivos IoT e prover um mecanismo de busca possibilitou evidenciar sua relevância do catálogo no contexto de disponibilidade extensa e diversa de Coisas. A característica principal deste ambiente é a adoção da recomendação W3C *Web of Things* como estratégia para uniformizar a maneira de expor as funcionalidades de um dispositivo e a forma de interagir com as funcionalidades disponíveis no mesmo. O objetivo é fornecer uma infraestrutura simples, porém útil, dotada de funcionalidades importantes para apoio às pesquisas que exploram cenários com um conjunto extenso de dispositivos IoT, com diversidade de interações (propriedades, ações e eventos).

Os experimentos realizados evidenciaram o potencial da camada catálogo, e permitiram comprovar seu propósito em requisitar *Thing Descriptions* em um conjunto de dispositivos IoT e a eficiência em selecionar dispositivos IoT por meio da busca dos metadados registrados em sua base de dados noSQL. Desta forma, foi possível compor uma estrutura capaz de armazenar e selecionar dispositivos IoT conforme os requisitos do domínio de aplicação, promovendo uma importante funcionalidade para a arquitetura proposta.

5.5. Realização da Camada Composição de Cenários

Como visto, a camada *Scenario* (Figura 50) disponibiliza às aplicações uma plataforma que suporta o processo de concepção de um cenário, incluindo as etapas de seleção de catálogos, seleção de dispositivos e composição de regras do domínio.

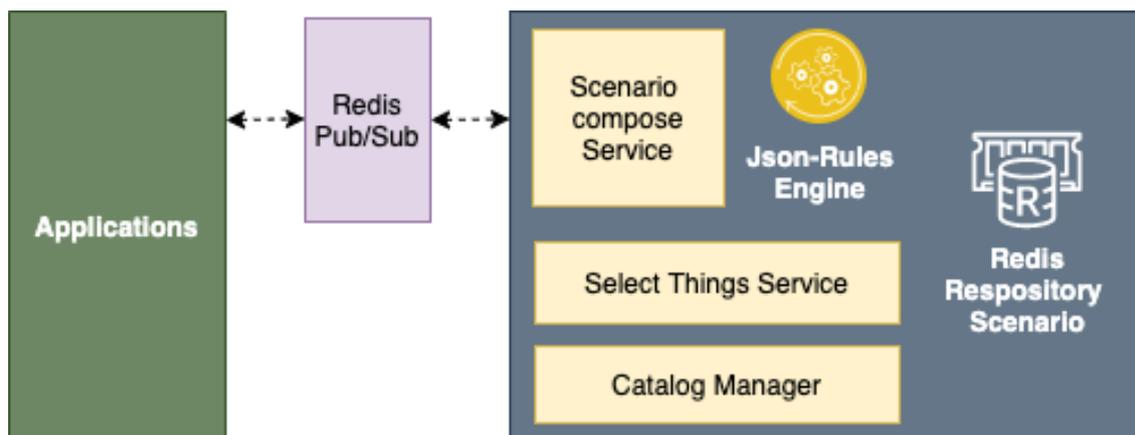


Figura 50 - Implementação da camada Scenario.

A composição de cenários consiste em duas principais atividades: (i) o processo de seleção de dispositivos que atendam aos requisitos funcionais do domínio e (ii) a descrição das regras do domínio que tem a função de analisar o estado dos dispositivos selecionados na etapa anterior. Com objetivo de realizar a atividade (i) foi implementado, utilizando as tecnologias *react* e *nodejs*, uma estrutura capaz de selecionar dispositivos, em múltiplas instâncias de catálogos, considerando a semântica formal associada aos elementos que compõe uma Coisa. Em relação a (ii), foi desenvolvido um mecanismo, utilizado a tecnologia *nodejs*, capaz de receber uma lista de dispositivos selecionados em (i), obter o estado dos mesmos que serão utilizados como fatos em uma máquina de regras. A máquina de regras deve propiciar a descrição de regras de alto nível do domínio de interesse, de forma dinâmica e flexível. Como resultado, serão retornados os dispositivos que atendem aos critérios de estado definido pelo especialista do domínio.

Em mais detalhes, foi desenvolvida uma funcionalidade capaz de receber uma lista de URLs – referentes aos dispositivos IoT - e realizar a obtenção do estado das propriedades legíveis de cada dispositivo, utilizando como parte da solução a biblioteca *thingweb.node-wot*¹¹ da Eclipse Foundation. Como resultado, é gerada uma lista de objetos JSON contendo o estado particular de cada dispositivo fornecido como entrada do algoritmo. Esta lista de objetos JSON se torna o valor de entrada, em conjunto com as regras que selecionam os estados de interesse, da máquina de regras adotada, à saber *json-rules-engine*¹². A máquina de regras retorna os objetos JSON cujos estados de realizam o casamento com as regras previamente estabelecidas.

¹¹ <https://github.com/eclipse/thingweb.node-wot>

¹² <https://www.npmjs.com/package/json-rules-engine>

A tecnologia Redis foi utilizada para armazenar a estrutura do cenário – nome, regras que estabelecem os estados de interesse e a lista de URLs de dispositivos IoT, desta forma, possibilitando estabelecer uma lista de todos os cenários previamente registrados. A estrutura de *publish/subscribe* fornecida pelo Redis¹³ é utilizada como forma de interação entre a infraestrutura e a aplicação interessada no cenário. Desta forma, aplicações que tem interesse em um determinado cenário podem realizar a subscrição no mesmo e ser notificado caso ocorra quaisquer mudanças na seleção dos dispositivos de interesse.

5.5.1. Definindo estados de interesse

Como visto na seção anterior, uma máquina de regras foi adotada como estratégia para definir e selecionar estados de interesse do cenário. De acordo com (CARDOSO; VALETTE, 1997), a modelagem deve buscar estabelecer a correspondência mais clara e direta possível entre os objetos físicos e as entidades matemáticas correspondentes. Nesse sentido, Redes de Petri se mostram uma escolha adequada, pois possuem uma relação direta com os elementos que compõe um sistema baseado em regras.

Sistemas baseados em regras são sistemas computacionais que empregam o paradigma declarativo de programação para a resolução de problemas (HAYES-ROTH, 1985). Neste paradigma, o conhecimento relativo ao domínio do problema é representado através de regras (instruções), descritas na forma *condição* → *ação*, as quais são usadas para derivar conclusões a partir de premissas (HILL, 2003). A solução do problema consiste em descrever o que deve ser feito, omitindo-se grande parte do como fazer, cabendo ao sistema decidir qual a melhor escolha para uma solicitação específica.

Um sistema baseado em regras é tipicamente formado por:

- Uma base de fatos, representando o conhecimento disponível sobre o sistema;
- Uma base de regras, que permite deduzir novos fatos;
- Um mecanismo de inferência, chamado motor de inferência, que permite realizar novas deduções, aplicando as regras aos fatos.

¹³ <https://redis.io>

O mecanismo de inferência mais elementar consiste em dispor o conjunto de regras numa lista, e percorrê-la sequencialmente. Se a parte condição da regra é verdadeira no contexto atual (correspondente a um fato na base de fatos), a regra é aplicada ou disparada. Se nenhuma regra é aplicável ou se um fato terminal (conclusão que se deseja deduzir) é deduzido, o mecanismo de inferência é finalizado.

Na maioria das vezes, em um dado contexto, várias regras podem ser aplicadas. O resultado da dedução, naturalmente, pode ser diferente segundo a regra escolhida. O mecanismo de inferência mais elementar escolhe a primeira regra, sem verificar se existem outras que podem também ser aplicadas.

Uma situação onde diversas regras podem ser aplicáveis constitui um conflito. A resolução do conflito é realizado pelo controle do motor de inferência, que pode ser bastante complexo. A correspondência entre a rede de Petri e os sistemas baseado em regras é baseada nas correspondências seguintes (CARDOSO; VALETTE, 1997):

- O conjunto de transições da rede, com suas condições e ações, representadas, respectivamente, pelas matrizes de *Pre* e *Post*, constituem a base de regras.
- A marcação inicial correspondente à base de fatos inicial, ou contexto inicial;
- O controle é dado pela estrutura da rede: se as transições são paralelas, a ordem de disparo é indiferente; se as transições estão em conflito efetivo, apenas uma poderá ser disparada.

A vantagem da utilização de rede de Petri para representar um sistema baseado em regras é devido seu formalismo, que permite a análise. Um sistema baseado em regras, diferente de um sistema descrito utilizando lógica clássica, não permite verificação formal. Entretanto, permite representar sistemas baseados em conhecimento e obter resultados quando não existe um algoritmo conhecido, como é o caso de sistemas especialistas.

Capítulo 6

6. Estudo de Caso

O objetivo deste capítulo é descrever cenários de mais alto nível, compostos por arranjos de dispositivos IoT virtuais, considerando suas funcionalidades e estados, podendo ser reutilizados por aplicações distintas a fim de explorar cenários de interesse. A seção 6.1 apresenta a visão geral do estudo de caso escolhido. A seção 6.2 trata da formalização do cenário utilizando redes de Petri a objetos e o respectivo mapeamento das transições para *string* de busca da Camada Catálogo e regras para seleção de estados de interesse da camada Cenário. A seção 6.3 apresenta a infraestrutura concebida para viabilização de um conjunto massivo de dispositivos IoT virtuais. Por fim, a seção 6.4 discute o estudo de caso e apresenta a interação entre os elementos da arquitetura e as escolhas tecnológicas da implementação realizada.

6.1. Estudo de caso – Contramedidas da poluição do ar – na perspectiva de uma cidade

Conforme já apresentado, o estudo (PRESSER; VESTERGAARD; GANEA, 2014) realizou um extenso levantamento de novos cenários a serem explorados com o advento da disponibilidade massiva de dispositivos no contexto de cidades inteligentes. Neste estudo de caso, foi escolhido o cenário nº 9 como uma amostra representativa de possibilidades que a arquitetura proposta nesta tese é capaz de prover. A seguir é descrito o cenário em questão.

Alice acabou de entrar no carro para dirigir dos subúrbios para a cidade onde trabalha. Seu GPS se conecta ao serviço de trânsito da cidade e sugere uma rota possível com o objetivo de reduzir as concentrações de poluentes no ar. Alice começa sua viagem. No caminho para a cidade, o serviço de trânsito sugere a Alice que ela deveria estacionar o carro em uma estação de trem e pegar o metrô. A taxa de estacionamento é gratuita, mas ela precisa pagar pela passagem de

metrô. O bilhete de metrô é mais barato que o imposto de congestionamento (dinâmico) imposto para áreas de tráfego intenso.

O cenário supracitado pode ser derivado em três cenários, conforme descrito a seguir:

- *C₁ - Carros em trânsito*

...entrar no carro para dirigir dos subúrbios para a cidade onde trabalha.

Seu GPS se conecta ao serviço de trânsito da cidade...

O cenário C₁ demonstra a necessidade de identificar veículos automotores que estão em uma determinada região geográfica e que estejam em funcionamento. Observe que considerando a disponibilidade massiva de dispositivos e o ambiente inerentemente dinâmico, em que diariamente novos carros são fabricados e o estado de interesse – em funcionamento, impõe um grande desafio no processo de seleção dos elementos que compõem o cenário C₁.

- *C₂ – Áreas com concentração de poluentes*

*...sugere uma rota possível com o objetivo de **reduzir as concentrações de poluentes no ar***

*...o **imposto de congestionamento (dinâmico)** imposto para áreas de tráfego intenso.*

O cenário C₂ evidencia a necessidade de identificar áreas em que foi identificado índices que caracterizam um fator de poluição acima do tolerável.

- *C₃ – Vagas de estacionamento*

*...o serviço de trânsito sugere a Alice que ela deveria **estacionar o carro em uma estação de trem e pegar o metrô...***

Por fim, o cenário C₃ demonstra a necessidade de identificar a disponibilidade de vagas de estacionamento em uma determinada região de interesse. A característica dinâmica está relacionada ao fator de ocupação dos estacionamentos e em eventuais novos estacionamentos.

A seção 6.2 discute a formalização do cenário supracitado. A seção 6.3 apresenta o uso do gerador de dispositivos IoT virtuais, utilizado com o objetivo de prover, de forma

simulada, a infraestrutura massiva de dispositivos existentes no contexto de uma cidade inteligente. A seção 6.4 discute o suporte que a arquitetura proposta nesta tese é capaz de oferecer ao desenvolvedor de aplicações que desejam explorar o cenário em questão. Por fim, a seção 6.5 apresenta uma discussão sobre o uso da arquitetura.

6.2. Formalização do cenário - Contramedidas da poluição do ar

Conforme já discutido na seção 4.2.4.1, Rede de Petri a objetos foi adotada como instrumento de formalização do conceito cenário. As seções 6.2.1, 6.2.2 e 6.2.3 apresentam a formalização dos cenários C_1 , C_2 e C_3 identificados na seção 6.1.

6.2.1. C_1 - Carros em trânsito

A representação gráfica do modelo de Rede de Petri a Objetos do cenário – “*Contramedidas da poluição do ar*” - “ *C_1 - Carros em trânsito*” é apresentada na Figura 51. O objetivo é fornecer às aplicações interessadas a relação dispositivos, possivelmente veículos que possuem motor, gps e odômetro, cujo motor esteja ligado.

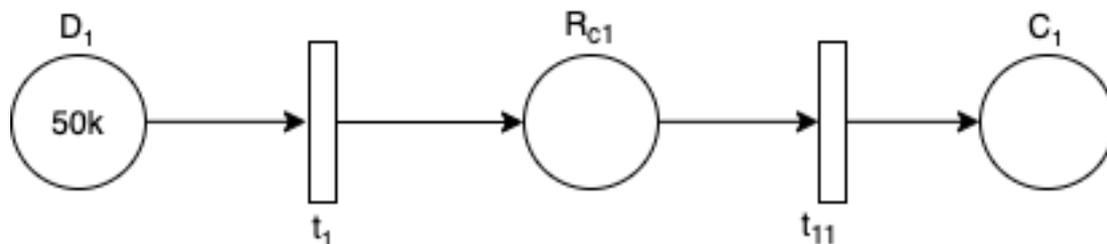


Figura 51 - Cenário C_1 .

A rede de Petri ilustrada pela Figura 51 é formalmente descrita como:

Um conjunto de lugares, $P = \langle D_0, R_{s1}, S_1 \rangle$, onde:

- D_1 representa a lista de dispositivos IoT disponíveis em um catálogo.
- R_{c1} representa a lista de dispositivos IoT que atendem aos critérios de seleção de características físicas, descrito em t_1 .
- C_1 ilustra o subconjunto de estados de interesse dos dispositivos IoT presentes no lugar R_{s1} , sendo os estados de interesse definidos em t_{11} .

Um conjunto de transições, $T = \langle t_1, t_{11} \rangle$, onde:

- t_1 define as regras de sensibilização que é mapeado para uma *string* de busca em conformidade com o *Thing Description Repository*.
- t_{11} define as regras de sensibilização que é mapeado para um conjunto de regras a ser utilizados a fim de definir os estados de interesse dos dispositivos presentes em R_{c1} .

Um conjunto de variáveis V , cujos tipos são dados por Classes $C_{class} = \{\text{Dispositivo, Interação}\}$ com:

- O conjunto de classes $C_{class} = \{\text{Dispositivo, Interação}\}$ com:

$$\text{Dispositivo} = \begin{cases} \text{nome: identificador do dispositivo} \\ \text{descrição: descreve o dispositivo} \\ \text{contexto: lista de ontologias} \\ \text{interação: lista de interações} \end{cases}$$

$$\text{Interação} = \begin{cases} \text{tipo: conceito ontológico que o define} \\ \text{descrição: descreve a interação} \\ \text{unidade de medida: conceito que define o valor retornado} \end{cases}$$

- As variáveis x do tipo Dispositivo: $\text{tipo}(x)=\text{Dispositivo}$.

A matriz Pre é a função precedente a cada arco de entrada de uma transição.

$$Pre = \begin{bmatrix} \langle x \rangle & 0 \\ 0 & \langle x \rangle \\ 0 & 0 \end{bmatrix}$$

A matriz $Post$ é a função seguinte a cada arco de saída de uma transição.

$$Post = \begin{bmatrix} 0 & \langle x \rangle \\ \langle x \rangle & 0 \\ 0 & 0 \end{bmatrix}$$

Uma etapa crucial é definir as regras que governam a Rede de Petri, em outras palavras, como ocorrerá a sensibilização das transições, no que diz respeito a seleção das características de interesse (a) e os estados de interesse (b).

- a) Requisitos funcionais dos dispositivos:

- $A_{tc}(t_1)$
 - $O_1 = \{\text{https://w3id.org/saref}, \text{http://purl.org/iot/vocab/m3-lite}, \text{http://ontology.eil.utoronto.ca/icity/Vehicle/1.2}\}$
 - $I_1 = \{\text{saref:Actuator}, \text{m3-lite:Odometer}, \text{m3-lite:GPSSensor}\}$
 - $U_m = \{\text{saref:OnOffFunction}, \text{m3-lite:Miles}, \text{geo:Point}\}$
 - $A_{tc}(t_1) = (x.\text{contexto} \in O_1) \wedge (x.\text{interação.tipo} \in I_1) \wedge (x.\text{interação.unidade de medida} \in U_m)$

Nesse sentido, a transição t_0 será disparada se o dispositivo contido em D_0 possuir, dentro do contexto que os definem as ontologias *Saref*¹⁴, *m3-lite*¹⁵ e *Vehicle*¹⁶, possua interações do tipo *saref:Actuator*, *m3-lite:Odometer* e *m3-lite:GPSSensor* tendo como unidade de medida respectivamente *saref:OnOffFunction*, *m3-lite:Miles* e *m3-lite:GPSSensor*.

b) Estados de interesse:

- $A_{tc}(t_{11}) = x.\text{interação.engine} = 1$

Por fim, a transição t_1 será sensibilizada caso o veículo selecionado na etapa anterior possua o motor em estado ligado.

Em termos de implementação, a regra supracitada deve ser mapeada em uma linguagem de busca compatível com a tecnologia adotada no *Thing Description Repository* do catálogo. Desta maneira, o *Search Description Service* é capaz de receber o parâmetro de busca e retornar a lista de URLs dos dispositivos que atendem aos requisitos funcionais elencados. Segue a seguir a *string* que reflete os requisitos:

```
"properties.engine.type": "saref:Actuator",
"properties.odometer.type": "m3-lite:Odometer",
"properties.gps.type": " m3-lite:GPS",
"@context.1.Vehicle": "http://ontology.eil.utoronto.ca/icity/Vehicle/1.2"
```

A lista de dispositivos retornada pelo *Search Description Service* é utilizada como parâmetro de entrada para o *Scenario Compose Services*, sendo este o responsável por obter o estado atual dos componentes dos dispositivos, compondo assim, a lista de fatos

¹⁴ <https://w3id.org/saref>

¹⁵ <http://purl.org/iot/vocab/m3-lite>

¹⁶ <http://ontology.eil.utoronto.ca/icity/Vehicle/1.2/>

para a máquina de regras *json-rules*. As regras são definidas em conformidade com a especificação definidas em b), sendo compatíveis com a tecnologia *json-rules*, conforme pode ser visto abaixo.

```
var veiculos = {
  all: [
    {
      fact: 'properties',
      path: '$.engine.value',
      operator: 'equal',
      value: 1
    },
  ]
}
```

6.2.2. C_2 - Áreas com concentração de poluentes

A representação gráfica do modelo de rede de Petri a objetos do cenário – “*Contramedidas da poluição do ar*” - “ C_2 - Áreas com concentração de poluentes” é apresentada na Figura 52:

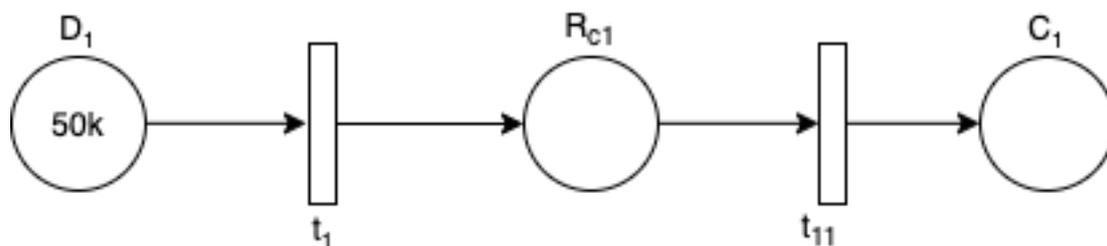


Figura 52 - Cenário C_2 .

A rede de Petri ilustrada pela Figura 52 é formalmente descrita como:

Um conjunto de lugares, $P = \langle D_0, R_{s1}, S_1 \rangle$, onde:

- D_1 representa a lista de dispositivos IoT disponíveis em um catálogo.
- R_{c1} representa a lista de dispositivos IoT que atendem aos critérios de seleção de características físicas, descrito em t_0 .
- C_1 ilustra o subconjunto de estados de interesse dos dispositivos IoT presentes no lugar R_{s1} , sendo os estados de interesse definidos em t_1 .

Um conjunto de transições, $T = \langle t_1, t_{11} \rangle$, onde:

- t_1 define as regras de sensibilização que é mapeado para uma *string* de busca em conformidade com o *Thing Description Repository*.
- t_{11} define as regras de sensibilização que é mapeado para um conjunto de regras a ser utilizados a fim de definir os estados de interesse dos dispositivos presentes em R_{c1} .

Um conjunto de variáveis V , cujos tipos são dados por Classes $C_{class} = \{\text{Dispositivo}, \text{Interação}\}$ com:

- O conjunto de classes $C_{class} = \{\text{Dispositivo}, \text{Interação}\}$ com:

$$\text{Dispositivo} = \begin{cases} \text{nome: identificador do dispositivo} \\ \text{descrição: descreve o dispositivo} \\ \text{contexto: lista de ontologias} \\ \text{interação: lista de interações} \end{cases}$$

$$\text{Interação} = \begin{cases} \text{tipo: conceito ontológico que o define} \\ \text{descrição: descreve a interação} \\ \text{unidade de medida: conceito que define o valor retornado} \end{cases}$$

- As variáveis x do tipo Dispositivo: $\text{tipo}(x) = \text{Dispositivo}$.

A matriz Pre é a função precedente a cada arco de entrada de uma transição.

$$\text{Pre} = \begin{bmatrix} \langle x \rangle & 0 \\ 0 & \langle x \rangle \\ 0 & 0 \end{bmatrix}$$

A matriz Post é a função seguinte a cada arco de saída de uma transição.

$$\text{Post} = \begin{bmatrix} 0 & \langle x \rangle \\ \langle x \rangle & 0 \\ 0 & 0 \end{bmatrix}$$

O passo seguinte se refere a definição das regras que estabelecem o funcionamento da Rede de Petri, ou seja, como será a sensibilização das transições, no que diz respeito à seleção das características de interesse (a) e os estados de interesse (b).

- a) Requisitos funcionais dos dispositivos:

- $A_{tc}(t_1)$
 - $O_1 = \{\text{https://w3id.org/saref}, \text{http://purl.org/iot/vocab/m3-lite}\}$
 - $I_1 = \{\text{saref:Sensor}, \text{m3-lite:GPSSensor}, \text{saref:Smoke}\}$
 - $U_m = \{\text{m3-lite:CO2}, \text{m3-lite:PPM}, \text{geo:Point}\}$
 - $A_{tc}(t_1) = (x.\text{contexto} \in O_1) \wedge (x.\text{interação.tipo} \in I_1) \wedge (x.\text{interação.unidade de medida} \in U_m)$

Nesse sentido, a transição t_0 será disparada se o dispositivo contido em D_0 possuir, dentro do contexto que os definem as ontologias *Saref*¹⁷ e *Om*¹⁸ e possua uma interação do tipo *saref:Temperature* e sua unidade de medida seja *om:Celsius_temperature_scale*.

b) Estados de interesse:

- $A_{tc}(t_{11})$
 - $A_{tc}(t_{11}) = x.\text{interação.carbon_monoxide} > 400 \vee x.\text{interação.smoke} > 2000$

Por fim, a transição t_{11} será sensibilizada caso o dispositivo detecte que o monóxido de carbono esteja superior a 400 partículas por milhão ou haja uma concentração superior a 2000 partículas por milhão oriundas do sensor de fumaça.

Em termos de implementação, a regra supracitada deve ser mapeada em uma linguagem de busca compatível com a tecnologia adotada no *Thing Description Repository* do catálogo. Desta maneira, o *Search Description Service* é capaz de receber o parâmetro de busca e retornar a lista de URLs dos dispositivos que atendem aos requisitos funcionais elencados em a). Segue a seguir a *string* que reflete os requisitos:

```

$or:[{"properties.engine.smoke": " saref:Smoke",
"properties.carbon_monoxide.type": " saref:Sensor"}],
"properties.gps.type": " m3-lite:GPS",

```

A lista de dispositivos retornada pelo *Search Description Service* é utilizada como parâmetro de entrada para o *Scenario Compose Services*, sendo este o responsável por obter o estado atual dos componentes dos dispositivos, compondo assim, a lista de fatos para a máquina de regras *json-rules*. As regras são definidas em conformidade com a

¹⁷ <https://w3id.org/saref>

¹⁸ <http://www.wurvoc.org/vocabularies/om-1.8>

especificação definidas em b), sendo compatíveis com a tecnologia *json-rules*, conforme pode ser visto abaixo.

```
var concentracao_poluentes = {
  any: [
    {
      fact: 'properties',
      path: '$. carbon_monoxide.value',
      operator: 'greaterThanInclusive',
      value: 400
    },
    {
      fact: 'properties',
      path: '$. smoke.value',
      operator: 'greaterThanInclusive',
      value: 2000
    }
  ],
}
```

6.2.3. C₃ - Vagas de estacionamento

A representação gráfica do modelo de rede de Petri a objetos do cenário – “*Contramedidas da poluição do ar*” - “*C₁ - Carros em trânsito*” é apresentada na Figura 51:

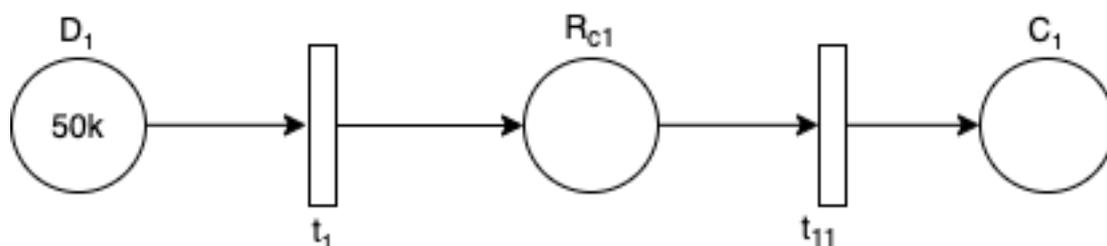


Figura 53 - Cenário C₃.

A rede de Petri ilustrada pela Figura 53 é formalmente descrita como:

Um conjunto de lugares, $P = \langle D_0, R_{s1}, S_1 \rangle$, onde:

- D_1 representa a lista de dispositivos IoT disponíveis em um catálogo.

- R_{c1} representa a lista de dispositivos IoT que atendem aos critérios de seleção de características físicas, descrito em t_0 .
- C_1 ilustra o subconjunto de estados de interesse dos dispositivos IoT presentes no lugar Rs_1 , sendo os estados de interesse definidos em t_1 .

Um conjunto de transições, $T = \langle t_1, t_{11} \rangle$, onde:

- t_1 define as regras de sensibilização que é mapeado para uma *string* de busca em conformidade com o *Thing Description Repository*.
- t_{11} define as regras de sensibilização que é mapeado para um conjunto de regras a ser utilizados a fim de definir os estados de interesse dos dispositivos presentes em R_{c1} .

Um conjunto de variáveis V , cujos tipos são dados por Classes $C_{class} = \{\text{Dispositivo}, \text{Interação}\}$ com:

- O conjunto de classes $C_{class} = \{\text{Dispositivo}, \text{Interação}\}$ com:

$$\text{Dispositivo} = \begin{cases} \text{nome: } \textit{identificador do dispositivo} \\ \text{descrição: } \textit{descreve o dispositivo} \\ \text{contexto: } \textit{lista de ontologias} \\ \text{interação: } \textit{lista de interações} \end{cases}$$

$$\text{Interação} = \begin{cases} \text{tipo: } \textit{conceito ontológico que o define} \\ \text{descrição: } \textit{descreve a interação} \\ \text{unidade de medida: } \textit{conceito que define o valor retornado} \end{cases}$$

- As variáveis x do tipo Dispositivo: $\text{tipo}(x) = \text{Dispositivo}$.

A matriz Pre é a função precedente a cada arco de entrada de uma transição.

$$\text{Pre} = \begin{bmatrix} \langle x \rangle & 0 \\ 0 & \langle x \rangle \\ 0 & 0 \end{bmatrix}$$

A matriz Post é a função seguinte a cada arco de saída de uma transição.

$$\text{Post} = \begin{bmatrix} 0 & \langle x \rangle \\ \langle x \rangle & 0 \\ 0 & 0 \end{bmatrix}$$

A seguir é apresentado como ocorrerá a sensibilização das transições, no que diz respeito à seleção das características de interesse (a) e os estados de interesse (b).

a) Requisitos funcionais dos dispositivos:

- $A_{tc}(t_1)$
 - $O_1 = \{\text{http://ontology.eil.utoronto.ca/icity/Parking/1.2}, \text{http://ontology.eil.utoronto.ca/icity/OM/1.2}\}$
 - $I_1 = \{\text{OM:Geo_Position}, \text{OM:CapacitySize}\}$
 - $U_m = \{\text{geo:point}, \text{xsd:positiveInteger}\}$
 - $A_{tc}(t_1) = (x.\text{contexto} \in O_1) \wedge (x.\text{interação.tipo} \in I_1) \wedge (x.\text{interação.unidade de medida} \in U_m)$

Nesse sentido, a transição t_0 será disparada se o dispositivo contido em D_0 possuir, dentro do contexto que os definem as ontologias *Saref*¹⁹ e *Om*²⁰ e possua uma interação do tipo *saref:Temperature* e sua unidade de medida seja *om:Celsius_temperature_scale*.

b) Estados de interesse:

- $A_{tc}(t_{11})$
 - $A_{tc}(t_{11}) = x.\text{interação.CapacitySize} > 0$

Por fim, a transição t_1 será sensibilizada caso o estacionamento possua uma capacidade superior a 0 vagas.

Em termos de implementação, a regra supracitada deve ser mapeada em uma linguagem de busca compatível com a tecnologia adotada no *Thing Description Repository* do catálogo. Desta maneira, o *Search Description Service* é capaz de receber o parâmetro de busca e retornar a lista de URLs dos dispositivos que atendem aos requisitos funcionais elencados em a). Segue a seguir a *string* que reflete os requisitos:

```
"properties.Geo_Position.type": " geo:point",
"properties.CapacitySize.type": " xsd:positiveInteger",
"@context.1.Parking": " http://ontology.eil.utoronto.ca/icity/Parking/1.2"
```

¹⁹ <https://w3id.org/saref>

²⁰ <http://www.wurvoc.org/vocabularies/om-1.8>

A lista de dispositivos retornada pelo *Search Description Service* é utilizada como parâmetro de entrada para o *Scenario Compose Services*, sendo este o responsável por obter o estado atual dos componentes dos dispositivos, compondo assim, a lista de fatos para a máquina de regras *json-rules*. As regras são definidas em conformidade com a especificação definidas em b), sendo compatíveis com a tecnologia *json-rules*, conforme pode ser visto abaixo.

```
var estacionamento = {
  all: [
    {
      fact: 'properties',
      path: '$. CapacitySize.value',
      operator: 'greaterThan',
      value: 0
    }, ]
}
```

6.3. Infraestrutura massiva de dispositivos IoT

O cenário Contramedidas da poluição do ar – na perspectiva de uma cidade inteligente ilustra um exemplo representativo do potencial retido em termos de desenvolvimento de novas soluções diante da disponibilidade massiva de dispositivos IoT. Com intuito de explorar este cenário e demonstrar a aplicação da arquitetura proposta, foi utilizado o ambiente de geração massiva IoT, apresentada na seção 5.2.1, para a geração de 50 mil dispositivos virtuais, em conformidade com a recomendação *W3C Web of Things*. Cada dispositivo tem até 10 interações distintas, no universo de 35 interações registradas, conforme Tabela 11.

Tabela 11 - interações registradas no gerador de dispositivos IoT virtuais.

Interação	Contexto	Unidade de medida
temperature	saref:Temperature	om:Celsius temperature scale
temperature	saref:Temperature	om:Fahrenheit temperature scale
temperature	saref:Temperature	om:Rankine temperature scale
luminosity	saref:Lighting device	om:Luminance unit
humidity	saref:Humidity	m3-lite:Humidity
gps	m3-lite:GPSSensor	geo:Point
ultrasonic	saref:Sensor	m3-lite:Centimetre
magnetic	m3-lite:Magnetometer	m3-lite:Tesla
pressure	saref:Pressure	m3-lite:Pascal

weight	m3-lite:WeightSensor	m3-lite:Kilo
heart_rate	m3-lite:HeartBeatSensor	m3-lite:BeatPerMinute
oxygen	saref:Sensor	m3-lite:PPM
speed	saref:Sensor	m3-lite:KilometerPerHour
speed	saref:Sensor	m3-lite:MeterPerSecond
height	saref:Sensor	m3-lite:Inch
precipitation	m3-lite:PrecipitationSensor	m3-lite:MilligramPerCubicMetre
carbon_monoxide	saref:Sensor	m3-lite:PPM
carbon_dioxide	saref:Sensor	m3-lite:CO2
Pedometer	m3-lite:Pedometer	xsd:positiveInteger
sound	m3-lite:SoundSensor	m3-lite:Decibel
led	saref:Lighting device	saref:OnOffFunction
light	m3-lite:LightSensor	m3-lite:Lux
engine	saref:Actuator	saref:OnOffFunction
smoke	saref:Smoke	m3-lite:PPM
accelerometer	saref:Sensor	om:Angular_acceleration_unit
gyroscope	m3-lite:GyroscopeSensor	om:Angular_acceleration_unit
odometer	m3-lite:Odometer	m3-lite:Miles
falldetector	m3-lite:FallDetector	saref:OnOffState
buzzer	saref:Actuator	saref:OnOffFunction
ozone	saref:Sensor	m3-lite:PPM
Glucometer	m3-lite:Glucometer	m3-lite:MmolPerLiter
CapacitySize	OM:CapacitySize	xsd:positiveInteger

A infraestrutura do Instituto Federal do Espírito Santo, composta com um cluster de servidores de alta disponibilidade foi adotada para hospedagem dos dispositivos, em que foi utilizado uma máquina virtual com 4 vCPUs, 20GB de memória RAM, sendo agrupados em 50 containers, em que cada container possuindo 1 mil dispositivos virtuais, conforme pode ser visto na Figura 54.

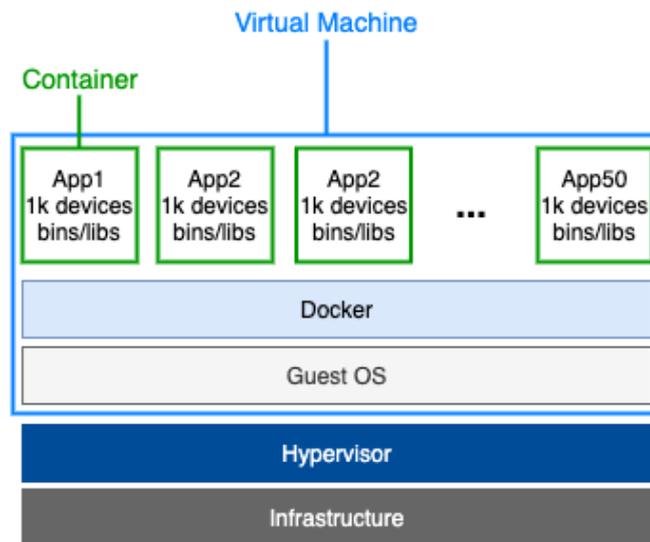


Figura 54 - Infraestrutura para provisão de um cenário massivo de dispositivos IoT virtuais

A infraestrutura física adotada é composta por servidores, storage e equipamentos de comunicação. O *Hypervisor* utilizado é o *Vmware* e o *Guest OS* – *Ubuntu 20.04*. O *docker* foi utilizado como plataforma de provisão de containers que possibilitaram a execução de 50 mil dispositivos virtuais na mesma máquina virtual. A utilização de containers foi uma solução encontrada para superar uma limitação em termos de execução simultânea de dispositivos IoT virtuais identificada na biblioteca utilizada no projeto *thingweb.node-wot*.

A Figura 55 apresenta a lista de dispositivos contidos no primeiro container. Cada container é identificado pela porta indicada no final da URL, variando entre as portas 8080 – 8129.



Figura 55 - Dispositivos IoT Virtuais hospedados na infraestrutura do Ifes.

Uma instância do componente *catalog* da arquitetura proposta foi hospedada na plataforma em nuvem *Amazon AWS* e o *Thing Description Manager Service* foi utilizado para obter os *Thing Descriptions* dos novos 50 mil dispositivos e persistir no *Repository Thing Description*.

Por fim, uma instância do componente *Scenario* da arquitetura proposta foi hospedada também na plataforma em nuvem *Amazon AWS*, viabilizando seu uso em conjunto com os 50 mil dispositivos gerados e o componente catálogo.

Conforme já discutido, para estabelecer um cenário, como primeiro passo, se faz necessário a definição dos requisitos funcionais de interesse, em seguida, compõe-se a *string* de busca e utiliza-se o componente *Search Thing Service*. O *Search Thing Service* retornará uma lista de URLs de dispositivos que atendem aos critérios do domínio em

questão. Esta lista de dispositivos é utilizada em conjunto com as regras que estabelecem os estados de interesse, para definir o cenário em questão.

6.4. Discussão

Com intuito de demonstrar a abordagem proposta, uma implementação foi realizada com base na arquitetura conceitual. Na camada *Thing*, o ambiente de geração massiva de dispositivos IoT foi utilizado a fim de prover um ambiente de disponibilidade massiva de dispositivos, no qual foram criados 50 mil dispositivos IoT virtuais com até 10 interações distintas – sendo compostas de sensores e atuadores, no universo de 35 interações previamente registradas, sendo hospedados na infraestrutura computacional do Instituto Federal do Espírito Santo. Uma instância da camada catálogo foi hospedada na plataforma de nuvem computacional Amazon AWS e sua funcionalidade – *Thing Description Manager Service* foi utilizada para obter e registrar os *Thing Descriptions* de todos os 50 mil dispositivos gerados, estabelecendo assim um catálogo populado e apto a responder a consultas implementadas pela funcionalidade *Search Thing Service*. Por fim, uma instância da camada *Scenario* foi hospedada na plataforma de nuvem computacional Amazon AWS, sendo capaz de obter e selecionar os dispositivos cujo estado de interesse atenda ao cenário Contramedidas da poluição do ar. Os requisitos funcionais foram descritos de maneira formal como uma condição de transição definidas por uma rede de Petri à Objetos e posteriormente mapeados em uma *string* de busca compatível com a linguagem de consulta do banco de dados *MongoDB*. Os estados de interesse foram definidos de maneira formal como uma condição de transição definidas por uma rede de Petri à Objetos e posteriormente mapeado em um conjunto de regras do tipo condição -> ação compatível com a tecnologia adotada como máquina de regras – *json rules engine*, sendo que os fatos inseridos na máquina de regras consiste da lista de objetos JSON que contém o registro do estado de todas as propriedades legíveis de um dispositivo IoT que atendem aos requisitos funcionais do domínio.

Foi possível observar que a aplicação conseguiu capturar, por meio da subscrição do cenário, os dispositivos cujos requisitos funcionais e estados de interesse atendiam ao domínio Contramedidas da poluição do ar. Desta maneira, aplicações podem subscrever cenários de interesse, abstraindo assim, a dinamicidade inerente ao domínio de IoT, no que se refere ao desenvolvimento e disponibilização de novos dispositivos IoT ao longo do tempo e a variação dos estados dos sensores e atuadores dos dispositivos selecionados

no âmbito do cenário. Esse nível de abstração possibilita as aplicações concentrarem esforços no desenvolvimento das regras de negócio específicas aos seu propósito, como no exemplo deste estudo de caso: “*sugere a Alice que ela deveria estacionar o carro em uma estação de trem e pegar o metrô*”, considerando o valor do imposto dinâmico e o custo do estacionamento mais próximo, abstraindo os detalhes do processo de seleção de dispositivos e estados de interesse gerais do domínio.

Capítulo 7

7. Conclusões

Neste capítulo, resumimos nossas principais conclusões, resultados acadêmicos e trabalhos futuros, discutindo brevemente o projeto da arquitetura à luz dos desafios identificados. O capítulo está estruturado como segue. A Seção 7.1 descreve o protocolo definido para a realização desta revisão; a Seção 7.2 apresenta os resultados acadêmicos obtidos; por fim, a Seção 7.3 traz os trabalhos futuros.

7.1. Considerações finais

No capítulo introdutório, iniciamos nossa discussão chamando a atenção para o crescente interesse no desenvolvimento soluções que abordam o problema da fragmentação existente no contexto de Internet das Coisas e do desafio de lidar em um cenário de disponibilidade massiva de dispositivos IoT. Enfatizamos a demanda por infraestruturas capazes de selecionar dispositivos IoT heterogêneos com requisitos que podem mudar a qualquer momento. Também ressaltamos que os ambientes de negócios são complexos devido à grande variedade de tecnologias, soluções de hardware e software integradas aos sistemas tradicionais. Defendemos, portanto, que as soluções deveriam fornecer flexibilidade, portabilidade e capacidade total de dissociação para suportar ambientes de negócios de TI tão complexos e heterogêneos. Dessa forma, os modelos de negócios podem ser redesenhados para incorporar objetos inteligentes no processo de desenvolvimento, facilitando a criação e o fornecimento de novos produtos e serviços de IoT. Isso torna as soluções de suporte, como a arquitetura proposta nesta tese, um componente estratégico para a implantação de aplicativos de IoT.

Diversidade de hardware, diferentes modelos de dados, software básico e protocolos de comunicação específicos, bem como interesses divergentes dos grandes players, fazem com que o ambiente de IoT continue sendo muito heterogêneo e, com isso, demandando uma ampla gama de ambientes para o desenvolvimento de soluções. Neste sentido, o surgimento de iniciativas que buscam tornar homogênea a maneira de descrever

e interagir com um domínio tão heterogêneo, tais como a recomendação *W3C Web of Things*, deve ser valorizada, investigada e o seu uso incentivado no projeto de novas plataformas de suporte à IoT, particularmente àquelas voltadas à IoT massiva, tendo em vista a escala da heterogeneidade. Considerando o crescimento esperado das soluções de IoT, é razoável esperar que a adoção de tais padrões impactem o sucesso de novos produtos e serviços. Em (GUBBI et al., 2013) já era reforçada a importância do desenvolvimento de tais plataformas.

Neste contexto, entendemos que a presente pesquisa, representada por uma arquitetura de código aberto, estruturada com base nos desafios da criação de aplicações finais em um contexto de disponibilidade massiva de dispositivos IoT heterogêneos, oferece uma solução alinhada aos interesses recentes de pesquisas da área, que investem e apostam na promoção da interoperabilidade semântica em IoT como o próximo passo na consolidação da Internet das Coisas.

Quando comparada com plataformas IoT que lidam com as questões de pesquisa aqui apontados, vemos que a arquitetura proposta nesta tese apresenta uma abordagem diferente, integrando elementos arquiteturais e funcionalidades que permitem capturar e modelar mais facilmente a dinamicidade e a heterogeneidade do ambiente IoT massivo, particularmente ao combinar o suporte à interoperabilidade semântica, à introdução dos catálogos de dispositivos IoT e à definição do conceito e cenários para descrever o comportamento dinâmico da aplicação com base nas situações de interesse do mundo real. Plataformas como as citadas em (HEJAZI et al., 2018), por exemplo, embora busquem estabelecer uma uniformidade em termos de sensoriamento, comunicação e integração com o ambiente de nuvem computacional, não possuem suporte semântico adequado para descrever dispositivos IoT e suas funcionalidades, e nem especificam uma maneira de como lidar com a seleção de dispositivos em um universo de disponibilidade extensiva. O trabalho de (KAEBISCH S., 2019), que inaugurou a discussão de como lidar com um conjunto massivo de dispositivos, apresenta sérias limitações de escopo, apesar de propor uma maneira de descrever de dispositivos IoT e uma forma de lidar com grupos de dispositivos. Por exemplo, a abordagem de KAEBISH não captura a dinamicidade inerente do ambiente de IoT em termos do surgimento de novos dispositivos e a variação dos seus estados. Nesse sentido, entendemos que a arquitetura proposta nesta tese apresenta uma visão mais completa e integrada dos problemas de pesquisa aqui reportados do que essas iniciativas recentes do domínio de aplicações IoT massivas.

7.2. Resultados Acadêmicos

É importante notar que, do ponto de vista acadêmico, além da produção e apresentação de artigos científicos e capítulo de livros, no Brasil e no exterior, um resultado relevante desta pesquisa de doutorado foi a escrita e aprovação de um projeto de pesquisa Universal – FAPES, captando recursos utilizados no desenvolvimento desta pesquisa, além da orientação e co-orientação de alunos de iniciação científica, focados no desenvolvimento de soluções de IoT baseadas na arquitetura proposta nesta tese. De forma complementar, ainda durante o período do doutorado, obtive a aprovação de um projeto de iniciação científica júnior, financiado também pela FAPES, onde coordenei uma equipe de pesquisa composta por 10 alunos, dois monitores e um tutor – em que foi desenvolvida uma pesquisa com intuito de implementar um dispositivo IoT para monitoramento de enchentes da bacia hidrográfica do rio Jacaraípe. Como resultados, obtivemos duas publicações de artigos e divulgação da pesquisa em entrevista para emissora de TV Local. A seguir, a lista de publicações, projetos, orientações e entrevistas de divulgação:

- Coordenador - Projeto Universal FAPES em andamento (2019 – 2021). Recurso total aprovado: R\$ 22.924,00
 - Artigo intitulado “*Semantic Interoperability in IoT: A Systematic Mapping*” foi aceito para apresentação oral em *The 12th conference on Internet of Things and Smart Spaces ruSMART*, ocorrido em agosto de 2019 em São Petersburgo – Rússia.
 - Publicação de um capítulo no livro intitulado “*Internet of Things, Smart Spaces, and Next Generation Networks and Systems*”, cujos organizadores são Olga Galinina; Sergey Andreev; Sergey Balandin; Yevgeni Koucheryavy. *Lecture Notes in Computer Science*. 01ed. Switzerland: *Springer International Publishing*, 2019, v. 11660, p. 53-64.
 - Artigo intitulado “*Um ambiente de geração massiva de dispositivos Virtuais em Internet das Coisas*” foi aceito para apresentação oral no *Brazilian Symposium on Computing Systems Engineering (SBESC 2020)*, ocorrido em novembro de 2020 – online.

- Orientação de Iniciação Científica: Lucas G. de P. Salgado (bolsista) e Rafael B. Pedruzzi (voluntário).
- Co-orientação de Iniciação científica: Nathânia Oliveira Queiroz (bolsista).
- Coordenador - Projeto PICjr FAPES concluído (2015 – 2017). Recurso total aprovado: R\$ 75.800,00
- Artigo intitulado “*Uma infraestrutura para Monitoramento de rios para previsão de enchentes*” foi aceito para apresentação oral no *SEGeT – Simpósio de Excelência em Gestão e Tecnologia* (<http://www.aedb.br/seget/>). Trata-se de um evento científico de abrangência nacional ocorrido em outubro de 2017 o seu 14º encontro em Resende – RJ.
- Artigo intitulado “*Projeto e desenvolvimento de um protótipo para monitoramento da bacia hidrográfica do rio Jacaraípe para previsão de enchentes*”. Aceito para apresentação oral no *XXI Encontro Latino Americano de Iniciação Científica*, ocorrido em outubro de 2017, São José dos Campos, São Paulo.
- Divulgação – TV Vitória – Record
 - <https://youtu.be/qtHuT-CRI54>
- Divulgação online:
 - <https://fapes.es.gov.br/Not%C3%ADcia/sistema-desenvolvido-por-alunos-de-escola-publica-preve-possibilidade-de-enchente-de-rio>
- Consultor AD-HOC Fapes.
 - Avaliação de projetos em desenvolvimento e concluídos, financiados pela Fapes, em empresas da grande vitória - ES.

7.3. Trabalhos futuros

Existe um conjunto de questões de pesquisa relacionadas a esta tese que estão sendo analisadas com perspectivas de desenvolvimento de novos trabalhos em um futuro próximo, cobrindo algumas dessas questões de pesquisa. Nossa proposta para a evolução da arquitetura é envolver os estudantes da UFES, IFES ou outras instituições no

desenvolvimento de novas funcionalidades a partir da versão da atualmente implementada, cujo código está disponível no Github em <https://github.com/si4-iot/>. Dentre os trabalhos futuros vislumbrados, podemos listar::

- Mapeamento automatizado da Rede de Petri a objetos para os elementos da arquitetura, tornando a definição conceitual do cenário o elemento central e computacionalmente funcional, criando um nível de abstração quanto à sintaxe utilizada para interagir com a implementação da camada catálogo e a camada cenário.
- Fundamentar ontologicamente o conceito Cenário, aprofundando as relações entre contexto, entidades e situações, construindo uma base teórica robusta, alinhando com os elementos da arquitetura.
- Aperfeiçoar os dispositivos IoT virtuais gerados pelo componente da arquitetura *Virtual Thing Generator*, tornando-os mais fidedignos a um dispositivo físico real, incluindo características como as encontradas nos experimentos realizados na seção 5.2.1, como tempo médio de resposta, tempo de médio de leitura, os tipos de falhas entre outros.
- Automatizar a configuração da infraestrutura necessária para a disponibilização dos dispositivos IoT virtuais gerados pelo componente da arquitetura *Virtual Thing Generator*.
- Integrar a arquitetura - ou parte dela - a outros trabalhos em desenvolvimento no LPRM, particularmente os trabalhos de telemonitoramento de pacientes em domicílio, que lidam com uma massa enorme de dados, um número grande de usuários e dispositivos heterogêneos de coleta de dados vitais dos pacientes.

REFERÊNCIAS

ABERER, K.; HAUSWIRTH, M.; SALEHI, A. **Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks**. 2007 International Conference on Mobile Data Management. **Anais...2007**

ABOWD, G. D. et al. **Towards a better understanding of context and context-awareness**. International symposium on handheld and ubiquitous computing. **Anais...1999**

AGARWAL, R. et al. **Unified IoT ontology to enable interoperability and federation of testbeds**. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). **Anais...2016**
AHMEDI, L.; JAJAGA, E.; AHMEDI, F. An Ontology Framework for Water Quality Management. **Semantic Sensor Networks 2013 (SSN2013)**, 2013.

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

BERMUDEZ-EDO, M. et al. **IoT-Lite: A Lightweight Semantic Model for the Internet of Things**. (B. J. El Baz D., Ed.)2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld). **Anais...Institute of Electrical and Electronics Engineers Inc.**, 2016a

BERMUDEZ-EDO, M. et al. **IoT-Lite: A Lightweight Semantic Model for the Internet of Things**. 2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld). **Anais...2016b**

BONINO, D.; CORNO, F.; DE RUSSIS, L. A Semantics-Rich Information Technology

Architecture for Smart Buildings. **Buildings**, v. 4, p. 880–910, 2014.

BORST, W. N. **Construction of Engineering Ontologies for Knowledge Sharing and Reuse**. Netherlands: Centre for Telematics and Information Technology (CTIT), 1997.

BOTTS, M. et al. OGC{®} sensor web enablement: Overview and high level architecture. In: **GeoSensor networks**. [s.l.] Springer, 2006. p. 175–190.

BOTTS, M.; ROBIN, A. OpenGIS Sensor Model Language (SensorML) Implementation Specification; OpenGIS Implementation Specification OGC 07-000. **Open Geospatial Consortium Inc**, 2007.

CALBIMONTE, J.-P. et al. **XGSN: An open-source semantic sensing middleware for the web of things**. CEUR Workshop Proceedings. **Anais...**Riva del Garda, Trentino, Italy: 2014

CALBIMONTE, J. J. et al. Semantic Sensor Data Search in a Large-Scale Federated Sensor Network Semantic Sensor Networks 2011 Semantic Sensor Networks 2011. **Semantic Sensor Networks**, p. 14–29, 2011.

CALHAU, R. F.; DE ALMEIDA FALBO, R. **An ontology-based approach for semantic integration**. 2010 14th IEEE International Enterprise Distributed Object Computing Conference. **Anais...**2010

CARDOSO, J.; VALETTE, R. **Redes de petri**. [s.l.] Editora da UFSC Florianópolis, 1997.

CHARPENAY, V.; KÄBISCH, S.; KOSCH, H. Introducing thing descriptions and interactions: An ontology for the web of things. **CEUR Workshop Proceedings**, v. 1783, p. 55–66, 2016.

CHEN, C.; HELAL, A. Device integration in SODA using the device description language. **Proceedings - 2009 9th Annual International Symposium on Applications and the Internet, SAINT 2009**, p. 100–106, 2009.

COMPTON, M. et al. The SSN ontology of the W3C semantic sensor network incubator group. **Journal of Web Semantics**, v. 17, p. 25–32, 2012.

CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Methodologies, tools and languages for building ontologies. Where is their meeting point? **Data & knowledge engineering**, v. 46, n. 1, p. 41–64, 2003.

COSTA, P. D. et al. Towards conceptual foundations for context-aware applications. **AAAI Workshop - Technical Report**, v. WS-06-12, p. 54–58, 2006.

COX, S. Observations and Measurements-Part 1-Observation schema (OpenGIS Implementation Standard OGC 07-022r1). **Open Geospatial Consortium Inc., Tech. Rep**, v. 8, 2007.

COX, S. **An explicit OWL representation of ISO/OGC Observations and Measurements**. SSN@ ISWC. **Anais...**2013

DANIELE, L.; DEN HARTOG, F.; ROES, J. **Created in Close Interaction with the Industry: The Smart Appliances REference (SAREF) Ontology**. Lecture Notes in Business Information Processing. **Anais...**Berlin, Germany: 2015

DATTA, S. K.; BONNET, C. Describing things in the Internet of Things: From CoRE link format to semantic based descriptions. **2016 IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW 2016**, n. i, p. 1–2, 2016.

DATTA, S. K.; BONNET, C. Advances in Web of Things for IoT Interoperability. **2018 IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW 2018**, p. 2–3, 2018.

DE SOUZA, S. C.; PEREIRA FILHO, J. G. **Semantic interoperability in {IoT:} a systematic mapping**. The 19th International Conference on Next Generation Wired/Wireless Advanced Networks and Systems + The 12th conference on Internet of Things and Smart Spaces (NEW2AN ruSMART 2019). **Anais...**St.Petersburg, Russia: ago. 2019

FALBO, R. DE A. et al. **Ontology patterns: clarifying concepts and terminology**. Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns. **Anais...**2013

FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A.; JURISTO, N. **METHONTOLOGY: From Ontological Art Towards Ontological Engineering**. Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series. **Anais...**American Association for Artificial Intelligence, 1997

FRUHWIRTH, T.; KASTNER, W.; KRAMMER, L. **A methodology for creating reusable ontologies**. Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018. **Anais...**Saint Petersburg, Russia: 2018

GANGEMI, A. et al. **Sweetening ontologies with DOLCE**. International Conference on Knowledge Engineering and Knowledge Management. **Anais...**2002

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. **International Journal of Human-Computer Studies**, v. 43, n. 5–6, p. 907–928, 1995.

GUARINO, N. **Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy**. [s.l.] IOS press, 1998. v. 46

GUARINO, N. The ontological level: Revisiting 30 years of knowledge representation. In: **Conceptual modeling: Foundations and applications**. [s.l.] Springer, 2009. p. 52–67.

GUBBI, J. et al. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future generation computer systems**, v. 29, n. 7, p. 1645–1660, 2013.

GUIZZARDI, G. **Ontological foundations for structural conceptual models**. [s.l.] CTIT, Centre for Telematics and Information Technology, 2005.

GUIZZARDI, G. On ontology, ontologies, conceptualizations, modeling languages, and

(meta) models. **Frontiers in artificial intelligence and applications**, v. 155, p. 18, 2007.

GUIZZARDI, G.; FALBO, R. A.; GUIZZARDI, R. S. S. A importância de Ontologias de Fundamentação para a Engenharia de Ontologias de Domínio: o caso do domínio de Processos de Software. **IEEE Latin America Transactions**, v. 6, n. 3, p. 244–251, 2008.

GYRARD, A. et al. Assisting IoT Projects and Developers in Designing Interoperable Semantic Web of Things Applications. n. 1, 2015.

GYRARD, A.; ZIMMERMANN, A.; SHETH, A. Building IoT based applications for Smart Cities: How can ontology catalogs help? **IEEE Internet of Things Journal**, 2018.

HASAN, S.; CURRY, E. Approximate semantic matching of events for the Internet of Things. **ACM Transactions on Internet Technology (TOIT)**, v. 14, n. 1, p. Article 2, 2014.

HAYES-ROTH, F. Rule-based systems. **Communications of the ACM**, v. 28, n. 9, p. 921–932, 1985.

HEJAZI, H. et al. **Survey of platforms for massive IoT**. 2018 IEEE International Conference on Future IoT Technologies (Future IoT). **Anais...2018**

HEVNER, A.; CHATTERJEE, S. Design science research in information systems. In: **Design research in information systems**. [s.l.] Springer, 2010. p. 9–22.

HILL, E. F. **Jess in action: Java rule-based systems**. [s.l.] Manning Publications Co., 2003.

IZZA, S. Integration of industrial information systems: from syntactic to semantic integration approaches. **Enterprise Information Systems**, v. 3, n. 1, p. 1–57, 2009.

JANOWICZ, K. et al. Semantic enablement for spatial data infrastructures. **Transactions in GIS**, v. 14, n. 2, p. 111–129, 2010.

JANOWICZ, K. et al. SOSA: A lightweight ontology for sensors, observations, samples,

and actuators. **Journal of Web Semantics**, 2018.

JAYARAMAN, P. P.; CALBIMONTE, J.; MAU, H. N. The Schema Editor of OpenIoT for Semantic Sensor Networks. **First Joint International Workshop on SEMANTIC SENSOR NETWORKS AND TERRA COGNITA SSN-TC 2015**, 2015.

KÄBISCH, S.; ANICIC, D. Thing Description as Enabler of Semantic Interoperability on the Web of Things. [s.d.].

KAEBISCH S., K. T. M. M. C. V. K. M. **Web of Things (WoT) Thing Description**, dez. 2019.

KHALED, A. E. et al. IoT-DDL-Device Description Language for the “T” in IoT. **IEEE Access**, v. 6, p. 24048–24063, 2018.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. 2004, p. 1–26, 2004.

KITCHENHAM, B. A.; BUDGEN, D.; BRERETON, O. P. Using mapping studies as the basis for further research--a participant-observer case study. **Information and Software Technology**, v. 53, n. 6, p. 638–651, 2011.

KRANENBURG, R. VAN et al. **The Internet of Things**. In 1st Berlin Symposium on Internet and Society: Exploring the Digital Future. **Anais...2011**

LE, K.-H. et al. WoT-AD: A Descriptive Language for Group of Things in Massive IoT. **2019 IEEE 5th World Forum on Internet of Things (WF-IoT)**, p. 257–262, 2019.

M. SERRANO, P. BARNAGHI, F. CARREZ, P. COUSIN, O. V. AND; FRIESS, P. Internet of Things iot semantic interoperability: Research Challenges, Best Practices, Recommendations and Next Steps. 2015.

MACIEL, P. R. M.; LINS, R. D.; CUNHA, P. R. F. **Introdução às redes de Petri e aplicações**. [s.l.] UNICAMP-Instituto de Computacao, 1996.

MARRANGHELLO, N. Redes de petri: Conceitos e aplicações. **São Paulo: DCCE/IBILCE/UNESP**, 2005.

MCGUINNESS, D. L.; VAN HARMELEN, F.; OTHERS. OWL web ontology language overview. **W3C recommendation**, v. 10, n. 10, p. 2004, 2004.

MITCHELL, R. J. **Managing complexity in software engineering**. [s.l.] IET, 1990.

MOORE, K. E.; BRENNAN, J. E. **Petri Nets and Simulation: A Tutorial**. SUMMER COMPUTER SIMULATION CONFERENCE. **Anais...**1995

NARDI, J. C.; DE ALMEIDA FALBO, R.; ALMEIDA, J. P. A. **A Panorama of the Semantic EAI Initiatives and the Adoption of Ontologies by these Initiatives**. (M. van Sinderen et al., Eds.)Enterprise Interoperability. **Anais...**Berlin, Heidelberg: Springer Berlin Heidelberg, 2013

NILES, I.; PEASE, A. **Towards a standard upper ontology**. Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. **Anais...**2001

NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology**. [s.l: s.n.].

PETRI, C. A. Kommunikation mit automaten. 1962.

PRESSER, M.; VESTERGAARD, L.; GANEA, S. Smart City Use Cases and Requirements. **CityPulse Project Deliverable D**, v. 2, p. 1, 2014.

QU, C.; LIU, F.; TAO, M. Ontologies for the Transactions on IoT. **International Journal of Distributed Sensor Networks**, v. 2015, n. 1, p. 1–12, 2015.

RIJGERSBERG, H.; VAN ASSEM, M.; TOP, J. Ontology of units of measure and related concepts. **Semantic Web**, v. 4, n. 1, p. 3–13, 2013.

RODRIGUES, T. et al. A Model-Based Approach for Building Ubiquitous Applications

Based on Wireless Sensor Network. **Mobile and Ubiquitous Systems: Computing, Networking, and Services**, p. 350–352, 2012.

SANCHEZ, L. et al. SmartSantander: IoT experimentation over a smart city testbed. **Computer Networks**, v. 61, p. 217–238, 2014.

SARMA, S.; BROCK, D.; ASHTON, K. The networked physical world. TR MIT-AUTOID-WH-001 MIT Auto-ID Centre,. **Auto-ID Center White Paper MIT- ...**, p. 1–16, 2000.

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

SHETH, A.; HENSON, C.; SAHOO, S. S. Semantic Sensor Web. **IEEE Internet Computing**, v. 12, n. 4, p. 78–83, 2008.

SHETH, A. P. Changing focus on interoperability in information systems: from system, syntax, structure to semantics. In: **Interoperating geographic information systems**. [s.l.] Springer, 1999. p. 5–29.

SHI, F. et al. A survey of data semantization in internet of things. **Sensors (Switzerland)**, v. 18, n. 1, 2018.

SIMPERL, E. Reusing ontologies on the Semantic Web: A feasibility study. **Data & Knowledge Engineering**, v. 68, n. 10, p. 905–925, 2009.

SIMPERL, E. P. B.; TEMPICH, C. **Ontology engineering: a reality check**. OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". **Anais...2006**

SONG, Z.; CÁRDENAS, A. A.; MASUOKA, R. Semantic middleware for the internet of things. **2010 Internet of Things, IoT 2010**, 2010.

STOCKER, M. et al. **Emrooz: A scalable database for SSN observations**. CEUR Workshop Proceedings. **Anais...**Bethlehem, PA, United states: 2015

SUÁREZ-FIGUEROA, M. C.; GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M. The NeOn Methodology for Ontology Engineering. In: SUÁREZ-FIGUEROA, M. C. et al. (Eds.). . **Ontology Engineering in a Networked World**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 9–34.

TEIXEIRA, S. et al. Modeling and automatic code generation for wireless sensor network applications using model-driven or business process approaches: A systematic mapping study. **Journal of Systems and Software**, v. 132, p. 50–71, 2017.

THULUVA, A. S.; ANICIC, D.; RUDOLPH, S. **IoT Semantic Interoperability with Device Description Shapes**. European Semantic Web Conference. **Anais...**2018

WANG, F. et al. A Survey from the Perspective of Evolutionary Process in the Internet of Things. **International Journal of Distributed Sensor Networks**, v. 2015, p. 1–9, 2015.

WANG, X.; ZHANG, X.; LI, M. A survey on semantic sensor web: Sensor ontology, mapping and query. **International Journal of u- and e- Service, Science and Technology**, v. 8, n. 10, p. 325–342, 2015.

WEGNER, P. **Interoperability**. ACM computing surveys. **Anais...**1996

WEISER, M. The Computer for the 21 st Century. **Scientific american**, v. 265, n. 3, p. 94–105, 1991.

WORKING, W. O. **OWL 2 Web Ontology Language**, 2009.

ZAMBORLINI, V.; GONÇALVES, B.; GUIZZARDI, G. **Codification and application of a well-founded heart-ECG ontology**. Proceedings of the 3rd Workshop on Ontologies and Metamodels in Software and Data Engineering, Campinas, Brazil. **Anais...**2008

ZHANG, X.; ZHAO, Y.; LIU, W. Transforming sensor data to rdf based on ssn ontology.
Advanced Science and Technology Letters, v. 81, p. 95–98, 2015.