

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

IMPLEMENTAÇÃO DE UM MOTOR SEMÂNTICO PARA AMBIENTES IOT

ALISSON SOLITTO DA SILVA

ORIENTADOR: PROF. ME. FABIO PIOLA NAVARRO

Marília - SP
Novembro/2017

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

IMPLEMENTAÇÃO DE UM MOTOR SEMÂNTICO PARA AMBIENTES IOT

ALISSON SOLITTO DA SILVA

Monografia apresentada ao Centro Universitário Eurípides de Marília como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Prof. Me. Fabio Piola Navarro

Marília - SP
Novembro/2017

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ALISSON SOLITTO DA SILVA

**IMPLEMENTAÇÃO DE UM MOTOR SEMÂNTICO PARA
AMBIENTES IOT**

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para a obtenção do Título de Bacharel em Ciência da Computação.

Prof. Me. Fabio Piola Navarro
Orientador

Professor
Convidado 1

Professor
Convidado 2

Marília - SP
Novembro/2017



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
MANTIDO PELA FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Alisson Sólitto da Silva

Implementação de um motor semântico para ambientes IoT

Banca examinadora da monografia apresentada ao Curso de Bacharelado em
Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de
Bacharel em Ciência da Computação.

Nota: 100 (Dez)

Orientador: Fabio Piola Navarro Fabio P. Navarro

1º.Examinador:Fábio Dacêncio Pereira Fabio

2º.Examinador:Jorge Luiz Barbosa Maciel Junior Jorge L. B. Maciel Junior

Marília, 30 de novembro de 2017.

*Dedico este trabalho aos meus pais **Adolfo** e **Cleide**, que se esforçaram muito para que eu sempre estivesse em busca do conhecimento. A conclusão deste trabalho e de mais uma etapa acadêmica é graças a todo o esforço e dedicação dos meus pais com a minha educação desde cedo.*

AGRADECIMENTOS

Agradeço primeiramente à **Deus**, por sempre ter iluminado o meu caminho, e pela saúde para viver e seguir com os meus objetivos.

À toda minha família, em especial meu pai, **Adolfo**, minha mãe, **Cleide** e ao meu irmão **Christian**, por todo apoio e carinho, além dos meus avós.

A minha namorada, **Maira**, presente praticamente desde o início da minha graduação, sempre esteve comigo nos momentos mais difíceis. Sempre muito companheira e amorosa.

Ao meu orientador, Prof. **Fabio Navarro**, que me ajudou e incentivou no desenvolvimento do presente trabalho.

A todos os professores do curso pela dedicação e comprometimento durante esse longo período.

Obrigado à todos!

Estamos aqui para fazer alguma diferença no universo, se não, porque estar aqui?
Steve Jobs

RESUMO

Com a ascensão da Internet of Things muitos dados serão gerados por estes ambientes, porém estes dados não possuem nenhum nível semântico para que possam ser compreensíveis não somente por humanos, mas também por máquinas através de agentes inteligentes. Para a estruturação formal e semântica desses dados foram utilizadas técnicas do paradigma da Web Semântica e uma ontologia pré-definida para ambientes IoT. O desenvolvimento de sistemas semânticos exige um conhecimento apurado de suas tecnologias e ontologias, assim, foi implementando um motor semântico que possui uma API para a manipulação dos dados na ontologia, as informações geradas pelos sensores podem ser enviadas utilizando a API do motor semântico, possibilitando a interoperabilidade entre diferentes tecnologias e a facilitando o uso para os desenvolvedores. Com a criação da API para a manipulação da ontologia é possível a criação de aplicações Web e até mesmo aplicativos para smartphones, abrindo um leque enorme de possibilidades de sistemas que utilizam semântica para ambientes IoT.

Palavras-chave: Internet of Things, Web Semântica, API semântica, motor semântico, ontologia, sensores.

ABSTRACT

With the rise of the Internet of Things many data will be generated by these environments, but these data have no semantic level so that they can be understood not only by humans, but also by machines through intelligent agents. For the formal and semantic structuring of these data, Semantic Web paradigm techniques and a predefined ontology for IoT environments were used. The development of semantic systems requires a thorough knowledge of its technologies and ontologies, thus, it has been implementing a semantic engine that has an API for the manipulation of the data in the ontology, the information generated by the sensors can be sent using the semantic engine API, making possible interoperability between different technologies and making it easier for developers to use. With the creation of the API for the manipulation of the ontology it is possible to create Web applications and even applications for smartphones, opening a huge range of possibilities of systems that use semantics for IoT environments.

Keywords: Internet of Things, Semantic Web, semantic API, semantic engine, ontology, sensors.

LISTA DE FIGURAS

Figura 2.1 – Paradigma Internet of Things	20
Figura 2.2 – Domínios de aplicações IoT	22
Figura 2.3 – Modelo de arquitetura vertical Internet of Things	23
Figura 2.4 – Hierarquia dos níveis de conhecimento	26
Figura 2.5 – Ontologia de sistema de geladeira inteligente	29
Figura 2.6 – Representação de uma requisição HTTP	31
Figura 2.7 – Arquitetura funcional do Servidor WoT	35
Figura 2.8 – Servidor WoT no dispositivo	36
Figura 2.9 – Cliente WoT no Smartphone	37
Figura 2.10 – Servidor WoT no Smart Home Hub	37
Figura 2.11 – Servidor WoT no servidor de nuvem	38
Figura 2.12 – Estrutura da Web Semântica	39
Figura 2.13 – Classificação da ontologia	42
Figura 2.14 – Ontologia Semantic Sensor Network	43
Figura 3.1 – Ontologia utilizada no trabalho	48
Figura 3.2 – Arquitetura motor semântico	50
Figura 3.3 – Tela principal da interface	56
Figura 3.4 – Tela para inclusão da instância na ontologia	57
Figura 3.5 – Tela para a criação de um sensor na ontologia	58
Figura 3.6 – Tela para incluir Measurement Capability	60
Figura 3.7 – Instâncias da ontologia criada na interface	61
Figura 3.8 – Tela de configuração dos dados para simulação	62
Figura 3.9 – Tela de simulação do envio de dados	63
Figura 3.10 – Instâncias de observação	63
Figura 3.11 – Visualização da ontologia	64
Figura 3.12 – Interface administração Fuseki	64

LISTA DE ABREVIATURAS E SIGLAS

API - *Application Programming Interface*
HTTP - *Hypertext Transfer Protocol*
IOT - *Internet of Things*
ITU - *International Telecommunication Union*
M2M - *Machine-to-Machine*
NIC - *National Intelligence Council*
OWL - *Web Ontology Language*
REST - *Representational State Transfer*
RDF - *Resource Description Framework*
RFID - *Radio-Frequency IDentification*
SQL - *Structured Query Language*
SSN - *Semantic Sensor Network*
UML - *Unified Modeling Language*
URI - *Uniform Resource Identifier*
URN - *Uniform Resource Name*
WOT - *Web of Things*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contexto	13
1.2	Motivação e Objetivos	14
1.3	Metodologia de Desenvolvimento do Trabalho	15
1.4	Organização do Trabalho	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Internet of Things	17
2.1.1	Perspectiva histórica	20
2.1.2	Motivação para evolução da IoT	21
2.1.3	Arquitetura	23
2.1.4	Fundamentos de desenvolvimento IoT	24
2.1.5	Contexto	25
2.1.6	Modelagem	26
2.1.6.1	Modelagem baseada em ontologias	28
2.2	Web of Things	30
2.2.1	Uniform Resource Identifiers	30
2.2.2	Hypertext Transfer Protocol	31
2.2.2.1	Mensagem HTTP	31
2.2.3	Métodos HTTP	32
2.2.3.1	GET	32
2.2.3.2	POST	32
2.2.3.3	PUT	32
2.2.3.4	DELETE	33
2.2.4	Representational State Transfer	33
2.2.5	Arquitetura WoT	34
2.2.5.1	Servidor Funcional WoT	34
2.2.5.2	Cenários da arquitetura WoT	36
2.2.5.3	Servidor WoT no dispositivo	36
2.2.5.4	Cliente WoT no Smartphone	36
2.2.5.4.1	Servidor WoT no Smart Home Hub	37
2.2.5.4.2	Servidor WoT no servidor de nuvem	37
2.3	Web Semântica	38
2.3.0.1	Camadas da Web Semântica	39
2.3.1	Ontologia	40

2.3.1.1	Tipos de ontologia	41
2.3.2	Ontologia e IoT	42
2.3.2.1	Semantic Sensor Network	43
2.3.3	Desenvolvimento	43
2.3.3.1	Desenvolvimento em Triplas	44
2.3.3.2	Desenvolvimento Orientado a Objetos	44
	3 DESENVOLVIMENTO	46
3.1	Tecnologias	46
3.1.1	Apache Jena	46
3.1.2	Fuseki	47
3.1.3	Protégé	47
3.1.4	Tomcat	47
3.1.5	Angular 4	47
3.2	Ontologia SSN e GEO	48
3.3	Motor Semântico	49
3.3.1	Camadas da Arquitetura	50
3.3.1.1	Internet of Things	50
3.3.1.2	Web of Things	51
3.3.1.3	Internet	52
3.3.1.4	Inteligência	52
3.3.1.4.1	API Motor Semântico	52
3.3.1.4.2	GET	52
3.3.1.4.3	POST	53
3.3.1.4.4	DELETE	54
3.3.1.4.5	Motor Semântico	54
3.3.1.4.6	Banco Semântico	55
3.3.1.5	Aplicações	55
3.3.2	Aplicação Semantic Internet of Things	55
3.3.2.0.1	Sensing Device	57
3.3.2.0.2	Property	57
3.3.2.0.3	Feature of Interest	57
3.3.2.0.4	Sensor	58
3.3.2.0.5	Simulator	61
3.3.2.0.6	Ontology	64
3.3.2.0.7	Fuseki	64
	4 CONCLUSÃO	65
4.1	Contribuições e Limitações	65
4.1.1	Contribuições	65

4.1.2	Limitações	66
4.2	Lições aprendidas	66
4.3	Trabalhos Futuros	66
	REFERÊNCIAS	68

Capítulo 1

INTRODUÇÃO

1.1 Contexto

Com o constante avanço da tecnologia a quantidade de objetos conectados à Internet está em uma constante crescente. De acordo com um relatório da Gartner Inc. “prevê que 6,4 milhões de objetos estarão conectados e em uso em todo o mundo no ano de 2016, um aumento de trinta por cento em relação ao ano de 2015 e chegará a 20,8 bilhões até 2020” (GARTNER, 2015). Esse grande número de objetos conectados à Internet geram e disseminam uma miríade de informações que podem interagir com outros sistemas e proporcionar o poder de tomada de decisão.

Os objetos que fazem parte da IoT vão além de smartphones e dispositivos vestíveis, o constante crescimento e a gama de sensores acoplados a objetos do mundo real como lâmpadas, máquinas de café, televisões, portas e etc, formam uma rede que permite coletar e analisar informações para as pessoas ou sistemas interagirem em um ambiente.

Com a ascensão da Internet of Things surge a necessidade de consumir serviços e informações, tornando-os interoperáveis com outros sistemas e redes. Nesse cenário surgiu um novo paradigma de desenvolvimento conhecido Web of Things, este novo paradigma faz uma abstração virtual do mundo real (IoT), algo como uma camada de software acima dos objetos reais, onde pode-se adaptar, criar modelos e protocolos, pois esta habilita acesso aos recursos (sensores) através da Web, o que pode ainda propiciar acesso semântico e desta forma proporcionar uma maior interação entre dispositivos, sistemas e pessoas (CHRISTOPHE; VERDOT; TOUBIANA, 2011).

Nos cenários mais comuns de IoT ocorre uma disseminação de dados heterogênea, esses dados possuem um bom nível sintático. No entanto, quando se trata do sentido desses dados estes repositórios não estão aptos a armazenar informação

semântica e contextualizada.

Para que seja possível a agregação de semântica aos dados dos sistemas informacionais atuais, é possível utilizar o paradigma da Web Semântica que propõe uma estrutura para as páginas web onde seja possível a compreensão tanto por seres humanos quanto por máquinas, ou seja, as informações da Web estariam estruturadas de forma semântica dando significado as páginas. A solução para este problema é fornecida por um dos componentes básicos da Web Semântica, uma coleção de informações chamada de ontologia.

Na Ciência da Computação, a ontologia é um conjunto de informações e dados obtidos que pode ser definido como um conjunto de conceitos fundamentais e suas relações existentes no domínio em questão, permitindo a representação e o entendimento do mundo-alvo de maneira formal e compreensível por humanos e computadores (MIZOGUCHI, 2004).

A Web Semântica não é apenas uma extensão da Web atual, podemos utilizar seus conceitos e tecnologias para estruturar de maneira formal e semântica diferentes sistemas que necessitam de semântica em sua estrutura. Em ambientes de IoT utilizando a camada WoT para que seja realizada a abstração virtual de uma rede de sensores, torna-se possível a criação de modelos de domínio de conhecimento para diferentes cenários. O domínio de conhecimento é composto de ontologias, conjuntos de dados e regras usadas para anotar semanticamente dados da IoT. Com o domínio de conhecimento definido é possível aplicar o paradigma da Web Semântica, permitindo o armazenamento da informação semântica e contextualizada produzida pela rede de sensores IoT.

Ainda que a aplicação da Web Semântica resolva o grande problema de agregar sentido aos dados brutos gerados pelos sensores de IoT, o desenvolvimento de sistemas semânticos exige um conhecimento apurado sobre o paradigma da Web Semântica, ontologias e suas linguagens de representação. A ausência desse conhecimento da maior parte dos desenvolvedores de sistemas é em razão de não ser algo trivial e, portanto mais predominante no mundo acadêmico e científico, tornando complexo e dificultoso a aplicação das tecnologias de semântica em sistemas informacionais e em ambientes IoT.

1.2 Motivação e Objetivos

A IoT ou Internet of Things é uma grande revolução de modo a conectar objetos utilizados no nosso dia-a-dia, que podem ser controlados e acessados através da Internet. O objetivo final é criar um mundo melhor para os seres humanos, onde objetos ao nosso redor saibam o que gostamos, o que queremos, e o que precisamos,

sem a intervenção humana e instruções explícitas, para isso esses ambientes devem ser compreendidos tanto por seres humanos quanto por máquinas.

Com a utilização do paradigma da Web Semântica é possível introduzir um nível semântico aos dados coletados pelos sensores, tornando-os compreensíveis tanto por humanos quanto por máquinas. Porém, o desenvolvimento de sistemas semânticos exige um conhecimento apurado da ontologia e de suas linguagens de representação, dificultando o desenvolvimento de sistemas semânticos e interoperáveis.

O objetivo geral do presente trabalho é propor um motor semântico que agregue sentido aos dados brutos gerados pelos sensores nas redes IoT, além de abstrair todo o conhecimento necessário para a criação de aplicações semânticas, proporcionando a interoperabilidade entre diferentes aplicações.

Para cumprimento dos objetivos gerais, os seguintes objetivos específicos foram realizados:

- Definir uma estrutura ontológica de representação do domínio de ambientes IoT;
- Criar um motor semântico utilizando o paradigma da Web Semântica e suas tecnologias;
- Criar uma API para comunicação do motor semântico com os dispositivos IoT e aplicações em diferentes tecnologias;
- Criar lógica para persistência da ontologia utilizando um banco semântico;
- Criar um protótipo de aplicação para validação do motor semântico proposto.

1.3 Metodologia de Desenvolvimento do Trabalho

A abordagem metodológica desta pesquisa, considerando os objetivos, está embasado na integração das áreas de Internet of Things e Web Semântica, relacionadas com a semântica da informação nos novos ambientes de sensores. Caracteriza-se por ser uma pesquisa de análise experimental e qualitativa descritiva do tema.

Para que os objetivos possam ser alcançados, os procedimentos metodológicos utilizados baseiam-se no estudo pautado em pesquisa bibliográfica e aplicada dividido da seguinte forma:

- Estudo pautado e levantamento bibliográfica sobre temas como: Internet of Things, Web Semântica, Web of Things, ontologias, métodos de comunicação HTTP e tecnologias utilizadas no desenvolvimento do projeto.
- Estudo da relação dos objetos dentro da ontologia utilizada.
- Tecnologias da Web Semântica para persistência da ontologia em disco.

- Construção do motor semântico baseado no paradigma da Web Semântica e comunicação HTTP, e do estudo de técnicas para a manipulação da ontologia.
- Implementação da interface para comunicação com o motor semântico, é realizado a análise de como é feita a comunicação com o motor semântico, e posteriormente, como as informações são persistidas na ontologia.

1.4 Organização do Trabalho

No Capítulo 2 é apresentado um estudo bibliográfico pautado sobre Internet of Things, suas perspectivas e quais técnicas utilizadas para a modelagem de contextos nesses ambientes, logo após é apresentado o paradigma da Web of Things e suas possíveis arquiteturas, e finalmente, são apresentados os conceitos e tecnologias da Web Semântica.

O Capítulo 3 expõe a criação do motor semântico utilizando todo o conhecimento adquirido e pesquisado no Capítulo 2. O Capítulo 3 inicia-se com uma breve descrição das tecnologias utilizadas, expõe a ontologia utilizada, a arquitetura proposta e por fim a aplicação desenvolvida para a utilização do motor semântico.

No Capítulo 4 são apresentados as contribuições, limitações e os resultados da pesquisa, assim como as lições aprendidas com o desenvolvimento do presente trabalho e as pesquisas futuras.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

A IoT ou Internet of Things é uma grande revolução a fim de conectar objetos utilizados no nosso dia-a-dia à Internet. Neste cenário surge uma grande rede não apenas de documentos, mas de objetos físicos que podem ser controlados e acessados através da Internet.

Com o propósito dos objetos que fazem parte da IoT consumirem serviços e informações surge um novo paradigma de desenvolvimento conhecido como WoT ou Web of Things. A WoT é utilizada para realizar a abstração virtual de uma rede IoT, tornando possível a criação de modelos de domínio para diferentes cenários.

Esta seção tem por objetivo abordar os conceitos e paradigmas que compõem a Internet of Things, Web of Things e Web Semântica, seus protocolos e métodos de comunicação, assim como os padrões de arquitetura, o processo de modelagem de contexto e, finalmente, um panorama sobre a inteligência de uma rede utilizando a modelagem baseada em ontologias.

2.1 Internet of Things

Durante a última década, a IoT ou Internet of Things ganhou uma atenção significativa na academia, bem como na indústria. As principais razões por trás deste interesse são as capacidades que a IoT irá oferecer. Este novo paradigma promete criar um mundo onde todos os objetos (também chamados objetos inteligentes) ao nosso redor estarão conectados à Internet e se comunicarão uns com os outros com a mínima intervenção humana. O objetivo final é criar um mundo melhor para os seres humanos, onde objetos ao nosso redor saibam o que gostamos, o que queremos, e o que precisamos, sem a intervenção humana e instruções explícitas (PERERA et al., 2014).

Este termo foi inicialmente cunhado por Kevin Ashton em uma apresentação

em 1998. Segundo (PERERA et al., 2014), Kevin mencionou, "A Internet of Things tem o potencial de mudar o mundo, assim como a Internet. Talvez ainda mais". Posteriormente, a IoT foi formalmente introduzida pela ITU (International Telecommunication Union) em 2005 e passou a receber grande importância pelas organizações.

Segundo o (ITU, 2012), a IoT é uma infraestrutura global de informação para a sociedade, permitindo serviços avançados interligando coisas (físicas e virtuais) existentes na evolução interoperável das tecnologias de informação e comunicação.

Existem múltiplas definições para o conceito de IoT dentro da comunidade de pesquisa, muitos pesquisadores durante décadas contribuem com esforços significativos para tornar a IoT uma realidade para o mundo real. Muitos pontos de vista são apresentados por inúmeros pesquisadores, (TAN, 2010, tradução nossa) explica que: "As coisas têm identidades e personalidades virtuais, esses objetos operam usando interfaces inteligentes para se conectarem e se comunicarem dentro de contextos sociais, ambientais e de usuários". As coisas no contexto da IoT são todos os objetos conectados à Internet, desde smartphones até uma torradeira.

Outro ponto de vista apresentado por (COMMISSION, 2008) descreve a origem semântica da expressão IoT, (COMMISSION, 2008) completa que a IoT é composta por dois termos distintos: Internet e Thing, onde a Internet pode ser definida como a rede mundial de redes de computadores interligadas, com base em protocolos de comunicação padrão, enquanto Thing é um objeto não identificável, podendo ser qualquer coisa. Portanto, semanticamente, a Internet das Coisas significa uma rede mundial de objetos interligados, baseados em protocolos de comunicação padrão. Para complementar essa visão (VERMESAN et al., 2011, tradução nossa) diz ainda que "A Internet das coisas permite que pessoas e coisas sejam conectadas a qualquer momento, em qualquer lugar, com qualquer coisa e qualquer pessoa".

Uma última definição publicada mais recentemente por (GUINARD; TRIFA, 2016, tradução nossa) explica: "A Internet das Coisas é um sistema de objetos físicos que pode ser descoberto, monitorado, controlado com dispositivos eletrônicos que se comunicam através de várias interfaces de rede e eventualmente podem ser conectado à Internet".

Com tantas definições existe uma real dificuldade em entender o que realmente significa a IoT, quais ideias básicas estão por trás desse conceito e quais as implicações sociais, econômicas e técnicas. A razão da aparente imprecisão em torno deste termo é uma consequência do nome "Internet das Coisas", que sintaticamente é composto de dois termos. O primeiro empurra para uma visão orientada para a rede de IoT, enquanto o segundo move o foco em "objetos" genéricos a serem integrados em um quadro comum (ATZORI; IERA; MORABITO, 2010).

Para complementar esta visão geral, três pontos destacam a complexidade da IoT. (COMMUNITIES, 2009)

- (a) não deverá ser vista como uma mera extensão da Internet, mas sim uma extensão de novos sistemas independentes que operam com suas próprias infra-estruturas. No entanto ainda dependem em parte das infra-estruturas da Internet.
- (b) as novas redes formadas pela IoT serão implementadas em conjunto com novos serviços
- (c) englobará diferentes modos de comunicação: comunicação de pessoa a pessoa e comunicações de coisa a coisa, incluindo a comunicação Máquina a Máquina (M2M).

Múltiplas definições, conceitos e tecnologias giram em torno deste paradigma na comunidade de pesquisadores, a Internet of Things pode ser definida como uma convergência de três visões (AGGARWAL; ASHISH; SHETH, 2013):

- Internet orientada: nesta visão a Internet atua como um Middleware¹ entre o usuário e as coisas inteligentes. O Middleware permitirá a abstração dinâmica do mundo real/físico entre o espaço digital/virtual combinando características da rede de sensores e protocolos utilizados na Web para permitir que objetos inteligentes sejam ligados à Internet.
- Coisas orientadas: são as coisas inteligentes, que representam sensores e atuadores responsáveis pelos estímulos do ambiente.
- Semântica orientada: onde o ponto principal é o conhecimento obtido através representação, armazenamento, organização e uso das informações. A semântica aborda as questões de representação do conhecimento que são trocadas pelos objetos inteligentes.

¹ Middleware é uma camada oculta de software que permite a comunicação e o transporte de informações entre aplicações de diferentes protocolos

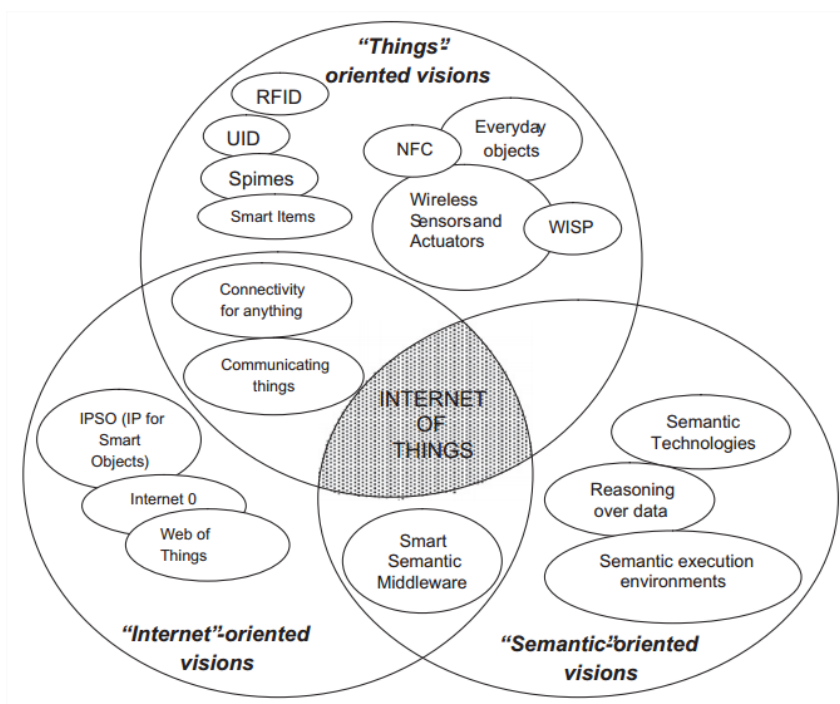


Figura 2.1 – Paradigma Internet of Things - resultado da convergência das diferentes visões mencionadas acima, destacando os principais conceitos, tecnologias e padrões de cada visão (ATZORI; IERA; MORABITO, 2010).

2.1.1 Perspectiva histórica

Este novo paradigma está ganhando espaço rapidamente, além de smartphones e dispositivos vestíveis, o constante crescimento e a gama de sensores acoplados a objetos do mundo real formam uma rede que permite coletar e analisar informações para as pessoas ou sistemas interagirem em um contexto. A ideia básica deste conceito é a presença generalizada à nossa volta de uma variedade de coisas ou objetos, como tags de identificação por radiofrequência (RFID), sensores, atuadores, telefones celulares, etc. Estes objetos são capazes de interagir e cooperar uns com os outros afim de alcançar objetivos comuns. Inquestionavelmente, a principal força da IoT é o alto impacto que terá em vários aspectos da vida cotidiana de seus usuários (ATZORI; IERA; MORABITO, 2010). Esses objetos possuirão seus próprios endereços da Internet, e serão incorporados em sistemas complexos que poderão ser usados para obter informações do ambiente (por exemplo, produtos alimentares que estão em falta na geladeira, temperatura do ambiente). (COMMUNITIES, 2009).

Muitas pesquisas, estatísticas, previsões e afirmações feitas por grandes órgãos e importantes fontes no meio apresentam a magnitude dessa nova tecnologia e a justificativa do foco da pesquisa nessa área. De acordo com um relatório da (GARTNER, 2015) “prevê que 6,4 milhões de objetos estarão conectados e em uso em todo o mundo no ano de 2016, um aumento de trinta por cento em relação ao ano de 2015

e chegará a 20,8 bilhões até 2020”. Partindo dessas considerações, não deve ser surpreendente que a IoT seja incluído pela NIC (Conselho Nacional de Inteligência) dos Estados Unidos da América na lista das seis tecnologias com maior impacto sobre o poder nacional (ATZORI; IERA; MORABITO, 2010).

Esse grande número de objetos conectados à Internet disseminarão uma miríade de informações que podem interagir com outros sistemas e impactar diretamente no cotidiano da sociedade, melhorando a qualidade de vida dos cidadãos e proporcionando novos empregos e oportunidades de negócio e crescimento para a indústria, muitos acreditam que a IoT será a Indústria 4.0, ou seja, a quarta revolução industrial – a primeira foi a manufatura e a mecanização, a segunda foi a produção em massa e a terceira foi a revolução digital, a próxima será a informatização da maquinaria e a automatização com recurso à robótica, a análise inteligente de dados melhorando a eficiência, a rentabilidade e a segurança.

De acordo com o (CASOTTI, 2016), muitos analistas consideram a IoT como um dos principais vetores da próxima revolução industrial, como foram a máquina a vapor, a eletricidade e a eletrônica. Essas grandes inovações trouxeram grandes impactos para a sociedade, na revolução atual, a quarta da história, as diferenças entre homem e máquinas são moderadas e o valor passa a ser a informação.

2.1.2 Motivação para evolução da IoT

A IoT é uma revolução silenciosa, no entanto as possibilidades que surgem são infinitas, passando por todos os estágios do ciclo de vida do produto até ocasiões de uso em indústrias verticais específicas. (ASIN; GASCON, 2012) listaram 54 domínios de aplicação sob doze categorias, algumas delas são: cidades inteligentes, ambientes inteligentes, medição inteligente, segurança e emergências, varejo, logística, controle industrial, agricultura inteligente, pecuária inteligente, automação doméstica, entre outros.

Para (ATZORI; IERA; MORABITO, 2010) os domínios de aplicações e cenários mais relevantes são apresentados na Figura 2.2:

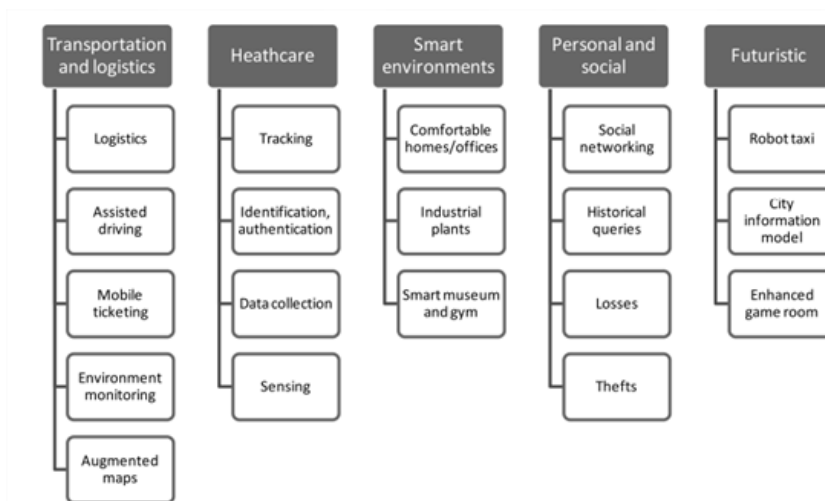


Figura 2.2 – Domínios de aplicações IoT com maior relevância (ATZORI; IERA; MORABITO, 2010).

As iniciativas de adoção de IoT tem diferentes perspectivas de valor e inúmeros benefícios que são esperados a partir da transformação digital. Para o (CASOTTI, 2016) alguns desses benefícios são: redução dos custos operacionais, aumento da produtividade, expansão para novos mercados e desenvolvimento de novos produtos e serviços (ou evolução dos existentes).

Segundo (CASOTTI, 2016), as diferentes perspectivas de valores com a adoção da IoT são:

- (a) Visibilidade (Conectar e Monitorar): Os objetos conectados e autônomos geram dados específicos sobre processos que podem ser monitorados com base nos alertas.
- (b) Agilidade (Controlar e Atuar): Os dados recuperados de forma automatizada permitem o gerenciamento e acompanhamento remoto do processo.
- (c) Inteligência (Prever e Adaptar): A fusão dos dados em tempo real de diferentes processos subsidiam as análises preditivas de performance e ações preventivas.
- (d) Inovação (Transformar e Explorar): Os dados coletados são utilizados para a criação e inovação de novos modelos de negócio, produtos e serviços.

Um exemplo que caracteriza o universo de possibilidades geradas pela adoção da IoT e engloba todos os valores gerados por ela a sua aplicação para monitoramento remoto. Imagine um pacote ou produto equipado com um sensor, transmissor e um identificador (RFID), deste modo a empresa não só poderá rastrear o item como também obter dados em tempo real sobre o desempenho e o modo como está sendo usado. Em contra partida, os clientes podem rastrear continuamente em tempo real

o andamento da entrega do pacote que estão aguardando e compartilhar, na mesma velocidade, suas experiências com produtos e serviços (CASOTTI, 2016).

2.1.3 Arquitetura

Segundo (ZHENG et al., 2011), há duas visões quanto aos modelos de arquitetura, uma é a visão vertical que concentra-se na implementação técnica, a segunda é a visão horizontal que prioriza a implementação e gerenciamento. Na visão vertical, a IoT é caracterizada pela transmissão confiável e processamento inteligente.

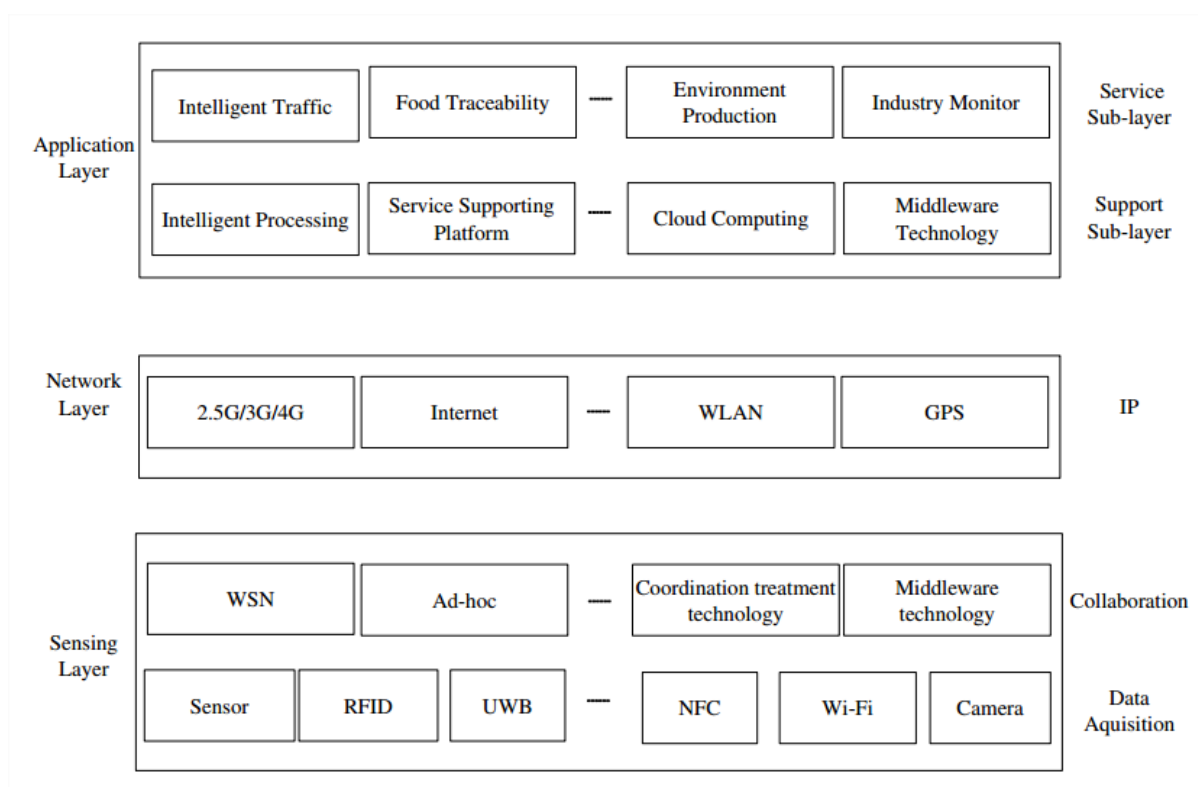


Figura 2.3 – Modelo de arquitetura vertical Internet of Things (ZHENG et al., 2011).

- Camada sensores: é responsável pela detecção e o recolhimento de dados em tempo real através de vários sensores. Os dados são provenientes da detecção do estado das coisas no mundo físico como a temperatura, os dados de multimídia adquiridos por sensores como, RFID, câmeras e GPS.
- Camada de rede: essa camada é responsável pela transmissão de dados adquiridos pela camada de sensores para a camada de aplicação, utilizando tecnologias de computação distribuída e computação em nuvem.
- Camada de Aplicação: a ultima camada executa o processamento e a análise de dados para fins de controles inteligentes.

2.1.4 Fundamentos de desenvolvimento IoT

Sabe-se que há muitos desafios a serem enfrentados no processo de desenvolvimento e implantação da IoT, (PERERA et al., 2014) identificou sete características com base em pesquisas anteriores que devem ser consideradas em todo o processo de desenvolvimento de soluções IoT, são elas:

- **Inteligência:** significa a aplicação do conhecimento, os dados brutos são coletados e então é aplicado um raciocínio transformando os dados em conhecimento de alto nível. Essa transformação pode ser feita através da modelagem e raciocínio do contexto. O contexto pode ser usado para fundir os dados dos sensores para inferir novos conhecimentos. Uma vez que temos conhecimento, ele pode ser aplicado para a interação, comunicação e recuperação da informação mais inteligente.
- **Arquitetura:** a IoT deve apresentar uma arquitetura híbrida compreendendo várias arquiteturas diferentes, essa arquitetura híbrida englobaria todos os tipos de sensores separando-os por evento e tempo. Sensores que produzem dados quando ocorre algum evento (por exemplo, sensor de porta), os restantes produzem dados continuamente com base em intervalos de tempo (por exemplo, sensor de temperatura)
- **Sistema:** o grande número de objetos que interagem de forma autônoma em uma rede IoT, as interações podem diferir significativamente dependendo dos recursos de cada objeto. Alguns podem ter muito poucos recursos, como armazenar informações muito limitadas e não fazer nenhum processamento. No entanto, alguns objetos podem ter recursos maiores de memória, processamento e raciocínio, o que os torna mais inteligentes.
- **Considerações de tamanho:** bilhões de dispositivos estarão conectados à Internet e a IoT precisa facilitar a interação entre esses objetos, semelhante ao número de objetos as interações entre eles aumentaram exponencialmente.
- **Considerações de tempo:** O processamento de dados em tempo real é essencial em um ambiente com bilhões de eventos paralelos e simultâneos.
- **Considerações de espaço:** com o crescimento exponencial o rastreamento e a localização geográfica dos objetos na rede torna-se um requisito fundamental, pois as interações podem diferir significativamente dependendo dos recursos dos objetos.
- **Tudo como serviço (Everything as a service):** com a popularidade da computação em nuvem, este modelo é altamente eficiente, escalável e fácil de usar. A IoT exige uma quantidade significativa de infra estrutura a ser posta em prática, a fim de tornar sua visão uma realidade, onde este modelo serviria muito bem toda a rede.

Para este projeto será dado destaque a característica de inteligência, pois este conceito é primordial para o desenvolvimento de ambientes inteligentes e interoperáveis. A aplicação do conhecimento no contexto em que encontra-se a rede IoT, possibilita não somente a interação e a comunicação mais inteligente, mas também a recuperação da informação de ambientes que possuem uma grande quantidade de dados sem nenhum sentido semântico.

2.1.5 Contexto

Segundo (ABOWD et al., 1999, tradução nossa), "contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e um aplicativo, incluindo o usuário e as próprias aplicações".

Independente do contexto em qual a rede de sensores encontra, são gerados muitos dados, porém muito pouco transforma-se em informação relevante para o usuário, ou seja, em um contexto são gerados dados brutos e informações de contexto.

- Dados brutos: são dados que encontram-se na forma original em que são coletados, podem ou não estar organizados, ou seja, é um dado coletado que não sofreu qualquer agregação de outros dados de forma a constituir uma informação (CORREIA, 2009).
- Informação de contexto: é o processamento do dado por meio de convenções usadas para sua representação que colaboram para a interpretação ou significado dos dados (CORREIA, 2009).

Um exemplo entre dados brutos e informações de contexto pode ser observado em uma rede de sensores que produzem informações de temperatura no contexto agropecuário, as leituras produzidas por esses sensores são considerados dados brutos, portanto essa rede gera apenas dados. Se a partir destes dados coletados pela rede de sensores agregarmos dados de latitude e longitude que representa a localização geográfica do sensor geramos informações de contexto relevantes para o usuário.

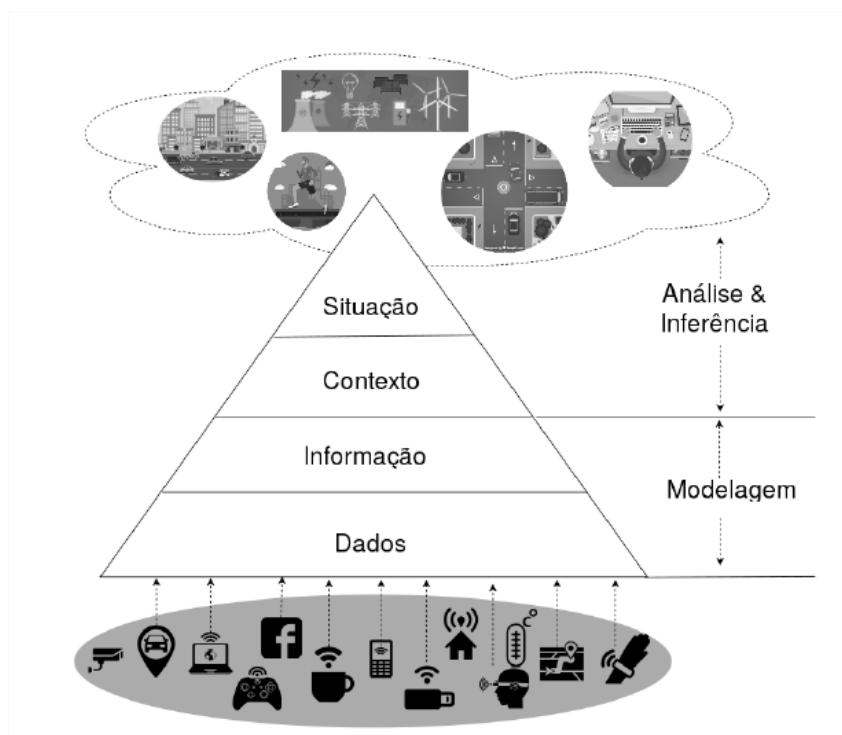


Figura 2.4 – Hierarquia dos níveis de conhecimento a partir de dados brutos de sensores (SANTOS et al., 2016).

2.1.6 Modelagem

Segundo (BATALLA et al., 2016), a modelagem de contexto é organizada em duas etapas. Na primeira etapa são definidos os atributos, características e relações das informações de contexto já com o contexto previamente definido. Na segunda etapa é validado o resultado da primeira e as informações de contexto são adicionadas e mescladas ao repositório existente, gerando assim novas informações de contexto disponíveis para uso.

As técnicas de modelagem de contexto mais populares pesquisadas apontam as forças e as fraquezas de cada técnica utilizada. São listadas seis técnicas anunciadas abaixo: (PERERA et al., 2014; BATALLA et al., 2016).

- **Key-Value Modelling (Modelagem Chave-Valor):** A técnica chave-valor modela as informações de contexto como pares, podendo ser utilizada em diferentes formatos de representação, como arquivos de texto e arquivos binários. Essa é uma forma muito simples de representação do contexto, é uma forma fácil de gerenciar quando as quantidades de dados são menores. Porém essa técnica não é escalável e não é adequada para armazenamento de estruturas complexas, já que não é possível representação relações hierárquicas usando pares chave-valor, além disso não permite o anexo de meta informação, que é essencial para

melhorar as informações de contexto.

- Markup Schema Modelling (Modelagem de esquema de marcação): Esta técnica modela as informações de contexto utilizando tags, é uma melhoria da técnica de modelagem valor-chave. A maior vantagem é a recuperação e a validação dos dados que é suportada através das definições do esquema. Uma tecnologia muito utilizada em aplicativos de diferentes domínios é a tecnologia XML. Além de suas vantagens e melhorias com relação a técnica anterior, está ainda carece de capacidades expressiva de raciocínio.
- Graphical Modelling (Modelagem Gráfica): Esta técnica modela o contexto com relacionamentos entre suas entidades, alguns exemplos dessa técnica são as modelagens Unified Modeling Language (UML) e Object Modeling (ORM). Essa técnica expressa maior riqueza de modelo de contexto em relação as anteriores, além disso seu uso pode ser considerado mais fácil, já que os desenvolvedores estão mais familiarizados com tecnologias de banco de dados SQL ou noSQL, os bancos de dados possibilitam o armazenamento de grande quantidade de dados e fornecem uma maneira simples de recuperação dos dados. Em contrapartida não existe uma padronização nas modelagens para o contexto, um mesmo contexto pode ser modelado de inúmeras formas, dificultando a interoperabilidade entre os sistemas.
- Object Based Modelling (Modelagem baseada em objetos ou orientada a objetos): A modelagem orientada a objeto tem como propósito modelar dados usando heranças e relacionamento de classes, sua estrutura possibilita o encapsulamento e a reutilização. Esse tipo de modelagem pode ser facilmente integrada em diversos sistemas com reconhecimento de contexto. No entanto, esta modelagem não fornece recursos de raciocínio inerentes e sua validação é difícil devido à falta de padrões e especificações
- Logic Based Modelling (Modelagem baseada em lógica): Essa técnica fornece muito mais riqueza e expressividade para representar informações sobre o contexto em comparação com os outros modelos apresentados. Nesse modelo são usados fatos, expressões e regras permitindo que novas informações de contexto de alto nível sejam geradas utilizando um contexto de baixo nível. Porém, a falta de padronização reduz a reutilização e aplicabilidade em outros sistemas.
- Ontology Based Modelling (Modelagem Baseada em Ontologia): O contexto é modelado utilizando tecnologias semânticas (RDF, RDFS, OWL) aumentando a capacidade de raciocínio do contexto. No entanto, a recuperação das informações de contexto pode se tornar computacionalmente mais intensiva de acordo com a quantidade de dados produzida.

2.1.6.1 Modelagem baseada em ontologias

Nos cenários mais comuns de IoT ocorre uma disseminação de dados heterogênea, esses dados possuem um bom nível sintático. No entanto, quando se trata do sentido desses dados estes repositórios não estão aptos a retornar informação semântica e contextualizada, dificultando a recuperação da informação e a tomada de decisão de forma inteligente. Neste cenário a melhor técnica de modelagem para o repositório é o uso de ontologias aumentando a capacidade semântica dos dados gerados pela rede de sensores, tornando-as redes inteligentes.

Segundo (STUDER; BENJAMINS; FENSEL, 1998), uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada. (STUDER; BENJAMINS; FENSEL, 1998) ainda explica, uma "conceitualização" é o modelo abstrato da representação de conceitos e relações de um domínio. "Explícito" significa que as restrições e conceitos são explicitamente definidos. "Formal" refere-se que a ontologia deve ser legível não apenas por humanos, mas compreendido também por máquinas. E por fim, "compartilhado" em que o conhecimento capturado pela ontologia não é privado, mas aceito por um grupo.

A maioria das abordagens de modelagem de contexto são baseadas em um cenário especial, por essa razão os modelos de informação de contexto são limitados não permitindo a integração do conhecimento em diferentes domínios a aplicações. Devido a necessidade de interoperabilidade entre diferentes ambientes e aplicações a modelagem de contexto baseada em ontologia é muito apropriada para o gerenciamento do contexto (TOPCU, 2011).

As razões mais comuns para o uso da modelagem baseada em ontologia são o compartilhamento e a compreensão comum da estrutura da informação, a reutilização de conhecimento de domínio e a inferência de conhecimento de alto nível no ambiente. Além disso as ontologias desacoplam o conhecimento dos código de aplicação e programa e permitem a integração de conhecimento em diferentes domínios em aplicações quando necessário (PERERA et al., 2014).

Existem várias abordagens e exemplos de uso de ontologias em ambientes de IoT, um possível cenário para enfatizar esta modelagem de contexto é um sistema de geladeira inteligente em um ambiente doméstico.

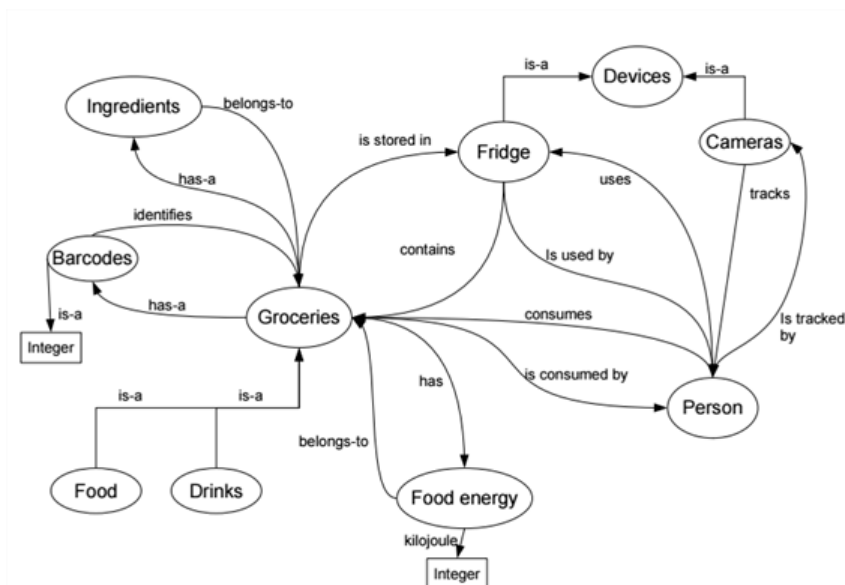


Figura 2.5 – Ontologia de sistema de geladeira inteligente (TOPCU, 2011).

A Figura 2.5 mostra uma notação gráfica simples de uma possível ontologia de um sistema de geladeira inteligente. As elipses definem as classes da ontologia, essas são as entidades que podem ser coisas, indivíduos ou abstrações do mundo real e cada seta define a relação entre essas entidades no domínio proposto. Por fim, os retângulos mostram os tipos de dados que são aceitos pelas entidades.

Agora imaginamos que a pessoa relacionada a entidade "Person" na ontologia chama-se João que usa uma geladeira e possui uma relação com os mantimentos que são armazenados. São definidas mais relações, como a classe "Person" consome este mantimento ou usa a geladeira, além disso os mantimentos podem ser identificados com um código de barras e possuem valor calórico. Outra fonte de dados importante é o uso da geladeira ou câmeras que podem acompanhar e analisar João quando ele está comendo. Todo esse sistema obtém dados de provedores diferentes, sendo assim o sistema deve configurar um repositório que armazenará todos os dados coletados e analisar os hábitos de João, após o período de coleta de informações o sistema poderá dar alguns conselhos, caso João esteja fazendo uma refeição com um alto nível calórico o sistema poderia dar-lhe algumas dicas para comer uma salada ou para cozinhar algo mais saudável (TOPCU, 2011).

Conforme mencionado anteriormente, os modelos de contexto baseados em ontologia oferecem vantagens em termos de expressividade e interoperabilidade. No exemplo da Figura 2.5, o modelo de ontologia atua como um Middleware oferecendo interoperabilidade entre as diferentes informações de dados (códigos de barras, câmeras e sensores dentro da geladeira).

2.2 Web of Things

Nos últimos anos com a ascensão da Internet of Things e o enorme progresso no campo de dispositivos e sistemas embarcados surge a necessidade de consumir serviços e informações, tornando-os interoperáveis com outros sistemas e redes.

Infelizmente, a construção de um ecossistema único e global de coisas que interajam sem problemas é praticamente impossível, pois não há um protocolo universal de aplicação para a IoT, existe uma heterogeneidade de dispositivos com diferentes hardwares e softwares neste ecossistema, sem um padrão para comunicação, deste modo essas redes ficam simplesmente isoladas. Para tornar essas redes interoperáveis precisa-se de uma única linguagem para dispositivos e aplicações que seja possível a trocar de informações, independentemente de como eles estão fisicamente conectados (GUINARD; TRIFA, 2016).

Nesse cenário surgiu um novo paradigma conhecido como WoT ou Web of Things. Segundo (CHRISTOPHE; VERDOT; TOUBIANA, 2011) a WoT é um paradigma e faz a abstração virtual do mundo real (IoT), algo como uma camada de software acima dos objetos reais, onde pode-se adaptar, criar modelos e protocolos, pois esta habilita acesso aos recursos (sensores) através da Web, o que pode ainda proporcionar acesso semântico e desta forma proporcionar uma maior interação entre dispositivos, sistemas e pessoas, são os novos ambientes digitais.

Na camada WoT pode ser realizada a abstração virtual de uma rede de sensores, tornando-se possível a criação de modelos de domínio de conhecimento para diferentes cenários. O maior objetivo é alavancar a conectividade entre dois mundos, o físico e o digital, fazendo com que a Internet atual englobe também objetos do mundo físico dos quais serão tratados como recursos Web.

Para tornar-se o ecossistemas de dispositivos IoT interoperáveis o paradigma da WoT utiliza protocolos e padrões amplamente aceitos na World Wide Web eliminando problemas de compatibilidade entre diferentes fabricantes, protocolos e formatos específicos.

2.2.1 Uniform Resource Identifiers

Uniform Resource Identifiers (URI) é a combinação de URL (Uniform Resource Locators) e URN (Uniform Resource Name). No que diz respeito ao protocolo de comunicação HTTP, URI são sequencias formatadas que identificam via nome, local ou qualquer outra característica um recurso na Internet (FIELDING et al., 1999).

2.2.2 Hypertext Transfer Protocol

O Hypertext Transfer Protocol (HTTP) é um protocolo de comunicação para nível de aplicação de sistemas de informação distribuídos, colaborativos e hipermídia. O HTTP foi usado pela iniciativa de informação global da World Wide Web desde 1990, basicamente o HTTP é um protocolo para a troca ou transferência de hipertexto (FIELDING et al., 1999).

Este protocolo permite que as mensagens trocadas contenham meta informações sobre os dados transferidos. Um cliente envia uma solicitação ao servidor na forma de um método de solicitação, seguido por uma mensagem contendo informações de clientes, conteúdos corporais e dados referentes a conexão com um servidor. O servidor responde com uma linha de status, incluindo a versão do protocolo da mensagem e um código de êxito ou erro, seguido por uma mensagem contendo informações do servidor. A Figura 2.6 representa essa troca de informações entre cliente e servidor.

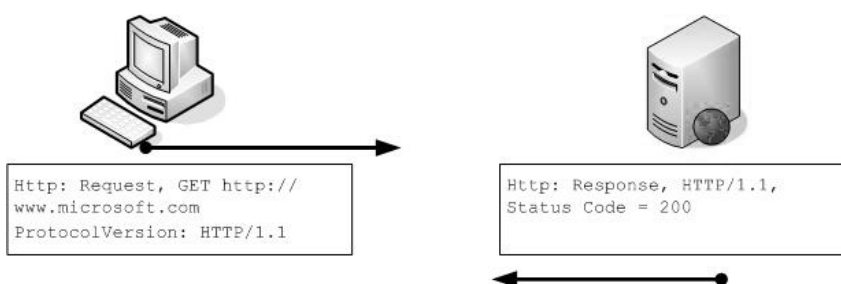


Figura 2.6 – Representação de uma requisição HTTP (MICROSOFT, 2007).

2.2.2.1 Mensagem HTTP

O protocolo HTTP faz a comunicação entre o cliente e o servidor por meio de mensagens, o cliente envia uma mensagem de requisição informando um recurso (URI) para o servidor, o servidor por sua vez busca o recurso e retorna uma mensagem de resposta ao cliente contendo o recurso solicitado.

A mensagem de requisição é composta por vários campos, abaixo será apresentado os principais campos que compõem a estrutura de uma requisição do protocolo HTTP (FIELDING et al., 1999).

- **REQUEST:** é a mensagem de solicitação de um cliente para um servidor, pode ser composta pelo cabeçalho e opcionalmente por um corpo. Na requisição é informado o método e a identificação do recurso utilizando uma URI.
- **HEADER:** o cabeçalho da requisição HTTP é utilizado para transmitir informações na mensagem entre cliente e servidor. Alguns cabeçalhos podem ser incluídos na mensagem de requisição, são eles: general-header, request-header,

response-header e entity-header. Estes campos são utilizados para transmitir informações adicionais entre cliente e servidor.

- BODY: o corpo da mensagem é opcional, quando houver é usado para receber o pedido ou resposta solicitado ao servidor. Para mensagens de resposta, se um corpo de mensagem está ou não presente depende tanto do método de pedido como da resposta do servidor.

2.2.3 Métodos HTTP

O HTTP define um conjunto de métodos que são utilizados nas requisições, dentre os quais se destacam os métodos GET, POST, PUT e DELETE. Os métodos indicam a ação desejada em um recurso indicado pela URI.

2.2.3.1 GET

O método GET é utilizado para recuperar qualquer informação (na forma de uma entidade) do servidor identificado pela URI de requisição. Se a URI de solicitação se refere a um processo de produção de dados, são os dados produzidos que devem ser retornados como a entidade no corpo da resposta e não o texto-fonte do processo, a menos que esse texto seja a saída do processo (FIELDING et al., 1999).

2.2.3.2 POST

O método POST é utilizado para solicitar a criação de um recurso no servidor identificado pela URI. Este método destina-se a publicar uma mensagem em um quadro de boletim, grupo de notícias, lista de discussão ou grupo similar de artigos, submeter um formulário a um processo de tratamento de dados, entre outros recursos de criação que podemos solicitar para o servidor. A função real executada pelo método POST é determinada pelo servidor e normalmente depende da URI de solicitação. A entidade postada é subordinada a esse URI da mesma forma que um arquivo é subordinado a um diretório que o contém. A ação executada pelo método POST pode não resultar em um recurso que pode ser identificado por um URI. Neste caso é retornado um código de status na resposta (FIELDING et al., 1999).

2.2.3.3 PUT

O método PUT é utilizado para solicitar a atualização de um recurso no servidor identificado pela URI. Se o URI informado na mensagem refere-se a um recurso já existente, a entidade incluída deve ser considerada como uma versão modificada daquele que reside no servidor de origem. Se o URI de solicitação não aponta para um recurso existente e esse URI é capaz de ser definido como um novo recurso, o servidor de origem pode criar o recurso com esse URI. Assim como no método POST

a ação executada pode não resultar em um recurso e então retornar um código de status na resposta (FIELDING et al., 1999).

2.2.3.4 DELETE

O método DELETE solicita ao servidor que exclua o recurso identificado pela URI. Este método pode ser substituído pela intervenção humana no servidor de origem. O cliente não pode garantir que a operação foi realmente executada, mesmo se o código de status retornado do servidor de origem indicar que a ação foi concluída com êxito. No entanto, o servidor não deve indicar sucesso a menos que, no momento em que a resposta é dada, ele pretende excluir ou mover o recurso para um local inacessível (FIELDING et al., 1999).

2.2.4 Representational State Transfer

O termo REST (Representational State Transfer) foi cunhado por (FIELDING; TAYLOR, 2000) em sua tese de doutorado. O REST define um conjunto de princípios a serem aplicados na construção de sistemas com uma arquitetura orientada a recursos (URI), este recurso pode ser tudo aquilo que deve ser acessado pelo cliente e transferido entre o mesmo e um servidor.

Ao desenvolver uma API REST existem algumas considerações, segundo (FIELDING, 2008) algumas regras são necessárias para diferenciar uma interface baseada em HTTP para uma API REST, a implementação concreta desse estilo considera os conceitos de recursos, endereçamento, interface uniforme, statelessness(sem estado, ou seja, não armazena a sessão no servidor) e hipermídia como o motor do estado da aplicação.

- Recursos (Resources): um recurso é qualquer componente de um aplicativo que tenha importância suficiente para ser identificado por uma URI. Um recurso pode ser um documento, uma página da web e até mesmo objetos físicos como sensores, motores, objetos inteligentes e tags RFID. A representação de um recurso contém as informações úteis sobre o estado atual de um recurso que são entregues a um cliente HTTP (RICHARDSON; RUBY, 2008; MAYER, 2010).
- Endereçamento (addressability): este princípio afirma que a informação sobre o recurso é mantida no endereço URI e não transmitida no corpo da solicitação HTTP (MAYER, 2010).
- Interface uniforme (Uniform Interface): este conceito define que um servidor deve ser capaz de determinar uma ação através da solicitação recebida por uma URI, ou seja, os recursos devem estar disponíveis com uma semântica de interação bem definida. O servidor determina a ação pelo método solicitado na requisição

HTTP (GET, POST, PUT e DELETE), dessa forma um serviço REST que possua essa característica está em conformidade com os princípios de endereçamento e interface uniforme. Um breve exemplo para esse princípio é quando um servidor recebe uma solicitação "GET/Usuario/1", a requisição solicita o recurso "Usuario" cujo código seja igual a um. Neste exemplo o "Usuario" é o recurso e "1" é um sub recurso de "Usuario", o servidor analisa o método "GET" presente na requisição HTTP e executa a ação (MAYER, 2010).

- Interação sem estado (Stateless): exige que a requisição efetuada pelo cliente seja independente, ou seja, todas as informações para atender a requisição devem fazer parte do pedido HTTP. Para isso não deve ser mantido qualquer estado de sessão do lado do servidor (GUINARD; TRIFA; WILDE, 2010).
- Hipermídia como o motor do estado de aplicação: os clientes de serviços REST devem fornecer links para outros recursos, isso permite aos clientes explorar os serviços simplesmente seguindo links, eliminando a necessidade de formatos de descoberta específicos (GUINARD; TRIFA; WILDE, 2010; MAYER, 2010).

2.2.5 Arquitetura WoT

Segundo (KAZUO et al., 2017) uma arquitetura funcional de WoT deve seguir alguns requisitos: flexibilidade, compatibilidade e segurança.

- Flexibilidade: Há uma grande variedade de dispositivos físicos, configurações e fornecedores para implementações de redes IoT. Estes não possuem um padrão de comunicação e muito menos conseguem interoperabilidade com outras redes. A arquitetura funcional do WoT deve ser capaz de ser mapeada e cobrir todas as variações.
- Compatibilidade: Quando ocorre uma padronização em paralelo a produção e desenvolvimento de uma tecnologia, no caso WoT, deve atentar-se as soluções IoT legadas. A arquitetura funcional do WoT deve fornecer uma ponte entre as soluções IoT legadas e a tecnologia WoT baseada em conceitos da Web, garantindo a compatibilidade das soluções atuais e legadas.
- Segurança: A arquitetura de WoT funcional deve fornecer funcionalidades de segurança, nas soluções IoT quando ocorre a quebra de segurança cibernética será desencadeado uma série de problemas, isso ocorre porque os dispositivos IoT hackeados geralmente controlam atividades essenciais no cotidiano das pessoas como sistemas de aquecimento, dispositivos móveis, carros e geladeiras.

2.2.5.1 Servidor Funcional WoT

A arquitetura funcional do Servidor WoT proposta por (KAZUO et al., 2017), é apresentada na Figura 2.7, o dispositivo virtual denominado "WoT Servient",

fornece acesso e controle a obtenção de valores dos dispositivos físicos da IoT.

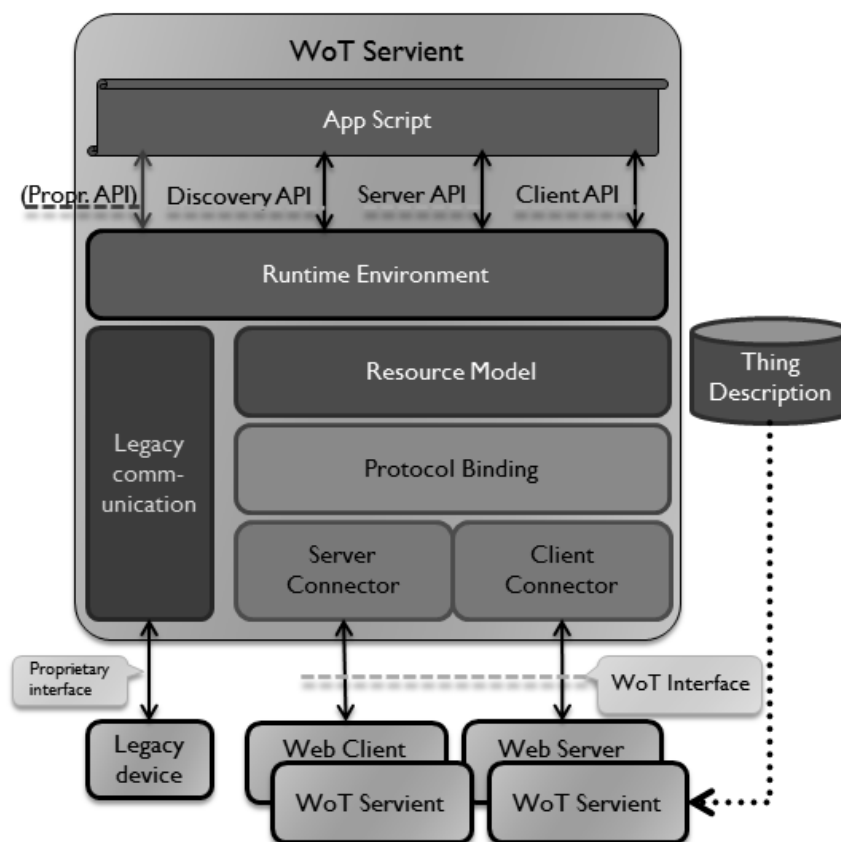


Figura 2.7 – Arquitetura funcional do Servidor WoT (KAZUO et al., 2017).

- **Server Connector (Conector do servidor):** é responsável por receber solicitações e enviar respostas aos clientes. Essa camada utiliza o protocolo HTTP para efetuar a transferência de dados entre cliente e servidor.
- **Client Connector (Conector do cliente):** essa camada acessa outros servidores da Web ou servidores WoT utilizando o protocolo HTTP.
- **Legacy Communication (Comunicação legado):** como existem uma heterogeneidade de dispositivos com diferentes hardwares e softwares no ecossistema de IoT, é necessário um módulo para comunicar com tais dispositivos herdados. Para isso essa camada inclui o mecanismo de vinculação de protocolos proprietários e o gerenciamento de recursos definidos por cada organização de gerenciamento de protocolo.
- **Protocol Binding (Protocolo de ligação):** essa camada converte interações com dispositivos usando informações dos objetos de acordo com a camada inferior.
- **Resource Model (Modelo de recurso):** fornece uma abstração entre os diferentes protocolos, como na Web, permite identificar e tratar as interações utilizando URI.
- **Things Description (Descrição das coisas):** fornece a documentação do objeto para o cliente externo requisitar a interface WoT.

- App Script (Script de aplicação): responsável por implementar a lógica da aplicação de forma modular e portátil, podendo assim acessar o hardware local, dispositivos remotos e legados da interface WoT.
- Runtime Environment (Ambiente de tempo de execução): o ambiente de tempo de execução fornece funções de servidor que aceita solicitações através da interface de WoT de outros clientes.

2.2.5.2 Cenários da arquitetura WoT

Como forma de ilustrar e exemplificar os a arquitetura WoT são apresentados alguns possíveis cenários utilizando WoT Servient, WoT Server e WoT Client, descritos por (KAZUO et al., 2017).

2.2.5.3 Servidor WoT no dispositivo

Trata-se de um aparelho eletrônico como um ar condicionado com funcionalidade de servidor Web conectado diretamente à rede doméstica local. Deste modo um controlador remoto com WoT cliente pode acessar o ar condicionado diretamente da rede local, por meio de um aplicativo ou navegador.

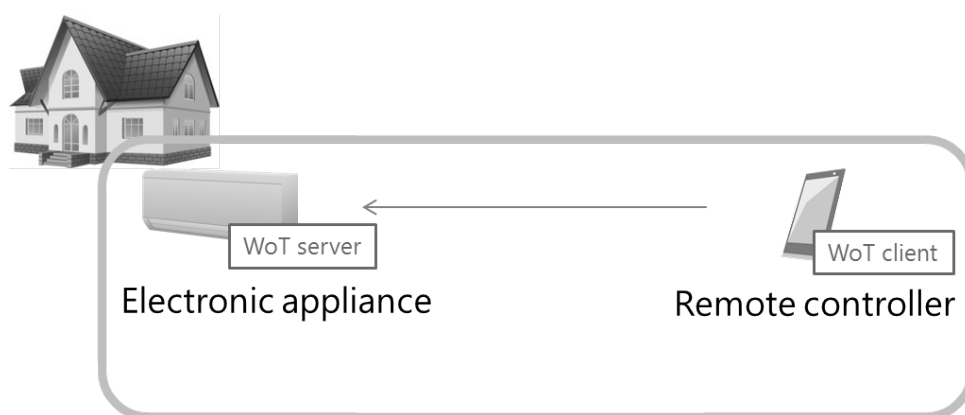


Figura 2.8 – Servidor WoT no dispositivo (KAZUO et al., 2017).

2.2.5.4 Cliente WoT no Smartphone

Esse caso é similar ao anterior. No entanto, o controlador remoto pode controlar diretamente ou remotamente o ar condicionado dependendo da rede em que está conectado. Caso esteja fora da rede doméstica local o controle remoto do objeto pode ser feito pela conexão Wi-Fi.

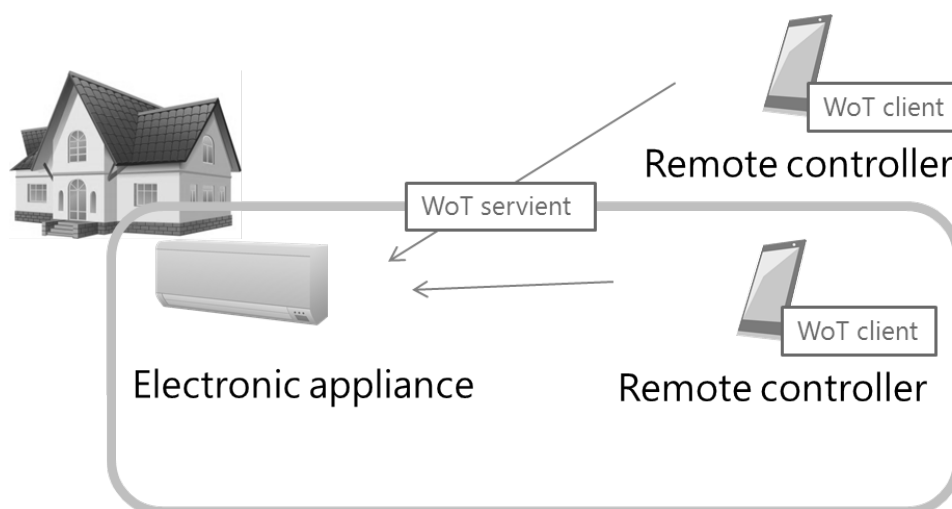


Figura 2.9 – Cliente WoT no Smartphone (KAZUO et al., 2017).

2.2.5.4.1 Servidor WoT no Smart Home Hub

Nesse cenário o controlador remoto não acessa diretamente o ar condicionado. Um Hub com servidor WoT é utilizado como ponte entre a rede doméstica e a Internet, gerenciando todos os aparelhos eletrônicos conectados na casa. Quando um controlador remoto solicita o acesso a um objeto na rede, primeiramente a requisição é enviada para o Hub e este gerencia os objetos.

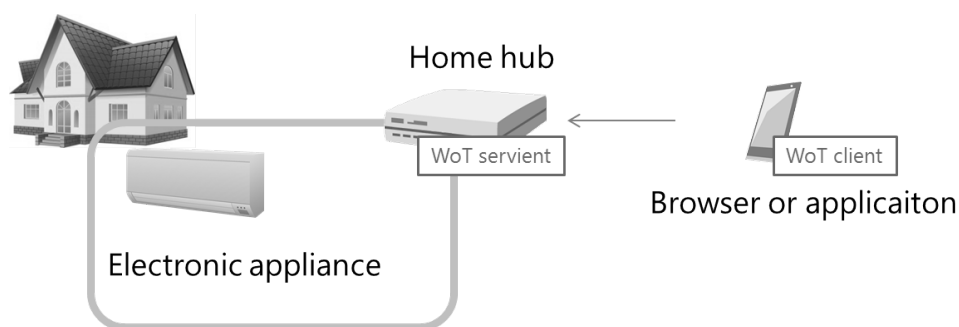


Figura 2.10 – Servidor WoT no Smart Home Hub (KAZUO et al., 2017).

2.2.5.4.2 Servidor WoT no servidor de nuvem

Esse cenário é parecido com o anterior, porém o controlador remoto não envia requisições diretamente ao Hub, e sim para um servidor WoT na nuvem colocado entre a rede doméstica e a Internet. O servidor WoT na nuvem gerencia as redes que estão conectadas e o Hub doméstico fica responsável por gerenciar os dispositivos conectados a casa.

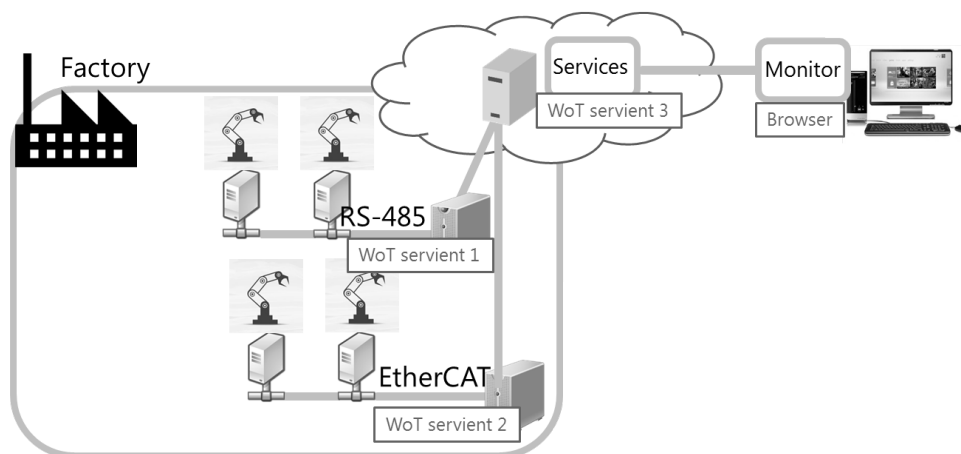


Figura 2.11 – Servidor WoT no servidor de nuvem em uma indústria inteligente (KAZUO et al., 2017).

2.3 Web Semântica

Devido ao crescimento exponencial dos dados gerados pelos sistemas atualmente e o constante avanço no desenvolvimento da IoT, a quantidade de dados produzidos será desmedido. Dentro desse contexto os desafios propostos a organização e a recuperação da informação tornam-se extremamente complexos, em redes IoT a interoperabilidade dos dados entre diferentes redes e sistemas é muito relevante, além da compreensão dessa informação tanto por seres humanos quanto por máquinas.

De modo a organizar e modelar a informação a fim de que possam ter significado e principalmente que se tornem passíveis de interpretação tanto por máquinas quanto por humanos, Tim Berners-Lee, em maio de 1994, na primeira conferência internacional sobre a World Wide Web, em Genebra propôs uma iniciativa que visava solucionar esse problema. Nesta conferência segundo Tim Berners-Lee, a forma que os documentos web estavam estruturados tornava compreensível apenas para seres humanos, para as máquinas os documentos eram praticamente inacessíveis por não terem um significado.

Mas somente passados sete anos Tim Berners-Lee publicou um artigo na revista Scientific American marcando o início da pesquisa relacionada à Web Semântica, neste artigo propõe essencialmente que se estruture os dados da Web de forma que eles possam ter significado e principalmente que se tornem passíveis de interpretação por máquinas, através de agentes computacionais. (ISOTANI; BITTENCOURT, 2015).

Em suma essa proposta visa a extensão da web como conhecemos hoje, para a web semântica. A ideia é prover uma estrutura para as páginas web onde seja possível ser compreendidas tanto por seres humanos quanto por máquinas, ou seja, as informações da web estariam estruturadas de forma semântica dando significado

as páginas.

Ainda na conferência internacional sobre a WWW, em Genebra, (BERNERS-LEE, 1994) complementa a ideia afirmando:

Adicionar semântica à web envolve duas coisas: permitir documentos que tenham informações em formulários legíveis por máquina e permitir que os links sejam criados com valores de relacionamento. Somente quando tivermos esse nível extra de semântica poderemos usar o poder do computador para nos ajudar a explorar a informação em maior medida do que a nossa própria leitura. (BERNERS-LEE, 1994, tradução nossa)

2.3.0.1 Camadas da Web Semântica

Objetivando ir além de conceitos e dar início a essa nova extensão da web surgiram várias representações de uma possível estrutura da web semântica e suas tecnologias necessárias para sua implementação. Porém sua primeira estrutura foi proposta por Tim Berners-Lee, descrevendo os recursos e as linguagens apresentados em camadas. Essa estrutura é conhecida como "bolo de noiva" ou "pirâmide da Web Semântica" (ISOTANI; BITTENCOURT, 2015).

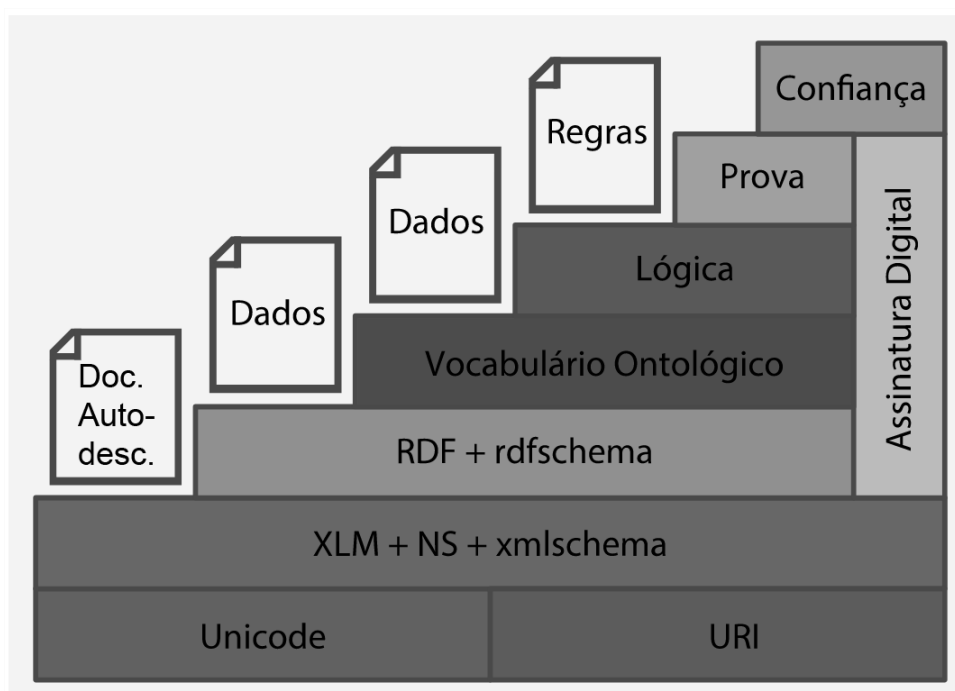


Figura 2.12 – Estrutura da Web Semântica (ISOTANI; BITTENCOURT, 2015).

- Unicode: O Unicode fornece um padrão universal de codificação, evitando que dois codificadores usem o mesmo número para dois caracteres diferentes ou usar números diferentes para o mesmo caractere (HEWLETT-PACKARD; INTERNATIONALIZATION; CUPERTINO, 1991).

- URI: são sequências que identificam via nome, local ou qualquer outra característica um recurso na Internet (FIELDING et al., 1999).
- XLM + NS + xmlschema: Extensible Markup Language (XML) é uma representação de dados simples projetado para a publicação eletrônica em larga escala (BRAY et al., 1997).
- RDF + rdfschema: RDF estrutura a ligação de recursos da web usando URIs para nomear a relação entre as coisas, é basicamente um modelo padrão para troca estruturada de dados na Web, porque possui recursos que facilitam a fusão de dados mesmo se os esquemas subjacentes diferem (DL, 2007).
- Assinatura Digital: é um código incluído em uma mensagem ou texto, para identificar o remetente da mensagem. No escopo da Web semântica, as assinaturas digitais tem como objetivo garantir a autenticidade e a veracidade das informações, desde a fonte de um documento RDF até a conferência de provas (HENRIQUE; IVAN; RADI, 2013).
- Vocabulário ontológico: A Web Semântica tem a capacidade de incluir repositórios de dados na Web, construindo vocabulários e regras para interoperarem com esses dados com o uso de tecnologias como RDF (LIMA; CARVALHO, 2004).
- Lógica: A adição da lógica é essencial para a consolidação da web semântica, o uso de regras para inferências torna possível a derivação de novos dados a partir de dados que já são conhecidos, ou seja, tem a intenção de transformar o documento em uma linguagem logica (LIMA; CARVALHO, 2004).
- Prova: Após passar por várias camadas a camada de prova analisará a veracidade das informações, para que as aplicações processem e decidam se confiam nestas informações, ou seja, pode-se provar hipóteses a partir de informações (LIMA; CARVALHO, 2004).
- Confiança: A camada de topo da estrutura tem como objetivo avaliar a correitude da prova. Essa avaliação é feita através da verificação da autenticidade e resolução de contradições (HENRIQUE; IVAN; RADI, 2013).

2.3.1 Ontologia

Quando trabalhamos com uma pluralidade de sistemas e aplicações que possuem diferentes estruturas para o tratamento da informação a interoperabilidade entre esses sistemas é dificultoso. Ao tentar combinar informações através de diferentes sistemas um mesmo conceito presente em um programa deverá saber quais termos estão sendo utilizados com o mesmo significado na aplicação correspondente.

Segundo (LIMA, 1998), um termo corresponde a um conceito particular dentro de um campo conceitual, os conceitos equivalem a um objeto que possuem um conjunto de propriedades e relações com outros conceitos em um determinado contexto.

Tal termo apresenta muitas variações de significado e interpretações distintas.

Uma solução para este problema é fornecida por um dos componentes básicos da Web Semântica, uma coleção de informações chamada de ontologia.

A concepção e o uso das ontologias sempre fizeram parte da proposta da Web Semântica, ela está no centro da arquitetura proposta por Tim Berners-Lee, e ao longo das últimas décadas tem-se tornado uma das principais tecnologias mais adequadas para lidar com grandes volumes de dados e fornecer a recuperação da informação de forma inteligente (ISOTANI; BITTENCOURT, 2015).

Historicamente o termo “ontologia” tem origem em um ramo da Filosofia (Metafísica) que estuda a natureza do “ser” e a “existência”. Para os filósofos, a Ontologia visa a explicar todas as coisas do mundo, estabelecendo sistematicamente sua linguagem conceitual (ISOTANI; BITTENCOURT, 2015). No entanto na Ciência da Computação, o significado e a finalidade do termo são distintos, a ontologia nesse ramo é um conjunto de informações e dados obtidos que pode ser definido como um conjunto de conceitos fundamentais e suas relações existentes no domínio em questão, permitindo a representação e o entendimento do mundo-alvo de maneira formal e compreensível por humanos e computadores, de forma semelhante a tomada na Filosofia (MIZOGUCHI, 2004).

A definição de ontologia mais frequentemente encontrada na literatura da Web Semântica é proposta por (GRUBER, 1995), “Uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada”.

Aqui conceitualização representa um modelo abstrato de algum fenômeno que identifica os conceitos relevantes para o mesmo. Explícita significa que os elementos e suas restrições estão claramente definidos; formal significa que a ontologia deve ser passível de processamento automático, e compartilhada reflete a noção de que uma ontologia captura conhecimento consensual, aceito por um grupo de pessoas. (BREITMAN, 2000).

Para (SEGUNDO, 2010) na Ciência da Computação, o termo “ontologia” é originário dos estudos de Inteligência Artificial, os estudos de ontologias estão inteiramente ligados a obtenção de conhecimento baseado em dados semiestruturados, aplicando um conjunto de métodos, técnicas ou processos automáticos ou semiautomáticos.

2.3.1.1 Tipos de ontologia

Segundo (GUARINO et al., 1998), as ontologias podem ser classificadas de acordo o objetivo de uso e com o contexto do domínio a ser aplicado. Guarino descreve quatro principais tipos de ontologias:

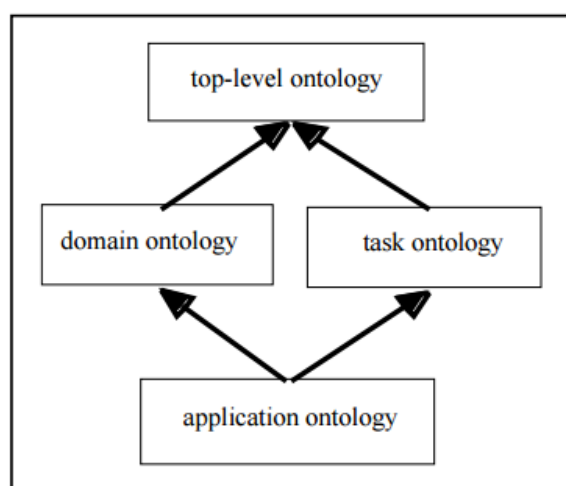


Figura 2.13 – Classificação da ontologia (GUARINO et al., 1998).

- Ontologias de topo (top-level ontologies): são ontologias que descrevem conceitos muito gerais como espaço, tempo, matéria, objeto, evento, entre outros. Essas ontologias não dependem de um determinado problema ou domínio.
- Ontologias de domínio (domain ontologies): têm a função de descrever o vocabulário relacionado a um domínio genérico, especializando os termos introduzidos na ontologia de topo. Alguns exemplos seriam ontologias descrevendo um domínio genérico como medicina ou automobilismo.
- Ontologias de tarefa (task ontologies): estas descrevem uma tarefa ou atividade genérica dentro da ontologia de domínio, como prontuários médicos dentro do domínio de medicina ou a compra de veículos, no domínio automobilístico.
- Ontologias de aplicação (application ontologies): são ontologias que descrevem uma aplicação dependente tanto de um domínio quanto de uma tarefa específica, que são muitas vezes especializações de ambas as ontologias relacionadas. Tais conceitos, correspondem ao papel desempenhado por alguma entidade dentro de um domínio, durante a execução de uma determinada tarefa.

2.3.2 Ontologia e IoT

O uso de tecnologias que agregam semântica podem ser tecnologias chaves para as redes de sensores, pois melhoram a interoperabilidade e a integração entre redes heterogêneas, além da comunicação humano-computador, já que passa a existir um relacionamento semântico definido, permitindo assim organizar, instalar e gerenciar dados e sensores por meio de especificações de alto nível. Existem inúmeros tipos de sensores e domínios em que podem ser utilizados, tais diferenças devem ser consideradas, como relações temporais e espaciais. Quando essas restrições e diferenças são formalmente representadas em uma ontologia, técnicas de inferência

podem ser aplicadas com maior facilidade. A ontologia para a redes de sensores deve ser muito abrangente visto que existe uma miríade de domínios e dispositivos distintos, além disso a ontologia deve capturar informações sobre as capacidades físicas do sensor, o desempenho e as condições nas quais ele pode ser usado. (ROB et al., 2017)

2.3.2.1 Semantic Sensor Network

Com o objetivo de padronizar e criar redes de sensores interoperáveis permitindo a representação mais expressiva e o acesso inteligente de recursos da rede, o consórcio W3C criou a ontologia Semantic Sensor Network (SSN), afim de agregar semântica com metadados espaciais, temporais e temáticos a rede de sensores.

A ontologia SSN é dividida em vários módulos com o objetivo de envolver todos os conceitos principais das redes de sensores.

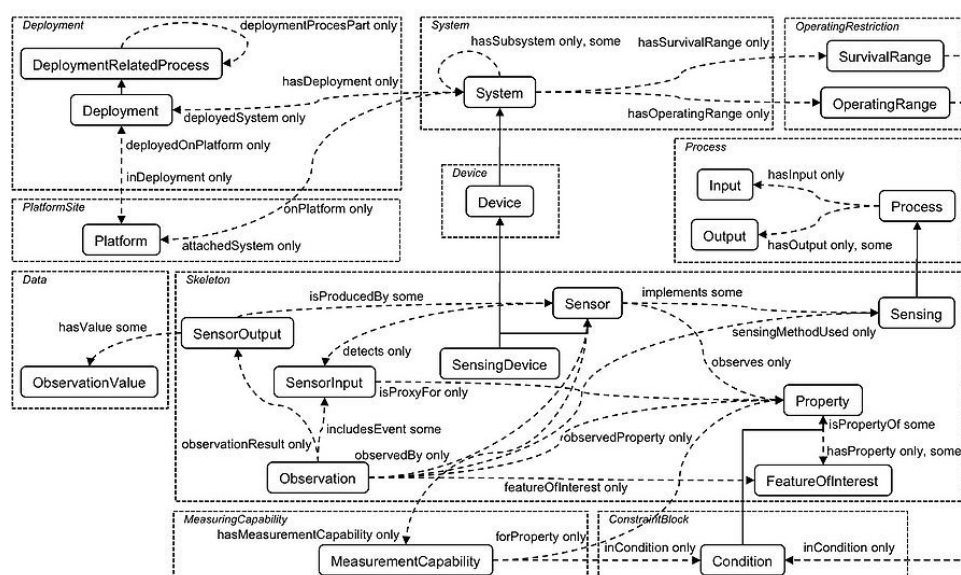


Figura 2.14 – Visão geral das classes e propriedades de ontologia SSN (ROB et al., 2017).

A modularização apresentada na ontologia SSN permite refinar e agrupar diferentes perspectivas presentes na rede de forma detalhada ou abstrata. Cada módulo contém classes e propriedades que podem ser usadas para representar aspectos particulares de um sensor ou suas observações (ROB et al., 2017).

2.3.3 Desenvolvimento

O desenvolvimento de uma aplicação baseada em ontologia deve considerar a manipulação de suas instâncias, algo complexo para desenvolvedores que utilizam padrões orientado a objeto. Segundo (ISOTANI; BITTENCOURT, 2015) atualmente,

existem duas abordagens principais utilizadas pelas ferramentas: desenvolvimento em triplas e desenvolvimento orientado a objetos.

2.3.3.1 Desenvolvimento em Triplas

A maioria das ferramentas atuais trabalha com o desenvolvimento baseado em triplas, o desenvolvedor deve conhecer muito bem a ontologia para poder criar e manipular cada tripla(sujeito, predicado e objeto). Para adicionar um recurso na ontologia são necessárias diversas linhas de código para descrever a tripla e seu recuso, as quais capturam apenas um valor de cada propriedade (ISOTANI; BITTENCOURT, 2015). O trecho de código abaixo exemplifica o desenvolvimento baseado em triplas em um repositório do Sesame.

```
...
ValueFactory f = myRepository.getValueFactory();
URI alisson = f.createURI("http://example.org/people/alisson");
URI name = f.createURI("http://example.org/ontology/name");
URI person = f.createURI("http://example.org/ontology/Person");
Literal alissonName = f.createLiteral("Alisson");
RepositoryConnection con = myRepository.getConnection();
con.add(alisson.RDF.TYPE, person);
con.add(alisson, name, alissonName);
...
```

2.3.3.2 Desenvolvimento Orientado a Objetos

Para os desenvolvedores já assemelhados com o padrão orientado a objeto, existem algumas ferramentas que traduzem as operações em objetos para o schema de representação em triplas. Algumas ferramentas foram criadas para prover esse paradigma para manipulação de instâncias em ontologias, neste caso podemos citar a poderosa ferramenta AliBaba (ISOTANI; BITTENCOURT, 2015).

O AliBaba é uma biblioteca de aplicativos que possui uma coleção de módulos que fornecem abstrações RDF, simplificando e acelerando o desenvolvimento e a manutenção de aplicativos semânticos. Além disso o AliBaba disponibiliza um repositório de objetos proporcionando uma grande expressividade e um ambiente de programação simplificado para o domínio.

Com essa ferramenta os desenvolvedores não precisam ter um conhecimento apurado da ontologia e de suas linguagens de representação. Um objeto representa uma instância na ontologia e seus respectivos atributos são mapeados com as pro-

priedades e as classes da ontologia. Logo, para adicionar um recurso da ontologia, basta adicionar um objeto.

O trecho de código abaixo exemplifica o desenvolvimento utilizando a ferramenta Alibaba.

```
...  
ObjectConnection con = repository.getConnection();  
Person alisson = new Person();  
alisson.setName("Alisson");  
con.addObject(alisson);  
...
```

Capítulo 3

DESENVOLVIMENTO

Esta seção tem por objetivo abordar os conceitos e técnicas envolvidos no desenvolvimento do presente trabalho. Inicialmente são abordados as tecnologias utilizadas para o desenvolvimento, a ontologia empregada para a estruturação semântica dos dados, os métodos de comunicação, assim como a arquitetura e a interface de manipulação dos dados.

3.1 Tecnologias

Para o desenvolvimento do projeto foi utilizado várias técnicas e tecnologias visando atingir o objetivo de abstrair o conhecimento relacionado a Web Semântica para os programadores.

Abaixo é apresentado uma breve descrição das principais tecnologias utilizada no projeto.

3.1.1 Apache Jena

O JENA é um framework de código aberto criado para o desenvolvimento de aplicações semânticas, originalmente desenvolvido por pesquisadores da HP Labs, no ano de 2000, dez anos mais tarde foi integrado pela equipe Apache Software Foundation e mantido até hoje (APACHE, 2017b).

Este framework possui um vasto conjunto de ferramentas e bibliotecas desenvolvido em Java para a manipulação de ontologias, além disso possui mecanismos de armazenamento para manipular grandes volumes de dados escritos em triplas RDF e motores de inferências baseados em raciocínio. Outras de suas principais ferramentas são:

- API RDF: ferramenta para a criação, leitura e escrita de dados em arquivos RDF.

- API OWL: ferramenta para a manipulação de modelos ontológicos em OWL (Web Ontology Language).
- API SPARQL: mecanismo de consulta utilizando a nova especificação do SPARQL.

3.1.2 Fuseki

Apache Jena Fuseki é um servidor SPARQL responsável por fazer a persistências de triplas RDF em um banco de dados, essa ferramenta é similar aos Sistemas de Gerenciamento de Banco de Dados (SGBD), contudo não é utilizado um modelo relacional para isso. Essa ferramenta está integrado ao Apache Jena e fornecendo uma camada de armazenamento persistente e transacional que incorpora consultas e armazenamentos em RDF.

O Fuseki pode ser executado como um serviço, aplicação ou servidor autônomo, fornecendo protocolos para consulta SPARQL. Além de possuir um interface para monitoramento e administração do servidor (APACHE, 2017a)

3.1.3 Protégé

O Protégé é uma ferramenta de código aberto criado em Java, atualmente está sendo desenvolvido e mantido pela Universidade de Stanford. O software pode ser instalado localmente em um computador ou ser acesso através da Web (STANFORD, 2017).

Essa ferramenta é muito poderosa para o desenvolvimento de sistemas baseados em conhecimento, permitindo manipular e criar modelos baseados em ontologia, além de possuir ferramentas para a construção de diagramas e gráficos. (<https://protege.stanford.edu/>)

3.1.4 Tomcat

O Apache Tomcat é um software de código aberto baseado em Java, criado para executar aplicações Web que utilizam as tecnologias JavaServer Pages, Java Servlet e Java WebSocket. O Tomcat é um servidor bastante estável com todas as características que um container comercial de aplicações web possui, atualmente é mantido por um grupo de voluntários da comunidade de código aberto do Java (TOMCAT, 2017).

3.1.5 Angular 4

Angular é um framework desenvolvido pela equipe do Google com um vasto conjunto de ferramentas que possibilita a criação de aplicativos Web, mobile e desktop.

Esse framework combina modelos declarativos, injeção de dependência e uma grande quantidade de bibliotecas e componentes para a manipulação do Document Object Model (DOM), utilização do protocolo HTTP, ganchos do ciclo de vida da página e muitas outras ferramentas para resolver diferentes desafios de desenvolvimento. Seu princípio é uma programação declarativa, estendendo as anotações HTML criando uma linguagem para desenvolvimento de interfaces web dinâmicas (GOOGLE, 2017).

3.2 Ontologia SSN e GEO

A modelagem de contexto utilizada para o desenvolvimento do motor semântico contempla a estrutura principal da ontologia Semantic Sensor Network (SSN), visando modelar de maneira formal os dados dos sensores, suas capacidades e as observações do ambiente. Além da utilização do módulo *Skeleton* da ontologia SSN foi utilizada uma ontologia externa para definir a geolocalização do sensor (GROUP, 2017).

A princípio, foi utilizado o módulo *Skeleton* da ontologia SSN para contemplar apenas os sensores que coletam dados, neste projeto não são considerados os estímulos do ambiente para a ativação de outros sensores na rede.

Na Figura 3.1 é apresentada a ontologia utilizada para o desenvolvimento do motor semântico, resultado da ontologia SSN e a fusão da ontologia de geolocalização.

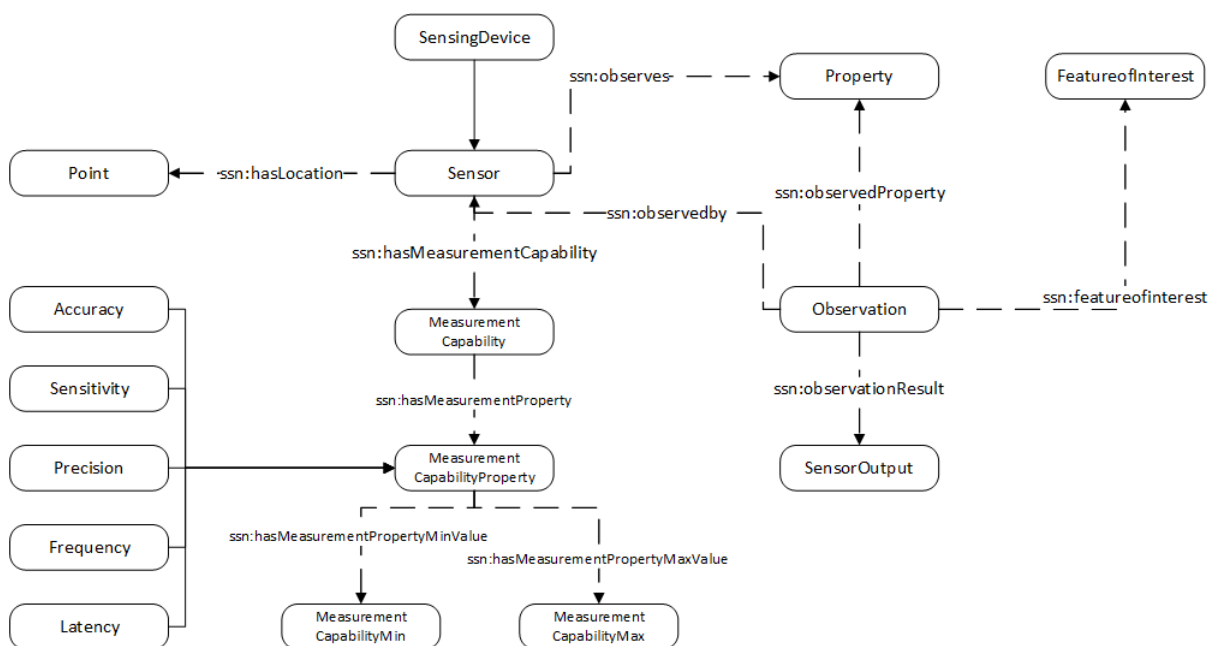


Figura 3.1 – Representação da ontologia utilizada no desenvolvimento do trabalho (Adaptado de (ROB et al., 2017)).

- Sensing Device: São os tipos de dispositivos de detecção do sensor, pode ser temperatura, umidade e etc;
- Sensor: Dispositivo físico que realiza as observações no ambiente de acordo com suas propriedades;
- Point: Ponto no espaço geográfico que indica um lugar;
- Property: Refere-se a uma qualidade observável pelo sensor, pode ser a altura de uma árvore, a profundidade de um corpo de água ou a temperatura de uma superfície;
- Feature of Interest: Indica uma característica de interesse que pode ser observada, uma característica de interesse da observação pode ser uma árvore, um ambiente ou um copo de água;
- Measurement Capability: Capacidade de medição do sensor;
- Measurement Capability Property: Refere-se as propriedades da capacidade de medição do sensor, frequência e etc;
- Measurement Capability Property (Min/Max): São os valores mínimos e máximos das propriedades da capacidade de medição do sensor;
- Observation: É o ato de realizar uma observação, estimar ou calcular o valor de uma propriedade de interesse (FeatureofInterest);
- Sensor Output: Armazena o valor resultante da observação (Observation).

3.3 Motor Semântico

Atualmente as redes de sensores ainda não expandiram da forma esperada, existem muitos problemas para se atingir o verdadeiro propósito da Internet of Things de maneira eficiente e inteligente. Para que seja possível que os objetos conectados ao nosso redor saibam o que gostamos, o que queremos, e o que precisamos, sem a intervenção humana e instruções explícitas, estes precisam entender o mundo-alvo onde estão inseridos.

Com a finalidade dos objetos entenderem o contexto que estão conectados é utilizado a modelagem baseada em ontologia provinda do paradigma da Web Semântica, com o uso de ontologias é possível definir um conjunto de conceitos fundamentais e suas relações existentes no domínio em questão, permitindo a representação formal e o entendimento do mundo-alvo, viabilizando a compreensão de seres humanos e computadores.

A adição da semântica nesses ambientes resultará em uma melhor recuperação da informação por agentes inteligentes, além da interoperabilidade entre diferentes redes já que as restrições e conceitos são explicitamente definidos na ontologia.

A proposta de motor semântico permite aos desenvolvedores criarem redes

de sensores inteligentes sem a necessidade de possuir conhecimento apurado da ontologia e seus linguagens de representações, além disso é feita a comunicação utilizando JSON (JavaScript Object Notation), um formato de texto simples que pode ser utilizado independente da linguagem de programação e já é conhecido por todos os desenvolvedores de software.

A criação do motor semântico de modo a resolver as problemáticas mencionadas possui a seguinte arquitetura:

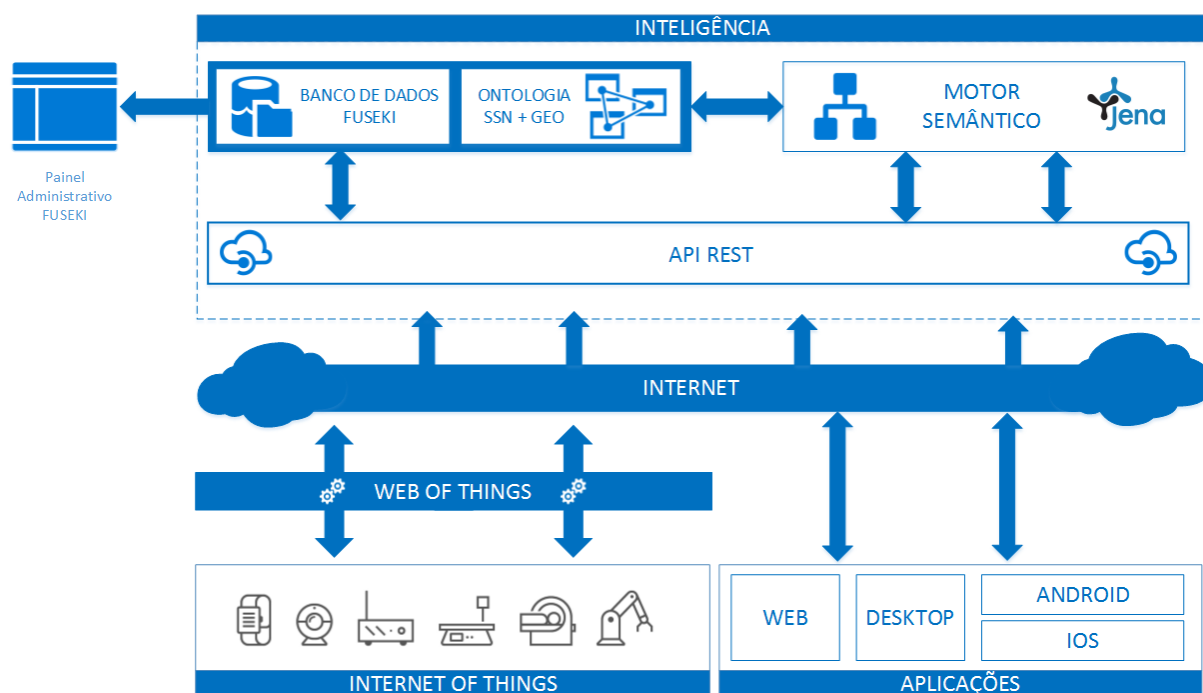


Figura 3.2 – Arquitetura do motor semântico (Autoria Própria).

3.3.1 Camadas da Arquitetura

3.3.1.1 Internet of Things

A camada da Internet of Things é onde encontra-se os sensores e as redes de sensores com os objetos aptos para a coleta de dados, esses objetos também devem estar preparados para a comunicação com a Internet. Os objetos da rede IoT estão além de smartphones, tablet e notebook já utilizados no nosso cotidiano para conectar à Internet, há outros equipamentos que se conectam à Internet para realizar atividades específicas, como exemplo uma câmera de segurança que monitore a sua casa à distância quando o estabelecimento está fechado, e caso ocorra alguma anomalia no ambiente o responsável recebe uma notificação no seu smartphone avisando sobre a ocorrência.

3.3.1.2 Web of Things

A camada Web of Things é responsável por fazer a lógica de comunicação do sensor com a Internet. Os sensores da rede coletam os dados brutos do ambiente, que então são enviados através do protocolo HTTP, o envio dos dados coletados para o motor semântico exige uma requisição na API desenvolvida utilizando o método POST. Os sensores obrigatoriamente precisam estar conectados à Internet para que seja possível a comunicação com a API do motor semântico.

A requisição para a API pode ser feita de acordo com a necessidade de coleta dos dados do ambiente, o desenvolvimento da lógica de envio de dados do sensor não exige nenhum tipo de conhecimento apurado do desenvolvedor sobre o sistema e suas tecnologias, apenas é necessário enviar um objeto do tipo JSON conforme apresentado abaixo:

```
{
  "nameSensor": "Sensor01",
  "descLabel": "Descrição",
  "descComment": "Descrição mais detalhada sobre a coleta do dado",
  "observedProperty": "Temperatura",
  "featureOfInterest": "Ambiente",
  "value": "10",
  "symbol": "C",
  "unit": "Celsius"
}
```

Abaixo é apresentada a documentação dos campos necessários para o envio dos dados.

- nameSensor: Nome do sensor cadastrado na rede, não é necessário a URI da ontologia.
- descLabel: Informar de forma breve a descrição da coleta, é um campo de preenchimento livre.
- descComment: Descrição detalhada da coleta, é um campo de preenchimento livre.
- observedProperty: Propriedade observada pelo sensor, essa propriedade é cadastrada no motor semântico e não é necessário informar a URI da ontologia.
- featureOfInterest: Define a característica de interesse que o sensor observa para a coleta dos dados, pode ser o ambiente, uma árvore, uma porta e etc. Não é necessário informar a URI da ontologia.
- value: Este campo é responsável por enviar o valor coletado pelo sensor

- unit: Unidade de medida do valor coletado
- symbol: Símbolo da unidade de medida do valor coletado

3.3.1.3 Internet

A camada de Internet implementa os protocolos necessários para o transporte de dados e a comunicação com outros dispositivos conectados à Internet. Além disso possui as tecnologias necessárias para a comunicação dos objetos IoT com a Internet.

É de fundamental importância que esta camada esteja operando para que os dados coletados sejam enviados e processados pelo motor semântico.

3.3.1.4 Inteligência

A camada de inteligência é a seção principal do projeto, responsável pela aplicação do conhecimento sobre os dados brutos coletados pelos sensores. É então aplicado um raciocínio transformando os dados coletados em informação semântica através da modelagem formal baseada em ontologia.

Nesta camada está localizado o motor semântico que possui três aplicações:

3.3.1.4.1 API Motor Semântico

A API desenvolvida em Java utiliza o estilo arquitetural REST, essa camada atua recebendo os dados dos sensores ou das interfaces de comunicação e quando necessário retorna os dados para consumo nas aplicações, o tráfego das informações é feito no formato JSON, este formato é utilizado para atingir a interoperabilidade entre diferentes linguagens de programação que pretendem utilizar os recursos da API.

A utilização da API é realizada indicando a URI do recurso e a ação desejada de acordo com o conjunto de métodos utilizados pelo protocolo HTTP.

3.3.1.4.2 GET

Para efetuar uma consulta é realizada uma requisição utilizando o método GET, indicando o domínio da API e o recurso, no exemplo abaixo será apresentado como fazer um consulta de todas as instâncias do tipo Property na ontologia.

- Domínio da API: `http://localhost:8084/FrameworkSSIOT/api/`
- Recurso: `property`
- Método: `GET`

O pedido (Request) será feito utilizando os elementos citados acima, o cabeçalho (Header) da requisição será feito da seguinte forma:

GET <http://localhost:8084/FrameworkSSIOT/api/property>

O retorno (Result) do pedido é apresentado no formato JSON com uma lista de todos os objetos e seus respectivos atributos na ontologia.

```
[
  {
    "type": "Property",
    "comment": "Temperature property",
    "label": "Temperature",
    "nome": "PropertyTemperature",
    "uri": "http://www.ssn.org/iot#PropertyTemperature"
  },
  {
    "type": "Property",
    "comment": "Humidity property",
    "label": "Humidity",
    "nome": "PropertyHumidity",
    "uri": "http://www.ssn.org/iot#PropertyHumidity"
  }
]
```

Abaixo é apresentada a documentação dos campos retornados:

- Type: Classe da instância
- Comment: Propriedade da instância
- Label: Propriedade da instância
- Nome: Apenas o identificador do recurso
- URI: URI completa do recurso

3.3.1.4.3 POST

Para realizar a criação de uma instância na ontologia utilizamos o mesmo procedimento da consulta, porém o método utilizado é o POST e em sua requisição informamos no formato JSON o objeto que deve ser instanciado na ontologia.

O pedido é efetuado da seguinte maneira:

- POST <http://localhost:8084/FrameworkSSIOT/api/property>
- Body:

```
{  
  "indName": "PropertyTemperature",  
  "label": "Temperature",  
  "comment": "Temperature property"  
}
```

Abaixo é apresentada a documentação dos campos enviados no corpo da requisição:

- indName: Identificador do recurso
- Label: Propriedade da instância
- Comment: Propriedade da instância

3.3.1.4.4 DELETE

O método DELETE é repensável por excluir um recurso na ontologia, a requisição é feita da mesma forma do método GET, com exceção do método utilizado e do identificador do recurso na ontologia.

- Domínio da API: <http://localhost:8084/FrameworkSSIoT/api/>
- Recurso: property
- Identificador na ontologia: PropertyTemperature
- Método: DELETE

O pedido (Request) será feito utilizando os elementos citados acima, o cabeçalho (Header) da requisição será feito da seguinte forma:

```
DELETE http://localhost:8084/FrameworkSSIoT/api/property/PropertyTemperature
```

3.3.1.4.5 Motor Semântico

O motor semântico é responsável por serializar o JSON recebido pela API e executar a lógica responsável por manipular a ontologia, realizando consultas e persistindo novas instâncias.

Ao fazer uma requisição de busca com o método GET é retornado um conjunto de objetos do tipo JSON com os dados da ontologia. Primeiramente a aplicação faz o carregamento e a validação da ontologia armazenada no banco semântico, logo após a validação da ontologia é enviado ao motor semântico a solicitação de busca na ontologia, o motor analisa se a busca é por classes ou indivíduos, além de analisar as subclasses e a URI de domínio de acordo com a busca solicitada. Ao completar

a busca da instância ou classe na ontologia é retornado a quantidade de objetos em formato JSON com os principais atributos vinculados.

O método POST responsável por criar uma nova instância na ontologia tem seu fluxo muito parecido com uma requisição de busca, a aplicação faz o carregamento e a validação da ontologia armazenada no banco semântico, logo após a validação da ontologia é enviado ao motor semântico o objeto JSON contendo os dados para persistência na ontologia. Na próxima etapa é feita a busca do tipo da classe e seguidamente gerado a instância do objeto na ontologia, quando é vinculada alguma propriedade a instância criada o motor se encarrega de criá-la, caso não exista na ontologia.

Por fim, o processo de exclusão de uma instância na ontologia utiliza o mesmo fluxo do método de busca, porém ao encontrar o objeto na ontologia não é retornado a lista de objetos e sim feita a exclusão do mesmo.

3.3.1.4.6 Banco Semântico

O banco de dados semântico utiliza a tecnologia de armazenamento de ontologias do FUSEKI, esta camada é responsável pelo armazenamento da ontologia no servidor, desta forma é possível manipular e persistir a ontologia em um único lugar, como um banco de dados convencional, para que seja acessível através de todas as aplicações desenvolvidas sempre com os dados atualizados.

Com a interface HTTP do FUSEKI podemos administrar o banco semântico, fazer consultas em SPARQL e alterações na ontologia.

3.3.1.5 Aplicações

A camada de aplicações é um dos principais resultados na arquitetura em virtude do desenvolvimento da API REST do motor semântico. A API possibilita a criação de interfaces em diferentes linguagens de programação, concedendo a eficácia na interoperabilidade, visto que o protocolo de comunicação é o HTTP e a troca de dados é feita no formato JSON, compatível com qualquer linguagem de programação.

3.3.2 Aplicação Semantic Internet of Things

Neste projeto foi desenvolvida uma interface utilizando a tecnologia Angular4, essa aplicação tem como objetivo administrar e parametrizar uma rede IoT utilizando o motor semântico que emprega a modelagem de dados baseada em ontologia.

A interface possui um menu lateral para acessar as principais funcionalidades da aplicação, as quatro primeiras funcionalidades são para efetuar a parametrização

inicial da rede de sensores na ontologia. Abaixo é apresentado de forma detalhada como está distribuída as funcionalidades da interface.

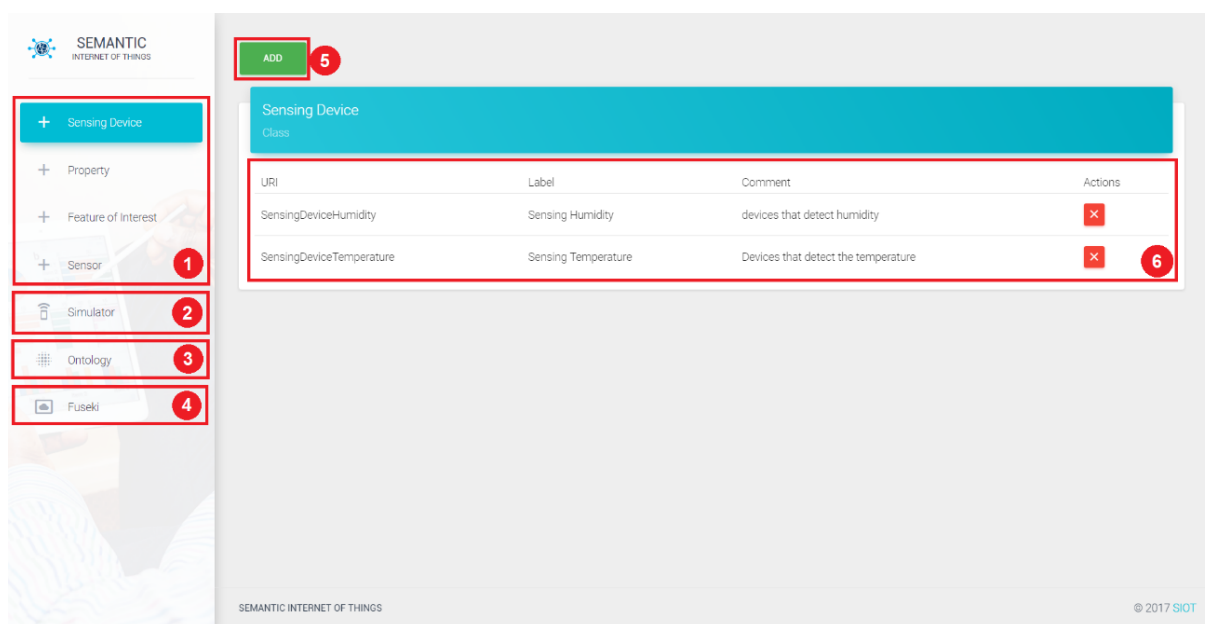


Figura 3.3 – Tela principal da interface (Autoria Própria).

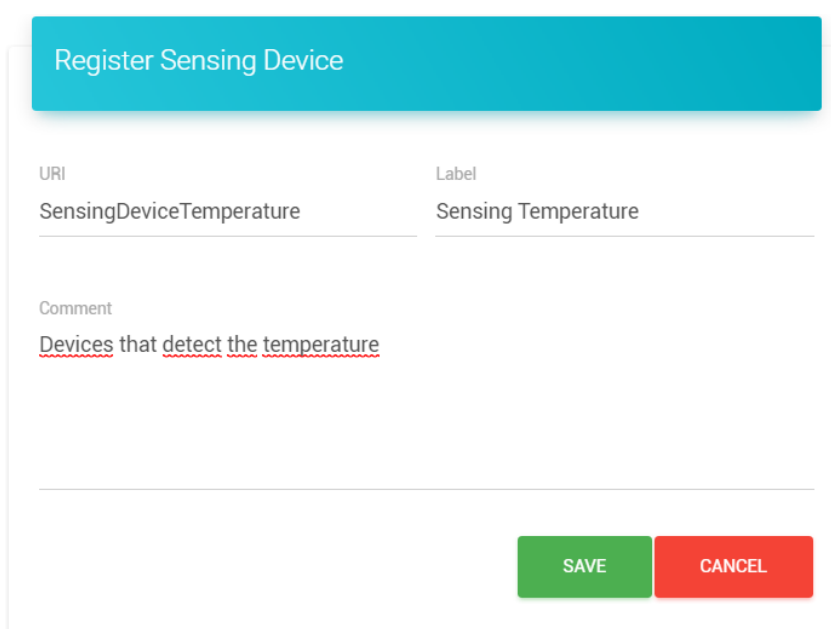
1. Todas as opções de instâncias e consultas podem ser acessadas a partir destes menus, as opções Sensing Device, Property e Feature of Interest possuem a mesma interface, apresentando uma lista das instâncias criadas, um botão para adicionar e a opção para excluir a instância. A interface mais complexa é a opção do Sensor que será tratada mais adiante;
2. Na opção Simulator é possível incluir os sensores instanciados na ontologia e criar simulações de observações;
3. Essa funcionalidade apresenta o código da ontologia salva no banco de dados semântico;
4. Na opção Fuseki é possível visualizar o painel administrativo do banco semântico;
5. Essa opção apresenta uma nova tela para incluir os valores dos atributos em cada instância;
6. No quadro é listado todas as instâncias referentes a funcionalidade selecionada no menu lateral, podendo também excluir uma instância.

3.3.2.0.1 Sensing Device

Nesta funcionalidade o usuário poderá criar tipos de dispositivos de detecção, na tela é demonstrado todas as subclasses de SensingDevice que estão persistidas na ontologia, é possível obter alguns detalhes da URI, label e comment, além da opção de excluir a classe da ontologia.

No botão “ADD” (Item 5) é apresentada uma nova tela permitindo a inclusão de uma nova classe e seus atributos. A URI é abstraída, desta forma não é necessário indicar o domínio, mas apenas o identificador.

SensingDeviceTemperature => <http://www.ssn.org/iot#> SensingDeviceTemperature



Register Sensing Device

URI
SensingDeviceTemperature

Label
Sensing Temperature

Comment
Devices that detect the temperature

SAVE CANCEL

Figura 3.4 – Tela para inclusão da instância na ontologia (Autoria Própria).

3.3.2.0.2 Property

A funcionalidade Property permite que sejam criados os tipos de propriedades observadas pelo sensor (Temperatura, Humidade, Ar), além de demonstrar todas as instâncias da subclasse Property persistidas na ontologia. Essa tela segue a mesma estrutura da tela Sensing Device, permitindo a visualização e a exclusão das instâncias e o botão “ADD” para persistir uma instância na ontologia.

3.3.2.0.3 Feature of Interest

A opção Feature of Interest podem ser instanciadas as características de observação do sensor, um sensor pode ter a propriedade de temperatura tendo como

característica de interesse um ambiente ou o berço de um bebê. Essa tela segue a mesma estrutura da tela Sensing Device, permitindo a visualização e a exclusão das instâncias e o botão “ADD” para persistir uma instância na ontologia.

3.3.2.0.4 Sensor

Diferente das funcionalidades de parametrizações anteriores a persistência do sensor na ontologia necessita que existam na ontologia classes de SensingDevice e instâncias do tipo Property.

The screenshot displays a web form titled "Register Sensor" with a light blue header. Below the header, a grey button labeled "SELECT SENSOR TYPES" is highlighted with a red box and a red circle containing the number 1. The form is divided into several sections, each with a light blue header: "Register Property", "Register Measurement", and "Register Spatial Thing". The "Register Property" section contains a green "ADD PROPERTY" button, followed by input fields for "URI", "Label", "Comment", and "Actions". The "Register Measurement" section contains a grey "PROPERTY SELECT" button, a green "ADD MEASUREMENT" button, and input fields for "URI", "Type", and "Actions". The "Register Spatial Thing" section contains input fields for "URI", "Label", "Lat", "Long", "Alt", and "Comment". A red box and a red circle containing the number 2 highlight the "URI" and "Label" fields in the "Register Property" section. A red box and a red circle containing the number 3 highlight the "ADD PROPERTY" button and the "URI", "Label", "Comment", and "Actions" fields in the "Register Property" section. A red box and a red circle containing the number 4 highlight the "PROPERTY SELECT" and "ADD MEASUREMENT" buttons and the "URI", "Type", and "Actions" fields in the "Register Measurement" section. A red box and a red circle containing the number 5 highlight the "URI", "Label", "Lat", "Long", "Alt", and "Comment" fields in the "Register Spatial Thing" section. At the bottom right of the form, there are two buttons: a green "SAVE" button and a red "CANCEL" button.

Figura 3.5 – Tela para a criação de um sensor na ontologia (Autoria Própria).

1. Neste item são selecionados todos os tipos de sensores incluídos na tela “Sensing Device”, primeiramente é necessário definir o tipo do sensor.

2. Após definir o tipo podem ser incluídas algumas informações básicas sobre o sensor: URI, Label, Comment.
3. Na opção “Register Property” é possível selecionar as propriedades cadastradas na tela “Property”, podendo ser incluídas ao sensor na opção “ADD PROPERTY”, ao acionar este botão é apresentada uma tela com a lista de todas as propriedades instanciadas na ontologia, permitindo adicionar a propriedade desejada.
 - a) Ao adicionar a propriedade é possível visualizá-las na tela do sensor, se necessário também é possível excluir as propriedades selecionadas
4. Após selecionadas as propriedades do sensor é possível definir as suas capacidades de medição, cada propriedade possui as suas capacidades de medição. Ao selecionar a propriedade desejada o campo “Property Select” indica a URI da propriedade selecionada.
 - a) Ao acionar o botão “ADD MEASUREMENT” uma nova tela é apresentada para que seja feita a inclusão dos atributos para a capacidade de medição da propriedade do sensor.

The image shows a web interface for registering measurement capabilities. At the top, there is a 'FREQUENCY' dropdown menu. Below it, the 'URI' label is visible. The main form consists of two identical-looking sections, one for 'Min' and one for 'Max' measurements. Each section has a blue header bar with the title 'Register Measurement Min' or 'Register Measurement Max'. Inside each section, there is a 'Value' input field containing the number '0'. Below the value field, there are two input fields labeled 'Unit' and 'Symbol'. At the bottom right of the form, there are two buttons: a green 'SAVE' button and a red 'CANCEL' button.

Figura 3.6 – Tela para incluir Measurement Capability (Autoria Própria).

- b) Inicialmente é definida a propriedade da capacidade de medicação, essas propriedades já estão definidas na ontologia e não são parametrizadas.
 - c) Após definir a propriedade podemos adicionar a URI e os atributos de medição mínimo e máximo.
5. A ultima etapa permite que seja definido uma geolocalização para sensor, podem ser indicados os valores de latitude, longitude e altitude, além das informações básicas como URI, label e comment.

A Figura 3.7 apresenta as instâncias no software Protégé, as instâncias foram criadas a partir da interface desenvolvida simulando a parametrização de um ambiente IoT para a utilização do motor semântico.

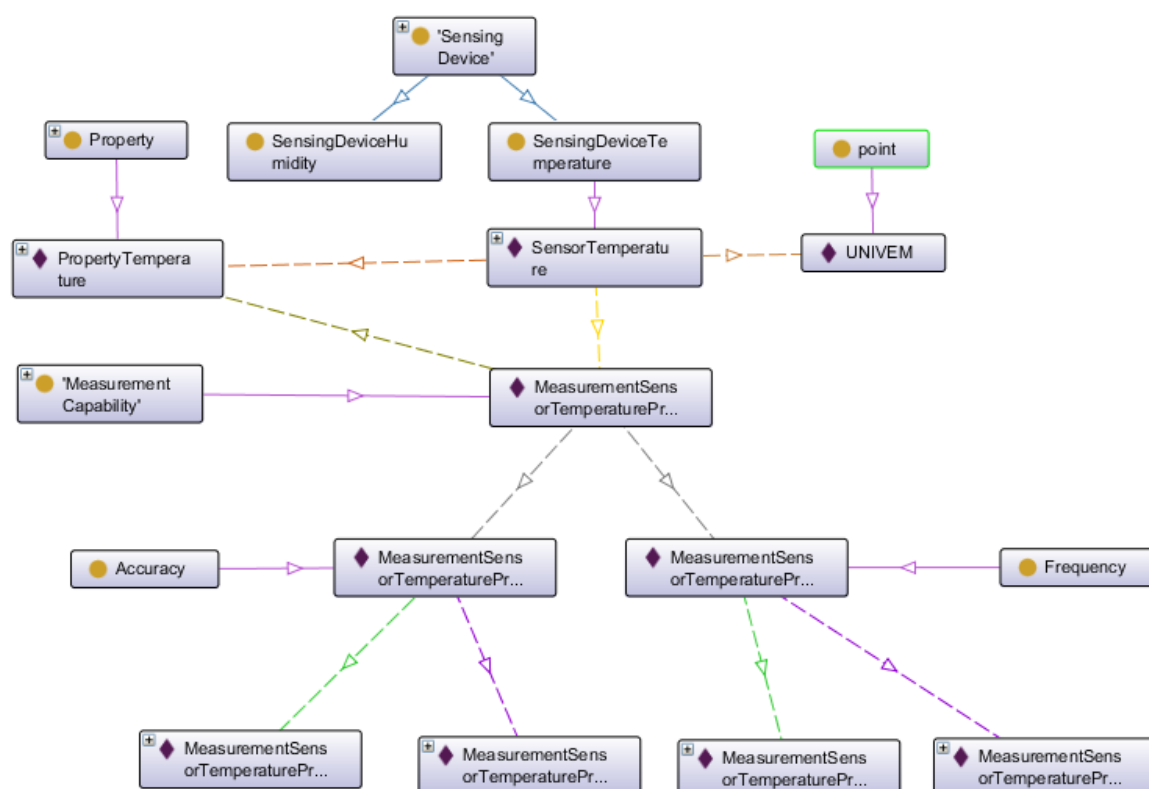


Figura 3.7 – Instâncias da ontologia criadas na interface (Autoria Própria).

3.3.2.0.5 Simulator

Sendo uma das principais funcionalidades da interface, a tela de simulação permite que seja simulado as observações dos sensores que estão persistidos na ontologia, gerando as observações de maneira dinâmica.

Acionando o botão “ADD SENSOR” é apresentada uma tela onde podemos seleccionar os sensores que estão cadastrados. Ao incluir o sensor são apresentados os dados de tipo de detecção, label, comment, property.

O botão “SETTINGS” apresenta uma nova tela permitindo que seja configurado as opções de observação do sensor, inicialmente é selecionado a característica de interesse que o sensor está observando e depois podem ser incluídos alguns atributos básicos para a simulação. No final da tela é demonstrado o objeto JSON da configuração, esse objeto pode ser utilizado no sensor físico para efetuar observações e enviá-las ao motor semântico.

Settings ×

ENVIRONMENT ▾

Label	Value
Descrição	10

Unit	Symbol
C	Celsius

Comment
Descrição mais detalhada sobre a coleta do dado

JSON

```
{ "nameSensor": "SensorTemperature", "descComment": "Descrição mais detalhada sobre a coleta do dado", "descLabel": "Descrição", "observedProperty": "PropertyTemperature", "featureOfInterest": "FTEnvironment", "value": "10", "symbol": "Celsius", "unit": "C" }
```

Figura 3.8 – Tela de configuração dos dados para simulação (Autoria Própria).

Após definir as configurações de observação podemos iniciar ou parar a simulação do sensor utilizando os botões “START” e “STOP” respectivamente. Ao iniciar a simulação é apresentado um gráfico em tempo real dos valores observados pelo sensor e logo abaixo um log de retorno da API. Os valores das observações são atualizados podendo variar de -3 à 3 com relação ao valor definido na configuração de simulação.

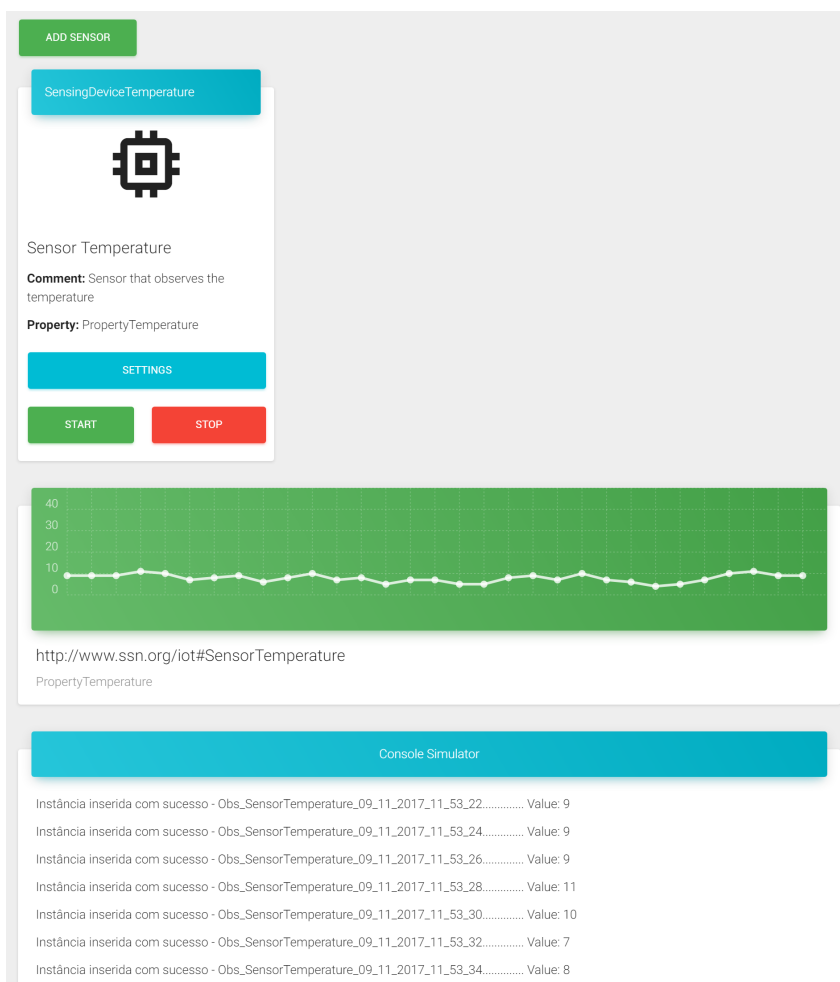


Figura 3.9 – Tela de simulação do envio de dados (Autoria Própria).

Após a parametrização inicial da rede IoT na ontologia é possível que sejam enviados os dados coletados pelos sensores para o motor semântico. A Figura 3.10 representa as instâncias criadas na ontologia a partir dos dados transmitidos pelo simulador.

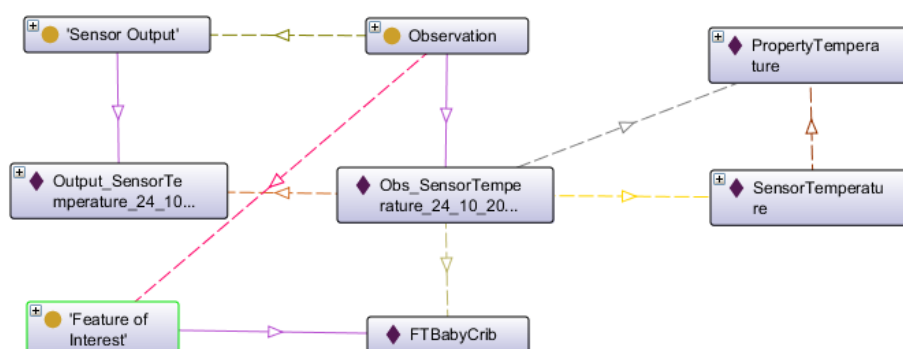


Figura 3.10 – Instâncias de observação (Autoria Própria).

3.3.2.0.6 Ontology

Essa funcionalidade permite visualizar a ontologia atualizada no banco semântico no formato de triplas



Figura 3.11 – Visualização da ontologia (Autoria Própria).

3.3.2.0.7 Fuseki

A opção FUSEKI permite acessar a interface de administração HTTP do banco semântico FUSEKI, nessa tela podemos verificar se servidor do banco de dados está funcionando, além de desfrutar de toda a interface do FUSEKI na aplicação.

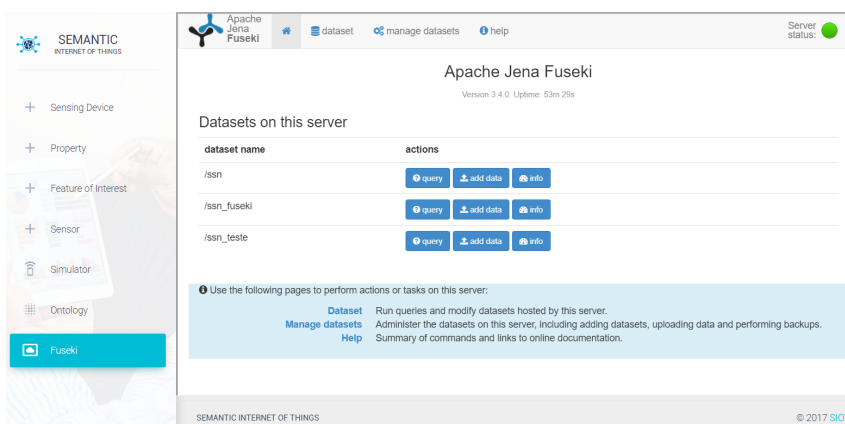


Figura 3.12 – Interface administração Fuseki (Autoria Própria).

Capítulo 4

CONCLUSÃO

4.1 Contribuições e Limitações

4.1.1 Contribuições

Este trabalho teve como objeto um motor semântico para auxiliar no desenvolvimento de aplicações utilizando a modelagem baseada em ontologias nos ambientes IoT. O motor semântico pretende abstrair toda a complexidade do desenvolvimento que é encontrada no paradigma da Web Semântica. A API desenvolvida torna possível a interoperabilidade entre diferentes linguagens de programação, pois a transmissão e o recebimento das informações utilizam o formato JSON que pode ser utilizado em qualquer ambiente.

Os objetos IoT conectados à Internet formam uma grande rede de sensores com o objetivo de coletar dados e contribuir em um mundo melhor para os seres humanos, devido à grande quantidade de objetos haverá um crescimento exponencial dos dados gerados por estes novos ambientes, sem a adição de semântica aos dados coletados não é possível gerar informações relevantes ao usuário, muito menos criar sistemas interoperáveis e utilizar técnicas de inferências e inteligência artificial para tornar estes ambientes realmente inteligentes atingindo seu propósito. Diferentemente do modelo relacional a modelagem baseada em ontologia possibilita a estruturação semântica e formal dos dados, viabilização a compreensão dos dados tanto por seres humanos quanto por computadores através de agentes inteligentes.

Com a utilização da Web Semântica e o motor semântico apresentado neste trabalho, cuja sua principal contribuição é a API que facilita o desenvolvimento abstraindo o conhecimento apurado em Web Semântica é possível obter dupla interoperabilidade, a primeira com relação ao desenvolvimento que pode ser feito em qualquer linguagem que utiliza o protocolo HTTP, e a segunda que contribui para a interoperabilidade entre diferentes ambientes e sistemas IoT já que passa a existir uma relação

entre as informações e suas relações no domínio em questão. Além da interoperabilidade produzida pela API entre as linguagens de programação, podemos empregar este benefício para a disponibilização de Dados Abertos Conectados que fazem parte da Web Semântica.

4.1.2 Limitações

As limitações do projeto no presente trabalho estão relacionadas diretamente a quantidade de dados produzidos pelos sensores da rede IoT. A capacidade de requisições recebidas pela API do motor semântico e a manipulação da ontologia podem vir a ser um problema.

A ontologia ainda não está completa para atender os diferentes tipos de sensores e propriedades existentes nos mais diversos ambientes, para tornar isso possível pode ser feita a agregação de outros vocabulários com a ontologia SSN, enriquecendo a sua estrutura.

4.2 Lições aprendidas

Foram diversas lições aprendidas com o desenvolvimento deste trabalho, primeiramente foi despertado o anseio pela pesquisa do conhecimento com relação a Web Semântica e Internet of Things. Devido a Web Semântica nunca fazer parte das minhas pesquisas foi muito complexo buscar o entendimento de suas tecnologias e a sua proposta, mas isso foi um grande incentivo para a leitura de diversos artigos e alguns livros sobre o tema.

A falta de conhecimento prévia sobre Web Semântica tornou o trabalho um pouco mais difícil e demorado, além da compreensão e utilização de suas tecnologias para a manipulação da ontologia e a criação do motor semântico.

Quanto ao desenvolvimento da interface e a criação da estrutura da API não houve dificuldade, pois são tecnologias já utilizadas e consolidadas no mercado facilitando a pesquisa.

4.3 Trabalhos Futuros

Para trabalhos futuros, o processo de recuperação da informação nos ambientes IoT que possuem a modelagem de dados baseada em ontologia é algo muito relevante, a velocidade e a inteligência desse processo exigem um estudo pautado, pois a medida que esses ambientes estão evoluindo a quantidade de informação gerada será demasiada, gerando dificuldades no processo de recuperação da informa-

ção. A inteligência do processo de recuperação da informação também é de extrema importância, por meio dela podemos fazer buscas na rede IoT utilizando linguagem natural.

REFERÊNCIAS

- ABOWD, G. D. et al. Towards a better understanding of context and context-awareness. In: SPRINGER. *International Symposium on Handheld and Ubiquitous Computing*. [S.l.], 1999. p. 304–307.
- AGGARWAL, C. C.; ASHISH, N.; SHETH, A. The internet of things: A survey from the data-centric perspective. In: *Managing and mining sensor data*. [S.l.]: Springer, 2013. p. 383–428.
- APACHE, J. *Apache Jena Fuseki*. 2017. Disponível em: <<https://jena.apache.org/documentation/fuseki2/>>.
- APACHE, J. *What is Jena?* 2017. Disponível em: <https://jena.apache.org/about_jena/about.html>.
- ASIN, A.; GASCON, D. 50 sensor applications for a smarter world. *Libelium Comunicaciones Distribuidas, Tech. Rep*, 2012.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- BATALLA, J. M. et al. *Beyond the Internet of Things: Everything Interconnected*. [S.l.]: Springer, 2016.
- BERNERS-LEE, T. *Plenary at WWW Geneva 94*. 1994.
- BRAY, T. et al. Extensible markup language (xml). *World Wide Web Journal*, v. 2, n. 4, p. 27–66, 1997.
- BREITMAN, K. K. *Web semântica: a internet do futuro*. [S.l.]: Grupo Gen-LTC, 2000.
- CASOTTI, R. F. d. N. M. lot na indústria: a quarta revolução está chegando. 2016.
- CHRISTOPHE, B.; VERDOT, V.; TOUBIANA, V. Searching the 'web of things'. In: IEEE. *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*. [S.l.], 2011. p. 308–315.
- COMMISSION, E. Internet of things in 2020 road map for the future. In: *RFID WORKING GROUP OF THE EUROPEAN TECHNOLOGY PLATFORM ON SMART SYSTEMS INTEGRATION (EPOSS)*. [S.l.: s.n.], 2008.

- COMMUNITIES, C. of the E. Internet of things — an action plan for europe. *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions*, 2009.
- CORREIA, M. B. F. *Infográficos: Uma Abordagem Ergonômica*. Tese (Doutorado) — PUC-Rio, 2009.
- DL, O. Resource description framework (rdf). 2007.
- FIELDING, R. et al. *Hypertext transfer protocol—HTTP/1.1*. [S.l.], 1999.
- FIELDING, R. T. Rest apis must be hypertext-driven. *Untangled musings of Roy T. Fielding*, 2008.
- FIELDING, R. T.; TAYLOR, R. N. *Architectural styles and the design of network-based software architectures*. [S.l.]: University of California, Irvine Doctoral dissertation, 2000.
- GARTNER, I. *Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015*. 2015. Disponível em: <<http://www.gartner.com/newsroom/id/3165317>>.
- GOOGLE. *What is Angular?* 2017. Disponível em: <<https://angular.io/docs>>.
- GROUP, W. S. W. I. *Basic Geo Vocabulary*. 2017. Disponível em: <<https://www.w3.org/2003/01/geo/>>.
- GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, Elsevier, v. 43, n. 5-6, p. 907–928, 1995.
- GUARINO, N. et al. Formal ontology and information systems. In: *Proceedings of FOIS*. [S.l.: s.n.], 1998. v. 98, n. 1998, p. 81–97.
- GUINARD, D.; TRIFA, V. *Building the web of things: with examples in node.js and raspberry pi*. Manning Publications Co., 2016.
- GUINARD, D.; TRIFA, V. M.; WILDE, E. *Architecting a mashable open world wide web of things*. [S.l.]: ETH, Department of Computer Science, 2010.
- HENRIQUE, P. B.; IVAN, G. J. J.; RADI, M. T. *Web semântica*. 2013.
- HEWLETT-PACKARD; INTERNATIONALIZATION, E.; CUPERTINO. *What is Unicode?* 1991. Disponível em: <www.unicode.org/standard/translations/portuguese.html>.
- ISOTANI, S.; BITTENCOURT, I. I. *Dados Abertos Conectados: Em busca da Web do Conhecimento*. [S.l.]: Novatec Editora, 2015.
- ITU. Next Generation Networks – Frameworks and functional architecture models . In: INTERNATIONAL TELECOMMUNICATION UNION. *GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS*. [S.l.], 2012. (SERIES Y, Y.2060).
- KAZUO, K. et al. *Web of Things (WoT) Architecture*. [S.l.], 2017.

- LIMA, J. C. de; CARVALHO, C. L. de. *Uma Visão da Web Semântica*. [S.l.], 2004.
- LIMA, V. M. A. *Terminologia, comunicação e representação documentária*. Tese (Doutorado) — Universidade de São Paulo, 1998.
- MAYER, S. Deployment support for an infrastructure for web-enabled devices. 2010.
- MICROSOFT. Utilizando o Network Monitor 3 para Entender o Tráfego HTTP. In: . [s.n.], 2007. Disponível em: <blogs.technet.microsoft.com/latam/2007/09/10/utilizando-o-network-monitor-3-para-entender-o-trfego-http/>.
- MIZOGUCHI, R. Part 3: Advanced course of ontological engineering. *New Generation Computing*, Springer, v. 22, n. 2, p. 193–220, 2004.
- PERERA, C. et al. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 1, p. 414–454, 2014.
- RICHARDSON, L.; RUBY, S. *RESTful web services*. [S.l.]: "O'Reilly Media, Inc.", 2008.
- ROB, A. et al. *Semantic Sensor Network Ontology*. [S.l.], 2017.
- SANTOS, B. P. et al. Internet das coisas: da teoria à prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2016.
- SEGUNDO, J. E. S. Representação iterativa: um modelo para repositórios digitais. Universidade Estadual Paulista (UNESP), 2010.
- STANFORD, P. *Why Protégé*. 2017. Disponível em: <<https://protege.stanford.edu>>.
- STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge engineering: principles and methods. *Data & knowledge engineering*, Elsevier, v. 25, n. 1-2, p. 161–197, 1998.
- TAN, N. W. L. Future internet: The Internet of Things. In: *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*. [S.l.]: IEEE, 2010. ISBN 978-1-4244-6542-2.
- TOMCAT, A. *Apache Tomcat*. 2017. Disponível em: <<http://tomcat.apache.org>>.
- TOPCU, F. Context modeling and reasoning techniques. In: *SNET Seminar in the ST*. [S.l.: s.n.], 2011.
- VERMESAN, O. et al. Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends*, Rivers Publishers Aalborg, v. 1, p. 9–52, 2011.
- ZHENG, L. et al. Technologies, applications, and governance in the internet of things. *Internet of Things—Global Technological and Societal Trends from Smart Environments and Spaces to Green ICT*, River Publishers, 2011.