

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE  
TECNOLOGIAS

ANDRÉ LUIS BONI DÉO

PROPOSTA DE UM MODELO DE REFERÊNCIA OPEN  
SOURCE PARA IOT – FOCO EM PME

CAMPINAS

2018

ANDRÉ LUIS BONI DÉO

PROPOSTA DE UM MODELO DE REFERÊNCIA OPEN  
SOURCE PARA IOT – FOCO EM PME

Dissertação apresentada ao Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Omar Carvalho Branquinho.

PUC-CAMPINAS

2018

Ficha Catalográfica Elaborada por Marluce Barbosa  
Sistema de Bibliotecas e Informação - SBI – PUC-Campinas

t621.3851 Déo, André Luis Boni.  
D418p Proposta de um modelo de referência Open Source para IoT: foco em  
PME / André Luis Boni Déo. - Campinas: PUC-Campinas, 2018.  
149f.

Orientador: Omar Carvalho Branquinho  
Dissertação (mestrado) – Pontifícia Universidade Católica de Campinas,  
Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação  
em Engenharia Elétrica.  
Inclui anexo e bibliografia.

1. Redes de sensores sem fio. 2. Pequenas e médias empresas. 3.  
Engenharia de software - Gerência. 4. Internet - Programas de computador.  
I. Branquinho, Omar Carvalho. II. Pontifícia Universidade Católica de Cam-  
pinas. Centro de Ciências Exatas, Ambientais e de Tecnologias. Pós-Gra-  
duação em Engenharia Elétrica. III. Título.

22.ed. CDD – t621.3851

**ANDRÉ LUIS BONI DÉO**

**“PROPOSTA DE UM MODELO DE REFERÊNCIA OPEN SOURCE PARA IOT”**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito parcial para obtenção do título de Mestre em Gestão de Redes de Telecomunicações.

Área de Concentração: Engenharia Elétrica.  
Orientador: Prof. Dr. Omar Carvalho Branquinho

Dissertação defendida e aprovada em 11 de dezembro de 2018 pela Comissão Examinadora constituída dos seguintes professores:



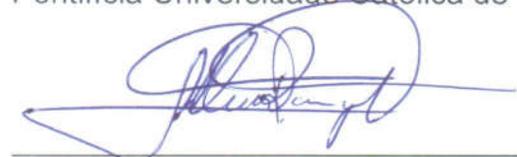
---

Prof. Dr. Omar Carvalho Branquinho  
Orientador da Dissertação e Presidente da Comissão Examinadora  
Pontifícia Universidade Católica de Campinas



---

Prof. Dr. Frank Herman Behrens  
Pontifícia Universidade Católica de Campinas



---

Prof. Dr. Rangel Arthur  
Universidade Estadual de Campinas

Dedico este trabalho a minha esposa, meu porto seguro em todos os momentos da minha vida.

## AGRADECIMENTOS

Primeiramente a Deus,

A Ele todo louvor e toda glória.

A minha família,

Em especial a minha esposa Gislaine, pelo apoio incondicional e pelo incentivo nos momentos mais difíceis, você sempre foi minha maior incentivadora.

Ao meus amigos e companheiros de trabalho,

Adriano R. Paganotto e Marcos de A. Corá, obrigado por tudo.

Ao Prof. Dr. Omar Carvalho Branquinho,

Incentivador, guia e amigo nesta parte da jornada.

Ao colega e amigo Rogério Nunes de Freitas,

Por todo apoio, dicas e cumplicidade nesta trajetória.

À Pontifícia Universidade Católica de Campinas,

Pela concessão da bolsa no Programa de Mestrado, sem o qual este trabalho não poderia ter sido concretizado.

Aos membros do Grupo de Trabalho no qual estou inserido,

Carlinho Caixeta, Eloisa Matthiesen, Guilherme Lopes, Pedro Chaves, Raphael Montali, Rogerio Nunes de Freitas, eu não tenho como expressar o quanto aprendi com vocês.

Aos colegas e companheiros da turma,

Pelo apoio e toda solidariedade.

Aos professores do programa de Mestrado,

Vocês foram verdadeiros mestres.

*“Se dois homens vêm andando por uma estrada, cada um com um pão, e, ao se encontrarem, trocarem os pães, cada um vai embora com um.*

*Se dois homens vêm andando por uma estrada, cada um com uma ideia, e, ao se encontrarem, trocarem as ideias, cada um vai embora com duas.”*

(Provérbio Chinês)

# RESUMO

DÉO, André Luis Boni. Proposta de Um Modelo de Referência *Open Source* para IoT. 2018. 149 pgs. Dissertação (Mestrado Profissional em Gestão de Redes de Telecomunicações) - Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Campinas, 2018.

A proposta desse trabalho é apresentar um modelo de referência para a criação de aplicações de Internet das Coisas - IoT do inglês *Internet of Things*, considerando todos os níveis necessários para a criação desse tipo de solução. O foco principal é criar uma arquitetura utilizando apenas soluções *Open Source*, balizada pelo modelo de referência. Essa abordagem visa atender as pequenas e médias empresas, que não possuem recursos para soluções custosas, que serão comercializadas e suportadas por grandes empresas. Foram analisados diversos paradigmas sobre a adoção de soluções de IoT, dentre os quais, destaca-se a falta de clareza do mercado de grandes fabricantes, ao indicar que IoT é uma solução *plug n' play*, não sendo a realidade encontrada nos estudos de casos realizados. Buscou-se ao longo do trabalho combater essa visão segmentada, ao mesmo tempo em que se objetivou demonstrar que para obter sucesso no desenvolvimento de uma plataforma, é preciso entender que as soluções de IoT, necessitam, obrigatoriamente de gerência, pois nenhuma solução é perene, infalível ou a prova de ação humana. Foram implementados estudo de casos, que permitiram a avaliação e testes do modelo de referência, chegando em uma estrutura de seis níveis necessários para a criação de uma aplicação de IoT. Os resultados demonstraram a factibilidade do modelo na implementação, em diferentes cenários.

**Termos de indexação:** Modelo de Referência IoT, Internet das Coisas, Redes de Sensores Sem Fio, Gerência, IoT

# ABSTRACT

DÉO, André Luis Boni. Proposal of an Open Source Reference Model for IoT. 2018. 149 pp. Dissertation (Master in Telecommunication Network Management) - Pontificia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Campinas, 2018.

The purpose of this work is to present a reference model for the creation of Internet of Things applications, considering all the levels necessary for the creation of this type of solution. The main focus is to create an architecture using only Open Source solutions, benchmarked by the reference model. The approach aims to serve small and medium-sized enterprises that do not have the resources to invest in costly solutions, that will be marketed and supported by large companies. Several paradigms about the adoption of IoT solutions were analyzed, among which the lack of clarity of large manufacturers stands out, of which IoT is a plug n 'play solution, not being the reality found in the case studies carried out. It was sought throughout the work to combat this segmented view, at the same time that it was aimed to demonstrate that to succeed in the development of a platform, it is necessary to understand that IoT solutions, necessarily, need management, since no solution is perennial, infallible, or the proof of human action. Case studies were implemented, which allowed the evaluation and testing of the reference model, arriving at a six-level structure necessary for the creation of a IoT application. The results demonstrated the feasibility of the model in the implementation in different scenarios.

Keywords: IoT Reference Model, Internet of Things, Wireless Sensor Networks, Management, IoT

## LISTA DE FIGURAS

Figura 1 - Humanos transformam Dados em Sabedoria .....	25
Figura 2 - Modelo de Referência CISCO.....	26
Figura 3 - Modelo de Referência CISCO Atualizado.....	27
Figura 4 - Operador de RSSF .....	28
Figura 5 - Proposta de uma Arquitetura para Gerenciamento de RSSF baseada em três dimensões.....	30
Figura 6 - modelo de arquitetura funcional para aplicações de IoT.....	32
Figura 7 - Integração de plataformas verticais da cidade na Plataforma de Cidades Inteligentes.....	35
Figura 8 - Pontos de referência para interoperabilidade de SCP com provedores externos .....	36
Figura 9 – Modelo de Referência de IoT – ITU-T Y.2060.....	38
Figura 10 – Papéis e Relacionamentos em Negócios de IoT.....	39
Figura 11 - Introdução de um Proxy Manager para o Gerenciamento de uma RSSF .....	43
Figura 12 - Controles de Gerenciamento Possíveis em uma RSSF .....	44
Figura 13 - Distribuição das funções de Gerência .....	45
Figura 14 - O papel do negócio na solução de IoT .....	47
Figura 15 - Relação entre Clientes, Regras de Negócio e Operador da RSSF.....	49
Figura 16 - Visão de Dados e Visão de Rede .....	52
Figura 17 - Modelo de Referência (Parcial) com Tecnologias.....	53
Figura 18 - Protocolos Suportados por Plataformas de IoT .....	54
Figura 19 - Modelo de Referência .....	59
Figura 20 - Visão Funcional de um Modelo de Referência de IoT .....	78
Figura 21 – Exemplo de Outlier.....	80
Figura 22 - Precisão e Acurácia .....	81
Figura 23 - Exemplo de Condicionamento .....	82
Figura 24 – Sensor para coleta de Temperatura e Umidade com conectores do tipo Grove .....	83
Figura 25 - Conversão de Variação de Tensão para Sinais Digitais .....	84
Figura 26 – Medição de Nível de Umidade do Solo com Vegetronix .....	84
Figura 27 - Arduino Uno .....	86
Figura 28 - Módulo de rádio BE900.....	87
Figura 29 - Raspberry Pi3 .....	90
Figura 30 - Visão de Dados e Visão de Rede .....	98
Figura 31 - Tamanho da Base de Dados do Zabbix (GB) .....	100
Figura 32 - Shield Arduino Mega.....	105
Figura 33 – Modos de execução do script.....	106
Figura 34 - meduZa com o LED aceso.....	107
Figura 35 - Resfriamento Evaporativo.....	108
Figura 36 - Interface do Negócio .....	109
Figura 37 - Interface do Especialista .....	109
Figura 38 - Interface do Operador de RSSF .....	110
Figura 39 - Abstração do Conhecimento do Especialista.....	111

Figura 40 - Raspberry Pi 3 com a Base do BE900.....	112
Figura 41 - Experimento Resfriamento por Evaporação - Figura 01 .....	113
Figura 42 - Experimento Resfriamento por Evaporação – Figura 02 .....	114
Figura 43 - Disposição dos elementos no Campo de Futebol - PUC Campinas ....	116
Figura 44 - Dados Coleta Caso Campo de Futebol – PUC Campinas .....	117
Figura 45 - Mapa com Triggers .....	120
Figura 46 - Ícones Personalizados .....	121
Figura 47 - Menus Personalizados.....	122
Figura 48 - Mapa Planta Baixa .....	123
Figura 49 - Menu de contexto .....	124
Figura 50 - Script executado pelo Zabbix.....	124
Figura 51 - Mapa refletindo a interação com a meduZa.....	125
Figura 52 - Recurso de Geolocalização .....	126
Figura 53 - Coleta de Intensidade de Sinal .....	127
Figura 54 - Coleta de Temperatura, Ponto de Orvalho e Umidade .....	127
Figura 55 - Zabbix Sender para Android .....	129
Figura 56 - Zabbix Geo para Android .....	130

## LISTA DE TABELAS

Tabela 1 – Sensores Utilizados no Casos.....	85
Tabela 2 - Camadas do protocolo Radiuno.....	88
Tabela 3 - Classificação da RSSI.....	119

## LISTA DE ABREVIATURAS E SIGLAS

ACK	=	<i>Acknowledgement</i>
AD	=	Conversão de Analógico para Digital
ANATEL	=	Agência Nacional de Telecomunicações
API	=	<i>Application Programming Interface</i>
CoAP	=	<i>Constrained Application Protocol</i>
DIY	=	<i>Do It Yourself</i>
FCAPS	=	<i>Fault, Configuration, Accounting, Performance e Security</i>
FEEC	=	Faculdade de Engenharia Elétrica e de Computação
GPS	=	<i>Global Positioning System</i>
HTML	=	<i>HyperText Markup Language</i>
HTTP	=	<i>Hypertext Transfer Protocol</i>
IA	=	Inteligência Artificial
IBSG	=	<i>Internet Business Solutions Group</i>
IoT PM	=	<i>IoT Proxy Manager</i>
IoT	=	<i>Internet of Things</i>
ISM	=	<i>Industrial, Scientific and Medical</i>
ISO	=	<i>International Organization for Standardization</i>
ITU-T	=	<i>ITU Telecommunication Standardization Sector</i>
JSON	=	<i>JavaScript Object Notation</i>
LED	=	<i>Light Emitting Diode</i>
LORA	=	<i>Long Range</i>
LR-WPAN	=	<i>Lower Rate Wireless Personal Area Network</i>
MAC	=	<i>Media Access Control</i>
MQTT	=	<i>Message Queuing Telemetry Transport</i>
NMS	=	<i>Network Management System</i>
NS	=	Nós Sensores
NTP	=	<i>Network Time Protocol</i>
OSI	=	<i>Open System Interconnection</i>
PIB	=	Produto Interno Bruto
PM	=	<i>Proxy Manager</i>
PME	=	Pequenas e Médias Empresas

QoS	=	<i>Quality of Service</i>
REST	=	<i>Representational State Transfer</i>
SCADA	=	<i>Supervisory Control and Data Acquisition</i>
SCP	=	<i>Smart City Platform</i>
SDN	=	<i>Software Defined Networking</i>
SLA	=	<i>Service Level Agreement</i>
SO	=	<i>Sistema Operacional</i>
SOHO	=	<i>Small Office and Home Office</i>
TMN	=	<i>Telecommunications Management Network</i>
VPS	=	<i>Values Per Second</i>

# SUMÁRIO

<b>RESUMO .....</b>	<b>8</b>
<b>ABSTRACT .....</b>	<b>9</b>
<b>LISTA DE FIGURAS .....</b>	<b>10</b>
<b>LISTA DE TABELAS .....</b>	<b>12</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>13</b>
<b>SUMÁRIO .....</b>	<b>15</b>
<b>1. INTRODUÇÃO .....</b>	<b>19</b>
1.1. Justificativa .....	21
1.2. Motivação.....	22
1.3. Objetivos.....	23
1.4. Organização do Trabalho .....	24
<b>2. REVISÃO BIBLIOGRÁFICA .....</b>	<b>25</b>
2.1. Da Coleta de Dados à Tomada de Decisões .....	25
2.2. Operador da Rede de IoT.....	27
2.3. Gerência de Redes.....	28
2.3.1. Gerência de Redes Clássica .....	28
2.3.2. Proposta de uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio baseada em três dimensões .....	29
2.3.3. Recomendações do Setor de Normatização das Telecomunicações.....	31
2.3.4. Soluções para coleta e apresentação de dados.....	41
2.4. Ferramentas de Gerência .....	41
2.4.1. Sistemas Supervisórios .....	41
2.4.2. <i>Proxy Manager</i> .....	43
<b>3. REQUISITOS DESEJÁVEIS EM UM MODELO DE REFERÊNCIA PARA IOT ....</b>	<b>46</b>
3.1. Requisitos de Usuário .....	46
3.2. O papel do negócio na solução de IoT.....	47
3.3. Coleta de Dados .....	48
3.4. Critérios para o Tratamento e Controle dos Dados .....	49
3.5. Definição de Limites e Execução de Ações.....	50
3.6. Operador da Rede de Sensor Sem Fio .....	51
3.7. Interfaces de Comunicação.....	52
3.8. Armazenamento Local e na Nuvem.....	54

<b>3.9. Interação com o Usuário .....</b>	<b>55</b>
<b>3.10. Adaptação para Múltiplos Cenários .....</b>	<b>56</b>
<b>4. PROPOSTA DE UM MODELO DE REFERÊNCIA DE IOT .....</b>	<b>57</b>
<b>4.1. Arquitetura .....</b>	<b>58</b>
<b>4.2. Visão Conceitual de um Modelo de Referência de IoT .....</b>	<b>60</b>
4.2.1. Nível 0 – Nível das Coisas.....	61
4.2.2. Nível 1 – Nível de dispositivo, condicionamento e controle .....	61
4.2.3. Nível 2 – Nível de conectividade .....	62
4.2.4. Nível 3 – Nível de borda .....	63
4.2.5. Nível 4 – Nível de armazenamento .....	65
4.2.6. Nível 5 – Nível de abstração.....	65
4.2.7. Nível 6 – Nível de apresentação.....	66
4.2.8. Gerência e segurança .....	67
<b>4.3. Exemplo de um negócio hipotético .....</b>	<b>68</b>
4.3.1. Nível 0 – Nível das Coisas – Exemplo de Aplicação .....	69
4.3.2. Nível 1 – Nível de dispositivo, condicionamento e controle – Exemplo de Aplicação .....	69
4.3.3. Nível 2 – Nível de conectividade – Exemplo de Aplicação .....	70
4.3.4. Nível 3 – Nível de borda – Exemplo de Aplicação.....	71
4.3.5. Nível 4 – Nível de armazenamento – Exemplo de Aplicação .....	71
4.3.6. Nível 5 – Nível de abstração – Exemplo de Aplicação .....	72
4.3.7. Nível 6 – Nível de apresentação – Exemplo de Aplicação .....	72
4.3.8. Gerência e segurança – Exemplo de Aplicação .....	73
<b>5. IMPLEMENTAÇÃO DE UMA SOLUÇÃO OPEN SOURCE BASEADA NO MODELO DE REFERÊNCIA .....</b>	<b>74</b>
<b>5.1. Estudos de Casos .....</b>	<b>74</b>
5.1.1. Caso Fazenda Cafeeira.....	74
5.1.2. Caso Campo de Futebol – PUC Campinas .....	75
5.1.3. Caso Fila do Restaurante Universitário – UNICAMP .....	76
5.1.4. Caso Monitoramento Ambiental para Condições de Emergência .....	76
5.1.5. Caso Protocolo repetidor para estender a cobertura de sinal.....	77
5.1.6. Caso Segurança.....	77
<b>5.2. Arquitetura Implementada.....</b>	<b>78</b>
5.2.1. Nível 0 – Nível das Coisas.....	79
5.2.2. Nível 1 – Nível de dispositivo, condicionamento e controle .....	80
5.2.3. Nível 2 – Nível de conectividade .....	85
5.2.4. Nível 3 – Nível de borda .....	89
5.2.5. Nível 4 – Nível de armazenamento .....	99

5.2.6.	Nível 5 – Nível de abstração.....	102
5.2.7.	Nível 6 – Nível de apresentação.....	102
5.2.8.	Gerência .....	103
5.2.9.	Segurança .....	103
<b>5.3.</b>	<b>Aplicativos de Celular.....</b>	<b>104</b>
5.3.1.	Zabbix Sender para celular.....	104
5.3.2.	App de Geolocalização.....	104
<b>5.4.</b>	<b>Caso Atuador.....</b>	<b>105</b>
<b>5.5.</b>	<b>Caso Implementação Prática do Modelo de Referência .....</b>	<b>107</b>
5.5.1.	Nível 7 – Negócio .....	108
5.5.2.	Nível 6 – Nível de apresentação.....	108
5.5.3.	Nível 5 – Nível de abstração.....	110
5.5.4.	Nível 4 – Nível de armazenamento .....	111
5.5.5.	Nível 3 – Nível de Borda.....	112
5.5.6.	Nível 2 – Nível de Conectividade.....	112
5.5.7.	Nível 1 – Nível de Dispositivo, Condicionamento e Controle.....	113
5.5.8.	Nível 0 – Nível das Coisas.....	114
<b>6.</b>	<b>RESULTADOS .....</b>	<b>115</b>
<b>6.1.</b>	<b>Nível 1 – Nível de dispositivo, condicionamento e controle .....</b>	<b>115</b>
<b>6.2.</b>	<b>Nível 2 – Nível de conectividade .....</b>	<b>115</b>
<b>6.3.</b>	<b>Nível 3 – Nível de borda .....</b>	<b>116</b>
<b>6.4.</b>	<b>Nível 4 – Nível de armazenamento.....</b>	<b>117</b>
<b>6.5.</b>	<b>Nível 5 – Nível de abstração .....</b>	<b>118</b>
<b>6.6.</b>	<b>Nível 6 – Nível de apresentação.....</b>	<b>118</b>
6.6.1.	Alertas Personalizados .....	119
6.6.2.	Ícones Personalizados.....	120
6.6.3.	Menus Personalizados .....	121
6.6.4.	Integração com Atuadores.....	122
6.6.5.	Integração de Plugins .....	125
6.6.6.	Demais Personalizações .....	126
6.6.7.	Aplicativos de Celular .....	128
<b>7.</b>	<b>CONCLUSÃO.....</b>	<b>131</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>133</b>
	<b>APÊNDICES .....</b>	<b>144</b>
	<b>Apêndice A – Questionário .....</b>	<b>144</b>
	<b>Apêndice B – Arquivos meduZa .....</b>	<b>146</b>
	<b>Apêndice C – Arquivos Arduino Emulando Radiuino .....</b>	<b>147</b>

<b>Apêndice D – Instruções para Personalização do Idioma da Interface Gráfica do Zabbix .....</b>	<b>148</b>
<b>ANEXOS .....</b>	<b>149</b>
<b>Anexo A – <i>Script</i> Gerar XML para Importar Imagens em Massa no Zabbix.....</b>	<b>149</b>

## 1. INTRODUÇÃO

A sociedade brasileira já se tornou conectada, as tecnologias de comunicação sem fio, como redes *wireless* e telefonia celular, fazem parte do dia-a-dia da população. De acordo com dados da Agência Nacional de Telecomunicações (ANATEL, 2018) o Brasil registrou 236,2 milhões de linhas móveis em operação em janeiro de 2018, mais de uma linha celular por habitante (IBGE, 2018). Além disso, de acordo com os dados da pesquisa TIC Domicílios, em 2017 o Brasil possuía 120,7 milhões de usuários conectados (CETIC, 2017).

Diante deste cenário, é possível inferir que o ser humano já está totalmente habituado ao fato de estar conectado, o próximo passo é conectar as coisas. De fato, esse movimento já está acontecendo, por meio de tecnologias como, por exemplo, IoT (*Internet of Things*), Indústria 4.0, SDN (*Software-Defined Networking*), Cidades e Campus Inteligentes.

Focando apenas no mercado de IoT, observa-se que ele continua em crescimento acelerado em todo o mundo, e o Brasil não é exceção frente a esse fenômeno. Segundo estudo da Frost & Sullivan (IDGNOW, 2017), o Brasil movimentou US\$ 1,35 bilhão no ano de 2016 e deve alcançar receitas de US\$ 3,29 bilhões em cinco anos.

Os grandes *players* do mercado estão desenvolvendo soluções para atender essas demandas. A EMERSON, empresa multinacional americana listada na Fortune 500, fabrica produtos e fornece serviços de engenharia para uma ampla gama de mercados industriais, comerciais e de consumo e já possui soluções voltadas para indústrias de porte extremamente grandes e atuantes em mercados específicos, como sua solução Smart Wireless Solutions (EMERSON, 2017) aplicada a refinarias e petrolíferas.

A CISCO, conhecida pela fabricação de equipamentos de conectividade de rede, apresenta soluções de IoT (CISCO, 2017) para mercados como Manufatura, Energia e Transporte, enquanto que a IBM aposta em sua plataforma de serviços cognitivos para negócios, o Watson (IBM, 2017).

Todas estas soluções, irão alavancar o desenvolvimento de aplicações e colaborar para a adoção de IoT nas grandes empresas. O problema é que isso tende a criar uma exclusão tecnológica em uma faixa da sociedade.

De um lado, existem as empresas familiares, artesanais e/ou focadas em mercados muito específicos, que não possuem interesse e/ou visão suficiente para

adoção de novas tecnologias. E do outro lado, há os grandes mercados que adotaram as soluções fornecidas pelas grandes empresas.

Porém, entre estas duas realidades, existirá um vale de excluídos tecnológicos, as Pequenas e Médias Empresas (PME) que não possuem condições de arcar com custos de implementação, manutenção e atualização, fornecidas pelos grandes *players* do mercado.

No cenário do Brasil apenas as Micro e Pequenas Empresas representam 98,5% do total de empresas, movimentam 27% do Produto Interno Bruto (PIB) e geram mais da metade dos empregos (CARTA, 2017).

Para entender e atender as demandas dessas PME, é possível realizar um paralelo com o universo dos aplicativos de dispositivo móveis. Nesse âmbito, a IoT será a plataforma, assim como o ANDROID (2018), as soluções desenvolvidas para essas empresas, serão como os aplicativos. Soluções personalizadas, adaptadas as necessidades e condições financeiras, e, o mais importante, providas por PME, se fazem necessárias nesse cenário.

Assim como os aplicativos de celular são personalizados para atender as necessidades de um grupo de indivíduos, as soluções de IoT para PME necessitam de personalização para cada ambiente. Todavia, é necessário respeitar a natureza financeira na qual essas empresas estão inseridas.

Contudo, deve-se obter um equilíbrio entre as restrições financeiras das PME e a continuidade do negócio ao se propor uma solução de IoT para uma empresa, pois o negócio da empresa será diretamente afetado pela indisponibilidade da solução. Portanto, produtos e soluções SOHO (*Small Office and Home Office*) não devem ser considerados nessas implementações.

Diferentemente de uma implantação em redes tradicionais, as soluções de IoT, podem utilizar-se de uma série de tecnologias, que envolvem comunicações sem fio, redes de sensores, softwares, comunicação via Internet, tratamento de dados, inteligência artificial (IA), apenas para citar algumas.

Para que seja possível atender as necessidades da PME, propõem-se separar a coleta de dados em duas visões distintas, a área de redes, que proverá as informações referentes a continuidade do negócio e a área de dados, responsável pelas informações de auxílio à tomada de decisão.

Essa separação é essencial para que possamos determinar os papéis dos profissionais envolvidos, a natureza dos dados coletados e o que se pretende obter com essas coletas.

Caso contrário não é possível aplicar, por exemplo, uma estratégia de SLA (*Service Level Agreement*), que basicamente é um acordo que determina os níveis de execução da rede, o que garante o funcionamento e continuidade da solução dentro de níveis aceitáveis. E tampouco prover níveis de serviço adequados às necessidades do cliente, por meio de priorização de tráfego no momento da transmissão, provendo assim condições para utilização de técnicas de QoS (*Quality of Service*).

Outro ponto a ser discutido é a apresentação dos dados, sendo o foco PME, espera-se que a solução atenda à uma vasta variedade de áreas, nesse quesito é importante que os ícones utilizados, os termos técnicos empregados, os gráficos apresentados sejam compatíveis com a área de atuação da PME.

Finalmente é proposto um modelo de referência, cujo objetivo é auxiliar os envolvidos a entenderem todos os elementos e tecnologias que precisam ser abordados ao se propor uma solução de IoT.

### **1.1. Justificativa**

Ao realizar uma análise entre as necessidades da IoT e dos seres humanos, percebe-se que o universo dos seres humanos, é extremamente restrito, se comparado ao universo das coisas.

O desenvolvimento de produtos para seres humanos segue uma padronização muito bem definida. Por exemplo, não faz sentido pensar em um teclado de computador de 3 centímetros, tampouco um de 3 metros. Os seres humanos, quando analisados estatisticamente possuem certo grau de uniformidade.

Porém a natureza das coisas é completamente diferente, é possível que duas máquinas troquem informações entre si, de maneira precisa, sem a existência de teclados, monitores ou botões de acionamento, necessitando apenas de uma interface de comunicação entre elas, muitas vezes, uma interface serial.

Por essa ótica, uma rede de computadores pretende atender as necessidades dos usuários, que assim como os produtos, são padronizadas, enquanto que uma rede de IoT, a princípio não possui uma padronização, no sentido de objetivo da rede.

Obviamente existem redes especializadas e padronizadas para atender setores específicos, onde o valor do produto, petróleo, por exemplo, é muito superior ao valor da implementação.

Para esses setores o custo da rede, será repassado ao produto, mesmo que esse custo seja extremamente alto, dificilmente impactará o custo do produto. Entretanto, os pequenos e médios empresários precisam de soluções personalizadas para a realidade de suas áreas de atuação.

## **1.2. Motivação**

Existe uma grande quantidade de publicações acadêmicas sobre IoT. Por outro lado, há uma carência de trabalhos acadêmicos e experimentais focados em PME, principalmente quando o objetivo for atender as necessidades dessas PME de forma profissional.

Ao se observar as soluções de IoT propostas pelo mercado, percebe-se que o cerne tem sido a divulgação de ideias a serem implementadas, mercados que podem surgir, criação e transformação de produtos.

Porém, tais soluções não levam em consideração as questões de infraestrutura de rede necessárias para que esses serviços e soluções sejam implementadas, mantidos e ampliados, de acordo com a necessidade do cliente, atingindo níveis de estabilidade e garantias de continuidade do negócio, como ocorre em qualquer rede tradicional.

É possível observar (COMPUTERWORLD, 2014) que essa é uma realidade que ocorre, pelo menos, nos últimos 4 anos.

Diante desse cenário, propõe-se a criação de uma arquitetura de soluções modulares, que possam ser customizadas com o objetivo de atender às necessidades das PME.

Esta plataforma precisa ser flexível, multissegmentada, personalizável e permitir que a IoT tenha uma interface com o mundo físico por meio das RSSF, gerência e visualização de informações na Internet.

Criar essa plataforma não é trivial, é preciso um corpo de profissionais qualificados em diversas áreas para que o objetivo possa ser alcançado.

Precisa-se de engenheiros de telecomunicações, desenvolvedores de software e administradores de redes com profundos conhecimentos em gerência de redes. Todos trabalhando de forma cooperativa e colaborativa visando atender o objetivo final.

Implementar essa solução de maneira profissional com foco nos objetivos do negócio, envolve uma grande quantidade de processos, tecnologias e conhecimentos de áreas distintas. Nesse cenário surge o Modelo de Referência.

### **1.3. Objetivos**

O objetivo desta dissertação é apresentar um Modelo de Referência, que possibilite a implementação de uma plataforma para gerenciamento e coleta de dados para PME com foco no negócio, por meio de RSSF, possibilitando realizar a gerência da rede e de dados, por meio de uma única solução.

Toda rede exige monitoramento e gerenciamento para atingir níveis confiáveis. Diante deste cenário, é proposta a utilização de uma arquitetura de soluções de RSSF e softwares, que permitem serem customizados e adaptados para se tornarem uma plataforma de IoT com foco em PME.

Devido à necessidade de atendimento de diversos nichos de mercado, personalizações e adaptações de cenários, sem criar dependência de fabricantes ou fornecedores específicos, optou-se no presente trabalho pela busca de soluções exclusivamente *Open Source*.

A vantagem das soluções *Open Source*, é que qualquer profissional capacitado pode realizar alterações e implementações naquele elemento da arquitetura, seja o desenvolvimento de um sensor ou alguma personalização na plataforma de gerência.

A principal desvantagem está no fato de nem sempre essas soluções possuem uma empresa responsável pelo suporte dos produtos e treinamento de profissionais. Escolher soluções que possuam essa característica é um ponto importante.

Outro ponto importante a ser observado, é que a capacitação dos profissionais, pode não ser necessariamente tradicional, tem crescido o número de profissionais adeptos do movimento DIY (*Do It Yourself*).

Isso pode ser positivo no sentido de que aumenta as possibilidades de profissionais disponíveis para as PME. Porém, deve ser um ponto de atenção ao contratar os serviços. É preciso a comprovação do conhecimento por parte dos profissionais envolvidos na implementação da solução.

Por fim, visa demonstrar que para atender as demandas personalizadas das PME, pode-se criar uma IoT de maneira mais simples, direta, robusta, personalizável, com tecnologia nacional, provida por PME com foco no atendimento das necessidades das PME.

#### 1.4. Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma:

- **Capítulo 2:** é apresentada uma revisão de conceitos necessários para a compreensão da proposta;
- **Capítulo 3:** apresenta os paradigmas para a adoção de IoT e a proposta do trabalho;
- **Capítulo 4:** é detalhada a arquitetura da proposta e, então, é proposto um negócio hipotético para melhor compreensão das necessidades e como elas são distribuídas na proposta;
- **Capítulo 5:** são apresentados os materiais e métodos utilizados na proposta;
- **Capítulo 6:** são apresentados os resultados obtidos e analisados;
- **Capítulo 7:** as conclusões e perspectivas para futuros trabalhos.

## 2. REVISÃO BIBLIOGRÁFICA

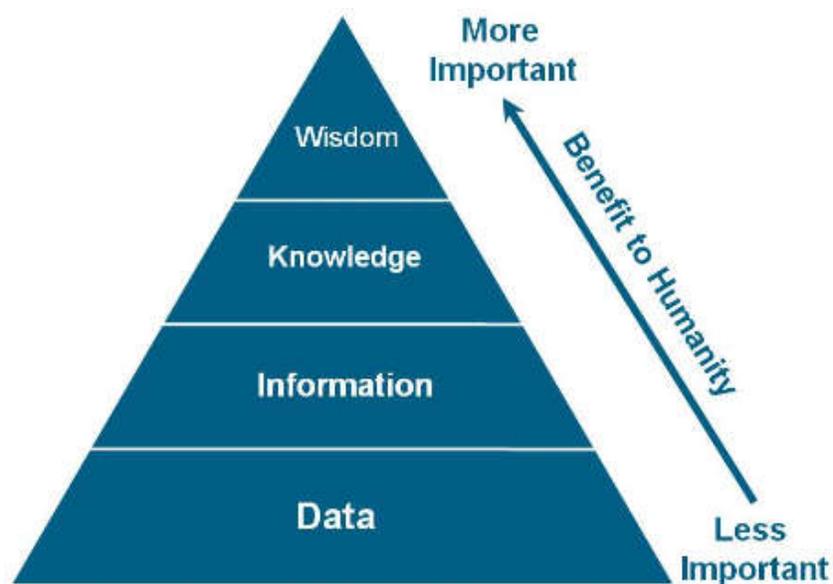
O universo de IoT é composto de uma série de elementos, como comunicação, sensores, aplicações, tratamento e visualização dos dados, para citar alguns. Neste capítulo, serão abordados temas pertinentes ao trabalho, encontrados na literatura.

### 2.1. Da Coleta de Dados à Tomada de Decisões

O Cisco Internet Business Solutions Group (IBSG) é uma consultoria global da empresa, que engloba mais de 200 especialistas de indústrias e estrategistas de negócios que possuem profunda experiência em 10 setores industriais globais (CISCO, 2018a).

Em 2011 foi publicado um documento (*White Paper*) onde o autor expõe a visão de que os humanos evoluem porque eles se comunicam e compartilham informações (EVANS, 2011, p. 6), e que esse conceito pode ser melhor compreendido analisando como os humanos processam os dados, partindo do que é menos importante e mais abundante, os dados (*Data*), até chegar ao que é mais importante e escasso, a sabedoria (*Wisdom*), conforme Figura 1.

Figura 1 - Humanos transformam Dados em Sabedoria



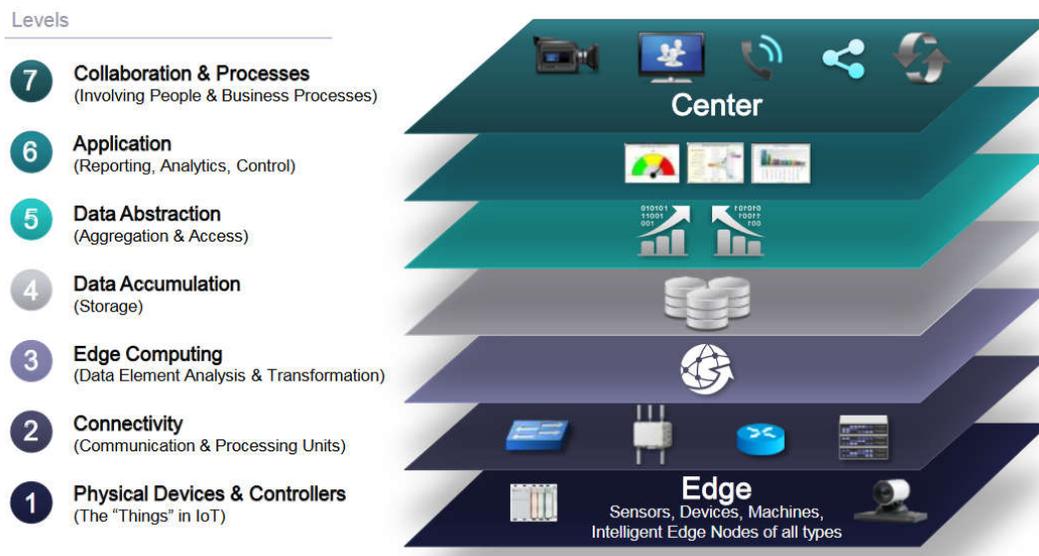
Fonte: EVANS, 2011

Segundo análise do autor, os dados são a matéria-prima processada em informações. Dados individuais por si só não são muito úteis, mas quando armazenados em grandes volumes podem identificar tendências e padrões, ou seja, informação. Estas e outras fontes de informação se reúnem para formar conhecimento.

No sentido mais simples, o conhecimento é uma informação da qual alguém está ciente. Sabedoria é a junção do conhecimento mais experiência. Enquanto o conhecimento muda com o tempo, a sabedoria é atemporal. E tudo começou com a aquisição de dados.

Analisando o modelo de referência (CISCO, 2014) proposto no Internet of Things World Forum (IOTWF, 2014), de 2014, Figura 2, observa-se a mesma linha de raciocínio apresentado por EVANS.

Figura 2 - Modelo de Referência CISCO



Fonte: CISCO, 2014

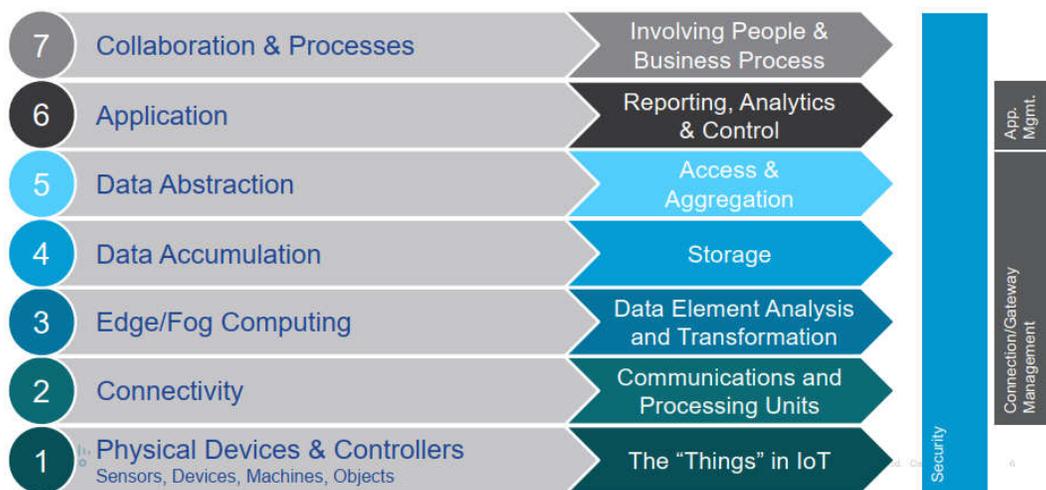
O processo é dividido em sete camadas, a primeira camada, é a de coleta de dados brutos, ou seja, a matéria-prima, que auxiliam as camadas superiores na identificação de tendências e padrões.

A medida que os dados são processados e entregues às camadas superiores, eles vão se tornando informações que atendem as necessidades de determinado processo do modelo, até chegarem à última camada, onde

podem se tornar “sabedoria”, auxiliando pessoas e negócios nas tomadas de decisões.

Um ponto importante a ser destacado, é que o modelo foi apresentado em 2014, porém continua válido e sendo utilizado em apresentações oficiais da CISCO (BOLAND, 2017), sofrendo é claro atualizações, como por exemplo, abordagem de questões como gerência e segurança, que podem ser observados na Figura 3.

Figura 3 - Modelo de Referência CISCO Atualizado



Fonte: BOLAND, 2017

## 2.2. Operador da Rede de IoT

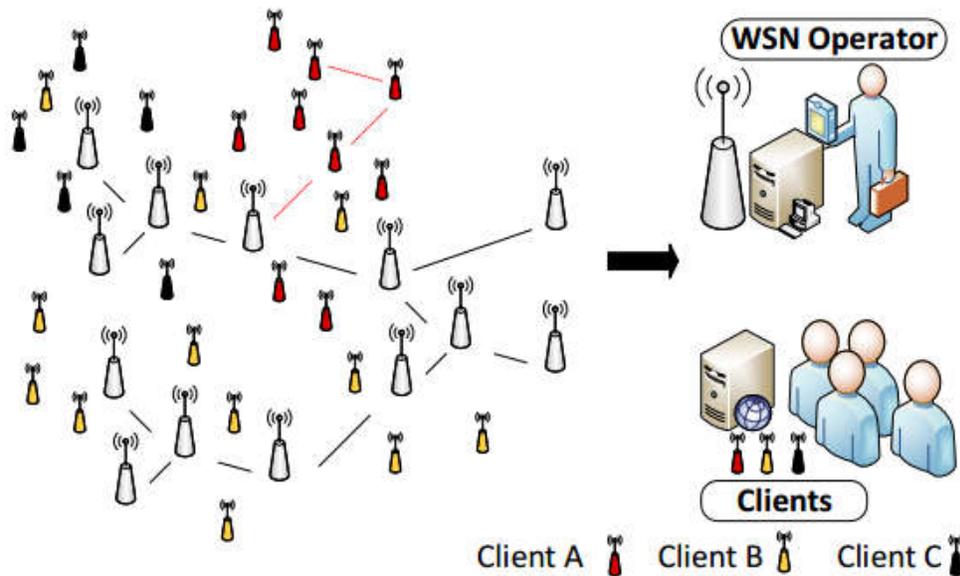
Gaillard et al. (GAILLARD, 2014) defendem a necessidade da existência de um operador de RSSF. Apesar desse conceito não ser especificamente para as redes das PME, ele é absolutamente indispensável nas RSSF's.

Pode-se observar na Figura 4 a existência de uma RSSF compartilhada por diversos clientes, nenhum cliente adicional pode ser acrescentado à rede, sem a prévia análise do Operador de RSSF. O papel deste operador é:

- Realizar o levantamento das necessidades do cliente;
- Avaliar se a rede possui capacidade para suprir a demanda;
- Adequar, se necessário, a rede para ter condições de atender a demanda do cliente;

- Estabelecer o contrato de SLA.

Figura 4 - Operador de RSSF



Fonte: GAILLARD, 2014

Apesar dessa importante contribuição, o foco do trabalho é apenas na visão de rede, ignorando questões importantes, como o impacto dessa rede no negócio do cliente, e quais os objetivos do cliente ao contratar essa rede.

### 2.3. Gerência de Redes

Nessa seção é realizada uma breve análise sobre os modelos de gerência providos pela literatura.

#### 2.3.1. Gerência de Redes Clássica

O padrão internacional OSI (*Open System Interconnection*) define cinco áreas funcionais para realização da gerência da rede (STALLINGS, 1988), conhecidas pelo acrônimo FCAPS (*Fault, Configuration, Accounting, Performance e Security*):

- **Gerência de Falhas (*Fault Management*):** Deve detectar e resolver rapidamente situações que degradam o funcionamento da rede.

- **Gerência de Configuração (*Configuration Management*):** Utilizado para que se tenha controle sobre diferentes versões de elementos de hardware e software presentes na Rede.
- **Gerência de Contabilização (*Accounting Management*):** Distribuição justa dos custos da rede, fazendo apropriação e tarifação de acordo com a utilização.
- **Gerência de Desempenho (*Performance Management*):** Conjunto de funções para medir, monitorar, avaliar e relatar os níveis de desempenho alcançados pela rede garantindo um nível de serviço aceitável ao usuário.
- **Gerência de Segurança (*Security Management*):** A meta é controlar o acesso aos recursos da rede de acordo com as definições de cada empresa para evitar sabotagens e proteger os dados para que não sejam acessados por pessoas não autorizadas.

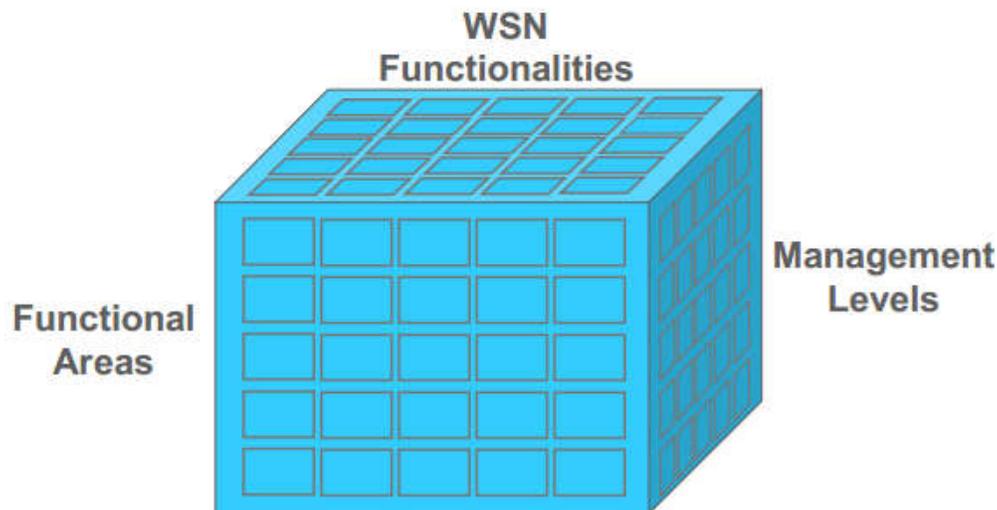
Apesar dos conceitos de gerência de redes serem amplamente conhecidos e utilizados em redes tradicionais, e serem uma recomendação do *ITU Telecommunication Standardization Sector (ITU-T)*, conforme abordado na recomendação ITU-T Y.4418, aparentemente, devido à escassez de artigos que abordem o tema, as soluções de IoT não tem focado nessas questões.

### **2.3.2. Proposta de uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio baseada em três dimensões**

A despeito dessa tendência, RUIZ (2003) propôs uma arquitetura, totalmente nova, para realizar o gerenciamento das RSSF, como pode ser observado na Figura 5, baseada em três dimensões:

- Áreas Funcionais do modelo OSI (*Open System Interconnection*)/ISO (*International Organization for Standardization*);
- Níveis de Gerenciamento do TMN (*Telecommunications Management Network*);
- Funcionalidades de RSSF, criada pela autora.

Figura 5 - Proposta de uma Arquitetura para Gerenciamento de RSSF baseada em três dimensões



Fonte: RUIZ, 2003

O obstáculo em relação ao trabalho, é que seu foco é baseado na ideia de RSSF totalmente autônomas, inteligentes e auto gerenciáveis, totalmente compatível com a visão de IoT em 2003.

Por volta desse período, o foco era a norma IEEE 802.15.4 (IEEE, 2018), que define redes de comunicação sem fio que operam com baixa taxa de transmissão de dados, conhecidas pela sigla LR-WPAN *Lower Rate Wireless Personal Area Network*).

O protocolo Zigbee (ZIGBEE, 2017) permite duas topologias distintas, estrela e peer-to-peer, porém a base para o protocolo é a peer-to-peer. Nesse período foram gerados muitos artigos explorando essa tecnologia.

Porém, ao longo do tempo observou-se que essas redes são extremamente especializadas, e que a complexidade de implementação e gerência dessas redes é muito maior do que se esperava.

Além disso, as características desse modelo de rede não condizem com as necessidades de uma Cidade Digital ou *Smart Campus*. Por exemplo, para atender a esses cenários surgiram outras tecnologias como Sigfox (SIGFOX, 2018a), LoRa e LoRaWAN (LORA, 2018).

### 2.3.3. Recomendações do Setor de Normatização das Telecomunicações

O ITU Telecommunication Standardization Sector (ITU-T), produz uma série de recomendações técnicas, sobre vários temas relacionados à tecnologia, principalmente de telecomunicações.

Nesta seção serão analisados aspectos pertinentes ao desenvolvimento do trabalho, de alguns documentos da série Y: *Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities*.

#### A. ITU-T Y.4418 - Gateway Functional Architecture for Internet of Things Applications

Primeiro são tratados os *Gateways* clássicos, chamados de *message-forwarding gateway*, eles simplesmente fazem a conversão de duas tecnologias de comunicação, por exemplo, de uma aplicação que utiliza o protocolo HTTP (*hypertext transfer protocol*) e dispositivos que são gerenciados pelo protocolo CoAP (*constrained application protocol*). Não existe lógica de aplicação nesse modelo.

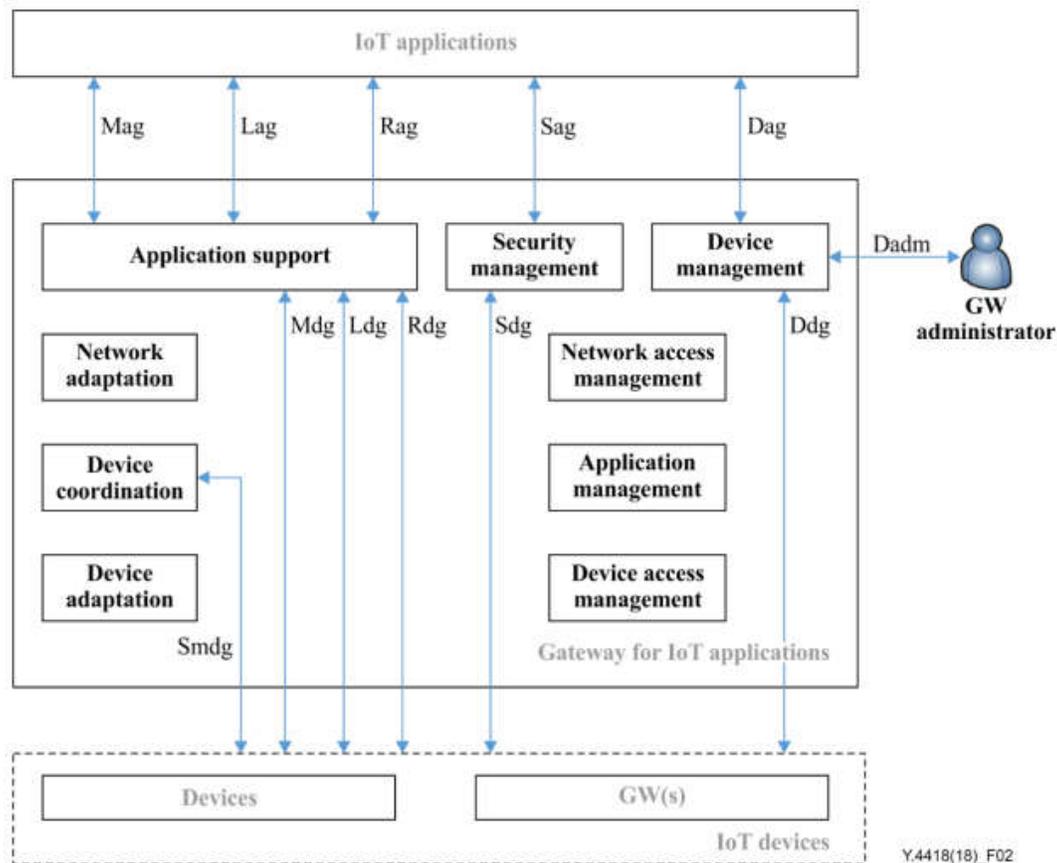
A segunda categoria de *Gateways* tratados no documento é o *edge computing*, que é projetado com uma lógica específica para atender as necessidades do usuário.

Esses equipamentos podem coletar dados dos dispositivos, analisá-los, e, ao encontrarem situações pré-determinadas, enviar uma notificação para uma aplicação de IoT específica, ou ainda, coletar os dados de um equipamento e disparar uma *trigger* em outro equipamento, que realizará determinada ação.

E finalmente, *Gateways* que são projetados para atender apenas à uma aplicação específica, eles podem realizar toda a lógica da aplicação localmente, e por isso recebem o nome de *local-processing gateway*.

O documento apresenta um modelo de arquitetura funcional para aplicações de IoT, apresentado na Figura 6, que contempla os três tipos de *Gateways* citados anteriormente.

Figura 6 - modelo de arquitetura funcional para aplicações de IoT



Fonte: ITU-T, 2018a

#### Pontos de referência entre o gateway e as aplicações de IoT:

**Mag:** Suporte de funções de encaminhamento de mensagens;

**Lag:** Suporte de funções de processamento local;

**Rag:** Suporte de funções de abertura de recursos;

**Sag:** Suporte de funções de gerenciamento de segurança;

**Dag:** Suporte de funções de gerenciamento de dispositivos.

#### Pontos de referência entre o gateway e os dispositivos de IoT:

**Smdg:** Suporte de funções relacionadas à descoberta de serviço;

**Mdg:** Suporte de funções de encaminhamento de mensagens;

**Ldg:** Suporte de funções de processamento local;

**Rdg:** Suporte de funções de abertura de recursos;

**Sdg:** Suporte de funções de gerenciamento de segurança;

**Ddg:** Suporte de funções de gerenciamento de dispositivos.

**Ponto de referência entre o *gateway* e o administrador do *gateway*:**

**Dadm:** Suporte de funções de gerenciamento de dispositivos.

Dos elementos apresentados no modelo, serão destacados a “Entidade funcional de suporte de aplicativo” e a “Entidade funcional de gerenciamento de dispositivos”, pois abordam pontos importantes para compreensão dos papéis de um *Gateway* de IoT.

#### **Entidade funcional de suporte de aplicativo**

A entidade funcional de suporte ao aplicativo é responsável pelo fornecimento de recursos, incluindo encaminhamento de mensagens, processamento local e abertura de recursos.

#### **Funcionalidades do suporte à capacidade de encaminhamento de mensagens:**

- Encaminhamento de mensagens entre os dispositivos e redes de comunicação;
- Encaminhamento de mensagens entre diferentes dispositivos ou *gateways* conectados ao *gateway*;
- Encaminhamento de mensagens de acordo com a ordem sequencial dos dados do dispositivo;
- Ajuste da ordem sequencial dos dados do dispositivo com base nas políticas;
- Uma função de tradução de protocolos entre diferentes protocolos conforme necessário ao comunicar com dispositivos e aplicativos.

### **Funcionalidades do suporte à capacidade de processamento local:**

Processamento local de aplicativos, incluindo a agregação de dados de dispositivos e aplicativos, análise de dados, transformação do formato de dados entre diferentes formatos, conforme dispositivos e aplicativos, encapsulando dados baseados em protocolos de aplicação, mediação de dados de dispositivos e aplicativos, coleta ou geração de dados relacionados à metadados, conforme aplicável, etc.

Processamento e execução local interagindo com aplicativos IoT remotos.

### **Entidade funcional de gerenciamento de dispositivos**

As funcionalidades do suporte ao gerenciamento de dispositivos incluem:

Gerenciamento de capacidades, incluindo gerenciamento de configuração, gerenciamento de desempenho, gerenciamento de falha, gerenciamento de segurança, mecanismo do protocolo de gerenciamento de dispositivo, gerenciamento contábil e exposição de serviço.

Gerenciamento de identificadores de dispositivos, incluindo criação, atualização, exclusão e recuperação de identificadores e gerenciamento de mapeamento de identificadores;

Agrupamento de dispositivos IoT com base nos atributos do dispositivo, como tipo de dispositivo e localização;

Autogerenciamento do *Gateway* e manutenção remota;

Configuração do *Gateway* de acordo com vários modos de configuração, por exemplo, configuração local e remota, configuração automática e manual e configuração dinâmica baseada em políticas.

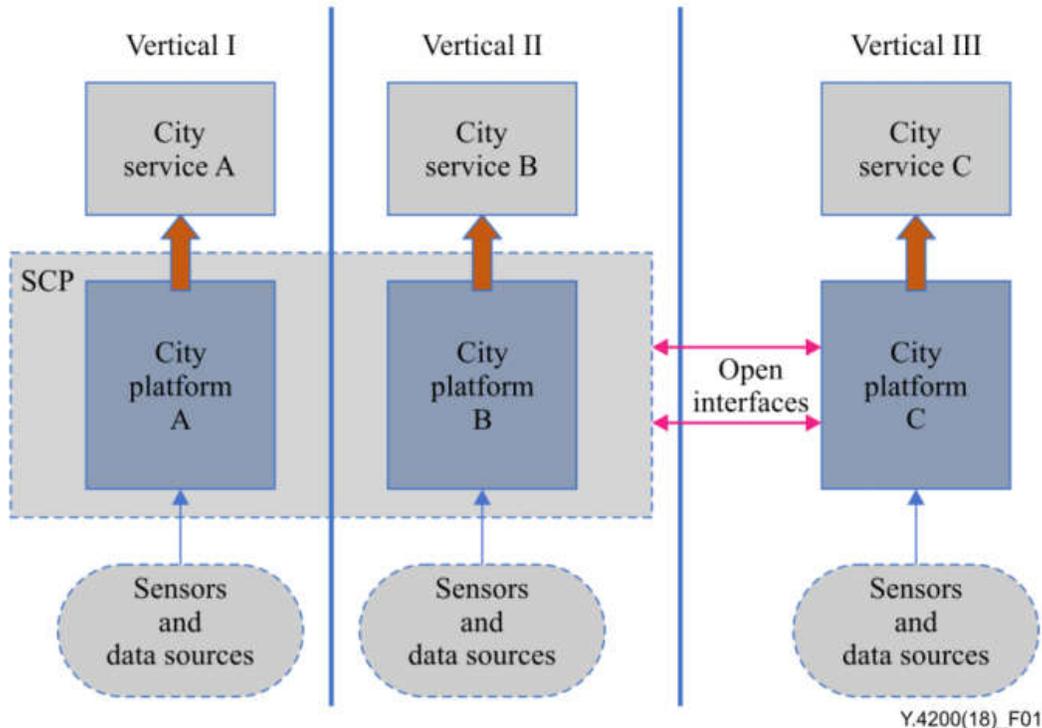
## B. ITU-T Y.4200 - Requirements for the Interoperability of Smart City Platforms

O documento destaca que toda cidade já possui uma série de plataformas para prover serviços em diversas áreas, no inglês *Smart City Platform* (SCP), e que esses serviços comumente são conhecidos como “verticais” - Figura 7.

Esses serviços são providos por plataformas IoT, SCADA (*Supervisory Control and Data Acquisition*), plataformas tradicionais - não IoT, *Big Data* e etc. O problema é que muitas vezes essas plataformas são independentes e não compartilham os recursos entre si, ou fazem isso apenas com parte dos dados.

Outro ponto importante é que nas plataformas urbanas, os sistemas proprietários são maioria e as possibilidades de oferecer interfaces, necessárias para a integração com as plataformas da cidade e as fontes de dados, são raras.

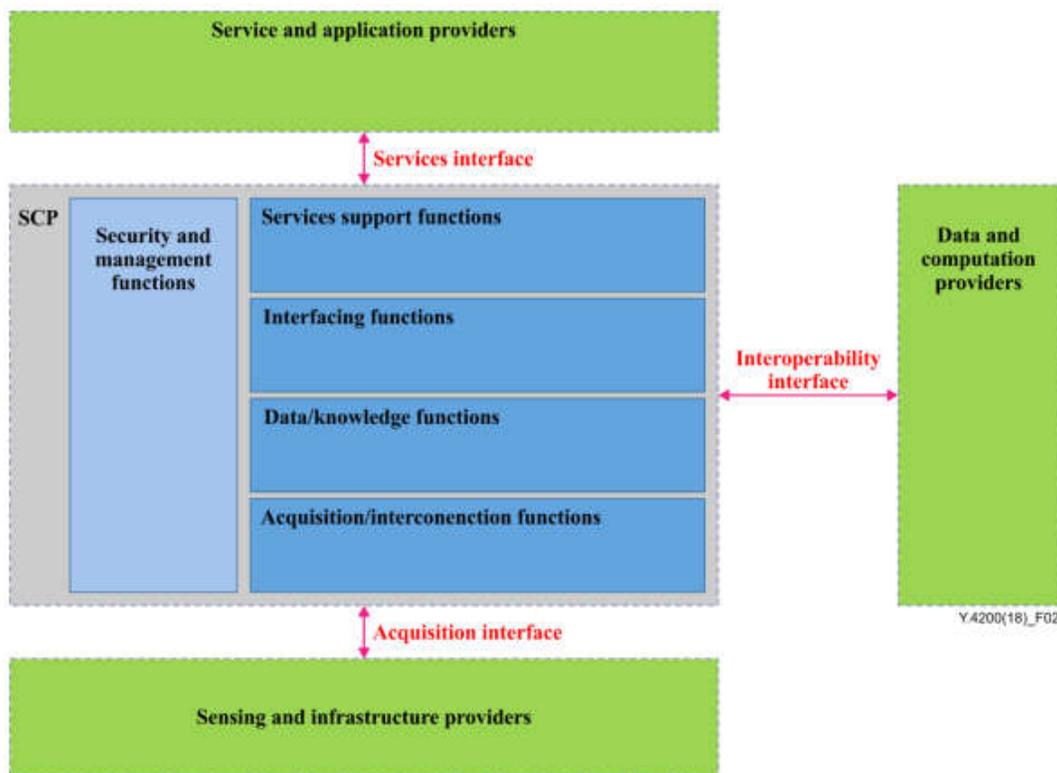
Figura 7 - Integração de plataformas verticais da cidade na Plataforma de Cidades Inteligentes



Fonte: ITU-T, 2018b

O documento descreve também os pontos de referência para interoperabilidade de SCP com provedores externos e indica outros materiais de referência para o grupo de funções apresentadas na Figura 8.

Figura 8 - Pontos de referência para interoperabilidade de SCP com provedores externos



Fonte: ITU-T, 2018b

O SCP fornece o seguinte grupo de funções conforme indicado em [ITU-T Y.4201]:

- **Funções de aquisição/interconexão:** Fornecem mecanismos de captura de dados dos sistemas de coleta (ver seção 8.2 da [ITU-T Y.4201]).
- **Funções de dados/conhecimento:** Suportam processamento de dados, agregando valor e transformando informação em conhecimento (ver cláusula 8.3 de [ITU-T Y.4201]).
- **Funções de interface:** Permitem o acesso à informação em diferentes níveis (ver cláusula 8.4 de [ITU-T Y.4201]).

- **Funções de suporte de serviço:** Coordenam todos os serviços possíveis envolvidos em cada ação desenvolvido a partir de funções de interoperabilidade (ver cláusula 8.5 da [ITU-T Y.4201]).
- **Funções de segurança e gerenciamento:** Fornecem funcionalidades horizontais, como auditorias, monitoramento e segurança (ver cláusula 8.6 da [ITU-T Y.4201]).

A Figura 8 mostra interfaces que permitem a comunicação entre as funções, indicadas pelas setas. Detalhes dessas interfaces são descritos na recomendação:

- **Interface de Aquisição:** Essa interface em conjunto com o SCP permite a coleta de informações dos elementos externos;
- **Interface de Interoperabilidade:** Esta interface em conjunto com o SCP permite a comunicação com provedores de dados externos e sistemas de computação de terceiros;
- **Interface de Serviço:** Essa interface em conjunto com o SCP permite que as aplicações tenham acesso às funções de suporte fornecidas pelo SCP.

A recomendação destaca ainda os requisitos para interoperabilidade de plataformas de cidades inteligentes, a fim de garantir um serviço de interoperabilidade adequado:

- **Interoperabilidade com Diferentes Tecnologias:** Capacidade de suportar diferentes tecnologias para a captura de padrões de informação e comunicação, bem como sistemas de informação internos, corporativos e, ou, externos;
- **Desempenho:** Capacidade de lidar com um grande número de dispositivos, serviços e processos de forma eficiente;
- **Escalabilidade:** Capacidade de aumentar os volumes de processamento, interligação e armazenamento sem precisar mudar a arquitetura;

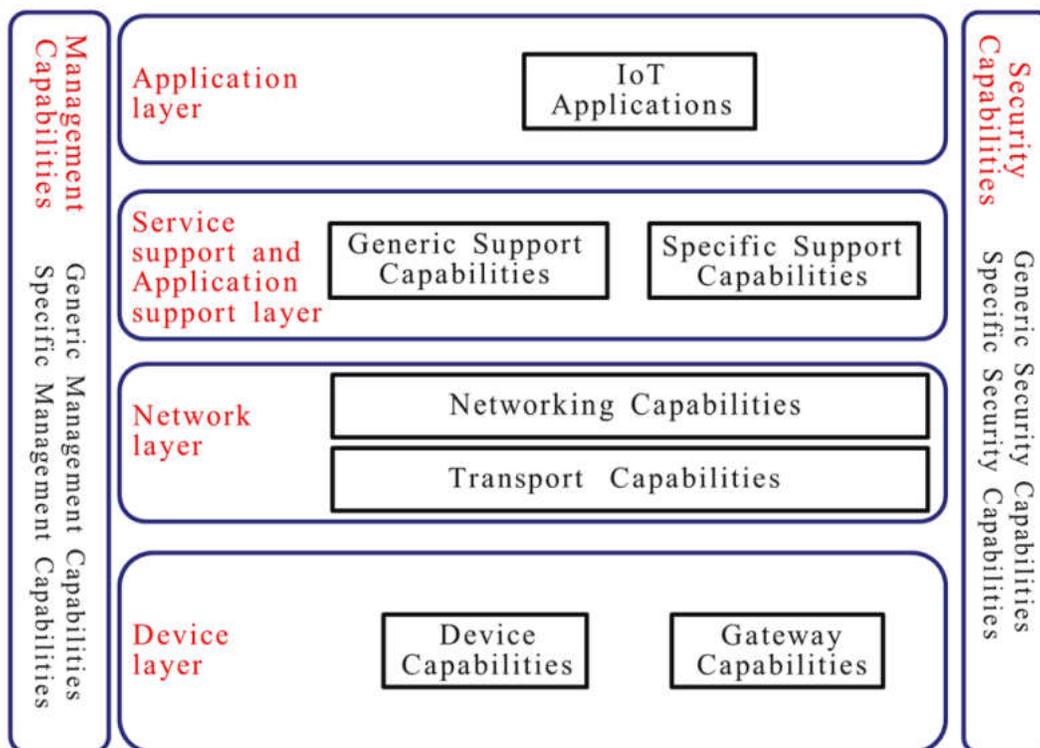
- **Robustez e Resiliência:** Capacidade de continuar enfrentando problemas;
- **Segurança:** Garantia de segurança e confiabilidade;
- **Extensibilidade:** Adaptabilidade para atender novas necessidades.

### C. ITU-T Y.2060 - Overview of the Internet of things

Na seção 8 da recomendação é apresentado um modelo de referência de IoT, dividido em quatro camadas: Camada de aplicação, Camada de suporte ao serviço e a aplicação, Camada de rede e Camada de dispositivo.

Além disso são destacadas as necessidades de Gerenciamento e Segurança, o modelo pode ser visualizado na Figura 9.

Figura 9 – Modelo de Referência de IoT – ITU-T Y.2060



Fonte: ITU-T, 2012

Na seção 8.5 são detalhados os aspectos de Gerência. A recomendação deixa claro, que IoT possui particularidades, mas a gerência

clássica por meio do modelo FCAPS faz-se necessária tanto quanto nas redes tradicionais.

Os recursos genéricos de gerenciamento, que são essenciais segundo a recomendação, incluem:

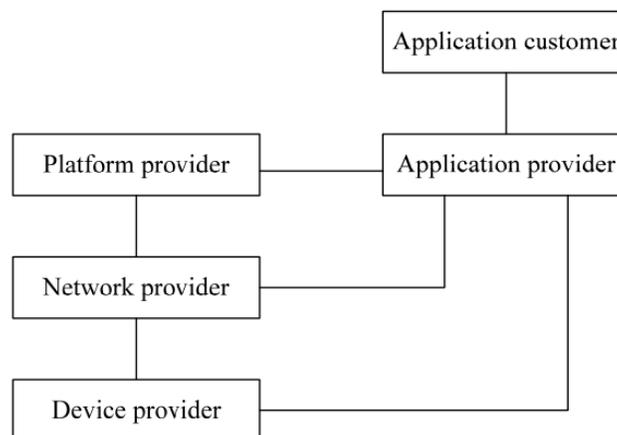
- Gerenciamento de dispositivos, como ativação e desativação de dispositivos remotos, diagnósticos, atualização de firmware e/ou software e gerenciamento de status de funcionamento do dispositivo;
- Gerenciamento de topologia de rede local;
- Gerenciamento de tráfego e congestionamentos, como a detecção de condições de limite de fluxo de rede e a implementação de reserva de recursos para fluxos de dados essenciais e/ou críticos.

O gerenciamento específico está associado aos requisitos específicos das aplicações, como por exemplo, monitoramento de linha de transmissão de energia de redes inteligentes.

No apêndice A são tratadas as questões do ecossistema de IoT e modelos de negócios.

É destacado o ponto de que cada empresa possui pelo menos uma regra de negócio, e, então é proposto modelo que exhibe os papéis e relacionamentos relevantes que podem ser encontrados em implantações de negócios da IoT, que pode ser observado na Figura 10.

*Figura 10 – Papéis e Relacionamentos em Negócios de IoT*



Fonte: ITU-T, 2012

A recomendação possui uma nota dizendo que o modelo não contempla todos os possíveis papéis e relacionamentos, e em seguida é realizado uma descrição dos mesmos:

- **Device Provider:** É o responsável pelos dispositivos que fornecem dados brutos e/ou conteúdo para o *Network Provider* e para o *Application Provider* de acordo com a lógica de serviço.
- **Network Provider:** Ele desempenha um papel central no ecossistema da IoT, executando as seguintes funções principais:

Acesso e integração de recursos fornecidos por outros provedores;

Suporte e controle da infraestrutura de recursos da IoT;

Oferta de recursos de IoT, incluindo recursos de rede e exposição de recursos a outros fornecedores.

- **Platform Provider:** Fornece recursos de integração e interfaces abertas. Diferentes plataformas podem fornecer recursos diferentes para *Application Providers*.

Os recursos de plataforma incluem capacidades de integração, bem como armazenamento de dados, processamento de dados ou gerenciamento de dispositivos, e, também é possível prover suporte para diferentes tipos de aplicações IoT.

- **Application Provider:** Utiliza recursos fornecidos pelo *Network Provider*, *Device Provider* ou *Platform Provider*, a fim de fornecer aplicações IoT para os clientes dessas aplicações.
- **Application Customer:** É o usuário das aplicações IoT fornecidas pelo *Application Provider*. Um *Application Customer* pode representar vários usuários das aplicações.

A recomendação encerra o apêndice destacando que em aplicações reais os participantes do ecossistema de IoT, podem ter outros relacionamentos não descritos.

O motivo para essa variedade de relacionamentos são os diferentes modelos de negócios possíveis.

Por fim são apresentados cinco modelos de negócios de IoT, sob a perspectiva do serviço de telecomunicações e operadores de rede.

### **2.3.4. Soluções para coleta e apresentação de dados**

Possuir uma única ferramenta que possa ser utilizada para monitoramento da rede de computadores, aplicações, sensores, protocolos indústrias e etc., aumenta a produtividade, pois os profissionais envolvidos precisam dominar apenas uma tecnologia, além disso, a troca de experiências entre setores, em relação à utilização da ferramenta, se torna natural.

A possibilidade de adaptar-se para múltiplos cenários como IoT, Indústria 4.0, Cidades Inteligentes, Redes Definidas por Software (SDN, do inglês, *Software Defined Networking*), é demasiadamente desejada, pois estes são temas relevantes atualmente, e a tendência é de que esta relevância aumente ainda mais nos próximos anos.

## **2.4. Ferramentas de Gerência**

Conforme abordado na seção 2.3.3 a gerência é uma recomendação recorrente. Nessa seção serão analisadas algumas das opções disponíveis para o gerenciamento e apresentação dos dados.

Em seguida será analisado um dos elementos que possibilitam a ligação de RSSF com a Internet, o *Proxy Manager*. Destacando-se os pontos elementares para a compreensão do trabalho.

Por fim, será analisado o software de gerência de redes Zabbix, evidenciando as características da ferramenta, seus componentes e utilitários. Tal análise se faz necessária para elucidar as razões que levaram o autor a optar por essa solução em detrimento de outras, uma vez que um capítulo de comparativo de ferramentas não agregaria ao objetivo do trabalho.

### **2.4.1. Sistemas Supervisórios**

Existem duas categorias de sistemas supervisórios amplamente conhecidos e utilizados no mercado: os *Network Management System* (NMS), utilizados no ambiente de redes de computadores e o SCADA (*Supervisory Control and Data Acquisition*), utilizados em ambientes industriais.

Apesar de serem utilizados em ambientes diferenciados, e fazerem uso de protocolos de comunicação distintos, estes sistemas possuem basicamente o mesmo propósito: obter informações sobre o estado de operação dos equipamentos e coletar dados.

Nesse sentido foram analisadas algumas soluções de SCADA e soluções específicas para IoT, além de algumas características comparadas ao Zabbix (ZABBIX, 2017a).

Como o objetivo do trabalho não é uma comparação de soluções de IoT, foram escolhidas algumas opções, por serem encontradas comumente em tutoriais e palestras de IoT, com o objetivo de exemplificar funcionalidades, e comparar com algumas das características da solução que o autor utiliza para implementar a plataforma.

O ScadaBR (SCADABR, 2017) é um software livre, de código-fonte aberto, voltado para o desenvolvimento de aplicações SCADA. O desenvolvimento desta solução foi realizado em ambiente JAVA (2017), o que traz como vantagem o fato do servidor ser multiplataforma. No entanto, os requisitos de hardware para utilização do ScadaBR são maiores em relação ao Zabbix que é desenvolvido em linguagem de programação C e PHP (ZABBIX, 2017a).

Soluções online como TAGO (2017), THINKSPEAK (2017), SIGFOX (2017), entre outras, são plataformas para envio de dados de sensores com a proposta de facilitar a visualização dos dados. A primeira desvantagem evidente, é que no caso de interrupção de conexão com a Internet as coletas de dados são perdidas, problema este, resolvido no Zabbix por meio do uso de *Proxies*; essa estratégia poderia ser implementada em outras soluções, entretanto, no Zabbix o *Proxy* é um item nativo. Outro problema, é que em todas essas plataformas, existirá algum tipo de restrição sobre a quantidade de sensores, intervalo de coleta e até mesmo exigência de tecnologia específica, no caso do SIGFOX.

A plataforma DOJOT (2017) é desenvolvida pelo CPqD em parceria com outras instituições de ciência e tecnologia. Esta solução, visa prover uma infraestrutura para desenvolvimento e utilização de tecnologias de IoT e

idades digitais. Apesar de promissor no momento, o projeto ainda é embrionário.

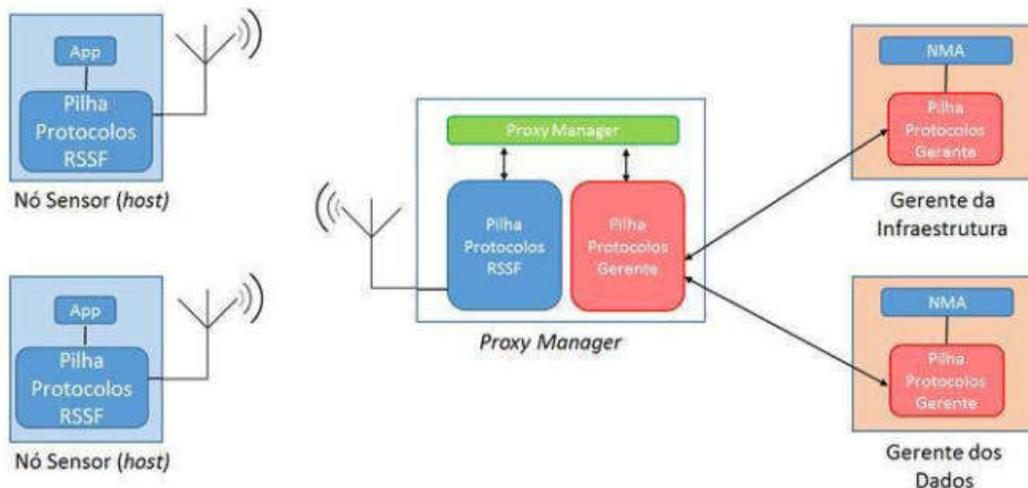
Com relação ao Zabbix, a principal desvantagem da solução DOJOT é o fato de ser distribuída como um único pacote via imagem DOCKER (2017) ou máquina virtual. Enquanto que no Zabbix é possível habilitar apenas os recursos necessários para o ambiente de atuação, como por exemplo, suporte ao monitoramento de aplicações Java ou ambiente VMWARE (2017).

### 2.4.2. Proxy Manager

OLIVEIRA (2016) propõe um novo elemento na arquitetura de RSSF, ao invés de um simples *gateway* que converte pacotes de RSSF em pacotes TCP/IP, esse elemento é programado para interagir com a RSSF, alterando rotas de comunicação em decorrência de problemas de intensidade de sinal, ou falha de um dos nós sensores.

Devido a essa nova característica de atuação, o autor passou a adotar o termo *Proxy Manager*, pois como pode ser observado na Figura 11, seu papel passa a ser o de um gerenciador dos elementos da RSSF.

Figura 11 - Introdução de um Proxy Manager para o Gerenciamento de uma RSSF



Fonte: OLIVEIRA, 2016

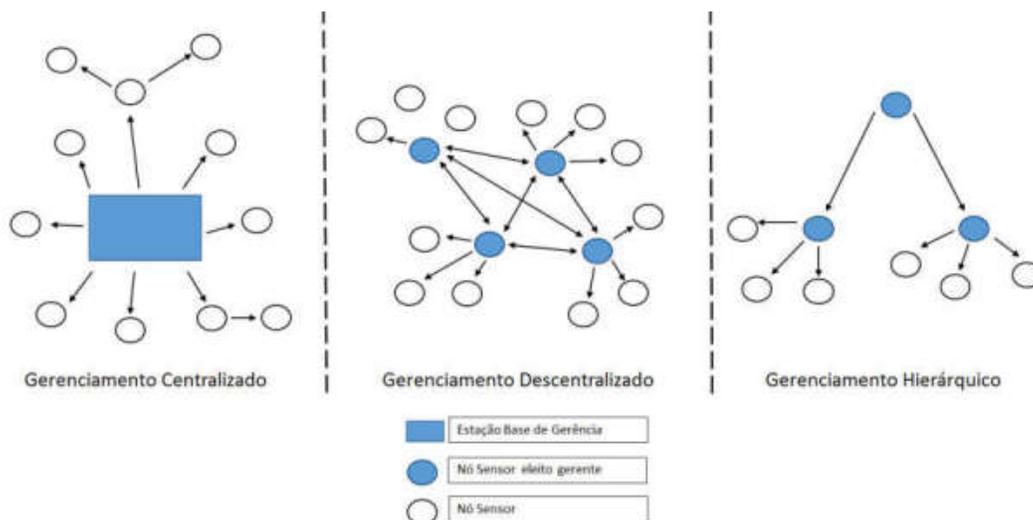
Além disso o autor utiliza a estratégia de *Gateway/Proxy* atuando como *front-end*, estratégia na qual os dados a serem enviados para as duas redes

são armazenados em um banco de dados local, acessível tanto da RSSF como da Internet.

Do ponto de vista de utilização dos recursos essa abordagem é eficiente, pois podem ser realizados tratamentos e agregações nos dados coletados antes do envio dos mesmos, descartando dados desnecessários para a aplicação.

Outro ponto destacado por OLIVEIRA (2016, p. 33) é que “o controle de uma plataforma de gerenciamento para RSSF pode ser classificado de três formas: centralizado, descentralizado e hierárquico”, como apresentado na Figura 12.

Figura 12 - Controles de Gerenciamento Possíveis em uma RSSF



Fonte: OLIVEIRA, 2016

A proposta do autor é a implementação de um *Proxy Manager*, com arquitetura de gerenciamento centralizado, com a finalidade de gerenciar RSSF de múltiplos saltos, instalada em ambientes previamente conhecidos.

Por fim o autor demonstra que a proposta atende às demandas das camadas horizontais, por meio das áreas funcionais (ISO) e verticais, por meio do foco da atuação. O conceito é exibido na Figura 13.

Figura 13 - Distribuição das funções de Gerência

		Focos de Atuação	
		DADOS	INFRAESTRUTURA
Áreas de Gerência	CONFIGURAÇÃO	- Configuração dos Sensores	- Configuração dos Nós Sensores - Parametrização do PM - Estabelecimento de rotas
	DESEMPENHO	(A ser definido pela aplicação)	- Monitoramento da PER nos NS
	FALHAS	(A ser definido pela aplicação)	- Detecção de PER alta

Fonte: OLIVEIRA, 2016

Segundo OLIVEIRA (2016, p. 39), “as áreas de gerência escolhidas (Configuração, Desempenho e Falhas) foram escolhidas, pois dizem respeito as funcionalidades de parametrização, inicialização, monitoração e manutenção da RSSF, sem os quais esta não pode operar de maneira estável”.

### **3. REQUISITOS DESEJÁVEIS EM UM MODELO DE REFERÊNCIA PARA IOT**

Para elaborar o modelo de referência foi preciso levantar as necessidades que envolvem uma solução de IoT, desde a coleta dos dados até a apresentação dos mesmos.

Durante esse processo várias abordagens foram necessárias, da revisão da literatura apresentada nas seções anteriores, ao desenvolvimento de um questionário para interação com o cliente (apêndice A).

Ao se adotar uma solução de software dentro de uma plataforma de IoT, com foco nas PME, faz-se necessário observar se a solução apresenta certas características. O modelo de referência atua como um guia na observação destas questões.

#### **3.1. Requisitos de Usuário**

O foco da construção do modelo de referência, foi permitir que o mesmo seja genérico, para que possa ser aplicado à diferentes tipos de clientes, de áreas de atuação diversas.

E devido à essa natureza genérica, é necessário criar um roteiro para que as entrevistas com os usuários sejam padronizadas, garantindo que as informações essenciais para o desenvolvimento da aplicação sejam atendidas.

Nesse sentido foi desenvolvido o formulário (apêndice A), que é dividido em duas partes, a primeira responsável por entender as necessidades do cliente e como ele deseja exibir essas informações, ou seja, levantar os requisitos do negócio, que serão tratados na seção 3.2.

Na segunda parte do questionário são tratadas as questões de como os dados serão coletados, que tipo de sensores serão utilizados, questões de comunicação, alimentação e demais informações úteis ao Operador da RSSF, papel que foi tratado na revisão de literatura e será abordado novamente na seção 3.6.

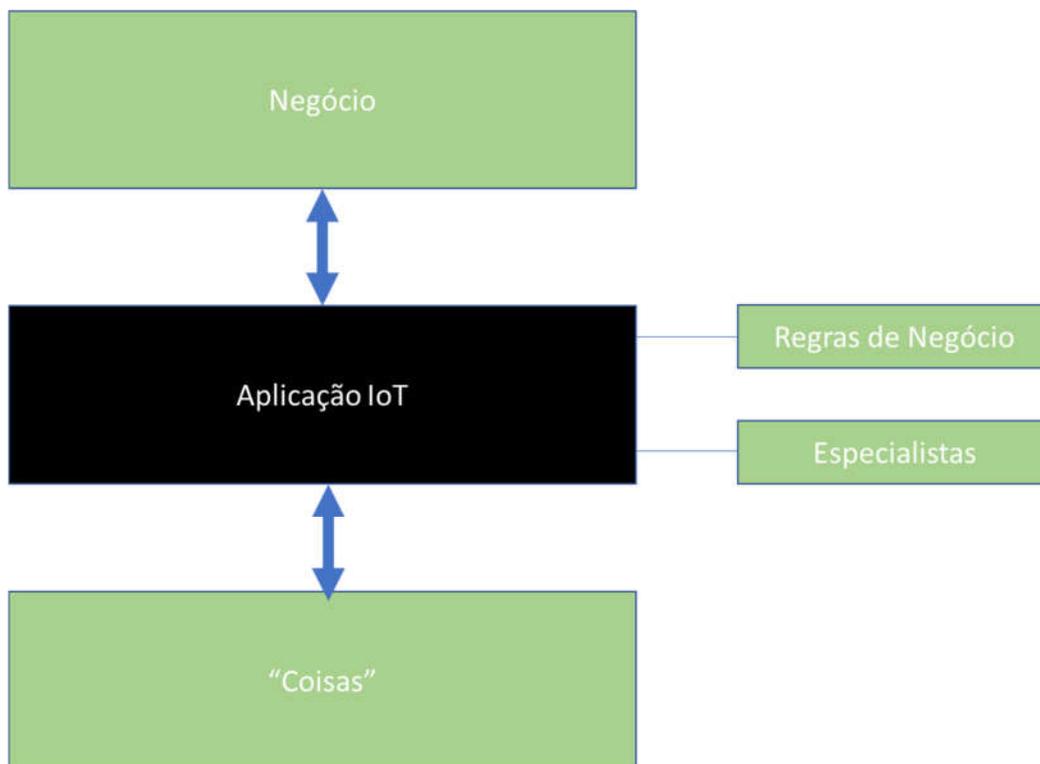
### 3.2. O papel do negócio na solução de IoT

Um negócio possui elementos bem definidos e que não podem ser ignorados no momento da criação de uma solução de IoT:

- O negócio visa lucro, sejam tangíveis ou intangíveis;
- Existem regras de negócio que devem ser refletidas na solução;
- O negócio possui especialistas em sua área de atuação, e os mesmos devem ser envolvidos na elaboração da solução;
- As “coisas” que farão parte da solução, precisam atender as necessidades do negócio.

A Figura 14 mostra o papel de uma aplicação IoT com as interdependências das coisas e negócio, por meio de suas regras e especialistas.

Figura 14 - O papel do negócio na solução de IoT



Fonte: Elaboração Própria

A questão não é somente realizar um controle local, uma automação, como já acontece em muitos processos industriais. Um exemplo é no setor

agrícola onde já existem grandes avanços na questão de automação, porém focado na produção. O objetivo é apenas produzir mais e/ou mais rápido e assim disponibilizar o produto ao mercado.

O negócio pode ser melhor atendido se obtiver um refinamento da análise dos dados, assim como, análise de fatores externos que podem influenciar no controle dos processos.

Uma solução IoT leva em consideração esse refinamento, criando para isto condições de atendimento de necessidades específicas.

Por exemplo, a temperatura de um processo que depende da temperatura ambiente, e, talvez seja necessário, analisar a previsão do tempo das próximas horas para identificar possíveis situações de não atendimento de algum requisito.

No exemplo do setor agrícola, poderiam ser observadas as condições climáticas, a variação do valor do produto, e tentar encontrar mecanismos para que a produção esteja em sintonia com o melhor momento da venda.

Naturalmente uma solução IoT tem acesso à Internet e pode fazer uso da previsão do tempo para determinar as condições de controle, pode consultar sites especializados em cotação de produtos, entre outras ações.

Também é possível, em função do poder de processamento de alguns elementos, uma análise para obter informações de dados brutos, fazendo, por exemplo, correlação entre variáveis do sistema.

### **3.3. Coleta de Dados**

Para que os dados possam ser transformados em informações, eles precisam ser coletados. Nesse sentido é importante que a ferramenta suporte o maior número de protocolos possível. Também é importante que a ferramenta seja flexível, permitindo a execução de comandos do Sistema Operacional (SO), *scripts* personalizados e etc.

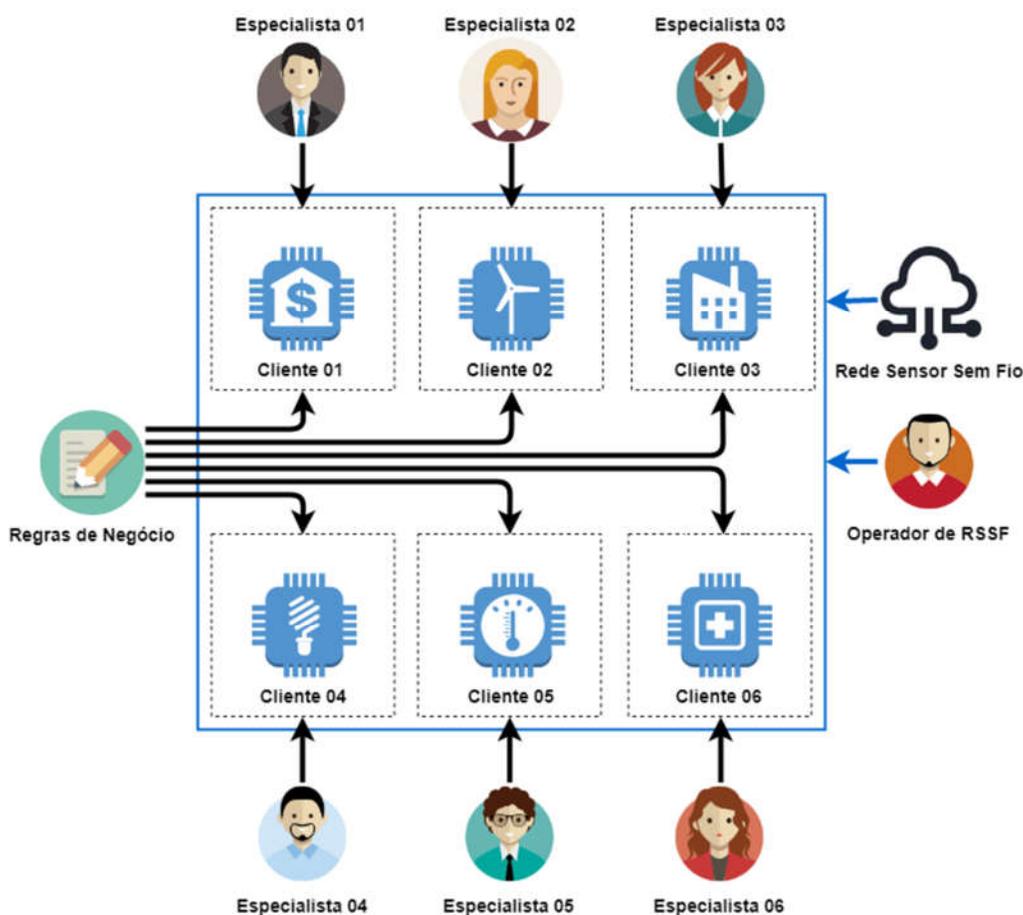
Atualmente, é essencial que a ferramenta possua uma API (*Application Programming Interface*) que basicamente, é um conjunto de instruções que permite a interação com a ferramenta por meio de linguagens de programação.

A API facilita o desenvolvimento de novas maneiras de coleta de dados e envio para a ferramenta. Permitindo, por exemplo, coletar informações referentes à uma aplicação e enviá-las para outra aplicação.

### 3.4. Critérios para o Tratamento e Controle dos Dados

Coletar os dados é uma parte essencial do processo, porém, uma vez que os dados estejam sendo coletados, é preciso transformá-los em informações. Tais alterações, são obtidas, por meio da realização de tratamento de dados, esse tratamento é feito sob diferentes pontos de vista e de maneiras distintas. Essas correlações podem ser visualizadas na Figura 15, os papéis serão descritos em sequência.

Figura 15 - Relação entre Clientes, Regras de Negócio e Operador da RSSF



Fonte: Elaboração Própria

Um especialista no negócio a ser atendido é quem vai determinar os parâmetros aceitáveis desses dados. O que é uma situação normal de operação e o que é uma situação que exige uma notificação.

Já as regras de negócio determinam como esses dados devem ser tratados. Por exemplo, um negócio pode ter a necessidade de receber apenas dados agrupados por média de hora, e notificações em caso de situações fora do normal, enquanto que outro negócio exigirá uma cópia local dos dados brutos, todas as coletas, e o envio de dados agrupados por média de 5 minutos.

Existe ainda o papel do Operador da RSSF, cabe a ele a responsabilidade de manter o sistema em funcionamento, ele deve garantir que a coleta e transmissão dos dados estão ocorrendo e agir em casos de inconformidade. Este papel será detalhado na seção 3.6.

Outro ponto que deve ser considerado nessa fase da implementação é o tratamento e controle de exceções. Não faz sentido enviar uma coleta que apresentou problemas, por exemplo, uma temperatura de  $-60^{\circ}\text{C}$  ou  $300^{\circ}\text{C}$ . O tratamento e controle, faz-se necessário também neste sentido.

Os dados coletados e considerados válidos, serão tratados de acordo com as necessidades do cliente. Mas de maneira geral pode-se aplicar conceitos matemáticos e estatísticos de tratamento de dados, como desvio padrão, tendência e projeção dos dados, para citar alguns exemplos.

### **3.5. Definição de Limites e Execução de Ações**

Nas seções anteriores discutiu-se que é necessário interagir com os dados para atender as demandas do negócio, e a principal maneira de alcançar esse objetivo é por meio da definição de limites e da execução de ações em decorrência dos mesmos.

Os limites representam na ferramenta as limitações existentes no ambiente real. Como, por exemplo, atingir certa temperatura, nível de determinado líquido, proximidade de uma área específica, detecção ou não de movimento, entre outros.

A ação é o que acontece quando os limites são ultrapassados, por exemplo, ao atingir determinada temperatura, um sistema de resfriamento ou aquecimento deve ser acionado, uma mensagem deve ser enviada ao operador, um alerta sonoro deve ser executado na ferramenta e o ícone no mapa deve ser trocado.

### **3.6. Operador da Rede de Sensor Sem Fio**

Conforme observado nas recomendações do ITU-T, uma solução de IoT pode conter elementos que não existem em uma rede tradicional, porém, a RSSF exige gerenciamento.

A RSSF é composta de sensores, atuadores, relés, módulos rádios e etc. Todos esses elementos são compostos de componentes eletrônicos, e, portanto, podem apresentar problemas.

A aplicação de um cliente pode estar em uma RSSF compartilhada, como ilustrado na Figura 15, ou possuir uma estrutura dedicada. Em ambas as situações, cabe ao Operador de RSSF garantir o funcionamento da plataforma e atuar em caso de necessidade, seja substituindo um sensor, realizando um ajuste nas configurações de rádio, trocando uma bateria e etc.

Diante deste cenário, surge a necessidade de visualização dos dados referentes à RSSF por parte do Operador. É necessário que a comunicação da RSSF seja monitorada, que regras de controle sejam estabelecidas, que telas de monitoramento e alertas sejam criados com foco no Operador de RSSF.

Como pode ser observado na Figura 16, os dados no nível de borda possibilitam que o nível de aplicação seja alimentado com duas visões distintas, para públicos específicos, o cliente e o Operador da RSSF, que possuem objetivos claros e bem definidos, que foram adquiridos durante o planejamento da solução.

*Figura 16 - Visão de Dados e Visão de Rede*

Fonte: Elaboração Própria

Sendo assim, tudo que foi discutido nas seções anteriores, se fará presente novamente, porém com foco na visão da rede. Inclusive o papel do especialista pode ser necessário, dependendo da solução de rádio adotada e do nível de conhecimento do Operador.

Cabe ainda salientar que dependendo do modelo de implementação e do tamanho da RSSF, pode existir uma interface de visualização no elemento de borda, não sendo necessário enviar os dados referentes à visão de rede para a Internet.

### **3.7. Interfaces de Comunicação**

O modelo de referência, como citado nas seções anteriores, precisa contemplar da coleta dos dados à exibição na Internet. Na Figura 17 pode-se observar os níveis iniciais do modelo, suas interconexões e elementos presentes em cada camada. Nessa seção serão abordadas as interfaces de comunicação do modelo.

Figura 17 - Modelo de Referência (Parcial) com Tecnologias



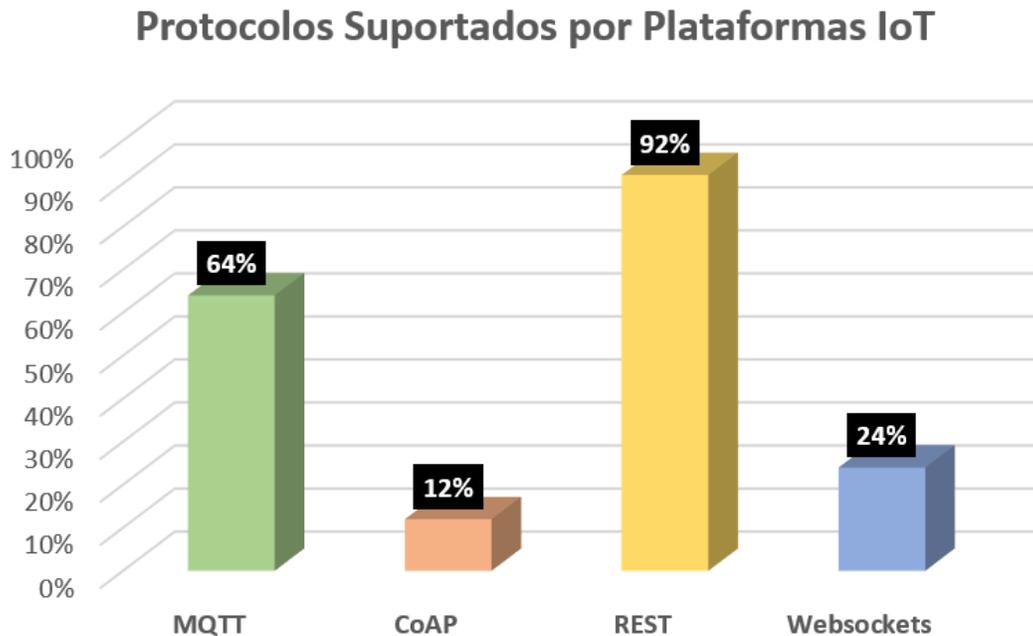
Fonte: Elaboração Própria

Para que seja possível coletar os dados dos sensores e atuadores e entregá-los ao elemento de borda, é preciso comunicação. Pela própria natureza das aplicações de IoT, comumente essa comunicação é sem fio.

Conforme pode-se observar na Figura 17, existem basicamente duas categorias de redes. Soluções que utilizam tecnologia TCP/IP e soluções de rádio, e dentro de cada categoria, existem várias tecnologias à disposição.

O passo seguinte é enviar os dados do elemento de borda para a aplicação na Internet. CRUZ et al. (2018) analisaram 33 plataformas de IoT, destas são citadas as tecnologias de comunicação suportadas por 25 delas. Na Figura 18 observa-se, das ferramentas analisadas pelos autores, que a metodologia REST é a mais comum, seguida do protocolo MQTT.

Figura 18 - Protocolos Suportados por Plataformas de IoT



Fonte: Elaboração Própria

Ressaltando que algumas plataformas suportam mais de um protocolo, o gráfico representa a porcentagem de protocolos suportados em relação ao número de plataformas analisadas pelos autores, como no caso da metodologia REST que é amplamente suportada.

Segundo as recomendações do ITU-T analisadas no presente trabalho, deve-se ter a preocupação com as questões de gerência e segurança na transmissão dos dados. O modelo destaca que tecnologias diferentes podem ser utilizadas em conjunto, para atender momentos distintos da transmissão:

- Comunicação dos elementos de coleta com o elemento de borda;
- Comunicação do elemento de borda com a aplicação na Internet.

### 3.8. Armazenamento Local e na Nuvem

Essa distinção entre os dados locais e os dados na nuvem, faz-se necessária por uma série de razões que serão elencadas ao longo dessa seção. A primeira delas, está diretamente ligada à seção anterior: O que acontece se o elemento de borda perder a comunicação com a Internet?

A solução proposta neste trabalho, é de utilizar um mecanismo para que os dados continuem sendo coletados e armazenados localmente, e posteriormente quando a comunicação com a Internet for reestabelecida, os dados sejam enviados à aplicação.

Outro ponto já discutido na seção 3.4 é que clientes possuem necessidades diferentes, esses dados precisam ser tratados localmente. Parte deles ou sua totalidade, dependendo da demanda, serão armazenados localmente, para servirem, por exemplo, de fonte para tecnologias como *Data Mining*.

Além disso deve-se considerar que nem todo dado coletado será enviado para a aplicação na Internet, o processo de tratamento e controle de dados irá eliminar alguns dados, como no exemplo de temperatura, citado na seção 3.4.

Por fim, os dados referentes à visão de rede, ou seja, informações que dizem respeito ao funcionamento da RSSF, não precisam necessariamente serem enviados à nuvem, como citado na seção 3.6. Em determinadas situações o Operador da RSSF visualizará esses dados localmente. Noutro cenário, apenas os alertas de situação de não conformidade precisam ser enviados ao Operador, enquanto que os dados permanecem armazenados localmente.

### **3.9. Interação com o Usuário**

A automatização das ações não é suficiente para que um sistema seja considerado como IoT, sendo necessária também, a interação com o usuário.

O primeiro nível de interação é por meio de uma interface, preferencialmente web, para que possa ser acessada a qualquer momento de qualquer dispositivo conectado ao ambiente da solução ou Internet, dependendo da abordagem.

Nesse sentido é preciso exibir gráficos, telas com informações atualizadas, mapas que representam determinadas áreas, alertas visuais e/ou sonoros, entre outros.

É necessário que o usuário do sistema se sinta confortável com as informações apresentadas, e que elas sejam as mais claras possíveis. Para atender esse quesito é possível utilizar diferentes ícones para situações de normalidade e problema, diferentes cores para diferentes níveis de problema.

Porém, outras estratégias podem ser utilizadas para interação com o usuário, mesmo que ele não tenha acesso à ferramenta. Pode-se enviar notificações por e-mail e/ou mensagens de SMS.

### **3.10. Adaptação para Múltiplos Cenários**

Possuir uma ferramenta que possa ser personalizada para atender as necessidades de múltiplos clientes, é essencial ao se trabalhar com um cenário de IoT, principalmente com foco em PME.

A solução deve adaptar-se ao cliente e não o inverso, adquirir um produto e depois, em função desse produto, ter que mudar o método de trabalho, ou se adaptar a termos técnicos que não são familiares é algo desgastante para o cliente.

Não basta que a solução possua ícones, esses ícones devem fazer sentido para o usuário, é preciso que as telas possam ser configuradas de acordo com o conceito que o usuário possui de interface, seja uma planta baixa ou interface com dois botões: “OK” e “Problema”, por exemplo.

Se houver opção de envio de mensagens por e-mail, SMS, ou algum software de comunicação, a mesma necessidade se aplica, as mensagens devem fazer sentido para quem as recebe.

#### 4. PROPOSTA DE UM MODELO DE REFERÊNCIA DE IOT

Uma vez detectadas as necessidades, propõe-se um modelo para atender as demandas. A proposta deste trabalho é apresentar um modelo de referência que contemple a divisão das atividades necessárias para a conceitualização, implementação e operação de uma solução de IoT, abrangendo desde o universo das “coisas” até o negócio.

Com base nesse modelo será implementada uma solução fim a fim, totalmente *Open Source* com foco nas pequenas e médias empresas. Faz-se necessária a criação de um modelo de referência, devido à natureza do universo de IoT. Para isso, é necessário envolver as soluções de muitas áreas para prover uma solução IoT.

Os conceitos dessas áreas são multidisciplinares, o que dificulta a visualização e entendimento do que será necessário para a solução durante o desenvolvimento da ideia.

Com o modelo de referência é possível exibir quais são as tecnologias envolvidas nos diversos níveis. O objetivo de dividir a solução em níveis é facilitar a visualização das etapas e possibilitar o detalhamento das necessidades e desafios de cada etapa. Isso possibilita criar uma base para análise das diferentes tecnologias.

Inspirado no modelo da Cisco, discutido na seção 2.1, é proposto um Modelo de Referência, denominado “**Modelo de Referência Aberto para Serviços de IoT**”, adaptado para a realidade de uma implementação prática, que tem foco na utilização por pequenas e médias empresas no Brasil.

A proposta é identificar todas as etapas para implantação de uma solução IoT, realista e implementável com foco nesse segmento de mercado.

A atenção com este tipo de aplicação se deve à questão de custo das soluções proprietárias providos pelas grandes empresas, o que cria um “vale dos excluídos”, no caso as PME, que não tem condições de implementar soluções onerosas. Isso causa uma situação em que essas empresas não conseguiriam usufruir dos benefícios da Internet das Coisas.

Diante desta situação a proposta permite a implementação completa de uma solução utilizando plataformas *Open Source* em todas as etapas da solução.

Esta proposta permite ao pequeno e médio empresário até mesmo utilizar outras soluções como parte de um sistema completo, mas dando um referencial para que os mesmos consigam optar pelas melhores solução de acordo com sua realidade.

#### **4.1. Arquitetura**

Toda solução de IoT nasce da necessidade de se coletar alguma medida ou controlar algo, para atender algum objetivo do negócio. Se não existe um negócio envolvido, está apenas realizando-se experimentos, que posteriormente podem vir a se tornar um produto ou não.

Porém, quando se parte de uma necessidade real, existe um interessado e um objetivo claro a ser atendido, que refletirá a agregação de valor em algum ponto do negócio e portando necessita de estabilidade e garantia de funcionamento, ou seja, essa solução precisa ser confiável, gerenciável e segura.

A Figura 19 demonstra o modelo de referência, que cobre todas as etapas a serem realizadas, da coleta do dado à disponibilização da informação, auxiliando o negócio na tomada de decisão. É possível observar os níveis, suas correlações e seus elementos. Cada nível, elemento e necessidade será detalhado nas seções subsequentes.

Figura 19 - Modelo de Referência



Fonte: Elaboração Própria

Para facilitar a análise da proposta, na seção 0 será utilizado um exemplo simples, para que fique claro a utilização de uma solução completa, envolvendo todas as etapas, e, resultando em um serviço para IoT. Entretanto, antes disso o Modelo de Referência será detalhado na próxima seção.

#### **4.2. Visão Conceitual de um Modelo de Referência de IoT**

Como já discutido anteriormente, toda solução de IoT, parte da necessidade de monitorar algo para que determinados objetivos do negócio sejam atendidos. Para que essa ligação das “coisas” com o negócio aconteça, algumas etapas precisam ser realizadas:

- Os dados precisam ser coletados e tratados localmente, próximo ao processo;

- Os dados precisam ser transmitidos até uma central de coletas para tratamento e posterior envio para a Internet. Sendo quase sempre utilizada uma solução de comunicação sem fio;

- É preciso um elemento de borda, que receba esses dados, realize determinados tratamentos, com algoritmos de controle e processos refinados, que permitam a aplicação de regras para controle do processo local, e, envie os dados para a Internet;

- Uma solução na nuvem que receba e armazene esses dados tratados pelo elemento de borda;

- Por meio de regras bem definidas e alinhadas com o negócio ocorre a abstração dos dados, transformando-os em informações. Nesse ponto é necessário um especialista, que seja capaz de identificar as necessidades para coleta e tratamento dados;

- Essas informações são exibidas por meio de alguma interface, que reflita as regras de negócio, auxiliando na tomada de decisões, como, por exemplo, por meio de exibição de status, alertas e alarmes;

- Para garantir que todas as etapas sejam cumpridas de forma satisfatória, garantindo a integridade, confidencialidade e disponibilidade, são empregadas as questões de gerência e segurança.

Esse é o fluxo que os dados percorrem da coleta à exibição, sendo transformados em informações ao longo do processo. Como se pode observar, existe uma relação direta de entrega entre os níveis, ao mesmo tempo que existe um objetivo maior que conduz o processo.

#### 4.2.1. Nível 0 – Nível das Coisas

Nesse nível estão presentes os elementos físicos, com os quais a solução de IoT precisa interagir para atender as necessidades do negócio. Aqui são apresentadas as necessidades do cliente: coletar uma temperatura ou umidade do solo, ligar ou desligar um motor, emitir um alerta sonoro em determinada situação e etc.

#### 4.2.2. Nível 1 – Nível de dispositivo, condicionamento e controle

Uma vez que existe um interessado e uma necessidade definida, é preciso descobrir como coletar esse dado e o que ele representa ao negócio. Por que é importante coletar essa temperatura? Quais são os limites estabelecidos e quais os impactos para o negócio, caso eles sejam ultrapassados?

Portanto, o Nível 1 está diretamente ligado ao Nível 7. A coleta de uma grandeza como temperatura, representa a vida dos alevinos em uma empresa de piscicultura, enquanto representa o controle do estoque em um fornecedor de chocolates e pode impedir um acidente gravíssimo em um ambiente industrial de caldeiras.

Esse nível é composto de alguns elementos:

- **Sensor:** Responsável por captar as grandezas;
- **Atuador:** Atua em algum processo;
- **Condicionamento e controle:** Faz a captura e tratamento das grandezas medidas e possui os algoritmos de controle local.

Os dispositivos que transformam as grandezas físicas em sinais elétricos, comumente são chamados de sensores ou transdutores, eles podem ser digitais ou analógicos.

O sinal elétrico proveniente dos sensores deve ser tratado para avaliar possíveis *outliers*, leituras discrepantes, como por exemplo 300° em temperatura ambiente, garantido assim a consistência dos dados.

O tratamento é considerado um condicionamento e neste ponto são possíveis estratégias de processamento digital dos sinais, para não somente

medir a grandeza, mas criar outras informações, como realizar uma média móvel para avaliar a velocidade de mudança da grandeza.

Vale ressaltar que localmente existe um microcontrolador com poder de processamento para realizar estratégias locais de decisão. Também, localmente, existe uma certa capacidade de processamento, o que cria a possibilidade de tratar dados a uma certa taxa de amostragem, não utilizar esta capacidade é transferir a responsabilidade para outra entidade, que possuem outras funções.

No condicionamento estão algoritmos de decisão local, que foram parametrizados por outras entidades do modelo de referência, como por exemplo, informações provenientes do especialista, ou determinada regra de negócio. De acordo com os parâmetros recebidos das leituras é possível acionar atuadores, como por exemplo, um relê.

Nesse nível pode-se utilizar algoritmos capazes de ajustar a temperatura numa dada faixa de valores, e, com diferentes tempos, por meio de controle dos processos. Ou medir e avaliar a consistência da temperatura com a umidade, já que existe uma relação conhecida entres essas duas grandezas, como por exemplo, na pressão atmosférica (MUNDO, 2017).

#### **4.2.3. Nível 2 – Nível de conectividade**

O objetivo de todo dado coletado é chegar à nuvem como uma informação útil ao negócio, mas antes disso ele precisa sair do sensor e chegar até um elemento de borda.

Existem várias opções para conectividade, sejam elas sistemas proprietários ou de código fonte aberto. Pode-se classifica-las em três categorias:

– **Soluções de Internet:** Protocolos proprietários que utilizam para a conectividade, protocolos próprios na RSSF, e disponibilização dos dados na Internet por meio do protocolo TCP/IP. Como, por exemplo, SigFox, LoraWan, LTE NoB (3GPP, 2018) e WI-Fi (2018).

– **Soluções de Conectividade:** Protocolos proprietários que utilizam protocolos específicos. Como, por exemplo, LoRa, ZigBee, Z-WAVE (2018) e BLUETOOTH (2018);

– **Soluções *Open Source*:** Protocolos de código fonte aberto que permitem a construção de redes especializadas para atender problemas específicos. Como, por exemplo, o RADIUINO (2017). Soluções como essa, permitem a adequação da solução às necessidades, ao contrário de outras propostas, onde as necessidades precisam se adaptar à solução.

Nesse nível é importante que a solução adotada possua uma capacidade de escoamento dos dados compatível com o serviço que está sendo prestado. Em geral as tecnologias possuem limitação de taxa de comunicação em função de questões como:

- Baixa velocidade de transferência;
- Perda de pacote;
- Compartilhamento de grande quantidade de sensores.

Um equívoco comum é imaginar que todos os dados produzidos localmente serão transferidos para o armazenamento na nuvem. Os dados a serem transmitidos, sua frequência, possíveis tratamentos, são definidos de acordo com a necessidade do negócio e das tecnologias escolhidas.

Deve ser possível nessa etapa, transmitir um dado coletado, como, por exemplo, uma temperatura. Receber a parametrização da temperatura para definição de novo comportamento local no Nível 1. E ainda, transmitir as ações de controle enviadas pelo cliente, no caso de necessidades específicas, como parar de atuar na temperatura e desligar o dispositivo.

#### **4.2.4. Nível 3 – Nível de borda**

Como os dados coletados são provenientes de uma RSSF, mas tem como objetivo final a Internet, precisa-se de um elemento que realize a ligação desses dois universos. Elemento normalmente chamado de *Gateway*. Porém um *Gateway* é um elemento que apenas converte uma tecnologia em outra, ou, liga uma rede em outra.

No presente modelo, seu papel é muito maior que isso, ele realiza a ligação da rede local com a Internet, mas também provê outros aspectos, como compreensão de dados, armazenamento local e priorização de processos.

Está presente também nesse elemento um software que faz o papel de *middleware*. Nesse software serão configuradas as estratégias de armazenamento local, tratamento dos dados e encaminhamento para a nuvem.

Por meio do *middleware* pode-se prover a gerência dos dados e da própria rede, essa divisão e controle é necessária para se chegar a uma solução madura, que permita a criação de serviços que possuam qualidade (QoS) e a capacidade de criar os acordos de níveis de serviço (SLA).

Pode-se receber uma leitura de temperatura e analisar sua consistência, realizar cálculos para estabelecer tendências, utilizar regras, para por exemplo, ligar ou desligar uma ventoinha para otimização do processo.

O principal diferencial da presente proposta em relação às soluções atuais é a presença desse elemento como um armazenamento temporário dos dados localmente, é necessário em caso de perda de comunicação com a Internet, armazenar os dados, por um período configurável, até que a conexão seja reestabelecida.

Já existe a capacidade de acessar a Internet e verificar, por exemplo, alguma alteração no clima, que possa de alguma forma, afetar o processo e garantir uma atuação proativa. O que permite tomar uma ação de controle do processo, caso exista alguma variação brusca da temperatura. É importante destacar que essa ação é executada com base nas regras oriundas do nível de abstração.

#### 4.2.5. Nível 4 – Nível de armazenamento

Uma vez que os dados chegaram na Internet, não basta exibi-los em tempo real, é preciso armazená-los, para isso será preciso dois elementos:

– **Servidor na nuvem:** Em geral é contratado para que não exista a dificuldade de gerenciar o funcionamento deste armazenamento. Além da facilidade de alterar a capacidade do mesmo de maneira simples, geralmente alterando o plano de contratação;

– **Capacidade de armazenamento:** O principal impedimento nesse quesito, são os custos envolvidos.

Uma vez que ao adotar uma solução na nuvem, a cobrança é realizada de acordo com a utilização e ao adotar uma solução de armazenamento próprio, surgem outras questões como *backup*, capacidade de ampliação do espaço de acordo com a necessidade, consumo de energia entre outros.

Porém, para algumas empresas, por questões de segurança e criticidade dos dados envolvidos, justifica-se a necessidade de um armazenamento próprio.

A solução escolhida deve ser capaz de armazenar, por exemplo, todas as leituras de temperatura e umidade enviadas pelo elemento de borda. É interessante também, dependendo da área de atuação, que sejam mantidas séries históricas longas, como 10 anos de coletas, considerando uma medida de temperatura feita a cada minuto, ou históricos anuais.

Pode-se ainda, se a solução permitir, criar elementos que representam situações do universo das “coisas”, por exemplo, armazenar as atuações em uma caldeira, para avaliar o desgaste ao longo do tempo.

#### 4.2.6. Nível 5 – Nível de abstração

Nesse ponto os dados são transformados em informações, para isso é preciso dois elementos:

– **Abstração:** Capacidade de tratar os dados e deles extrair informações dos processos;

– **Especialista:** Elemento externo que determina a estratégia de transformação dos dados em informação.

A abstração deve ser capaz, de, por meio do tratamento dos dados brutos, retirar informações úteis para o negócio.

O especialista é alguém, ou empresa, que seja capaz de gerar as regras para retirar as informações dos dados brutos. O especialista pode ser alguém da própria empresa ou um consultor externo especialista no negócio.

Situações, como, por exemplo, fazer a correlação da temperatura e umidade do processo, com a temperatura e umidade ambiente coletada pela Internet na região, ou, avaliar a produtividade do processo, por meio da análise do número de atuações em função do tempo, são contempladas nesse nível.

#### **4.2.7. Nível 6 – Nível de apresentação**

No nível 6 é estabelecida a interface com o cliente final, é definido como as informações serão apresentadas, se existem alertas e como eles se comportam, e, se esses dados serão utilizados para interação com outras ferramentas.

Faz-se necessária a capacidade de representar as informações de forma útil para o negócio. É preciso representar as informações na interface, respeitando as regras do negócio estabelecidas no nível anterior.

Esta exibição também está ligada a um especialista, uma vez que para cada tipo de atividade existem necessidades diferentes, como por exemplo, a utilização de gráficos e terminologias adequados para aquela área. Telas, Gráficos, Ícones, Alarmes e Alertas, em forma de mensagens e sonoros fazem parte da interface.

Pode-se visualizar, por exemplo, um gráfico com a correlação entre a produção e a variação da temperatura, ou, um gráfico com a correlação entre a temperatura interna da casa de vegetação e a temperatura e umidade ambiente. E ainda receber alertas sobre aquecimento ou variações da temperatura acima do esperado.

#### 4.2.8. Gerência e segurança

De forma vertical, deve-se ter a preocupação com as questões de gerência e segurança. Como dito anteriormente, não é possível atingir níveis de QoS e atender requisitos de SLA sem possuir mecanismos de controle da solução. A gerência é dividida em duas partes:

– **Gerência da Rede:** Responsável pelos aspectos relativos à operação da rede nos seus diversos níveis. Garante o bom funcionamento por meio de avaliação de métricas para determinar a qualidade de serviço (QoS) e dá suporte ao nível de serviço (SLA);

– **Gerência dos Dados:** Responsável pelos dados nos diversos níveis. Interage com cada um dos níveis para garantir que os dados coletados, processados, abstraídos e exibidos estejam em conformidade com as necessidades dos clientes.

Nos últimos anos, conforme previsto pelos especialistas (CONVERGENCIA, 2018), (ECOMMERCENEWS, 2018), (IDGNOW, 2018), o universo de IoT se tornou alvo de inúmeros ataques. Pensar em uma estratégia de segurança é essencial para qualquer modelo atual. A segurança é tratada em duas partes:

– **Nível Local:** Trata-se da segurança desde a coleta do dado, até que a mesma alcance o nível de borda. Refere-se à segurança relacionada à RSSF;

– **Nível Internet:** Nesse nível a preocupação é proteger a informação desde o momento que ela sai do nível de borda até o armazenamento do dado no nível de aplicação. Refere-se à segurança na Internet, da transmissão ao armazenamento.

### 4.3. Exemplo de um negócio hipotético

Para demonstrar a aplicabilidade do modelo de referência, será proposto um negócio hipotético, supondo um negócio que necessite controlar uma temperatura.

Pode-se imaginar algum processo industrial que necessite desse controle. Mas nesse caso, o negócio possui uma particularidade, ele tem necessidades dinâmicas, é preciso alterar as características do processo em função de fatores externos.

Uma vez que a regra de negócio foi estabelecida, o próximo passo é identificar as “coisas”:

- Sensor de temperatura;
- Um sistema ventilação.

Um produto “X”, por exemplo, tomate cereja, pode ter seu crescimento variado pelas condições de um sistema de evapotranspiração, aplicado em uma casa de vegetação, uma estufa, em termos leigos.

Porém a montagem e parametrização do sistema, as variações de condicionamento, como acelerar ou retardar o crescimento do tomate, entre outros fatores, só podem ser definidos por um especialista que entende dos processos envolvidos e sua relação com o negócio.

Foi estabelecido também, como regra de negócio que a previsão do tempo precisa ser consultada para a tomada de decisão de aumentar ou diminuir essa variação.

Deve-se variar, uma série de fatores, e a solução deve correlacionar como essas variáveis vão influenciar a qualidade do produto. Porém para a demonstração prática da aplicação, vamos considerar apenas a umidade e temperatura em função do tempo.

A execução dos processos deve estar em sintonia com o atendimento das regras de negócio, que permitam, por exemplo, retardar o processo em função do valor do mercado do produto. Enquanto que a apresentação das informações deve refletir as necessidades do negócio.

#### **4.3.1. Nível 0 – Nível das Coisas – Exemplo de Aplicação**

O nível zero tem dois sensores de temperatura e umidade, e, uma ventoinha que será acionada de acordo com a necessidade.

Uma casa de vegetação real possui muitos elementos que devem ser controlados, mas para o experimento proposto, o foco será apenas na temperatura e umidade.

Para o negócio, este aparato serve para influenciar alguma variável do processo, por exemplo, regular a temperatura interna da casa de vegetação  $0,5^{\circ}\text{C}$  abaixo da temperatura ambiente, como regra geral, e coletar a variação da umidade em relação à temperatura. A velocidade da queda da temperatura influi no crescimento de algum produto que está sendo cultivado, tomate cereja, por exemplo.

A regra geral de ligar e desligar com  $0,5^{\circ}\text{C}$  atende minimamente ao que se espera do processo. Entretanto, se for analisado o tempo de queda da temperatura, pode ser necessário variar a velocidade da ventoinha, para aumentar este tempo, evitando cenários onde não podem ocorrer quedas bruscas de temperatura, cenário, entretanto, que não será implementado.

#### **4.3.2. Nível 1 – Nível de dispositivo, condicionamento e controle – Exemplo de Aplicação**

Localmente vai existir uma regra em que ao ser ligado, o sensor mede a temperatura ambiente externa, subtrai  $0,5^{\circ}\text{C}$  e liga a ventoinha, esse é o limite inicial, quando a temperatura interna baixa alcançando o valor limite, a ventoinha é desligada.

É importante observar que este controle independe dos outros níveis, sendo executado só localmente, isto significa que se não existirem os outros níveis, o processo vai continuar funcionando com estas regras, assim como ocorre em um processo tradicional de automação.

Esse comportamento padrão, é extremamente importante para casos de perda de conexão com a Internet, ou mesmo, de problemas de comunicação com o elemento de borda.

Podem existir algoritmos locais para verificação do valor correto da temperatura, obtendo, por exemplo, 10 amostras e realizando a média ou utilizando um filtro de média móvel baseada em um tempo específico, 1 minuto, por exemplo.

A criação do aparato, sensores de temperatura e ventoinha, foi determinada por um especialista. Também foi este especialista que determinou, como regra geral, ligar a ventoinha para que a temperatura interna esteja 0,5° C abaixo em relação à temperatura externa.

Localmente é possível realizar outros cálculos que sejam do interesse de outros níveis, como medir o tempo médio entre ligar e desligar a ventoinha.

O Nível 1, para atender ao negócio, deve fornecer para os níveis superiores à temperatura e umidade interna média nos últimos 10 ciclos. Já o Operador de RSSF, visando a gerência da rede, solicitou as seguintes informações:

- Tempo para execução dos 10 ciclos;
- Tempo médio da ventoinha ligada;
- Tempo médio da ventoinha desligada.

#### **4.3.3. Nível 2 – Nível de conectividade – Exemplo de Aplicação**

Nesse nível obtém-se a capacidade de comunicação com o elemento de borda. Em termos práticos, é possível:

- Realizar a transmissão da temperatura e umidade;
- Receber a parametrização da temperatura, para definição de um novo comportamento local, no Nível 1. Por exemplo, ligar a ventoinha para que a temperatura interna caia de 1,0° C;
- Transmitir as ações de controle enviadas pelo cliente, no caso de necessidades específicas, como parar de atuar na temperatura e desligar o dispositivo.

#### **4.3.4. Nível 3 – Nível de borda – Exemplo de Aplicação**

Neste nível é analisada a performance do processo, para isso ele recebe algumas informações do nível 1:

- Temperatura e umidade média nos últimos 10 ciclos;
- Tempo para execução dos 10 ciclos;
- Tempo médio da ventoinha ligada;
- Tempo médio da ventoinha desligada.

Também devem ser tomadas decisões com base nos resultados recebidos, por exemplo, mudar o delta de temperatura interna de 0,5° C para 1,0° C.

#### **4.3.5. Nível 4 – Nível de armazenamento – Exemplo de Aplicação**

Aqui são armazenadas as informações coletadas, na presente proposta, isso se dá por meio da implementação de um servidor na nuvem. Os dados armazenados correspondem aos dados coletados para atender as necessidades do cliente:

- Temperatura e umidade internas da casa de vegetação;

Também o que for necessário para alimentar os níveis superiores e auxiliar na tomada de decisão e validação das regras de negócio:

- Tempo médio para baixar de 0,5° C;
- Tempo médio para baixar de 1,0° C, quando esta estratégia for necessária;
- Número de ciclos por hora para 0,5° C;
- Número de ciclos por hora para 1,0° C;

Se estes tempos não forem suficientes para atender as demandas do negócio, será preciso revisar o hardware escolhido. Pois é o negócio quem determina a necessidade.

Como o foco da presente proposta são as pequenas e médias empresas, pode-se chegar à conclusão que, devido ao custo de um hardware mais especializado ou preciso ser muito alto, será necessário alterar o processo de produção do produto, gerando novas regras de negócio, o que automaticamente exige a revisão da solução de IoT.

Nesse nível também estão presentes as informações necessárias para garantir o funcionamento contínuo da solução:

- Tempo em que a ventoinha ficou ligada;
- Funcionamento dos sensores de temperatura e umidade.

#### **4.3.6. Nível 5 – Nível de abstração – Exemplo de Aplicação**

Com base nos dados coletados, pode-se gerar informações que serão necessárias para o Operador de RSSF garantir o funcionamento da rede:

– Calcular o tempo médio de utilização da ventoinha e inferir o número previsto de horas que a ventoinha estará ligada no próximo ano.

Com base no histórico de utilização e na previsão para o ano seguinte, comparar essas informações com as especificações fornecidas pelo fabricante da ventoinha, e, prever a data aproximada da substituição da mesma, evitando assim interrupções na solução.

Além de gerar informações que podem ser úteis para o negócio:

– Correlaciona o número de vezes que liga e desliga em função da temperatura ambiente;

– Gerar ações para a chegada de uma frente fria obtida por meio da previsão do tempo na Internet.

Com base nesses indicadores novas estratégias de produção podem ser adotadas.

#### **4.3.7. Nível 6 – Nível de apresentação – Exemplo de Aplicação**

Para atender as demandas do negócio foram criados uma série de gráficos para demonstração do ambiente:

- Variação da temperatura ambiente ao longo do tempo;
- Variação da temperatura e umidade interna ao longo do tempo;
- Estado da ventoinha: ligada ou desligada;
- Gráfico correlacionando a temperatura e umidade com o estado da ventoinha.

Também são configurados alertas que acionam *triggers* em determinadas situações:

- Mudanças bruscas de tempo dos processos;
- Problema na ventoinha;
- Temperatura acima do limite.

É importante destacar que os alertas devem ser configurados com níveis diferentes de criticidade.

Na ocorrência de uma *trigger*, pode-se acionar determinadas ações pré-configuradas, como por exemplo:

- Enviar e-mail com detalhes dos alertas;
- Disparar um alerta sonoro na interface;
- Enviar uma mensagem para o aplicativo TELEGRAM (2017);
- Acionar um atuador, como por exemplo, um LED de alerta, uma sirene, um rele que desliga o equipamento.

#### **4.3.8. Gerência e segurança – Exemplo de Aplicação**

As questões de gerência da rede foram tratadas por meio de coletas do funcionamento do ambiente:

- Estado da ventoinha: ligada ou desligada;
- Delta de tempo sem receber dados da ventoinha ou temperatura e umidade.

São gerados gráficos, *triggers* e notificações para essas situações. As questões de segurança não foram tratadas no experimento, mas serão dados exemplos na seção 5.1.6.

## **5. IMPLEMENTAÇÃO DE UMA SOLUÇÃO OPEN SOURCE BASEADA NO MODELO DE REFERÊNCIA**

Como já citado, o objetivo do trabalho é criar um modelo de referência de IoT, e como validação do modelo implementar uma solução que abrange desde as necessidades do negócio até a implementação das soluções, utilizando soluções *Open Source* voltada para o mercado de Pequenas e Médias Empresas.

Nessa seção são discutidas as soluções adotadas, mas nada impede que outras soluções sejam utilizadas, para um ou mais elementos da arquitetura, criando assim sua própria implementação.

Porém, para demonstração dos conceitos, fez-se necessário a montagem de uma plataforma, na qual optou-se exclusivamente por tecnologias de código fonte aberto.

### **5.1. Estudos de Casos**

O modelo nasce de várias experimentações ocorridas no grupo de trabalho no qual o mestrando está inserido. Nesta seção serão apresentados os casos ocorridos.

Vale ressaltar que o mestrando contribuiu com todos os casos citados, ainda que não tenha implementado todos eles. Outro ponto a ser destacado é que o modelo de referência nasce das necessidades enfrentadas durante a elaboração e execução destes casos. Em muitos deles o modelo era inexistente ou parcialmente existente.

#### **5.1.1. Caso Fazenda Cafeeira**

Este projeto surgiu da necessidade de monitoração especificada por um produtor de café da região Sul do estado de Minas Gerais em uma área de 32 alqueires. A quantidade e periodicidade da irrigação nos cafezais são calculadas em função da umidade do solo e da temperatura e umidade relativa do ar.

Considerando-se que em um futuro próximo o sistema de outorga será substituído pela medição do volume de água retirado dos mananciais, o acompanhamento destas informações proporcionará economia de recursos naturais e financeiros.

O objetivo primário desse caso foi validar a cobertura do sinal em uma área abrangente e verificar a estratégia de envio dos dados para a Internet por meio do IoT *Proxy Manager* (IoT PM), que será detalhado na seção 5.2.4.

Ao constatar que a solução funcionava e que sua implementação não demandou grandes ajustes, foi proposto ao responsável pela fazenda a coleta de determinadas grandezas como temperatura, umidade e umidade do solo em dois pontos distintos.

Além disso, foi instalado um GPS (*Global Positioning System*) em um trator da fazenda, no qual os dados eram coletados, e via IoT PM, atualizava-se a posição do trator na ferramenta de gerência a cada nova leitura.

### **5.1.2. Caso Campo de Futebol – PUC Campinas**

Este projeto surgiu da necessidade da própria universidade em monitorar e controlar a irrigação do campo de futebol. A irrigação atual é feita por meio de aspersores com irrigação automatizada. Os funcionários responsáveis por essa tarefa podem agendar para que a irrigação aconteça três vezes por semana, por exemplo, porém mesmo que esteja chovendo, a programação será executada. Essa situação produz uma série de problemas:

- As condições do campo podem ou não estar adequadas para a prática de esportes;
- A grama pode ou não receber a quantidade de irrigação necessária, e caso receba em excesso também é um problema;
- Os gastos com o sistema atual são extremamente elevados;
- Existe uma preocupação com as questões de meio ambiente, sendo a água um recurso tão precioso.

### **5.1.3. Caso Fila do Restaurante Universitário – UNICAMP**

SILVA (2017) participou do concurso Smart Campus, da Faculdade de Engenharia Elétrica e de Computação (FEEC) na Unicamp, em que propôs uma solução baseada Redes de Sensores sem Fio aplicadas no conceito de Internet das Coisas; a plataforma escolhida para a exibição e análise das informações foi o Zabbix.

O objetivo do trabalho foi identificar o nível de ocupação das filas dos restaurantes universitários de acordo com os dados colhidos por meio da RSSF. Como a universidade possui três restaurantes disponíveis aos alunos, eles poderiam optar pelo de menor fila, ou mesmo aguardar o horário com menos procura para se dirigir ao restaurante.

Os dados colhidos ao longo do tempo, poderiam ainda ser utilizados para adequações no negócio. Este trabalho foi um dentre os três vencedores do concurso e foi apresentado no Inova Campinas em outubro de 2017 (INOVA, 2017).

### **5.1.4. Caso Monitoramento Ambiental para Condições de Emergência**

TOMASINI et al. (2017) abordou a questão da necessidade de monitoramento e diagnóstico de áreas de risco ou situações nas quais o monitoramento pode alertar sobre condições emergenciais causadas por desastres naturais. A principal dificuldade encontrada nessas situações é a questão de alimentação energética.

Como alternativa ao problema é proposto a utilização de RSSF com estratégias de otimização da bateria (*SleepMode*), que é basicamente um modo de operação em que são desabilitados praticamente todos os módulos do nó sensor. O que resulta em um consumo de energia muito baixo.

O Zabbix foi adotado como servidor de aplicação, armazenando todos os dados coletados, no total o experimento totalizou aproximadamente 14 dias de coletas ininterruptas, sendo destes dez dias de tempo nublado e chuvoso.

### 5.1.5. Caso Protocolo repetidor para estender a cobertura de sinal

Matthiesen et al. (2017) discorrem sobre as vantagens de utilização de um repetidor em RSSF para expandir a área de cobertura da rede. O ponto principal do artigo a ser destacada no presente trabalho, são os benefícios que a utilização do Zabbix, nesse cenário, traz ao operador da rede:

“Usando o Zabbix, as estatísticas serão mais fáceis, sendo compiladas enquanto os testes estiverem em execução, estimando a média do sinal e seu desvio padrão.”, (MATTHIESEN et al., 2017, p. 4).

### 5.1.6. Caso Segurança

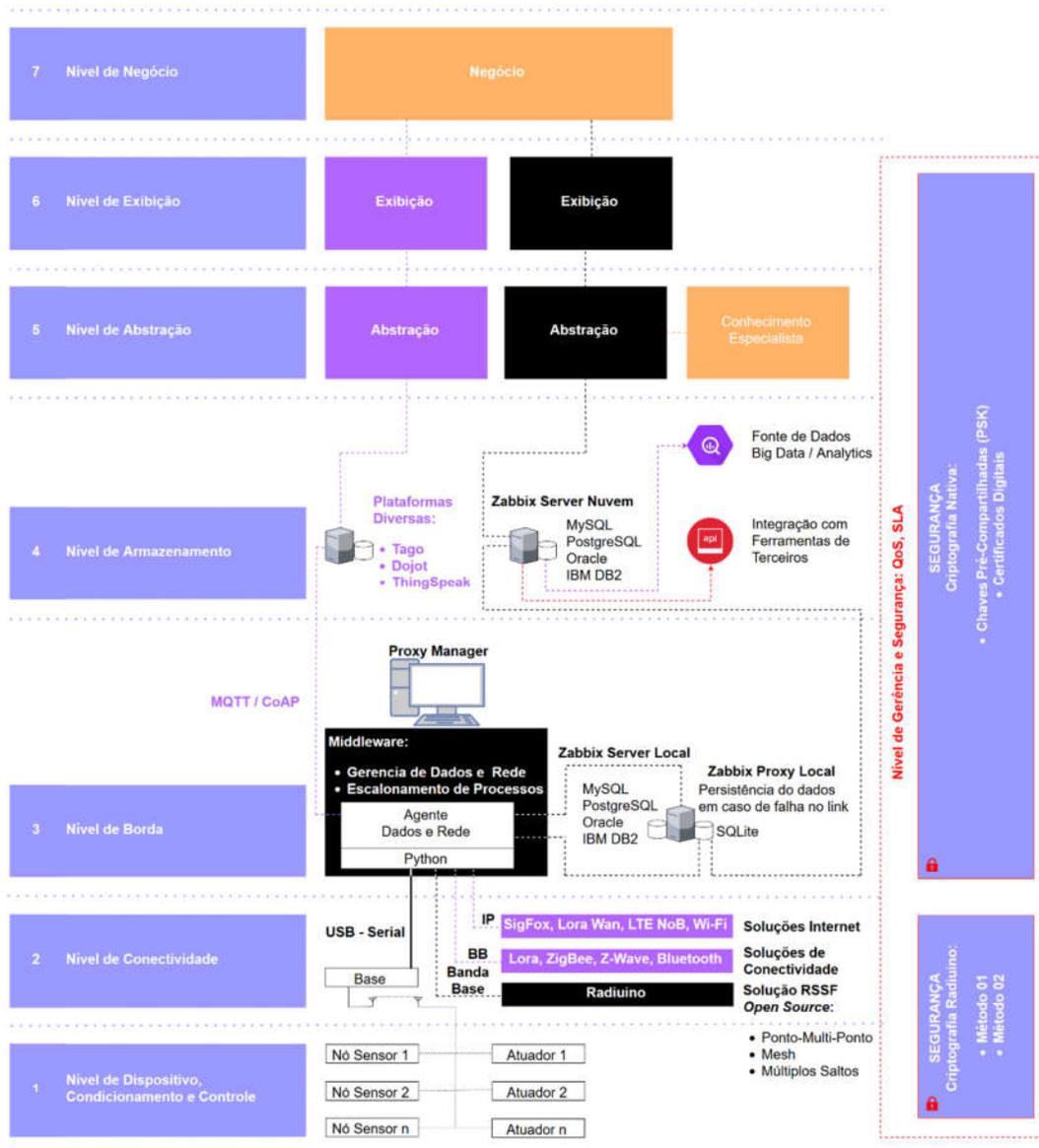
FREITAS propõe duas soluções de segurança para o protocolo de RSSF Radiuino. O primeiro método é o uso de uma cifragem polialfabética, denominada “or exclusivo”; esta solução é detalhada em FREITAS e BRANQUINHO (2017).

O trabalho envolve ainda o desenvolvimento de um *sniffer* para RSSF, que é um programa utilizado para capturar e inspecionar dados trafegados em redes, que foi utilizado para demonstrar a eficiência do método. A segunda proposta, FREITAS (2018), é a utilização da técnica de criptografia *One Time Pad*, na qual envolve um algoritmo criptográfico simétrico e uma função *hash*.

## 5.2. Arquitetura Implementada

Na Figura 20 observa-se o Modelo de Referência, é possível observar todos os níveis descritos, com o incremento de alguns exemplos de tecnologias e soluções disponíveis no mercado. Na seção 5.2.4 será realizada uma análise desses elementos adicionais.

Figura 20 - Visão Funcional de um Modelo de Referência de IoT



Fonte: Elaboração Própria

Na seção 5.1 foram apresentados vários casos, nas próximas seções serão analisados os elementos de cada nível de acordo com o modelo de referência. Os casos serão citados de acordo com a relevância para o nível.

O objetivo dessa seção é evidenciar que mesmo utilizando a mesma tecnologia de RSSF e ferramenta de apresentação de dados, as necessidades são distintas e a divisão em níveis proposta pelo modelo de referência possibilitam um melhor entendimento destas.

### **5.2.1. Nível 0 – Nível das Coisas**

Como citado ao longo do trabalho, as “coisas” variam de acordo com o tipo de negócio, área de atuação, objetivos da solução e etc. Nessa seção serão analisadas as necessidades apresentadas durante o desenvolvimento dos projetos.

Caso Fazenda Cafeeira:

- Umidade do Solo;
- Cobertura de Sinal;
- Posição geográfica do trator.

Caso Campo de Futebol – PUC Campinas:

- Umidade e Temperatura do ar;
- Umidade do Solo.

Caso Fila do Restaurante Universitário – UNICAMP:

- Tamanho das filas dos restaurantes universitários.

Caso Monitoramento Ambiental para Condições de Emergência:

- Umidade do Solo;
- Nível de bateria.

Caso Protocolo repetidor para estender a cobertura de sinal:

– Cobertura de Sinal.

Caso Segurança:

– Dados transmitidos e recebidos.

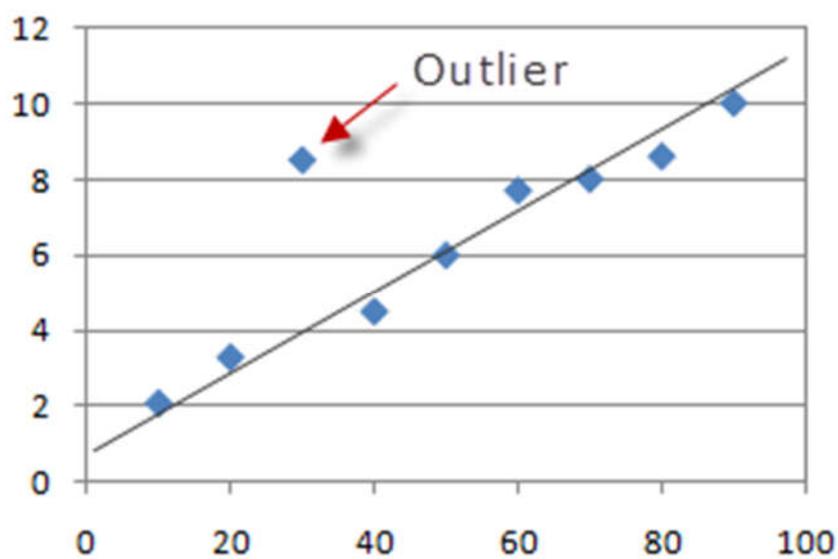
### 5.2.2. Nível 1 – Nível de dispositivo, condicionamento e controle

Nesse nível são abordadas as questões relativas ao condicionamento e controle dos dados, pode-se traduzir isso como o zelo pelo dado. Nesse momento questões como precisão, acurácia e *outliers* precisam ser abordadas.

Ao coletar uma informação por um determinado período, observa-se uma padronização dos dados, uma temperatura sofre variações ao longo do dia, porém, não é esperado que de uma leitura para outra aconteçam variações extremamente bruscas e fora da faixa. Por exemplo, uma leitura informa 22,3° C, enquanto que a leitura seguinte, informa 315° C.

Esses casos são considerados *outliers*, como pode-se observar na Figura 21 o dado recebido é totalmente destoante em relação ao comportamento das demais leituras.

Figura 21 – Exemplo de Outlier



Fonte: MATH, 2018

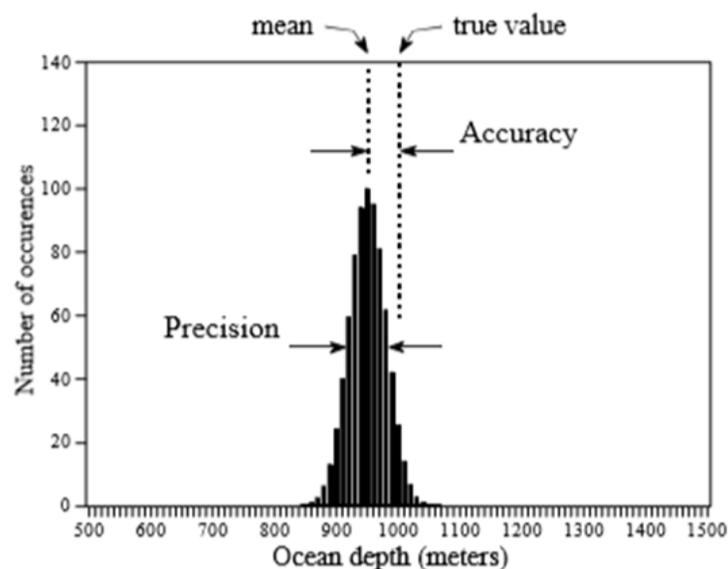
Neste quesito dois pontos devem ser observados, primeiro, que localmente existe um microcontrolador que possui uma certa capacidade de processamento. Segundo que independentemente da tecnologia escolhida o canal de transmissão de dados será sempre o elemento mais sensível e limitado da solução.

Ao utilizar a tecnologia SigFox, por exemplo, possui-se a limitação de envio de apenas seis mensagens de 12 bytes por hora ou cento e quarenta mensagens por dia. Essa limitação se deve ao regulamento da Europa, que afirma que a faixa pública só pode ser utilizada 1% do tempo; e a decisão da empresa é de manter a limitação para todos os países independente da legislação local, (SIGFOX, 2018b).

Este é um exemplo claro de uma situação, onde o poder de processamento do microcontrolador deve ser utilizado para descartar os *outliers*, poupando o canal de transmissão de um dado que não reflita a realidade do ambiente, exceto em casos onde o especialista ou as regras do negócio especificuem que eles devem ser encaminhados.

Outro caso, em que o poder de processamento do microcontrolador pode ser aproveitado, é na questão de condicionamento dos dados, para realizar ajustes de precisão e acurácia das coletas. A Figura 22 demonstra esses conceitos.

Figura 22 - Precisão e Acurácia



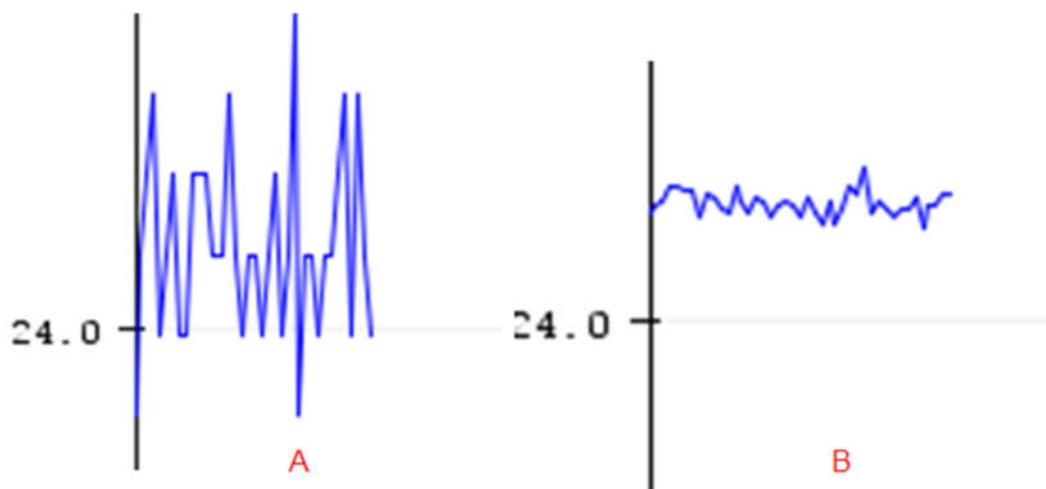
Fonte: SMITH, 2006

De maneira bem simples pode-se definir precisão como a variação das coletas ao longo do tempo, em relação à média. Enquanto que a acurácia é a diferença entre a média e o valor real coletado.

Para demonstrar o poder do microcontrolador, observe que o mesmo é capaz de coletar um dado a cada milissegundo, agrupar cem coletas e calcular a média, isso resulta seis médias por minuto. Transmitir um valor a cada dez segundos, provavelmente não será necessário na maioria dos casos, um valor por minuto, presumivelmente atende as demandas da maioria das PME.

Porém como demonstração, partindo de um cenário onde ocorre uma coleta a cada dez segundos, observa-se na Figura 23A os dados sendo enviados após a coleta, sem condicionamento algum, e, na Figura 23B os dados condicionados pelo microcontrolador, ou seja, são enviadas apenas as médias.

*Figura 23 - Exemplo de Condicionamento*



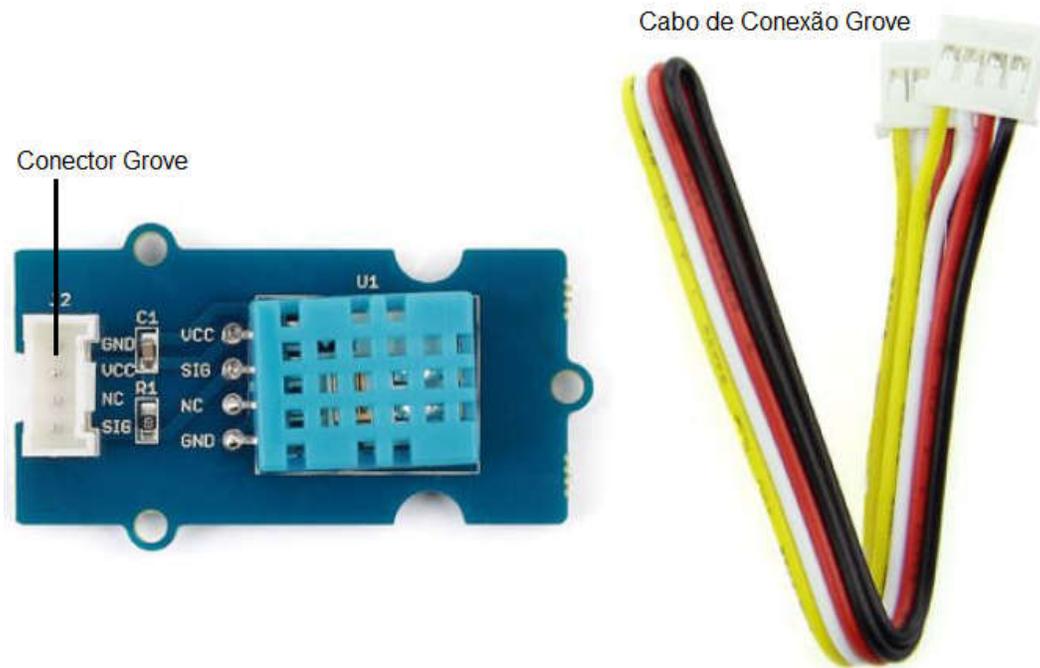
Fonte: Elaboração Própria

Observa-se que aplicando o condicionamento e controle, é possível obter uma exibição dos dados mais próxima da realidade do ambiente. Como já citado na seção 4.2.2, não utilizar esta capacidade é transferir a responsabilidade para outra entidade, que possuem outras funções, como por exemplo, sobrecarregar o canal de comunicação.

## Sensores

Para realizar a coleta dos dados é preciso utilizar nós sensores (NS). Dentre as opções disponíveis, optou-se pela utilização de sensores com conectores GROVE (2017), devido à facilidade de conexão e substituição dos mesmos. Na Figura 24 observa-se um sensor de temperatura e umidade, modelo DHT11.

Figura 24 – Sensor para coleta de Temperatura e Umidade com conectores do tipo Grove



Fonte: Adaptado de SOLARBOTICS, 2017

### A. Sensor Digital

O sensor digital é ligado diretamente em uma porta digital e entrega uma sequência de bits. No caso do DHT11, o *datasheet* (AKIZUKIDENSHI, 2018) informa que essa sequência é de 40 bits, sendo divididos da seguinte forma:

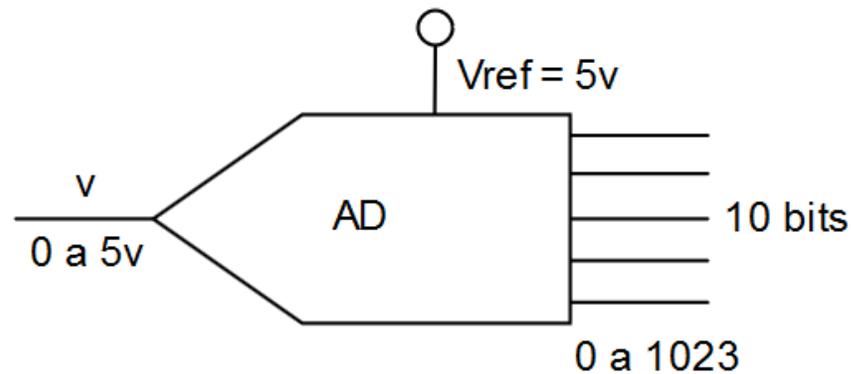
- 8 bits para a parte inteira da umidade;
- 8 bits para a parte decimal da umidade;
- 8 bits para a parte inteira da temperatura;
- 8 bits para a parte decimal da temperatura;
- 8 bits para a paridade

- A sequência recebida para que a leitura do sinal seja considerada válida é 0100 1101.

### B. Sensor Analógico

Os sensores analógicos são ligados em uma porta de conversão de analógico para digital (AD), que converte a variação de tensão em uma variação de bits, conforme representado na Figura 25.

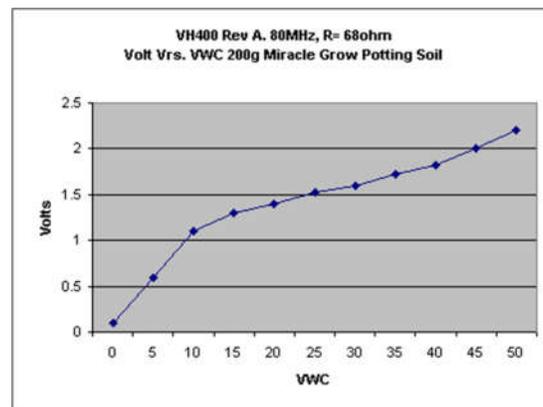
Figura 25 - Conversão de Variação de Tensão para Sinais Digitais



Fonte: Elaboração Própria

Para calcular valor da leitura, pode-se existir situações distintas, de acordo com o modelo do sensor. No *datasheet* do sensor vegetronix VH400 (VEGETRONIX, 2018), observa-se que as medidas são separadas em 5 faixas de leitura, o próprio documento já fornece a fórmula a ser aplicada. E um exemplo de medição, Figura 26.

Figura 26 – Medição de Nível de Umidade do Solo com Vegetronix



Fonte: VEGETRONIX, 2018

Bastando assim, coletar apenas a medida de tensão e depois realizar o cálculo adequado, de acordo com a faixa.

### C. Sensores Utilizados nos Casos

Na Tabela 1 pode-se visualizar os sensores que foram utilizados em cada um dos casos. Os casos não citados são os que não necessitaram de sensores.

*Tabela 1 – Sensores Utilizados no Casos*

Caso	Sensores
Fazenda Cafeeira	VH400, Módulo GPS, DHT22
Campo de Futebol – PUC Campinas	VH400, DHT22
Monitoramento Ambiental para Condições de Emergência	VH400, DHT22
Segurança	DHT22

Fonte: Elaboração Própria

#### 5.2.3. Nível 2 – Nível de conectividade

Este é o nível responsável por levar os dados até o elemento de borda, em geral possui limitação de taxa de comunicação, em função de questões como baixa velocidade de transferência, perda de pacotes e compartilhamento da banda com centenas de sensores.

Portanto é muito importante escolher uma tecnologia com capacidade de escoamento de dados compatível com o serviço que está sendo prestado.

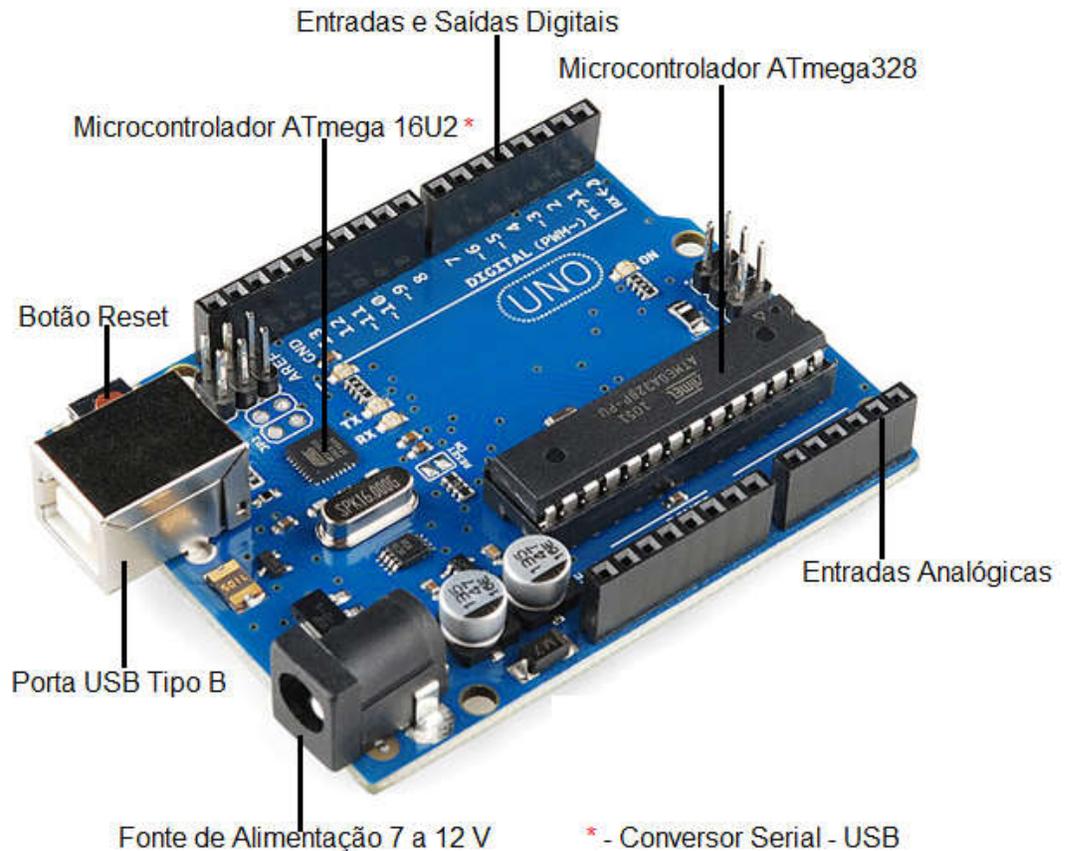
#### Plataforma de Prototipagem Eletrônica de Hardware Livre e de Placa Única

Arduino é uma plataforma de prototipagem eletrônica, constituída de hardware e software, ambos livres. Implementado em uma placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++.

O microcontrolador possui a capacidade computacional para realizar uma série de tratativas das informações coletadas pelos sensores. Após isso

é realizada a transmissão dos dados via um meio pré-determinado. Os elementos principais da placa podem ser observados na Figura 27.

Figura 27 - Arduino Uno



Fonte: Adaptado de FILIPEFLOP

### Transmissão

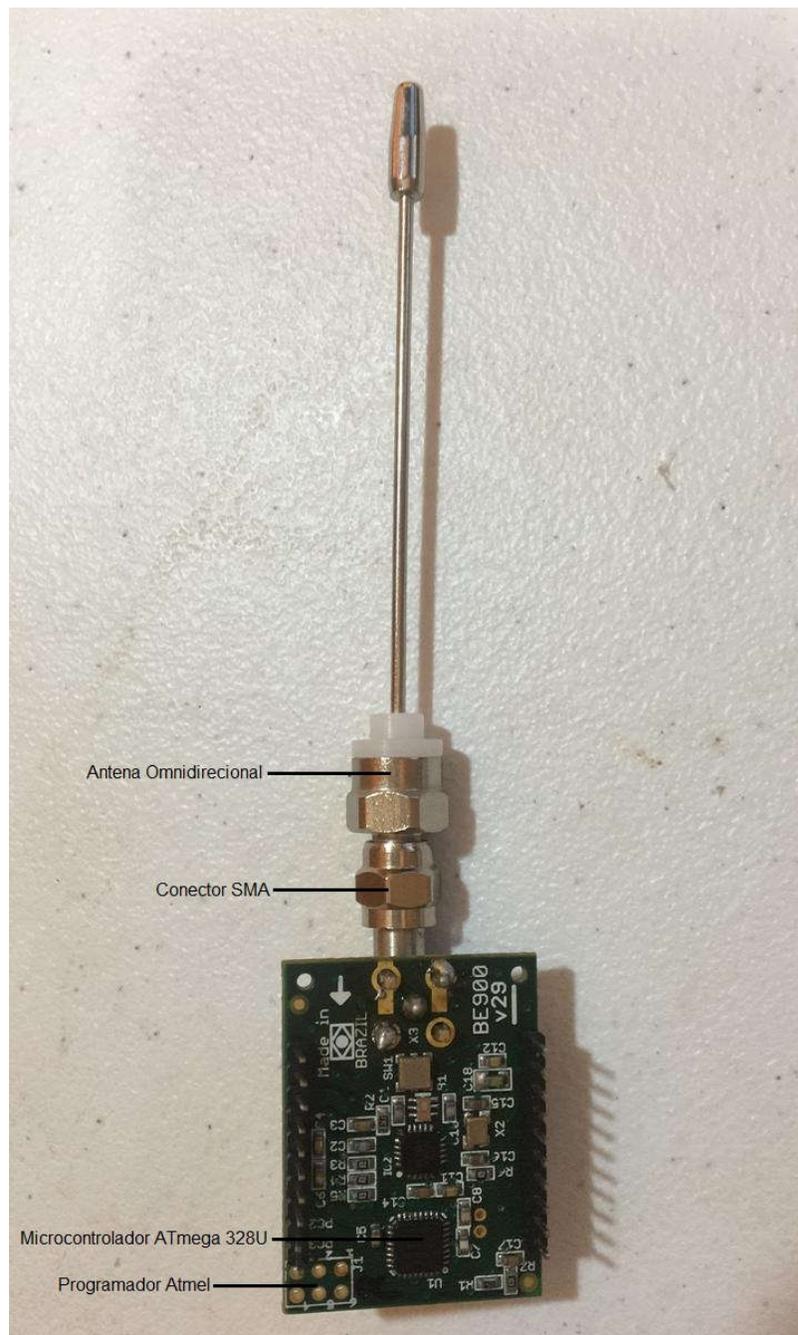
Visando a utilização de RSSF foi escolhido como método de transmissão links de rádio, por meio de módulos BE-900 e protocolo Radiuino.

Na Base e nos NS, foram utilizados módulos de rádio BE-900 que utiliza o processador AVR ATmega328P e o Transceptor TI CC1101 RF.

Os rádios foram ajustados para operar na banda não licenciada ISM (*Industrial, Scientific and Medical*) de 915MHz (902-907,5MHz e 915-928MHz), o que atende às regulamentações da ANATEL, os módulos podem

atingir uma taxa de transmissão de até 250 Kbps (RADIOIT, 2017). Na Figura 28 observa-se uma imagem real do módulo BE900.

Figura 28 - Módulo de rádio BE900



Fonte: Elaboração Própria

### Protocolo de Comunicação de Rede de Sensores Sem Fio

A comunicação dos NS com a base se dá por meio do protocolo de RSSF, denominado Radiuino (RADIUINO, 2017). Podem ser utilizadas outras tecnologias de comunicação sem fio, inclusive proprietárias, mas esse não é o foco deste trabalho.

O Radiuino pode ser definido como um protocolo de comunicação para Redes de Sensores Sem Fio, que implementa um pacote de comunicação de 52 bytes. O protocolo é organizado em 5 camadas: Física, MAC, Rede, Transporte e Aplicação. Esta estrutura é conceitualmente equivalente à pilha TCP/IP, um resumo de cada camada está disponível na Tabela 2.

*Tabela 2 - Camadas do protocolo Radiuino*

Camada	Área de Estudo
Aplicação	Funções ligadas ao desenvolvimento das aplicações diretamente, como medidas de grandezas e controle de processos.
Transporte	Funções de controle da comunicação como ACK ( <i>Acknowledgement</i> ), contagem de pacotes, disciplina de transmissão (Controle de Acesso ao Meio - MAC).
Rede	Identificação do sensor e rede, contemplando funções para roteamento de pacotes na rede.
MAC	Funções para controlar os processos de recepção e principalmente transmissão. Política de economia de energia.
Física	Funções relacionadas com a parte de rádio como: potência, canal. Possível evoluir para alterar outras características do rádio.

Fonte: Elaboração Própria

#### 5.2.4. Nível 3 – Nível de borda

Este é um dos níveis mais importantes do ponto de vista de implementação, nesse nível os dados são transformados em informações. Uma série de controles acontecem nesse nível: recebimento dos dados das camadas inferiores, tratamento e armazenamento das informações, regras de coleta, interação com as ações de condicionamento e controle e com os atuadores, envio dos dados para as camadas superiores.

##### ***IoT Proxy Manager***

Seguindo a linha de raciocínio de OLIVEIRA (2016), partiu-se de um *Proxy Manager* (PM), no qual foram acrescentadas novas funcionalidades, como por exemplo, a possibilidade de comunicação com o software de gerência de redes.

Nesse ponto, esse elemento evoluiu novamente e passou a se comportar como um *IoT Proxy Manager* (IoT PM), visto que agora ele tem condições de controlar elementos da RSSF, mas também controlar quais dados e de que maneira eles serão enviados ao software de gerência de redes.

##### **Computador de Placa Única**

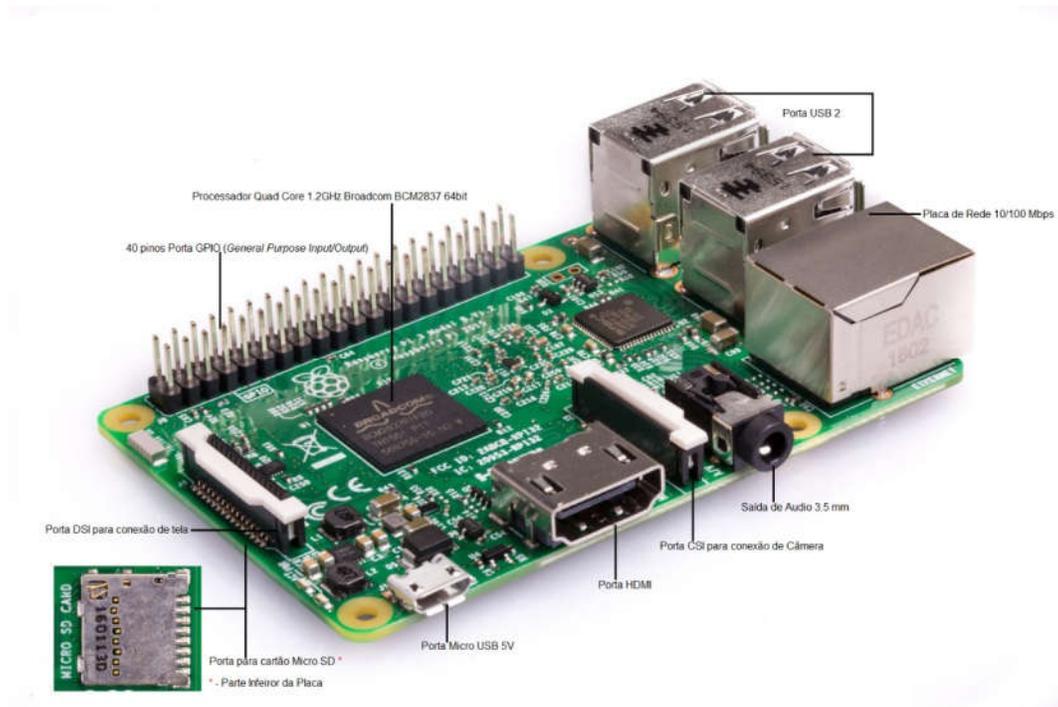
No presente trabalho, foi utilizado o computador de placa única Raspberry Pi 3, que possui especificações de hardware (RASPBerry, 2017) suficientes para a execução dos softwares necessários para o trabalho, dentre os quais uma versão do Sistema Operacional GNU/Linux, o aplicativo Zabbix e os *scripts* em linguagem de programação Python.

O Raspberry Pi 3 possui um processador Broadcom BCM2837 de 64 bits e clock de 1,2GHz, Wi-fi e Bluetooth 4.1 integrados, 1G de memória RAM, adaptador para cartão microSD e GPU Videocore IV 3D.

Suas dimensões e consumo de energia são ideais para uso em qualquer ambiente. É possível instalar diversas distribuições do Sistema Operacional GNU/Linux como o Raspbian (RASPBIAN, 2018) e Ubuntu

(UBUNTU, 2018), além do Windows 10 IoT (WINDOWS, 2018). Os principais elementos do Raspberry Pi 3 podem ser visualizados na Figura 29.

Figura 29 - Raspberry Pi3



Fonte: Adaptado de RASPBERRY

Existem outros computadores de placa única que podem ser utilizados para substituição do Raspberry, como por exemplo, o BeagleBone Black (BEAGLEBOARD, 2018), apesar de possuir hardware inferior. E o CubieBoard (CUBIEBOARD, 2018), que apesar de possuir hardware superior ao Raspberry, tem uma relação custo benefício muito maior.

### **Linguagem de Programação para Desenvolvimento dos *Scripts* do *Proxy Manager***

Dentro do computador, é executado um *script* na linguagem de programação *Open Source* PYTHON (2017), que possui a função de gerenciar a comunicação com a rede de sensores sem fio, enviar comandos para os atuadores, além de realizar alguns cálculos estatísticos e a comunicação com

o Zabbix Server. Esse *script* Python desenvolve o papel de *Middleware* e está presente no IoT PM.

Foi adotada a linguagem Python pela facilidade de compreensão da linguagem e a grande variedade de bibliotecas disponíveis.

### **Características do Zabbix**

O software de Gerenciamento de Redes Zabbix, foi escolhido por ser um software Livre, de código fonte aberto, sem versões comerciais (ZABBIX, 2017b), *true open source*, o que torna possível personalizar a ferramenta e criar estratégias que atendam às demandas de IoT das PME.

Possui um sistema de monitoramento distribuído por meio de agentes e *proxies* (ZABBIX, 2017c), que são agentes remotos, com funções adicionais, de, por exemplo, armazenar os dados localmente em caso de interrupção com a Internet, e posteriormente enviar essas informações ao servidor.

Apesar do uso de agentes e *proxies* para coleta de dados, a ferramenta possui administração centralizada via interface web, com a vantagem de que todos os dados de coletas e configurações são armazenados em uma base de dados relacional.

Uma vez que os dados estejam armazenados, é possível seu processamento e reutilização posterior, por exemplo, em sistemas de simulação, integração com outros softwares e plataformas, aplicação de *Big Data* e *Analytics*, que consiste em extrair e combinar resultados de várias fontes de dados, para descobrir padrões, correlações desconhecidas e outras informações úteis para ajudar na tomada de decisão. Apenas para citar alguns exemplos.

Apesar de o Zabbix ser considerado um NSM, existem exemplos de uso do mesmo atuando como SCADA (ZABBIX, 2016; 2017e).

O Zabbix é adotado por empresas do mundo todo (ZABBIX, 2017f), incluindo mais de 50 empresas presentes no FORTUNE500 (2017).

A Zabbix SIA (ZABBIX, 2017e), empresa responsável pelo Zabbix, tem divulgado a intenção de utilização do Zabbix como plataforma de IoT há alguns anos (UBUNTU, 2015; UNIREDE, 2017; SERVERWORKS, ZABBIX, 2017f), como consequência dessa ação, algumas palestras foram realizadas

abordando esse tema (ZABBIX, 2017d; UNICAMP, 2017; ZABBIX, 2018c; ZABBIX, 2018d).

### **A. Zabbix Server**

Esse é o elemento central da solução, é por meio dele que as informações são coletadas, processadas, gerenciadas e exibidas (HORST; PIRES; DÉO, 2015).

Ele é composto de pelo menos três elementos: Processo Zabbix Server, Banco de Dados e Interface Web.

### **B. Banco de Dados**

No banco de dados são armazenadas as informações coletadas, as configurações da ferramenta e dados históricos de auditoria e alertas, além de todos os gráficos, telas, ícones e demais elementos gráficos da solução.

Segundo o manual oficial da ferramenta (ZABBIX, 2017h) são suportadas várias soluções de banco de dados: MySQL, PostgreSQL, Oracle, IBM DB2 e SQLite. Os processos do Zabbix Server e a Interface Gráfica consultam e gravam informações no Banco de Dados.

### **C. Interface Web**

É a interface com o usuário, é por meio dessa interface que o usuário configura a maneira como o ambiente vai se comportar, as informações que deseja monitorar, os alertas que deseja visualizar e os alarmes que serão recebidos.

Também é por meio desse elemento que se torna possível acessar gráficos e telas que exibem informações sobre a coleta de dados, inclusive histórica e a saúde do ambiente.

## **Componentes do Zabbix**

Na seção anterior foram analisadas algumas das características da ferramenta que demonstram que o mesmo pode ser utilizado como uma plataforma para coleta e exibição de dados para IoT. Nessa seção serão analisados os elementos que compõem a solução.

### **A. Processo Zabbix Server**

Este é o serviço que fica em execução no servidor, são executadas múltiplas instâncias de coletores: *Pollers*, SNMP, JAVA, IPMI, VMware, dentre outros.

Eles controlam as coletas dos dados, gravação no banco de dados, exibição de alertas, envio de alarmes, comunicação com os Agentes e *Proxies*.

### **B. Coletores**

Os dados podem chegar ao Zabbix por meio de dois sentidos, quando o servidor vai até o dispositivo solicitar uma informação, processo conhecido por *Poller*, ou quando o dado é enviado ao servidor, processo conhecido como *Trapper*.

O Zabbix suporta uma série de coletores (ZABBIX, 2017i), Zabbix Agente, Checagens Simples, SNMP, Zabbix Internal, IPMI, Monitoramento JMX, dentre outros. Nas seções subsequentes serão detalhados os itens que são relevantes para a proposta.

### **C. Checagens Simples**

A monitoração com *Simple check* é a mais básica de todas, ela permite realizar um ping em um endereço TCP/IP, seja de rede ou Internet. E assim detectar se aquele endereço está acessível a partir da conexão de rede ou Internet do ambiente onde a solução está implementada.

Verificar se algum serviço TCP responde em uma porta, por exemplo, por padrão, um serviço web HTTP responde na porta 80, enquanto que um serviço web HTTPS responde na porta 443. Além de checar o serviço UDP NTP (*Network Time Protocol*).

A principal vantagem dessa checagem se dá pelo fato de não ser necessário nenhuma instalação adicional no destino.

#### **D. Zabbix Agente Passivo**

Nesse tipo de checagem a solicitação tem origem no Servidor Zabbix ou *Proxy Zabbix*, e, como destino o Agente Zabbix instalado no equipamento.

A principal vantagem dessa solução é que ela é mais simples.

#### **E. Zabbix Agente Ativo**

Nesse tipo de checagem a solicitação tem origem no Agente, e, como destino o Servidor Zabbix ou *Proxy Zabbix*.

O agente quando iniciado, envia uma solicitação a quem ele responde, solicitando uma lista de todos os itens que devem ser monitorados, bem como o intervalo de coletas. Ciclicamente essa solicitação é realizada novamente, para verificar se ocorreram alterações.

Uma vez que o agente sabe quais dados devem ser coletados, ele realiza a coleta, e então envia a informação ao destino, Zabbix Server ou *Proxy Zabbix*.

A principal vantagem dessa solução é que ela transfere parte da carga de processamento do servidor para o elemento monitorado.

#### **F. Zabbix Trapper**

Itens desse tipo são utilizados quando desejamos enviar um dado para o Zabbix, mas o *script* e/ou comando que gera o dado leva um tempo maior que 30 segundos, que é o tempo máximo de espera por um dado processado pelo agente (ZABBIX, 2017j).

A principal vantagem dessa solução está na ausência de tempo para enviar o dado, situação ideal onde utilizamos checagens de dados na Internet, interação com softwares de terceiros e conexões com bancos de dados, situações em que a mesma consulta pode variar bastante o tempo da resposta em função dos fatores envolvidos.

### **Zabbix Proxy**

Pode-se entender esse elemento como um superagente, ele coleta dados em nome do Servidor Zabbix, com um desempenho de milhares de dados por segundo com um consumo de hardware modesto (HORST; PIRES; DÉO, 2015).

Apesar de ser um elemento opcional na arquitetura de uma solução Zabbix, sua utilização proporciona um diferencial, muito desejado no cenário de IoT, a persistência dos dados em caso de queda do link de Internet.

O Zabbix *Proxy* possui um banco de dados local, geralmente um SQLite, para que não ocorram impactos no objetivo de utilização de hardware modesto.

Quando ocorre uma falha no link de Internet, os dados continuam sendo coletados e armazenados no banco de dados local, quando o link for reestabelecido, essas informações são enviadas ao Servidor Zabbix.

Dentre as vantagens, além da persistência dos dados, pode-se citar o fato da necessidade de hardware menor no Servidor Zabbix, uma vez que o processamento será distribuído entre os Zabbix *Proxies*.

### **Utilitários de Linha de Comando**

O Zabbix possui alguns utilitários de linha de comando que auxiliam nos testes, detecção de problemas e validação das coletas.

#### **A. Zabbix Get**

Utilitário utilizado para validar a comunicação do elemento monitorado com o Zabbix Server ou Zabbix *Proxy*, no sentido do servidor para o elemento monitorado.

Pode-se verificar se a comunicação está ocorrendo, o tipo de dado que será retornado e se a resposta obtida, é a esperada. Pode-se obter uma mensagem de erro de execução, por exemplo, detectando assim uma falha de configuração ou de permissionamento em determinado arquivo.

Estes testes são úteis quando se está na fase de planejamento das coletas onde é preciso validar as questões de comunicação e formato dos

dados recebidos. Detalhes sobre seu funcionamento podem ser obtidos na documentação oficial (ZABBIX, 2017c).

### **B. Zabbix Sender**

Utilitário utilizado para validar a comunicação do elemento monitorado com o Zabbix Server ou *Proxy*, no sentido do elemento monitorado para o servidor.

Pode-se enviar um dado esporadicamente, sem a necessidade de configuração de uma coleta recorrente.

Sua utilização é extremamente útil, quando utilizamos um *script* para coletar determinado dado e ao final enviar esse dado ao Zabbix.

Um *script* pode realizar uma série de etapas, tratamentos, tomadas de decisão, o que pode primeiramente levar um tempo maior que 30 segundos, que é o limite máximo de espera de um item. E em última análise decidir se o dado deve ou não ser enviado ao Zabbix.

Detalhes sobre seu funcionamento podem ser obtidos na documentação oficial (ZABBIX, 2017k).

### **Middleware**

A função do *middleware* é ser um software intermediário entre a RSSF e a Internet, como já citado no início da seção, no presente trabalho ele foi desenvolvido em linguagem Python. Ele assume várias funções, detalhadas abaixo, ao longo da sua execução e está armazenado no IoT PM.

### **A. Comunicação**

A comunicação parte sempre do *middleware*, ele monta o pacote do protocolo de comunicação de RSSF Radiuino, via serial envia o pacote a ser transmitido para a base, que por sua vez envia ao sensor ou ao repetidor; quando a base recebe um pacote, envia para a serial, o *script*, que continuamente monitora a serial, detecta e recebe o pacote.

Uma vez recebido o pacote, o *script* pode realizar determinados tratamentos de dados ou tomada de decisões e então enviar os dados para o destino. Nesse ponto podem existir basicamente, três situações: Enviar para

um Zabbix *Proxy* local, enviar para um Zabbix Server local ou enviar para uma plataforma na nuvem, que pode ou não ser um Zabbix Server.

O envio dos dados para a plataforma, se dá por meio da implementação de bibliotecas no *script* Python, no caso de envio de dados para o Zabbix, independentemente de ser o Server ou o *Proxy*, é utilizada a função `zabbix_sender` por meio da biblioteca PY-ZABBIX (2017). Para outras plataformas, são utilizadas as bibliotecas específicas da plataforma.

### **Plataformas de Armazenamento e Visualização de Dados**

O primeiro objetivo de possuir múltiplas plataformas de exibição e visualização de dados é a compatibilidade com diversas soluções, com o objetivo de atender, da melhor maneira possível, as demandas dos negócios. No caso da solução proposta, dependendo do cenário, pode-se utilizar estratégias diferentes.

#### **A. Zabbix Server Local**

Numa situação onde o cliente pode desejar uma cópia de todos os dados coletados, inclusive os *outliers*, para utilizar por exemplo, como fonte para uma ferramenta de inteligência artificial ou *Big Data*, faz sentido possuir um Zabbix Server local.

Em outro cenário onde um negócio necessita de uma grande infraestrutura de RSSF, pode ser necessário, possuir um Operador de RSSF presente no local em tempo integral. Neste caso, o papel do Zabbix Server local é prover apresentação do que está acontecendo, visão de rede, em tempo real.

Tempo real, nesse cenário, é o tempo de coleta e exibição dos dados, por exemplo, se um dado é coletado a cada minuto, o tempo real desse elemento é de um minuto de atualização.

## B. Zabbix Proxy Local

Essa é a situação mais comum, o objetivo final da aplicação é apresentar os dados em uma aplicação na nuvem. Porém para garantir a persistência dos dados em caso de interrupção do link da Internet, é utilizado o Zabbix Proxy.

Ele possui um banco de dados local, temporário, que pode continuar recebendo os dados localmente, até que o link de Internet seja reestabelecido, nesse momento ele envia os dados ao Zabbix Server. Preservando assim o sistema de perda de dados.

## C. Visão de Dados e Visão de Rede

Na seção 3.6 foi feita uma breve introdução sobre o assunto, porém nesse ponto do trabalho, faz-se necessário, um detalhamento do tema. O negócio e o Operador da RSSF trabalham em conjunto, porém possuem objetivos diferentes, e, portanto, necessidades distintas sobre como visualizar o que acontece na solução.

O objetivo do negócio é claro, pois foi determinado pelas regras de negócio e interações com o especialista. Já os objetivos do Operador da RSSF, podem variar dependendo do tipo de tecnologia e solução implementada.

Nesse cenário, é importante ter claro que existem duas visões da solução, conforme apresentado na Figura 16, e, reproduzido novamente na Figura 30.

Figura 30 - Visão de Dados e Visão de Rede



Fonte: Elaboração Própria

A visão de rede proporciona ao Operador da RSSF elementos necessários para a gerência da RSSF, e atuação quando necessário. A ferramenta apresenta nesse cenário, dados referentes à intensidade de sinal,

momentos em que os atuadores são acionados e alertas referentes a esses elementos.

Por exemplo, se um sensor não se comunicar por um período pré-estabelecido, se o sinal em determinado ponto da RSSF se degrada, se um atuador que deveria estar funcionando falhou, em todos esses casos o Operador recebe um alerta. A solução é única, mas as visões são distintas.

#### **D. Escalonamento de Processos**

O foco do presente trabalho são as PME, porém o Modelo de Referência pode ser aplicado em negócios de qualquer dimensão, e a medida que existe um aumento de demanda de recursos da solução, faz-se necessário que a solução seja escalável. Uma das maneiras de atingir esse objetivo é por meio do escalonamento de processos. Apesar de não ser esse o foco do presente trabalho, o assunto foi inserido no modelo de referência, pelas razões já apresentadas.

Quanto ao Zabbix, a ferramenta já possui nativamente a opção de ajustes na configuração para iniciar mais de um processo, responsável por determinada ação, receber pacotes do zabbix\_sender, por exemplo, evitando assim um gargalo nesse ponto da solução.

No caso do *script* em Python, estudos estão sendo realizados, no grupo de pesquisa no qual o mestrando está inserido, para permitir que o mesmo seja utilizado de maneira a gerar vários processos das mesmas tarefas, aproveitando a capacidade dos processadores de vários núcleos, técnica conhecida em computação como *multithreading*.

#### **5.2.5. Nível 4 – Nível de armazenamento**

Nesse nível são tratadas as questões de armazenamento de informações na nuvem. Existem várias plataformas que podem ser utilizadas. Na presente proposta optou-se pela disponibilização de Zabbix Server hospedado na nuvem.

Ao utilizar-se uma plataforma na nuvem, paga-se pelo recurso consumido, dependendo da solução escolhida esse valor é fechado: compra-

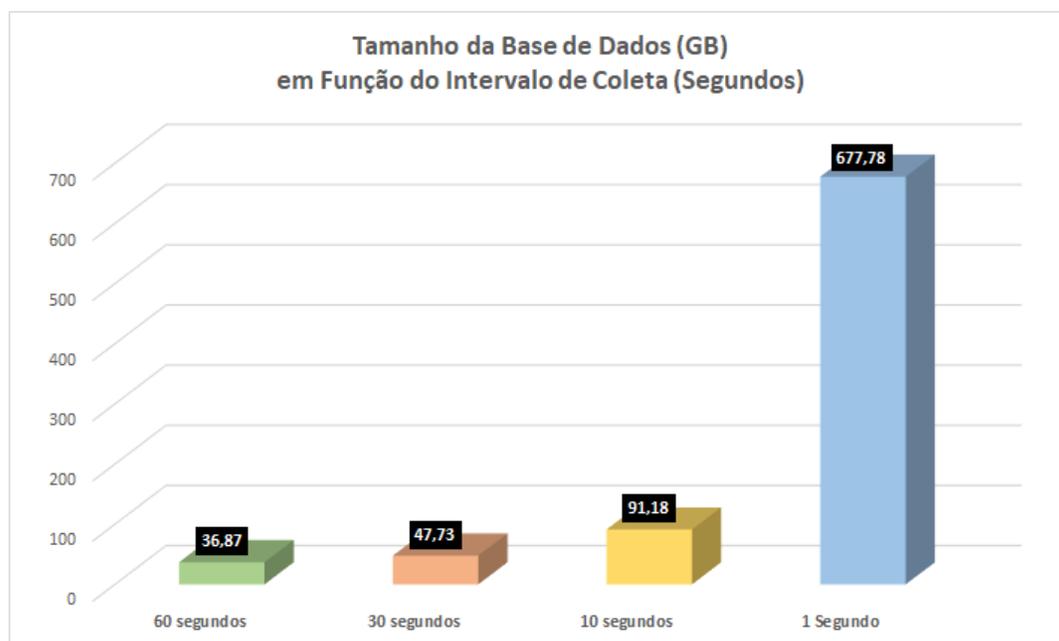
se uma configuração que inclui processamento, memória, espaço em disco e tráfego mensal. Enquanto que em outras soluções paga-se pelo tempo de utilização do recurso, permitindo assim que a aplicação fique desligada quando não estiver em operação. De qualquer maneira existe um custo, e, portanto, existe a preocupação em manter esse custo baixo.

Quando é realizada a entrevista com o cliente, este tende a ter uma visão exacerbada das necessidades, por exemplo solicitando que as coletas de dados aconteçam a cada um segundo. Nesse ponto é importante a participação do especialista, para ajudar a ajustar as reais necessidades do negócio.

O Operador de RSSF, também deve demonstrar as estratégias que podem ser adotadas, por exemplo realizar a coleta no nível do sensor, e enviar uma média ao elemento de borda, que novamente pode realizar tratamentos específicos antes de enviar esses dados para a nuvem.

Para exemplificar essa situação ao cliente, pode-se utilizar diversas estratégias, porém uma maneira simples de demonstrar essa diferença é por meio de tabelas e gráficos. Utilizando-se a fórmula de estimativa de tamanho da base de dados do ZABBIX (2017h). Desenvolveu-se a Figura 31.

*Figura 31 - Tamanho da Base de Dados do Zabbix (GB)*



Fonte: Elaboração Própria

Os dados para desenvolvimento do gráfico são detalhados a seguir:

- Configuração do Zabbix: 10 MB;
- Número de itens coletados: 3000;
- Intervalo de atualização (Nova coleta): Variando de acordo com o apresentado no gráfico, 60 segundos, 30 segundos, 10 segundos e 1 segundo;
- Número Itens Processados por Segundo (VPS, do inglês *Values Per Second*): Calculado pelo número de itens dividido pelo intervalo de coleta;
- Histórico de coletas: 30 dias. Nesse item são armazenadas cada uma das coletas individuais;
- Histórico de médias de coletas: 05 anos. Nesse item são armazenados, a cada hora, um conjunto de valores: máximo, mínimo, média, contagem, para cada item coletado;
- Eventos: 03 anos, no pior cenário possível, que é a ocorrência de um evento por segundo.

Apenas variando o intervalo de coleta de 60 segundos para 1 segundo temos um aumento de 18,39 vezes no tamanho da base de dados, o que pode resultar em uma diferença de custos substancial dependendo da plataforma contratada pelo cliente.

Uma vez que estes dados estejam disponíveis em um banco de dados, pode-se utilizar essas informações em outras plataformas, e aplicar técnicas diversas. Eles podem, por exemplo, servir como fonte de dados para aplicações de *Big Data* e *Analytics*. Podem ainda ser utilizados por plataformas de Inteligência Artificial com o IBM Watson ou KONKER (2018), que buscam encontrar novos padrões e correlações para as informações coletadas.

Finalmente esses dados poderiam ainda servir de referência para simuladores científicos, por exemplo, utilizar a plataforma proposta por RUIZ (2003) para analisar e correlacionar os resultados que a autora obteve na época com os dados coletados no presente, em uma arquitetura totalmente diferente da original.

### **5.2.6. Nível 5 – Nível de abstração**

Nesse nível busca-se entender aspectos do negócio que não estão explicitamente declarados, como no caso das regras de negócio. Pode ser uma percepção que o especialista tenha, mas que não tenha sido validada, ele pode por exemplo, solicitar uma alteração da configuração padrão por um período, para depois avaliar os dados e tomar uma decisão.

Aqui se enquadrariam também as correlações que a inteligência artificial detectou, o especialista não precisa ser necessariamente uma pessoa, pode ser também uma entidade de IA, em última análise um software.

Na situação do Caso da Fila do Restaurante Universitário, por exemplo, poderia se observar se o comportamento das pessoas seria alterado. As pessoas que tem por preferência um dos locais, estariam dispostas a frequentar outro local em função dos tamanhos das filas?

Outro nível de abstração aplicado foi o uso do Radiuino emulado diretamente no código do Arduino. Isso possibilitou o desenvolvimento e validação de regras de negócio, alertas, mapas, geração de gráficos, sem a necessidade de montar uma RSSF física.

### **5.2.7. Nível 6 – Nível de apresentação**

Durante a elaboração dos casos, e, também em conversas com especialistas de outras áreas para entender suas necessidades e validar o modelo de referência, foi diagnosticado que, na amostra que experimentamos, as pessoas possuem ideias diferentes de interfaces.

Em conversas com a Profa. Dra. Cláudia Cotrim Pezzuto, docente permanente do mestrado em Sistemas de Infraestrutura Urbana (PUC, 2017), as necessidades eram de visualização de plantas baixas, relação da temperatura com a umidade, relação de temperaturas de um mesmo local, mas em alturas distintas.

Essas demandas chamaram a atenção por serem totalmente diferentes do que tínhamos realizado em termos de apresentação até então. Após essas conversas, foi iniciado no grupo as discussões sobre apresentação dos dados,

e como as regras de negócio e opinião do especialista influenciavam nesse ponto.

### **5.2.8. Gerência**

As questões de Gerência clássica, mais especificamente o modelo FCAPS, estão presentes desde o princípio da elaboração do modelo:

- Gerência de Falhas: Preocupação com o funcionamento da rede, detecção de deterioração do sinal, detecção de falhas de comunicação, perda de pacotes, falta de comunicação dos nós sensores com a base ou repetidor;

- Gerência de Configuração: Ajustes de parâmetros do protocolo, ajustes de configuração da transmissão e recepção, configuração dos arquivos do Zabbix;

- Gerência de Contabilização: Levantamento das necessidades do cliente para precificação da solução, papel desenvolvido pelo Operador de RSSF;

- Gerência de Desempenho: Validação dos níveis de sinais da RSSF, leitura dos dados de acordo com as necessidades do negócio, ajustes no banco de dados do Servidor Zabbix;

- Gerência de Segurança: Presente desde o início por meio do desenvolvimento da dissertação de FREITAS (2018).

Porém durante o processo de revisão bibliográfica, principalmente nas recomendações do ITU-T, fica claro que outros aspectos de gerência podem ser abordados.

### **5.2.9. Segurança**

As questões de segurança não foram abordadas no presente trabalho. Apesar de estarem presentes desde o princípio da elaboração do modelo de referência, por meio do trabalho de outro membro do grupo, FREITAS (2018), e que resultou em sua dissertação.

### **5.3. Aplicativos de Celular**

Durante os testes realizados nos casos, foram detectadas algumas situações, onde um aplicativo de celular seria útil para validar necessidades específicas e pontuais.

Foram visualizadas também novas oportunidades de experimentação, como por exemplo, no caso da fazenda cafeeira, seria possível ter usado um celular ao invés do módulo de GPS instalado no trator?

Diante desse cenário foram desenvolvidos dois aplicativos.

#### **5.3.1. Zabbix Sender para celular**

Durante o processo de testes do envio de dados para a ferramenta de gerência, foram encontradas várias situações onde os dados não alcançavam o destino.

Para saber se o problema estava na comunicação do nó sensor até o IoT PM, ou no envio dos dados para a ferramenta de gerência, utilizou-se o Zabbix Sender Mobile (ZABBIXSENDER, 2017b), para enviar dados via rede de telefonia celular ou Wi-Fi.

Se a comunicação ocorria de forma transparente os esforços eram concentrados no ambiente de RSSF, caso contrário os esforços se davam na ferramenta de gerência.

#### **5.3.2. App de Geolocalização**

Com fruto da prova de conceito, realizada na fazenda cafeeira, foi criado o App de Geolocalização (ZABBIXGEO, 2018b), que lê os dados de GPS do celular e atualiza as informações na ferramenta de gerência.

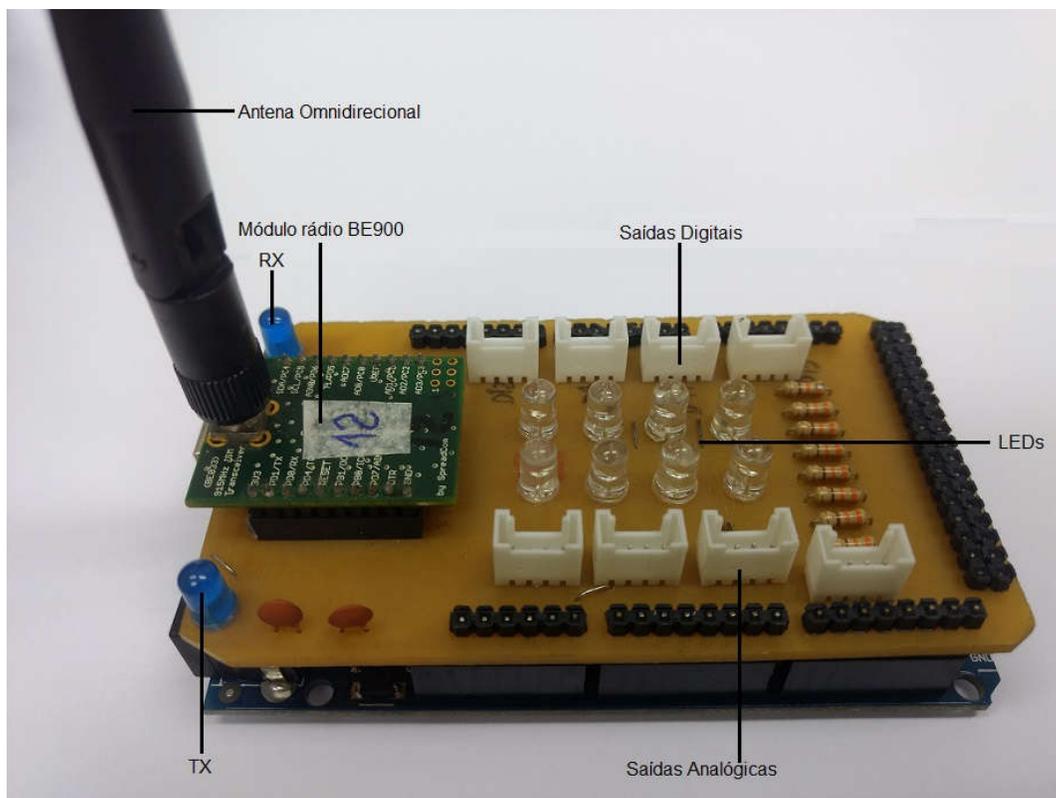
Validando assim a hipótese de que o GPS do celular é suficiente para uma solução de geolocalização, aliada à plataforma de IoT, atingindo o objetivo com um baixo custo, uma vez que geralmente as pessoas que dirigem um veículo, já possuem um celular.

Esse app unido ao Lokito (LOCKITO, 2018), que possibilita enganar o GPS do celular, com rotas criadas pelo aplicativo, permite demonstrar o conceito sem necessidade de deslocamento.

#### 5.4. Caso Atuator

Com o objetivo de demonstrar o uso da solução como um atuador, o Engenheiro Eletricista e mestrando em Engenharia Elétrica na Faculdade de Engenharia Elétrica e de Computação (FEEC) na Unicamp, Guilherme Lopes da Silva, desenvolveu um *shield* para Arduino Mega (ARDUINO, 2017), os detalhes podem ser observados na Figura 32. Silva, faz parte do mesmo grupo do qual o mestrando está inserido.

Figura 32 - Shield Arduino Mega



Fonte: Elaboração Própria

No elemento de borda foi desenvolvido um *script* em linguagem Python para interagir com o projeto, nomeado de “meduZa”. O projeto é composto de

uma série de saídas analógicas e digitais que podem ser utilizadas para conexão de sensores diversos.

Além disso, possui uma série de LEDs que podem ser acionados para simular, por exemplo, lâmpadas de uma residência, conforme será abordado na seção 6.6.4.

O *script* possui dois modos de operação: interativo e comando. Se for executado passando o parâmetro 1, ele entra no modo interativo, Figura 33A; se for executado no modo comando, é preciso passar a posição do objeto com o qual deseja atuar e a operação, “1” para ligar e “0” para desligar, Figura 33B.

Figura 33 – Modos de execução do script

```

root@localhost:~# ./meduZa.py 1
ID_base: 0
ID_sensor: 1

Escolha um comandos abaixo e pressione enter:
1 - Realiza Medidas:
s - Para sair:
Entre com a Opção = s
Fim da Execução

```

A

```

root@localhost:~# ./meduZa.py 43 1
Comando Executado com sucesso!

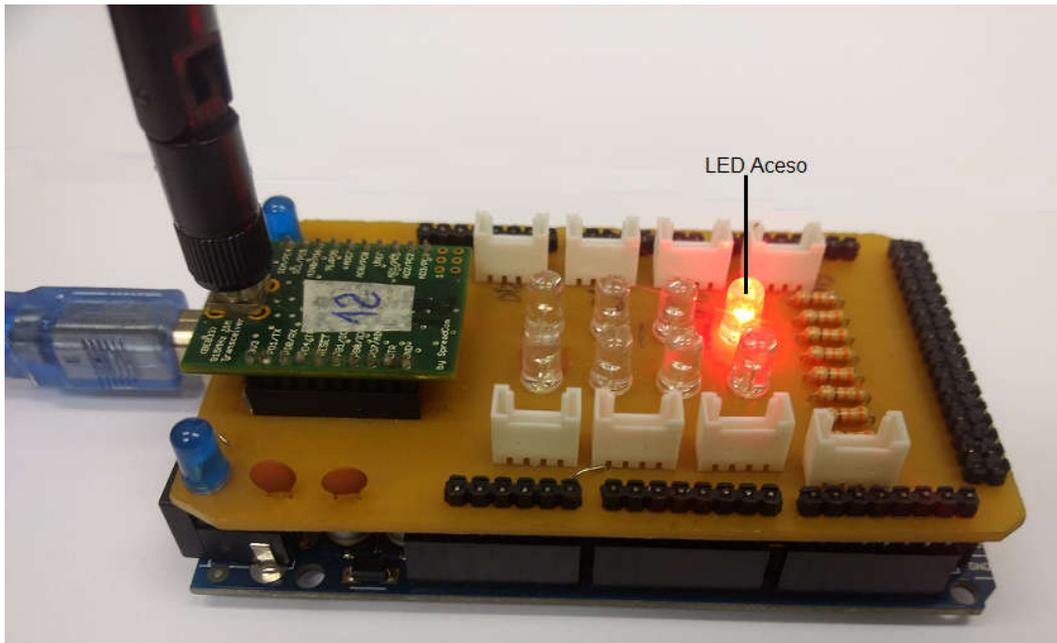
```

B

Fonte: Elaboração Própria

Ao executar o *script* com passagem de parâmetro, um pacote Radiuino será criado, enviado via RSSF até o *shield*, onde será recebido pelo módulo rádio e repassado ao Arduino Mega, que foi previamente configurado com um *firmware* para interpretar a instrução e executar a ação no LED correspondente. O resultado da interação é refletido na meduZa, Figura 34.

Figura 34 - meduZa com o LED aceso



Fonte: Elaboração Própria

Apesar do *shield* possuir saídas analógicas e digitais, o objetivo neste ponto do trabalho não é abordar esse tema, pois o funcionamento é idêntico aos apresentados nos casos anteriores.

### 5.5. Caso Implementação Prática do Modelo de Referência

Os casos e implementações citados até agora serviram para aperfeiçoar pontos específicos do modelo de referência e ajudar no amadurecimento do mesmo, porém nenhum deles contemplou o modelo de referência por completo.

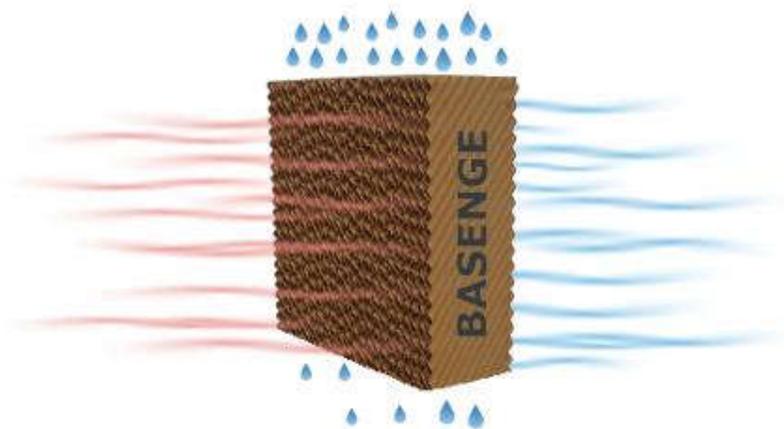
Com esse objetivo em mente, foi criado o protótipo que será detalhado nas próximas Seções. E para elucidar o fluxo da solução de maneira mais clara, o modelo de referência será apresentado em uma abordagem *top-down*.

### 5.5.1. Nível 7 – Negócio

Em diversas áreas existe a necessidade de resfriamento do ambiente, como por exemplo, estufas em agricultura. Para construção de um exemplo será criada uma aplicação para resfriamento de um ambiente formado por um duto e sua comparação com o ambiente externo.

No caso do experimento o princípio de resfriamento evaporativo é obtido fazendo circular ar por uma esponja umedecida.

*Figura 35 - Resfriamento Evaporativo*



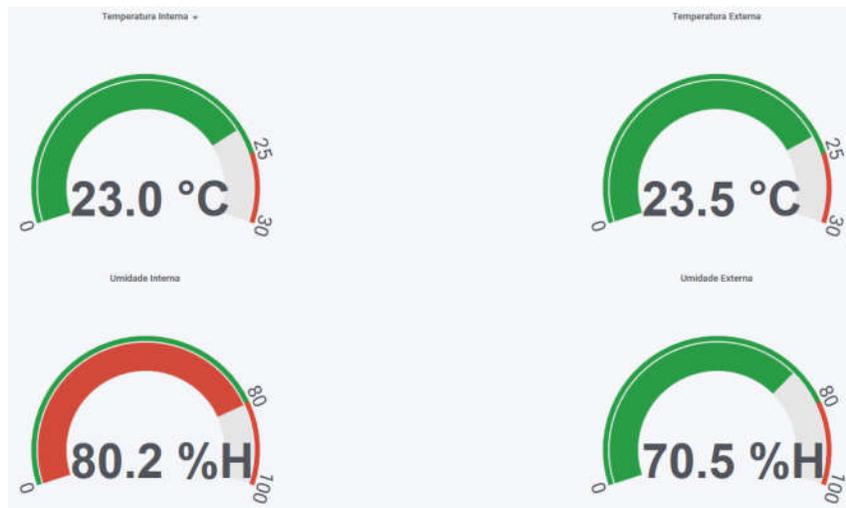
FONTE: BASENGE, 2018

### 5.5.2. Nível 6 – Nível de apresentação

Foram desenvolvidas três interfaces de visualização dos dados, uma para o Operador da RSSF gerada no Zabbix e duas para o cliente; sendo uma delas destinada ao negócio produzida no Grafana e outra destinada ao especialista, desenvolvida em Python.

A interface do negócio é utilizada no dia-a-dia da operação, Figura 36, seu formato de velocímetro permite que um operário que não entende do negócio consiga acompanhar o sistema e atuar, se necessário.

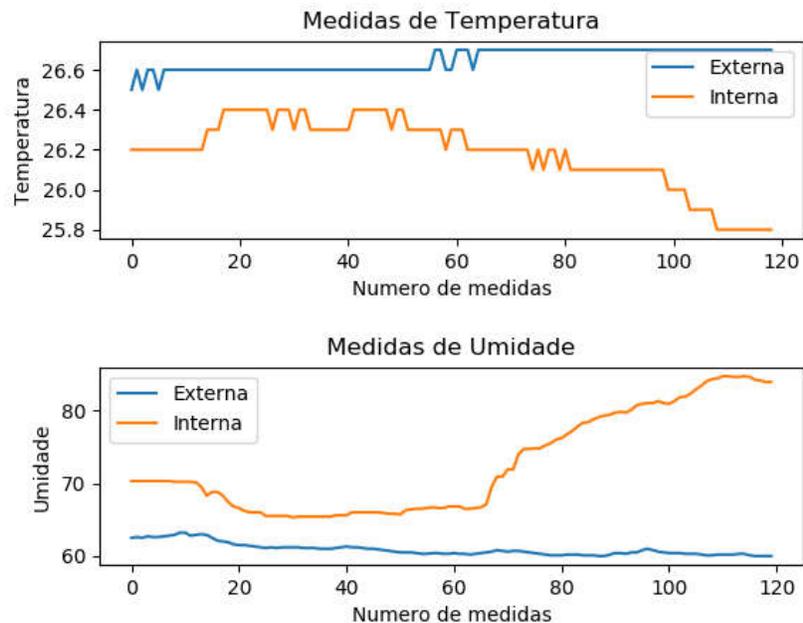
Figura 36 - Interface do Negócio



Fonte: Elaboração Própria

A interface do especialista, Figura 37 , permite que ele acompanhe em tempo real as coletas e que possa atuar no ambiente. Essa abordagem é extremamente útil nas fases iniciais de implementação, após isso essa atuação passa a ser mais refinada, como abordado na próxima seção.

Figura 37 - Interface do Especialista

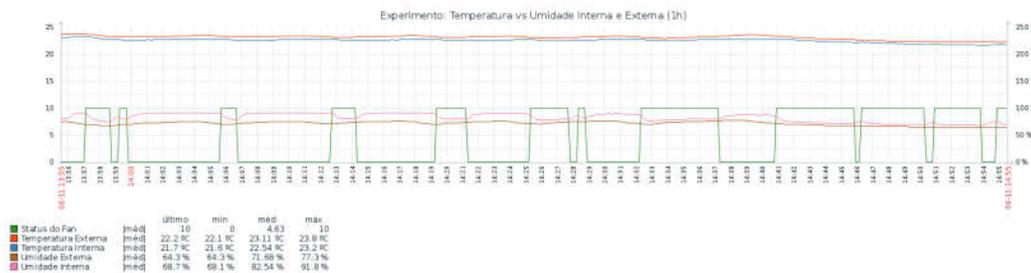


Fonte: Elaboração Própria

No exemplo da Figura 37 é observada a diminuição da temperatura em cerca de  $0,6^{\circ}\text{C}$  com o aumento da umidade de 68% a 85%, demonstrando o efeito do resfriamento evaporativo.

A interface do Operador de RSSF, Figura 38, não necessita de grandes refinamentos. Apesar de ser importante, principalmente em situações onde o cliente possa solicitar alguma informação histórica, por exemplo.

Figura 38 - Interface do Operador de RSSF



Fonte: Elaboração Própria

Apesar de simples a interface possui as informações relevantes ao negócio. Um detalhe a ser destacado é a temperatura no eixo esquerdo em escala de  $5^{\circ}\text{C}$  por faixa e a umidade no eixo direito em escala de 20% por faixa.

### 5.5.3. Nível 5 – Nível de abstração

O fenômeno que permite o resfriamento parte do princípio que a água ao evaporar retira calor do ambiente (BASENGE, 2018). Este é o mesmo fenômeno que permite com que a água em uma moringa fique mais fresca, uma vez que o barro da moringa permite a evaporação da água roubando calor da moringa. Esse é o tipo de informação que é obtido por meio do especialista.

Com o intuito de demonstrar como o conhecimento do especialista é refletido na solução, foi criada uma tela para simular essa atuação, Figura 39. Ela permite alterar o Delta Temperatura, diferença entre a temperatura interna e externa, o Tempo de Decisão, de quanto em quanto tempo será checada essa diferença e o intervalo de coleta dos dados dos sensores.

Figura 39 - Abstração do Conhecimento do Especialista



Fonte: Elaboração Própria

Em uma solução real, os dados coletados seriam enviados para uma plataforma de IA, como já discutido anteriormente, e as correlações descobertas, seriam enviadas ao especialista, que por sua vez, solicitaria alterações no ambiente.

Porém essa tela permite demonstrar essa atuação do especialista e acompanhar os efeitos no ambiente em tempo real. Com destaque para o fato de que a tela conversa com o IoT-PM, que tem a capacidade de alterar o ambiente de acordo com a necessidade do negócio de forma imediata. O fluxo técnico é descrito a seguir.

O Python envia para o nó sensor sem fio as informações do Delta de Temperatura, Tempo de Decisão e Intervalo entre Requisições. O firmware que está no nó sensor sem fio realiza o processo de controle de forma automática, utilizando inicialmente um valor *default* e depois recebe do computador os parâmetros para fazer a regulação da diferença de temperatura.

#### 5.5.4. Nível 4 – Nível de armazenamento

O armazenamento dos dados é realizado em um Zabbix Server na Nuvem. A base de dados funciona como fonte para o Grafana gerar os gráficos.

### 5.5.5. Nível 3 – Nível de Borda

É utilizado um Raspberry Pi 3 como computador local; os *scripts* em linguagem Python realizam a função de *Middleware*. Eles contêm a função de recebimento dos dados dos níveis inferiores e envio para o Zabbix *Proxy*, por meio da implementação do `zabbix_sender` via biblioteca.

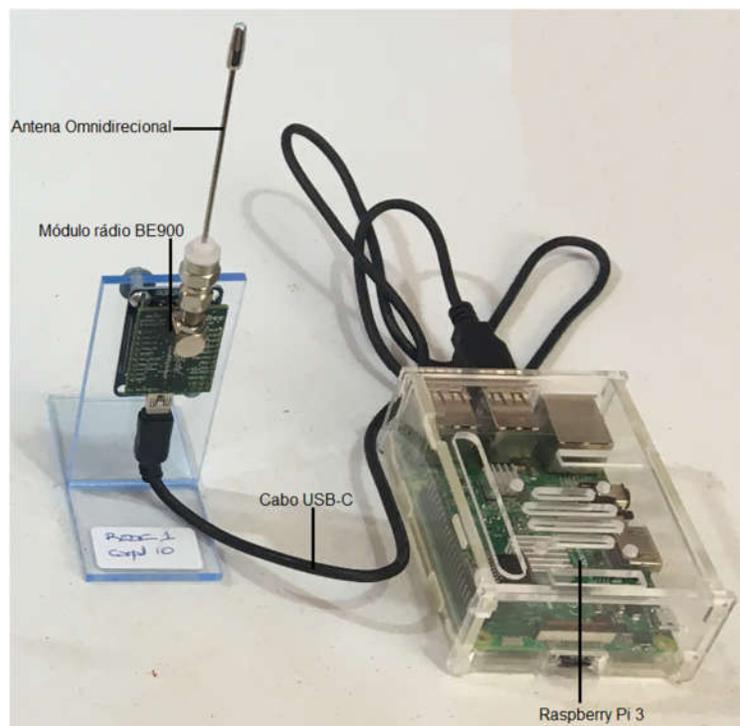
No Zabbix *Proxy*, que está instalado no próprio Raspberry Pi 3, existe o armazenamento local em banco de dados SQLite, o que garante a persistência dos dados mesmo em caso de interrupção do link com a Internet.

A interface demonstrada na Figura 39, responsável pela interação do especialista, também pode estar presente nesse elemento, alterando os parâmetros de atuação e coleta das camadas inferiores.

### 5.5.6. Nível 2 – Nível de Conectividade

É utilizada uma base de rádio conectada diretamente ao Raspberry Pi 3 via cabo USB-Serial, Figura 40, a transmissão ocorre via protocolo de RSSF Radiuino.

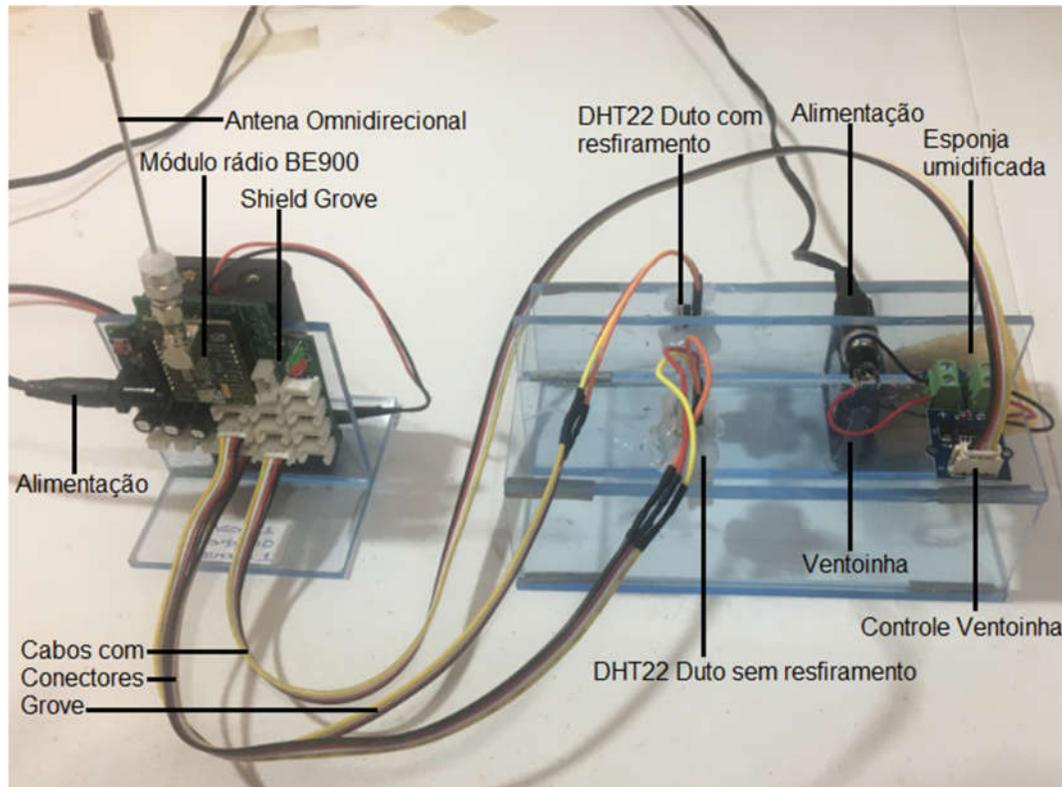
Figura 40 - Raspberry Pi 3 com a Base do BE900



Fonte: Elaboração Própria

No experimento, Figura 41 , é conectado um módulo de rádio, que é o responsável por receber as requisições do elemento de borda, enviar o comando para coletar os dados e/ou atuar no dispositivo, nesse caso ligar ou desligar a ventoinha, e por fim, enviar os dados coletados ao elemento de borda.

Figura 41 - Experimento Resfriamento por Evaporação - Figura 01



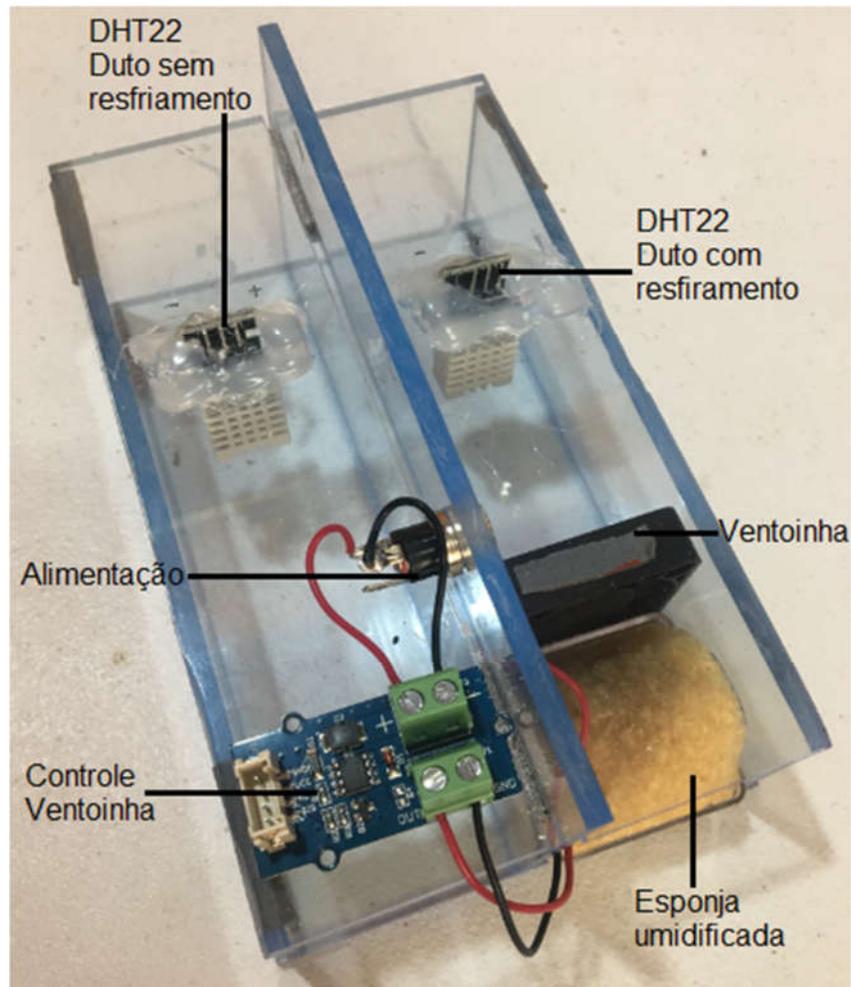
Fonte: Elaboração Própria

### 5.5.7. Nível 1 – Nível de Dispositivo, Condicionamento e Controle

O experimento conta com dois dutos, um com resfriamento e outro sem resfriamento. O objetivo é mostrar que existe uma diferença entre as duas temperaturas, que pode ser controlada pelo fluxo de ar da ventoinha.

O objetivo do experimento não é criar um sistema eficiente de resfriamento, mas sim utilizar o princípio para criar uma aplicação IoT. Essa solução foi elaborada com base na necessidade do negócio, discutido na seção 0, e os detalhes do experimento podem ser visualizados na Figura 42.

Figura 42 - Experimento Resfriamento por Evaporação – Figura 02



Fonte: Elaboração Própria

### 5.5.8. Nível 0 – Nível das Coisas

Nesse experimento as coisas são temperatura e umidade, que estão diretamente ligadas as necessidades estabelecidas no nível 7.

## 6. RESULTADOS

Nesta seção são apresentados os resultados das implementações dos temas discutidos nas seções anteriores.

### 6.1. Nível 1 – Nível de dispositivo, condicionamento e controle

Durante a elaboração do presente trabalho e nos casos apresentados uma série de sensores foram utilizados, dentre eles VH400, Módulo GPS, DHT22 e um atuador, a ventoinha.

Demonstrou-se que foi possível coletar os dados desses sensores e interagir com a ventoinha de maneira automática, utilizando apenas ferramentas *Open Source*.

Entretanto, é importante destacar que durante a realização dos casos foram detectados uma série de problemas. Essas situações serviram para aperfeiçoar o modelo, pois conforme os problemas ocorriam, novas necessidades eram detectadas.

A medida que for pertinente esses problemas serão abordados nos níveis respectivos. Por exemplo, durante o Caso da Fazenda Cafeeira, foi registrada uma leitura do trator a uma velocidade de 300 km/h.

Claro que esse dado era um *outlier*, dessa situação surgiu a compreensão de utilizar o poder de processamento do microcontrolador para tratar esses dados, gerando o condicionamento no nível 1.

### 6.2. Nível 2 – Nível de conectividade

O Protocolo de RSSF Radiuino se mostrou eficiente para realizar a coleta e transmissão dos dados. Porém, sendo uma tecnologia de rádio, enfrentou problemas como áreas sem cobertura e barreiras que impediam a propagação do sinal.

As soluções abordadas nos Casos da Fazenda Cafeeira e do Protocolo repetidor, foram essenciais para desenvolver mecanismos para lidar com essas situações.

Como por exemplo, realizar a validação da área, conhecida como *Site Survey*, a utilização ou não de repetidores, além claro da possibilidade de alteração de parâmetros do protocolo.

### 6.3. Nível 3 – Nível de borda

A estratégia de possuir um armazenamento temporário no elemento de borda se mostrou essencial ao longo dos testes, no Caso Campo de Futebol – PUC Campinas, as condições eram tidas como ideais, Figura 43.

*Figura 43 - Disposição dos elementos no Campo de Futebol - PUC Campinas*



Fonte: MATTHIESEN, 2018

Os dois nós sensores e o nó repetidor, possuíam baterias e painéis fotovoltaicos e nenhuma situação de sombra; a estação rádio base e o IoT PM



Em todas as situações onde a comunicação era realizada diretamente com o Servidor na nuvem, e, onde não ocorreram problemas de comunicação, os dados foram recebidos e armazenados.

Esta ação permitiu assim a consulta dos mesmos em qualquer período posterior à coleta. Respeitando claro os parâmetros de armazenamento que foram configurados na ferramenta.

No Caso Campo de Futebol – PUC Campinas, a perda dos dados poderia ter sido evitada, se a estratégia do *Proxy* tivesse sido utilizada.

### **6.5. Nível 5 – Nível de abstração**

Durante o desenvolvimento do trabalho, ficou claro que o grupo no qual o mestrando está inserido, tinha domínio sobre o “como”, os membros do grupo e o professor orientador eram plenamente capazes de coletar e transmitir dados via RSSF.

Porém, faltava o propósito, o “para que”, a conversa com outros especialistas e a busca dos casos surgiu dessa necessidade; para ser uma solução, é preciso ter um objetivo, resolver algum problema.

Logo que as conversas se iniciaram, ficou claro, que o papel do especialista era essencial, como citado anteriormente, detectou-se que as áreas possuíam visões diferentes do conceito de interface, termos técnicos e objetivos específicos.

### **6.6. Nível 6 – Nível de apresentação**

Apesar da intenção da Zabbix SIA de tornar o Zabbix um software mais próximo de IoT, e da grande flexibilidade da ferramenta, alguns ajustes são necessários para proporcionar ao usuário uma melhor experiência. Nessa seção serão apresentadas as personalizações para que a ferramenta se adeque aos usuários: na maneira que apresenta os dados, nos textos exibidos nos menus e nos recursos que disponibiliza.

Na arquitetura proposta, esses são exemplos de demandas que serão solicitadas pelo negócio e pelo especialista.

### 6.6.1. Alertas Personalizados

O Zabbix possui nativamente o recurso de delimitação de limites (*thresholds*), gatilhos de eventos, do inglês *triggers* (ZABBIX, 2018a) e ações de *triggers* (ZABBIX, 2018b), esse conjunto de facilidades permite a definição de uma grande variedade de alertas e notificações.

Porém é preciso criar estratégias de notificação que atendam às necessidades do ambiente, gerência das redes, dos clientes e gerência dos dados. Nesse quesito, novamente existe uma variedade de necessidades muito grande, influenciadas pelas necessidades e particularidades de cada tipo de negócio.

Um dos recursos mais interessantes, nos mapas, é a possibilidade de representar *triggers*, com isso, é possível representar até cinco níveis de não conformidade, de acordo com a gravidade do problema.

REAL (2015) desenvolveu uma tabela para expressar a relação da intensidade do sinal com a qualidade do mesmo, como pode ser observado nas duas primeiras colunas da Tabela 3.

Apesar de ser uma tabela com base nos resultados que ela encontrou no ambiente onde testes foram realizados, utilizando o recurso de dependências de *triggers* do Zabbix, é possível expressar essa dependência, que seria um exemplo de informação fornecida pelo especialista.

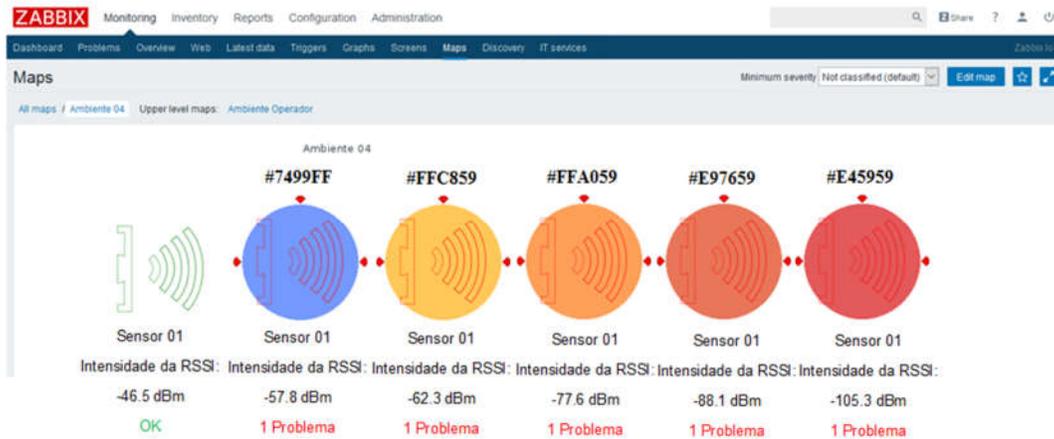
*Tabela 3 - Classificação da RSSI*

Classificação	RSSI	Dependência	Cor (HTML)
Excelente	> -50 dBm		
Ótimo	entre -50 e -59,99 dBm	↓	#7499FF
Bom	entre -60 e -69,99 dBm	↓	#FFC859
Regular	entre -70 e -79,99 dBm	↓	#FFA059
Ruim	entre -80 e -89,99 dBm	↓	#E97659
Péssimo	<= -90 dBm	↓	#E45959

Fonte: Adaptado de REAL, 2015

Na Figura 45, observa-se que além dos níveis de não conformidade, existe um ícone para situações de conformidade, verde, e outro para situações de não conformidade, vermelho.

Figura 45 - Mapa com Triggers



Fonte: Elaboração Própria

Cada círculo possui uma cor diferente, que representa a gravidade da não conformidade. Na imagem foi destacado o código da cor em notação HTML (*HyperText Markup Language*).

Como citado anteriormente, devido ao recurso de dependências de triggers, apenas uma imagem seria exibida, a imagem com a cor que representa a situação de intensidade de sinal atual.

### 6.6.2. Ícones Personalizados

Cada área de atuação possui um conceito de identidade visual, além disso, o próprio design evolui ao longo dos anos, por isso é importante desenvolver ícones com os quais os usuários possuam identificação.

Na Figura 46, são apresentados exemplos de ícones que foram criados para representar coletas em locais específicos, como por exemplo, sensores em um lago, em uma mata, no transporte público, para aplicações de segurança e etc.

Além disso, existem três cores distintas para cada ícone: cinza que representa um sensor cadastrado, mas que não está sendo monitorado, verde

que representa um sensor operando em estado de conformidade e vermelho que representa um sensor operando com alguma não conformidade.

Figura 46 - Ícones Personalizados



Fonte: Elaboração Própria

Os ícones cadastrados, podem ser utilizados em mapas, o que facilita a visualização de problemas pelo Operador de RSSF. Podendo ser disponibilizado à um operador, que simplesmente acompanha as mudanças e atua de acordo com a necessidade.

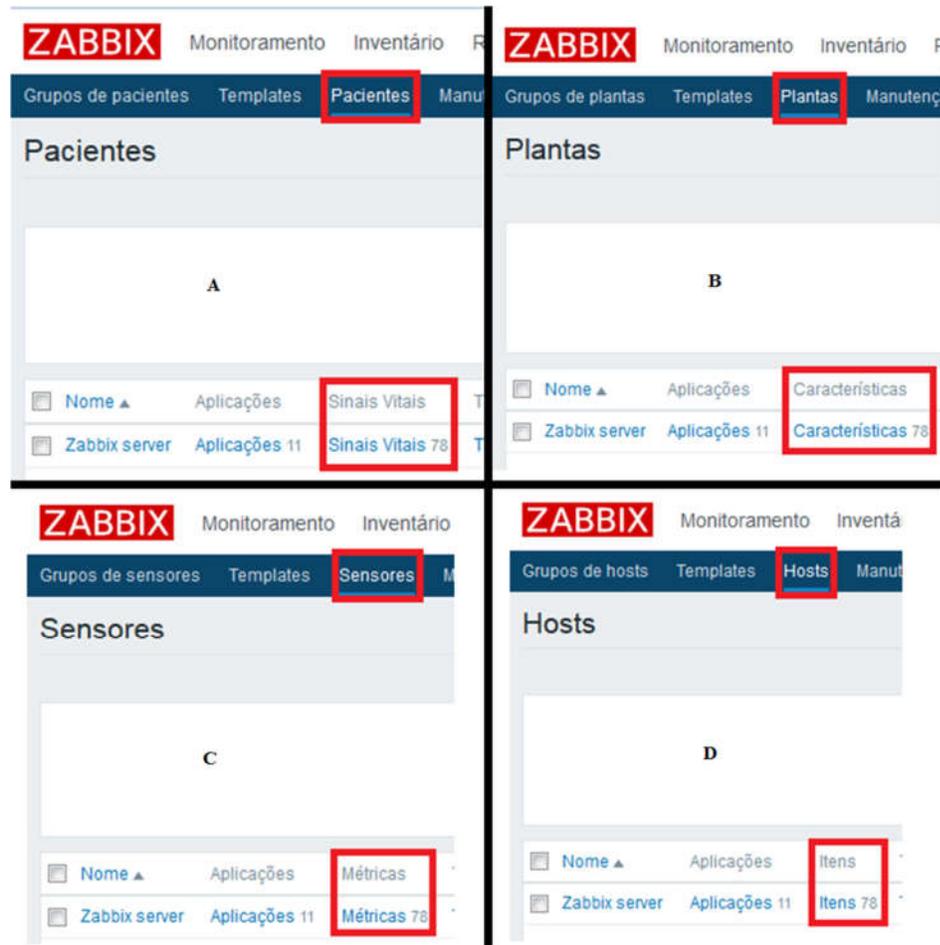
O Zabbix possui a opção de importar imagens de ícones, e, fundo para serem utilizadas nos mapas. Para agilizar o processo de importação em lote, pode-se utilizar um *script* (GITHUB, 2011).

### 6.6.3. Menus Personalizados

A interface padrão do Zabbix possui opções de cadastrar “Hosts”, e cada host possui “itens”. Isso faz sentido para redes de computadores, mas para um Engenheiro Agrônomo provavelmente não, tampouco para um Médico. Com o pensamento de que a solução deve-se adaptar ao usuário, é preciso personalizar os menus (ANDREDEO, 2017).

Pode-se observar na Figura 47, a personalização dos menus para atender áreas distintas de atuação, proporcionando ao usuário da ferramenta maior produtividade, visto que o usuário possui familiaridade com os termos envolvidos.

Figura 47 - Menus Personalizados



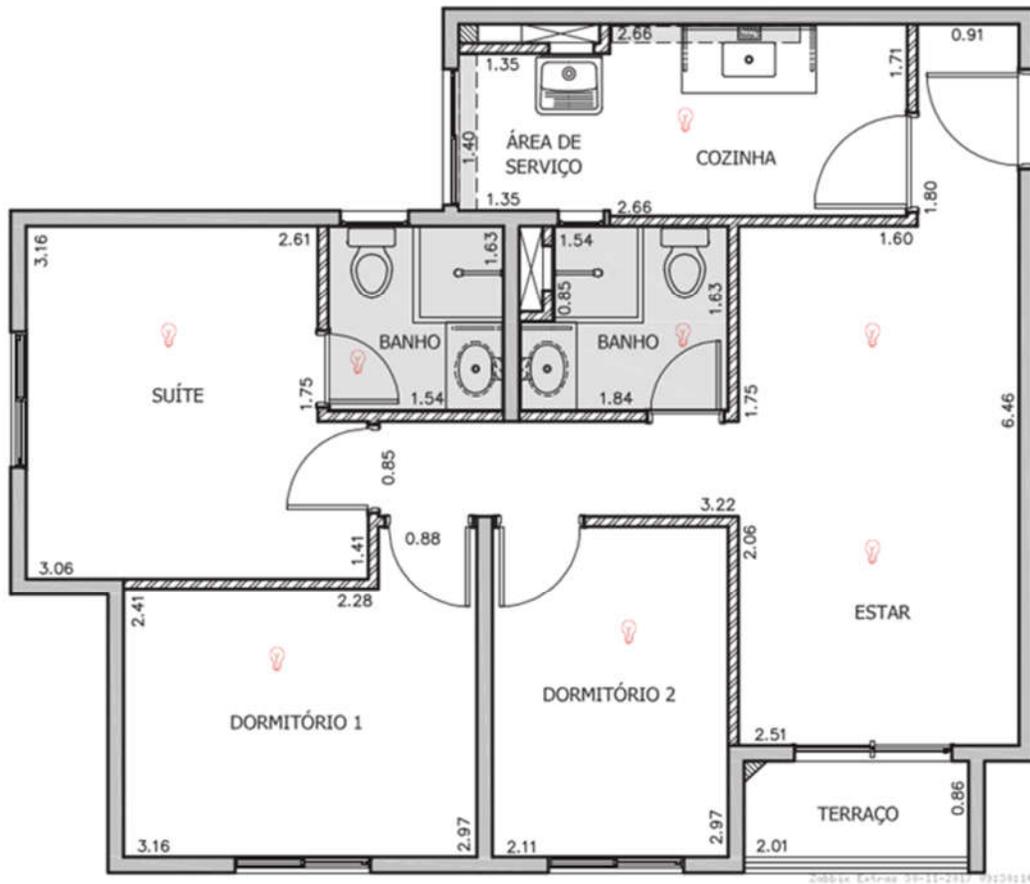
Fonte: Elaboração Própria

Para atender a necessidade de atuação em áreas diversas, esse tipo de personalização é essencial e indispensável. Quanto maior o conforto do usuário com a ferramenta, menor será a resistência na adoção e utilização da mesma.

#### 6.6.4. Integração com Atuadores

O projeto meduZa, citado na seção 5.4, foi integrado na interface do Zabbix, por meio do recurso de mapas, como pode ser observado na Figura 48, foi utilizada uma planta baixa representando um apartamento com oito lâmpadas. Cada lâmpada representada no mapa, corresponde à um LED no *shield*.

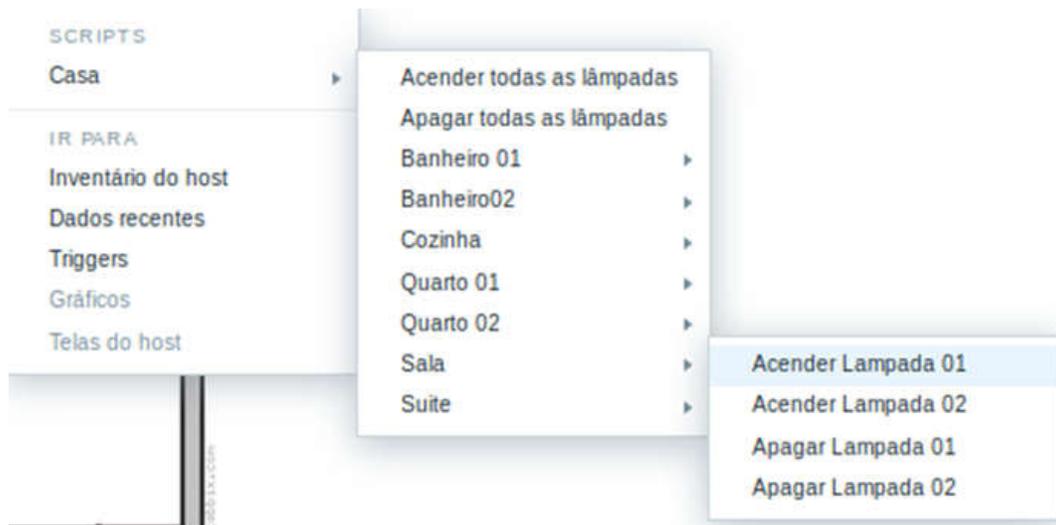
Figura 48 - Mapa Planta Baixa



Fonte: Elaboração Própria

Nos mapas do Zabbix é possível criar menus de contexto, utilizando esse recurso, foi criado um menu “Casa”, com as opções de “Acender todas as lâmpadas”, “Apagar todas as lâmpadas” ou navegar por cada um dos cômodos; Ao navegar por um dos cômodos, “sala”, por exemplo, é possível acender ou apagar as lâmpadas individualmente, Figura 49.

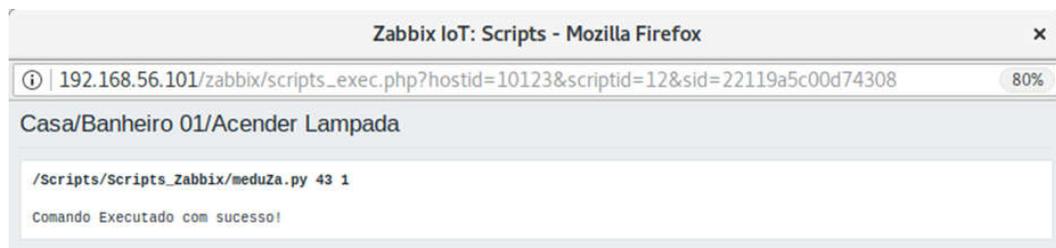
Figura 49 - Menu de contexto



Fonte: Elaboração Própria

Ao acessar qualquer opção do menu, o Zabbix executa o *script* em linguagem Python, passando os parâmetros necessários, Figura 50.

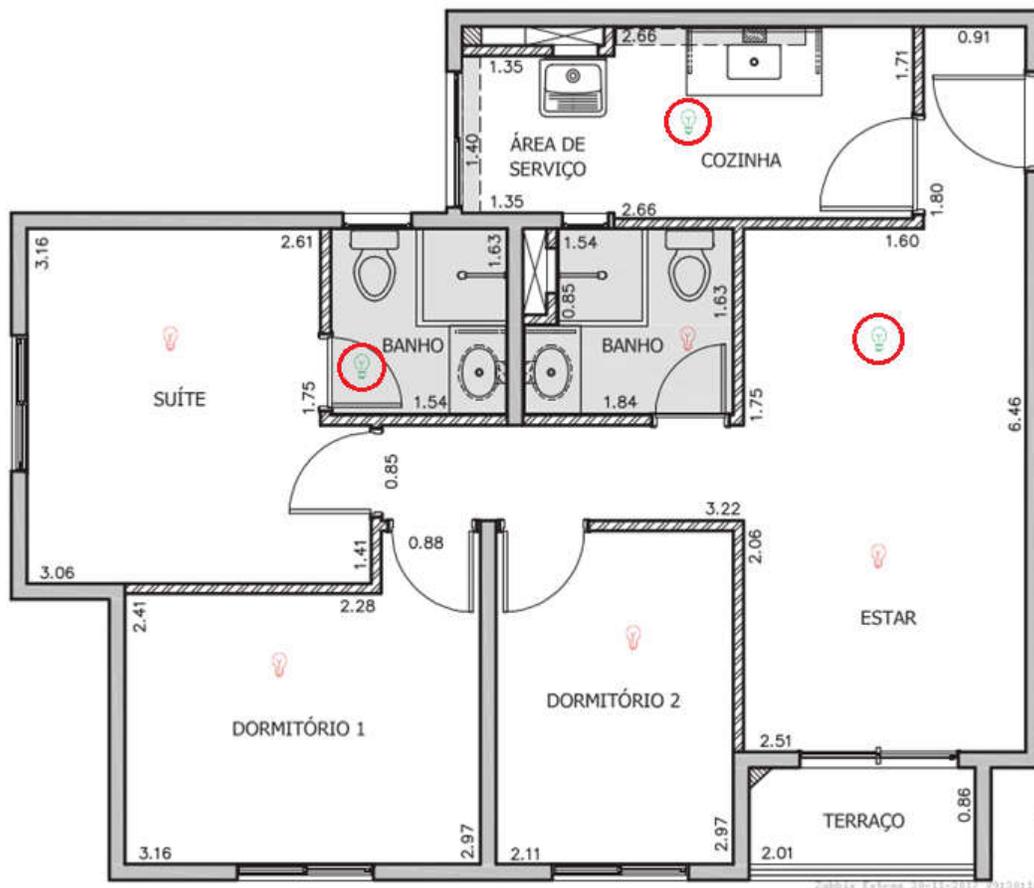
Figura 50 - Script executado pelo Zabbix



Fonte: Elaboração Própria

O resultado será refletido na meduZa, como já demonstrado na seção 5.4, e no mapa do Zabbix, como pode-se observar na Figura 51. Essa implementação traz segurança ao usuário do sistema, pois ele possui na interface uma representação real do estado do elemento físico, a “coisa”.

Figura 51 - Mapa refletindo a interação com a meduZa



Fonte: Elaboração Própria

### 6.6.5. Integração de Plugins

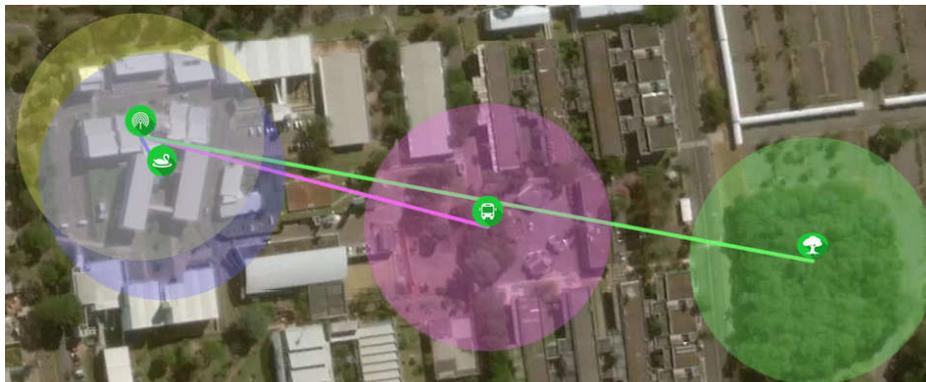
Apesar do Zabbix ser uma ferramenta extremamente flexível, algumas funções desejadas pelos usuários (ZABBIX, 2017g) ainda não estão presentes na ferramenta. Para contornar esse problema, foram desenvolvidos plug-ins, sendo o EveryZ (EVERYZ, 2017a) um dos mais utilizados.

Dentre as várias funcionalidades, destaca-se o módulo de Geolocalização (EVERYZ, 2017b) que permite trazer para a ferramenta mapas das soluções OpenStreetMap (OPENSTREETMAP, 2017) e mapbox (MAPBOX, 2017).

O recurso de geolocalização é demonstrado na Figura 52. Para cada sensor é possível utilizar um ícone, exibir um círculo, que representa, por

exemplo, a intensidade de sinal de RSSF e ainda criar representação de links com cores personalizadas.

*Figura 52 - Recurso de Geolocalização*



Fonte: Elaboração Própria

Além disso, é possível, ler o posicionamento de um sensor de GPS e atualizar em tempo real, a cada nova leitura, o posicionamento do ícone no mapa, equivalente ao sensor.

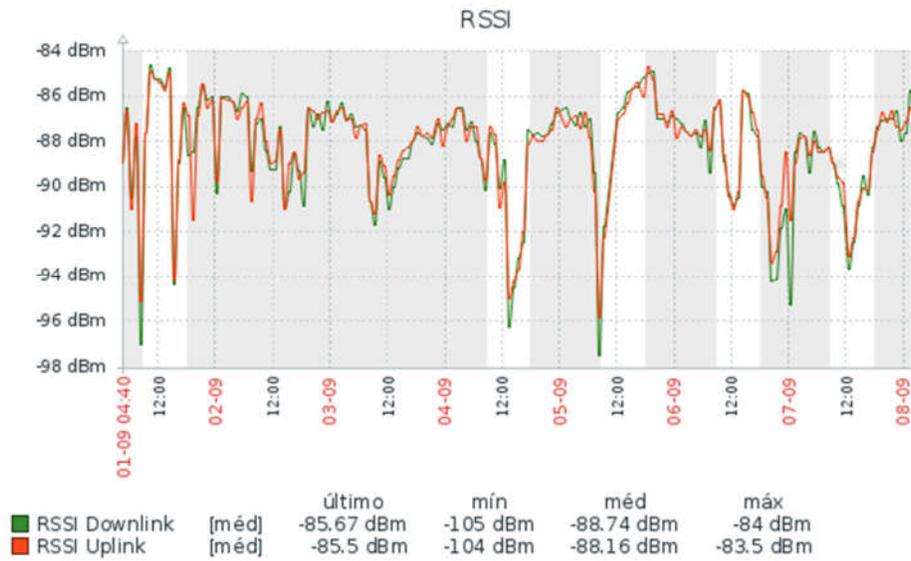
Essa leitura de posicionamento, pode também ser realizada, por meio do aplicativo de celular (ZABBIXGEO, 2018b).

#### **6.6.6. Demais Personalizações**

As funções nativas da ferramenta, também trazem recursos que podem ser utilizados de acordo com a necessidade do cliente, como por exemplo, realizar personalizações na maneira que os dados são apresentados.

É apresentado na Figura 53, um exemplo de coleta de RSSI, que é um dos parâmetros a ser utilizado, pelo operador da rede, para avaliar a qualidade da mesma.

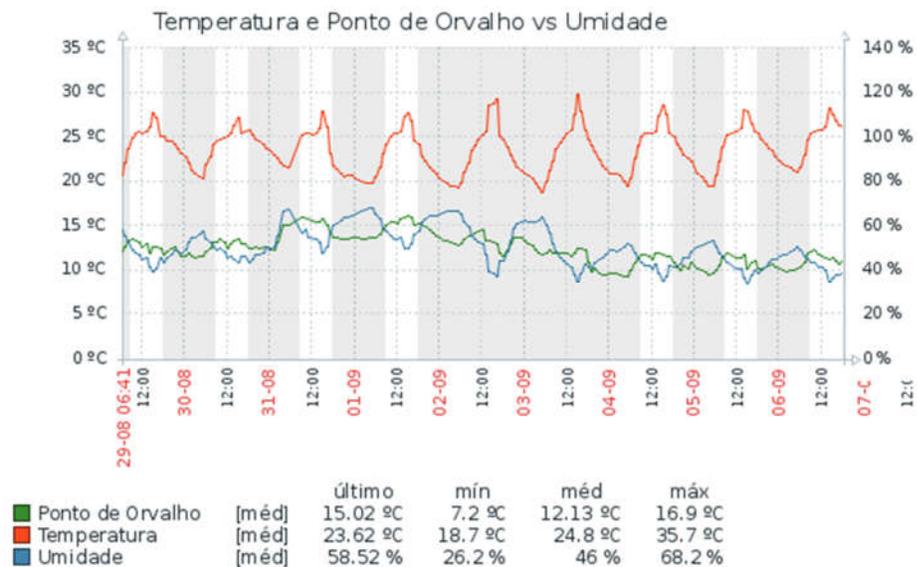
Figura 53 - Coleta de Intensidade de Sinal



Fonte: Elaboração Própria

Observa-se na Figura 54, o desenvolvimento de um gráfico com eixos Y personalizados, enquanto o eixo da esquerda apresenta escala de 5° C em 5° C, o eixo da direita exibe escala de 10% em 10 % de umidade.

Figura 54 - Coleta de Temperatura, Ponto de Orvalho e Umidade



Fonte: Elaboração Própria

Apesar da umidade atingir no máximo 100%, como o gráfico é dinâmico, além de serem utilizadas duas escalas diferentes, é necessário ampliar a escala da direita em função da escala da esquerda, por isso o indicativo no gráfico de 140%, mesmo sabendo que esse valor nunca será alcançado.

### **6.6.7. Aplicativos de Celular**

Utilizando o MIT App Inventor 2 (MIT, 2018), é possível criar aplicações para Android utilizando blocos de programação.

Em situações de implementação, por exemplo, pode-se encontrar problemas de comunicação. Nesse ponto o Zabbix Sender é utilizado para validar se o Servidor Zabbix está operacional e recebendo os dados, facilitando assim o foco na resolução dos problemas.

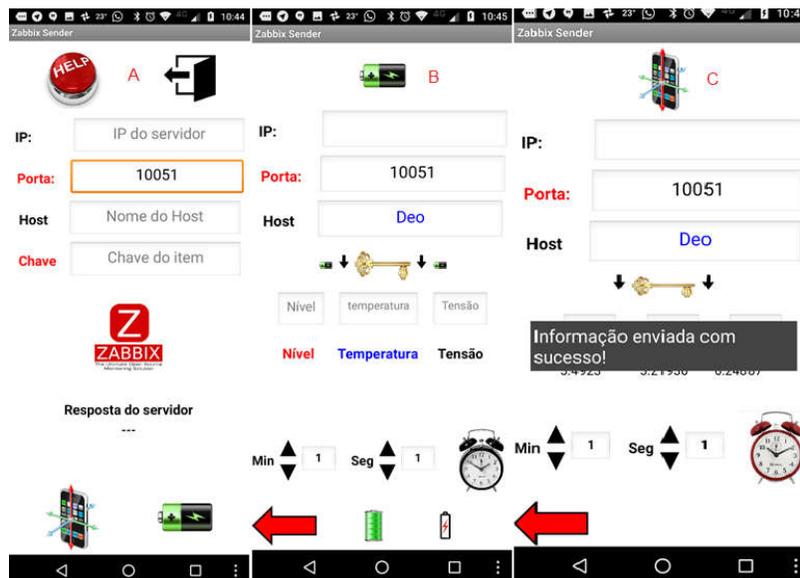
Pois, se o Servidor está operacional, o problema está na coleta de dados, ou na comunicação do elemento de borda com a Internet.

#### **Zabbix Sender**

Visto que já tinha sido implementada a função de Zabbix Sender via Linguagem de Programação Python, desenvolveu-se o Zabbix Sender para Android. O programa está disponível na Google Play Store e os códigos fontes foram disponibilizados (ZABBIXSENDER, 2017a).

É possível enviar dados para um item específico do Zabbix, Figura 55A, bastando para isso informar a chave; enviar o nível da bateria, Figura 55B, e o posicionamento do acelerômetro do celular, Figura 55C.

Figura 55 - Zabbix Sender para Android



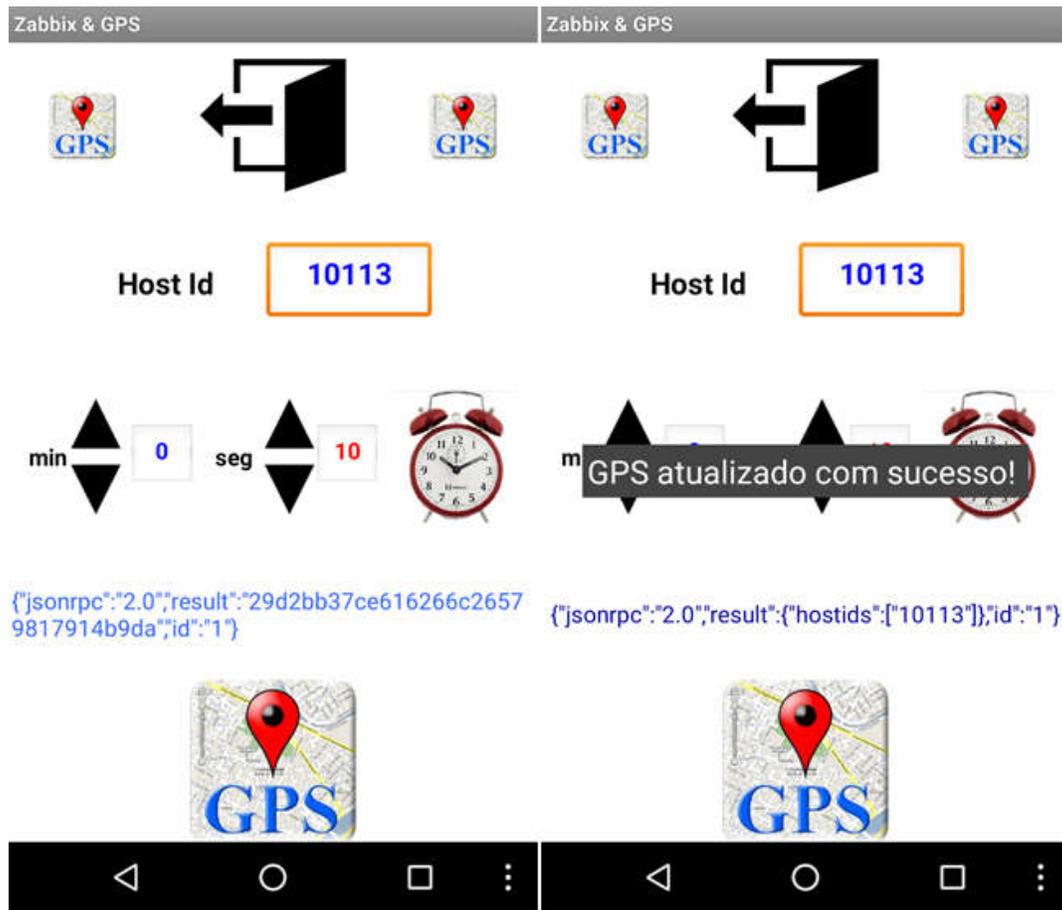
Fonte: Elaboração Própria

### Zabbix Geo

Uma vez que o MIT App Inventor 2 se mostrou eficiente para criar aplicativos Android compatíveis com o Zabbix, e devido à experiência do Caso da Fazenda Cafeeira, onde foi utilizado o monitoramento por GPS. Criou-se o aplicativo Zabbix Geo, disponível na Google Play Store e disponibilizou-se os códigos (ZABBIXGEO, 2018a).

É possível coletar as informações do GPS do celular e atualizar um item no Zabbix Server, Figura 56. Necessitando saber apenas o *Host Id* do elemento que contém o item a ser atualizado e informar o intervalo de atualização dos dados.

Figura 56 - Zabbix Geo para Android



Fonte: Elaboração Própria

As aplicações para essa solução são diversas, mas apenas para citar um exemplo, no caso da Fazenda Cafeeira, o proprietário poderia fornecer um celular para o tratorista, com o programa instalado e acompanhar em tempo real a posição do trator.

## 7. CONCLUSÃO

Atualmente, na mídia especializada vende-se a ideia de que a Internet das Coisas é algo simples de ser implementado, como por exemplo, a geladeira inteligente (TECHTUDO, 2018). O presente trabalho constatou que esse simplismo na realidade não existe, uma vez que as soluções precisam atender vários níveis de necessidades, para se atingir uma implementação real de IoT. As tecnologias existentes no mercado devem atender as necessidades da aplicação IoT e não o contrário.

O presente trabalho teve como objetivo inicial simplesmente uma abordagem de gerência de dados e redes, utilizando a ferramenta Zabbix. Porém, ao longo dos experimentos observou-se que não era possível atingir uma solução de IoT em sua completude sem pensar em todos os pontos do começo ao fim do processo. Nesse contexto, surgiu a necessidade do desenvolvimento de um modelo de referência que contemplasse a segmentação necessária para atender a aplicação de IoT em sua plenitude.

O modelo de referência proposto permite obter uma visão de todas as etapas necessárias para a implementação de uma solução de IoT. Ele abrange do atendimento das necessidades do negócio até a coleta dos dados, descrevendo os pontos de atenção em cada um dos níveis.

O modelo é *Open Source*, permitindo assim que o mesmo seja alterado e adaptado de acordo com as necessidades. Este é um benefício interessante tanto para estudantes que desejam desenvolver seus próprios projetos de IoT, quanto para profissionais que desejam prover soluções para o mercado de PME. Além disso, as tecnologias utilizadas na proposta também são *Open Source*.

Essa estratégia é principalmente interessante para PME, cujo o orçamento é restrito em função da própria natureza da empresa, que não dispõe de recursos para adoção de uma solução *Turn Key*, ou seja, de forma completa. Entretanto, essas soluções são onerosas e compatíveis com as necessidades de grandes empresas. Não sendo a realidade das PME.

As soluções propostas pelo Modelo de Referência podem ser substituídas por outras, parcialmente ou totalmente. Por exemplo, no nível 2,

os estudos de casos realizados utilizaram como tecnologia de RSSF o protocolo Radiuino e os módulos rádio BE900, porém, uma vez que seja mais viável a utilização da solução LoRa, por exemplo, não existe qualquer impedimento nessa substituição.

Outro exemplo é o do nível 4, onde utilizou-se um Zabbix Server para armazenamento dos dados na nuvem, que poderia ser substituído por uma plataforma como a Dojot, que se propõe além do armazenamento, realizar os níveis de abstração e exibição.

A desvantagem de adoção de tecnologias proprietárias é que elas não permitem a personalização, obrigando o negócio a se adaptar à solução. As tecnologias utilizadas no modelo de referência, por outro lado, permitem a escolha das soluções que sejam mais adequadas a realidade das PME, além de permitir personalizações.

Um exemplo, seria a adoção de uma solução que contemplasse os níveis 4 ao 6. Neste caso, a PME estaria condicionada a utilizar as soluções de armazenamento, abstração e exibição fornecidos por essa solução. Isto inviabilizaria a busca de novas soluções de abstração (nível 5), que seriam mais adequadas a realidade das PME.

Como trabalho futuro, sugere-se:

- Desenvolvimento de *scripts* capazes de utilizar a metodologia REST;
- Gerar uma saída em padrão JSON, com os elementos da rede, para aproveitar as funcionalidades do Zabbix de cadastro automático dos elementos;
- Implementações utilizando escalonamento de processos no elemento de borda;
- Detalhamento dos processos de comunicação, gerando o modelo lógico da solução;
- Criação de uma comunidade em volta do Modelo de Referência, para compartilhamento de experiências e desenvolvimento de novas funcionalidades.

## REFERÊNCIAS BIBLIOGRÁFICAS

3GPP. **The Mobile Broadband Standard**. 2018. Disponível em <<http://www.3gpp.org/technologies/keywords-acronyms/98-lte>>. Acesso em 20 de outubro de 2018.

AKIZUKIDENSHI. **Temperature and humidity module: DHT11 Product Manual**. 2018. Disponível em <<https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>>. Acesso em 14 de abril de 2018.

ANATEL. **Brasil registra redução de 2,88% no número de acessos em operação na telefonia móvel em 12 meses**. 2018. Disponível em <<http://www.anatel.gov.br/dados/destaque-1/283-brasil-tem-236-2-milhoes-de-linhas-moveis-em-janeiro-de-2018>>. Acesso em 17 de março de 2018.

ANDREDEO. **Personalizando o Idioma da Interface Gráfica do Zabbix**. 2017. Disponível em <<http://andredeo.blogspot.com.br/2017/06/personalizando-o-idioma-da-interface.html>>. Acesso em 02 de novembro de 2017.

ANDROID. **Android**. 2018. Disponível em <<https://www.android.com>>. Acesso em 05 de abril de 2018.

ARDUINO. **Arduino Mega 2560 Rev3**. 2017. Disponível em <<https://store.arduino.cc/usa/arduino-mega-2560-rev3>>. Acesso em 05 de março de 2017.

BASENGE. **Resfriamento Evaporativo**. 2018. Disponível em <<http://www.basenge.com.br/resfriamento-evaporativo>>. Acesso em 18 de outubro de 2018.

BEAGLEBOARD. **BeagleBone Black**. 2018. Disponível em <<https://beagleboard.org/black>>. Acesso em 14 de abril de 2018.

BLUETOOTH. **Bluetooth**. 2018. Disponível em <<https://www.bluetooth.com>>. Acesso em 20 de outubro de 2018.

BOLAND. **IoT Architecture and Design Patterns**. 2017. Disponível em <<https://panopto.aarnet.edu.au/Panopto/Pages/Viewer.aspx?id=3cbbc50c-49a5-4b54-8156-123ec79182e0>>. Acesso em 12 de outubro de 2017.

CARTA campinas. **54% dos empregos gerados no Brasil estão em micros e pequenas empresas.** 2017. Disponível em <<http://cartacampinas.com.br/2017/09/54-dos-empregos-gerados-no-brasil-estao-em-micros-e-pequenas-empresas>>. Acesso em 07 de outubro de 2017.

CETIC. **TIC DOMICÍLIOS 2017.** 2017. Disponível em <[https://cetic.br/media/analises/tic\\_domicilios\\_2017\\_coletiva\\_de\\_imprensa.pdf](https://cetic.br/media/analises/tic_domicilios_2017_coletiva_de_imprensa.pdf)>. Acesso em 22 de outubro de 2018.

CISCO. **Building the Internet of Things.** 2014. Disponível em <[https://daue6ehqissah.cloudfront.net/breakouts/2014/H-ARC-01\\_Cisco-Intel-IBM\\_FINAL.pdf](https://daue6ehqissah.cloudfront.net/breakouts/2014/H-ARC-01_Cisco-Intel-IBM_FINAL.pdf)>. Acesso em 20 de outubro de 2018.

CISCO. **Internet das Coisas (IoT): A IoT conecta objetos à Internet, gerando dados e informações aos quais nunca tivemos acesso antes.** 2017. Disponível em <[https://www.cisco.com/c/pt\\_br/solutions/internet-of-things/overview.html#~stickynav=3](https://www.cisco.com/c/pt_br/solutions/internet-of-things/overview.html#~stickynav=3)>. Acesso em 16 de setembro de 2017.

CISCO. **Welcome to Cisco IBSG.** 2018a. Disponível em <<http://keminet.net/source/web/about/ac79/index.html>>. Acesso em 20 de outubro de 2018.

COMPUTERWORLD. **IoT é um grande e confuso campo à espera de explodir.** 2014. Disponível em <<https://computerworld.com.br/2014/11/25/iot-e-um-grande-e-confuso-campo-a-espera-de-explodir/>>. Acesso em 07 de agosto de 2017.

CONVERGENCIA digital. **Quadruplica o número de ataques DDoS originados por dispositivos IoT no Brasil.** 2018. Disponível em <<http://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&inford=47544&sid=18>>. Acesso em 25 de maio de 2018.

CRUZ, M. A. A da et al. **A Proposal for Bridging the Message Queuing Telemetry Transport Protocol to HTTP on IoT Solutions.** 2018. Disponível em <<https://ieeexplore.ieee.org/document/8267034>>. Acesso em 20 de outubro de 2018.

CUBIEBOARD. **CubieBoard: A series of open source hardware.** 2018. Disponível em <<http://cubieboard.org>>. Acesso em 14 de abril de 2018.

DOCKER. **WHAT IS DOCKER.** Disponível em <<https://www.docker.com/what-docker>>. Acesso em 20 de outubro de 2017.

DOJOT. dojot. 2017. Disponível em <<http://www.dojot.com.br>>. Acesso em 20 de outubro de 2017.

ECOMMERCENEWS. **Trend Micro alerta: roteadores são principal alvo em IoT no Brasil**. 2018. Disponível em <<https://ecommercenews.com.br/noticias/pesquisas-noticias/trend-micro-alerta-roteadores-sao-principal-alvo-em-iot-no-brasil/>>. Acesso em 25 de maio de 2018.

EMERSON. **Emerson's Virtual Wireless Plant Tour**. 2017. Disponível em <<http://videos.emersonprocess.com/detail/videos/wireless/video/4005559660001/emerson-s-virtual-wireless-plant-tour?autostart=true>>. Acesso em 16 de setembro de 2017.

EVANS. **The Internet of Things - How the Next Evolution of the Internet Is Changing Everything**. 2011. Disponível em <[https://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf)>. Acesso em 20 de outubro de 2018.

EVERYZ. **EveryZ.org | The enterprise Zabbix Plugin**. 2017a. Disponível em <<http://www.everyz.org/pt/>>. Acesso em 02 de novembro de 2017.

EVERYZ. **ZabGeo – Módulo de Geolocalização**. 2017b. Disponível em <<http://www.everyz.org/pt/2017/04/10/geolocalizacao/>>. Acesso em 02 de novembro de 2017.

FILIFELOP. **Placa Uno R3 + Cabo USB para Arduino**. 2018. Disponível em <<https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino>>. Acesso em 22 de maio de 2018.

FORTUNE500. **FORTUNE500**. 2017. Disponível em <<http://fortune.com/fortune500>>. Acesso em 02 de novembro de 2017.

FREITAS, R. N. BRANQUINHO, O. C. **Segurança em Redes de Sensores Sem Fio**. BTSym 2017 Proceedings. ISSN 2447-8326. V.1. Disponível em <<http://lcv.fee.unicamp.br/images/BTSym-17/Papers/73482.pdf>>. 2017. Acesso em 22 de maio de 2018.

FREITAS, Rogério Nunes de. **Impacto no desempenho de RSSF com diferentes técnicas de cifragem**. 2018. 124p. Dissertação (mestrado) - Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Gestão de Redes de Telecomunicações. Campinas - SP.

GAILLARD Guillaume, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois. **Service Level Agreements for Wireless Sensor Networks: a WSN Operator's Point of View**. IEEE/IFIP NOMS - Network Operations and Management Symposium, May 2014, Krakow, Poland. 2014. <hal-00998906>

GITHUB. **A script to generate Zabbix map XML from PNG images**. 2011. Disponível em <[https://github.com/zabbix/zabbix/blob/trunk/misc/images/png\\_to\\_xml.sh](https://github.com/zabbix/zabbix/blob/trunk/misc/images/png_to_xml.sh)>. Acesso em 02 de novembro de 2017.

GROVE. **Grove System**. 2017. Disponível em <[http://wiki.seeedstudio.com/Grove\\_System](http://wiki.seeedstudio.com/Grove_System)>. Acesso em 05 de março de 2017.

HORST, A. H. S.; PIRES, A. S.; DÉO, A. L. B. **De A a Zabbix**. 1. ed. São Paulo: Novatec Editora Ltda., 2015. Cap. 01.

IBGE. **Projeção da população do Brasil e das Unidades da Federação**. 2018. Disponível em <<https://www.ibge.gov.br/apps/populacao/projecao/>>. Acesso em 17 de março de 2018.

IBM. **The Internet of Things becomes the Internet that thinks with Watson IoT**. 2017. Disponível em <<https://www.ibm.com/internet-of-things>>. Acesso em 16 de setembro de 2017.

IDGNOW. **Em 2021, mercado brasileiro de Internet das Coisas será de US\$ 3,29 bilhões**. 2017. Disponível em <<http://idgnow.com.br/ti-corporativa/2017/06/18/em-cinco-anos-mercado-de-iot-no-brasil-vai-valer-us-3-29-bilhoes>>. Acesso em 16 de setembro de 2017.

IDGNOW. **Pesquisadores descobrem 'rede zumbi' que ataca dispositivos IoT**. 2018. Disponível em <<http://idgnow.com.br/internet/2018/10/10/pesquisadores-descobrem-rede-zumbi-que-ataca-dispositivos-iot/>>. Acesso em 15 de novembro de 2018.

IEEE. **IEEE 802.15 WPAN™ Task Group 4 (TG4)**. 2018. Disponível em <<http://www.ieee802.org/15/pub/TG4.html>>. Acesso em 14 de abril de 2018.

INOVA. **Alunos desenvolvem projetos para melhorar o cotidiano na universidade**. 2017. Disponível em <<https://www.inova.unicamp.br/noticia/alunos-desenvolvem-projetos-para-melhorar-o-cotidiano-na-universidade>>. Acesso em 25 de maio de 2018.

IOTWF. **Internet of Things World Forum**. 2014. Disponível em <<https://www.iotwf.com>>. Acesso em 20 de outubro de 2018.

ITU-T. **Recommendation Y.2060 - Overview of the Internet of things**. 2012. Disponível em <<https://www.itu.int/rec/T-REC-Y.2060-201206-I>>. Acesso em 15 de outubro de 2018.

ITU-T. **Recommendation Y.4200 - Requirements for the Interoperability of Smart City Platforms**. 2018b. Disponível em <<https://www.itu.int/rec/T-REC-Y.4200/en>>. Acesso em 15 de outubro de 2018.

ITU-T. **Recommendation Y.4418 - Gateway Functional Architecture for Internet of Things Applications**. 2018a. Disponível em <<https://www.itu.int/rec/T-REC-Y.4418-201806-I/en>>. Acesso em 15 de outubro de 2018.

JAVA. **Software Java**. 2017. Disponível em <<https://www.oracle.com/br/java/>>. Acesso em 12 de outubro de 2017.

KONKER. **A Plataforma Konker**. 2018. Disponível em <<http://www.konkerlabs.com>>. Acesso em 18 de outubro de 2018.

LOCKITO. **Lockito – Fake GPS itinerary**. 2018. Disponível em <[https://play.google.com/store/apps/details?id=fr.dvilleneuve.lockito&hl=pt\\_BR](https://play.google.com/store/apps/details?id=fr.dvilleneuve.lockito&hl=pt_BR)>. Acesso em em 14 de abril de 2018.

LORA. **LoRa Alliance**. 2018. Disponível em <<https://www.lora-alliance.org>>. Acesso em 17 de março de 2018.

MAPBOX. **mapbox**. 2017. Disponível em <<https://www.mapbox.com>>. Acessado em 02 de novembro de 2017.

MATH Open Reference. **Outlier**. Disponível em <https://www.mathopenref.com/outlier.html>. Acesso em 22 de maio de 2018.

MATTHIESEN et al. **Repeater Protocol to Extend Signal Coverage**. Sensors & Transducers, Vol. 218, Issue 12, December 2017, pp. 27-31. 2017. Disponível em [http://www.sensorsportal.com/HTML/DIGEST/december\\_2017/Vol\\_218/P\\_2964.pdf](http://www.sensorsportal.com/HTML/DIGEST/december_2017/Vol_218/P_2964.pdf). Acesso em 22 de maio de 2018.

MATTHIESEN, Eloisa Alexandre Nielsen. 2018. **Proposta de escopo de engenharia de IoT: estudos de casos**. 2018. 97p. Dissertação (mestrado) - Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Programa de Pós-Graduação em Infraestrutura Urbana. Campinas - SP.

MIT App Inventor. **Anyone Can Build Apps That Impact the World**. 2018. Disponível em <http://appinventor.mit.edu/explore/>. Acesso em 10 de Janeiro de 2018.

MUNDO Educação. **Umidade do Ar**. 2017. Disponível em <https://mundoeducacao.bol.uol.com.br/geografia/umidade-ar.htm>. Acesso em 12 de outubro de 2017.

OLIVEIRA, Rodolfo Francisco de. **Proposta de um Proxy Manager para a Internet das coisas**. 2016. 160p. Dissertação (mestrado) - Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Gestão de Redes de Telecomunicações. Campinas - SP.

OPENSTREETMAP. **OpenStreetMap**. 2017. Disponível em <https://www.openstreetmap.org>. Acesso em 02 de novembro de 2017.

PUC Campinas. **Programa de Pós-Graduação em Sistemas de Infraestrutura Urbana**. 2017. Disponível em <https://www.puc-campinas.edu.br/pos-graduacao/programa-de-pos-graduacao-em-sistemas-de-infraestrutura-urbana-mestrado/>. Acesso em 18 de outubro de 2018.

PYTHON. **Python**. 2017. Disponível em <https://www.python.org>. Acesso em 05 de novembro de 2018.

PY-ZABBIX. **adubkov/py-zabbix**. 2017. Disponível em <<https://github.com/adubkov/py-zabbix>>. Acesso em 05 de março de 2017.

RADIOIT. **BE-900 - Módulo de comunicação sem fio**. 2017. Disponível em: <<http://www.radioit.com.br/datasheets/BE900-Datasheet-v1.5.pdf>>. Acesso em 13 de outubro de 2017.

RADIUINO. **Radiuino**. 2017. Disponível em <<http://radiuino.cc>>. Acesso em 05 de março de 2017.

RASPBERRY. **Raspberry Pi 3 Model B**. 2017. Disponível em <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b>>. Acesso em 28 de outubro de 2017.

RASPBIAN. **Welcome to Raspbian**. 2018. Disponível em <<https://www.raspbian.org>>. Acesso em 14 de abril de 2018.

REAL, Cilene Renata. **Medida e análise de comportamento da RSSI de uma rede de sensor sem fio em ambiente industrial**. 2015. 100p. Dissertação (mestrado) - Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Engenharia Elétrica. Campinas - SP.

RUIZ, Linnyer Beatrys. **MANA: Uma Arquitetura para Gerenciamento de Redes de Sensores Sem Fio**. Dissertação (Doutorado em Ciência da Computação) – Programa de Pós-graduação em Ciência da Computação, Universidade Federal de Minas Gerais, 2003. Disponível em <<https://www.dcc.ufmg.br/pos/cursos/defesas/128D.PDF>>. Acesso em 13 de fevereiro de 2018.

SCADABR. **ScadaBR**. 2017. Disponível em <<http://www.scadabr.com.br>>. Acesso em 02 de novembro de 2017.

SERVERWORKS. **[Zabbix How To] Get an environmental data with feelers [Zabbix IoT]**. 2017. Disponível em <<http://blog.serverworks.co.jp/tech/2017/04/05/zabbix-how-to-get-an-environmental-data-with-feelers-zabbix-iot>>. Acesso em 28 de outubro de 2017.

SIGFOX. **sigfox**. 2017. Disponível em <<https://www.sigfox.com/en>>. Acesso em 20 de outubro de 2017.

SIGFOX. **Sigfox Technology Overview**. 2018a. Disponível em <<https://www.sigfox.com/en/sigfox-iot-technology-overview>>. Acesso em 17 de março de 2018.

SIGFOX. **Sigfox Technical Overview**. 2018b. Disponível em <<https://www.sigfox.com/en/sigfox-iot-technology-overview> >. Acesso em 15 de setembro de 2018.

SILVA, Guilherme Lopes da. **Alunos desenvolvem projetos para melhorar o cotidiano na universidade**. 2017. Disponível em <<https://www.inova.unicamp.br/noticia/alunos-desenvolvem-projetos-para-melhorar-o-cotidiano-na-universidade>>. Acesso em 15 de setembro de 2018.

SMITH, S. W., *The Scientist and Engineer's Guide to Digital Signal Processing*, San Diego, CA: California Technical Publishing, 2006. Chap. 2.

SOLARBOTICS. **Grove - Temperature and Humidity Sensor**. 2017. Disponível em <<https://solarbotics.com/product/29080/>>. Acesso em 05 de março de 2017.

STALLINGS, W. **SNMP, SNMPv2, SNMPv3, and RMON 1 and 2**. Third Edition. Reading, MA: Addison-Wesley, 1998.

TAGO. **Tago**. 2017. Disponível em <<https://tago.io>>. Acesso em 20 de outubro de 2017.

TECHTUDO. **O que é Internet das Coisas? Dez fatos que você precisa saber sobre IoT**. 2018. Disponível em <<https://www.techtudo.com.br/listas/2018/08/o-que-e-internet-das-coisas-dez-coisas-que-voce-precisa-saber-sobre-iot.ghtml>>. Acesso em 15 de outubro de 2018.

TELEGRAM. **Telegram a new era of messaging**. 2017. Disponível em <<https://telegram.org>>. Acesso em 05 de março de 2017.

THINKSPEAK. **ThinkSpeak**. 2017. Disponível em <<https://thingspeak.com>>. Acesso em 20 de outubro de 2017.

TOMASINI, O. C. et al. **Environmental Monitoring Using Wireless Sensor Network For Emergency Conditions**. BTSYM 2017 Proceedings. ISSN 2447-8326. V.1. 2017. Disponível em <<http://lcv.fee.unicamp.br/images/BTSym-17/Papers/76901.pdf>>. Acesso em 22 de maio de 2018.

UBUNTU. **Zabbix + IoT/Snappy: allows users to feel the pulse of things**. 2015. Disponível em <<https://insights.ubuntu.com/2015/06/12/zabbix-iotsnappy-allows-users-to-feel-the-pulse-of-things/>>. Acesso em 11 de julho de 2017.

UBUNTU. **Ubuntu**. 2018. <<https://www.ubuntu.com>>. Acesso em 14 de abril de 2018.

UNICAMP. **[Cinfotec 2017] Zabbix para IoT**. 2017. Disponível em <[http://cameraweb.ccuec.unicamp.br/watch\\_video.php?v=4AG2AD6RUBRR](http://cameraweb.ccuec.unicamp.br/watch_video.php?v=4AG2AD6RUBRR)>. Acesso em 28 de outubro de 2017.

UNIREDE. **Zabbix 3.0 Virando a página na evolução do monitoramento**. 2016. Disponível em <<https://www.unirede.net/zabbix-3-0-virando-pagina-na-evolucao-do-monitoramento/>>. Acesso em 11 de julho de 2017.

VEGETRONIX. **VH400 Rev A to Rev G: VWC to Voltage Curves**. 2018. Disponível em <<http://www.vegetronix.com/Curves/VH400-RevA/VG400-RevA-Curves.phtml>>. Acesso em 22 de maio de 2018.

VMWARE. **vmware**. 2017. Disponível em <<https://www.vmware.com/br.html>>. Acesso em 12 de outubro de 2017.

WI-FI. WI-FI ALLIANCE. 2018. Disponível em <<https://www.wi-fi.org>>. Acesso em 20 de outubro de 2018.

WINDOWS. **Windows 10 Internet of Things**. 2018. Disponível em <<https://developer.microsoft.com/pt-br/windows/iot>>. Acesso em 14 de abril de 2018.

ZABBIX. **Zabbix in Industry AT**. 2016. Disponível em <[https://www.zabbix.com/conf\\_latam\\_2016\\_agenda](https://www.zabbix.com/conf_latam_2016_agenda)>. Acesso em 01 de novembro de 2017.

ZABBIX. **ZABBIX The Enterprise-class Monitoring Solution for Everyone**. 2017a.

Disponível em <<https://www.zabbix.com>>. Acesso em 10 de outubro de 2017.  
ZABBIX. **True Open Source**. 2017b. Disponível em <[https://www.zabbix.com/true\\_open\\_source](https://www.zabbix.com/true_open_source)>. Acesso em 10 de outubro de 2017.

ZABBIX. **Zabbix Documentation 3.4**. 2017c. Disponível em <<https://www.zabbix.com/documentation/3.4/manual/concepts>>. Acesso em 02 de novembro de 2017.

ZABBIX. **CASE STUDY: Using Zabbix in an IoT architecture**. 2017d. Disponível em <[https://www.zabbix.com/conf2017\\_agenda](https://www.zabbix.com/conf2017_agenda)>. Acesso em 27 de outubro de 2017.

ZABBIX. **About Zabbix LLC**. 2017e. Disponível em <<https://www.zabbix.com/about>>. Acesso em 02 de novembro de 2017.

ZABBIX. **Zabbix in 2017**. 2017f. Disponível em <[https://www.zabbix.com/files/zabconf2017/alexei\\_vladishev-opening\\_speech.pdf](https://www.zabbix.com/files/zabconf2017/alexei_vladishev-opening_speech.pdf)>. Acesso em 02 de novembro de 2017.

ZABBIX. **ZABBIX BUGS AND ISSUES**. 2017g. Disponível em <<https://support.zabbix.com/browse/ZBX/?selectedTab=com.atlassian.jira.jira-projects-plugin:issues-panel>>. Acesso em 02 de novembro de 2017.

ZABBIX. **Pré-requisitos**. 2017h. Disponível em <<https://www.zabbix.com/documentation/3.4/pt/manual/installation/requirements>>. Acesso em 20 de dezembro de 2017.

ZABBIX. **Tipos de itens**. 2017i. Disponível em <<https://www.zabbix.com/documentation/3.4/pt/manual/config/items/itemtypes>>. Acesso em 20 de dezembro de 2017.

ZABBIX. **Zabbix Agent (UNIX)**. 2017j. Disponível em <[https://www.zabbix.com/documentation/3.4/pt/manual/appendix/config/zabbix\\_agentd](https://www.zabbix.com/documentation/3.4/pt/manual/appendix/config/zabbix_agentd)>. Acesso em 20 de dezembro de 2017.

ZABBIX. **Sender**. 2017k. Disponível em <<https://www.zabbix.com/documentation/3.4/pt/manual/concepts/sender>>. Acesso em 20 de dezembro de 2017.

ZABBIX. **Triggers.** 2018a. Disponível em <https://www.zabbix.com/documentation/3.4/pt/manual/config/triggers>>. Acesso em em 21 de abril de 2018.

ZABBIX. **Ações.** 2018b. Disponível em <https://www.zabbix.com/documentation/3.4/pt/manual/config/notifications/action>>. Acesso em em 21 de abril de 2018.

ZABBIX. **Zabbix Summit 2018 Agenda.** 2018c. Disponível em [https://www.zabbix.com/zabbix\\_summit2018\\_agenda#day1\\_1630](https://www.zabbix.com/zabbix_summit2018_agenda#day1_1630)>. Acesso em 01 de novembro de 2018.

ZABBIX. **024 zabbix para iot.** 2018d. Disponível em <https://www.youtube.com/watch?v=g9eBR5knBAA>>. Acesso em 08 de novembro de 2018.

ZABBIXGEO Android. **Arquivos fonte no formato do projeto MIT APP Inventor.** 2018a. Disponível em [https://drive.google.com/drive/folders/1aQ1kC12txoKCx8xGI\\_hJSQaPDpW76se6](https://drive.google.com/drive/folders/1aQ1kC12txoKCx8xGI_hJSQaPDpW76se6)>. Acesso em 12 de maio de 2018.

ZABBIXGEO. **Zabbix Geo - Zabbix Brasil.org.** 2018b. Disponível em [https://play.google.com/store/apps/details?id=appinventor.ai\\_GuiTalos.Post\\_GPS\\_Zabbix](https://play.google.com/store/apps/details?id=appinventor.ai_GuiTalos.Post_GPS_Zabbix)>. Acesso em 27/04/2018.

ZABBIX Sender Android. **Arquivos fonte no formato do projeto MIT APP Inventor.** 2017a. Disponível em [https://drive.google.com/drive/folders/1aQ1kC12txoKCx8xGI\\_hJSQaPDpW76se6](https://drive.google.com/drive/folders/1aQ1kC12txoKCx8xGI_hJSQaPDpW76se6)>. Acesso em 15 de novembro de 2017.

ZABBIXSENDER. **Zabbix Sender - Zabbix Brasil.org** . 2017b. Disponível em [https://play.google.com/store/apps/details?id=appinventor.ai\\_GuiTalos.Zabbix\\_post\\_autenticacao](https://play.google.com/store/apps/details?id=appinventor.ai_GuiTalos.Zabbix_post_autenticacao)>. Acesso em 14 de abril de 2018.

ZIGBEE. **Zigbee Alliance.** 2017. Disponível em <http://www.zigbee.org>>. Acesso em 17 de março de 2018.

Z-WAVE. **Safer. Smarter. Z-Wave.** 2018. Disponível em <https://www.z-wave.com>>. Acesso em 20 de outubro de 2018.

## APÊNDICES

### Apêndice A – Questionário

#### Questionamentos Rede <Nome da Empresa/Cliente/Projeto>

##### Parte 01 – Informações do Cliente

1. Quais coletas serão realizadas?
2. Para cada item:
  - A) Como o dado será enviado?
    - a. Digital
    - b. Analógico
  - B) Qual o tempo de coleta?
  - C) Qual o período de retenção de coletas individuais?
  - D) Qual o período de retenção de média diária de coletas?
3. Você gostaria de apresentar os dados em forma de gráficos? Quais dados e quais tipos de gráficos?
4. Você gostaria de apresentar os dados em forma de mapas? Quais mapas - Planta Baixa, Mapa para o Operador da Rede, Exibição de Links?
5. Você gostaria de agregar vários valores em uma única tela? Quais valores?
6. Você gostaria de apresentar essas telas de forma dinâmica (Slide Shows)? De quanto em quanto tempo?
7. Você gostaria de estabelecer *thresholds*? Quais valores e para quais métricas?
8. Você gostaria de ser alertado caso algum sensor pare de funcionar? De que maneira (E-mail, SMS, Telegram)? Após quanto tempo de inatividade?
9. Você gostaria de ser alertado caso alguma métrica atinja determinado *thresholds*? De que maneira (E-mail, SMS, Telegram)? Após quanto tempo de inatividade?

##### Parte 02 – Informações do Operador de RSSF

1. Quantos sensores serão instalados?
2. Quantas bases serão instaladas?
3. Qual canal será utilizado?
4. Qual o tipo de Antena da Base?
5. Qual o tipo de Antena dos Sensores?
6. A antena da Base é do tipo Interna ou Externa?
7. A antena dos Sensores é do tipo Interna ou Externa?
8. Qual o tipo de Rádio?

9. Qual a potência ajustada de Transmissão?
10. Será realizado offset do medidor de potência?
11. Qual a taxa de transmissão?
12. Será utilizado repetidor?
13. Caso Afirmativo - Qual a distância física:
  - A) Repetidor em relação à base
  - B) Cada um dos sensores em relação ao repetidor
  - C) O Repetidor realizará alguma coleta? Qual (is)
14. Caso Negativo - Qual a distância física:
  - A) Cada um dos sensores em relação à Base

**Para cada um dos sensores responda:**

1. Os sensores são analógicos ou digitais?
2. Qual o modelo do sensor?
3. Como será a alimentação desse sensor?
  - A) Alimentação Externa (Tomada)
  - B) Alimentação Interna por bateria (Pilha)
  - C) Alimentação Externa por bateria (12V)
  - D) Alimentação Interna por bateria de Chumbo
  - E) Alimentação Interna por bateria de Lítio
4. O Sensor possui Paineis Fotovoltaico?
5. Caso Afirmativo:
  - A) Quantos Painéis?
  - B) Qual a Potência?
  - C) Qual a Tensão?
6. O sensor será Indoor ou Outdoor?
7. O sensor precisará de proteção extra? Qual a classificação de acordo com o Índice de Proteção?
  - A) Camuflagem
  - B) Prova d'água

**Apêndice B – Arquivos meduza**

<https://github.com/andredeo/meduza>

**Apêndice C – Arquivos Arduino Emulando Radiuino**

<https://github.com/andredeo/arduino/tree/master/Radiuino%20Emulado>

## **Apêndice D – Instruções para Personalização do Idioma da Interface Gráfica do Zabbix**

<http://andredeo.blogspot.com/2017/06/personalizando-o-idioma-da-interface.html>

## ANEXOS

### **Anexo A – *Script* Gerar XML para Importar Imagens em Massa no Zabbix**

[https://github.com/zabbix/zabbix/blob/trunk/misc/images/png\\_to\\_xml.sh](https://github.com/zabbix/zabbix/blob/trunk/misc/images/png_to_xml.sh)