ADEMIR FERREIRA DA SILVA

Sensoriamento automático e participativo em cidades

SÃO PAULO

2016

ADEMIR FERREIRA DA SILVA

Sensoriamento automático e participativo em cidades

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para a obtenção do título de Mestre em
Ciências.

SÃO PAULO

2016

ADEMIR FERREIRA DA SILVA

Sensoriamento automático e participativo em cidades

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para a obtenção do título de Mestre em
Ciências.

Área de Concentração: Sistemas
Eletrônicos

Orientador: Prof. Dr. Marcelo Knorich Zuffo

SÃO PAULO

2016

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Silva, Ademir Ferreira

Sensoriamento Automático e Participativo em Cidades / A. F. Silva --
versão corr. -- São Paulo, 2016.
191 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Cidades Inteligentes 2.Redes de Sensores 3.Middlewares 4.Participação
Comunitária I.Universidade de São Paulo. Escola Politécnica. Departamento
de Engenharia de Sistemas Eletrônicos II.t.

AGRADECIMENTOS

Aos amigos da IBM Research, que ao seu modo me estenderam a mão quando precisei.

A minha família, que gentilmente entendeu que os momentos da minha ausência foram fundamentais para a conclusão desta jornada.

Agradeço especialmente a minha querida esposa, Marcia Cristina, que incondicionalmente me apoiou neste projeto. Sem o seu ombro amigo nada disto seria possível.

Resumo

As cidades estão a seu tempo e a seu modo, modernizando os serviços prestados à população. Entre os diversos fatores que estão contribuindo para esta evolução estão a diversificação e proliferação de sensores, nos diversos domínios de serviços das cidades, e os novos canais de comunicação com os munícipes, entre eles, as redes sociais e mais recentemente os sistemas *crowdsensing*, motivados pelos anseios sociais, por melhores serviços públicos e pela popularização dos dispositivos móveis. Nesta direção, a eficiência administrativa é um fator essencial, uma vez que as cidades estão se mostrando mais complexas na medida em que cresce a população nas áreas urbanas. A utilização de técnicas de sistemas distribuídos para que múltiplos domínios de serviços usufruam da mesma infraestrutura computacional, pode auxiliar na eficiência das cidades, evitando gastos administrativos duplicados e até mesmo, possibilitando a correlação de eventos entre os serviços, favorecendo a identificação de fatores de causalidades e assim, a tomada de decisões administrativas mais objetivas e precisas. Neste contexto, este trabalho concentra-se na análise de um *middleware* direcionado à gestão de cidades para coleta, integração e interpretação dos dados de sensores, pertencentes aos serviços disponíveis da própria cidade, junto com os dados do sensoriamento colaborado pelos cidadãos. Para avaliação do conceito foi investigado o cenário de monitoração da conservação de vias públicas. Após 3 meses de coletas de dados por um sistema de sensoriamento automático, totalizando mais de 360 mil pontos e também mais de 90 relatórios pelo sensoriamento participativo, verificou-se que um sistema distribuído pode realizar a interpretação de séries históricas, engajar os munícipes apoiar a manutenção dos serviços da cidade e também indicar objetivamente aos gestores públicos os pontos que devem ser prioritariamente atendidos. Aliar ferramentas pelas quais o cidadão pode, de acordo com sua necessidade, convicção e altruísmo, exercer influência nos gestores públicos com o suporte de informação contínua e critérios objetivos das redes de sensores, pode estimular a continua excelência dos serviços públicos.

Palavras chaves: Rede de sensores. Cidades inteligentes. *Middleware*. *Crowdsensing*

Abstract

The cities in their own way and time are improving their services provided to the population. Among several factors that are contributing to this trend are the diversification and proliferation of sensors in various services domains of cities and new communication ways with citizens, for instance, social networks and more recently, crowdsensing systems, motivated by social expectations for better public services and the popularity of mobile devices. In this direction, administrative efficiency is a key factor, since the cities are proving more complex with increasing the population in urban areas. Techniques of distributed systems to share the same computing infrastructure to multiple service domains, can assist in the efficiency of cities, avoiding duplicate administrative costs and even allowing event correlation between services, providing the identification of causality factors, thus making management decisions more objective and accurate. In this context, this research focuses on analysis of a middleware directed to city management for collection, integration and interpretation of sensors data, present in city services, along with the sensing data contributed by citizens. For concept evaluation, was investigated the scenario of conservation of public streets. After 3 months of data collection by an automatic sensing system comprising more than 360 thousand points and also 94 reports of collaborative sensing, it was found that a distributed system can perform the interpretation of historical series; engage the citizens to support maintenance of city services and indicate objectively the points that should primarily be fix by public managers. Combining tools, which citizens can, according to their need, conviction, altruism, exert their own influence in public management and the continuous information support to objective criteria of sensor networks, can stimulate the continued excellence of public services.

Keywords: *Sensor network, smart cities, middleware, crowdsensing*

LISTA DE SIGLAS E ABREVIATURAS

3G	<i>Third Generation</i>
4G	<i>Fourth Generation</i>
ADC	<i>Analog Digital Converter</i>
AES	<i>Advanced Encryption Standard</i>
BPSK	<i>Binary Phase Shift Keying</i>
CAP	<i>Contention Access Period</i>
CFP	<i>Contention-Free Period</i>
CPU	<i>Central Processing Unit</i>
CRC	<i>Cyclic Redundancy Check</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CSS	<i>Cascading Style Sheets</i>
DARPA	<i>Defense Advanced Research Project Agency</i>
DBaaS	<i>Database as a Service</i>
DC	<i>Direct Current</i>
DSM	<i>Distributed Shared Memory</i>
DSN	<i>Distributed Sensor Networks</i>
FFD	<i>Full Functional Device</i>
GPS	<i>Global Position System</i>
GPRS	<i>General Packet Radio Service</i>
GTS	<i>Guaranteed Time Slots</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IaaS	<i>Infrastructure as a Service</i>
IDL	<i>Interface Definition Language</i>
IETF	<i>Internet Engineering Task Force</i>
IOS	<i>iPhone Operating System</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
IPsec	<i>Internet Protocol Security</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
ISM	<i>Industrial, Scientific and Medical radio bands</i>
JDBC	<i>Java Database Connectivity</i>

JMS	<i>Java Message Service</i>
JSON	<i>JavaScript Object Notation</i>
KNN	<i>K - Nearest Neighbour</i>
LAN	<i>Local Area Network</i>
MAC	<i>Media Access Control</i>
MEMS	<i>Micro-Electromechanical Systems</i>
MER	Modelo de Entidade e Relacionamento
MQTT	<i>Message Queuing Telemetry Transport</i>
MQTT-SN	<i>Message Queuing Telemetry Transport for Wireless Sensor Networks</i>
NaaS	<i>Network as a Service</i>
NFC	<i>Near Field Communication</i>
O-QPSK	<i>Offset Quadrature Phase-Shift Keying</i>
PaaS	<i>Platform as a Service</i>
PCI	Placa de Circuito Impresso
QoS	<i>Quality of Service</i>
QR	<i>Quick Response</i>
REST	<i>Representational State Transfer</i>
RFD	<i>Reduced Functional Devices</i>
RMI	<i>Remote Method Invocation</i>
RNA	Rede Neural Artificial
RPC	<i>Remote Procedure Call</i>
RSSF	Rede de Sensores Sem Fio
RSSI	<i>Received Signal Strength Indicator</i>
S2aaS	<i>Sensing as a Service</i>
SaaS	<i>Software as a Service</i>
SD	Secure Digital Card
SQL	<i>Structured Query Language</i>
SVM	<i>Support Vector Machine</i>
TCP	<i>Transmission Control Protocol</i>
TI	Tecnologia de Informação
UDP	<i>User Datagram Protocol</i>
URL	<i>Uniform Resource Locator</i>
UUID	<i>Universally unique identifier</i>
WSN	<i>Wireless Sensor Network</i>
XML	<i>Extensible Markup Language</i>

LISTA DE FIGURAS

Figura 1: Componentes fundamentais de cidades inteligentes	17
Figura 2: Arranjo típico de RSSF	24
Figura 3: Composição de um nó sensor	25
Figura 4: Visão geral de aplicações em RSSF	29
Figura 5: Pilha de protocolos de rede de sensores.....	32
Figura 6: Arquitetura de aplicações de sensoriamento participativo	40
Figura 7: Interações do modelo de publicar e assinar	53
Figura 8: Camadas lógicas de serviço de software e hardware em sistemas distribuídos	56
Figura 9: Arquitetura de <i>middleware</i> para RSSF	58
Figura 10: Camadas de composição de <i>middleware</i> para <i>crowdsensing</i>	60
Figura 11: Foco do estudo - Intersecção dos conceitos de rede de sensores, <i>crowdsensing</i> e sistemas distribuídos	65
Figura 12: Elementos básicos do sistema: sensores, <i>crowdsensing</i> e sistema de gerenciamento	66
Figura 13: Visão geral da arquitetura	68
Figura 14: Arquitetura da rede de sensor	69
Figura 15: Arquitetura da camada de rede do dispositivo móvel.....	70
Figura 16: Arquitetura da aplicação dos nós sensores	71
Figura 17: Arquitetura da aplicação do sorvedouro	73
Figura 18: Arquitetura da aplicação <i>crowdsensing</i>	75
Figura 19: Arquitetura da aplicação do Servidor – Serviço de Computação em Nuvem.....	78
Figura 20: Componentes do gerenciador de eventos	79
Figura 21: Ilustração do experimento.....	83
Figura 22: Vista do <i>hardware</i> que compõe o nó sensor e a montagem final	84
Figura 23: Vista do <i>hardware</i> do sorvedouro e a montagem final	85
Figura 24: Montagem do sensor no veículo.....	86
Figura 25: Telas da aplicação <i>crowdsensing</i> : tela de apresentação, tela de contextos de interação disponíveis e tela de elaboração de relatórios.....	88
Figura 26: Telas da aplicação <i>crowdsensing</i> : Tela de histórico, alerta característico para auditoria da informação dos sensores e mapa de navegação	89
Figura 27: Infraestrutura de programação integradas no servidor para a construção do <i>middleware</i>	90
Figura 28: Modelo de dados e representação da alocação das tabelas no respectivo banco	91
Figura 29: Detalhes construtivos e de integração entre o servidor de mensagens, o servidor de aplicação e o banco de dados.....	92
Figura 30: Exemplo de configurações iniciais da aplicação <i>crowdsensing</i>	93
Figura 31: Detalhes construtivos e de integração do servidor de aplicação e o banco de dados, com ênfase no serviço de relatórios e gerenciador do banco de dados	94

Figura 32: Detalhes construtivos e de integração do servidor de aplicação e o banco de dados, com ênfase no serviço de relatórios e gerenciador do banco de dados	95
Figura 33: Detalhes construtivos e de integração do servidor de aplicação e o banco de dados, com ênfase no serviço de notificações	96
Figura 34: Visão geral das interfaces utilizadas no experimento.....	97
Figura 35: Representação das rotas analisadas no experimento.....	98
Figura 36: Características do conteúdo da mensagem do sensor	100
Figura 37: Distribuição da amostragem dos dados coletados pelo sensor	101
Figura 38: Perfil de aceleração das categorias analisadas	102
Figura 39: Quantidade e distribuição dos pontos de referências usados	103
Figura 40: Resultado da análise dos métodos de classificação do estado da via pública usando dados de velocidade e de um acelerômetro tri-axial	104
Figura 41: Análise dos métodos de classificação para detecção de buracos em via pública usando dados de velocidade e de um acelerômetro tri-axial.....	105
Figura 42: Resultado da análise de classificação do experimento	105
Figura 43: Visualização dos buracos encontrados pelo modelo analítico de classificação na interface gráfica do sistema de gerenciamento	106
Figura 44: Resultados dos relatórios coletados de <i>crowdsensing</i>	107
Figura 45: Visualização dos pontos do trajeto 1 avaliado de <i>crowdsensing</i>	108
Figura 46: Conjunto de 8 imagens coletadas por <i>crowdsensing</i>	108
Figura 47: Interface de gerenciamento da aplicação <i>crowdsensing</i>	109
Figura 48: Resultado das opiniões dos participantes no experimento	110
Figura 49: Testes de carga de processamento do <i>middleware</i>	111
Figura 50: Arquitetura SmartSantander	113
Figura 51: Arquitetura do projeto MetroSense	115
Figura 52: Arquitetura G-Sense do lado do Cliente	117
Figura 53: Arquitetura G-Sense do lado do Servidor	117
Figura 54: Ciclo de produção e consumo de dados de sensoriamento comunitário	118
Figura 55: Diagrama funcional do Centro de Operações de domínio Cruzado	119
Figura 56: Arquitetura de monitoração de microdrenagem urbana	128
Figura 57: Representação do cenário de sorvedouro móvel	128
Figura 58: Detalhes da arquitetura da plataforma de internet das coisas para monitoramento de motores industriais	130
Figura 59: Exemplo de topologia de RSSF	154
Figura 60: Detalhes do super quadro do IEEE 802.15.4	154
Figura 61: Detalhes do quadro de dados do IEEE 802.15.4 operando em 2450MHz.....	155
Figura 62: Modelo de referência da pilha de camadas para o protocolo 6LoWPAN.....	157
Figura 63: Configuração mínima e completa usando 6LoWPAN no quadro IEEE 802.15.4.....	157
Figura 64: Representação do encaminhamento local de pacotes em uma RSSF em malha com o protocolo 6LoWPAN.....	158

Figura 65: Arquitetura MQTT-SN	161
Figura 66: Modos de funcionamento do <i>gateway</i> MQTT-SN.....	162
Figura 67: Procedimento de conexão MQTT-SN.....	162
Figura 68: Procedimento de publicação do MQTT-SN	163
Figura 69: Modelos de negócios	165

LISTA DE TABELAS

Tabela 1: Trajetos percorridos na avaliação <i>crowdsensing</i>	107
Tabela 2: Principais características definidas pelo padrão IEEE 802.15.4	153
Tabela 3: Comparação entre 6LoWPAN, Zigbee e Zensys	156
Tabela 4: Comparação entre MQTT e HTTP	159
Tabela 5: Tabela do banco de dados - <i>Pothole</i>	167
Tabela 6: Tabela do banco de dados - <i>Categories Pothole</i>	167
Tabela 7: Tabela do banco de dados - <i>Marked Pothole</i>	168
Tabela 8: Tabela do banco de dados - <i>Applications</i>	168
Tabela 9: Tabela do banco de dados - <i>Search Area</i>	169
Tabela 10: Tabela do banco de dados - <i>Citizen Reports</i>	169
Tabela 11: Tabela do banco de dados - <i>Attributes</i>	169
Tabela 12: Tabela do banco de dados - <i>Categories</i>	170
Tabela 13: Tabela do banco de dados - <i>Participants</i>	170
Tabela 14: Conteúdo da interface <i>Set Search Area</i> - Notificações	172
Tabela 15: Conteúdo da interface <i>Warn</i> - Notificações	172
Tabela 16: Resultado da interface <i>Path</i> - Relatórios	173
Tabela 17: Resultado da interface <i>Classified Path</i> - Relatórios.....	173
Tabela 18: Resultado da interface <i>Crowdsensing</i> - Relatórios.....	174
Tabela 19: Conteúdo da interface <i>Post Sensors</i> - Gerenciador de dados	175
Tabela 20: Conteúdo da interface <i>Get Sensors</i> - Gerenciador de banco de dados.....	175
Tabela 21: Resultado da interface <i>Track Sensors</i> - Gerenciador de dados.....	176
Tabela 22: Conteúdo da interface <i>Post Report</i> - Gerenciador de dados.....	176
Tabela 23: Resultado da interface <i>Get Report</i> - Gerenciador de banco de dados.....	177
Tabela 24: Resultado da interface <i>Citizen Reports</i> - Gerenciador de dados	177
Tabela 25: Resultado da interface <i>Categories</i> - Gerenciador de banco de dados.....	178
Tabela 26: Conteúdo da interface <i>Participants</i> -Gerenciador de banco de dados	178
Tabela 27: Resultado da interface <i>Applications</i> - Gerenciador de banco de dados.....	178
Tabela 28: Resultado da interface <i>Attributes</i> - Gerenciador de banco de dados	179
Tabela 29: Resultado da interface <i>Classified Points</i> - Gerenciador de dados	179
Tabela 30: Conteúdo da interface <i>Classified Points</i> – Gerenciador de dados	180
Tabela 31: Resultado da interface <i>Search Points</i> - Gerenciador de dados.....	181
Tabela 32: Resultado da interface <i>Categories Pothole</i> - Gerenciador de dados	181
Tabela 33: Resultado da interface <i>get Search Area</i> - Gerenciador de dados	182
Tabela 34: Conteúdo da interface <i>post Search Area</i> - Gerenciador de dados.....	182
Tabela 35: Conteúdo do tópico <i>Application</i> - Gerenciador de eventos.....	183
Tabela 36: Conteúdo do tópico <i>Sensors</i> - Gerenciador de eventos	183
Tabela 37: Conteúdo do tópico <i>Report</i> - Gerenciador de eventos	184
Tabela 38: Conteúdo do tópico <i>Categories Attributes</i> - Gerenciador de eventos	184

Sumário

LISTA DE TABELAS	13
1 INTRODUÇÃO.....	16
1.1 MOTIVAÇÃO E OBJETIVOS	18
1.2 ORGANIZAÇÃO DO TRABALHO	20
2 REDE DE SENSORES	21
2.1 INTRODUÇÃO	21
2.2 CARACTERÍSTICAS DAS REDES DE SENSORES.....	23
2.3 DESAFIOS DAS REDES DE SENSORES.....	26
2.4 APLICAÇÕES DAS REDES DE SENSORES.....	28
2.5 ARQUITETURA DAS REDES DE SENSORES	31
2.6 CONSIDERAÇÕES FINAIS	34
3 CROWDSENSING.....	36
3.1 INTRODUÇÃO	37
3.3 ARQUITETURA DE SISTEMAS <i>CROWDSENSING</i>	40
3.4 DESAFIOS DE SISTEMAS <i>CROWDSENSING</i>	42
3.5 LIMITAÇÕES DE RECURSOS	44
3.6 MECANISMOS DE AGREGAÇÃO E ANÁLISE	45
3.7 CONSIDERAÇÕES FINAIS	46
4 SISTEMAS DISTRIBUÍDOS.....	49
4.1 INTRODUÇÃO	49
4.2 DESAFIOS DOS SISTEMAS DISTRIBUÍDOS	50
4.3 PARADIGMAS DE COMUNICAÇÃO DE SISTEMAS DISTRIBUÍDOS.....	51
4.4 <i>MIDDLEWARE</i>	55
4.5 <i>MIDDLEWARE</i> PARA REDE DE SENSORES	56
4.6 CARACTERÍSTICAS DE <i>MIDDLEWARE</i> PARA REDE DE SENSORES.....	58
4.7 <i>MIDDLEWARE</i> PARA <i>CROWDSENSING</i>	59
4.8 CARACTERÍSTICAS DE <i>MIDDLEWARE</i> PARA <i>CROWDSENSING</i>	61
4.9 CONSIDERAÇÕES FINAIS	63
5 DESENVOLVIMENTO	65
5.1 ESCOPO	67
5.2 ARQUITETURA DE REDE	68
5.3 FUNCIONALIDADES ESPERADAS DO <i>MIDDLEWARE</i> PARA O NÓ SENSOR, SORVEDOURO E DISPOSITIVO MÓVEL DO PARTICIPANTE	70
5.4 CARACTERÍSTICAS DO <i>MIDDLEWARE</i> NO SERVIDOR.....	77

5.5	CENÁRIO DE APLICAÇÃO	81
5.6	EXPERIMENTO	82
5.7	RESULTADOS.....	98
6	TRABALHOS RELACIONADOS	112
6.1	SMARTSANTANDER	112
6.2	METROSENSE.....	114
6.3	G-SENSE.....	116
6.4	OPENSENSE	118
6.5	CENTRO DE OPERAÇÕES INTELIGENTE IBM	119
7	CONCLUSÃO	122
7.1	TRABALHO PUBLICADO DESTA PESQUISA: SIMULAÇÃO DE MICRODRENAGEM EM CIDADES CONSCIENTES – XX CONGRESSO BRASILEIRO DE AUTOMÁTICA.....	127
7.2	TRABALHO PUBLICADO DESTA PESQUISA: <i>A CLOUD-BASED ARCHITECTURE FOR THE INTERNET OF THINGS TARGETING INDUSTRIAL ENGINE REMOTE – SENSORS TO CLOUD ARCHITECTURES WORKSHOP – 21° IEEE SYMPOSIUM ON HIGH PERFORMANCE COMPUTER ARCHITECTURE (HPCA - 2015)</i>	129
	REFERÊNCIAS.....	131
	APÊNDICE	153
A.	O PADRÃO IEEE 802.15.4	153
B.	O PROTOCOLO 6LOWPAN.....	156
C.	MESSAGE QUEUING TELEMETRY TRANSPORT – MQTT.....	159
D.	MESSAGE QUEUING TELEMETRY TRANSPORT FOR WIRELESS SENSOR NETWORKS – MQTT- SN.....	161
E.	COMPUTAÇÃO EM NUVEM.....	164
F.	TABELAS DO BANCO DE DADOS	167
G.	DEFINIÇÕES DAS INTERFACES.....	171
H.	MODELOS DE ALGORITMOS DE CLASSIFICAÇÃO UTILIZADOS	185
I.	QUESTIONÁRIO DE PESQUISA DE OPINIÃO SOBRE APLICATIVO DE SENSORIAMENTO PARTICIPATIVO	190

1 INTRODUÇÃO

As cidades estão se mostrando mais complexas e mais importantes na medida em que cresce a população nas áreas urbanas (NAM e PARDO, 2011). De acordo com as Nações Unidas, 2008 marcou o ano em que mais da metade das pessoas já viviam em áreas urbanas, sendo que até 2030 é esperado que esta população alcance 5 bilhões de indivíduos (MARTINE e MARSHALL, 2007). Especialmente impulsionada pela tecnologia de informação (TI) como as redes sociais e as iniciativas de sensoriamento por multidão, denominado em inglês por *crowdsensing* (GANTI, YE e LEI, 2011), os centros urbanos se tornaram locais vibrantes onde os cidadãos estão mais engajados na contribuição de melhorias. Entre os exemplos destas iniciativas *crowdsensing*, estão às tecnologias em aplicativos de dispositivos móveis como o Waze¹, 1746² ou o Rota Acessível³ (KOCK, CARDONHA, *et al.*, 2013), que permitem ao cidadão pró-ativamente participar no aprimoramento dos serviços nas cidades, a partir de informações coletadas por eles.

Outro viés para contribuição no aprimoramento dos serviços prestados nas cidades é o monitoramento automático por sensores. As tecnologias de rede de sensores sem fio (RSSF), do termo em inglês *wireless sensor networks* (AKYILDIZ, SU, *et al.*, 2002), direcionada ao transporte (TUBAISHAT, ZHUANG, *et al.*, 2009), distribuição de água (STOIANOV, NACHMAN, *et al.*, 2007), infraestrutura (KIM, PAKZAD, *et al.*, 2007) e meio ambiente (MURTY, G., *et al.*, 2008), podem proporcionar a estes serviços mecanismos de controle mais eficientes. Este tipo de rede é composto por dispositivos distribuídos que cooperativamente trabalham para a monitoração de fenômenos como temperatura, pressão, umidade, luminosidade, presença de certos gases, etc. (YICK, MUKHERJEE e GHOSAL, 2008).

Na direção de cidades eficientes há diferentes estudos que apontam o conceito de cidades inteligentes, a qual a definição mais abrangentemente aceita refere-se justamente em melhorar a eficiência dos serviços e também estimular o desenvolvimento sustentável das cidades, com a integração de fatores tecnológicos,

¹ www.waze.com/pt-BR

² <http://www.1746.rio.gov.br/>

³ <http://rotaacessivel.com/>

humanos e institucionais (NAM e PARDO, 2011). A figura 1 representa a integração destes 3 fatores:

Figura 1: Componentes fundamentais de cidades inteligentes

Fonte: Adaptado de NAPHADE, et al, 2011

No entanto, a fim de avançar efetivamente para tais ambientes urbanos inteligentes, os cidadãos devem ser motivados a este esforço (SANCHEZ, GALACHE, *et al.*, 2011), como também os recursos utilizados nas cidades devem ser maximizados. O *middleware* é um arcabouço de *software* exatamente direcionado para maximização de recursos e a integração de sistemas heterogêneos (TANENBAUM e VAN STEEN, 2007), o que neste contexto são representados pelos sensores que monitoram os serviços da cidade e as informações que os cidadãos desejam compartilhar. O *middleware* está localizado entre o sistema operacional e a aplicação, cuja principal função é fornecer por meio de interfaces o reuso de funcionalidades, facilitando o trabalho no desenvolvimento de sistemas distribuídos eficientes (COULOURIS, DOLIMORE, *et al.*, 2012).

Nota-se que tipicamente os sistemas voltados a *crowdsensing* (KHAN, XIANG, *et al.*, 2013) e a rede de sensores (YICK, MUKHERJEE e GHOSAL, 2008) atuam

separadamente. A integração e correlação destas fontes de dados em um sistema centralizado propício para tratamento de dados heterogêneos como um *middleware*, podem gerar conhecimento e suporte a coordenação de ações em prol da eficiência dos serviços nas cidades. Por exemplo, entre os desafios encontrados em aplicações *crowdsensing* está a própria coordenação desse sistema (STEVENSON e D'HONDT, 2010), uma vez que a coleta de dados excessivamente espaçados geograficamente ou temporalmente, pode ser um fator negativo para agregação dos dados (GANTI, YE e LEI, 2011). Neste caso, um *middleware* que combine dados gerados por sensores pode proporcionar, através de módulos analíticos, indicações ao cidadão colaborador de pontos que prioritariamente precisam ter informações coletadas.

Esta pesquisa é voltada a um *middleware* direcionado à gestão de cidades para coleta, integração e interpretação dos dados de sensores, pertencentes aos serviços disponíveis da própria cidade, junto com os dados do sensoriamento compartilhado altruisticamente pelos cidadãos. Presume-se que este sistema poderá realizar a interpretação de séries históricas, alertas aos munícipes e apoio a manutenção dos serviços da cidade, por exemplo, com o suporte a tomada de decisão aos gestores públicos.

1.1 Motivação e objetivos

O compartilhamento de recursos por diversos dispositivos heterogêneos é uma característica típica de sistemas distribuídos (COULOURIS, DOLIMORE, *et al.*, 2012). Neste trabalho será adotada a utilização de um *middleware* como a estratégia para coleta, integração e interpretação dos dados de um sistema distribuído direcionado a cidades. Entre os desafios inicialmente encontrados neste sistema estão:

- Padronização dos dados provenientes de duas fontes distintas, neste caso representado por:

- i) **Sensores** pertencentes aos serviços da cidade;
- ii) **Sensoriamento colaborativo.**

- Infraestrutura de *hardware* e *software* que comporte o volume de dados, conexões e transações compatíveis com possivelmente milhões de dispositivos que podem estar presentes nesta rede, representados aqui pelos ativos das cidades (ex.: postes de iluminação, semáforo de trânsito, bocas de lobo, etc.) e também pelos próprios cidadãos.

- Modelos analíticos que correlacionem informações das fontes distintas de dados para gerar conhecimento, alertas e confirmações, auxiliando assim os gestores públicos na tomada de decisão.

- Algoritmos e estratégias de diminuição de dados expirados e redundantes gerados pelas fontes de dados.

A pesquisa será desenvolvida ao redor dos seguintes objetivos e hipóteses:

Objetivo da pesquisa. Entendimento de tecnologias e métodos ubíquos para a coleta, integração e interpretação de dados dos serviços prestados nas cidades.

Hipótese. Uma infraestrutura de coleta e integração de dados de sensores e *crowdsensing* para os serviços das cidades, em um *middleware* com módulos analíticos capazes de interpretar integradamente fontes de informação, permitirá aos agentes públicos melhorarem a monitoração dos serviços e planejar ações administrativas com prioridade.

Desta maneira, as perguntas de pesquisas endereçadas são:

“Qual a arquitetura de *middleware* para dados de sensores e de *crowdsensing* para equipar cidades, permitindo o monitoramento e análise de eventos nesses ambientes? ”

“Como coletar dados, prover pré-processamento (filtragem) e análise a sistemas de gerenciamento dos serviços das cidades? ”

“Quais são os aspectos para a integração de informações em ambientes urbanos envolvendo pessoas e os serviços das cidades? ”

A pesquisa será desenvolvida sobre os conceitos de:

- i) Rede de sensores;
- ii) *Crowdsensing*;
- iii) Sistemas distribuídos, com foco em *middleware*. Essas são ferramentas que provêm métodos e tecnologias para a coleta, integração e análise de dados das fontes de informação propostas.

Esse trabalho será desenvolvido dentro do escopo do projeto IBM *Smarter Cities*⁴, parte do programa IBM *Smarter Planet*⁵. Nesse programa, são estudadas formas de instrumentar, interconectar e implantar inteligências nos elementos ao nosso redor, tornando dados em informação inteligente para a tomada de decisão. A IBM está explorando novos tipos de soluções para gestão de cidades que analisa e sincroniza os esforços entre os diversos setores e agências governamentais.

1.2 Organização do trabalho

As redes de sensores (abordadas no capítulo 2), os sistemas *crowdsensing* (abordados no capítulo 3) e os sistemas distribuídos (abordados no capítulo 4) são os fundamentos utilizados para a construção do trabalho. Inicialmente em cada um destes capítulos é contextualizado a escolha do conceito e em seguida, são abordadas as definições, características e desafios referente ao tema. No final destes capítulos são descritas algumas considerações que enfatizam as principais ideias destes conceitos adotadas na pesquisa.

O desenvolvimento do *middleware* é realizado no capítulo 5, com a proposta da arquitetura, definição das funções e um cenário para a prova de conceito deste sistema. Também no mesmo capítulo são apresentados os resultados alcançados na pesquisa. No capítulo 6 são discutidos os trabalhos relacionados a esta dissertação. Por fim as conclusões desta pesquisa e contribuições de publicações já alcançadas por este trabalho, são apresentadas no capítulo 7.

⁴ http://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/

⁵ <http://www.ibm.com/smarterplanet/us/en/overview/ideas/>

2 REDE DE SENSORES

As redes de sensores surgiram com a motivação de defesa militar na década de 1970, pelo o programa de redes de sensores distribuídos (DSN, *Distributed Sensor Networks*) da agência americana DARPA (*Defense Advanced Research Project Agency*) (CHONG e KUMAR, 2003). Em meados de 2000, com avanço dos sistemas micro eletromecânicos (MEMS, *Micro-Electromechanical Systems*), de tecnologias de comunicação sem fio e da eletrônica digital, habilitaram o desenvolvimento de dispositivos sensores multifuncionais minúsculos, de baixo custo, consumo energético e alcance de comunicação (AKYILDIZ, SU, *et al.*, 2002). A ideia é que o esforço colaborativo de diversos dispositivos sensores, organizados em uma rede sem fio, monitore um fenômeno em uma região de interesse (AKYILDIZ, SU, *et al.*, 2002).

Estas redes de sensores podem ser aplicadas no monitoramento e rastreamento em muitos contextos diferentes (YICK, MUKHERJEE e GHOSAL, 2008). Atualmente verifica-se a utilização desta tecnologia em diversos serviços das cidades como no transporte (TUBAISHAT, ZHUANG, *et al.*, 2009), distribuição de água (STOIANOV, NACHMAN, *et al.*, 2007), infraestrutura (KIM, PAKZAD, *et al.*, 2007), meio ambiente e meteorologia (MURTY, G., *et al.*, 2008), proporcionando a estes serviços mecanismos de controle mais eficientes.

Neste capítulo são realizadas as revisões de conceitos, características de funcionamento, desafios, áreas de aplicações e arquiteturas das redes de sensores.

2.1 Introdução

Define-se rede de sensores sem fio (RSSF), ou pelo acrônimo da língua inglesa *wireless sensor network* (WSN), como um conjunto de nós sensores que se comunicam livremente sem fio em pequenas distâncias para um esforço colaborativo na interpretação de um fenômeno (AKYILDIZ, SU, *et al.*, 2002). Também segundo os autores, as RSSF possibilitam a coleta de dados de forma distribuída em ambientes onde o uso de fios é inviável.

As RSSF são projetadas para aplicações específicas e consideram as diversas limitações dos nós, por exemplo, energia, processamento, armazenamento, alcance e largura de banda de comunicação (YICK, MUKHERJEE e GHOSAL, 2008). Segundo os mesmos autores são divididas nos seguintes tipos: **subaquática**, **subterrânea**, **móvel**, **multimídia** e **terrestre**.

Subaquática. As RSSF subaquáticas são direcionadas a aplicações como oceanografia, exploração no mar, navegação assistida e vigilância subaquática. Os nós são instalados debaixo d'água e têm importante restrição de banda de comunicação e atraso de propagação, uma vez que neste ambiente a comunicação é estabelecida por ondas acústicas. Uma característica deste tipo de RSSF é a mobilidade típica por conta das ondas marinhas. (AKYILDIZ, POMPILI e MELODIA, 2005).

Subterrânea. Os nós são enterrados para o sensoriamento do solo, de rochas, ou outros minerais, em aplicações como agricultura e monitoramento de encostas. Este tipo de ambiente impõe alta atenuação do sinal rádio dos nós e alto custo de manutenção devido à dificuldade ao acesso dos sensores (AKYILDIZ e STUNTEBECK, 2006).

Móvel. As RSSF com características predominantes de mobilidade requerem a constante auto-organização dos nós, uma vez que as rotas de comunicação são dinamicamente alteradas. As aplicações para rastreamento e localização de alvos são exemplos comuns para este tipo de RSSF (AL-KARAKI e KAMAL, 2004).

Multimídia. São compostas predominantemente por câmeras e microfones dispostos de modo a rastrear e monitorar uma determinada região. Os nós além de compartilharem entre si os dados coletados podem mutuamente fazer operações de compressão e correlação. O fluxo de dados cadenciado, típico de conteúdo multimídia, requer maior largura de banda de comunicação do que os demais tipos de RSSF (AKYILDIZ, MELODIA e CHOWDHURY, 2007).

Terrestre. As redes terrestres são o tipo de RSSF que têm menos restrições. Podem ser beneficiadas por outras fontes de alimentação de energia como a solar, de modo a não depender exclusivamente das baterias. Também podem ser beneficiadas

pelos diversos mecanismos de otimização de consumo de energia como a transmissão a baixa distância, agregação de dados, eliminação de dados redundantes e otimização dos saltos da rota (ANASTASI, CONTI, *et al.*, 2009).

2.2 Características das redes de sensores

Os componentes de uma rede sem fio são: o **nó sensor**, o **nó sorvedouro** e o **servidor**.

Nó sensor. É constituído de um ou mais elementos para sensoramento do fenômeno e também de componentes para processamento de dados, armazenamento e comunicação (SOHRABY, MINOLI e ZNATI, 2007). Pode ser ou não distribuído aleatoriamente no ambiente de monitoração. Normalmente a rota de comunicação até o sorvedouro é constituída passando pelos nós vizinhos, os chamados “saltos”.

Nó sorvedouro. É o ponto da RSSF central pelo qual os dados dos demais nós sensores da rede serão encaminhados (SOHRABY, MINOLI e ZNATI, 2007). Geralmente possui recursos energéticos e de processamento maiores que os demais nós da rede. É no sorvedouro que ocorre a conexão com a rede externa (LAN, GPRS, 3G, 4G, etc.).

Servidor. É a estação que coordena as atividades globais da RSSF, por exemplo, faz solicitações de tarefas para os nós sensores ou manipula os dados recebidos pelos sensores.

De uma maneira geral os nós sensores são todos iguais, exceto o nó sorvedouro que possui uma conexão adicional para a rede externa. Porém em algumas situações é preferível que alguns nós tenham funções especializadas para realizar agrupamentos de sensores (ABBASI e YOUNIS, 2007). Por exemplo, em RSSF densas, o consumo energético dos nós pode ser diminuído abaixando o alcance do rádio, porém os nós de agrupamentos ainda manterão a potência do rádio alta para alcançar outros nós de agrupamento ou o próprio sorvedouro, construindo hierarquias na rede (ABBASI e YOUNIS, 2007). A figura 2 mostra um exemplo de uma RSSF típica:

Figura 2: Arranjo típico de RSSF

Fonte: Adaptado de SOHRABY, MINOLI e ZNATI, 2007

Um típico nó sensor é composto de **unidade de processamento**, **unidade de energia**, **transceptor** e **unidade de sensoriamento** podendo ainda conter opcionalmente **módulos para localização**, **mobilidade** e **fonte de alimentação**. A figura 3 apresenta a composição de um nó sensor.

Unidade de processamento. É o conjunto composto pelo processador e memória, onde são realizados o tratamento e o armazenamento de dados local e as funções de controle do nó. Além do processamento para o controle do nó, a amostragem e a precisão do fenômeno analisado são fatores crítico do processamento. Por exemplo, aplicações baseadas na coleta de variáveis como temperatura e umidade normalmente requerem menos processamento de amostras por segundo do que aplicações que coletam sinais de vibração ou áudio.

Unidade de energia: Normalmente é composta pela bateria e eventualmente por conversores DC para DC para adaptação de tensão dos demais módulos do nó.

Transceptor. É o grande consumidor da energia do nó, composto pelo rádio e a antena que o conecta na RSSF. Tipicamente opera nas frequências de ISM (*Industrial, Scientific and Medical*).

Figura 3: Composição de um nó sensor

Fonte: Adaptado de SOHRABY, MINOLI e ZNATI, 2007

Unidade de sensoriamento. É formado pelo transdutor sensor e o conversor de sinal analógico para digital - ADC (*Analog Digital Converter*). Opcionalmente pode existir mais unidades de sensoriamento.

Módulo de localização. Frequentemente as aplicações de RSSF necessitam o conhecimento das posições dos nós, principalmente quando eles são móveis. Um dos sensores mais usados para detectar a posição é o GPS, com o uso limitado a locais abertos.

Módulo de mobilidade. Existem nós que possuem recursos próprios de mobilidade, chamados de nós ativos, os quais são comumente utilizados para aplicações de resgate e vigilância (SOHRABY, MINOLI e ZNATI, 2007).

Fonte de alimentação. Este módulo é adicional e tem o objetivo de prover autossuficiência ou minimizar as restrições energéticas, a partir da coleta de energia

do ambiente onde o nó está localizado. Entre as fontes mais populares de energia reutilizada estão a solar e a piezelétrica (GILBERT e BALOUCI, 2008).

Embora muitas das técnicas de economia de energia são utilizadas no *software* embarcado dos nós (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000), também em RSSF são estudados no *hardware* métodos para construção de rádios customizados e deste modo, mais eficientes energeticamente (HEMPSTEAD, LYONS, *et al.*, 2008). É possível que com o avanço na pesquisa em técnicas para abordagem holística entre a aplicação, arquitetura e fabricação de circuitos integrados, brevemente serão populares hardwares para RSSF compatível com autonomia de décadas de uso (HEMPSTEAD, LYONS, *et al.*, 2008).

2.3 Desafios das redes de sensores

Os desafios de RSSF para AKYILDIZ et al. (2002) são o **consumo de energia, tolerância a falhas, escalabilidade, ambiente de operação, topologia, restrição de hardware, meio de transmissão e custo de produção.**

Consumo de energia. O consumo de energia é uma importante limitação de RSSF (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000). As características da aplicação como a quantidade de amostragem do fenômeno, o processamento de dados necessário e principalmente a frequência que estes dados precisam ser atualizados no sistema central, são impactadas pela quantidade de energia que os nós da rede dispõem, em especial, as aplicações com um confinamento energético por bateria (HEINZELMAN, CHANDRAKASAN e BALAKRISHNAN, 2000). Dependendo da aplicação, como as subaquáticas e subterrâneas, pode ser inviável ou até mesmo impossível substituir a bateria (YICK, MUKHERJEE e GHOSAL, 2008). Além disso, o mau funcionamento de alguns nós pode acarretar em mudanças das rotas da RSSF ou até mesmo comprometer a área de cobertura da rede. A otimização do consumo energético não se limita apenas a aumentar o tempo de vida de um único nó sensor, mas prolongar o tempo de funcionamento de toda a rede (AL-KARAKI e KAMAL, 2004).

Tolerância a falhas. Os nós podem se tornar temporariamente ou definitivamente inoperantes por diversos motivos, tais como, interferências do ambiente, danos físicos, falta de energia ou congestionamento da rede, porém estas falhas não devem afetar significativamente a tarefa global da RSSF (AKYILDIZ, SU, *et al.*, 2002). Os protocolos devem ser construídos de modo a alcançar os níveis necessários de tolerância a falhas da aplicação (PARADIS e HAN, 2007).

Escalabilidade. As necessidades das aplicações de RSSF podem exigir a presença milhares de nós, tanto para aumentar a área de cobertura ou a resiliência a falhas (AKYILDIZ, SU, *et al.*, 2002). Porém, a concorrência de acesso ao meio e a distribuição das rotas dos diversos nós podem comprometer a eficiência da rede conforme é aumentado a quantidade de sensores. Desse modo, os esforços de tornar as RSSF expansíveis se concentram nos protocolos da camada de enlace e de rede (YE, HEIDEMANN e ESTRIN, 2002) e (AL-KARAKI e KAMAL, 2004).

Ambiente de operação. Apenas observando pelo ponto de vista do local onde é instalada uma RSSF, podem surgir grandes desafios a impactar significativamente o desempenho da rede. Alguns destes lugares podem ser até inóspitos ao ser humano, por exemplo, dentro de um furacão, no fundo oceano, em campo de batalhas, no interior de uma caverna, enterrados em uma plantação, no interior de máquinas, conectados a animais, em locais quimicamente ou biologicamente contaminados (AKYILDIZ, SU, *et al.*, 2002). O ambiente de operação irá impor diversas restrições a RSSF como o tamanho físico do nó, a densidade da rede, a velocidade de propagação, a atenuação do sinal de comunicação e a latência (YICK, MUKHERJEE e GHOSAL, 2008).

Topologia. Devido à enorme quantidade de nós que uma RSSF pode assumir pela forma arbitrária que os nós são instalados ou ainda pela inerente susceptibilidade a falhas destes, a tarefa de criar e manter a topologia da rede é um grande desafio. A manutenção da topologia é também dificultada pelo fato que novos nós podem ser incluídos na rede depois de formada ou até mesmo pela mobilidade deles (AKYILDIZ, SU, *et al.*, 2002). Diversas são as formas empregadas para a manutenção da topologia, como usar as técnicas de localização ou modificar o ciclo de sono de alguns nós (LI e YANG, 2006). Esta é uma tarefa contínua a fim de evitar falhas na cobertura da rede (LI e YANG, 2006).

Restrições de hardware. As dimensões do hardware, precisão de medição, capacidade de processamento, comunicação e operação em um meio inóspito, são exemplos de características do hardware ao qual interferem significativamente no desempenho de uma RSSF (AKYILDIZ, SU, *et al.*, 2002) e (YICK, MUKHERJEE e GHOSAL, 2008).

Meio de transmissão. Existem alguns meios de transmissão de RSSF, por exemplo, as ondas sonoras, o infravermelho e as ondas eletromagnéticas. As ondas sonoras são predominantemente usadas para aplicações subaquáticas e o infravermelho pode ser usado em locais com ruídos eletromagnéticos, mas os nós precisam de linha de visada direta. Exceto para aplicações específicas, o meio de transmissão mais utilizado é o de ondas eletromagnéticas na faixa ISM (*Industrial Scientific and Medical*) (AKYILDIZ, SU, *et al.*, 2002).

Custo de produção. Também por conta da quantidade de nós que uma RSSF pode possuir, o valor do dispositivo pode impactar na densidade e na cobertura da rede.

2.4 Aplicações das redes de sensores

Existem diversas aplicações para RSSF. Segundo YICK, MUKHERJEE e GHOSAL (2008) podem ser divididas em monitoração e rastreamento, figura 4.

Algumas destas aplicações de RSSF já foram construídas para monitoramento de serviços de cidades, com experimentos em escala compatível a cidades e fora do ambiente de laboratório, por exemplo, o projeto *SmartSantander* (SANCHEZ, MUNOZ, *et al.*, 2013). Este projeto exhibe uma plataforma aberta em larga escala para testar tecnologias de RSSF conectadas à internet, também chamada de internet das coisas ou *internet of things* (IoT) (ATZORI, IERA e MORABITO, 2010). A infraestrutura do experimento consistiu de 790 nós fixos com sensores de temperatura, intensidade luminosa, ruído, umidade do solo, anemômetro, pressão atmosférica, radiação solar, pluviômetro e monóxido carbônico; 390 nós instalados no asfalto para detecção de presença de carros em pontos de estacionamento da rua e 150 nós móveis instalados em ônibus e táxis para monitorar o trânsito da cidade. Porém um dos projetos

precursores de RSSF direcionados a cidades foi o *CitySense* (MURTY, G., *et al.*, 2008). Neste projeto aproximadamente 100 nós foram espalhados pela cidade de Cambridge, Estados Unidos, em poste de iluminação ou em construções públicas e eram constituídos por sensores de temperatura, umidade do ar, anemômetro, pluviômetro e dióxido de carbono. Os nós não tinham restrições de energia, uma vez que eram diretamente ligados na rede elétrica e possuíam capacidades de processamento e armazenamento. Também eram construídos com duas interfaces 802.11, sendo uma para administração e outra para o plano de monitoração.

Figura 4: Visão geral de aplicações em RSSF

Fonte: Adaptado de YICK, MUKHERJEE e GHOSAL, 2008

Alguns experimentos de RSSF foram desenvolvidos para monitoração da vida animal, uma vez que a presença de pessoas no ambiente pode atrapalhar o padrão de comportamento dos próprios animais de estudo. Um dos projetos precursores consistiu em aplicar uma RSSF com 32 nós estáticos para monitoração de ninhos de aves marítimas na ilha *Great Duck*, no estado de Maine, Estados Unidos

(MAINWARING, POLASTRE, *et al.*, 2002). No projeto os ninhos eram monitorados por nós que continham sensores de temperatura, luminosidade, umidade e pressão barométrica. Os dados eram repassados a um sorvedouro que tinha uma conexão por satélite até um centro de monitoração no continente. Outro projeto dedicado a monitoração da vida de animais silvestres foi o Zebranet, o qual continha 30 nós com sensores de localização na forma de colares instalados em zebras (JUANG, OKI, *et al.*, 2002), na região de Laikipia, Quênia. Neste projeto tanto os nós (zebras) quanto o ponto para a coleta dos dados, o sorvedouro, eram móveis. Os sorvedouros eram instalados em veículos terrestres ou até mesmo em aviões para a coleta dos dados. Os biólogos queriam saber como, porque e quando as zebras migravam, usando sensores GPS (*Global Position System*) para a pesquisa.

Outro segmento de destaque na utilização de redes de sensores é o ambiente industrial. O segmento industrial possui uma grande diversidade de aplicações para RSSF, entre elas, a cooperação com robôs, rastreamento de objetos em tempo real, monitoração de equipamentos e monitoração de ambientes (LOW, WIN e ER, 2005). Um dos projetos precursores neste segmento, com aplicação real, foi o desenvolvimento de uma RSSF para a coleta da assinatura de vibração em equipamentos de purificação de água de uma planta de fabricação de semicondutores (KRISHNAMURTHY, ADLER, *et al.*, 2005). Os autores instalaram 150 acelerômetros em 26 nós sensores durante 4 meses de experimento. Assim como este exemplo, o uso de RSSF para obtenção de métodos preditivos mais eficientes em equipamentos ainda são investigados. Porém, agora incluem formas de integração com serviços de computação em nuvem, adicionando serviços de recomendações, relatórios de falha e modelos de análise do equipamento, mais diversificados e sob demanda (SILVA, AZEVEDO, *et al.*, 2015).

Entre os tipos de aplicações de RSSF, as aplicações militares têm a importância histórica de ter motivado os esforços iniciais de pesquisa (CHONG e KUMAR, 2003). Os fenômenos mais importantes analisados a estas aplicações são as ondas eletromagnéticas, a luz, a pressão e o som. Porém, além do tipo de sensoriamento dos nós as características que definem as capacidades das RSSF para uso militar, são a cobertura da rede e a geração do conhecimento que ela proporciona, a partir da fusão dos dados (DURISIC, TAFAR, *et al.*, 2012). Atualmente, entende-se que não

somente o campo de batalha atende aos interesses do segmento militar, mas também cenários de guerra urbana, circunstâncias de proteção militar, manutenção da paz e atendimento a desastres. Devido à variedade de interesses, é diversificado o uso da tecnologia RSSF para fins militar, por exemplo, o sensoriamento da condição de operação e a navegação de mísseis, a proteção de perímetro, a localização de atiradores de elite em guerras urbanas, o monitoramento dos sinais vitais de soldados, etc. (DURISIC, TAFA, *et al.*, 2012).

A agricultura é outro exemplo de segmento beneficiado pela tecnologia de RSSF. A monitoração do solo, do clima, da presença e disseminação de certas pragas, do crescimento do cultivo e em criação de animais, a monitoração do ambiente de confinamento, do rastreamento do rebanho e até mesmo a gestão do ciclo de vida do animal são estratégias utilizadas para gerenciamento e suporte a decisão no agronegócio (GARCIA, LUNADEI, *et al.*, 2009). Um dos exemplos pioneiros de implantação de RSSF para agricultura foi a instalação de 65 nós sensores para monitorar o cultivo de uvas em uma área de 2 hectares durante 6 meses (BECKWITH, TEIBEL e BOWEN, 2004). Segundo os autores, a rede era relativamente densa com os nós distribuídos entre 10 e 20 metros e enfrentou variações de temperatura entre -30°C a +45°C, com rajadas de ventos de 140km/h. O resultado deste projeto foi a detecção e o entendimento das regiões do cultivo que eram mais afetadas com pontos de geadas ou superaquecimento.

2.5 Arquitetura das redes de sensores

Tipicamente os nós de uma RSSF operam por pilha de protocolos. Este conceito de pilha de protocolos é similar ao empregado em rede de computadores tradicionais a qual a aplicação não interage diretamente com o *hardware* da rede, ao invés disto, usa diversas camadas de *software* com funções definidas até alcançar a camada física, onde está o *hardware* que executará a propagação dos dados pela rede (KUROSE e ROSS, 2010).

A pilha de protocolo em RSSF, figura 5, tem normalmente as seguintes camadas: **aplicação, transporte, rede, enlace e física** as quais farão um esforço colaborativo

para, além de lidar com a tarefa de sensoriamento em si, também resolver desafios de mobilidade e da energia limitada na rede (AKYILDIZ, SU, *et al.*, 2002).

Figura 5: Pilha de protocolos de rede de sensores

Fonte: Adaptado de AKYILDIZ, et al, 2002

Os planos de gerenciamento de tarefa, mobilidade e energia são distribuídos pelas camadas de protocolos de modo que a rede se mantenha em funcionamento pelo máximo tempo possível. Por exemplo, uma tarefa de sensoriamento em uma região específica pode ser concentrada em apenas poucos sensores na região, ao invés de usar toda a rede. Em termos de mobilidade é útil um determinado nó conhecer os vizinhos e o trajeto para calcular quais são as conexões disponíveis naquela rota, ao invés de deliberadamente recriar novas conexões. Um nó pode entrar em modo de baixo consumo de energia, se conhecer o cronograma de comunicação dos nós vizinhos (AKYILDIZ, SU, *et al.*, 2002).

Aplicação. Mesmo considerando que as RSSF são específicas para cada cenário de atuação, otimizações na aplicação podem reduzir o número de transmissões, e portanto, o consumo de energia da rede. Entre as funções desta camada podem estar a agregação de dados, o agrupamento de sensores, as funções de autenticação e localização. Um padrão proprietário popular para RSSF e que também abrange a camada de aplicação é o Zigbee (ZIGBEE ALLIANCE, 2014). O MQTT-SN (*Message Queuing Telemetry Transport for Wireless Sensor Networks*) é um protocolo centrado a dados que pertence a esta camada e será detalhado no apêndice D.

Transporte. Os protocolos de transporte para RSSF são desenvolvidos considerando a dinâmica da topologia, a diversidade da aplicação, as características do tráfego e os recursos disponíveis como o armazenamento de dados e a energia disponível (LIU, TANG e LIU, 2012). Porém, as duas restrições mais significativas são a economia de energia e a distribuição heterogênea do tráfego na rede, uma vez que pontos mais próximos das bordas da rede estão suscetíveis a um tráfego menor do que pontos mais centrais (WANG, SOHRABY, *et al.*, 2006). Para uma RSSF o protocolo de transporte precisa prover eficiência energética, confiabilidade com flexibilidade e quando necessário, a qualidade de serviço (WANG, SOHRABY, *et al.*, 2006).

Rede. A função da camada de rede em RSSF é encontrar caminhos entre a origem e o destino da mensagem, considerando a eficiência energética, mobilidade ou a qualidade de serviço (AKKAYA e YOUNIS, 2005). Esta função é classicamente chamada de roteamento (KUROSE e ROSS, 2010), podendo o pacote de dados percorrer diversos nós, através de saltos, até alcançar o destino. O roteamento pode considerar diversos propósitos conforme a necessidade da aplicação, entre eles estão a tolerância a falhas, agregação de dados, basear-se em consultas e QoS (AL-KARAKI e KAMAL, 2004). Com relação à distribuição dos nós na rede, existem protocolos de roteamento que criam hierarquias, outros que se baseiam na localização física dos nós e ainda os protocolos que simplesmente empregam as técnicas de saltos até alcançar o destino (AL-KARAKI e KAMAL, 2004).

Enlace. Uma das características marcantes das RSSF é a densidade dos nós na rede, a qual traz implicações na concorrência de acesso ao meio sem fio (YE, HEIDEMANN e ESTRIN, 2002). A função do protocolo de enlace, conhecido também como protocolo MAC (*Media Access Control*), é justamente prover métodos eficientes para os nós acessarem o meio com a menor utilização de energia e ainda adaptável a dinâmica da rede, seja pela mobilidade, falha dos nós ou escalabilidade (HUANG, XIAO, *et al.*, 2013). As características de vazão e latência podem ser consideradas importantes também em algumas aplicações que necessitem de QoS, mas em geral são deixadas como métricas secundárias (DEMIRKOL, ERSOY e ALAGOZ, 2006). Entre as técnicas tratadas nesta camada a fim de aperfeiçoar o consumo energético estão, por exemplo, evitar a colisão de pacotes, otimizar os pacotes de controle e

manter desligado o transceptor pelo período que previamente se conhece que não haverá comunicação entre os vizinhos (DEMIRKOL, ERSOY e ALAGOZ, 2006).

Física: Conforme discutido na seção 2.4, as RSSF podem operar em diversos meios de transmissão, os quais interferem diretamente na camada de rede física. Considerando que este trabalho tem o foco em aplicações direcionada a centros urbanos, serão considerados fundamentalmente o meio de transmissão por onda eletromagnética. Embora alguns trabalhos de pesquisa abordam o tema da utilização de faixas de rádios licenciadas, por meio de técnicas que oportunamente se beneficiam de intervalos de ociosidade destes canais (AKAN, KARLI e ERGUL, 2009), predominantemente as RSSF são usadas na faixa de ISM. De modo similar as pesquisas com abordagem holística entre a aplicação, arquitetura e fabricação de circuitos integrados, para a construção de hardware de RSSF compatível com autonomia de décadas de operação (HEMPSTEAD, LYONS, *et al.*, 2008), as técnicas que consideram abordagens inter-relacionadas entre as camadas de física e enlace são as mais promissoras. A mais popular seguindo nesta direção é o padrão IEEE 802.15.4 (IEEE, 2006) que foi desenvolvido, confrontando os desafios típicos de comunicação sem fio, entre eles o ruído e interferência eletromagnética, com técnicas na camada de controle de acesso ao meio, por exemplo, o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) e controle de erro de dados CRC (*Cyclic Redundancy Check*) típicas da camada de enlace. O IEEE 802.15.4 será detalhado no apêndice A.

2.6 Considerações finais

Após o surgimento na década de 1970 (CHONG e KUMAR, 2003) até os dias de hoje, as RSSF foram continuamente evoluindo. Esta evolução atuou nos mais diversos desafios citados neste capítulo, por exemplo, em termos de métodos para economia de energia, tolerância a falhas, escalabilidade, custo de produção, etc. Atualmente nota-se que o foco de pesquisa das RSSF está direcionado para conectividade dos diversos itens do cotidiano com a internet, com o objetivo de usufruir dos benefícios destes dispositivos de modo ubíquo e pervasivo, com o nome de internet das coisas (ATZORI, IERA e MORABITO, 2010).

Neste capítulo foi observado que pela diversidade de aplicações, a estratégia de utilização de sensores tem ganhos significativos na interpretação dos mais diversos fenômenos aplicados, em especial para quantiza-los. Especificamente para avaliação dos diversos serviços nas cidades como o transporte (TUBAISHAT, ZHUANG, *et al.*, 2009), a distribuição de água (STOIANOV, NACHMAN, *et al.*, 2007), a infraestrutura (KIM, PAKZAD, *et al.*, 2007), o meio ambiente e meteorologia (MURTY, G., *et al.*, 2008) ou até mesclado em múltiplos serviços (SANCHEZ, MUNOZ, *et al.*, 2013) os sensores podem proporcionar mecanismos de controle mais eficientes.

A tecnologia de RSSF discutida neste capítulo, composta pelas três camadas fundamentais: nós sensores, sorvedouro e servidor e também pela arquitetura de camada da rede: aplicação, transporte, rede, enlace e física, serão usadas neste trabalho para a obtenção da parte correspondente ao serviço de monitoração automático direcionado a cidades proposto. Especificamente para o serviço da camada física e enlace será utilizado o padrão IEEE 802.15.4 que foi desenvolvido justamente para aplicações com baixa transferência de informações e consumo de energia (IEEE, 2006). Ele será detalhado no apêndice, junto com o 6LoWPAN que será a tecnologia adotada para a camada de rede. O 6LoWPAN é o protocolo que habilita o endereçamento de pacotes IPv6 em redes de baixo consumo de energia (MULIGAN e GROUP, 2007) compatível com a quantidade de ativos das cidades. Para a camada de aplicação será empregado o paradigma de comunicação de sistema distribuído, publicar e assinar baseado em tópicos, o qual será discutido no capítulo 4.

3 CROWDSENSING

As diversas iniciativas de utilizar a multidão de usuários da internet em prol de receber informações mais abrangentes, mais baratas e em muitos casos, mais exatas do que especialistas individuais locais, culminou no surgimento do conceito de *crowdsourcing* (HOWE, 2006). Este conceito parte do pressuposto que o todo seja capaz de se autocorrigir e assim os resultados serão globalmente mais confiáveis do que a resposta de um indivíduo ou de um grupo pequeno local (BRABHAM, 2008). Especializações desta estratégia com designações próprias surgiram posteriormente como o *crowdfunding*: financiamento de projetos, o *crowdvoting*: votação por júri popular e o *crowdcreation*: criação de bens intelectuais (HOWE, 2008).

A especialização estudada neste trabalho é aquela destinada ao sensoriamento móvel efetuado por um grupo indefinido de pessoas, definido como *mobile crowdsensing* (GANTI, YE e LEI, 2011). Graças a pré-requisitos cada vez mais comuns: um indivíduo com *smartphone* e conectividade a sistemas remotos, não necessariamente *on-line*, a facilidade de disseminação do projeto, o espalhamento dos participantes e o baixo custo de implantação e operação, estas aplicações vem tornando-se atualmente populares, por exemplo, o Waze⁶ já conta com mais de 50 milhões de usuários⁷. No Brasil, projetos como o 1746⁸, com apoio da prefeitura municipal do Rio de Janeiro, ou o RotaAcessível⁹ com a participação de organizações não governamentais como a AACD¹⁰, tem conseguido a adoção de milhares de pessoas.

No momento, verifica-se a utilização desta tecnologia em diversos serviços das cidades, tais como o monitoramento da poluição do ar (PAULOS, HONICKY e GOODMAN, 2007), o relato de ruído excessivo localizado (RANA, CHOU, *et al.*, 2010; PAULOS, HONICKY e GOODMAN, 2007), a indicação da velocidade atual da via (HERRERA, WORK, *et al.*, 2010), a localização de buracos em via pública (MOHAN, PADMANABHAN e RAMJEE, 2008), etc., proporcionando a estes serviços mecanismos adicionais para suporte a tomada de decisão dos gestores públicos.

⁶ <https://www.waze.com/pt-BR>

⁷ <http://en.wikipedia.org/wiki/Waze>

⁸ <http://www.rio.rj.gov.br/web/guest/exibeconteudo?id=4658555>

⁹ <http://rotaacessivel.com/>

¹⁰ <http://aacd.org.br/>

Neste capítulo são realizadas as revisões de conceitos, paradigmas, arquitetura, desafios e também discutido as limitações de recursos e mecanismos de agregação dos sistemas *mobile crowdsensing*.

3.1 Introdução

O termo em inglês *crowdsourcing* criado por Jeff Howe (HOWE, 2006) resulta das palavras **crowd** - multidão e **outsourcing** – subcontratação. Atualmente a definição mais integrada é: "*Crowdsourcing* é um tipo de atividade *on-line* participativa em que um indivíduo, uma instituição, uma organização sem fins lucrativos ou uma empresa propõe para um grupo de indivíduos com conhecimento variado, heterogêneo e numeroso, através de um convite aberto e flexível, o compromisso voluntário da realização de uma tarefa. O compromisso da tarefa de complexidade e modularidade variável, na qual, a multidão participa com seu trabalho, dinheiro, conhecimento e / ou experiência, sempre implica em benefício mútuo. O usuário receberá a satisfação de um determinado tipo de necessidade, seja ela, econômica, de reconhecimento social, de autoestima, ou o desenvolvimento de competências individuais, enquanto o provedor da atividade vai obter e utilizar a seu favor o que o usuário trouxe para o empreendimento, cuja forma depende do tipo de atividade definida" (AROLAS e GUEVARA, 2012).

A especialização de *crowdsourcing* foco deste trabalho será a de *mobile crowdsensing*, definida por GANTI, YE e LEI (2011): "*Mobile crowdsensing* é uma categoria de aplicações a qual indivíduos com sensores ou dispositivos computacionais, como os *smartphones*, coletivamente compartilham informações afim de medir ou mapear fenômenos de comum interesse" (GANTI, YE e LEI, 2011). Também, segundo os mesmos autores, *crowdsensing* está dividido em duas categorias de utilização: o **sensoriamento social** e o **comunitário**. Estas categorias independentemente podem se encaixar em paradigmas de funcionamento **participativo**, com ação proativa do próprio participante ou **oportunista**, uma vez que os dispositivos móveis podem gerar automaticamente informações sobre o ambiente que estão inseridos.

3.2 Paradigmas e escala de sensoriamento

Considerando o envolvimento do participante, existem dois paradigmas de funcionamento de *crowdsensing*: o **participativo** e o **oportunista** (GANTI, YE e LEI, 2011).

Participativo. Neste paradigma o participante está diretamente envolvido, coletando e compartilhando informações sobre o local que está inserido (BURKE, ESTRIN, *et al.*, 2006). Áudio, vídeo, imagens, coordenadas de GPS e textos sobre o local, são exemplos de dados que podem ser compartilhados pelos participantes. Uma vantagem poderosa do sensoriamento participativo é que as pessoas conseguem discernir contextos complexos e compartilhar informações de alta relevância (BURKE, ESTRIN, *et al.*, 2006). Porém é preciso de métodos de engajamento para continuidade de compartilhamento de informações por estes indivíduos (GANTI, YE e LEI, 2011). Uma aplicação que remete a este tipo de paradigma é, por exemplo, o *Citizen Sensing* (KOCK, CARDONHA, *et al.*, 2013) a qual um cidadão relata problemas na região da cidade como a necessidade de poda de árvore, problemas de iluminação pública, pontos de alagamentos, etc.

Oportunista. No sensoriamento oportunista o envolvimento do participante é mínimo, sendo considerado mais escalável do que o paradigma participativo (LANE, EISENMAN, *et al.*, 2008). Por outro lado, questões de privacidade podem ser atingidas fora do contexto do sensoriamento, por exemplo, a divulgação indireta da localização do participante sem que o mesmo tome conhecimento (LANE, EISENMAN, *et al.*, 2008). Uma aplicação típica deste paradigma é o monitoramento de trânsito em vias urbanas, onde o cidadão apenas ativa aplicação e então a aplicação se encarrega de compartilhar com os demais participantes do sistema qual a velocidade atual da via (HERRERA, WORK, *et al.*, 2010).

Já pelo enfoque de utilização dos dados, as aplicações *crowdsensing* podem estar divididas em duas escalas de sensoriamento: **sociais** ou **comunitárias** (GANTI, YE e LEI, 2011).

Sociais. Neste tipo de escala de sensoriamento, as informações são compartilhadas dentro de uma rede social. Devido à relativa confiança inerente entre os indivíduos do grupo de uma rede social, os cuidados com a privacidade ou a veracidade dos dados captados, pode ser reduzido (LANE, MILUZZO, *et al.*, 2010). Um dos projetos pioneiros desta categoria é o *CenceMe*, onde os participantes compartilhavam no *Facebook*¹¹ os resultados de suas atividades físicas, por exemplo, a distância percorrida em uma caminhada ou situações do cotidiano, como fotos tiradas aleatoriamente e fora do controle do participante (MILUZZO, LANE, *et al.*, 2008). Estes resultados captados por sensores, tais como, o acelerômetro, o GPS, o microfone e a câmera no *smartphone* do participante, segundo os autores, despertavam a curiosidade e o engajamento em atividades físicas dos demais usuários da rede social.

Comunitárias. As aplicações de sensoriamento comunitário têm o objetivo de compartilhar informações para o bem público. O baixo comprometimento e a privacidade são premissas para que os participantes pratiquem o altruísmo ou incentivem a melhoria dos serviços públicos pelo compartilhamento de dados coletados por eles (LANE, MILUZZO, *et al.*, 2010). Entre os exemplos de aplicações típicas estão o monitoramento da poluição do ar (PAULOS, HONICKY e GOODMAN, 2007), o relato de ruído excessivo localizado (RANA, CHOU, *et al.*, 2010; PAULOS, HONICKY e GOODMAN, 2007), a indicação da velocidade atual ou relatos da posição de buracos em vias pública (MOHAN, PADMANABHAN e RAMJEE, 2008), etc.

Existem alguns trabalhos que envolvem múltiplos aspectos de paradigmas e escala de sensoriamento, por exemplo, o projeto *Bikenet* (EISENMAN, MILUZZO, *et al.*, 2009). Neste projeto, os ciclistas oportunamente coletavam dados ambientais e do próprio desempenho da atividade esportiva durante o trajeto. O participante poderia compartilhar no momento do exercício, ou no final do dia, o seu desempenho em uma rede social, ou ainda, divulgar os dados coletados de gás carbônico dos pontos geolocalizados coletados automaticamente durante a rota.

¹¹ www.facebook.com

3.3 Arquitetura de sistemas *crowdsensing*

A arquitetura de sistemas *mobile crowdsensing* descrita por alguns autores GANTI, YE e LEI (2011) e CHON, LANE e LI, et al. (2012) são baseadas na arquitetura de sensoriamento participativo, inicialmente descrita por BURKE, et al. (2006) e também apresentada por CHRISTIN, et al. (2011), figura 6.

Figura 6: Arquitetura de aplicações de sensoriamento participativo

Fonte: Adaptado de CHRISTIN, et al, 2011

Nesta arquitetura as partes interessadas são compostas por três grupos: os **participantes**, o **administrador do sistema** e os **usuários finais** (CHRISTIN, REINHARDT, et al., 2011).

Participantes. É a multidão de pessoas que colabora com os dados coletados. Podem ser concomitantemente consumidores do resultado final pelos dados agregados do sistema.

Administrador do sistema. São os responsáveis pela coordenação do sistema. Geralmente são grupos de pesquisa ou membros de uma organização que iniciam o

sistema, por exemplo, disseminando em lojas virtuais de aplicativos e que também efetuam o gerenciamento, atualização e manutenção do servidor de aplicação (CHRISTIN, REINHARDT, *et al.*, 2011).

Usuários finais. São as pessoas que utilizaram o resultado do sistema para os diversos fins destas aplicações. Podem incluir pesquisadores, gestores públicos, empresas particulares e os próprios participantes.

Os **dispositivos** desempenham o papel da ferramenta elementar para o sensoriamento pelo participante. Possuem funções similares as encontradas em rede de sensores: **sensoriamento**, **processamento**, **armazenamento**, além da geração e transmissão dos **relatórios**.

Sensoriamento. Tipicamente os *smartphones* e *tablets* são os dispositivos alvos das aplicações *crowdsensing*. Logo, as aplicações mais populares são aquelas que usam os sensores normalmente disponíveis destes dispositivos (CHRISTIN, REINHARDT, *et al.*, 2011). Também segundo os mesmos autores, os principais sensores atualmente usados nestas aplicações estão: o GPS, câmera, microfone, acelerômetro e giroscópio. Algumas aplicações usam também sensores externos em conjunto com o dispositivo pessoal do participante, para divulgar dados por *crowdsensing* (EISENMAN, MILUZZO, *et al.*, 2009).

Processamento. Em aplicações *crowdsensing*, a capacidade de processamento tem como característica ser um recurso variável. Este motivo se deve pelas diferenças de dispositivos dos participantes e a competição de recursos com outros aplicativos no dispositivo (GANTI, YE e LEI, 2011).

Armazenamento. Assim como a capacidade de processamento, a capacidade de armazenamento compete por recursos com os outros aplicativos e necessidades do participante. Para estas aplicações é conveniente o recurso de armazenamento para temporariamente guardar os relatórios criados, visto que, pela mobilidade do participante, nem sempre o dispositivo está conectado. Também, o armazenamento é conveniente por questões de privacidade, pois o participante pode verificar, depois da coleta, quais as informações ele permite compartilhar no sistema (CHRISTIN, REINHARDT, *et al.*, 2011).

Relatório. Para que o servidor de aplicações compreenda os dados coletados por diversos participantes, normalmente, as informações coletadas são formatadas em relatórios padronizados. No paradigma participativo o relatório é preenchido pelo participante, enquanto que no paradigma oportunista é criado, armazenado e enviado automaticamente pelo dispositivo do participante.

Servidor de aplicação. É o sistema remoto que coordena as **tarefas** dos dispositivos como a atualização da aplicação, a solicitação para atividades especiais de coleta e a criação de canal para contato com os desenvolvedores. Geralmente implantado em infraestruturas de computação em nuvem (KOCK, CARDONHA, *et al.*, 2013), possui recursos de **processamento e armazenamento** com características bem distintas dos dispositivos dos participantes (GANTI, YE e LEI, 2011). Como o volume de dados pode variar com a adesão de mais participantes e consequentemente mais relatórios, a infraestrutura de computação em nuvem é mais adequada para as aplicações *crowdsensing*, uma vez que pode dinamicamente suprir mais recursos em momentos de picos de processamento ou desprovê-los em momentos ociosos do sistema (LANE, MILUZZO, *et al.*, 2010).

A **visualização** é sem dúvida uma função importante do servidor de aplicação, pois é onde os dados coletados pelos diversos participantes são agrupados a fim de expressar uma representação do resultado final.

3.4 Desafios de sistemas *crowdsensing*

Crowdsensing como as outras especializações de *crowdsourcing* é baseado no engajamento das pessoas. Estes sistemas que contam com a participação de indivíduos dispostos a coletar dados, por alguns autores, são definidos como sensoramento participativo (BURKE, ESTRIN, *et al.*, 2006), cujos desafios também envolvem o **recrutamento e retenção de participantes**, a **qualidade dos dados**, a **colaboração e coordenação** e a **proteção da privacidade**.

Recrutamento e retenção de participantes. A adesão e retenção de participantes são fatores críticos para o sucesso de projetos *crowdsensing*. É importante a compreensão de quais são as motivações dos participantes para

contribuir inicialmente e de forma contínua neste sistema (REDDY, ESTRIN e SRIVASTAVA, 2010). Por exemplo, o estudo destes mesmos autores mostra que a região de atuação interfere na adesão de participantes e também que os participantes tendem a se desmotivar ao longo da campanha. Algumas pesquisas apontam que a motivação está relacionada à obtenção de incentivos monetários ou à obtenção de recompensas intangíveis como a satisfação de interesses pessoais, divertimento ou reconhecimento social (KAUFMANN, SCHULZE e VEIT, 2011). Os incentivos nestes sistemas se classificam em: **ganho pessoal, ganho monetário e reconhecimento social** (OMOKARO, 2012). Na primeira categoria o incentivo é preocupação moral com o problema em estudo. Na segunda categoria é um pagamento pela contribuição. Já a última categoria, os participantes obtêm reconhecimento dentro de uma comunidade, por exemplo, ganhando gradualmente troféus digitais a cada objetivo atingido (OMOKARO, 2012).

Qualidade dos dados. Em sistemas *crowdsensing*, uma parte do desafio da qualidade dos dados está na imprevisibilidade do comportamento humano (STEVENS, 2012). As ações voluntárias dos participantes, como as intenções maliciosas, e também as involuntárias, por exemplo, a propensão a alguma escolha, podem afetar na precisão dos dados. Disponibilizar instruções para métodos normalizados ajuda a educar os participantes sobre os limites máximos e mínimos da aplicação e assim, diminuir eventuais desvios (GOUVEIA, FONSECA, *et al.*, 2004). A outra parte do desafio é o lado técnico, uma vez que os sensores nos dispositivos dos participantes são menos precisos do que em equipamentos especializados (STEVENS, 2012). Para lidar com ambos os desafios, alguns pesquisadores pressupõem que estes sistemas possuem dados redundantes em termos de posição e tempo, de modo que ferramentas estatísticas podem ser empregadas (GANTI, YE e LEI, 2011).

Colaboração e coordenação. Principalmente em aplicações com o paradigma participativo, a ausência de coordenação do sistema pode se tornar um fator negativo para agregação dos dados, especialmente aquelas voltadas para cidade. A coleta de dados demasiadamente espaçados geograficamente ou temporalmente, pode invalidar a utilização dos dados (STEVENS e D'HONDT, 2010). Os sistemas de

coordenação podem indicar rotas ou pontos que precisam ter informações coletadas (STEVENSON e D'HONDT, 2010).

Privacidade. A percepção de privacidade é uma característica específica para cada indivíduo (FOGEL e NEHMAD, 2009). Também existem diferenças na criticidade de privacidade quanto aos paradigmas de *crowdsensing*, por exemplo, o oportunista, comparado ao participativo, tem a privacidade mais ameaçada, já que os participantes têm controle restrito dos dados coletados (KHAN, XIANG, *et al.*, 2013). Desta maneira, controles personalizados pelo próprio participante de quando e quais informações podem ser disponibilizadas são recomendáveis (CHRISTIN, REINHARDT, *et al.*, 2011). A geolocalização, com detalhamento de tempo e frequência, com granularidade por indivíduo é um exemplo de ameaça a privacidade, uma vez que pode ser inferido o local de residência, o trajeto ou até os momentos de deslocamentos que são normalmente realizados pelos participantes (CHRISTIN, REINHARDT, *et al.*, 2011). O áudio também é outro exemplo de dados coletados que os participantes sentem restrições de privacidade (CHON, LANE, *et al.*, 2013). Já em prol da privacidade, estão mecanismos de agrupamento de dados e generalização do contexto (CHRISTIN, REINHARDT, *et al.*, 2011). Outra característica deste tipo de sistema é que os participantes têm majoritariamente mínimo contato com os administradores do sistema, às vezes apenas por e-mail. Assim, precisa-se haver entre as partes certa relação de confiança de privacidade (CHRISTIN, REINHARDT, *et al.*, 2011).

3.5 Limitações de recursos

As aplicações de *crowdsensing* possuem limitações similares a aquelas encontradas em redes de sensores, porém pela dinâmica de utilização do participante e heterogeneidade de dispositivos, assumem requerimentos diferentes (GANTI, YE e LEI, 2011). Embora os recursos dos dispositivos, tais como, energia, processamento, armazenamento e comunicação são geralmente mais robustos do que em rede de sensores, também são limitados. Devem ser analisados de forma a prever o funcionamento global do dispositivo e não exclusivamente para a aplicação de *crowdsensing* (GANTI, YE e LEI, 2011). Logo, antes do desenvolvimento, é

necessário refletir como a aplicação e os demais potenciais aplicativos no mesmo dispositivo irão compartilhar os recursos sem significativamente afetar no desempenho de cada um (GANTI, YE e LEI, 2011). Por exemplo, aplicações que operam com o paradigma oportunista, devem prever que são secundárias e funcionarão em pequenos períodos intermitentes (CAMPBELL, EISENMAN, *et al.*, 2008).

As aplicações *crowdsensing* são limitadas também pelo tipo de sensor disponível originalmente no aparelho. Exemplos característicos são aquelas que usam acelerômetros, as quais foram beneficiadas pela incorporação destes no *smartphones* com intuito de aperfeiçoar a interface do usuário (LANE, MILUZZO, *et al.*, 2010).

A natureza operacional é outro fator de limitação. Os dispositivos estão vinculados mais as interações entre pessoas e o seu entorno do que no sensoramento de fenômenos físicos, portanto, estão normalmente disponíveis em lugares convenientes para os usuários como no bolso da roupa ou dentro de uma bolsa e não necessariamente no ponto de coleta mais confiável para a leitura do sensor (CAMPBELL, EISENMAN, *et al.*, 2008).

3.6 Mecanismos de agregação e análise

Em aplicações de *crowdsensing* a agregação de dados pode ocorrer em dois locais distintos, com funções complementares: **localmente no dispositivo** e **remotamente no servidor de aplicação** (GANTI, YE e LEI, 2011).

Localmente no dispositivo. No dispositivo, a agregação ocorre por dois motivos: economia de recursos e simplificação no tratamento dos dados do servidor de aplicações (GANTI, YE e LEI, 2011). Por exemplo, nas aplicações que monitoram sons específicos, a primeira qualificação de dados será realizada no *smartphone* (RANA, CHOU, *et al.*, 2010). Esta tarefa será menos dispendiosa de recursos energéticos e banda de comunicação para o próprio dispositivo, do que transmitir todas as amostras realizadas pelo microfone, além disso, diversos dispositivos executando simultaneamente estas funções podem criar gargalos no sistema remoto, tanto na recepção dos dados, quanto no processamento.

Remotamente no servidor de aplicação. Em última análise a agregação no sistema remoto é o objetivo fim de sistemas *crowdsensing*, sendo usada para detectar padrões e prever comportamento futuros. Esta parte do sistema tira proveito das informações geradas pela multidão de participantes, de modo que quanto maior o número de indivíduos contribuindo no sistema e mais informações compartilhadas por eles, mais fidedignas serão os modelos analíticos e os resultados (BRABHAM, 2008). Estes modelos analíticos empregam técnicas de mineração de dados e interpretações estatísticas para reconhecer padrões e prever problemas. Por exemplo, aplicações *crowdsensing* que monitoram o trânsito, usam ferramentas estatísticas para inferir se existe congestionamento em uma via utilizando os dados coletados pelos cidadãos e a capacidade da via (MOHAN, PADMANABHAN e RAMJEE, 2008). Desta forma, organismos de gestão pública de trânsito, a partir da mineração de dados de diversos dias, sobre o padrão de formação de congestionamento destas vias, poderiam aperfeiçoar os tempos dos semáforos desta região e melhorar a fluidez (GANTI, YE e LEI, 2011). Os modelos analíticos normalmente operam sobre as informações do banco de dados, mas dependendo do volume de dados coletado pelo sistema, pode operar diretamente no próprio fluxo de dados, gerando economia de recursos de armazenamento (BIEM, BOUILLET, *et al.*, 2010).

As técnicas de agregação são importantes também para a confidencialidade dos dados enviados pelos participantes, pois as informações de um indivíduo específico não são tipicamente relevantes para as aplicações *crowdsensing* comunitárias (MILUZZO, LANE, *et al.*, 2008). Porém estas técnicas de agregação de dados são específicas para cada aplicação (GANTI, YE e LEI, 2011).

3.7 Considerações finais

Devido a atual popularidade dos *smartphones*, *tablets* e das respectivas aplicações móveis, por exemplo, no Brasil o número de aquisição de *smartphones* já é maior do que celulares comuns (ABINEE, 2015), nota-se que os aplicativos *crowdsensing* estão conseguindo a confiança dos gestores públicos para atuação em suas políticas públicas. Entre os exemplos de adoção destes aplicativos em cidades, destacam-se aqueles os quais o cidadão participante relata problemas com os

serviços públicos. Por exemplo, o 1746¹² na cidade do Rio de Janeiro responde por 12% das solicitações da prefeitura¹³, o COLAB.re¹⁴ tem adesão de diversas prefeituras, inclusive das capitais do Paraná e Piauí e o Cidadão Online¹⁵, também já conta com ao menos 5 cidades paulistas se beneficiando do aplicativo.

Neste capítulo, foi discutido que o foco das aplicações *crowdsensing* é o compartilhamento de informações coletadas por sensores pertencentes aos dispositivos móveis de indivíduos que desejam mapear fenômenos de comum interesse. Em cidades, se enquadra o modelo de utilização de dados comunitário, cujas características são a necessidade para fácil disseminação, o potencial de espalhamento pela cidade, aproveitando o padrão de mobilidades dos participantes e adicionalmente, o suporte de análises complexas, considerando o recebimento das opiniões do fenômeno interpretadas pelos participantes. Também, considera-se que o baixo comprometimento e a privacidade são premissas para os participantes praticarem o altruísmo neste modelo (LANE, MILUZZO, *et al.*, 2010). Além disso, como discutido na seção 3.5, um ponto fundamental é que a ausência de coordenação do sistema pode se tornar um fator negativo para agregação dos dados, uma vez que a coleta de dados demasiadamente espaçadas geograficamente ou temporalmente, pode invalidar a utilização dos dados (STEVENS e D'HONDT, 2010). Um exemplo de coordenação para estes sistemas é a indicação de rotas ou pontos que precisam ter informações coletadas (STEVENS e D'HONDT, 2010). O próximo capítulo trata justamente deste ponto, discutindo técnicas de computação distribuída que podem ajudar a construção de sistemas de coordenação de informação em cidades.

A tecnologia *crowdsensing* discutida neste capítulo, composta pelos três grupos fundamentais: participantes, administrador do sistema e usuários finais, será usada neste trabalho para a obtenção da parte correspondente ao serviço de monitoração participativo direcionado a cidades proposto. Considerando que cada indivíduo tem a sua própria escolha de conectividade para participação, por exemplo, 3G, 4G ou até mesmo 802.11, as camadas de rede e enlace devem ser flexíveis para suportar esta comodidade do participante, portanto neste trabalho, a única restrição considerada

¹² http://www.1746.rio.gov.br/?page_id=5

¹³ <http://www.gestaoporprocessos.com.br/12-dos-contatos-com-a-prefeitura-do-rio-sao-feitos-via-app-e-website/>

¹⁴ <http://www.colab.re/>

¹⁵ <https://cidadaoonline.sistemas4r.com.br/>

nestas camadas ao participante, será o indivíduo ter acesso a internet, não necessariamente em tempo real.

4 SISTEMAS DISTRIBUÍDOS

Fundamentalmente, o foco desta pesquisa está no estudo de um arcabouço de tecnologia da informação que integre dados gerados por dois tipos distintos de fonte de informação: sensores instalados nos serviços das cidades e relatórios coletados por cidadãos, por sistemas *crowdsensing*, pertinentes ao contexto do sistema. Para tanto serão analisados neste capítulo os sistemas distribuídos, em especial, uma abstração característica destes sistemas, denominada *middleware*. A definição, os desafios e paradigmas de sistemas distribuídos serão apresentados, como também a definição e características de *middlewares* relacionados ao contexto em estudo: rede de sensores e sistemas *crowdsensing*.

4.1 Introdução

Define-se sistema distribuído como aquele no qual os componentes de *hardware* ou *software*, localizados em computadores interligados em rede, comunicam-se e coordenam suas ações apenas enviando mensagens entre si (COULOURIS, DOLIMORE, *et al.*, 2012). Também segundo os autores, a principal motivação para construir e usar sistemas distribuídos é proveniente do desejo de compartilhar recursos, tanto de *hardware* como de *software*. As consequências no uso destas técnicas são:

Concorrência. Execução de programas concorrendo pelos mesmos recursos.

Inexistência de relógio global. Existem limites na precisão do sincronismo para a ordenação da troca de mensagens.

Falhas independentes. Cada componente do sistema pode falhar independentemente, deixando os outros ainda em funcionamento.

4.2 Desafios dos sistemas distribuídos

Os desafios de sistemas distribuídos, segundo COLOURIS, et al. (2012), são a **heterogeneidade, divulgação do sistema, segurança, escalabilidade, tratamento de falhas, concorrência, transparência e qualidade de serviço.**

Heterogeneidade. Devido à variedade de redes, sistemas operacionais, *hardwares* e linguagens de programação existentes, técnicas de padronização devem ser implantadas, como o uso de algoritmos e protocolos únicos, para que as entidades do sistema possam se beneficiar do compartilhamento de recursos entre elas (ZHOU, ZHENG, *et al.*, 1993). O *middleware* está entre as técnicas de padronização, sendo uma camada de *software* que fornece uma abstração de programação, mascaramento da heterogeneidade das redes, do *hardware*, dos sistemas operacionais e das linguagens de programação subjacentes (COULOURIS, DOLIMORE, *et al.*, 2012). Esta técnica será detalhada durante o decorrer deste capítulo.

Divulgação do sistema. Para integração de componentes desenvolvidos por diversos programadores aos sistemas distribuídos é necessário que as interfaces estejam publicadas, ou em outras palavras, que o sistema seja aberto. A estratégia do desenvolvimento de sistemas abertos evita a estagnação, o que é especialmente indesejável a sistemas distribuídos voltado a rede de sensores, principalmente pela dinâmica característica deste tipo de rede, gerada pela mobilidade, inclusão e falha dos nós (HADIM e MOHAMED, 2006).

Segurança. Como o processamento é efetuado colaborativamente por diversas entidades do sistema, mecanismos de autenticação e criptografia podem ser usados para proteção da confidencialidade, integridade e disponibilidade aos recursos e informações compartilhadas. Não apenas mecanismos para prover a segurança de comunicação podem ser utilizados, mas também a execução do processamento (THAIN, TANNENBAUM e LIVNY, 2005).

Escalabilidade. Controlar o custo dos recursos físicos, a perda de desempenho e impedir que recursos de *software* se esgotem, são medidas que sistemas distribuídos devem desempenhar conforme aumenta a quantidade de entidades (COULOURIS, DOLIMORE, *et al.*, 2012). Ainda segundo os mesmos autores, um

sistema distribuído é considerado escalável se o custo da adição de um usuário for um valor constante, em termos dos recursos que devem ser adicionados.

Tratamento de falhas. Em um sistema distribuído qualquer parte pode falhar individualmente. O desafio encontra-se na necessidade dos demais componentes possuírem mecanismos de tratar as eventuais falhas das partes do sistema que dependem (TANENBAUM e VAN STEEN, 2007).

Concorrência. Quanto maior a presença de usuários também aumenta a concorrência pelos recursos do sistema. Devido justamente ao impacto no desempenho, alguns autores usam a concorrência de processos como métrica de comparação entre sistemas distribuídos (ZHANG, FRESCHL e SCHOPF, 2003).

Transparência. A operação da distribuição do sistema deve ser transparente para o desenvolvedor de integração, de modo que este apenas se preocupe com o seu projeto propriamente dito. O uso de técnicas de transparência aplicadas a abstrações de sistemas distribuídos, como o *middleware* (VINOSKI, 1997), visa à percepção de um sistema integrado, ao invés de uma coleção de componentes desconexos.

Qualidade de serviço. O sistema precisa prover aspectos mínimos de garantia de acesso aos serviços como confiabilidade, segurança e desempenho, sendo uma característica também desejável a rede de sensores (WANG, CAO, *et al.*, 2008).

4.3 Paradigmas de comunicação de sistemas distribuídos

Existem diversos modos possíveis de comunicação entre as entidades de um sistema distribuído. Para EUGESTER, *et al.* (2003), são definidos como paradigmas de comunicação e para COLOURIS, *et al.* (2012), estão divididos em **comunicação entre processos**, **invocação remota** e **comunicação indireta**.

Comunicação entre processos. Refere-se ao suporte de comunicação de nível mais baixo entre o processo remetente e o destino como a passagem de mensagens do tipo enviar e receber. Entre os modos possíveis em sistemas distribuídos estão à

comunicação por datagrama, o UDP e por fluxo de dados, o TCP, podendo ser usado entre um par de processos específico (*unicast*) ou envolvendo a comunicação entre um processo para vários processos (*multicast*) (TANENBAUM e VAN STEEN, 2007).

Invocação remota. Cobre uma variedade de técnicas baseadas na troca bilateral entre as entidades que se comunicam em sistema distribuído, resultando na chamada de uma operação, um procedimento ou um método remoto (COULOURIS, DOLIMORE, *et al.*, 2012). Também segundo os autores, entre as técnicas para invocação remota estão os **protocolos de requisição-resposta**, as **chamada de procedimento remoto** e a **invocação de método remoto**.

- **Protocolos de requisição-resposta.** O HTTP se enquadra nesta categoria de protocolos, os quais envolvem a comunicação pela troca do par de mensagens do tipo requisição e resposta, do cliente para o servidor (FIELDING, GETTYS, *et al.*, 1999). A primeira mensagem geralmente contém a codificação da operação que será executada pelo servidor, com eventuais argumentos associados para efetivar a operação.

- **Chamada de procedimento remoto.** O RPC (*remote procedure call*) faz as interações remotas aparecerem como locais. Uma chamada de procedimento remoto é iniciada pelo cliente, ao enviar uma mensagem para um servidor remoto executar um procedimento específico. O cliente comumente permanece bloqueado, aguardando até a resposta do servidor (BIRRELL e NELSON, 1984). Para permitir que os servidores sejam acessados por diferentes clientes, é comum usar a linguagem de descrição de interface (IDL).

- **Invocação de método remoto.** O RMI (*remote method invocation*) é muito parecido com o RPC, porém é estendido para objetos distribuídos. Um objeto chamador local pode invocar um método em um objeto remoto de forma transparente, o que torna este método com integração mais forte às linguagens de programação (MAASSEN, VAN NIEUWPOORT, *et al.*, 1999).

Comunicação indireta. As técnicas anteriores apresentam uma relação bilateral entre remetentes e destinatários, com os remetentes encaminhando explicitamente as mensagens / invocações para os destinatários. Já no caso da comunicação indireta,

existe uma terceira entidade que possibilita o desacoplamento entre remetentes e destinatários (COULOURIS, DOLIMORE, *et al.*, 2012). Um exemplo de funcionamento desta terceira entidade é representado na figura 7, pelo serviço de notificação da técnica de **publicar e assinar**. Para os mesmos autores, entre as técnicas de comunicação indireta estão a **comunicação em grupo**, **memória compartilhada distribuída**, **espaço de tupla**, **filas de mensagens** e a técnica de **publicar e assinar**.

Figura 7: Interações do modelo de publicar e assinar

Fonte: Adaptado de COULOURIS, et al, 2012

- **Comunicação em grupo.** Este modelo de comunicação emprega a abstração de grupos, a qual é formada por um conjunto de destinatários com a mesma afinidade. O remetente não conhece a identidade dos destinatários, é uma comunicação de um indivíduo da rede para muitos (BIRMAN, SCHIPER e STEPHENSON, 1991).

- **Memória compartilhada distribuída.** Os sistemas DSM (*Distributed Shared Memory*) fornecem uma abstração para compartilhamento de dados entre dispositivos que não compartilham a memória física. Os processos destes dispositivos acessam o sistema DSM por meio de leituras e gravações de modo transparente, como se fosse à memória local dentro de seus espaços de endereçamento (LI e HUDAK, 1989).

- **Espaço de tupla.** É um modelo pelo qual os processos podem colocar itens de dados estruturados arbitrários, chamados tuplas, em um espaço persistente. Outros

processos podem ler ou remover tais tuplas desse espaço, especificando padrões de interesse (GELERNTER, 1985).

- **Filas de mensagens.** É um serviço ponto a ponto onde os processos produtores podem enviar mensagens para uma fila especificada e os processos consumidores recebem mensagens das filas ou são notificados da chegada de novas mensagens na fila. Este paradigma oferece desacoplamento temporal entre os processos produtores e consumidores (HAPNER, BURRIDGE, *et al.*, 2002).

- **Publicar e assinar.** É definido com um paradigma de interação entre módulos produtores (publicadores) e consumidores de dados (assinantes) através de eventos (EUGSTER, FELBER, *et al.*, 2003). Nele as informações criadas poderão ser consumidas pelos usuários do sistema, os módulos assinantes, que demonstrarem interesse em recebê-las. Neste esquema de interação a troca de mensagens, denominada de eventos, é realizada assincronamente, fornecendo o desacoplamento entre os elementos do sistema e assim favorecendo a flexibilidade e escalabilidade para as aplicações (EUGSTER, FELBER, *et al.*, 2003). O modelo possui um serviço de mediação de eventos entre os publicadores e assinantes, o qual armazena as subscrições dos clientes além de receber e enviar as notificações dos eventos. O funcionamento se inicia quando ocorre a subscrição a um evento específico ou a um padrão de eventos pelo publicador. Em seguida, os eventos criados e publicados pelos produtores serão notificados pelo serviço de gerenciamento, aos assinantes que particularmente se inscreveram a estes eventos para consumo. Para este sistema, segundo EUGESTER, *et al.* (2003), o desacoplamento será em termos de:

- i) Espaço: Os publicadores e assinantes não têm a necessidade de conhecer uns aos outros;
- ii) Tempo: As interações podem ser executadas em instantes diferentes, por exemplo, quando os publicadores ou assinantes estiverem desconectados;
- iii) Sincronismo: Existe uma independência do fluxo de execução, já que os módulos do sistema não precisam interromper seu funcionamento principal para a produção ou consumo dos eventos.

Um exemplo de funcionamento deste paradigma é representado na figura 7. Também de acordo com EUGESTER et al. (2003), quando os clientes realizam a subscrição no sistema estão demonstrando interesse a um padrão de eventos, que pode ser classificado em diferentes esquemas:

- i) Baseado em tópicos: Ocorre à publicação ao sistema de gerenciamento por uma palavra-chave única, que indicará o conteúdo do evento. Isto permite ao assinante definir sua subscrição a partir desta palavra-chave;
- ii) Baseado em conteúdo: Permite ao assinante expressar seus interesses utilizando uma consulta no conteúdo dos eventos e não a um tópico previamente atribuído ao evento;
- iii) Baseado em tipo: Os eventos são tratados como objetos e a classificação é realizada de acordo com o tipo do objeto.

4.4 Middleware

O *middleware* é uma camada de *software* que fornece uma abstração de programação, do *hardware*, do sistema operacional, das linguagens de programação subjacentes e da heterogeneidade da rede (COULOURIS, DOLIMORE, et al., 2012). Também segundo os autores, o *middleware* é representado por processos em um conjunto de computadores que interagem entre si para implantar o suporte para comunicações e compartilhamento de recursos para sistemas distribuídos. Seu objetivo é fornecer elementos básicos úteis para a construção de componentes de *software* que possa interagir em um sistema distribuído. Normalmente eleva o nível das atividades de comunicação e aplicação por meio do suporte a abstrações.

Além de resolver os problemas de heterogeneidade, o *middleware* fornece um modelo computacional uniforme para ser usado pelos desenvolvedores de aplicativos distribuídos. Os modelos possíveis incluem a invocação remota de objetos, a notificação remota de eventos, o acesso remoto a banco de dados e o processamento de transações distribuídas. A figura 8 representa as camadas lógicas de serviço de *software* e *hardware* em sistemas distribuídos e indica o posicionamento do *middleware* entre estas camadas.

Figura 8: Camadas lógicas de serviço de software e hardware em sistemas distribuídos

Fonte: Adaptado de COULOURIS, et al, 2012

4.5 *Middleware* para rede de sensores

Os *middlewares* direcionados a rede de sensores têm como objetivo apoiar o desenvolvimento, manutenção, implantação e execução das aplicações baseadas em sensores em rede, o que inclui (ROMER, KASTEN e MATTERN, 2002):

- i) Mecanismos para a formulação das tarefas de sensoriamento;
- ii) Comunicação dessas tarefas para a rede de sensores;
- iii) Coordenação dos nós para dividir as tarefas e distribuí-la aos nós sensores individuais;
- iv) Agregação dos dados das leituras individuais;
- v) Apresentação do resultado para o administrador da tarefa.

Para HADIM e MOHAMED (2006), os desafios do *middleware* em rede de sensores são: **o gerenciamento da energia e dos recursos limitados, escalabilidade, mobilidade e topologia dinâmica da rede, heterogenidade, organização dinâmica da rede, integração com o mundo real, conhecimento da aplicação, agregação de dados, qualidade de serviço e segurança.** Embora muitos destes desafios são os mesmos já citados no capítulo 2 para rede de sensores, alguns assumem restrições adicionais em um *middleware*.

Heterogenidade. As rede de sensores são voltadas para aplicações específicas (YICK, MUKHERJEE e GHOSAL, 2008), logo o *hardware* e as funcionalidades assumem características distintas conforme a aplicação. O *middleware* precisa considerar as especificidades das aplicações.

Organização dinâmica da rede. Como o conhecimento da rede é essencial para que a rede de sensores funcione corretamente, o *middleware* deve fornecer mecanismos para descoberta do estado da rede (HADIM e MOHAMED, 2006) e suporte para a configuração automática e tratamento de erros pelos nós, uma vez que estes operam sem supervisão, em contraste com os sistemas tradicionais (ROMER, KASTEN e MATTERN, 2002).

Conhecimento da aplicação. Enquanto que os *middlewares* tradicionais são projetados para acomodar uma grande variedade de aplicações sem necessariamente precisar do conhecimento da aplicação, no entanto, para rede de sensores, devem fornecer mecanismos para a inclusão do conhecimento da aplicação no sistema. Esta estratégia pode significativamente aperfeiçoar a eficiência energética e o uso dos recursos da rede pelos nós (WANG, CAO, *et al.*, 2008).

Agregação de dados. Tipicamente, aplicações de rede de sensores envolvem redundância de dados em um conjunto de nós localizados. A agregação dos dados destas diferentes fontes, em pontos específicos da rede, reduz globalmente o número de transmissões para o sorvedouro, o que conseqüentemente melhora a eficiência energética da rede de sensores (KRISHNAMACHARI, ESTRIN e WICKER, 2002).

Segurança. Tanto pela limitação de recursos, quanto pelo uso do canal de comunicação ou ainda pelo tipo de sensoramento dos nós, por exemplo, em aplicações que coletam dados sensíveis, a segurança é um grande desafio em rede de sensores (WANG, ATTEBURY e RAMAMURTHY, 2006). Os *middlewares* devem concentrar esforços no desenvolvimento e integração de segurança para alcançar diferentes requisitos como a confidencialidade, autenticação e integridade (HADIM e MOHAMED, 2006).

4.6 Características de *middleware* para rede de sensores

Um *middleware* para rede de sensores é composto por quatro componentes: **abstrações de programação, serviços do sistema, suporte a tempo de execução e mecanismos de QoS** (WANG, CAO, *et al.*, 2008) e também segundo os autores, a arquitetura é dividida em 3 camadas: **nó sensor, sorvedouro e terminal de usuário**, conforme mostrado na figura 9:

Figura 9: Arquitetura de *middleware* para RSSF

Fonte: Adaptado de WANG, et al, 2008

Abstrações de programação. Este componente define a interface entre o *middleware* e a aplicação e possui três aspectos, segundo WANG, et al. (2008):

- i) Nível de abstração: Se refere em como a aplicação enxerga o sistema, partindo do patamar em que é possível programar um nó exclusivamente, até o nível ao qual o programador expressa apenas um comportamento global do sistema para replicar-se aos demais nós da rede de sensores;
- ii) Tipo de interface: É o estilo de programação da interface com a aplicação, podendo ser mais descritivo, baseado em regras declarativas ou também especificar um conteúdo, por exemplo, nas interfaces baseadas em XML;
- iii) Paradigma de programação: Trata-se do modelo de programação das aplicações, como explorado na seção 4.4.

Serviços do sistema. Este componente incorpora as principais funcionalidades do *middleware* para rede de sensores como a execução e o gerenciamento da rede. Para WANG, et al. (2008), está dividido em duas categorias:

- i) Serviços comuns que auxiliam no gerenciamento da informação e na infraestrutura da rede de sensores;
- ii) Serviços de domínio que facilitam o desenvolvimento de uma aplicação com requisitos específicos, por exemplo, aumentando a taxa de amostragem do fenômeno monitorado.

Suporte a tempo de execução. Estende as funções do sistema operacional embarcado dos sensores para prover as funcionalidades dos serviços do sistema. Segundo WANG, et al. (2008), entre as funcionalidades providas por este componente para rede de sensores estão a comunicação entre processos, o controle de potência e o armazenamento de dados da rede.

Mecanismos de QoS. Define as restrições de qualidade de serviço do sistema. Para WANG, et al. (2008), este componente opera nas ambas as camadas de rede e aplicação, podendo considerar métricas de rede entre as quais estão a vazão de dados, latência e perda de pacotes ou métricas vinculadas a aplicação como a precisão dos dados, atraso na agregação ou tempo de vida do sistema.

4.7 *Middleware para crowdsensing*

Os *middlewares* para *crowdsensing* têm como objetivo apoiar o desenvolvimento, manutenção, implantação e execução das aplicações baseadas no sensoriamento participativo e oportunista (RA, LIU, et al., 2012) (AGARWAL, BANERJEE, et al., 2013) (CARDONE, FOSCHINI, et al., 2013). Tipicamente são divididos em duas camadas, sendo uma instalada no dispositivo móvel do participante e outra no servidor remoto (GANTI, YE e LEI, 2011), figura 10.

Entre os desafios apontados de *middlewares* para *crowdsensing* (AGARWAL, BANERJEE, et al., 2013), (DAS, MOHAN, et al., 2010), (RA, LIU, et al., 2012) e também observados em *middlewares* de redes sociais móveis (KARAM e MOHAMED,

2012) estão: **heterogeneidade de dispositivos, agregação, qualidade e confiabilidade dos dados, mobilidade dos participantes, eficiência energética, privacidade, escalabilidade, engajamento e retenção dos participantes.** Também, alguns destes desafios são os mesmos identificados para *crowdsensing* já citados no capítulo 3, porém têm restrições adicionais em um *middleware*.

Figura 10: Camadas de composição de *middleware* para *crowdsensing*

Fonte: Adaptado de GANTI, YE e LEI, 2011

Mobilidade dos participantes. Para a contextualização da informação coletada pelo participante no espaço e no tempo, os dispositivos móveis normalmente utilizam os dados do próprio sensor GPS do dispositivo. Deste modo, os serviços de localização dos *middlewares* assumem além do papel de auxiliar nos serviços de agregação dos dados (HACHEM, PATHAK e ISSARNY, 2014), também tratar eventuais aspectos de mobilidade da aplicação (PAN, ZHENG, *et al.*, 2013).

Eficiência energética. Este desafio é similar aos *middlewares* para rede de sensores, como observado na seção 4.7, embora os dispositivos móveis normalmente dispõem de mais recursos de processamento e armazenamento e as baterias são recarregadas pelos usuários, diferentemente dos sensores pertencentes as rede de sensores. Também como já apontado na seção 3.7, a utilização dos sensores do dispositivo móvel consome tipicamente mais energia e recursos. O desafio do *middleware* é ser sensível aos recursos disponíveis do dispositivo e assim adaptar o seu funcionamento (AGARWAL, BANERJEE, *et al.*, 2013).

Privacidade. Os sistemas voltados a aplicações *crowdsensing* devem possuir configurações de privacidade e anonimato (DAS, MOHAN, *et al.*, 2010). O *middleware* também deve garantir que os participantes não sejam rastreados entre eles, evitando inferências indiretas (KARAM e MOHAMED, 2012). O grande desafio é fornecer uma técnica de privacidade simples mas poderosa, consumindo relativamente pouco recurso (KARAM e MOHAMED, 2012).

Escalabilidade. O *middleware* para aplicações *crowdsensing* deve ser flexível o suficiente para gerenciar o aumento do número de participantes sem o comprometimento do desempenho do sistema (DAS, MOHAN, *et al.*, 2010). O entendimento da criticidade e o volume da carga de dados sobre a rede são pontos a ser planejados em prol da escalabilidade (AGARWAL, BANERJEE, *et al.*, 2013). Por exemplo, a utilização de técnicas limitando a participação de dados com sensoriamento redundante, pode favorecer a escalabilidade (HACHEM, PATHAK e ISSARNY, 2014).

Engajamento e retenção de participantes. A quantidade de cognição pelo lado do participante precisa ser considerada com atenção pelo *middleware* (AGARWAL, BANERJEE, *et al.*, 2013). Por exemplo, a quantidade de envolvimento dos participantes em aplicações de sensoriamento participativo é maior do que no paradigma oportunista. Tanto estas interações precisam ser antecipadamente identificadas no desenvolvimento do *middleware*, como os mecanismos de incentivos. As recompensas monetárias, o anonimato, as configurações de privacidade e o controle de recursos do dispositivo são funções que incentivam a participação dos usuários (RA, LIU, *et al.*, 2012).

4.8 Características de *middleware* para *crowdsensing*

As características dos *middlewares* para *crowdsensing* são similares aquelas observadas em aplicações de rede de sensores, de tal modo que alguns autores já utilizaram *middlewares* apropriados para RSSF em aplicações de *crowdsensing* (PERERA, ZASLAVSKY, *et al.*, 2012) e (PERERA, JAYARAMAN, *et al.*, 2014).

Similarmente aos *middlewares* para rede de sensores, as propostas para *crowdsensing* são compostas também por quatro componentes: **abstrações de programação, serviços do sistema e suporte a tempo de execução** (RA, LIU, *et al.*, 2012).

Abstrações de programação. Do mesmo modo do que em *middlewares* de rede de sensores, este componente possui a função de definir a interface entre a aplicação. É possível verificar diversos tipos de interface, por exemplo, baseado em regras por auxílio de interfaces gráficas (CARDONE, FOSCHINI, *et al.*, 2013) ou declarativas utilizando o XML (RA, LIU, *et al.*, 2012). Já com relação ao nível de abstração, em sistemas *crowdsensing* o *middleware* pode assumir uma condição de abstração ao qual o programador expressa apenas um comportamento global do sistema, tornando transparentes os aspectos sobre mobilidade e heterogeneidade dos dispositivos móveis participantes (YE, GANTI, *et al.*, 2012) ou pode conter políticas de atribuição de tarefas aos participantes como a localização da tarefa, porcentagem de participantes, etc. (CARDONE, FOSCHINI, *et al.*, 2013). Nas abstrações de programação, há abordagens de paradigmas de programação tanto com técnicas de requisição-resposta (HACHEM, PATHAK e ISSARNY, 2014) ou com técnicas de publicar e assinar (ZARKO, ANTONIC e PRIPUZIC, 2013). Especialmente os trabalhos com enfoque de paradigmas de comunicação indireta, são elaborados com viés para escalabilidade, grande quantidade de dados e também para dispositivos com poucos recursos (ZARKO, ANTONIC e PRIPUZIC, 2013).

Serviços do sistema. Este componente contém as principais funções do *middleware*. Entre os serviços comuns à sistemas *crowdsensing* estão: a inscrição dos dispositivos, métodos de localização, agregação de dados (HACHEM, PATHAK e ISSARNY, 2014), algoritmos para uso eficiente dos recursos do dispositivo (YE, GANTI, *et al.*, 2012), mecanismos de incentivos dos participantes, segurança (RA, LIU, *et al.*, 2012) e visualização (CARDONE, FOSCHINI, *et al.*, 2013).

Suporte a tempo de execução. Estende as funções do sistema operacional para prover as funcionalidades dos serviços do sistema (RA, LIU, *et al.*, 2012). Entre as funcionalidades providas por este componente para *crowdsensing* estão:

- i) No dispositivo móvel: A execução da tarefa de sensoriamento, mecanismos de privacidade e consciência sobre o estado de utilização de recursos (RA, LIU, *et al.*, 2012).
- ii) No servidor remoto: Normalmente apenas o gerenciamento do sistema. Alguns autores sugerem que a execução da atividade de *crowdsensing* pode ser distribuída no dispositivo móvel e no servidor remoto. Nesta circunstância, o servidor pode orquestrar a atividade de *crowdsensing* requerida pelo gestor do sistema (RA, LIU, *et al.*, 2012).

4.9 Considerações finais

Pela definição discutida, nota-se que os sistemas distribuídos abrangem muitos dos desenvolvimentos tecnológicos mais significativos da atualidade. Para COULOURIS, *et al.* (2012), as seguintes tecnologias são destaques de contribuições dos sistemas distribuídos: *e-commerce* (finanças e comércio), mecanismos de busca e repositório de conteúdo (sociedade da informação), jogos online (entretenimento), registros eletrônicos (assistência médica), *e-learning* (educação), serviços baseados em mapas (transporte e logística), grades computacionais para suporte ao *e-science* (ciências) e em gerenciamento ambiental, o uso de tecnologia de sensores interligados em rede para gerenciar e monitorar o ambiente natural. Além desses exemplos, uma incorporação de sistemas distribuídos neste momento popular é a **computação em nuvem**. Esta baseia-se em um modelo que, por meio de uma rede de dados conveniente, ubíqua e sob demanda, acessa um conjunto compartilhado de recursos computacionais configuráveis, por exemplo, processamento, armazenamento, aplicações e serviços que podem ser rapidamente provisionados e liberados com um esforço mínimo de gerenciamento ou interação com o provedor de serviços (MELL e GRANCE, 2011). Uma especificidade de computação em nuvem também particularmente interessante a cidades é a **infraestrutura como serviço**. Conhecida pelo acrônimo em inglês, *IaaS* (*Infrastructure as a Service*), refere-se a especialização de computação em nuvem, oferecida ao cliente, para provisionar processamento, armazenamento, rede e outros recursos computacionais fundamentais, em que o cliente é capaz de implantar e executar softwares arbitrários, o que pode incluir sistemas operacionais e aplicativos (MELL e GRANCE, 2011). A

tecnologia IaaS será utilizada no desenvolvimento deste trabalho e será detalhada no apêndice E, no tema de computação em nuvem.

Neste capítulo foi discutido diversas tecnologias de sistemas distribuídos. Destaca-se para a implantação em cidades o **paradigma de comunicação indireta publicar e assinar** e a abstração *middleware*. O método publicar e assinar definido por EUGSTER, et al. (2003), discutido na seção 4.4, baseia-se na troca de mensagens assíncrona, fornecendo desacoplamento entre os elementos do sistema e favorecendo a flexibilidade e escalabilidade para aplicações, sendo por estas características utilizado em cidades (PIRO, CIANCI, et al., 2013). Portanto, será usado este paradigma de comunicação neste trabalho. Considerando ter sido desenvolvido justamente em técnicas de transmissão de mensagens, por exemplo, aquelas usadas em sistemas *crowdsensing* (IBM CORPORATION; EUROTECH, 2010), quanto por existir uma adaptação construída exclusivamente para rede de sensores (HUNKELER, TRUONG e STANFORD-CLARK, 2008), foi selecionado o MQTT neste estudo como a tecnologia que personifica o paradigma publicar e assinar. O MQTT será detalhado no apêndice C e em seguida a especificidade para rede de sensores, o MQTT-SN, será apresentado no apêndice D.

O *middleware*, como dito na seção 4.5, é a camada de *software* que fornece a abstração de programação, *hardware*, sistema operacional e heterogeneidade da rede (COULOURIS, DOLIMORE, et al., 2012). Tanto os *middlewares* direcionados a rede de sensores (ROMER, KASTEN e MATTERN, 2002) quanto a *crowdsensing* (RA, LIU, et al., 2012) (AGARWAL, BANERJEE, et al., 2013) (CARDONE, FOSCHINI, et al., 2013) têm como objetivo apoiar o desenvolvimento, manutenção, implantação e execução destas aplicações. Os componentes desta tecnologia comum a estas fontes de dados, discutido neste capítulo: abstrações de programação, serviços do sistema e suporte a tempo de execução, serão usadas neste trabalho para a obtenção do arcabouço tecnológico direcionado a cidades proposto, responsável pela integração do serviço de monitoração automático e participativo.

5 DESENVOLVIMENTO

Até agora foram detalhados os conceitos que serão empregados para a construção deste trabalho: as redes de sensores, os sistemas participativos e oportunistas para coleta de dados (*crowdsensing*) e os sistemas distribuídos. É na intersecção destes conceitos que esta pesquisa é desenvolvida, figura 11.

Figura 11: Foco do estudo - Intersecção dos conceitos de rede de sensores, *crowdsensing* e sistemas distribuídos

Fonte: Autor

Conforme apresentado no capítulo 1, o objetivo é a pesquisa de um **sistema de gerenciamento** direcionado à gestão de cidades para coleta, integração e interpretação dos dados de sensores, pertencentes aos serviços disponíveis da própria cidade e dos dados coletados e compartilhados altruisticamente pelos cidadãos (*crowdsensing*). A figura 12 ilustra esta segmentação entre os elementos deste sistema, cujos os desafios inicialmente encontrados são:

1 - Padronização dos dados provenientes dos sensores pertencentes aos serviços da cidade e do sensoriamento colaborativo pelos cidadãos.

2 - Infraestrutura de *hardware* e *software* do sistema de gerenciamento que comporte o volume de dados, conexões e transações compatíveis com possivelmente milhões de dispositivos que podem estar presentes nesta rede, representados aqui pelos ativos das cidades (ex.: postes de iluminação, semáforo de trânsito, bocas de lobo, etc.) e também pelos dispositivos dos cidadãos participantes.

3 - Modelos analíticos que correlacionem informações das fontes distintas de dados para gerar conhecimento, alertas e confirmações, auxiliando os gestores públicos na tomada de decisão.

4 - Algoritmos e métodos de diminuição de dados expirados e redundantes gerados pelas fontes de dados.

Figura 12: Elementos básicos do sistema: sensores, *crowdsensing* e sistema de gerenciamento

Fonte: Autor

No apêndice estão brevemente descritas as tecnologias utilizadas no desenvolvimento desta pesquisa para superação destes desafios pontuados, entre elas estão: o **6LoWPAN** que estabelece o suporte ao IPv6 para endereçamentos dos sensores, o **MQTT** e **MQTT-SN** como o conjunto de protocolos da camada de aplicação, que geram o suporte de comunicação distribuída para padronização dos dados ao sistema de gerenciamento. Também a estratégia de **laaS**, servindo como a infraestrutura de *hardware* elástica compatível com o volume de dados, conexões e transações esperadas.

No decorrer deste capítulo é destacado o escopo da pesquisa e apresentado a arquitetura que se pretende testar. Pela diversidade dos serviços em cidades com características e requisitos distintos, por exemplo, transporte, segurança, saúde,

saneamento, etc., é sugerido a análise de um serviço como cenário de aplicação para testar o conceito, em seguida são detalhados as estratégias de desenvolvimento e os resultados alcançados no estudo.

5.1 Escopo

De acordo com as observações sobre os conceitos de sistemas distribuídos, apresentadas no capítulo 4, foi verificado que o compartilhamento de recursos por diversos dispositivos heterogêneos é um desafio típico de sistemas distribuídos. Neste trabalho será adotada a utilização de um *middleware* como a estratégia para tratamento da heterogeneidade destas duas fontes de dados à cidades. Também no mesmo capítulo, foi observado que já existem vários trabalhos relacionados a *middlewares* exclusivamente direcionados a rede de sensores ou a *crowdsensing*, porém não consideram os desafios da integração destas fontes de dados e as eventuais correlações possíveis. Portanto, prevendo que a especificação das funcionalidades reutilizáveis por um *middleware* nas camadas do sorvedouro, do nó sensor ou do dispositivo móvel do participante se assemelha com os esforços de pesquisa realizados por outros autores, em trabalhos específicos para *middleware* em rede de sensores ou *crowdsensing*, este objetivo está fora do escopo desta pesquisa. Ao invés disto, este trabalho concentra-se no detalhamento dos requisitos para a integração destes sistemas como as camadas de rede e os componentes necessários no servidor.

Também não é contemplado neste estudo a análise da interatividade direta entre pessoas e sensores através de dispositivos móveis, por exemplo, no contexto de realidade aumentada (VAN KREVELEN e POELMAN, 2010).

5.2 Arquitetura de rede

A visão geral da arquitetura é apresentada a seguir na figura 13. Ela está compartimentada em três subdivisões: a rede de sensores, os dispositivos móveis dos participantes, criando a rede *crowdsensing* e o serviço de computação em nuvem representado o servidor deste sistema.

Figura 13: Visão geral da arquitetura

Fonte: Autor

Além do sistema operacional, a rede de sensores é composta por uma arquitetura de rede com diversas camadas, figura 14. Estas camadas cuidadosamente articuladas irão favorecer a implantação deste sistema em cidades. Entre as camadas estão:

Física / enlace. O padrão IEEE 802.15.4, descrito brevemente no apêndice A, é a instrumentação para estas duas camadas recomendada neste trabalho. Ele foi desenvolvido justamente para aplicações com baixa transferência de informações e

consumo de energia e inclusive vem sendo utilizado para cidades (SANCHEZ, MUNOZ, *et al.*, 2013). Estará presente no nó sensor e também no sorvedouro do lado da rede de sensores, porém no sorvedouro, do lado da conexão com a rede do servidor, é possível a utilização de outras tecnologias, tais como o IEEE 802.11, Ethernet, fibra ótica e até mesmo o 3G e 4G.

Rede. Considerando a popularidade e os benefícios das redes IP, bem como o endereçamento de 128 bits, compatível com redes direcionadas a cidades (SHELBY e BORMANN, 2011), o IPv6 é a recomendação deste trabalho. A adaptação do IPv6 para rede de sensores, o 6LoWPAN, que foi originalmente desenvolvido para o padrão IEEE 802.15.4 (MULIGAN e GROUP, 2007), nesta pesquisa é utilizado para manter compatível a rede de sensores com o protocolo IPv6. As principais características são referenciadas no apêndice B. No sorvedouro, do lado da conexão com o servidor, considera necessária a adaptação do IPv6 para IPv4 afim de manter a compatibilidade de endereçamento com equipamentos IPv4 ainda atuais.

Figura 14 Arquitetura da rede de sensor

Fonte: Autor

Transporte: Enquanto o TCP é o protocolo usado para o sorvedouro do lado da rede do servidor, o UDP é o protocolo de transporte do lado da rede de sensores. A

garantia de entrega de mensagens, quando aplicável a rede de sensores, não será oferecida pela camada de transporte. Neste caso, o protocolo de aplicação responsável pelo paradigma publicar e assinar assumirá este papel. Um caso possível é o MQTT-SN que comporta níveis de qualidade de serviço, por exemplo, o QoS 2, exatamente para cumprir este tipo de necessidade (HUNKELER, TRUONG e STANFORD-CLARK, 2008).

Figura 15 Arquitetura da camada de rede do dispositivo móvel

Fonte: Autor

A figura 15 representa a arquitetura das camadas de rede para apoiar a aplicação *crowdsensing* no dispositivo do participante. Verifica-se que as camadas inferiores a aplicação já pertence a estes dispositivos, elas são utilizadas de modo transparente pelos usuários quando eles acessam um navegador de internet em seu dispositivo móvel. Porém neste trabalho é especificado a presença do protocolo publicar e assinar na camada de aplicação, pois esta estratégia irá favorecer a escalabilidade e a heterogeneidade de dados neste sistema.

5.3 Funcionalidades esperadas do *middleware* para o nó sensor, sorvedouro e dispositivo móvel do participante

Considera-se que todos os dispositivos deste sistema possuam além do sistema operacional e das camadas de rede apresentadas, alguns componentes do *middleware* na camada da aplicação. Estes componentes irão proporcionar diversas características para operação integrada com as demais entidades do sistema, por exemplo, com o próprio servidor. Embora tenha sido apresentado na seção 5.2 que não é foco principal deste trabalho a especificação destes componentes ao nó sensor,

sorvedouro ou ao dispositivo móvel do participante, espera-se que na camada de aplicação destes dispositivos, exista ao menos algumas funcionalidades específicas para operação do *middleware*. Por exemplo, para o nó sensor, pondera-se como os componentes básicos o **sensoriamento especializado**, **localização**, **comunicação**, **gerenciamento de energia**, **gerenciador de dados** e **gerenciamento de regras**. A figura 16 ilustra uma arquitetura com a organização destes componentes.

Figura 16: Arquitetura da aplicação dos nós sensores

Fonte: Autor

Gerenciador de regras: Considera-se a presença de um gerenciador de regras o qual tenha a tarefa de customizar as funções dos demais componentes do sensor, a partir das atualizações provindas do sorvedouro. Este componente também pode contar com um módulo de segurança para verificar a autenticidade da comunicação com o sorvedouro, em especial as eventuais atualizações recebidas. Desta forma, as informações válidas recebidas podem ser processadas e então criadas novas regras para disseminação aos demais componentes.

Sensoriamento especializado: É este componente que efetivamente captura o dado do fenômeno ao qual o sensor monitora. Através de configurações externas a este componente pode-se customizar o seu funcionamento, por exemplo, sobre o período de amostragem do transdutor. Ainda, um módulo subsequente de processamento pode verificar se o dado capturado está acima do limiar com que a aplicação da rede de sensores está predisposta a receber, evitando que informações desnecessárias sejam propagadas pela rede.

Localização. Este componente representa a funcionalidade de geolocalizar o ativo na cidade, o que pode estar disponível em sensores fixos quanto móveis. A utilização deste componente em sensores fixos é justificada pela presença de normalmente milhares ativos na cidade (postes de iluminação, bocas de lobo, etc.), de modo que a geolocalização manual destes sensores pode ser uma atividade trabalhosa e susceptível a erros. Para sensores móveis, por exemplo, presentes em ônibus e caminhões de coleta de lixo, a contextualização da geolocalização do dado coletado é uma característica intrínseca destes sensores. Um módulo de pré-processamento adicional pode verificar quando publicar a atual posição do sensor, por exemplo, nos casos onde a diferença da medição anterior estiver acima do valor da precisão do GPS.

Gerenciador de dados. Este componente representa a persistência dos dados coletados para posterior transmissão. Esta funcionalidade é desejável em circunstâncias onde o nó sensor não está continuamente em conexão com o sorvedouro, por exemplo, para sensores móveis. Um módulo de pré-processamento pode acrescentar a este componente o condicionamento dos dados em pacotes maiores, realizando assim a agregação dos dados e otimizando a transmissão.

Comunicação. Espera-se que o fluxo de dados de entrada e saída do nó sensor esteja contido neste componente. Um módulo de pré-processamento pode ser adicionado para realizar a cadência de transmissão, recebendo dados de localização, do gerenciador de dados ou diretamente dos componentes de sensoriamento especialista. O módulo publicar-assinar pertence a representação deste componente, operando em conjunto com o sorvedouro e o componente de notificação presente no servidor (*broker*), para prover escalabilidade e heterogeneidade dos dados deste sistema.

Gerenciamento de energia. A partir do momento que o ciclo de sensoriamento e transmissão estiver completado, um componente de gerenciamento de energia pode hibernar o sensor com objetivo de economizar energia. As eventuais configurações do período de hibernação devem ser compatíveis com o período de leitura do componente de sensoriamento especialista e também entre os demais sensores da rede local, com intuito de manter o sincronismo de conectividade entre os nós.

Figura 17: Arquitetura da aplicação do sorvedouro

Fonte: Autor

Como dito, similarmente aos nós sensores, também o sorvedouro, além do sistema operacional e das camadas de rede, irá dispor de componentes na camada da aplicação que irão apoiar a operação distribuída e flexível do *middleware*, figura 17. As funcionalidades básicas esperadas do *middleware* para o sorvedouro são a **comunicação com a rede de sensores**, **comunicação com o servidor**, **gerenciamento do sorvedouro** e **gerenciamento de regras**.

Gerenciamento de regras. Analogamente ao nó sensor, considera-se também que no sorvedouro exista um componente que dissemine as configurações aos demais componentes, partindo das atualizações do servidor. Um módulo de segurança neste componente pode garantir que apenas os dados enviados pelo servidor sejam aderidos e posteriormente processados, de modo que, se necessário, as regras atuais sejam substituídas e transformadas novamente em configurações dedicadas aos demais componentes do sorvedouro.

Comunicação rede sensores. Este componente representa a interface com os nós sensores da rede local, ao qual o sorvedouro está gerenciando. O protocolo publicar-assinar do lado da rede de sensores, pertencente a este componente, normalmente está permanentemente disponível para recebimento dos pacotes enviados pelos diversos sensores. Eventualmente, um módulo de configurações dedicadas pode estar presente, por exemplo, para manter as especificações sobre o tempo de conexão ativa com um nó sensor (HUNKELER, TRUONG e STANFORD-CLARK, 2008).

Comunicação com servidor. Este componente representa o canal de comunicação entre o sorvedouro e o servidor. É por esta representação onde a tradução dos dados da rede de sensores para o servidor é realizada, por meio do módulo publicar-assinar. Além dos dados compilados na rede de sensores local, espera-se também por este componente, o tráfego das atualizações enviadas pelo servidor.

Gerenciamento do sorvedouro. Considera-se a presença de um componente de gerenciamento, que contenha diversos módulos, para lidar com a integridade e eficiência da sub rede administrada pelo sorvedouro. Um exemplo, é presença de um módulo de autenticidade que verifique se o pacote recebido é de um sensor válido. A persistência de dados é outro módulo importante, que pode estar presente neste componente, com a função de armazenar além da informação de quais são os sensores válidos na rede local, também os dados recebidos pelos diversos sensores que o sorvedouro administra, durante a fase de agregação dos dados. A agregação dos dados é ainda outro exemplo de funcionalidade que pode estar presente no componente de gerenciamento, a qual pertence à estratégia de economia de recursos do sistema (HUNKELER, TRUONG e STANFORD-CLARK, 2008). As novas regras

de funcionamento da rede de sensores gerenciada, assim que recebidas por um módulo de configuração, podem ser processadas e disponibilizadas para disseminação a rede local, pelo módulo de gerenciamento dos nós.

Já pelo lado da rede *crowdsensing*, presume-se que o dispositivo do participante terá múltiplas funções concomitantemente em execução, logo, a aplicação deve operar em sintonia com esta utilização compartilhada de recursos do dispositivo (GANTI, YE e LEI, 2011). Também alguns componentes do *middleware* são esperados na aplicação do participante, são eles a **interface do usuário**, **preferências**, **comunicação**, **coordenação** e o **gerenciador de tarefas**. A figura 18 ilustra uma arquitetura com a organização destes componentes.

Figura 18: Arquitetura da aplicação *crowdsensing*

Fonte: Autor

Interface do usuário. Este componente representa a interface gráfica e de toque do usuário. Além da interface gráfica inerente a tarefa, outras visualizações podem estar presentes para despertar o interesse do participante, por exemplo, um mapa com a posição geolocalizada dos relatórios já emitidos, o próprio histórico dos relatórios e também gráficos com sumários estatísticos. Considerando a atividade fim

da aplicação *crowdsensing*, neste caso a geração de relatórios, a interface do usuário pode habilitar a personalização da aplicação e dos relatórios, o que incentiva o engajamento ao uso do sistema pelo participante (CHRISTIN, REINHARDT, *et al.*, 2011).

Preferências. Pondera-se também a presença de um componente de preferências que modifique as funcionalidades da aplicação a partir da interação do participante, em prol das escolhas pessoais. Espera-se como as principais interações neste componente a seleção do tipo de tarefa, a escolha da imagem captada, partindo da câmera do dispositivo e as recomendações inerentes da tarefa. Assim, em um exemplo sobre uma tarefa de relatos de buracos em via pública, estas recomendações do participante podem contemplar o tamanho do buraco e o tipo de urgência. Além da aplicação *crowdsensing* proporcionar ao participante escolher as tarefas de sensoriamento e as opções disponíveis inerentes a tarefa, pode incluir também as configurações personalizadas de privacidade. As customizações de anonimato do participante ou até mesmo, quando aplicável, generalizando a localização, por exemplo, identificando a fonte da informação por um bairro ao invés da posição específica do GPS, são opções possíveis de preferências neste componente.

Comunicação. Transmitir os dados coletados pelos cidadãos e receber as notificações do servidor são as funções esperadas nesta representação de componente. Também na aplicação *crowdsensing* é importante um módulo publicar-assinar neste componente, para permitir que o sistema de gerenciamento da cidade se mantenha escalável e o tratamento da mensagem seja padronizado com a rede de sensores.

Gerenciador de tarefas. Este componente representa o mecanismo de orquestração das tarefas, recebendo as preferências do usuário, executando a coleta de dados dos sensores no dispositivo e encaminhando estas informações como os relatos do participante para a transmissão. Nesta representação, as configurações personalizadas definidas pelo participante, por exemplo, ilustradas por um JSON (*JavaScript Object Notation*), parametrizam quais sensores do dispositivo efetivamente são usados e no caso do sensoriamento oportunista, também qual é a frequência de medição e transmissão. O módulo de sensores realiza a seleção e leitura do sensor definido pela tarefa. Já o georreferenciamento da tarefa pode ser

provido pelo módulo de localização. Prevendo existir circunstâncias onde não haja momentaneamente comunicação com o servidor, as medições podem ser armazenadas temporalmente em um módulo de persistência de dados para a posterior transmissão.

Coordenação. Este componente representa as funcionalidades na aplicação *crowdsensing* de receber pelo servidor, tarefas específicas e incentivos para colaboração do participante. Como será detalhado na próxima seção, o *middleware* presente no servidor tem o papel de coordenar tarefas de sensoriamento. Por exemplo, o participante pode receber por este componente uma mensagem indicando que uma determinada tarefa, próxima ao local do último relatório do participante, é mais efetiva no sistema. Assim, para aumentar a possibilidade de adesão a esta tarefa pelo participante, mecanismos de incentivos podem sugerir uma contrapartida de incremento de reputação ao participante ou algum outro tipo de bonificação se a tarefa sugerida for cumprida. Para garantir a confiabilidade da informação encaminhada pelo servidor, espera-se também presença de um módulo de segurança neste componente.

5.4 Características do *middleware* no servidor

A aplicação do servidor, representada na figura 19, cumprirá as principais tarefas do *middleware* estudado neste trabalho. Estas tarefas são descritas nesta seção como componentes e serviços reutilizáveis do sistema de gerenciamento. Desta forma é considerado que os serviços de **gerenciador de eventos, coordenador, notificações, relatórios, atualizações e gerenciador de bancos de dados** e os componentes comuns aos demais serviços, como os modelos **analíticos e regras**, estejam presentes no servidor. Também é previsto que aplicação seja instalada em uma infraestrutura de computação em nuvem, devido às características favoráveis a escalabilidade do sistema, deste modo, o sistema operacional do servidor e as camadas de rede abaixo da camada de aplicação são transparentes para o gestor do sistema.

Figura 19: Arquitetura da aplicação do Servidor – Serviço de Computação em Nuvem

Fonte: Autor

Gerenciador de eventos. Este serviço cumpre funções importantes neste sistema, sendo responsável por receber todo o fluxo de dados de entrada e saída, gerenciar os tópicos das mensagens e também realizar o pré-processamento das mensagens conforme a regra específica do tópico. O gerenciador de eventos, além de compartilhar dos componentes de modelos analíticos e de regras, é composto do **broker** e de um componente de **segurança**, figura 20. O *broker* representa o serviço de notificação do paradigma de comunicação publicar e assinar (EUGSTER, FELBER, *et al.*, 2003) e tem a função de tornar o *middleware* adaptável à oferta variável de sensores e participantes. Instalando este componente em uma IaaS, aumenta-se ou diminui-se o poder de processamento ou a quantidade de armazenamento das mensagens pela demanda de utilização. Conforme apresentado na seção 4.4, este tipo de componente torna a coleta de dados assíncrona, favorecendo escalabilidade do sistema. O componente de segurança garante a confiabilidade dos dados para o processamento do servidor.

Figura 20: Componentes do gerenciador de eventos

Fonte: Autor

No gerenciador de eventos todas as mensagens recebidas são de eventos ou alertas gerados pelos sensores e dispositivos móveis dos participantes. Como mencionado, as informações são processadas conforme as regras dos tópicos. Por exemplo, um tópico pode conter como regra que as mensagens encaminhadas a ele devem ser armazenadas no banco de dados de mensagens críticas, para receber posterior tratamento especial pelos demais serviços. Uma situação plausível a este exemplo, pode ser um sensor acusando temperatura acima do normal.

Gerenciador de bancos de dados. O serviço de gerenciamento do banco de dados administra diversos bancos de dados com funções exclusivas. Esta separação é conveniente considerando que alguns destes, pela própria natureza da informação, pode conter um grande volume de dados e frequentes acessos, como é previsto que seja o banco para armazenamento das mensagens comuns dos sensores, diferentemente do banco de dados para armazenar as configurações, por exemplo, com as características dos tipos de sensores pertencentes na rede, ou com as tarefas de *crowdsensing*. Para os participantes que desejarem identificar-se no sistema, existe um banco de dados exclusivo para manutenção dos eventuais dados sensíveis. Um banco de dados exclusivamente importante é utilizado para manter as mensagens críticas providas pelos sensores ou dos participantes. Estas informações precisam de destaque e eventualmente tomadas de decisões rápidas pelos gestores do sistema. Mesmo o banco de dados de mensagens comuns tem a importante função da coleta de dados para a criação de séries históricas, o que é útil para gerenciamento do serviço da cidade ao longo dos meses.

Coordenador. O coordenador é o serviço que busca por eventos extraordinários que podem ser comprovados por ambos os sistemas de *crowdsensing* e da rede de

sensores. Por exemplo, o coordenador pode identificar uma transação que necessite a transmissão de uma mensagem a um participante, solicitando ao participante a comprovação da veracidade dos dados de um sensor, neste exemplo, considerando a localização do participante e do sensor. Outra responsabilidade do coordenador é a chamada automática do serviço de notificações. No exemplo anterior, o coordenador pode utilizar a transação de confirmação do participante para notificar um organismo específico de gestão da cidade.

Notificações. O serviço de notificação tem o objetivo predominante de alertar os gestores do sistema sobre situações extraordinárias com os serviços monitorados da cidade, a partir dos dados coletados pela rede de sensores e dos participantes.

Estes alertas são concretizados através de notificações na aplicação e pode ser representado por ênfases na visualização de dados, registro de erros, *e-mails*, etc. Uma vez que o serviço de notificação faz uso compartilhado do componente de regras, é possível customizar regras pelo gestor para avisar os participantes sobre assuntos não relacionados aos dados coletados, por exemplo, vinculados a campanhas promocionais na cidade.

Atualizações. Este serviço controla as atualizações dos *softwares* instalados na rede de sensores e também nas aplicações *crowdsensing*. Em um sistema voltado a cidades, a ação de acrescentar, excluir ou aperfeiçoar funcionalidades na monitoração dos serviços pelas atualizações nos diversos dispositivos, precisa ser executada por processos automatizados e gerenciada de modo centralizado, evitando a interpretação de informações inconsistentes devido a dispositivos com versões diferentes. Além disso, o organismo gestor da cidade pode usar este serviço para administrar as aplicações *crowdsensing* disponíveis ou fomentar constantes modificações nestas aplicações, induzindo o dinamismo na cidade.

Relatórios. O serviço de relatórios tem o papel de apresentar a informação presente no banco de dados. Podem operar também em conjunto com os modelos analíticos de modo a apresentar dados que facilitem o suporte a decisão.

Analíticos. São os modelos analíticos que provêm a inteligência do *middleware*. Descobertas do que está ou irá ocorrer no serviço da cidade, seguindo

os padrões de comportamentos históricos dos dados de sensores e relatos colaborativos dos cidadãos, são valiosos para suporte a decisão dos gestores públicos, por exemplo, prevendo calamidades ou auxiliando a manutenção dos serviços. Na arquitetura do *middleware* deste trabalho, os modelos analíticos são utilizados como um componente comum aos diversos serviços, facilitando a reutilização.

Regras. As regras são as características pré-estabelecidas pelo gestor, para ações automáticas do *middleware* e são construídas em um componente comum aos serviços do *middleware*. Considera-se que podem estar sujeitas a subjetividade pelos gestores, de modo que precisam estar disponíveis para edição.

Interface de Programação. Prover que a informação coletada seja acessada aos gestores do sistema é a função da interface. É a interface de programação que possibilita a aplicação reutilizar os serviços do *middleware*.

5.5 Cenário de aplicação

Como observados nos trabalhos de ERIKSSON, et al. (2008), MOHAN, PADMANABHAN e RAMJEE (2008) e KOCK, et al. (2013), a identificação de buracos em vias públicas é uma prática já investigada no desenvolvimento de cidades inteligentes, realizada tanto por técnicas de *crowdsensing* (MOHAN, PADMANABHAN e RAMJEE, 2008), (KOCK, CARDONHA, et al., 2013) quanto por sensoriamento automático (ERIKSSON, GIROD, et al., 2008).

No trabalho desenvolvido por ERIKSSON, et al. (2008), nós sensores especializados equipados com acelerômetros e GPS foram instalados no interior de veículos para a detecção de buracos. Similarmente a este trabalho, porém ao invés de um sensor especializado, um *smartphone* com acelerômetro embutido foi utilizado para a identificação de buracos (MOHAN, PADMANABHAN e RAMJEE, 2008). Os autores KOCK, et al. (2013), estudaram a utilização de um aplicativo de *smartphone* para o cidadão denunciar ao órgão público apropriado buracos nas vias, utilizando uma foto do buraco georreferenciada selecionada pelo participante.

Um dos principais problemas constatados das técnicas de identificação automática dos buracos nas vias públicas propostas por ERIKSSON, et al. (2008) e MOHAN, PADMANABHAN e RAMJEE (2008) é o falso positivo. Já a técnica de *crowdsensing* que utiliza fotos georreferenciadas apresentada por KOCK, et al. (2013), é contestada pela imprecisão da informação repassada pelo cidadão (fotos fora de foco, de baixa resolução, pouca luminosidade, etc.) e ainda com um contexto subjetivo, por exemplo, se o buraco relatado tem a relevância para acionar os mecanismos de conservação da cidade.

Neste trabalho será utilizada a identificação de buracos nas vias públicas como forma de demonstração do modelo de *middleware* direcionado a cidade, apresentado nesta pesquisa. Representando o serviço de sensoriamento na cidade, será utilizado em um veículo um nó sensor com acelerômetros em conjunto com GPS. Pondera-se que este tipo de nó sensor pode ser utilizado em veículos públicos como ônibus e caminhões de lixo, o que de fato já foi descrito em projetos anteriores (DE ZOYSA, KEPPITIYAGAMA, et al., 2007). Representando os dados provindos por *crowdsensing*, será usado o aplicativo Rota Acessível¹⁶ adaptado para este trabalho.

5.6 Experimento

Para prova de conceito, representando o sensoriamento automatizado por rede de sensor, foi construído um nó sensor multifuncional e instalado em um veículo de passeio. Este nó sensor coletava a aceleração do veículo nos eixos X, Y e Z e também a posição georreferenciada do carro no mesmo instante. Os dados eram transmitidos para um nó sorvedouro quando o veículo estava dentro do alcance de comunicação da rede sem fio. Após o recebimento dos dados pelo nó sorvedouro, localizado em uma garagem, estes eram repassados pela rede local cabeada à uma instância de computação em nuvem.

No mesmo experimento, uma aplicação móvel representando o sistema de *crowdsensing* foi desenvolvida. O participante conseguia relatar um buraco de rua bastando apenas tirar uma fotografia do ponto em questão pelo *smartphone* e então

¹⁶ <http://www.rotaacessivel.com/>

enviá-la, na própria aplicação, como um relatório. Considerando que a customização do relatório pode favorecer o engajamento do participante para utilização da aplicação (CHRISTIN, REINHARDT, *et al.*, 2011) e maximizar a precisão do relatório, a opinião sobre o tamanho do buraco e a criticidade requerida para atenção do gestor público eram coletadas pelo participante no mesmo relatório. A posição georreferenciada do participante também era coletada automaticamente pela aplicação móvel. Por fim, os dados do relatório eram enviados a mesma instância de computação em nuvem usada pela rede do sensor. Na figura 21 está representado como foi construído o experimento.

Figura 21: Ilustração do experimento

Fonte: Autor

O nó sensor montado no veículo foi elaborado a partir de uma placa de desenvolvimento do fabricante Arduino, modelo Uno¹⁷, a qual executava o papel de CPU (Central Processing Unit) do sensor.

Neste sensor, o módulo responsável pela captura da aceleração do veículo era um MEMS do fabricante Freescale, modelo MMA7361¹⁸, já adaptado para montagem

¹⁷ <http://arduino.cc/en/Main/ArduinoBoardUno>

¹⁸ http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf

na placa de desenvolvimento Arduino. Entre as características deste módulo estão a compensação de temperatura, a sensibilidade de operação de $\pm 1.5g$ (indicado também pelo fabricante para a correção de estabilidade em imagens no celular, para parar o acesso ao disco rígido durante uma queda em *notebooks*, como contador de passos, etc.) e também a compensação de referência da aceleração zero, o que evita ao usuário procedimentos prévios de calibração do sensor a uma superfície plana.

Figura 22: Vista do *hardware* que compõe o nó sensor e a montagem final

Fonte: Autor

O módulo GPS captava a posição georreferenciada do veículo. No experimento foi utilizado do fabricante Globalsat, modelo EB365¹⁹, já adaptado à montagem na placa de desenvolvimento Arduino. As principais características deste módulo GPS é a capacidade de gerenciar a recepção de até 20 satélites, sensibilidade de -142dBm e a precisão de 2,5 metros segundo o fabricante. Adicionalmente, integrado a este módulo foi utilizada uma antena de GPS para o segmento automotivo do fabricante Amphenol, modelo GAN30043²⁰ com ganho de 28dB.

Suportando as camadas física e de enlace do padrão IEEE 802.15.4 responsáveis à comunicação sem fio do sensor ao sorvedouro, foi utilizado o módulo

¹⁹http://www.globalsat.com.tw/products-page.php?menu=4&gs_en_product_id=4&gs_en_product_cnt_id=64

²⁰ <http://amphenolafrica.com/cats/GPS.pdf>

XBee PRO S1²¹ do fabricante Digi. Destaca-se neste modelo, a potência de 63mW na transmissão e 100dBm de sensibilidade na recepção, possibilitando um alcance de comunicação em visada direta de até 1,6 km. Uma placa com conectores genéricos do fabricante Seeed Studio, modelo XBee Shield V2²² foi acrescentada para facilitar as conexões entre o módulo de comunicação, o módulo do acelerômetro e a CPU Arduino. A figura 22 demonstra os diversos *hardwares* descritos que compuseram o sensor e a montagem final.

Figura 23: Vista do *hardware* do sorvedouro e a montagem final

Fonte: Autor

Similarmente ao nó sensor, o sorvedouro foi desenvolvido a partir de uma CPU do fabricante Arduino, porém no sorvedouro utilizou-se o modelo Mega²³. Para a comunicação sem fio com o nó sensor montado no veículo, o mesmo modelo do módulo IEEE 802.15.4 usado no nó sensor foi utilizado, o XBee PRO S1 do fabricante Digi. O enlace com o servidor do sistema, era provido por uma PCI Ethernet, também do fabricante Arduino²⁴. Com o intuito de armazenar localmente os dados recebidos pelo nó sensor antes de enviar ao servidor do sistema, um cartão SD de 1G Bytes

²¹ <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/po-int-multipoint-rfmodules/xbec-series1-module#specs>

²² http://www.seeedstudio.com/wiki/XBee_Shield_V2.0

²³ <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

²⁴ <https://www.arduino.cc/en/Main/ArduinoEthernetShield>

realizava a persistência destes dados, disponibilizado pela entrada micro SD da PCI Ethernet. Adicionalmente uma PCI de Conexões do fabricante GEEETECH, modelo XBee Shield V3²⁵ foi utilizada para facilitar as conexões entre o módulo de comunicação sem fio e a PCI Ethernet. A figura 23 demonstra os diversos itens descritos que compuseram o sorvedouro.

Figura 24: Montagem do sensor no veículo

Fonte: Autor

O veículo utilizado para o experimento foi do fabricante Honda, modelo CIVIC²⁶, ano de fabricação 2013. Montado no compartimento do porta luvas, o sensor foi alimentado por uma conexão USB disponível da central multimídia do veículo. Instalou-se a antena do GPS fora do porta luvas, no vidro do para-brisa frontal. A figura 24 apresenta a montagem do sensor no veículo.

A aplicação *crowdsensing* foi criada no ambiente de desenvolvimento Eclipse²⁷, com auxílio da plataforma IBM MobileFirst²⁸, instalada como um módulo adicional no Eclipse. Afim de não restringir participantes pelo tipo de sistema operacional

²⁵ <http://www.geeetech.com/zigbeexbee-module-sheild-v30-p-565.html>

²⁶ <http://www.icarros.com.br/honda/civic/2013/ficha-tecnica>

²⁷ <http://www.eclipse.org/>

²⁸ <http://www.ibm.com/mobilefirst/us/en/>

pertencente ao *smartphone*, por exemplo, IOS²⁹ (*iPhone Operating System*), Android³⁰ ou Windows Phone³¹, este aplicativo foi construído com tecnologias de aplicações móveis híbridas (WASSERMAN, 2010). Para esta aplicação, a linguagem de programação utilizada foi o Javascript (ECMA INTERNATIONAL, 2011), sendo complementada com as bibliotecas abertas do projeto PhoneGap³² para integração dos recursos nativos do *smartphone*, por exemplo, o acesso as funções do GPS e da câmera fotográfica. O HTML 5 (*HyperText Markup Language*) foi usado para marcação de texto e elementos da visualização (HICKSON, BERJON, *et al.*, 2014) e o CSS (*Cascading Style Sheets*) para a definição do estilo destes elementos de visualização (BOS, CELIK, *et al.*, 2011). Com esta estratégia de desenvolvimento de aplicativos móveis híbridos, usando as bibliotecas PhoneGap para acesso aos recursos do aparelho, é possível a construção de aplicativos baseados nas funcionalidades de navegadores de internet (*webkits*) e deste modo, compatível com a maioria dos sistemas operacionais para dispositivos móveis atualmente existentes (IBM CORPORATION, 2012).

Conforme sugerido na arquitetura da aplicação *crowdsensing* na seção 5.4, um componente de gerenciamento de tarefas foi implantado. Este componente era responsável por construir a aplicação móvel dependendo da configuração desejável pelo gestor. Por exemplo, o gestor poderia definir as possibilidades de interações que o participante teria acesso e quais os tipos de contextos que ele queria receber relatos do participante. Neste experimento foi elaborado uma aplicação denominada “Cidades Conscientes”, a qual o participante pode reportar problemas como buracos em vias públicas, mas é possível ser acrescentados outros cenários como problemas na calçada, falha na iluminação pública, poda de árvore, etc. Com esta estratégia é possível o gestor escolher outros contextos para aplicação móvel, por exemplo, receber a opinião de participantes sobre eventos culturais disponíveis na cidade. A presença deste serviço de gerenciamento de tarefas em uma aplicação *crowdsensing*, devido a possibilidade da aplicação assumir diversas finalidades, favorece a longevidade e a escalabilidade por conta da diversidade de aplicações possíveis (GALLO, CARDONHA, *et al.*, 2014).

²⁹ <https://www.apple.com/ios/>

³⁰ <https://www.android.com/>

³¹ <http://www.windowsphone.com/pt-br>

³² <http://phonegap.com/>

Um módulo de mapa foi incluído para permitir ao participante visualizar a localização do seu relatório e também dos demais relatórios vigentes enviados por outros participantes. Além disso, quando necessário, o mapa indica a localização de pontos que requerem a atenção do participante, por exemplo, quando é solicitado ao participante uma tarefa de sensoriamento em um ponto específico da cidade.

Também foi incluído um módulo de histórico para aprimorar a experiência do participante ao uso da aplicação. Neste módulo é possível visualizar os relatórios enviados com sucesso e também aqueles que, por problemas de conectividade do participante no momento do envio do relatório, ainda não foram emitidos. Outra função deste módulo é a visualização do relatório gerado pelo participante, contendo a fotografia tirada e também as escolhas selecionadas.

Figura 25: Telas da aplicação *crowdsensing*: tela de apresentação, tela de contextos de interação disponíveis e tela de elaboração de relatórios

Fonte: Autor

A figura 25 e 26 mostram a interface gráfica pertencente a aplicação *crowdsensing* criada para o experimento. Na figura 25 estão destacadas a tela de apresentação, a tela com a lista de contextos possíveis para interação do participante, neste caso, “Problemas na calçada” e “Buracos na rua” e também a tela de captura de relatórios, a qual o participante após capturar a imagem do contexto inspecionado,

relata a sua percepção do problema, por exemplo, informando o tamanho do buraco e a sugestão de criticidade para atenção do gestor público. Na figura 26 está a representação da tela de histórico, a qual indica quais relatórios foram enviados com sucesso e aqueles que devido a falha de conectividade momentânea ainda não foram transmitidos ao servidor. Também na figura 26 está a tela com o mapa de navegação, que indica as posições coletadas dos relatórios vigentes e a posição atual do participante e por fim, a mensagem de notificação característica recebida pelo participante após ele enviar com sucesso um relatório e ao mesmo tempo, estar próximo a um ponto que necessita ser auditado pelo participante.

Figura 26: Telas da aplicação *crowdsensing*: Tela de histórico, alerta característico para auditoria da informação dos sensores e mapa de navegação

Fonte: Autor

O *middleware* montado para o experimento foi construído conforme a arquitetura proposta na seção 5.2. Ele foi instalado em uma infraestrutura de computação em nuvem (IaaS) do fabricante SoftLayer³³ contendo 4 núcleos, 4 GBytes de memória RAM e 100 GBytes de espaço em disco rígido, com o sistema operacional Linux Centos 7³⁴. Nesta instância foi integrado um servidor de aplicação, um banco de dados e um servidor de mensagens, conforme está representado na figura 27.

³³ <http://www.softlayer.com/our-platform>

³⁴ <http://www.centos.org/>

Figura 27: Infraestrutura de programação integradas no servidor para a construção do *middleware*

Fonte: Autor

O servidor de aplicação foi construído a partir do IBM Websphere Application Server 8.5.5³⁵. Além de prover as páginas WEB, também suporta a infraestrutura de operação dos *servlets* (HUNTER e CRAWFORD, 2001), empregados pelas diversas tarefas específicas do *middleware*. O servidor neste experimento também hospedou o IBM MobileFirst Platform Foundation Server³⁶, que executa partes do serviço de atualização do *middleware* proposto como a distribuição e atualização automática da aplicação *crowdsensing*.

No experimento, os dados foram armazenados de modo estruturado em um MER (Modelo de Entidade e Relacionamento) utilizando o programa de banco de dados IBM DB2 9.7 Enterprise³⁷. Na figura 28, encontra-se a visão geral das tabelas empregadas e o respectivo banco de dados ao qual foram instaladas. Como mencionado na seção 5.2, a divisão das tabelas por tipo de banco se faz necessária devido as características individuais dos dados. Desta forma é possível definir estratégias distintas de segurança, resiliência, escalonamento, armazenamento e processamento de dados.

³⁵ <http://www-03.ibm.com/software/products/en/appserv-was>

³⁶ <http://www-03.ibm.com/software/products/pt/mobilefirstfoundation>

³⁷ <http://www-01.ibm.com/software/data/db2/>

Figura 28: Modelo de dados e representação da alocação das tabelas no respectivo banco

Fonte: Autor

Neste experimento, o **banco de mensagens comuns** foi composto pela tabela “*Pothole*”, a qual recebe dados com granularidade de 1 segundo enviado pelo sensor montado no veículo. Devido a este tipo de carga de dados, este banco de dados pode necessitar de capacidade de armazenamento diferenciado. Outro exemplo de necessidade diferenciada de recursos é o **banco de mensagens críticas**. Ele foi desenvolvido com as tabelas “*Marked Pothole*” a qual armazena a classificação do ponto analisado pelo sensor (neste caso, indicando se o ponto da rua está em boas condições, ondulado e principalmente com buraco) e a tabela “*Citizen Reports*”, que é caracterizada pelos relatórios enviados dos participantes. Devido a estas características dos dados o banco de mensagens críticas pode receber recursos específicos para aprimoramento da resiliência de processamento. Já o **banco de dados sensíveis** eventualmente será exclusivo pelos sistemas de segurança nele suportados. Neste experimento, recebeu a tabela “*Participants*” com informação de dados pessoais de participantes que voluntariamente quiseram ser identificados no sistema através de apelidos específicos. No anexo 9.6 estão disponíveis o tipo construtivo e a função de cada campo das tabelas empregados no experimento.

Figura 29: Detalhes construtivos e de integração entre o servidor de mensagens, o servidor de aplicação e o banco de dados

Fonte: Autor

O **serviço de eventos** do *middleware* foi construído a partir de um servidor de mensagens IBM Websphere Message Queuing 7.5³⁸. Foi criado tópicos independentes para as mensagens dos dados provenientes da rede de sensores e para a aplicação *crowdsensing*. Esta estratégia se faz necessário visto que o volume de mensagens da rede de sensores é muito maior do que os relatórios enviados pelos participantes. Desta forma é possível alocar recursos específicos para cada tópico de mensagens, por exemplo, mais espaço em disco ou até mesmo uma instância de IaaS própria.

A figura 29 apresenta os detalhes construtivos e de integração entre o servidor de mensagens, o servidor de aplicação e o banco de dados. Neste experimento o serviço gerenciador de eventos foi construído pelo servidor de mensagens e os *servlets* presentes no servidor de aplicação. As conexões JMS (*Java Message Service*) entre o servidor de mensagens e o servidor de aplicação habilitavam o

³⁸ <http://www-03.ibm.com/software/products/en/ibm-mq>

tratamento das mensagens entre estes servidores (HAPNER, BURRIDGE, *et al.*, 2002). Já o serviço **gerenciador do banco de dados** era composto pelo banco de dados propriamente dito e também por *servlets* no servidor de aplicação. Similarmente as conexões JMS, foi construído conexões JDBC (*Java Database Connectivity*) entre o banco de dados e o servidor de aplicações, permitindo aos *servlets* enviar comandos SQL (*Structured Query Language*) para o banco de dados (REESE, 2000).

Figura 30: Exemplo de configurações iniciais da aplicação *crowdsensing*

Exemplo de formato das configurações iniciais da aplicação

```
{
  "categories": [
    {
      "id": 1,
      "name": "Problema na Calçada"
    },
    {
      "id": 2,
      "name": "Buraco em Rua"
    }
  ],
  "description": "Colabore para tornar a sua cidade mais acessível e inclusiva",
  "id": 1,
  "name": "Cidades Conscientes",
  "version": 1.0
}
```

Fonte: Autor

No gerenciador de eventos, o tópico *Sensors*, presente no servidor de mensagens, era responsável por receber os dados publicados pelo *gateway*. Logo, este tópico processava os dados gerados no sensor montado veículo, com a granularidade de 1 segundo (aceleração XYZ, velocidade, latitude, longitude, data e hora). O processamento deste tópico era finalizado pela gravação dos dados no banco de mensagens comuns, provida pela interface *post Sensors*, do serviço gerenciador de banco de dados. De modo análogo, o tópico *Report*, presente no servidor de mensagens, recebia os relatórios publicados pela aplicação do dispositivo móvel do participante contendo, por exemplo, os relatórios sobre os buracos encontrados. Porém, para esta funcionalidade, havia duas interfaces no serviço do gerenciador do banco de dados, sendo uma delas (*Report*) para a gravação do conteúdo de texto do relatório em formato JSON (tamanho, criticidade, data, hora, latitude, longitude) e a outra (*Images*) especificamente para guardar a fotografia enviada junto com o relatório. Neste experimento, o tópico *Config* era assinado pelos dispositivos dos participantes para receber os parâmetros iniciais das aplicações como as categorias

e atributos existentes. Um exemplo de configuração da aplicação enviada pelo servidor a aplicação *crowdsensing* pode ser visualizada na figura 30.

Figura 31: Detalhes construtivos e de integração do servidor de aplicação e o banco de dados, com ênfase no serviço de relatórios e gerenciador do banco de dados

Fonte: Autor

A figura 31 representa os aspectos construtivos entre o banco de dados e o servidor de aplicação, similarmente a figura 29. Porém a ênfase nesta representação está no relacionamento do **serviço de relatórios** com o serviço do gerenciador de banco de dados. Neste experimento o serviço de relatório foi construído totalmente no servidor de aplicação e foi composto pelas seguintes interfaces:

- ***Path (get)***. Apresenta os dados coletados pelos sensores a partir de uma rota selecionada.

- ***Classified Path (get)***. Retorna os pontos já classificados, partindo da escolha de uma rota e uma sugerida classificação. Esta interface, por exemplo, informa que na rota 'x' existe 'n' pontos que foram classificados como buracos.

- ***Crowdsensing (get)***. Lista os relatórios enviados pelos participantes até o momento.

- **Image (get).** Recebe a imagem que um determinado relatório gerou.

O serviço de relatórios, como mencionado na seção 5.5, tem o papel de apoiar as visualizações dos dados armazenado no banco de dados e desta forma pode ser acrescido de outras interfaces conforme a necessidade de análise dos dados. Embora não construído neste experimento é possível operar junto com modelos analíticos para interpretação de introspecções significativas ao gestor do sistema. Os detalhes das interfaces REST construídas estão descritas no apêndice G.

Figura 32: Detalhes construtivos e de integração do servidor de aplicação e o banco de dados, com ênfase no serviço de relatórios e gerenciador do banco de dados

Fonte: Autor

A figura 32 destaca os aspectos construtivos entre o banco de dados e o servidor de aplicação, similarmente as figuras 29 e 31, entretanto a ênfase nesta figura está no relacionamento do **serviço coordenador** com o serviço do gerenciador de banco de dados. A interpretação dos dados armazenados e a tomada de decisões automatizadas baseadas em regras pré-estabelecidas pelo gestor, são funções do serviço coordenador. Neste experimento, o coordenador, a partir de uma solicitação da interface *post Analisis* com a indicação de uma rota, executava a análise dos dados coletados pelo sensor no veículo. Este processamento utilizava as funcionalidades do **módulo de modelos analíticos** para padronizar a amostragem das medições (*Sampling*) e realizar a classificação do ponto coletado (*Classification*).

Esta etapa era concluída pela persistência dos dados através da interface *post Classified Points*. Exemplificando o processamento de ações baseadas em regras pré-estabelecidas pelo coordenador a interface *get Alert*, retornava o ponto identificado como buraco mais próximo a posição do participante. Este processamento considerava as informações de latitude e longitude presentes nos metadados do relatório emitido pelo participante e as regras da área de procura definida no **módulo de regras**.

Figura 33: Detalhes construtivos e de integração do servidor de aplicação e o banco de dados, com ênfase no serviço de notificações

Fonte: Autor

A figura 33 também mostra os aspectos construtivos entre o banco de dados e o servidor de aplicação, similarmente as figuras 29, 31 e 32, porém enfatizando o **serviço de notificações**. Foi desenvolvido uma interface *post Set Search Area* com o intuito testar a inserção de regras customizadas para o *middleware*. Neste caso, opera com uma regra sobre a área de procura ao qual o serviço coordenador utiliza para tomada de decisão automatizada, conforme mencionado na figura 32. No lado

direito da figura 32 está o fluxo de dados, por parte do serviço de notificações, quando recebe uma solicitação do serviço coordenador sobre uma notificação a ser publicada. Este serviço está conectado aos provedores de mensagens externos, neste experimento ao *Cloud Message for Android*³⁹ ou *Apple Push Notification*⁴⁰, resultando em mensagens de alerta no dispositivo móvel do usuário, independentemente se a aplicação *crowdsensing* ainda está ativa.

A figura 34 mostra a visão geral das interfaces utilizadas neste experimento. Os detalhes sobre as funcionalidades de cada interface são apresentados no apêndice G.

Figura 34: Visão geral das interfaces utilizadas no experimento

Fonte: Autor

Os modelos analíticos utilizados, implementados em *servlets*, foram elaborados a partir do conjunto de bibliotecas para inteligência artificial WEKA⁴¹. Um modelo analítico especial contemplou a classificação dos dados de aceleração enviados pelos sensores, identificando quais pontos na via pública existe um possível buraco, qual

³⁹ <https://developers.google.com/cloud-messaging/>

⁴⁰ <https://developer.apple.com/notifications/>

⁴¹ <http://www.cs.waikato.ac.nz/ml/weka/>

parte da via está irregular ou em boas condições e até mesmo, quais pontos o veículo parou. Outro modelo analítico utilizado interpreta pela distância euclidiana o provável buraco de rua mais próximo para a certificação de um participante, a partir da recepção da localização do relatório do participante e de uma área de procura previamente estipulada pelo gestor.

5.7 Resultados

O experimento foi realizado durante 104 dias ininterruptos, entre o período de dezembro de 2014 a março de 2015.

Figura 35: Representação das rotas analisadas no experimento

Fonte: Autor

O veículo utilizado no experimento percorreu uma distância pouco superior a 6000 km. Os traslados ocorreram principalmente no trajeto do trabalho para a casa, em torno de 80 ciclos de ida e volta, com distância de 35 km para cada ciclo e certa variabilidade da rota. Normalmente com duração de 45 minutos em cada trecho. Esporadicamente ocorreram viagens, com traslados superiores a 100 km, o que se verificou em aproximadamente duas dezenas de trajetos. Foi comum rotas não repetitivas para atendimentos das diversas necessidades do condutor voluntário,

principalmente próximo a sua residência, localizado no bairro de Santana, na cidade de São Paulo. A figura 35 representa todas as rotas coletadas durante o experimento na interface de visualização da aplicação WEB do sistema de gerenciamento. É possível constatar na escala de 5 km, uma concentração de rotas especialmente na zona norte da capital paulista e também na cidade de Guarulhos, particularmente na fronteira entre municípios.

No total, foi coletado 193 horas de dados durante o experimento. Este tempo era contabilizado começando no momento de partida do motor até o veículo ser desligado. Desta maneira, foi gerado pelo veículo 1011,2 M bytes. Em outras palavras, o nó sensor transferia 1,45 K bytes/segundo para o sorvedouro. Portanto, em média, eram processados 9,72 M bytes pela rede sensores ao sistema de gerenciamento por dia.

Para análise da transferência de dados entre o sorvedouro e o nó sensor, foram realizados testes considerando 1 hora de sensoriamento válido e amostragem de 120 medições. Estes testes geravam 5,4 M bytes. Como a comunicação da rede sem fio estava configurada a 56 Kbps de velocidade, a transmissão ocorria em 16 minutos com uma taxa de transferência de 5,5 K bytes/segundo.

Como dito, neste experimento o veículo circulava livremente durante a cidade, coletando dados sobre as vias e apenas quando estava próximo a residência onde estava instalado o nó sorvedouro, ocorria a transferência dos dados. Normalmente 1 vez por dia. Considerando apenas os ciclos de traslados entre a residência e trabalho, com 35 km de extensão, o sensor gerava por dia 3,5 M bytes. Assim, tipicamente, a transmissão dos dados pelo nó sensor no veículo ocorria durante 10 minutos. Esta taxa de transferência é devido a capacidade de transferência da rede IEEE 802.15.4.

Após a coleta dos dados do nó sensor foi identificado 225 rotas. Este volume de rotas está abaixo das 486 vezes que o motor do veículo foi ligado, identificado pelos dados contidos no cartão de memória do sensor. Este motivo se deu principalmente porque a aplicação do sensor não mantinha a tabela *almanac* do GPS. Portanto, toda vez que o sensor ligava precisava receber a tabela de constelação do GPS, o que normalmente ocorre a cada 12 minutos (HOFMANN-WELLENHOF, LICHTENEGGER e WASLE, 2007). Devido a esta implementação da aplicação do nó sensor, os dados dos pontos recebidos sem uma coordenada de latitude e longitude eram descartados

para classificação. Na prática, o veículo tinha que permanecer durante ao menos 12 minutos ligados em local aberto para gerar dados válidos para classificação. Esta situação não ocorreu em todas as rotas executadas. Também houve uma falha na antena do GPS entre os dias 7 e 16 de fevereiro, o que agravou a coleta de dados não-válidos para classificação neste período.

Figura 36: Características do conteúdo da mensagem do sensor

Fonte: Autor

A aplicação do nó sensor estava configurada para coletar a maior taxa de amostragem do acelerômetro, o que implicou no sistema de gerenciamento receber dados do sensor com taxas de amostragens diferentes. A figura 36 representa o conteúdo de um registro no banco de dados de mensagens comuns, gerado a partir de um pacote de dados típico transmitido pelo nó sensor. Neste exemplo, os valores sobre a data, hora, latitude, longitude e o número de satélites disponíveis, são gerados pelo GPS contido no nó sensor. Os valores sobre os eixos (X, Y e Z), são criados pelo acelerômetro. Já os dados sobre o número de amostras e o trajeto são gerados no sorvedouro. O ID é a referência do registro na tabela do banco de dados.

Ao receber uma mensagem diária, o *middleware* normalizava a amostragem em 120 amostras por segundo antes de executar a classificação. O gráfico a seguir da figura 37, indica a distribuição de amostragem do acelerômetro sobre os dados

coletados. Foi considerado para normalização o intervalo de amostragem entre 60 a 240 amostras por segundo.

Figura 37: Distribuição da amostragem dos dados coletados pelo sensor

Fonte: Autor

A aplicação do sensor criava um pacote de dados a cada 20 amostras de leitura no acelerômetro. Ao longo do experimento, o sensor coletou e transmitiu para o sorvedouro 4,17 milhões de pacotes. Estes pacotes foram processados e agrupados pelo sorvedouro em 361.145 pontos disponíveis no banco de dados para a classificação, cada um representando 1 segundo. Os pontos foram classificados em 4 categorias:

- **Carro Parado:** O veículo durante o experimento poderia estar em qualquer lugar, a qualquer velocidade, inclusive parado, em um congestionamento ou aguardando o semáforo da via oferecer passagem.

- **Via normal:** Esta categoria considera a via recém asfaltada ou a plena condição de dirigibilidade.

- **Via ondulada:** Existem imperfeições na via visíveis e sentidas pelo motorista quando está dirigindo, porém não atrapalha severamente o condutor de alcançar o seu destino.

- **Buraco:** Os buracos são as anomalias na via (fendas, rachaduras, depressão, etc.) que podem danificar o veículo, principalmente o sistema de suspensão, ou até mesmo ocasionar acidentes (DNIT, 2005).

Figura 38: Perfil de aceleração das categorias analisadas

Fonte: Autor

Cada ponto possui uma rugosidade distinta detectada pelo perfil de aceleração nos eixos X, Y e Z quando o veículo passa sobre ele. A figura 38 mostra o resultado

no domínio do tempo da aceleração nos três eixos para cada categoria com 120 amostras por segundo.

Foi utilizado pontos de referência para treinar o modelo analítico que detectava a categoria do ponto percorrido, neste caso, representado pelo algoritmo de classificação. Cada ponto de referência foi coletado sob uma condição previamente conhecida e notoriamente definida, por exemplo, carro parado. A figura 39 representa a quantidade e distribuição dos pontos de referências usados.

Figura 39: Quantidade e distribuição dos pontos de referências usados

Fonte: Autor

Como não é possível realizar esta análise meticulosa para todos os pontos que se deseja classificar durante o experimento, utiliza-se algoritmos matemáticos de classificação. Para isto, é comum dividir o conjunto original de dados, neste caso formado pelos pontos de referências, em dois conjuntos específicos denominados de conjunto de treinamento e conjunto de teste, com objetivo de construir um modelo matemático e testá-lo na classificação de pontos previamente conhecidos (FACELLI, LORENA, *et al.*, 2011). O conjunto de treinamento é usado para definição do modelo de classificação e o conjunto de teste é usado para examinar a acurácia do modelo (FACELLI, LORENA, *et al.*, 2011). Normalmente o conjunto de dados original é subdividido em diversas partes, onde cada uma contém um conjunto de treinamento e outro de teste. O desempenho final deste modelo de predição é dado pela média dos desempenhos observados pelos conjuntos de teste (ARLOT e CELISSE, 2010). Este método é denominado de validação cruzada (ARLOT e CELISSE, 2010).

Entre os diversos métodos matemáticos de classificação existentes, também encontrados por CHUGH, BANSAL e SOFAT (2014) em sua publicação *Road Condition Detection Using Smartphone Sensors: A Survey*, foram selecionados os algoritmos K-Vizinho mais próximo (K-NN), o classificador ingênuo de Bayes (*Naive Bayes*), a máquina de vetor de suporte (SVM), redes neurais, árvores de decisão e regras de decisão aplicáveis as técnicas de aprendizado de máquina para comparação. Uma breve descrição de cada modelo de algoritmo utilizado encontra-se disponível no apêndice H. O objetivo da comparação foi detectar qual é o método mais preciso para a detecção de buracos em vias públicas, usando dados de um acelerômetro tri-axial e também a informação sobre a velocidade instantânea do automóvel. A figura 40 representa os resultados da precisão da classificação destes métodos considerando as 4 possibilidades avaliadas (veículo parado, via normal, via ondulada e buraco). Neste gráfico, os resultados mostrados são a média da precisão do modelo e a margem de tolerância de erro, em porcentagem.

Figura 40: Resultado da análise dos métodos de classificação do estado da via pública usando dados de velocidade e de um acelerômetro tri-axial

Fonte: Autor

A figura 41 compara os modelos apenas para a classificação de buracos na via. Foram avaliados tanto a precisão do modelo quanto a totalidade dos pontos positivos que o modelo classificou corretamente. O resultado é apresentado em porcentagem.

Ambos os resultados da figura 40 e 41 foram obtidos pela técnica de validação cruzada subdividindo o conjunto de dados em 10 subconjuntos de treinamento e teste, para a construção do modelo e avaliação da precisão (ARLOT e CELISSE, 2010). O

modelo *Naive Bayes* foi escolhido o modelo matemático para esta tarefa no *middleware*, por possuir a maior precisão, 96,7% e menor erro absoluto, cerca de 4,9%. Também foi o modelo que mais acertou os pontos verdadeiramente positivo, com 97,1% de acertos.

Figura 41: Análise dos métodos de classificação para detecção de buracos em via pública usando dados de velocidade e de um acelerômetro tri-axial

Fonte: Autor

Em todo o experimento, nas 225 rotas avaliadas foram encontrados 11.305 buracos nas vias pelo lado do sensoriamento automático. A figura 42 representa os resultados da classificação do algoritmo durante o experimento.

Figura 42: Resultado da análise de classificação do experimento

Fonte: Autor

A figura 43 ilustra a visualização da localização dos buracos encontrados durante o experimento de acordo com o modelo analítico de classificação.

Figura 43: Visualização dos buracos encontrados pelo modelo analítico de classificação na interface gráfica do sistema de gerenciamento

Fonte: Autor

É possível por outros métodos o sensor executar além da coleta do perfil de aceleração da via, também a classificação do ponto, através de um software não atualizável (ERIKSSON, GIROD, *et al.*, 2008) ou com *hardware* com filtros dedicados (CHEN, LU, *et al.*, 2011), porém estas abordagens não foram o foco deste trabalho. Neste estudo é o sistema de gerenciamento que recebe os dados dos sensores e os classifica por modelos analíticos. Embora fora do escopo desta pesquisa, mas viabilizado pela arquitetura estudada, presume-se que o *middleware* aperfeiçoe ao longo do tempo de operação na cidade o modelo de classificação e então repasse a atualização com novas regras ao componente de regras presente no nó sensor, conforme discutido na seção 5.4. Desta maneira, o sensor pode prover dados já classificados ao sistema de gerenciamento de acordo com a realidade específica da cidade.

Para o teste com a aplicação *crowdsensing* foi percorrido duas vias consideradas comuns aos trajetos realizados pelo condutor. No total, foram coletados 94 relatórios em 3 Km. A figura 44 indica os problemas encontrados nas 2 vias analisadas nos

pontos referenciados com existência de buraco pelo sensoriamento automático. Embora a maior categoria de problemas foi devido a buracos, também se detectou outros defeitos na via, tais como, depressões, remendos e trincas. Também foi observado que o sensoriamento automático identificou os artefatos que pertencem normalmente a uma via pública como valetas, tampa de bueiro, lombadas e tachões.

Figura 44: Resultados dos relatórios coletados de *crowdsensing*

Fonte: Autor

Tabela 1: Trajetos percorridos na avaliação *crowdsensing*

Trajetos	Trajetos avaliados por <i>crowdsensing</i>	Repetições de coletas pelo sensor montado no veículo
1	R. Parque Domingos Luis (n° 1 – n° 800) e R. Pedro Cacunda (n° 100 – n° 500)	29
2	R. Tanque Velho (n°1 - n°2000)	28

Fonte: Autor

Com este resultado, cerca de 39% dos relatórios confirmaram o problema de buraco na via, detectado previamente pelo acelerômetro montado no veículo, além de 31% de outros problemas na via. Embora algumas categorias podem ser consideradas como falsos positivos (tampa de bueiro, lombada, valeta, tachão), não foram encontrados durante o experimento de *crowdsensing* verdadeiros negativos, ou seja, casos onde havia o buraco na via e o sistema automatizado não detectou.

Baseado nesta evidência, foram necessários pouco menos de 30 trajetos para cobrir todos os pontos com falhas nas vias locais de mão dupla analisadas. A tabela 1 indica os trajetos percorridos na avaliação *crowdsensing* e a quantidade de vezes que foram realizados as coletas de dados sobre a via, pelo sensoriamento automatizado.

Figura 45: Visualização dos pontos do trajeto 1 avaliado de *crowdsensing*

Fonte: Autor

A figura 45 mostra a visualização dos pontos no trajeto número 1 avaliado, incluindo os relatórios de *crowdsensing* e os pontos do sensoriamento automático.

Figura 46: Conjunto de 8 imagens coletadas por *crowdsensing*

Fonte: Autor

Também foi possível analisar as imagens coletadas e verificar se a subjetividade da imagem pelo participante pode comprometer a interpretação do gestor. A figura 46 mostra um conjunto de 8 fotos coletadas durante o experimento.

Considerando o serviço de atualização do *middleware*, assim que era disponibilizado uma nova versão, bastava o participante reiniciar a aplicação *crowdsensing* em seu *smartphone* e possuir uma conexão ativa com a internet, que ele recebia automaticamente um alerta avisando-o que já havia uma nova versão disponível para *download*. A figura 47 apresenta a interface de gerenciamento provida pela plataforma IBM MobileFirst⁴² que executava a distribuição das aplicações e gerenciamento das atualizações dos dispositivos móveis, incorporada ao *middleware*.

Figura 47: Interface de gerenciamento da aplicação *crowdsensing*

The screenshot shows the IBM MobileFirst Platform Operations Console. At the top, there's a header with the title 'IBM MobileFirst Platform Operations Console' and a user greeting 'Welcome, root | Logout | About'. Below the header, there are four tabs: 'Catalog', 'Devices' (which is selected), 'Push Notifications', and 'Log Profiles'. Under the 'Devices' tab, there's a search bar labeled 'Find a device:' with a placeholder 'User ID or Friendly Name' and a 'Search' button. To the right of the search bar is a 'Refresh' button. Below the search bar is a table listing devices. The table has columns for 'User ID', 'Friendly Name', 'Device Model', 'Device Version', 'Device ID', 'Status', and an action column with a '+' icon and a 'Delete' link. There are 8 rows of device data.

	User ID	Friendly Name	Device Model	Device Version	Device ID	Status	
+			SM-N7502	4.3	5d8f8ff1-fba1-384b-a11a-afd7778599c2	Active	Delete
+			SM-N9005	5.0	ae589674-f04b-3f59-9ed1-a4d3ffec361	Active	Delete
+			SM-G900M	5.0	fbac5a33-22fb-3428-8588-e5d3227d115d	Active	Delete
+			SM-A500M	4.4.4	f1d4bbf6-9fe3-3031-8153-dcd90cdc6467	Active	Delete
+			GT-I9100	2.3.4	d8acf3b4-2e36-3eec-b715-535f2123f1d7	Active	Delete
+			SM-G530H	4.4.4	6a5df4cb-5719-3eca-beac-48b38899e536	Active	Delete
+			SM-G920I	5.0.2	87ededc3-03bf-3cb3-87cb-8307bcab54d9	Active	Delete

Fonte: Autor

Foi disponibilizado para testes durante 15 dias a aplicação *crowdsensing* para 5 participantes, após o encerramento da coleta dos dados pelo sensoriamento automático. A figura 47 também apresenta os dispositivos que pertenceram a esta parte da pesquisa, incluindo 2 aparelhos que exerceram a função exclusiva de teste. Adicionalmente, foi realizado com estes participantes e mais outros 5, que não usaram a aplicação, uma pesquisa de opinião sobre este tipo de tecnologia. A figura 48 mostra

⁴² <http://www.ibm.com/mobilefirst/br/pt/>

o resultado das opiniões colhidas. O modelo do questionário encontra-se disponível no apêndice I.

Figura 48: Resultado das opiniões dos participantes no experimento

Fonte: Autor

Durante a aplicação desta consulta, observou-se que as pessoas de certa forma já conhecem tecnologias semelhantes e tendem a comparar com o Waze, embora tal aplicação é direcionada ao sensoriamento oportunista, conforme discutido no capítulo 3. Também foi expressado pelos participantes que a aplicação móvel proposta pode ajudar a cidades, mas eles precisam estar vivenciando o problema, ou notar mais claramente alguma recompensa para uso. Prevendo este tipo de necessidade, no questionário havia uma pergunta específica sobre incentivos monetários por parte do gestor público. A maioria das pessoas indicaram que o menor incentivo monetário já estimularia a adoção e utilização frequente. Outro fator encontrado na aplicação deste questionário foi a predisposição dos participantes em caminhar para auditar o ponto sugerido pelo gestor público. Embora a quantidade amostral de pessoas entrevistadas

seja muito reduzida, esta consulta verificou que algumas pessoas estão dispostas a contribuir com este tipo de tecnologia.

Por fim, foram realizados testes de carga para verificar a capacidade de processamento do *middleware*. Os resultados são mostrados na figura 49, com as métricas de atraso e perdas de pacotes. Estes testes foram baseados em simulações, as quais cada sensor enviava durante 1 hora dados com amostragem de 120 medições por segundo. O processamento consistia no gerenciador de mensagens receber os dados, o coordenador realizar a classificação do ponto e o serviço do gerenciador de banco de dados realizar a persistência dos dados. Foi observado que o tempo de processamento por mensagens era próximo a 300ms e o gerenciador de mensagens conseguia realizar o enfileiramento de mensagens a processar sem perdas de conteúdo, em até 5 sensores. Conforme discutido na seção 5.5, estas características podem ser melhoradas conforme aumenta-se a capacidade dos recursos disponíveis no serviço de computação em nuvem. Além disso, por não ter sido objetivo deste trabalho identificar a configuração ótima dos diversos módulos deste sistema (servidor de mensagens, aplicações ou banco de dados), acredita-se que existem condições de otimizar este sistema, incluindo mais sensores utilizando os mesmos recursos de computacionais.

Figura 49: Testes de carga de processamento do *middleware*

Fonte: Autor

6 TRABALHOS RELACIONADOS

Com a popularidade das aplicações móveis e também de aplicações em dispositivos embarcados, principalmente incentivados por projetos livres como o Arduino⁴³ e o Raspberry Pi⁴⁴ e também das reiteradas ondas de manifestações sociais por serviços públicos eficientes, nota-se um renovado interesse de pesquisa destas tecnologias, bem como trabalhos estudando a interação de ambas tecnologias. Este capítulo tem o objetivo de apresentar os trabalhos mais significativos encontrados no transcorrer desta pesquisa que relacionam as fontes de dados geradas por sensores e dispositivos móveis. Entre os mais interessantes projetos estão o SmartSantander (SANCHEZ, MUNOZ, *et al.*, 2013) e o centro de operações IBM, implantado na cidade do Rio de Janeiro (IBM CORPORATION, 2011). Estes projetos estão em franca operação e consideram importantes pontos de conectividade e utilização eficiente dos recursos e serviços da cidade para a construção de cidades inteligentes, inclusive com a utilização de métodos e tecnologias para aprimorar o diálogo com os seus cidadãos. Porém como os demais trabalhos encontrados, não vislumbram o potencial que a integração das informações coletadas por sensores e identificadas por seus cidadãos, em um mesmo serviço da cidade, podem exercer para melhorar a tomada de decisão dos gestores públicos.

6.1 SmartSantander

O projeto *SmartSantander*⁴⁵ exhibe uma plataforma aberta em larga escala para testar tecnologias de RSSF conectadas à internet, também chamada de internet das coisas ou *Internet of Things* (ATZORI, IERA e MORABITO, 2010). A infraestrutura do experimento publicada (SANCHEZ, MUNOZ, *et al.*, 2013), consistiu de 790 nós fixos com sensores das mais variadas funcionalidades: temperatura, intensidade luminosa, ruído, umidade do solo, anemômetro, pressão atmosférica, radiação solar, pluviômetro e monóxido carbônico; 390 nós instalados no asfalto para detecção de presença de carros em pontos de estacionamento da rua, 150 nós móveis instalados

⁴³ <https://www.arduino.cc/>

⁴⁴ <https://www.raspberrypi.org/>

⁴⁵ <http://www.smartsantander.eu>

em ônibus e táxis para monitorar o trânsito da cidade. Atualmente a infraestrutura possui 3000 sensores, segmentados para as atividades de: gerenciamento das vagas de carros nas ruas; intensidade do tráfego automotivo; irrigação automática de parques e jardins públicos e também para o monitoramento ambiental, por meio de sensores fixos e também móveis, instalados em veículos de transporte público.

Figura 50: Arquitetura SmartSantander

Fonte: SANCHEZ, MUNOZ, et al, 2013

Este projeto é o maior esforço de experimentação de tecnologias de internet das coisas, do inglês, *Internet of Things* (IoT) direcionada a cidades realizado fora do ambiente de laboratório, que considerou também a inclusão de 2000 etiquetas NFC (*Near Field Communication*) e QR (*Quick Response*) em pontos turísticos e de informação para suporte a realidade aumentada (SANCHEZ, MUNOZ, et al., 2013). Em conjunto a este esforço, também foi disponibilizada uma aplicação de sensoriamento participativo aos cidadãos. Esta tecnologia de *crowdsensing* também

tem objetivo de engajar o cidadão a reportar problemas como: buraco na rua, acidente de trânsito (SANCHEZ, GALACHE, *et al.*, 2011).

A principal diferença observada entre a proposta desta pesquisa e a apresentada no projeto SmartSantander é que esta pesquisa propõe a criação de métodos para correlação entre dados gerados por sensores e *crowdsensing*, através de um *middleware*, enquanto que o projeto SmartSantander tem o foco na utilização dos dados dos sensores e do sensoriamento participativo desvinculados. Embora na aplicação de realidade aumentada exista a interação localizada entre o dispositivo móvel do cidadão e uma etiqueta NFC, o relacionamento entre estas fontes de dados, mesmo sendo um interessante artifício para engajamento do cidadão, não é o objeto desta pesquisa. Esta pesquisa não visa o estudo da interação entre estes dispositivos heterogêneos, mas de maneira holística estuda quais os efeitos e causas que podem ser interpretados quando estas fontes de informação atuam integradamente. A arquitetura do projeto SmartSantander é apresentada na figura 50.

6.2 MetroSense

O projeto MetroSense⁴⁶ é uma arquitetura de rede para contemplar a interação entre sensores fixos, sensores móveis e pessoas portando dispositivos móveis em escala urbana. A motivação do projeto está no sensoriamento oportunista que pode ser explorado nesta interação (CAMPBELL, EISENMAN, *et al.*, 2006). Segundo os autores, as RSSF podem conter sensores móveis, fixos e sorvedouros, sendo eficientes para atividades de sensoriamento localizadas. Contudo, quando comparadas com outras redes de comunicação, como a GPRS, as RSSF possuem baixo alcance de comunicação e normalmente têm restrições energéticas. Porém as pessoas podem carregar dispositivos móveis com sensores (acelerômetro, GPS) os quais normalmente são mantidos carregados pelos usuários. A partir de interações com RSSF estes dispositivos podem ser utilizados para aumentar o alcance de comunicação ou receber instruções para realizar o sensoriamento de uma atividade específica.

⁴⁶ <http://metrosense.cs.dartmouth.edu/>

Figura 51: Arquitetura do projeto MetroSense

Fonte: A. T. CAMPBELL, S. B. EISENMAN e N. D. LANE, et al, 2006

A diferença entre a proposta desta pesquisa e a apresentada no projeto MetroSense é que esta pesquisa propõe a criação de métodos para correlação entre os dados da rede de sensores em serviços de cidades e de *crowdsensing*, enquanto que o projeto MetroSense se propõe a criar a infraestrutura de rede para interação entre a RSSF e os dispositivos móveis dos cidadãos. A abordagem desta pesquisa é prover métodos que correlacionem ambas as fontes de dados para suporte a decisão de gestores públicos por intermédio de um *middleware*, enquanto que o projeto MetroSense se restringe a aumentar a eficiência e a cobertura de uma tarefa de sensoriamento urbano pelo oportunismo das qualidades destes dois sistemas heterogêneos: a RSSF e o sensoriamento por pessoas. A arquitetura do projeto MetroSense é apresentada na figura 51.

6.3 G-Sense

O projeto G-Sense (PEREZ, LABRADOR e BARBEAU, 2010) propõe uma arquitetura que integra sensores fixos e móveis com as propostas de sensoriamento participativo (BURKE, ESTRIN, *et al.*, 2006), sensoriamento centrado em pessoas e serviços baseado em localização.

A arquitetura é voltada para aplicações militares, onde as tropas podem conter em seu corpo sensores de atividade física, por exemplo, sensores de batimento cardíaco. Também podem monitorar durante uma caminhada no campo de batalha a presença de gases ou as condições climáticas, através de sensores nos dispositivos móveis. Esta proposta também considera que sensores de presença georreferenciados, previamente espalhados na região de interesse, podem identificar a presença de soldados inimigos e alertar o centro de comando.

Neste mesmo trabalho, propõe-se a presença de diversos servidores para hospedar este sistema, compartilhando informações através de redes P2P. Embora esta prática seja muito relevante para a distribuição de responsabilidades e aumentar a qualidade de serviço em aplicações militares, uma vez que um servidor pode ser atacado sem comprometer a operação do sistema, não há sentido prático em aplicações voltadas a cidades, onde infraestrutura pode ser contemplada por computação em nuvem, oferecendo vantagens em termos segmentação do controle administrativo, de custo e escalabilidade.

As figuras 52 e 53 representam respectivamente a arquitetura do sistema G-Sense do lado do cliente e do servidor. Nela os autores não consideram protocolos de mensagem entre os sensores ou dispositivos móveis dos usuários aos servidores. Também não consideram mecanismos do *middleware* para recebimento assíncrono de mensagens como nos paradigmas publicar e assinar. Estes pontos restringem a escalabilidade do sistema para inclusão de sensores e participantes necessária em cidades.

Figura 52: Arquitetura G-Sense do lado do Cliente

Fonte: Adaptado de PEREZ, LABRADOR e BARBEAU, 2010

Figura 53: Arquitetura G-Sense do lado do Servidor

Fonte: Adaptado de PEREZ, LABRADOR e BARBEAU, 2010

Embora comentado como possível a análise e correlação de dados entre os sensores e os dados coletados pelos dispositivos móveis dos participantes, representados pelas tropas, os autores não caracterizaram no trabalho os modelos analíticos que poderiam ser desenvolvidos para esta aplicação.

6.4 OpenSense

Este trabalho descreve uma visão de um sistema de sensoriamento aberto, autossustentável, pelo ponto de vista entre produtores e consumidores de dados e também auto gerenciado pela comunidade (ABERER, SATHE, *et al.*, 2010). Os autores acreditam que uma abordagem utilitária dos dados possa manter este sistema sustentável, sendo necessários inicialmente explorar uma infraestrutura que comporte sensores heterogêneos carregados pelos indivíduos nos mais diversos modos, tais como: sensores pertencentes em dispositivos móveis, montados em veículos ou vestíveis como os relógios inteligentes.

Figura 54: Ciclo de produção e consumo de dados de sensoriamento comunitário

Fonte: Adaptado de ABERER, et al, 2010

Os autores indicam algumas diferenças de rede de sensores tradicionais como: propriedade dos dados, heterogeneidade dos sensores, confiabilidade dos sensores e também dos participantes. Também são apontados diversos desafios deste tipo de visão, tais como os modelos de sensoriamento, gerenciamento dos dados, tolerância a falhas, gerenciamento de energia, reputação e privacidade. A figura 54 representa o ciclo de produção e consumo de dados do Opensense.

Esta visão, embora envolva a participação de sensores e dispositivos móveis, é diferente da abordagem desta pesquisa, principalmente pela não-existência de um

sistema de gerenciamento definido. Especifica-se nesta pesquisa claramente um sistema de gerenciamento, preferivelmente baseado em um serviço em nuvem, que os gestores das cidades possam utilizá-lo para tomada de decisão, sendo apoiado por dados coletados por sensores e seus cidadãos.

6.5 Centro de Operações Inteligente IBM

Os autores definem o projeto como uma ferramenta inteligente que provê acesso às informações e capacidades da cidade, habilitando os gestores públicos a gerenciá-la e melhorá-la em suas operações (IBM CORPORATION, 2011).

Figura 55: Diagrama funcional do Centro de Operações de domínio Cruzado

Fonte: Adaptado de IBM Corporation, 2011

A figura 55 mostra a completa arquitetura do centu de operações proposto. Entre as principais funcionalidades proposta estão:

Colaboração entre domínios. As diversas agências (transporte, saneamento, saúde) trocam informações entre si, coordenando suas ações individualizadas em prol de um objetivo comum;

Gerenciamento de incidentes e crises. Administrando ações e informações dos diversos domínios da cidade em um mesmo lugar, os gestores podem direcionar os recursos disponíveis, priorizando as ações para a resolução otimizada do incidente;

Conjunto de políticas e regras. O conjunto de políticas da cidade formam os princípios para a construção e manutenção de procedimentos e protocolos de operação das diversas agências. A implementação destas políticas em regras definidas ajudam na tomada de ações automatizadas, provendo assim consistência nos serviços da cidade.

Vista operacional comum das condições atuais da cidade. Visualizações em tempo real e históricas são disponibilizadas em um local centralizado para acesso aos vários gestores da cidade. Indicadores de performance e alertas auxiliam em uma visualização operacional compreensiva da cidade.

Para os aspectos que consideram a integração dos cidadãos, alguns serviços são prestados, por exemplo, um portal de acesso WEB, uma central de chamadas telefônicas, além de um serviço de *chat* e notificações. Os autores comentam que as ações de alertas aos munícipes a eventos extraordinários da cidade, como um acidente em uma via importante, podem ser coordenando pelo centro de operações. Os cidadãos também podem utilizar aplicações móveis *crowdsensing* de modo a relatar problemas na cidade e acompanhar o atendimento da resolução do problema. Na cidade do Rio de Janeiro, um exemplo deste centro de operações foi implantado⁴⁷, sendo o serviço 1746 Rio⁴⁸ o portal de relacionamento *crowdsensing* da prefeitura.

Embora este projeto contemple diversas funcionalidades fora do escopo desta pesquisa, não prevê que os cidadãos podem interagir de modo automatizado com as informações coletadas pelos sensores da cidade, que é o escopo desta pesquisa. O incremento no contexto dos dados gerados por sensores e as informações adicionais

⁴⁷ <http://centrodeoperacoes.rio/institucional>

⁴⁸ <http://www.1746.rio.gov.br/>

dos munícipes, pode facilitar a priorização aos atendimentos das agências, como também gerar um novo perfil de engajamento do cidadão com sua cidade.

7 CONCLUSÃO

As cidades estão a seu tempo e a seu modo modernizando os serviços prestados à população. Entre os diversos fatores que estão contribuindo para esta evolução, estão a diversificação e proliferação de sensores nos diversos domínios de serviços. O que em algumas cidades pode ter iniciado com termômetros de ruas e estações meteorológicas, já é cotidiano o GPS em ônibus e caminhão de lixo para controle da frota e facilitar a previsibilidade do serviço ao usuário, ou ainda em algumas cidades, sensores para monitoração da lotação de vagas das ruas, em postes de luz para detectar falhas no serviço, na automação da irrigação de praças e até mesmo para monitoração de infraestruturas, como em pontes. Além das redes de sensores contribuir neste processo de modernização dos centros urbanos, novos canais de comunicação com os munícipes estão disponíveis, por exemplo, as redes sociais e mais recentemente, graças a popularização dos dispositivos móveis, os sistemas *crowdsensing*. O sensoriamento *crowdsensing* oportunista, ao qual destaca-se mundialmente o Waze⁴⁹, ou o participativo, disponível em algumas cidades como o 1746⁵⁰, são ferramentas as quais o cidadão pode, de acordo com sua necessidade, convicção ou altruísmo, exercer sua própria influência nos gestores públicos para a continua excelência dos serviços públicos.

As capacidades das tecnologias de sensoriamento automatizado, representado pela rede de sensores, ou do sensoriamento colaborativo observada por sistemas *crowdsensing*, foram estudadas durante esta pesquisa para o entendimento de como podem estar integradas na melhoria de um serviço público. Durante este estudo foi observado que embora cada domínio de serviço da cidade requer estratégias distintas de sensoriamento, por exemplo, câmeras de vídeo são muito utilizadas no contexto de vigilância enquanto que os termômetros atendem melhor as questões sobre monitoramento ambiental, é possível utilizar técnicas de sistemas distribuídos para que tais serviços usufruam da mesma infraestrutura computacional. Esta estratégia evita gastos administrativos duplicados e até mesmo possibilita a correlação de eventos, por exemplo, o serviço de emergência que atende acidentes de trânsito pode observar a causalidade de certos eventos devido à má conservação de um trecho em

⁴⁹ www.waze.com

⁵⁰ www.1746.rio.gov.br

uma via, o que em certas cidades é de responsabilidade de outro órgão público a conservação das vias. Esta mesma estratégia da utilização de sistemas distribuídos em prol da agregação das informações de diversos domínios de serviços das cidades foi verificada por outros trabalhos, comentados no capítulo 6, e explorada nesta pesquisa por um *middleware*.

Entende-se que o objetivo deste trabalho sobre o entendimento de tecnologias e métodos para a coleta, integração e interpretação de dados dos serviços prestados nas cidades com foco nas informações de sensores e *crowdsensing* foi cumprido. Nos capítulos 2, 3 e 4 foram discutidas os conceitos, características, tipos, aplicações e desafios de rede de sensores, *crowdsensing* e sistemas distribuídos. Já no capítulo 5, foi formulado uma arquitetura direcionada ao tratamento destas fontes de dados, que pode ser genericamente utilizada em vários domínios de serviços, a qual resumidamente possui as seguintes capacidades: i) administrar o fluxo de dados dos sensores e relatórios dos participantes; ii) tornar-se adaptável à demanda variável de sensores e participantes; iii) persistir os dados para prover mecanismos de séries históricas e evolução dos serviços; iv) manter critérios de armazenamento de dados distintos conforme a criticidade da informação; v) disponibilizar aos serviços de visualização relatórios previamente analisados com modelos analíticos pertinente ao serviço avaliado; vi) notificar os participantes sobre pontos de auditoria disponíveis; vii) identificar eventos extraordinários que podem ser comprovados por ambas as fontes de dados; viii) alertar os gestores sobre situações extraordinárias com os serviços monitorados, a partir dos dados coletados pelos sensores e participantes; ix) disponibilizar a inserção de regras para tratamentos de rotinas automáticas e x) controlar as versões dos *softwares* instalados na rede de sensores e também nas aplicações *crowdsensing*.

Considera-se que a hipótese testada da pesquisa, sobre se prover uma infraestrutura de coleta e integração de dados de sensores e *crowdsensing* para os serviços das cidades, em um *middleware* com módulos analíticos capazes de interpretar integradamente estas fontes de informação, permitirá aos agentes públicos melhorarem a monitoração dos serviços públicos e planejar ações administrativas priorizadas, foi confirmada. A arquitetura, o *middleware* e o cenário de uso elaborados no capítulo 5 foram direcionados a testar esta hipótese. Este cenário, a monitoração

da conservação das vias públicas, baseia-se no mapeamento georreferenciado da rugosidade das ruas por veículos públicos (ônibus, ambulância, viaturas policiais e veículos de fiscalização) montados com um nó sensor, composto por um acelerômetro tri-axial e GPS. Então, ao final do dia, estes veículos transferem os dados automaticamente a sorvedouros por uma rede sem fio. Estes dados são analisados por *middleware* que também recebe relatos de participantes, contendo fotos, sobre as condições das vias. Por fim, através de um conjunto de regras estabelecidas, este *middleware* pode notificar os participantes sobre pontos a ser auditados ou também alertar o gestor público sobre os locais que foram confirmados pelos participantes, priorizando desta forma a manutenção das vias nas regiões confirmadas.

Com os resultados de 3 meses de experimento, considerando mais de 360 mil pontos coletados pelo nó sensor, estima-se, a partir dos mais de 90 relatórios enviados por *crowdsensing*, que 29% dos defeitos na via percebidos pelo sensoramento automatizado tratam-se de falsos positivos (tampa de bueiro, valeta, lombada e tachão). Até mesmo modificando as características do agrupamento de dados para incluir eventos longos como lombadas e valetas, em comparação com buracos, ainda vão existir os falsos positivos, por exemplo, gerados pelas tampas de bueiro. Esta situação demonstra a importância do suporte da auditoria dos participantes, ao ponto avaliado, como critério de validação e prioridade ao atendimento. Pelo lado de *crowdsensing* a subjetividade da imagem e urgência do problema, devido a pluralidade de defeitos das vias, torna o suporte da leitura de sensores um critério objetivo de qualificação. É até mesmo possível incluir nos modelos analíticos do *middleware* (classificador), normas municipais sobre o tipo de defeito tratado pela prefeitura, por exemplo, o tamanho mínimo de um buraco na via, tornando o critério de qualificação do ponto dentro do controle administrável pelo gestor.

Desafios avaliados

Os desafios inicialmente encontrados foram superados da seguinte maneira:

- Modelos analíticos que correlacionem informações das fontes distintas de dados. Como dito, os diferentes domínios dos serviços das cidades (transporte, segurança, saneamento, etc.) requerem estratégias distintas para instrumentação por sensores. Para o cenário avaliado, a monitoração da conservação das vias públicas,

os modelos analíticos executavam: i) A curadoria dos dados, representada pela compatibilização dos dados coletados pelo sensor ao algoritmo de classificação; ii) A classificação dos dados, identificando quais os pontos na cidade eram realmente necessários para auditoria do participante, neste caso, representado pelos defeitos nas vias; iii) Identificação sobre quando um participante estava em um local apropriado para receber uma solicitação de auditoria, de acordo com uma regra pré-estabelecida pelo gestor.

- Infraestrutura de *hardware* e *software* que comporte o volume de dados em cidades. A estratégia endereçada foi a construção de um *middleware* com paradigma de comunicação publicar e assinar contendo os serviços de gerenciamento de eventos, gerenciamento do banco de dados, coordenador, notificações, relatórios e atualizações, hospedado em uma infraestrutura de computação em nuvem. Estes serviços com estas funcionalidades se mostraram apropriados ao cenário de uso, porém serão necessários incluir novas funcionalidades a estes em outros cenários de aplicação. Especialmente a proposta de hospedar o *middleware* em uma infraestrutura de computação em nuvem, se mostrou conveniente a aplicações em cidades pelos critérios de não haver um custo inicial e o serviço de computação ser dimensionado de modo escalável conforme a necessidade endereçada.

- Padronização dos dados provenientes das duas fontes distintas. A técnica de sistemas distribuído oferecida pelo paradigma de comunicação publicar e assinar se mostrou apropriado para a comportar as transações enviadas pelos sensores e também pelos relatórios dos participantes. Neste trabalho foi utilizado o MQTT.

Perguntas de Pesquisa

As perguntas de pesquisa foram endereçadas da seguinte maneira:

“Como coletar dados, prover pré-processamento (filtragem) e análise a sistemas de gerenciamento dos serviços das cidades? ”

Existem diversos fenômenos possíveis para a coleta de dados por sensores (temperatura, umidade, aceleração, detecção gases, etc.) que são dependentes do contexto monitorado. O tipo de variável e o contexto analisado afetam a estratégia de pré-processamento e subsequente análise, por exemplo, o fenômeno de aceleração

com técnicas de aprendizado de máquina foi usada nesta pesquisa para detecção da rugosidade da rua com o objetivo de encontrar falhas na via, porém para encontrar defeitos em máquinas rotativas, como apresentado na seção 7.2, analisar o sinal do acelerômetro no domínio da frequência é mais apropriado. Os métodos de comunicação com os nós sensores são outro fator importante para consideração. As tecnologias de longa distância como GPRS e 3G já são populares em cidades, porém o IEEE 802.15.4, especificamente desenhado rede de sensores, proporciona uma maior densidade de sensoriamento localizado, devido a eficiência energética e operação em rede com topologia em malha. Combinado com as tecnologias de longa distância é empregado em diversas situações, como discutido no capítulo 2, inclusive em cidades. No cenário de pesquisa explorado, o IEEE 802.15.4 foi utilizado para transmitir os dados entre o sensor montado no veículo e o sorvedouro instalado na garagem.

“Quais são os aspectos para a integração de informações em ambientes urbanos envolvendo pessoas e os serviços das cidades? ”

Pelo lado do sensoriamento colaborativo as aplicações principalmente de caráter participativo, a qual foi avaliada nos experimentos, são dependentes de divulgação, altruísmo, credibilidade do atendimento a solicitação e até mesmo de experiências anteriores do participante com o problema a ser relatado. Além disso, existe um decaimento de utilização com o tempo, quando a aplicação é dependente exclusivamente do altruísmo. Já pelo lado do sensoriamento automático há mais garantias de entrega de dados, porém ocorrem inevitavelmente falsos alertas, devido as inúmeras variações da realidade. O desafio da integração está em solicitar o engajamento do participante quando ele está envolvido com a causa. Uma situação plausível é quando o participante está com a aplicação aberta, mas não necessariamente interagindo com ela. Neste trabalho foi testado esta integração quando o usuário termina de submeter um relatório. Nesta caso, o *middleware* em seguida avalia se indica ao participante um ponto a auditar, de acordo com uma regra de distância previamente estabelecida pelo gestor. Este tipo de estratégia pareceu ser conveniente, embora a distância a ser percorrida para avaliar o problema afeta consideravelmente o engajamento.

Trabalhos futuros

Em certas vias verificou-se a incidência de dezenas de problemas concentrado em pouco menos de 1 km, por exemplo, em ruas com paralelepípedos. Isto pode ser considerado um problema endêmico da via necessitando da manutenção de um longo trecho, ao invés de apenas um reparo de buraco. Um modelo analítico para detectar este tipo de padrão pode ser útil sugerindo ações corretivas mais extensas e duradouras em um espaço objetivamente delimitado, evitando reparos com pouca efetividade em localidades notoriamente problemáticas.

Para os dois trechos analisados, cerca de 3 km em ruas locais de mão dupla, foram notados que não houve verdadeiro negativo, após cerca de 30 vezes o veículo passar pela mesma localidade. A avaliação da quantidade ideal de medições em uma via por um veículo portando este tipo de tecnologia, considerando também a quantidade de faixas de rolagem, pode favorecer a criação de modelos analíticos que interpretem temporalmente se um ponto previamente detectado já foi reparado ou tratava-se simplesmente de um falso positivo.

7.1 Trabalho publicado desta pesquisa: Simulação de microdrenagem em cidades conscientes – XX Congresso Brasileiro de Automática

O trabalho analisa a implantação de uma rede de sensores sem fio, baseada em sensores de ultrassom, para monitoramento de resíduos urbanos em bocas de lobo (SILVA, NOGIMA, *et al.*, 2014). A premissa é que o monitoramento contínuo destes pontos é um importante agente na prevenção de enchentes, além de otimizar o serviço de limpeza urbana indicando com precisão os pontos que devem ser prioritariamente atendidos. É proposta uma arquitetura de sistema de microdrenagem urbana considerando as restrições do ambiente de operação, a coleta dos dados dos sensores pelo sorvedouro e um serviço de computação em nuvem para tratamento dos dados. Considera-se que as informações destes sensores serão coletadas, por intermédio de sorvedouros móveis instalados em caminhões de coleta domiciliar de lixo, e transferidas a um serviço de análise de dados em nuvem. As figuras 56 e 57 representam a arquitetura proposta.

Figura 56: Arquitetura de monitoração de microdrenagem urbana

Fonte: SILVA, et al, 2014

Esta rede oportunista de transferência de dados foi simulada, entre os sensores e o sorvedouro móvel, na região da Cidade Universitária, na cidade de São Paulo. De modo geral, foi observado que as alternativas tecnológicas existentes, como as redes de sensores sem fio, são plausíveis de ser implantadas em cidades com resultados extremamente favoráveis ao gerenciamento da infraestrutura de microdrenagem de cidades.

Figura 57: Representação do cenário de sorvedouro móvel

Fonte: SILVA, et al, 2014

Foi executada também a integração com o projeto *Citizen Sensing* (KOCK, CARDONHA, et al., 2013) para visualização dos dados. O objetivo do *Citizen Sensing* é engajar o cidadão, com um *smartphone*, a relatar problemas na cidade, tais como, poda de árvore, buraco na rua, calçadas com falhas de acessibilidade, etc.,

mostrando-se como um valioso agente na conservação da cidade e em sintonia ao conceito de cidades conscientes (SHEPARD, 2011).

Este trabalho tem a visão que a correlação de dados coletados entre os relatos dos cidadãos, sensores meteorológicos e sensores de microdrenagem, possibilitará a construção de modelos analíticos com recomendações objetivas aos gestores sobre o sistema de drenagem, entre eles:

- i) Pontos de limpeza prioritária;
- ii) Pontos de alagamento potenciais;
- iii) Locais que a coleta de lixo está deficiente;
- iv) Modelos de alagamento em bairros;
- v) Análise indireta da saturação das galerias pluviais.

7.2 Trabalho publicado desta pesquisa: *A Cloud-based Architecture for the Internet of Things Targeting Industrial Engine Remote – Sensors to Cloud Architectures Workshop – 21° IEEE Symposium on High Performance Computer Architecture (HPCA - 2015)*

Este trabalho propõe uma arquitetura baseada em computação em nuvem, para aplicações de internet das coisas, com o objetivo de facilitar a implantação de sistemas de monitoramento industrial. A premissa é que a aquisição, processamento, análise e gerenciamento de dados gerados por sensores em redes de monitoramento de ambientes industriais, podem se beneficiar de plataformas modernas voltada a internet das coisas. A análise de tais conjuntos de dados pode permitir a tomada de decisão rápida e precisa tendo como alvo a melhoria da produtividade (SILVA, AZEVEDO, *et al.*, 2015).

Embora este trabalho não está diretamente relacionado a aplicações em cidades, foi estudado nele algumas funcionalidades na arquitetura que também são usados nesta pesquisa. A figura 58 representa os detalhes da arquitetura, a qual incluiu: as camadas de rede e aplicações nos dispositivos que compõem a rede de sensores de coleta de dados; as adaptações de dados entre os protocolos, as técnicas

de enfileiramento de mensagens, análise de dados em tempo real e relatórios para análise posterior.

Figura 58: Detalhes da arquitetura da plataforma de internet das coisas para monitoramento de motores industriais

Fonte: Adaptado de SILVA, AZEVEDO, et al, 2015

O cenário avaliado foi o monitoramento de um motor de indução elétrica. Os resultados preliminares indicaram que esta arquitetura pode suportar métodos de análise para a implantação de metodologias de manutenção preditiva. Além disso, os resultados também indicaram que com esta plataforma a atividade de manutenção preditiva pode ser executada sob demanda, gerando um novo modelo de negócios de serviços para pequenas empresas de assistência técnica, provido pela rápida e fácil implantação no ambiente industrial.

Conclui-se também que embora este estudo se concentrou num único dispositivo específico, a arquitetura proposta pode ser estendida para uma ampla gama de aplicações, por exemplo, saúde, meio ambiente, monitoramento de tráfego em cidades, entre outros.

REFERÊNCIAS

- ABBASI, A. A.; YOUNIS, M. A survey on clustering algorithms for wireless sensor networks. **Computer Communications**, v. 30, n. 14, p. 2826–2841, 2007. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366407002162>. Acesso em: Agosto 2015.
- ABERER, K. et al. Opensense: open community driven sensing of environment. In: INTERNATIONAL WORKSHOP ON GEOSTREAMING. **Workshop**. San Jose: ACM. 2010. p. 39-42. Disponível em: http://dl.acm.org/ft_gateway.cfm?id=1878509&ftid=904772&coll=DL&dl=GUIDE&CFID=542076495&CFTOKEN=70550726. Acesso em: Agosto 2015.
- ABINEE. **Panorama Econômico e Desempenho Setorial**. São Paulo. 2015.
- AGARWAL, V. et al. USense - A Smartphone Middleware for Community Sensing. In: INTERNATIONAL CONFERENCE ON MOBILE DATA MANAGEMENT (MDM). **Proceedings**. Milan: IEEE. 2013. p. 56-65. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6569122>. Acesso em: Agosto 2015.
- AKAN, O. B.; KARLI, O.; ERGUL, O. Cognitive Radio Sensor Networks. **IEEE Network**, v. 23, n. 4, p. 34-40, 2009. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6892784>. Acesso em: Agosto 2015.
- AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. **Ad Hoc Network**, v. 3, n. 3, p. 325–349, 2005. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1570870503000738>. Acesso em: Agosto 2015.
- AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. **Computer Networks**, v. 38, n. 4, p. 393–422, 2002. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128601003024>. Acesso em: Agosto 2015.
- AKYILDIZ, I. F.; MELODIA, T.; CHOWDHURY, K. R. A survey on wireless multimedia sensor networks. **Computer networks**, v. 51, n. 4, p. 921-960,

2007. Disponivel em:

<<http://www.sciencedirect.com/science/article/pii/S1389128606002751>>.

Acesso em: Agosto 2015.

AKYILDIZ, I. F.; POMPILI, D.; MELODIA, T. Underwater acoustic sensor networks: research challenges. **Ad hoc networks**, v. 3, n. 3, p. 257-279, 2005. Disponivel em: <<http://www.sciencedirect.com/science/article/pii/S1570870505000168>>. Acesso em: Agosto 2015.

AKYILDIZ, I. F.; STUNTEBECK, E. P. Wireless underground sensor networks: research challenges. **Ad Hoc Networks**, v. 4, n. 6, p. 669-686, 2006. Disponivel em:

<<http://www.sciencedirect.com/science/article/pii/S1570870506000230>>.

Acesso em: Agosto 2015.

AL-KARAKI, J. N.; KAMAL, A. E. Routing Techniques in Wireless Sensor Networks: a Survey. **Wireless Communications**, v. 11, n. 6, p. 6-28, 2004. Disponivel em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1368893>>. Acesso em: Agosto 2015.

ANASTASI, G. et al. Energy conservation in wireless sensor networks: a survey. **Ad Hoc Networks**, v. 7, n. 3, p. 537-568, 2009. Disponivel em: <<http://www.sciencedirect.com/science/article/pii/S1570870508000954>>. Acesso em: Agosto 2015.

ARLOT, S.; CELISSE, A. A survey of cross-validation procedures. **Statistics surveys**, v. 4, p. 40-79, 2010. Disponivel em: <http://projecteuclid.org/download/pdfview_1/euclid.ssu/1268143839>. Acesso em: Agosto 2015.

AROLAS, E. E.; GUEVARA, F. G. L. Towards an integrated crowdsourcing definition. **Journal of Information Science**, v. 38, n. 2, p. 189-200, 2012. Disponivel em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.371.9949&rep=rep1&type=pdf>>. Acesso em: Agosto 2015.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: a survey. **Computer Networks**, v. 54, n. 15, p. 2787-2805, 2010. Disponivel em: <<http://www.sciencedirect.com/science/article/pii/S1389128610001568>>. Acesso em: Agosto 2015.

- BARONTI, P. et al. Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards. **Computer Communications**, v. 30, n. 7, p. 1655-1695, 2007. Disponivel em: <http://www.sciencedirect.com/science/article/pii/S0140366406004749>. Acesso em: Agosto 2015.
- BECKWITH, R.; TEIBEL, D.; BOWEN, P. Report from the field: results from an agricultural wireless sensor network. In: INTERNATIONAL CONFERENCE ON LOCAL COMPUTER NETWORKS. **Proceedings**. Tampa: IEEE. 2004. p. 471-478. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1367268>. Acesso em: Agosto 2015.
- BIEM, A. et al. IBM infosphere streams for scalable, real-time, intelligent transportation services. In: ACM INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA. **Proceedings**. Indianapolis: ACM. 2010. p. 1093-1104. Disponivel em: http://dl.acm.org/ft_gateway.cfm?id=1807291&ftid=805275&coll=DL&dl=GUIDE&CFID=542076495&CFTOKEN=70550726. Acesso em: Agosto 2015.
- BIRMAN, K.; SCHIPER, A.; STEPHENSON, P. Lightweight causal and atomic group multicast. **ACM Transactions on Computer Systems**, v. 9, n. 3, p. 272-314, 1991. Disponivel em: http://dl.acm.org/ft_gateway.cfm?id=128742&ftid=7811&dwn=1&CFID=542076495&CFTOKEN=70550726. Acesso em: Agosto 2015.
- BIRRELL, A. D.; NELSON, B. J. Implementing remote procedure calls. **ACM Transactions on Computer Systems**, v. 2, n. 1, p. 38-59, 1984. Disponivel em: http://dl.acm.org/ft_gateway.cfm?id=357392&ftid=58947&dwn=1&CFID=542076495&CFTOKEN=70550726. Acesso em: Agosto 2015.
- BOS, B. et al. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, 07 Junho 2011. Disponivel em: <http://people.dsv.su.se/~jpalme/internet-course/compendium-9.pdf>. Acesso em: Agosto 2015.
- BRABHAM, D. C. Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. **International journal of research into new media technologies**,

v. 14, n. 1, p. 75-90, 2008. Disponivel em:

<<http://con.sagepub.com/content/14/1/75.full.pdf>>. Acesso em: Agosto 2015.

BURKE, J. et al. Participatory Sensing. In: WORLD-SENSOR-WEB - MOBILE DEVICE CENTRIC SENSOR NETWORKS AND APPLICATIONS. **Workshop**. Boulder: ACM. 2006. p. 1–5. Disponivel em:
<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.3024&rep=rep1&type=pdf>>. Acesso em: Agosto 2015.

CAMPBELL, A. T. et al. People-centric urban sensing. In: INTERNATIONAL WORKSHOP ON WIRELESS INTERNET. **Workshop**. New York: ACM. 2006. p. 1-14. Disponivel em:
<http://dl.acm.org/ft_gateway.cfm?id=1234179&ftid=415362&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

CAMPBELL, A. T. et al. The Rise of the rise of People-Centric Sensing. **Internet Computing**, v. 12, n. 4, p. 12-21, 2008. Disponivel em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4557974>>. Acesso em: Agosto 2015.

CARDONE, G. et al. Fostering participation in smart cities: a geo-social crowdsensing platform. **IEEE Communications Magazine**, v. 51, n. 6, p. 112-119, 2013. Disponivel em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6525603>>. Acesso em: Agosto 2015.

CHEN, K. et al. Road condition monitoring using on-board Three-axis Accelerometer and GPS Sensor. In: INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND NETWORKING IN CHINA - CHINACOM. **Proceedings**. Harbin: IEEE. 2011. p. 1032-1037. Disponivel em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6158308>>. Acesso em: Agosto 2015.

CHON, Y. et al. Automatically characterizing places with opportunistic crowdsensing using smartphones. In: ACM CONFERENCE ON UBIQUITOUS COMPUTING. **Proceedings**. Pittsburgh: ACM. 2012. p. 481-490.

CHON, Y. et al. Understanding the Coverage and Scalability of Place-centric CrowdSensing. In: INTERNATIONAL JOINT CONFERENCE ON PERVASIVE

AND UBIQUITOUS COMPUTING. **Proceedings**. Zurich: ACM. 2013. p. 3-12.

Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=2493498&ftid=1392269&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

CHONG, C. Y.; KUMAR, S. P. Sensor networks: evolution, opportunities, and challenges. **Proceedings of the IEEE**, v. 91, n. 8, p. 1247-1256, 2003.

Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1219475>>. Acesso em: Agosto 2015.

CHRISTIN, D. et al. A survey on privacy in mobile participatory sensing applications. **Journal of Systems and Software**, v. 84, n. 11, p. 1928-1946, 2011.

Disponível em:

<<http://www.sciencedirect.com/science/article/pii/S0164121211001701>>.

Acesso em: Agosto 2015.

CHUGH, G.; BANSAL, D.; SOFAT, S. Road Condition Detection Using Smartphone Sensors: a Survey. **International Journal of Electronic and Electrical Engineering**, v. 7, n. 6, p. 595-601, 2014. Disponível em:

<http://www.irphouse.com/ijeee_spl/ijeeev7n6_09.pdf>. Acesso em: Agosto 2015.

COPPIN, B. **Inteligência Artificial**. Rio de Janeiro: LTC, 2010.

COULOURIS, G. F. et al. **Distributed systems: concepts and design**. 5^o Ed. ed. Massachusetts: Addison Wesley, 2012. ISBN: 978-0-13-214301-1.

CURINO, C. et al. Relational cloud: a database-as-a-service for the cloud. In: CONFERENCE ON INNOVATIVE DATA SYSTEMS RESEARCH, CIDR. **Proceedings**. Massachusetts: MIT. 21 Junho 2011. p. 235-240. Disponível em: <http://www.cidrdb.org/cidr2011/Papers/CIDR11_Paper33.pdf>. Acesso em: Agosto 2015.

DAS, T. et al. PRISM: platform for remote sensing using smartphones. In: INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES. **Proceedings**. San Francisco: ACM. 2010. p. 63-76. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=1814442&ftid=510032&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

DE ZOYSA, K. et al. A public transport system based sensor network for road surface condition monitoring. In: WORKSHOP ON NETWORKED SYSTEMS FOR DEVELOPING REGIONS. **Workshop**. Kyoto: ACM. 2007. p. 1-6.

Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=1326585&type=pdf&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

DEERING, S.; HINDEN, R. RFC 2460 - Internet Protocol, Version 6 (IPv6), Fremont, 1 Dezembro 1998. Disponível em: <<https://www.ietf.org/rfc/rfc2460.txt>>. Acesso em: Agosto 2015.

DEMIRKOL, I.; ERSOY, C.; ALAGOZ, F. MAC Protocols for Wireless Sensor Networks: a Survey. **IEEE Communications Magazine**, v. 44, n. 4, p. 115-121, 2006. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1632658>>. Acesso em: Agosto 2015.

DNIT. **Manual de Conservação Rodoviária**. Rio de Janeiro: Departamento Nacional de Infra-estrutura de Transportes, 2005. Disponível em: <http://ipr.dnit.gov.br/publicacoes/710_Manual_de_Conservacao_Rodoviaria.pdf>. Acesso em: Agosto 2015.

DURISIC, M. P. et al. A survey of military applications of wireless sensor networks. In: MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO). **Proceedings**. Bar, Montenegro: IEEE. 2012. p. 196-199. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6268958>>. Acesso em: Agosto 2015.

ECMA INTERNATIONAL. ECMAScript Language Specification, Geneva, 1 June 2011. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>>. Acesso em: Agosto 2015.

EISENMAN, S. B. et al. BikeNet: a mobile sensing system for cyclist experience mapping. **ACM Transactions on Sensor Networks**, Sidney, v. 6, n. 1, p. 1-15, 2009. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=1653766&ftid=729372&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

ERIKSSON, J. et al. The pothole patrol: using a mobile sensor network for road surface monitoring. In: INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES. **Proceedings**. Breckenridge: ACM. 2008. p. 29-39. Disponível em: <http://dl.acm.org/ft_gateway.cfm?id=1378605&ftid=526929&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

EUGSTER, P. T. et al. The many faces of publish/subscribe. **ACM Computing Surveys (CSUR)**, v. 35, n. 2, p. 114-131, 2003. Disponível em: <http://dl.acm.org/ft_gateway.cfm?id=857078&ftid=213413&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

FACELLI, K. et al. **Inteligencia Artificial - Uma Abordagem de Aprendizado de Máquina**. Rio de Janeiro: LTC, 2011.

FIELDING, R. et al. Hypertext Transfer Protocol -- HTTP/1.1, 1 Junho 1999. Disponível em: <<https://www.ietf.org/rfc/rfc2616.txt>>. Acesso em: Agosto 2015.

FLACH, P. **Machine learning**: the art and science of algorithms that make sense of data. Cambridge: Cambridge University Press, 2012.

FOGEL, J.; NEHMAD, E. Internet social network communities: risk taking, trust, and privacy concerns. **Computers in Human Behavior**, v. 25, n. 1, p. 153-160, 2009. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563208001519>>. Acesso em: Agosto 2015.

FRANKEL, S.; KRISHNAN, S. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap, Fremont, 1 Fevereiro 2011. Disponível em: <<https://tools.ietf.org/html/rfc6071>>. Acesso em: Agosto 2015.

GALLO, D. S. et al. Taxonomy of Citizen Sensing for Intelligent Urban Infrastructures. **Sensors Journal**, v. 14, n. 12, p. 4154-4164, 2014. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6908996>>. Acesso em: Agosto 2015.

- GANTI, R. K.; YE, F.; LEI, H. Mobile Crowdsensing: current State and Future Challenges. **Communications Magazine**, v. 49, n. 11, p. 32-39, 2011. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6069707>. Acesso em: Agosto 2015.
- GARCIA, R. L. et al. A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends. **Sensors**, v. 9, n. 6, p. 4728-4750, 2009. Disponivel em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.271.6829&rep=rep1&type=pdf>. Acesso em: Agosto 2015.
- GELERNTER, D. Generative communication in Linda. **ACM Transactions on Programming Languages and Systems**, v. 7, n. 1, p. 80-112, 1985. Disponivel em: http://dl.acm.org/ft_gateway.cfm?id=2433&ftid=24929&dwn=1&CFID=542076495&CFTOKEN=70550726. Acesso em: Agosto 2015.
- GILBERT, J. M.; BALOUCHI, F. Comparison of Energy Harvesting Systems for Wireless Sensor Networks. **International Journal of Automation and Computing**, v. 9, n. 4, p. 334-347, 2008. Disponivel em: <http://rd.springer.com/content/pdf/10.1007%2Fs11633-008-0334-2.pdf>. Acesso em: Agosto 2015.
- GONÇALVES, V.; BALLON, P. Adding value to the network: mobile operators' experiments with Software-as-a-Service and Platform-as-a-Service models. **Telematics and Informatics**, v. 28, n. 1, p. 12-21, 2011. Disponivel em: <http://www.sciencedirect.com/science/article/pii/S0736585310000365>. Acesso em: Agosto 2015.
- GOUVEIA, C. et al. Promoting the use of environmental data collected by concerned citizens through information and communication technologies. **Journal of Environmental Management**, v. 71, n. 2, p. 135-154, 2004. Disponivel em: <http://www.sciencedirect.com/science/article/pii/S0301479704000428>. Acesso em: Agosto 2015.
- GUTIERREZ, J. A. et al. IEEE 802.15. 4: a developing standard for low-power low-cost wireless personal area networks. **Network**, p. 2529-2553, 2001.

- HACHEM, S.; PATHAK, A.; ISSARNY, V. Service-oriented middleware for large-scale mobile participatory sensing. **Pervasive and Mobile Computing**, v. 10, p. 66-82, 2014. Disponivel em: <http://www.sciencedirect.com/science/article/pii/S157411921300134X>. Acesso em: Agosto 2015.
- HADIM, S.; MOHAMED, N. Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks. **IEEE distributed systems online**, v. 7, n. 3, p. 1-23, 2006. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1621014>. Acesso em: Agosto 2015.
- HAPNER, M. et al. **Java message service**. Santa Clara: Sun Microsystems Inc, 2002.
- HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES. **Proceedings**. Maui: IEEE. 2000. p. 10. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=926982>. Acesso em: Agosto 2015.
- HEMPSTEAD, M. et al. Survey of hardware systems for wireless sensor networks. **Journal of Low Power Electronics**, v. 4, p. 11-20, 2008. Disponivel em: https://www.eecs.harvard.edu/~dbrooks/hempstead_JOLPE08.pdf. Acesso em: Agosto 2015.
- HERRERA, J. C. et al. Evaluation of Traffic Data Obtained via GPS-Enabled Mobile Phones: the Mobile Century Field. **Transportation Research Part C: Emerging Technologies**, v. 18, n. 4, p. 568–583, 2010. Disponivel em: <http://www.sciencedirect.com/science/article/pii/S0968090X09001430>. Acesso em: Agosto 2015.
- HICKSON, I. et al. HTML5 - A vocabulary and associated APIs for HTML and XHTML, 28 October 2014. Disponivel em: <http://www.w3.org/TR/html5/>. Acesso em: Agosto 2015.

HOFMANN-WELLENHOF, B.; LICHTENEGGER, H.; WASLE, E. **GNSS—global navigation satellite systems: GPS, GLONASS, Galileo, and more.**

Mörlenbach: Springer Science & Business Media, 2007.

HOWE, J. The rise of crowdsourcing. **Wired Magazine**, v. 14, n. 6, p. 1-4, 2006.

Disponível em: <http://sistemas-humano-computacionais.wdfiles.com/local--files/capitulo%3Aredes-sociais/Howe_The_Rise_of_Crowdsourcing.pdf>.

Acesso em: Agosto 2015.

HOWE, J. **Crowdsourcing: How the power of the crowd is driving the future of business.** New York: Random House, 2008.

HUANG, P. et al. The evolution of MAC protocols in wireless sensor networks: a survey. **IEEE Communications Surveys & Tutorials**, n. 15, p. 101-120, 2013.

Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6188353>>. Acesso em: Agosto 2015.

HUNKELER, U.; TRUONG, H. L.; STANFORD-CLARK, A. MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks. In: INTERNATIONAL CONFERENCE ON COMMUNICATION SYSTEMS SOFTWARE AND MIDDLEWARE. **Workshop**. Bangalore: IEEE. 2008. p. 791-798. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4554519>>. Acesso em: Agosto 2015.

HUNTER, J.; CRAWFORD, W. **Java servlet programming.** Sebastopol: O'Reilly Media, Inc., 2001.

IBM CORPORATION. **Smarter Cities: introducing the IBM City Operations and Management Solution.** Armonk: IBM, 2011. Disponível em:

<<http://www.redbooks.ibm.com/redpapers/pdfs/redp4734.pdf>>. Acesso em: Agosto 2015.

IBM CORPORATION. Native, web or hybrid mobile-app development, New York, 1 Abril 2012. Disponível em: <<ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf>>. Acesso em: Agosto 2015.

IBM CORPORATION; EUROTECH. MQTT V3.1 Protocol Specification, 19 Agosto 2010. Disponível em: <<http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>>. Acesso em: Agosto 2015.

IEEE. **802.15.4-200 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)**. New York: IEEE, 2006.

JUANG, P. et al. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In: CONFERENCE ON ARCHITECTURAL SUPPORT FOR PROGRAMMING LANGUAGES AND OPERATING SYSTEMS. **Proceedings**. New York: ACM. 2002. p. 96-107. Disponível em: <http://dl.acm.org/ft_gateway.cfm?id=605408&ftid=109294&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

KARAM, A.; MOHAMED, N. Middleware for mobile social networks: a survey. In: INTERNATIONAL CONFERENCE ON SYSTEM SCIENCE (HICSS). **Proceedings**. Maui: IEEE. 2012. p. 1482-1490. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6149064>>. Acesso em: Agosto 2015.

KAUFMANN, N.; SCHULZE, T.; VEIT, D. More than fun and money. Worker Motivation in Crowdsourcing – A Study on Mechanical Turk. In: AMERICAS CONFERENCE ON INFORMATION SYSTEMS. **Proceedings**. Detroit: Association for Information Systems. 2011. p. 1-11. Disponível em: <http://raptor1.bizlab.mtsu.edu/S-Drive/DMORRELL/Mgmt%204990/Crowdsourcing/Kaufmann_et%20al_2011.pdf>. Acesso em: Agosto 2015.

KHAN, W. Z. et al. Mobile phone sensing systems: a survey. **Communications Surveys & Tutorials**, v. 15, n. 1, p. 402-427, 2013. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6177188>>. Acesso em: Agosto 2015.

KIM, S. et al. Health monitoring of civil infrastructures using wireless sensor networks. In: INFORMATION PROCESSING IN SENSOR NETWORKS. **Simpósio**. Cambridge: IEEE. 2007. p. 254-263. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4379685>>. Acesso em: Agosto 2015.

KOCK, F. et al. A Platform for Citizen Sensing in Sentient Cities. **Citizen in Sensor Networks**, v. 7685, p. 57-66, 2013. Disponível em:
<http://link.springer.com/content/pdf/10.1007%2F978-3-642-36074-9_6.pdf>. Acesso em: Agosto 2015.

KRISHNAMACHARI, B.; ESTRIN, D.; WICKER, S. The impact of data aggregation in wireless sensor networks. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS. **Workshop**. VIENNA: IEEE. 2002. p. 575-578. Disponível em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1030829>>. Acesso em: Agosto 2015.

KRISHNAMURTHY, L. et al. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In: INTERNATIONAL CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS, SENSYS. **Proceedings**. San Diego: ACM. 2005. p. 64-75. Disponível em:
<http://dl.acm.org/ft_gateway.cfm?id=1098926&ftid=334246&coll=DL&dl=GUIDE&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

KSHETRI, N. Cloud computing in developing economies. **IEEE Computer**, v. 43, n. 10, p. 47-55, 2010. Disponível em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5530325>>. Acesso em: Agosto 2015.

KUNDRA, V. **State of public sector cloud computing**. Washigton. 2009.

KUROSE, J.; ROSS, K. **Redes de Computadores e Internet**. 5ª Edição. ed. São Paulo: Addison Wesley, 2010.

LAMPKIN, V. et al. **Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry**. New York: International Business Machines Corporation, 2012.

LANE, N. D. et al. Urban sensing systems: opportunistic or participatory? In: WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS.

Workshop. New York: ACM. 2008. p. 11-16. Disponivel em:

<http://dl.acm.org/ft_gateway.cfm?id=1411763&type=pdf&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

LANE, N. D. et al. A Survey of Mobile Phone Sensing. **Communications Magazine**, v. 48, n. 9, p. 140-150, 2010. Disponivel em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5560598>>. Acesso em: Agosto 2015.

LI, K.; HUDAK, P. Memory coherence in shared virtual memory systems. **ACM**

Transactions on Computer Systems, v. 7, n. 4, p. 321-359, 1989. Disponivel em:

<http://dl.acm.org/ft_gateway.cfm?id=75105&ftid=15312&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

LI, M.; YANG, B. A Survey on Topology issues in Wireless Sensor Network. In:

INTERNATIONAL CONFERENCE ON WIRELESS NETWORKS. **Proceedings**.

Nevada: CSREA. 2006. p. 503-510. Disponivel em:

<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.7613&rep=rep1&type=pdf>>. Acesso em: Agosto 2015.

LIU, S.; TANG, Y.; LIU, Y. A survey of transport protocol for wireless sensor. In:

INTERNATIONAL CONFERENCE ON CONSUMER ELECTRONICS, COMMUNICATIONS AND NETWORKS. **Proceedings**. Yichang: IEEE. 2012.

p. 2338-2341. Disponivel em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6202037>>. Acesso em: Agosto 2015.

LOW, K. S.; WIN, W. N. N.; ER, M. J. Wireless sensor networks for industrial environments. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE FOR MODELLING, CONTROL AND AUTOMATION.

Proceedings. Vienna: IEEE. 2005. p. 271-276. Disponivel em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1631480>>. Acesso em: Agosto 2015.

MA, X.; LUO, W. The analysis of 6LowPAN technology. In: IEEE Pacific-Asia

WORKSHOP ON COMPUTATIONAL INTELLIGENCE AND INDUSTRIAL

APPLICATION. **Workshop**. Wuhan: IEEE. 2008. p. 963-966. Disponivel em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4756702>>. Acesso em: Agosto 2015.

MAASSEN, J. et al. An efficient implementation of Java's remote method invocation.

ACM Sigplan notices, v. 34, n. 8, p. 173-182, 1999. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=301120&ftid=30951&dwn=1&CFID=542076495&CFTOKEN=70550726>.

MAINWARING, A. et al. Wireless Sensor Networks for Habitat Monitoring. In:

INTERNATIONAL WORKSHOP ON WIRELESS SENSOR NETWORKS AND APPLICATIONS. **Workshop**. New York: ACM. 2002. p. 88-97. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=570751&ftid=85570&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

MARTINE, G.; MARSHALL, A. **State of world population 2007**: unleashing the potential of urban growth. In *State of world population 2007: Unleashing the potential of urban growth*. New York: UNFPA, 2007.

MELL, P.; GRANCE, T. **The NIST definition of cloud computing**. Gaithersburg. 2011.

MILUZZO, E. et al. Sensing Meets Mobile Social Networks: the Design, Implementation and Evaluation of the CenceMe Application. In: ACM CONFERENCE ON EMBEDDED NETWORK SENSOR SYSTEMS.

Proceedings. Raleigh: ACM. 2008. p. 337-350. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=1460445&ftid=587534&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

MITCHELL, T. M. **Machine learning**. Boston: McGraw-Hill Boston, 1997.

MOHAN, P.; PADMANABHAN, V. N.; RAMJEE, R. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: ACM CONFERENCE ON EMBEDDED NETWORK SENSOR SYSTEMS. **Proceedings**. Raleigh: ACM. 2008. p. 323-336. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=1460450&ftid=587538&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

MULIGAN, G.; GROUP, 6. W. The 6LoWPAN Architecture. In: WORKSHOP ON EMBEDDED NETWORKED SENSORS. **Workshop**. Cork: ACM. 2007. p. 78-

82. Disponivel em:

<http://dl.acm.org/ft_gateway.cfm?id=1278992&ftid=428171&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

MURTY, R. N. et al. Citysense: an urban-scale wireless sensor network and testbed. In: IEEE CONFERENCE ON TECHNOLOGIES FOR HOMELAND SECURITY. **Proceedings**. Waltham: IEEE. 2008. p. 583-588. Disponivel em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4534518>>. Acesso em: Agosto 2015.

NAM, T.; PARDO, T. A. Conceptualizing smart city with dimensions of technology, people, and institutions. In: INTERNATIONAL DIGITAL GOVERNMENT RESEARCH CONFERENCE - DIGITAL GOVERNMENT INNOVATION IN CHALLENGING TIMES. **Proceedings**. College Park: ACM. 2011. p. 282-291. Disponivel em: <http://dl.acm.org/ft_gateway.cfm?id=2037602&ftid=1035791&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

NAPHADE, M. et al. Smarter cities and their innovation challenges. **Computer**, n. 44, p. 32-39, 2011. Disponivel em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5875937>>. Acesso em: Agosto 2015.

OMOKARO, O. A Framework to Promote User Engagement in Participatory Sensing Applications. In: CONFERENCE ON UBIQUITOUS COMPUTING. **Proceedings**. New York: ACM. 2012. p. 548-551. Disponivel em: <http://dl.acm.org/ft_gateway.cfm?id=2370306&ftid=1290134&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

PAN, B. et al. Crowd Sensing of Traffic Anomalies based on Human Mobility and Social Media. In: INTERNATIONAL CONFERENCE ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS. **Proceedings**. Orlando: ACM. 2013. p. 334-343. Disponivel em: <http://dl.acm.org/ft_gateway.cfm?id=2525343&ftid=1421499&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

PARADIS, L.; HAN, Q. Dealing with Faults in Wireless Sensor Networks. **Journal of Network and Systems Management**, v. 15, p. 171-190, 2007. Disponivel em:

<<http://inside.mines.edu/~qhan/students/Lilia/docs/WSNFaultsSurvey.pdf>>.

Acesso em: Agosto 2015.

PAULOS, E.; HONICKY, R. J.; GOODMAN, E. Sensing Atmosphere. **Human-Computer Interaction Institute**, v. 1, p. 201-204, 2007. Disponivel em: <<http://repository.cmu.edu/cgi/viewcontent.cgi?article=1202&context=hcii>>. Acesso em: Agosto 2015.

PERERA, C. et al. Capturing sensor data from mobile phones using global sensor network middleware. In: INTERNATIONAL SYMPOSIUM ON PERSONAL INDOOR AND MOBILE RADIO COMMUNICATIONS (PIMRC). **Simpósio**. Sydney: IEEE. 2012. p. 24-29. Disponivel em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6362778>>. Acesso em: Agosto 2015.

PERERA, C. et al. Mosden: an internet of things middleware for resource constrained mobile devices. In: INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HICSS). **Proceedings**. Waikoloa: IEEE. 2014. p. 1053-1062. Disponivel em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6758734>>. Acesso em: Agosto 2015.

PERERA, C. et al. Sensing as a service model for smart cities supported by internet of things. **Transactions on Emerging Telecommunications Technologies**, v. 25, n. 1, p. 81-93, 2014. Disponivel em: <<http://onlinelibrary.wiley.com/doi/10.1002/ett.2704/epdf>>. Acesso em: Agosto 2015.

PEREZ, A. J.; LABRADOR, M. A.; BARBEAU, S. J. G-sense: a scalable architecture for global sensing and monitoring. **Network**, v. 24, n. 4, p. 57-64, 2010. Disponivel em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5510920>>. Acesso em: Agosto 2015.

PIRO, G. et al. Information centric services in Smart Cities. **Journal of Systems and Software**, v. 88, p. 169-188, 2013. Disponivel em: <<http://www.sciencedirect.com/science/article/pii/S0164121213002586>>. Acesso em: Agosto 2015.

- RA, M. R. et al. Medusa: a programming framework for crowd-sensing applications. In: INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES. **Proceedings**. Low Wood Bay: ACM. 2012. p. 337-350. Disponivel em: http://dl.acm.org/ft_gateway.cfm?id=2307668&ftid=1257707&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.
- RANA, R. K. et al. Ear-Phone: an End-to-End Participatory. In: INFORMATION PROCESSING IN SENSOR NETWORKS. **Proceedings**. Stockholm: ACM. 2010. p. 105-116. Disponivel em: http://dl.acm.org/ft_gateway.cfm?id=1791226&ftid=809193&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.
- REDDY, S.; ESTRIN, D.; SRIVASTAVA, M. Recruitment Framework for Participatory Sensing Data Collections. **Pervasive Computing**, v. 6030, p. 138-155, 2010. Disponivel em: http://link.springer.com/content/pdf/10.1007%2F978-3-642-12654-3_9.pdf>. Acesso em: Agosto 2015.
- REESE, G. **Database programming with JDBC and Java**. Sebastopol: O'Reilly Media, Inc, 2000.
- REN, Y.; WEN, G.; LI, X. An SVM based algorithm for road disease detection using accelerometer. **TELKOMNIKA Indonesian Journal of Electrical Engineering**, v. 11, n. 9, p. 5169-5175, 2013. Disponivel em: <http://www.iaesjournal.com/online/index.php/TELKOMNIKA/article/view/3264/pdf>>. Acesso em: Agosto 2015.
- RIMAL, B. P.; CHOI, E.; LUMB, I. A taxonomy and survey of cloud computing systems. In: INTERNATIONAL JOINT CONFERENCE ON INC, IMS AND IDC. **Proceedings**. Seoul: IEEE. 2009. p. 44-51. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5331755>>. Acesso em: Agosto 2015.
- ROMER, K.; KASTEN, O.; MATTERN, F. Middleware challenges for wireless sensor networks. **ACM SIGMOBILE Mobile Computing and Communications Review**, v. 6, n. 4, p. 59-61, 2002. Disponivel em: http://dl.acm.org/ft_gateway.cfm?id=643556&ftid=149090&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, n. 6, p. 386-408, 1958. Disponível em: http://web.stanford.edu/class/psych209a/ReadingsByDate/01_30/Rosenblatt58Perceptron.pdf. Acesso em: Agosto 2015.
- SANCHEZ, L. et al. SmartSantander: internet of Things Research and Innovation through Citizen Participation. **Future Network & Mobile Summit**, v. 7858, p. Springer, 2011. Disponível em: http://link.springer.com/content/pdf/10.1007%2F978-3-642-38082-2_15.pdf. Acesso em: Agosto 2015.
- SANCHEZ, L. et al. SmartSantander: IoT experimentation over a smart city testbed. **Computer Networks**, v. 61, n. 14, p. 217–238, 2013. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128613004337>. Acesso em: Agosto 2015.
- SHELBY, Z.; BORMANN, C. **6LoWPAN**: The wireless embedded Internet. Chippenham: John Wiley & Sons, 2011.
- SHEPARD, M. **Sentient city**: Ubiquitous computing, architecture, and the future of urban space. Massachusetts: MIT Press, 2011.
- SILVA, A. F. et al. Simulação de Microdrenagem em Cidades Conscientes. In: Congresso Brasileiro de Automática. **Proceedings**. Belo Horizonte: SBA. 2014. p. 2414-2421. Disponível em: <http://www.eletrica.ufpr.br/anais/cba/2014/PDF/1569927177.pdf>. Acesso em: Agosto 2015.
- SILVA, A. F. et al. A Cloud-based Architecture for Internet of Things, Targeting Industrial Engine Remote Monitoring. In: INTERNATIONAL SYMPOSIUM ON HIGH-PERFORMANCE COMPUTER ARCHITECTURE - SENSOR TO CLOUD ARCHITECTURES WORKSHOP. **Workshop**. California: IEEE. 2015. p. 1-6.
- SOHRABY, K.; MINOLI, D.; ZNATI, T. **Wireless sensor networks**: technology, protocols, and applications. Hoboken: John Wiley & Sons, 2007.
- SONG, J. et al. WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. In: REAL-TIME AND EMBEDDED TECHNOLOGY AND

APPLICATIONS SYMPOSIUM. **Simpósio**. Saint Louis: IEEE. 2008. p. 377-386. Disponível em:
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4550808>. Acesso em: Agosto 2015.

STANFORD-CLARK, A.; TRUONG, H. L. MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2, Zurich, 14 Novembro 2013. Disponível em:
http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf. Acesso em: Agosto 2015.

STEVENS, M. **Community memories for sustainable societies. The case of environmental noise**. Ixelles: Vrije Universiteit Brussel, 2012. Disponível em:
<http://brussense.be/phd-matthias/Thesis-final.pdf>. Acesso em: Agosto 2015.

STEVENS, M.; D'HONDT, E. Crowdsourcing of Pollution Data using Smartphones. In: WORKSHOP ON UBIQUITOUS CROWDSOURCING. **Workshop**. Copenhagen: ACM. 2010. p. 1-4. Disponível em:
<http://soft.vub.ac.be/Publications/2010/vub-tr-soft-10-15.pdf>. Acesso em: Agosto 2015.

STOIANOV, I. et al. PIPENET: a wireless sensor network for pipeline monitoring. In: INTERNATIONAL SYMPOSIUM ON INFORMATION PROCESSING IN SENSOR NETWORKS. **Simpósio**. Cambridge: IEEE. 2007. p. 264-273. Disponível em:
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4379686>. Acesso em: Agosto 2015.

TANENBAUM, A.; VAN STEEN, M. **Distributed Systems**. New Jersey: Pearson Prentice Hall, 2007.

THAIN, D.; TANNENBAUM, T.; LIVNY, M. Distributed computing in practice: the Condor experience. **Concurrency and Computation: Practice and Experience**, v. 17, n. 2, p. 323-356, 2005. Disponível em:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.6.3035&rep=rep1&type=pdf>. Acesso em: Agosto 2015.

THE INTERNATIONAL SOCIETY OF AUTOMATION. **ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: process control and related applications**. Durham: ISA, 2011.

- TUBAISHAT, M. et al. Wireless sensor networks in intelligent transportation systems. **Wireless communications and mobile computing**, v. 9, n. 3, p. 287-302, 2009. Disponivel em: <http://onlinelibrary.wiley.com/doi/10.1002/wcm.616/epdf>. Acesso em: Agosto 2015.
- VAN KREVELEN, D. W. F.; POELMAN, R. A Survey of Augmented Reality Technologies, Applications and Limitations. **International Journal of Virtual Reality**, v. 9, n. 2, p. 1-20, 2010. Disponivel em: http://cierto.org/pdf/realidad_aumentada_historia.pdf. Acesso em: Agosto 2015.
- VAPNIK, V. N. **The nature of statistical learning theory**. New York: Springer-Verlag, 1995.
- VINOSKI, S. CORBA: integrating diverse applications within distributed heterogeneous environments. **Communications Magazine**, v. 35, n. 2, p. 46-55, 1997. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=565655>. Acesso em: Agosto 2015.
- WANG, C. et al. A Survey of Transport Protocols for Wireless Sensor Networks. **IEEE Network**, v. 20, n. 3, p. IEEE, 2006. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1637930>. Acesso em: Agosto 2015.
- WANG, M. M. et al. Middleware for wireless sensor networks: A survey. **Journal of computer science and technology**, v. 23, n. 3, p. 305-326, 2008. Disponivel em: <http://link.springer.com/content/pdf/10.1007%2Fs11390-008-9135-x.pdf>. Acesso em: Agosto 2015.
- WANG, Y.; ATTEBURY, G.; RAMAMURTHY, B. A survey of security issues in wireless sensor networks. **IEEE Communications Surveys & Tutorials**, v. 8, n. 2, p. 2-23, 2006. Disponivel em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4109893>. Acesso em: Agosto 2015.
- WASSERMAN, A. I. Software engineering issues for mobile application development. In: FUTURE OF SOFTWARE ENGINEERING RESEARCH. **Proceedings**.

Santa Fe: ACM. 2010. p. 397-400. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=1882443&ftid=863459&coll=DL&dl=GUIDE&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

YE, F. et al. MECA: mobile Edge Capture and Analysis Middleware for Social Sensing Applications. In: INTERNATIONAL CONFERENCE COMPANION ON WORLD WIDE WEB. **Proceedings**. Lyon: ACM. 2012. p. 699-702. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=2188184&ftid=1205807&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

YE, W.; HEIDEMANN, J.; ESTRIN, D. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In: INFOCOM. **Proceedings**. New York: IEEE. 2002. p. 1567 - 1576. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1019408>>. Acesso em: Agosto 2015.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey.

Computer Networks, v. 52, n. 12, p. 2292–2330, 14 Abril 2008. Disponível em:

<<http://www.sciencedirect.com/science/article/pii/S1389128608001254>>.

Acesso em: Agosto 2015.

ZARKO, I. P.; ANTONIC, A.; PRIPUZIC, K. Publish/subscribe middleware for energy-efficient mobile crowdsensing. In: CONFERENCE ON PERVASIVE AND UBIQUITOUS COMPUTING ADJUNCT PUBLICATION. **Proceedings**. Zurich: ACM. 2013. p. 1099-1100. Disponível em:

<http://dl.acm.org/ft_gateway.cfm?id=2499577&ftid=1393273&dwn=1&CFID=542076495&CFTOKEN=70550726>. Acesso em: Agosto 2015.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of internet services and applications**, v. 1, n. 1, p. 7-18, 2010. Disponível em:

<<http://rd.springer.com/content/pdf/10.1007%2Fs13174-010-0007-6.pdf>>.

Acesso em: Agosto 2015.

ZHANG, X.; FRESCHL, J. L.; SCHOPF, J. M. A performance study of monitoring and information services for distributed systems. In: IEEE INTERNATIONAL SYMPOSIUM ON HIGH PERFORMANCE DISTRIBUTED COMPUTING.

Simpósio. Seattle: IEEE. 2003. p. 270-281. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1210036>>. Acesso em: Agosto 2015.

ZHOU, S. et al. UTOPIA: a Load Sharing Facility for Large, Heterogeneous

Distributed Computer Systems. **Software: Practice and Experience**, p. 1305-1336, 1993. Disponível em:

<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.2991&rep=rep1&type=pdf>>. Acesso em: Agosto 2015.

ZIGBEE ALLIANCE. Specifications, Agosto 2014. Disponível em:

<<http://www.zigbee.org/Specifications.aspx>>. Acesso em: 13 Maio 2014.

APÊNDICE

A. O padrão IEEE 802.15.4

IEEE 802.15.4. É um padrão desenvolvido para aplicações com baixa transferência de informações, alcance e consumo de energia, sendo idealmente usado em RSSF (IEEE, 2006) (GUTIERREZ, NAEVE, *et al.*, 2001). O IEEE 802.15.4 define apenas as características da camada **física** e **enlace**. A tabela 2 apresenta as principais características definidas no padrão.

Camada física: Ativação / desativação do transceptor, detecção da energia recebida no canal, indicador de qualidade de conexão, seleção de canal, detector de canal livre e transmissão e recepção de pacotes pela camada física.

Camada enlace: Gerenciamento de *beacon*, acesso ao canal, gerenciamento de intervalos de tempo (*slot time*), validação de quadros, reconhecimento de entrega de quadros e associação na RSSF.

Tabela 2: Principais características definidas pelo padrão IEEE 802.15.4

Camada física			
Frequência de operação (MHz)	2450	915	868
Taxa de transmissão (Kbytes/segundo)	250	40	20
Quantidade de canais	16	10	1
Banda de comunicação (MHz)	5	2	0,6
Tipo de modulação	O-QPSK	BPSK	BPSK
Bit por símbolo	4	1	1
Sensibilidade de recepção	-85 dBm		
Camada de enlace			
Tamanho do quadro de comunicação	127 bytes		
Topologias de rede	Estrela e ponto-a-ponto		
Estratégia de hibernação dos nós	<i>Beacon</i> ou sem hibernação		
Estratégia de evitar colisão de pacotes	CSMA/CA		
Criptografia	AES-128		

Fonte: Autor

As vantagens do padrão são a facilidade de instalação, a confiabilidade na transmissão de dados, operação em baixo alcance, baixo custo do dispositivo e razoável tempo de duração da bateria, enquanto mantém uma simples e flexível pilha de protocolos (BARONTI, PILLAI, *et al.*, 2007).

O padrão suporta duas classes de dispositivos:

- i) **FFD (Full Functional Device):** Todas as funções do IEEE 802.15.4 estão incluídas e pode operar como coordenador de interconexão entre os nós, um simples nó sensor ou ainda como o próprio sorvedouro da rede

- ii) **RFD (Reduced Functional Devices)**: O dispositivo possui as funções apenas para coletar dados e se comunicar com os nós do tipo FFD.

As formas de topologia possíveis são a estrela e ponto-a-ponto. A figura 59 representa os arranjos possíveis entre os nós.

Figura 59: Exemplo de topologia de RSSF

Fonte: Adaptado de IEEE, 2006

Uma importante função suportada para economia de energia é o modo *beacon* (BARONTI, PILLAI, *et al.*, 2007). O *beacon* é um pacote de dados especial que contém informações da identificação do nó coordenador, do período de sincronização e da estrutura do super quadro. Quando ativo, age como mecanismo de sincronização entre os nós da rede. O funcionamento parte do princípio que os nós estão inicialmente acordados. Em seguida, os *beacons* são transmitidos pelos nós coordenadores e assim que os demais nós o recebem, verificam se é o turno corrente para própria comunicação. Caso positivo, aguarda o intervalo de tempo apropriado (*slot time*) para transmissão, ao contrário, entram em modo de hibernação até o período do próximo *beacon*.

Figura 60: Detalhes do super quadro do IEEE 802.15.4

Fonte: BARONTI, et al, 2007

O nó sorvedouro pode operar utilizando ou não o super quadro, figura 60. O super quadro é definido como o intervalo de dois *beacons*. Ele é subdividido em intervalos de tempo chamado de GTS (*Guaranteed Time Slots*), organizados em três períodos: CAP (*Contention Access Period*), CFP (*Contention-Free Period*) e período inativo. O período CAP é quando acontece a competição dos nós pelo meio, principalmente no processo de associação do nó na rede ou quando a rede não opera em modo *beacon*. O período CFP é composto por período com intervalos de tempo restritos para um nó exclusivamente transmitir dados, importante para aplicações que requeiram QoS. Já o período inativo é quando os nós da RSSF estão em modo hibernação e economizam energia.

O pacote de comunicação pode conter até 128 bytes e possui características bem definidas. A figura 61 representa conteúdo do quadro MAC.

Figura 61: Detalhes do quadro de dados do IEEE 802.15.4 operando em 2450MHz

Fonte: Adaptado de IEEE, 2006

B. O protocolo 6LoWPAN

6LoWPAN. É uma definição de protocolo para habilitar o endereçamento de pacotes IPv6 em redes de baixo consumo de energia (MULIGAN e GROUP, 2007). O acrônimo é proveniente da junção das palavras inglesas, *IPv6 over Low power Wireless Personal Area Network*. O conceito foi criado pelo grupo de trabalho de mesmo nome, 6LoWPAN, do IETF (*Internet Engineering Task Force*), o qual vislumbrava que o IP poderia e deveria ser aplicado também a dispositivos pequenos, como os nós das redes de sensores (MULIGAN e GROUP, 2007).

A popularidade do IP, as diversas ferramentas desenvolvidas e já difundidas, além da capacidade de 128 bits de endereçamento do IPv6 estão entre as vantagens que motivaram os esforços na adaptação do IP para rede de sensores sem fio (SHELBY e BORMANN, 2011). Além disso, o 6LoWPAN pode proporcionar características favoráveis ao uso como o menor tamanho do código, a menor carga adicional do cabeçalho e a utilização de protocolos de transporte populares, por exemplo, o UDP e TCP, em comparação a alguns padrões disponíveis demonstrado na tabela 3 (MULIGAN e GROUP, 2007).

Tabela 3: Comparação entre 6LoWPAN, Zigbee e Zensys

	Zigbee	Zensys	6LoWPAN
Tamanho do código em modo malha	32K até 64K	32K	22K
RAM	8K	<2K	4K
Carga adicional do cabeçalho	8 a 16 bytes	Proprietário	2 a 11 bytes
Tamanho da rede	~65K	232	2 ⁶⁴
Suporte de rádio	802.15.4	Proprietário	802.15.4
Camada de transporte	Nenhuma	Proprietário	UDP / TCP
Conexão com Internet	Zigbee Gateway	Zensys Gateway	Roteador

Fonte: Adaptado de MULIGAN e GROUP, 2007

O 6LoWPAN possui como pontos chaves a **coordenação com o IPv6 e o IEEE 802.15.4, atribuição e gerenciamento de endereços, gerenciamento da rede e segurança** (MA e LUO, 2008).

Coordenação com o IPv6 e o IEEE 802.15.4. Embora o 6LoWPAN também funcione em outros padrões como o WirelessHART (SONG, HAN, *et al.*, 2008) e o ISA100 (THE INTERNATIONAL SOCIETY OF AUTOMATION, 2011), foi construído baseado no funcionamento do IEEE 802.15.4 (SHELBY e BORMANN, 2011). Como representado na tabela 3, é possível conter no padrão IEEE 802.15.4 uma carga de dados de apenas 82 bytes, enquanto que no IPv6 o tamanho mínimo do quadro é 1280 bytes (DEERING e HINDEN, 1998). Logo, o 6LoWPAN foi introduzido na pilha de protocolos como uma camada de adaptação para compressão do IPv6 (MA e LUO, 2008), figura 62.

Figura 62: Modelo de referência da pilha de camadas para o protocolo 6LoWPAN

Fonte: Adaptado de MA e LUO, 2008

Atribuição e gerenciamento de endereços. Segundo MULIGAN e GROUP (2007), o protocolo foi construído com o conceito “de pagar pelo que será usado”. Em outras palavras, quando são necessários os nós comunicarem localmente, a sobrecarga de dados imposto pelo protocolo será mínima, neste caso, apenas de 2 bytes. Porém, quando é necessário um endereçamento para um destino na internet, pode existir a necessidade de incluir também 40 bytes do IPv6. A figura 63 indica as configurações mínima e completa usando o 6LoWPAN no quadro IEEE 802.15.4, com a camada de transporte UDP.

Figura 63: Configuração mínima e completa usando 6LoWPAN no quadro IEEE 802.15.4

Fonte: Adaptado de SHELBY e BORMANN, 2011

Gerenciamento da rede. A figura 64 demonstra uma representação do encaminhamento local de pacotes de uma RSSF em malha com o protocolo 6LoWPAN. Neste caso, o cabeçalho contendo o endereço do destino é comprimido de modo a indicar a informação do próximo nó da rede ao destino final. Porém, quando necessário, os nós podem ser localizados pelo endereço IPv6 como um dispositivo comum em redes IP.

Figura 64: Representação do encaminhamento local de pacotes em uma RSSF em malha com o protocolo 6LoWPAN

Fonte: Adaptado de SHELBY e BORMANN, 2011

Segurança. Embora os padrões que operam nas camadas abaixo do 6LoWPAN já possuam mecanismos de segurança, como o próprio IEEE 802.15.4 que usa o AES128, o 6LoWPAN fornece adicionalmente segurança pela camada de rede, por exemplo, disponibilizando o IPsec (FRANKEL e KRISHNAN, 2011). Mesmo considerando que os nós em uma RSSF são mais suscetíveis de acesso físico do que em redes convencionais (WANG, ATTEBURY e RAMAMURTHY, 2006), estas ferramentas de segurança adicionais e já existentes do IP, aumentam globalmente a segurança dos dados nas RSSF (SHELBY e BORMANN, 2011).

C. Message Queuing Telemetry Transport – MQTT

O MQTT (*Message Queuing Telemetry Transport*) é uma implementação do paradigma de comunicação de sistema distribuído publicar e assinar baseado em tópicos, desenvolvido para aplicações de telemetria. É definido como um protocolo da camada de aplicação que assume que a rede irá prover uma conexão fim a fim e um serviço de auto segmentação de dados, como o TCP/IP (IBM CORPORATION; EUROTECH, 2010). Também, segundo os desenvolvedores, o objetivo é que o cliente seja o mais simples possível de modo que toda complexidade do protocolo esteja embutida no *broker*.

No MQTT são empregados três níveis de QoS (Quality of Service):

- i) Nível 0: As mensagens são entregues sem nenhum tipo de confirmação de recebimento;
- ii) Nível 1: As mensagens são retransmitidas até o destinatário confirmar o recebimento;
- iii) Nível 2: No nível 1 pode existir na rede diversas cópias da mesma mensagem, o que em certas aplicações pode ser inaceitável, já o nível 2 de QoS é definido para enviar apenas uma mensagem e aguardar a confirmação do destinatário.

O *broker* supervisiona o tempo vida da conexão por um período configurável, denominado *keep-alive*. Se após este período não houver nenhuma mensagem disseminada pelo cliente, ele enviará um comando PING ao *broker*, ao qual devolverá ao cliente um reconhecimento a este comando. Esta função é o mecanismo de manutenção de conexão do MQTT (IBM CORPORATION; EUROTECH, 2010). O MQTT também possui um mecanismo opcional para indicar as ações tomadas pelo destinatário ou remetente, sobre uma desconexão anormal de um cliente, denominado *Last Will Testament*.

Tabela 4: Comparação entre MQTT e HTTP

	MQTT	HTTP
Orientação de projeto	Centrado a dados	Centrado a documentos
Paradigma de comunicação	Publicar e assinar	Requisição – Resposta
Tamanho da mensagem	Pequena. Cabeçalho com 2 bytes	Grande. Todo o protocolo é baseado em texto
Níveis de serviços	3 configurações de QoS	Todas as mensagens têm o mesmo nível de serviço
Bibliotecas adicionais	30KB para C e 100KB para Java	Depende da aplicação (XML, JSON)
Distribuição de dados	1 para zero, 1 para 1, 1 para n	1 para 1

Fonte: Autor

Cada mensagem MQTT possui 2 bytes de controle que indicam o nível de QoS para sinalizar se a mensagem precisar ser retida no *broker*, enquanto o destinatário não processa a mesma, o tipo de mensagem (conexão, publicar, assinar, etc.) e quantos bytes ainda têm o pacote de informação.

Devido à popularidade para aplicações de internet do protocolo HTTP é conveniente caracterizar as diferenças entre o MQTT e o HTTP. A tabela 4 indica pontos de comparação (LAMPKIN, LEONG, *et al.*, 2012).

D. Message Queuing Telemetry Transport for Wireless Sensor Networks – MQTT-SN

O MQTT-SN (*Message Queuing Telemetry Transport for Wireless Sensor Networks*) é uma variação do protocolo MQTT desenvolvido por HUNKELER, TRUONG, e STANFORD-CLARK (2008), para operar em dispositivos com restrições de energia e banda de comunicação, por exemplo, como os dispositivos para rede de sensores sem fio. Também segundo os autores, o protocolo prevê escalabilidade da rede de sensores e fácil integração com a rede tradicional.

Devido às características intrínsecas de falhas dos nós em uma rede de sensores por falta de energia ou pela degradação do canal de comunicação, a aplicação não necessariamente necessita ter consciência de falhas ou mudanças dos nós. A aplicação apenas precisa dos dados a partir do momento que estiverem disponíveis. Com este tipo de premissa, o MQTT-SN foi projetado a ser um protocolo centrado nos dados (HUNKELER, TRUONG e STANFORD-CLARK, 2008).

Figura 65: Arquitetura MQTT-SN

Fonte: Adaptado de HUNKELER, TRUONG e STANFORD-CLARK, 2008

A arquitetura possui além dos clientes, o *gateway* e o *broker*, figura 65. O *gateway* é o responsável pela tradução das mensagens MQTT-SN, provindas da RSSF, para o MQTT e vice-versa. Ele se relaciona com os clientes em modo transparente ou agregação dos dados. Embora o modo *gateway* transparente seja mais fácil de implantação no cliente, não é escalável, uma vez que cada nó da rede necessita de uma conexão com o *broker*. Por outro lado, no modo de agregação, existindo uma falha no canal de comunicação com o *broker*, todos os nós ligados a este *gateway*, estarão também desconectados da rede, figura 66.

Para os clientes se conectarem ao *gateway*, ou sorvedouro, eles iniciam o processo de descoberta, chamado de SEARCHGW. Este processo é transmitido por *broadcast* na rede e quando um *gateway* recebe esta mensagem, responde com uma mensagem denominada GWINFO. O cliente que iniciou a comunicação e agora possui os endereços dos *gateways* os quais responderam a mensagem SEARCHGW, irá responder à mensagem GWINFO finalizando o processo de descoberta.

Figura 66: Modos de funcionamento do gateway MQTT-SN

Fonte: Adaptado de HUNKELER, TRUONG e STANFORD-CLARK, 2008

O MQTT-SN opera sobre a arquitetura IEEE 802.15.4. Devido a carga de dados necessária para funcionamento das camadas de comunicação anteriores como a de enlace e rede, a aplicação irá usufruir de aproximadamente 60 bytes para comunicação. O processo de conexão atende a esta restrição, enviando separadamente as informações de conexão que podem ser grandes: CLIENTID, WILLTOPIC e WILLMSG. Já no processo de publicação dos dados, o cliente transmite inicialmente o nome do tópico e então o gateway responde com um *ID* correspondente a este tópico, gerado de uma tabela local do gateway. Finalmente, o cliente responde a esta mensagem com o *ID* e também com os dados referentes ao tópico que deseja transmitir. As figuras 67 e 68 ilustram o processo de conexão e publicação mencionadas.

Figura 67: Procedimento de conexão MQTT-SN

Fonte: Adaptado de HUNKELER, TRUONG e STANFORD-CLARK, 2008

Figura 68: Procedimento de publicação do MQTT-SN

Fonte: Adaptado de HUNKELER, TRUONG e STANFORD-CLARK, 2008

Embora não implementado na primeira versão do MQTT-SN todos os mecanismos de QoS e economia de energia, como o modo de hibernação, a distribuição atual já contempla estas funcionalidades (STANFORD-CLARK e TRUONG, 2013).

E. Computação em Nuvem

A computação em nuvem (em inglês, *cloud computing*) é um modelo que por meio de uma rede de dados conveniente, ubíqua e sob demanda, um consumidor utiliza um conjunto compartilhado de recursos computacionais configuráveis, por exemplo, processamento, armazenamento, aplicações e serviços. Estes recursos podem ser rapidamente provisionados e liberados por ele, com um mínimo esforço de gerenciamento ou interação com o provedor de serviços (MELL e GRANCE, 2011). Para os autores as características essenciais de computação em nuvem são:

Auto-serviço sob demanda. O consumidor pode provisionar capacidades de computação, tais como, o tempo de utilização do servidor e armazenamento em rede, de modo automatizado pelo provedor de serviço e conforme sua necessidade.

Ampla acesso à rede. Os recursos de computação estão disponíveis pela rede e são acessados através de mecanismos que promovem o uso de dispositivos heterogêneos. Mesmo até clientes com baixo recursos computacionais, por exemplo, *smartphones*, *tablets*, *notebooks* e estações de trabalho.

Utilização compartilhada de recursos. Existem diversos recursos computacionais disponíveis pelo provedor para servir aos múltiplos consumidores com as mais variadas demandas, de modo compartilhado.

Elasticidade. O consumidor pode aumentar ou diminuir os recursos a qualquer momento conforme sua necessidade, em alguns casos automaticamente.

Serviço mensurável. O uso dos recursos podem ser monitorados, controlados e relatados, proporcionando transparência tanto para o provedor, quanto para o consumidor do serviço.

Existem algumas preocupações para utilizar uma aplicação em um ambiente de computação em nuvem, por exemplo, o custo, a confiabilidade ou mesmo a segurança do serviço. Os sistemas de computação em nuvem podem ser especializados para atender tais diversidades de demanda (ZHANG, CHENG e BOUTABA, 2010):

Privado - São nuvens construídas exclusivamente para um único usuário, por exemplo, uma empresa. Devido a infraestrutura utilizada ser exclusiva, existe o controle total do usuário sobre como o sistema é acessado, distribuído e sobre as aplicações implementadas.

Público – Neste tipo de nuvem os provedores oferecem seus recursos como serviços para o público em geral. O investimento inicial nulo de infraestrutura computacional é um importante agente motivador para adesão a este tipo de modalidade, porém como não há um controle refinado sobre as configurações de rede, a segurança aos dados mantidos na aplicação é comprometida.

Híbrido - Nas nuvens híbridas os modelos de nuvens públicas e privadas são mesclados. Este tipo de nuvem permite que os recursos alocados para a nuvem privada possam ser ampliados a partir de uma reserva de recursos em uma nuvem pública.

Um provedor pode entregar os serviços de computação em nuvem por diversos modelos, os quais estão relacionados com as camadas da arquitetura de um sistema em nuvem, figura 69. A popularização de sistemas em nuvem atualmente criou diversas especificidades destes modelos: Sensoriamento como Serviço – S2aaS (PERERA, ZASLAVSKY, *et al.*, 2014), Banco de Dados como Serviço – DBaaS (CURINO, JONES, *et al.*, 2011), Rede como Serviço – NaaS (GONÇALVES e BALLON, 2011). Os modelos de serviços mais populares são (RIMAL, CHOI e LUMB, 2009):

Software como Serviço (SaaS). A capacidade fornecida ao consumidor é a utilização de aplicações sob demanda. Estas aplicações são disponibilizadas para acesso dos dispositivos clientes por diversas maneiras, as mais comuns são os navegadores de internet ou uma interface de programa.

Plataforma como Serviço (PaaS). Neste modelo de serviço, a capacidade fornecida ao consumidor é construir aplicativos usando linguagens de programação, bibliotecas, serviços e ferramentas suportadas pelo provedor.

Infraestrutura como Serviço (IaaS). Infraestrutura como serviço é a capacidade fornecida ao consumidor para provisionar processamento, armazenamento, rede e outros recursos computacionais fundamentais em que o consumidor é capaz de implantar e executar softwares arbitrários, que pode incluir também sistemas operacionais e aplicativos.

Figura 69: Modelos de negócios

Fonte: Adaptado de ZHANG, CHENG e BOUTABA, 2010

As vantagens da utilização de computação em nuvem em cidades foi analisada em 30 casos de uso pelo governo dos Estados Unidos e destaca-se (KUNDRA, 2009):

Econômica. A computação em nuvem é uma abordagem de pagamento sob demanda em TI, com um baixo investimento inicial. Desta forma, o fluxo de caixa da cidade encaixa-se melhor com o custo do sistema.

Flexibilidade. Quando é vislumbrado pelos departamentos de TI flutuações na utilização dos recursos computacionais da cidade, eles não precisam de grandes esforços operacionais para garantir *hardware* e *software* adicional.

Implementação rápida. Sem a necessidade de passar por processos de contratação e certificação adicional e com uma vasta quantidade de serviços, ferramentas e recursos de TI, a computação em nuvem ajuda a projetos públicos iniciarem rapidamente.

Serviço consistente. A computação em nuvem oferece um nível superior de serviço e confiabilidade e uma resposta imediata a situações de emergência do que manter hospedados os recursos por responsabilidade da cidade.

Eficácia. A computação em nuvem libera os gestores da cidade dos detalhes mais operacionais de configuração e manutenção do sistema de TI, permitindo estes se concentrar mais nas tarefas de missão crítica da cidade.

Eficiência energética. Como os recursos estão reunidos, cada cidade não precisa ter sua própria infraestrutura de TI dedicada. Várias cidades podem compartilhar recursos de computação, levando a maiores taxas de utilização, menos servidores e menos consumo de energia.

Por outro lado, as desvantagens da utilização de computação em nuvem estão associadas a segurança da informação como a cibercrime e a espionagem (KSHETRI, 2010).

Especialmente em cidades, a computação em nuvem aliada a dados de sensores tem potenciais aplicações que, segundo alguns autores, podem criar uma quantidade sem precedentes de oportunidades para construir soluções de valor ao processo de tomada decisão e torná-lo mais eficiente e eficaz (PERERA, ZASLAVSKY, *et al.*, 2014).

F. Tabelas do banco de dados

POTHOLE. Esta tabela pertencente ao caso de uso de detecção de buraco em via pública e armazena os dados brutos dos sensores que analisam a aceleração dentro de um veículo. Os registros são inseridos em lote após o *download* dos dados pelos sensores. A granularidade dos dados é 1 segundo.

Tabela 5: Tabela do banco de dados - *Pothole*

Campo	Tipo	Descrição
<i>ID</i>	<i>BIGINT</i>	Identificação do ponto incrementada automaticamente em uma unidade quando é adicionado os registros
<i>DATE</i>	<i>INTEGER</i>	Dia do ano fornecido pelo GPS. Formato: DD/MM/AA
<i>TIME</i>	<i>INTEGER</i>	Horário fornecido pelo GPS. Formato: LINUX <i>epoch</i> . Ex.: 1427064122
<i>MEASURES</i>	<i>INTEGER</i>	Quantidade de amostragem por segundo dos eixos (X, Y e Z)
<i>AXIS_X</i>	<i>LONG VARCHAR</i>	Medições de acelerações no eixo X.
<i>AXIS_Y</i>	<i>LONG VARCHAR</i>	Medições de acelerações no eixo Y.
<i>AXIS_Z</i>	<i>LONG VARCHAR</i>	Medições de acelerações no eixo Z.
<i>LATITUDE</i>	<i>DOUBLE</i>	Georreferência recebida pelo GPS. Formato: +/- XXX.XXXXXX. Ex.: -23.040012
<i>LONGITUDE</i>	<i>DOUBLE</i>	Georreferência recebida pelo GPS. Formato: +/- XXX.XXXXXX. Ex.: -46.680504
<i>SPEED</i>	<i>DOUBLE</i>	Velocidade calculada pelo GPS.
<i>SATELITES</i>	<i>INTEGER</i>	Quantidade de satélites presentes no momento da leitura do GPS
<i>PATH_ID</i>	<i>INTEGER</i>	Identificação do trajeto que foi realizado as medições.

Fonte: Autor

CATEGORIES POTHOLE. Esta tabela pertencente ao caso de uso de detecção de buraco em via pública e define as categorias existentes para a tabela *MARKED POTHOLE*. Os registros são inseridos manualmente pelo gestor do sistema.

Tabela 6: Tabela do banco de dados - *Categories Pothole*

Campo	Tipo	Descrição
<i>CATEGORY_ID</i>	<i>INTEGER</i>	Identificação da categoria existente.
<i>DESCRIPTION</i>	<i>VARCHAR (255)</i>	Descrição da categoria. Ex.: Buraco em via pública.

Fonte: Autor

MARKED POTHOLE. Esta tabela pertencente ao caso de uso de detecção de buraco em via pública e armazena a classificação dos pontos armazenados na tabela *POTHOLE*. Os registros são inseridos pelo algoritmo de classificação de dados após a análise do ponto, identificado na tabela com

ID. Os valores dos eixos (AXIS_X, AXIS_Y e AXIS_Z) são armazenados já com a amostragem normalizada de 120 amostras por segundo.

Tabela 7: Tabela do banco de dados - *Marked Pothole*

Campo	Tipo	Descrição
<i>ID_MARK</i>	<i>BIGINT</i>	Identificação incrementada automaticamente em uma unidade quando é adicionado os registros
<i>ID</i>	<i>BIGINT</i>	Identificação do ponto coletado na tabela POTHOLE
<i>CATEGORY_ID</i>	<i>INTEGER</i>	Identificação da categoria classificada
<i>PATH_ID</i>	<i>INTEGER</i>	Identificação do trajeto que foi realizado as medições.
<i>AXIS_X</i>	<i>LONG VARCHAR</i>	Medições de aceleração do eixo X. Valores normalizados em 120 registros.
<i>AXIS_Y</i>	<i>LONG VARCHAR</i>	Medições de aceleração do eixo Y. Valores normalizados em 120 registros.
<i>AXIS_Z</i>	<i>LONG VARCHAR</i>	Medições de aceleração do eixo Z. Valores normalizados em 120 registros.
<i>SPEED</i>	<i>DOUBLE</i>	Velocidade calculada pelo GPS.

Fonte: Autor

APPLICATIONS. Esta tabela define as aplicações existentes no aplicativo de *crowdsensing* do dispositivo móvel. As aplicações do dispositivo móvel do participante podem assumir enfoques distintos entre elas. Por exemplo, a aplicação “Cidades Conscientes” considera os relatos de problemas nos serviços das cidades, porém pode coexistir no mesmo aplicativo móvel do participante uma outra aplicação, como a coleta da opinião do cidadão sobre a qualidade de um evento cultural ocorrendo na cidade. Os registros nesta tabela são inseridos manualmente pelo gestor do sistema.

Tabela 8: Tabela do banco de dados - *Applications*

Campo	Tipo	Descrição
<i>ID</i>	<i>INTEGER</i>	Identificação incrementada automaticamente em uma unidade quando é adicionado os registros.
<i>NAME</i>	<i>VARCHAR (255)</i>	Nome da aplicação. Ex.: Cidades Conscientes
<i>DESCRIPTION</i>	<i>VARCHAR (255)</i>	Texto com um breve descritivo da finalidade da aplicação.
<i>VERSION</i>	<i>INTEGER</i>	Número da versão da aplicação.

Fonte: Autor

SEARCH AREA. Esta tabela pertencente ao caso de uso de detecção de buraco em via pública e define a área de procura por buracos já classificados, considerando a localização do relatório do participante. O registro é atualizado automaticamente pelo gestor do sistema, na interface de visualização. O campo denominado área é compreendido como a lateral (L) da área de procura (Área = L²). Este valor considera como início o ponto onde está a localização do relatório do participante.

Tabela 9: Tabela do banco de dados - Search Area

Campo	Tipo	Descrição
AREA	INTEGER	Valor da lateral (L) da área de procura partindo do ponto analisado.

Fonte: Autor

CITIZEN REPORTS. Esta tabela recebe o conteúdo dos relatórios enviado pelos participantes através do aplicativo de *crowdsensing* do dispositivo móvel, exceto os dados binários da fotografia. Os registros são inseridos automaticamente a cada novo relatório recebido.

Tabela 10: Tabela do banco de dados - Citizen Reports

Campo	Tipo	Descrição
ID_REPORT	INTEGER	Identificação incrementada automaticamente em uma unidade quando é adicionado os registros.
USER_ID	VARCHAR (255)	Identificação do participante. Normalmente o UUID do dispositivo.
TIMESTAMP	INTEGER	Horário fornecido pelo dispositivo móvel. Formato: LINUX epoch. Ex.: 1427064122
CATEGORY	INTEGER	Identificação da categoria utilizada no relatório.
TYPE	INTEGER	Tipo de atributo usado.
CRITICALITY	INTEGER	Criticidade sugerida para solução do problema.
LATITUDE	DOUBLE	Georreferência recebida pelo GPS do dispositivo móvel. Formato: +/-XXX.XXXXXX. Ex.: -23.040012
LONGITUDE	DOUBLE	Georreferência recebida pelo GPS do dispositivo móvel. Formato: +/-XXX.XXXXXX. Ex.: -46.680504
PHOTO_URI	VARCHAR (255)	Nome do arquivo da foto enviada. Ex.: 1427064122.jpg
COMMENTS	VARCHAR (255)	Comentários adicionais em texto livre do participante.

Fonte: Autor

Tabela 11: Tabela do banco de dados - Attributes

Campo	Tipo	Descrição
ID	INTEGER	Identificação incrementada automaticamente em uma unidade quando é adicionado os registros.
NAME	VARCHAR (255)	Nome do atributo. Ex.: Tamanho
DESCRIPTION	VARCHAR (255)	Texto com um breve descritivo da finalidade do atributo.
TYPE	VARCHAR (255)	Tipo do atributo.
POSSIBLEVALUES	VARCHAR (255)	Valores possíveis que podem ser incorporados pelo atributo.

Fonte: Autor

ATTRIBUTES. Esta tabela define os atributos existentes no aplicativo de *crowdsensing* do dispositivo móvel. Os registros são inseridos manualmente pelo gestor do sistema. Uma característica importante desta tabela é a definição do tipo de atributo e os valores possíveis. Estes dados serão

utilizados pela aplicação *crowdsensing* para customização das interações com o participante. Por exemplo, o atributo “TAMANHO” ele pode ser do tipo “RADIO_BUTTON” e ter incorporado como valores possíveis os textos: “GRANDE, MEDIO e PEQUENO”.

CATEGORIES. Esta tabela define as categorias existentes no aplicativo de *crowdsensing* do dispositivo móvel. Os registros são inseridos manualmente pelo gestor do sistema. As categorias são abstrações das funcionalidades que o gestor pretende coletar de informação, por exemplo, “BURACO EM RUA”.

Tabela 12: Tabela do banco de dados - *Categories*

Campo	Tipo	Descrição
ID	INTEGER	Identificação incrementada automaticamente em uma unidade quando é adicionado os registros.
NAME	VARCHAR (255)	Nome da categoria. Ex.: Buraco em Rua.
DESCRIPTION	VARCHAR (255)	Texto com um breve descritivo da finalidade da categoria.

Fonte: Autor

PARTICIPANTS. Esta tabela recebe dados dos participantes que voluntariamente se candidataram a modificar o seu apelido no sistema. Na tabela 13, estão incluídas as informações do e-mail e telefone do participante, que eventualmente podem ser utilizadas pelo gestor para entrar em contato com ele por outro canal de comunicação, ao invés da aplicação móvel.

Tabela 13: Tabela do banco de dados - *Participants*

Campo	Tipo	Descrição
ID	INTEGER	Identificação incrementada automaticamente em uma unidade quando é adicionado os registros.
DEVICE_ID	VARCHAR (255)	Identificação do dispositivo participante. Normalmente o UUID do dispositivo.
ALIAS	VARCHAR (255)	Apelido que o participante quer que esteja disponível na aplicação.
PHONE	VARCHAR (255)	Número do telefone do participante voluntariamente cedido.
EMAIL	VARCHAR (255)	E-mail do participante voluntariamente cedido.
PASSWORD	VARCHAR (255)	Senha do participante para modificação destes dados nesta tabela

Fonte: Autor

G. Definições das interfaces

As interfaces REST descritas pertencem aos serviços **coordenador**, **notificações**, **relatórios** e **gerenciador de banco de dados**. Elas enviam e recebem conteúdo do tipo JSON, deste modo o cabeçalho do método POST possui as variáveis:

Content-Type: application/json

Accept: application/json

Serviço: Coordenador

Analisis. Esta interface chama o método que é responsável pela análise dos pontos em uma rota selecionada. Cada ponto é classificado por modelos analíticos existentes conforme a aplicação corrente. Esta interface opera em conjunto com o serviço do gerenciador do banco de dados para armazenar além dos pontos em uma rota selecionada, também a categoria analisada.

Método: POST

URL: /coordinator/analysis/<pothole>/<device_id>/<path_id>

<pothole>: Nome da aplicação

<device_id>: Identificação do sensor.

<path_id>: Identificação do trajeto que foi realizado as medições.

A interface retorna sucesso ou falha sinalizando o processamento da tarefa para a aplicação solicitante.

Alerts: Esta interface chama o método responsável pela análise do relatório enviado pelo participante. Verifica-se nas proximidades do relatório recebido, pontos de atenção adicionais que podem ser auditados pelo participante, coletado por sensores próximos.

Método: GET

URL: /coordinator/alerts/<application_id>/<id_categories>/<id_report>

<Application_id>: Identificador da aplicação. Ex.: 1 - Cidades conscientes

<Id_categories>: Identificador da categoria. Ex.: 1 – Buraco em via pública

<id_report>: Identificação do registro dos relatórios *crowdsensing*

A interface retorna sucesso ou falha sinalizando o processamento da tarefa para a aplicação solicitante.

Serviço: Notificações

Set Search Area. Esta interface chama o método que define a métrica da regra de procura usada no *middleware*.

Método: POST

URL: /notification/set_search_area/<pothole>/

<pothole>: Nome da aplicação

Tabela 14: Conteúdo da interface Set Search Area - Notificações

Campo	Descrição
<i>AREA</i>	Valor da lateral (L) da área de procura partindo do ponto analisado. Área de procura = L^2 .

Fonte: Autor

Warn. Esta interface chama o método que dispara mensagens aos serviços de notificações dos provedores externos. Todos os provedores (Google, Apple) recebem a mesma mensagem simultaneamente, porém apenas o provedor que detém a conta do participante conclui o processo de alerta. Um exemplo da mensagem de texto que o provedor recebe para publicação ao participante é mostrada na figura 25.

Método: POST

URL: /notification/warn/<pothole>/<user_id>

<pothole>: Nome da aplicação

Tabela 15: Conteúdo da interface Warn - Notificações

Campo	Descrição
<i>DISTANCE</i>	Distância entre o último relatório do participante e o ponto coletado por sensor mais próximo.

Fonte: Autor

Serviço: Relatórios

Path. Esta interface chama o método que lista os pontos coletados pelos sensores.

Método: GET:

URL: /reports/path/<pothole>/<device_id>/<path_id>

<pothole>: Nome da aplicação

<device_id>: Identificação do sensor.

<path_id>: Identificação da rota.

Tabela 16: Resultado da interface *Path* - Relatórios

Campo	Descrição
<i>ID</i>	Identificação do ponto coletado.
<i>LATITUDE</i>	Georreferência recebida pelo GPS.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS.

Fonte: Autor

***Classified Path*:** Esta interface chama o método que lista os pontos já classificados, de acordo com uma classificação selecionada.

Método: GET

URL: /reports/classified_path/<pothole>/<category_id>

<pothole>: Nome da aplicação

<category_id>: Identificação da categoria classificada. Ex.: Via Ondulada

Tabela 17: Resultado da interface *Classified Path* - Relatórios

Campo	Descrição
<i>ID</i>	Identificação do ponto coletado.
<i>LATITUDE</i>	Georreferência recebida pelo GPS.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS.

Fonte: Autor

***Crowdsensing*:** Esta interface chama o método que lista todos os relatórios recebidos dos participantes.

Método: GET

URL: /reports/crowdsensing/<application_id>/<category>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<category>: Identificador da categoria. Ex.: 1 - Ex.: Buraco em Rua.

Tabela 18: Resultado da interface *Crowdsensing* - Relatórios

Campo	Descrição
ID_REPORT	Identificação do registro do relatório.
TYPE	Tipo de atributo usado.
CRITICALITY	Criticidade sugerida para solução do problema.
LATITUDE	Georreferência recebida pelo GPS do dispositivo móvel.
LONGITUDE	Georreferência recebida pelo GPS do dispositivo móvel.
PHOTO_URI	Nome do arquivo da foto enviada. Ex.: 1427064122.jpg
COMMENTS	Comentários adicionais em texto livre do participante.

Fonte: Autor

Image: Esta interface chama o método executa o download de uma fotografia reportada pelo participante.

Método: GET

URL: /reports/image/<application_id>/<photo_uri>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<photo_uri>: Nome do arquivo da foto procurada. Ex.: 1427064122.jpg

O resultado desta interface será a própria figura solicitada.

Serviço: Gerenciador de banco de dados

Post Sensors. Esta interface chama o método que grava os dados recebidos dos sensores no banco de dados.

Método: POST:

URL: /database/sensors/<pothole>/<device_id>/<timestamp>

<pothole>: Nome da aplicação

<device_id>: Identificação do sensor.

<timestamp>: Horário fornecido pelo GPS.

Tabela 19: Conteúdo da interface *Post Sensors* - Gerenciador de dados

Campo	Descrição
<i>MEASURES</i>	Quantidade de amostras por segundo dos eixos (X, Y e Z)
<i>AXIS_X</i>	Medições de acelerações no eixo X.
<i>AXIS_Y</i>	Medições de acelerações no eixo Y.
<i>AXIS_Z</i>	Medições de acelerações no eixo Z.
<i>SPEED</i>	Velocidade calculada pelo GPS.
<i>PATH_ID</i>	Número do trajeto que foi realizado as medições.

Fonte: Autor

Get Sensors. Esta interface chama o método retorna os dados sobre o conjunto dos pontos coletados pelos sensores em uma rota. A interface opera em conjunto com o serviço coordenador para a classificação dos pontos, de acordo com os modelos analíticos apropriados pela aplicação.

Método: POST:

URL: /database/sensors/<pothole>/<device_id>/<path_id>

<pothole>: Nome da aplicação

<device_id>: Identificação do sensor.

<path_id>: Identificação do trajeto que foi realizado as medições.

Tabela 20: Conteúdo da interface *Get Sensors* - Gerenciador de banco de dados

Campo	Descrição
<i>ID</i>	Identificação do ponto que foi realizado as medições
<i>MEASURES</i>	Quantidade de amostras por segundo dos eixos (X, Y e Z)
<i>AXIS_X</i>	Medições de acelerações no eixo X.
<i>AXIS_Y</i>	Medições de acelerações no eixo Y.
<i>AXIS_Z</i>	Medições de acelerações no eixo Z.
<i>SPEED</i>	Velocidade calculada pelo GPS.
<i>PATH_ID</i>	Número do trajeto que foi realizado as medições.

Fonte: Autor

Track Sensors. Esta interface chama o método que lista os pontos coletados pelos sensores. É usada em conjunto com o serviço de relatórios para auxiliar na visualização dos dados.

Método: GET:

URL: /database/track_sensors/<pothole>/<device_id>/<path_id>

<pothole>: Nome da aplicação

<device_id>: Identificação do sensor.

<path_id>: Identificação da rota.

Tabela 21: Resultado da interface *Track Sensors* - Gerenciador de dados

Campo	Descrição
<i>ID</i>	Identificação do ponto coletado.
<i>LATITUDE</i>	Georreferência recebida pelo GPS.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS.

Fonte: Autor

Report: Esta interface chama o método que grava os dados recebidos pelos relatórios dos participantes no banco de dados.

Método: POST

URL: /database/report/<application_id>/<user_id>/<timestamp>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<user_id>: Identificação do participante. Normalmente o UUID do dispositivo.

Tabela 22: Conteúdo da interface *Post Report* - Gerenciador de dados

Campo	Descrição
<i>TIMESTAMP</i>	Horário fornecido pelo dispositivo do participante.
<i>TYPE</i>	Tipo de atributo usado.
<i>CRITICALITY</i>	Criticidade sugerida para solução do problema.
<i>LATITUDE</i>	Georreferência recebida pelo GPS do dispositivo móvel.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS do dispositivo móvel.
<i>PHOTO_URI</i>	Nome do arquivo da foto enviada. Ex.: 1427064122.jpg
<i>COMMENTS</i>	Comentários adicionais em texto livre do participante.

Fonte: Autor

Report: Esta interface chama o método que retorna os dados sobre um relatório de participante do banco de dados. É usada em conjunto com o serviço coordenador para localizar pontos próximos coletado por sensores que podem ser auditados pelo participante.

Método: GET

URL: /database/report/<application_id>/<Id_categories >/<id_report>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<Id_categories>: Identificador da categoria. Ex.: 1 – Buraco em via pública

<id_report>: Identificação do registro dos relatórios *crowdsensing*.

Tabela 23: Resultado da interface *Get Report* - Gerenciador de banco de dados

Campo	Descrição
<i>LATITUDE</i>	Georreferência recebida pelo GPS do dispositivo móvel.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS do dispositivo móvel.

Fonte: Autor

Citizen Reports: Esta interface chama o método que lista todos os relatórios recebidos dos participantes. É usada em conjunto com o serviço de relatórios para auxiliar na visualização dos dados.

Método: GET

URL: /database/citizen_reports/<application_id>/<category_id>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<category_id>: Identificador da categoria. Ex.: 1 - Ex.: Buraco em Rua.

Tabela 24: Resultado da interface *Citizen Reports* - Gerenciador de dados

Campo	Descrição
ID_REPORT	Identificação do registro do relatório.
<i>TYPE</i>	Tipo de atributo usado.
<i>CRITICALITY</i>	Criticidade sugerida para solução do problema.
<i>LATITUDE</i>	Georreferência recebida pelo GPS do dispositivo móvel.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS do dispositivo móvel.
<i>PHOTO_URI</i>	Nome do arquivo da foto enviada. Ex.: 1427064122.jpg
<i>COMMENTS</i>	Comentários adicionais em texto livre do participante.

Fonte: Autor

Categories: Esta interface chama o método que é responsável para listar as categorias de relatórios de *crowdsensing* possíveis.

Método: GET

URL: /database/crowdsensing/categories

Tabela 25: Resultado da interface *Categories* - Gerenciador de banco de dados

Campo	Descrição
ID	Identificação da categoria.
NAME	Nome da categoria. Ex.: Buraco em Rua.
DESCRIPTION	Texto com um breve descritivo da finalidade da categoria.

Fonte: Autor

Participants: Esta interface chama o método que grava as informações sobre os participantes que voluntariamente se candidatam a modificar o seu apelido no sistema.

Método: POST

URL: /database/crowdsensing/participants/<device_id>

<device_id>: Identificação do dispositivo participante. Normalmente o UUID do dispositivo.

Tabela 26: Conteúdo da interface *Participants* - Gerenciador de banco de dados

Campo	Descrição
ALIAS	Apelido que o participante quer disponível na aplicação.
PHONE	Número do telefone do participante voluntariamente cedido.
EMAIL	E-mail do participante voluntariamente cedido.
PASSWORD	Senha do participante para modificação destes dados.

Fonte: Autor

Applications. Esta interface chama o método que define as aplicações existentes no aplicativo de *crowdsensing* do dispositivo móvel.

Método: GET

URL: /database/crowdsensing/applications

Tabela 27: Resultado da interface *Applications* - Gerenciador de banco de dados

Campo	Descrição
ID	Identificação do registro da aplicação.
NAME	Nome da aplicação. Ex.: Cidades Conscientes
DESCRIPTION	Texto com um breve descritivo da finalidade da aplicação.
VERSION	Número da versão da aplicação.

Fonte: Autor

Attributes. Esta interface chama o método que define os atributos existentes no aplicativo de *crowdsensing* do dispositivo móvel.

Método: GET

URL: /database/crowdsensing/attributes

Tabela 28: Resultado da interface *Attributes* - Gerenciador de banco de dados

Campo	Descrição
<i>ID</i>	Identificação do registro do atributo.
<i>NAME</i>	Nome do atributo. Ex.: Tamanho
<i>DESCRIPTION</i>	Texto descritivo da finalidade do atributo.
<i>TYPE</i>	Tipo do atributo.
<i>POSSIBLEVALUES</i>	Valores possíveis incorporados pelo atributo.

Fonte: Autor

Classified Points: Esta interface chama o método que lista os pontos já classificados de acordo uma classificação selecionada. É usada em conjunto com o serviço de relatórios para auxiliar na visualização dos dados.

Método: GET

URL: /database/classified_points/<pothole>/<category_id>

<pothole>: Nome da aplicação

<category_id>: Identificação da categoria classificada. Ex.: Via Ondulada

Tabela 29: Resultado da interface *Classified Points* - Gerenciador de dados

Campo	Descrição
<i>ID</i>	Identificação do ponto coletado.
<i>LATITUDE</i>	Georreferência recebida pelo GPS.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS.

Fonte: Autor

Classified Points: Esta interface chama o método que armazena os pontos já classificados pelo serviço coordenador.

Método: POST

URL: /database/classified_points/<pothole>/<device_id>/<Id>

<pothole>: Nome da aplicação

<device_id>: Identificação do sensor.

<id>: Identificação do ponto que foi realizado as medições.

Tabela 30: Conteúdo da interface *Classified Points* – Gerenciador de dados

Campo	Descrição
<i>CATEGORY_ID</i>	Identificação da categoria classificada
<i>AXIS_X</i>	Medições de acelerações no eixo X.
<i>AXIS_Y</i>	Medições de acelerações no eixo Y.
<i>AXIS_Z</i>	Medições de acelerações no eixo Z.
<i>SPEED</i>	Velocidade calculada pelo GPS.
<i>PATH_ID</i>	Identificação do trajeto que foi realizado as medições.

Fonte: Autor

Reported Image: Esta interface chama o método executa o *download* de uma fotografia reportada pelo participante. É usada em conjunto com o serviço de relatórios para auxiliar na visualização dos relatórios dos participantes.

Método: GET

URL: /database/reported_image/<application_id>/<photo_uri>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<photo_uri>: Nome do arquivo da foto procurada. Ex.: 1427064122.jpg

O conteúdo recebido desta interface será a própria figura solicitada.

Search Points: Esta interface chama o método que procura, a partir de uma posição provida por um relatório do participante e uma área de procura definida, qual o ponto coletado por sensores conforme uma categoria de problema. É usada em conjunto com o serviço coordenador para verificar se existe uma sugestão de ponto próximo ao participante que pode ser auditado.

Método: GET

URL:/database/search_points/<pothole>/<category_id>/<lat_min>/<long_min>/<lat_max>/<long_max>

<pothole>: Nome da aplicação

<category_id>: Identificação da categoria classificada. Ex.: Via Ondulada

<lat_min>: Georreferência de latitude mínima.

<long_min>: Georreferência de longitude mínima.

<lat_max>: Georreferência de latitude máxima.

<long_max>: Georreferência de longitude máxima.

Tabela 31: Resultado da interface *Search Points* - Gerenciador de dados

Campo	Descrição
<i>ID</i>	Identificação do ponto coletado.
<i>LATITUDE</i>	Georreferência recebida pelo GPS.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS.

Fonte: Autor

Categories Pothole. Esta interface pertencente ao caso de uso de detecção de buraco em via pública e chama o método que define as categorias existentes para a tabela *MARKED POTHOLE*.

Método: GET

URL: /database/categories_pothole/<pothole>

<pothole>: Nome da aplicação

Tabela 32: Resultado da interface *Categories Pothole* - Gerenciador de dados

Campo	Descrição
<i>CATEGORY_ID</i>	Identificação da categoria existente.
<i>DESCRIPTION</i>	Descrição da categoria. Ex.: Buraco em via pública.

Fonte: Autor

Search area: Esta interface chama o método que retorna o valor da área de procura estipulado pelo gestor como regra. É usada em conjunto com o serviço coordenador como a métrica de proximidade na procura por pontos coletados por sensores próximo ao participante.

Método: GET

URL: /database/search_area/<pothole>/

<pothole>: Nome da aplicação

Tabela 33: Resultado da interface *get Search Area* - Gerenciador de dados

Campo	Descrição
<i>AREA</i>	Valor da lateral (L) da área de procura partindo do ponto analisado. Área de procura = L^2 .

Fonte: Autor

Search area: Esta interface chama o método que grava a área de procura estipulada pelo gestor. É usada em conjunto com o serviço de notificações como a regra de procura automatizada por pontos coletados por sensores próximo ao participante.

Método: POST

URL: /database/search_area/<pothole>/

<pothole>: Nome da aplicação

Tabela 34: Conteúdo da interface *post Search Area* - Gerenciador de dados

Campo	Descrição
<i>AREA</i>	Valor da lateral (L) da área de procura partindo do ponto analisado. Área de procura = L^2 .

Fonte: Autor

Images: Esta interface chama o método executa o *upload* de uma fotografia reportada pelo participante.

Método: POSTURL: /database/images/<application_id>/<photo_uri>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<photo_uri>: Nome do arquivo da foto procurada. Ex.: 1427064122.jpg

O conteúdo recebido desta interface será a própria figura encaminhada.

Serviço: Gerenciador de eventos

O serviço gerenciador de eventos é constituído pelas interfaces de tópicos publicar e assinar. Elas enviam e recebem conteúdo do tipo JSON os quais estão detalhados nas tabelas nos tópicos respectivos.

Application. Este tópico é utilizado pela aplicação móvel para receber os parâmetros que serão utilizados pela aplicação *crowdsensing*.

Método: Assinar

Tópico: /event/application

Tabela 35: Conteúdo do tópico *Application* - Gerenciador de eventos

Campo	Descrição
<i>ID</i>	Identificação da aplicação.
<i>NAME</i>	Nome da aplicação. Ex.: Cidades Conscientes
<i>DESCRIPTION</i>	Texto com um breve descritivo da finalidade da aplicação.
<i>VERSION</i>	Número da versão da aplicação.

Fonte: Autor

Sensors. Este tópico é utilizado pelo sorvedouro para inserir os dados dos sensores no *middleware*. Os registros são enviados em lotes com a granularidade de 1 segundo.

Método: Publicar

Tópico: /event/sensors/<pothole>/<device_id>

<pothole>: Nome da aplicação

<device_id>: Identificação do sensor.

Tabela 36: Conteúdo do tópico *Sensors* - Gerenciador de eventos

Campo	Descrição
<i>MEASURES</i>	Quantidade de amostras por segundo dos eixos (X, Y e Z)
<i>DATE</i>	Dia do ano fornecido pelo GPS.
<i>TIME</i>	Horário fornecido pelo GPS.
<i>LATITUDE</i>	Georreferência recebida pelo GPS.
<i>LONGITUDE</i>	Georreferência recebida pelo GPS.
<i>AXIS_X</i>	Medições de acelerações no eixo X.
<i>AXIS_Y</i>	Medições de acelerações no eixo Y.
<i>AXIS_Z</i>	Medições de acelerações no eixo Z.
<i>SPEED</i>	Velocidade calculada pelo GPS.
<i>SATELITES</i>	Quantidade de satélites no momento da leitura do GPS.
<i>PATH_ID</i>	Número do trajeto que foi realizado as medições.

Fonte: Autor

Report. Este tópico é utilizado pela aplicação móvel para inserir os dados do relatório coletado pelo participante ao *middleware*.

Método: Publicar

Tópico: /event/report/<application_id>/<device_id>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

<user_id>: Identificação do participante. Normalmente o UUID do dispositivo.

Tabela 37: Conteúdo do tópico *Report* - Gerenciador de eventos

Campo	Descrição
TIMESTAMP	Horário fornecido pelo dispositivo do participante.
TYPE	Tipo de atributo usado.
CRITICALITY	Criticidade sugerida para solução do problema.
LATITUDE	Georreferência recebida pelo GPS do dispositivo móvel.
LONGITUDE	Georreferência recebida pelo GPS do dispositivo móvel.
PHOTO_URI	Nome do arquivo da foto enviada. Ex.: 1427064122.jpg
COMMENTS	Comentários adicionais em texto livre do participante.

Fonte: Autor

Categories_Attributes. Este tópico é utilizado pela aplicação móvel para receber as definições de categorias e atributos que serão utilizados pela aplicação *crowdsensing*.

Método: Assinar

Tópico: /event/application/<application_id>

<application_id>: Identificador da aplicação. Ex.: 1 - Cidades Conscientes

Tabela 38: Conteúdo do tópico *Categories_Attributes* - Gerenciador de eventos

Campo	Descrição
CATEGORY_ID	Identificação da categoria.
NAME	Nome da categoria. Ex.: Buraco em Rua.
DESCRIPTION	Texto com descritivo da finalidade da categoria.
ATTRIBUTE_ID	Identificação do atributo.
ATTRIBUTE_NAME	Nome do atributo. Ex.: Tamanho
ATTRIBUTE_DESCRIPTION	Texto descritivo da finalidade do atributo.
TYPE	Tipo do atributo.
POSSIBLEVALUES	Valores possíveis incorporados pelo atributo.

Fonte: Autor

H. Modelos de algoritmos de classificação utilizados

O aprendizado de máquina é uma área de pesquisa de computação que estuda o desenvolvimento de métodos capazes de extrair conhecimento a partir de amostras de dados. Diversos algoritmos existentes de aprendizado de máquina são utilizados para encontrar e atribuir um determinado rótulo de classe, a uma porção específica de um conjunto de dados. Este processo é denominado de método de classificação (MITCHELL, 1997). A seguir se encontram breves descrições dos métodos de classificação utilizados nesta pesquisa.

K - Vizinhos mais próximos. O método K-NN (*K - Nearest Neighbour*) baseia-se na proximidade entre os dados para efetuar a predição. O objetivo é classificar um ponto desconhecido com base nos exemplos do conjunto de treinamento que estão próximos a ele. A hipótese deste método é que dados similares tendem a estar concentrados em uma mesma região no espaço de entrada (FACELLI, LORENA, *et al.*, 2011).

O algoritmo de K-NN opera em situações onde o conjunto de dados de entrada, ou instância de dados, pode ser definido por um vetor de k dimensões. A comparação do ponto a ser classificado é realizada em todos os pontos do conjunto de treinamento, normalmente pela distância Euclidiana, de modo a encontrar a menor distância (COPPIN, 2010). Em um exemplo de espaço tridimensional, a distância entre o ponto que se deseja classificar (P1) e apenas um ponto do conjunto de treinamento (P2) é:

$$\sqrt{((Px1 - Px2)^2 + (Py1 - Py2)^2 + (Pz1 - Pz2)^2)}$$

Entre os aspectos positivos está a facilidade para o treinamento, uma vez que basta apenas armazenar o conjunto de treinamento em memória. O algoritmo é aplicável a problemas complexos e também é aprimorado de modo incremental, ou seja, quando existe um incremento de dados de treinamento basta incluí-lo ao conjunto anterior, melhorando assim a precisão do método.

Quanto aos aspectos negativos, considera-se que quanto maior o conjunto de treinamento, maior será o processamento para realizá-lo, pois o teste requer calcular a distância entre o ponto a classificar e os demais pontos do conjunto de treinamento. Além disso, uma alta dimensionalidade normalmente compromete a precisão (FACELLI, LORENA, *et al.*, 2011).

Classificador Ingênuo de Bayes. Este classificador, mais popularmente conhecido como *Naive Bayes*, baseia-se em métodos probabilísticos bayesianos. Estes métodos assumem que a probabilidade de um evento B , dado um evento A , não depende apenas da relação entre B e A , mas também da probabilidade de observar B independentemente de observar A (FACELLI, LORENA, *et al.*, 2011). Esta relação é expressa pelo teorema de Bayes (COPPIN, 2010):

$$P(B|A) = \frac{(P(A|B) * P(B))}{P(A)}$$

Onde:

$P(B)$ é a probabilidade a priori de B .

$P(A)$ é a probabilidade a priori de A .

$P(B|A)$, é a probabilidade de B , dado A . Esta notação além de ser chamada de probabilidade condicionada é também conhecida como probabilidade a posteriori de B (COPPIN, 2010).

O algoritmo *Naive Bayes* calcula a probabilidade de uma proposição verdadeira, dado que já se conhece um fragmento relacionado de informação. Na fase de aprendizado, considera que uma hipótese é verdadeira a partir de um conjunto específico de evidências, assim o classificador pode determinar em qual hipótese acreditar ao identificar a maior probabilidade a posteriori de cada uma delas (COPPIN, 2010).

O classificador ingênuo de Bayes assume que os atributos de uma classe são independentes entre si, por isto recebe o nome de ingênuo (FACELLI, LORENA, *et al.*, 2011). Entre os aspectos positivos está a facilidade de implementá-lo de forma incremental, também ter um bom desempenho em uma grande variedade de domínios, inclusive quando existe uma clara dependência entre os atributos, além de ser robusto a atributos irrelevantes (FACELLI, LORENA, *et al.*, 2011). Quanto aos aspectos negativos está a dificuldade em lidar com a dependência de atributos.

Redes neurais artificiais. Este método é inspirado na capacidade de aprendizado das redes neurais biológicas presentes no cérebro (FACELLI, LORENA, *et al.*, 2011). Baseia-se em unidades de processamentos, denominadas de neurônios, densamente interconectadas, as quais executam funções matemáticas. Estas conexões entre neurônios simulam as sinapses do cérebro e possuem pesos associados. O comportamento de um neurônio em uma rede neural artificial (RNA) é expresso por (COPPIN, 2010):

$$Y = \sum_{i=1}^n w_i x_i$$

Onde:

' Y ' é a soma ponderada das entradas do neurônio

' n ' é o número de entradas

' x ' é o valor de cada entrada

' w ' é o peso correspondente da entrada

Durante o processo de aprendizado de uma RNA os pesos são automaticamente ajustados. Por exemplo, a rede *perceptron*, desenvolvida por ROSENBLATT (1958), é uma das redes neurais artificiais mais populares, cujo método de ajustes dos pesos durante o aprendizado é dado por (ROSENBLATT, 1958):

$$w_i \leftarrow w_i + (a * x_i * e)$$

Onde:

' w ' é o peso correspondente da entrada

' a ' é a taxa de aprendizado

' x ' é o valor de cada entrada

' e ' é o erro que foi produzido

Entre a entrada e a saída deste método de classificação podem existir diversas camadas sequenciais de neurônios interconectadas. A arquitetura, além dos pesos associados nas conexões entre neurônios, é outro aspecto que interfere na capacidade de uma RNA. Este aspecto da complexidade do sistema gerada pelas dependências entre neurônios é a principal crítica as redes neurais artificiais (COPPIN, 2010). Por outro lado, o método é popular pela generalização e tolerância a falhas, destacando em aplicações de visão computacional (FACELLI, LORENA, *et al.*, 2011).

Árvores de decisão. Este classificador utiliza o princípio de dividir um problema inicial complexo em problemas mais simples, o que é aplicado recursivamente na construção do modelo. Formalmente, uma árvore de decisão é um grafo acíclico direcionado, que possui ou um nó de divisão, com dois ou mais sucessores, ou um nó folha (FACELLI, LORENA, *et al.*, 2011). Enquanto o nó de divisão realiza o teste condicional baseado nos valores do atributo, o nó folha encontra-se tipicamente na base da árvore com a função ou variável alvo.

Um dos maiores desafios desta categoria de classificadores é encontrar regras de divisão para a construção da árvore de decisão com menor tamanho possível, condizente com o conjunto de dados (FACELLI, LORENA, *et al.*, 2011). Um método utilizado para determinar quais características usar a cada estágio da árvore é o que oferece maior ganho de informação. Este ganho de informação, também conhecido como redução da entropia dos dados, mede a aleatoriedade das variáveis. A entropia dos dados de treinamento é definida por (COPPIN, 2010):

$$H(S) = -p_1 \log_2 p_1 - p_0 \log_2 p_0$$

Onde:

' p_1 ' é a proporção dos dados de treinamento que inclui exemplos positivos

' p_0 ' é a proporção dos dados de treinamento que inclui exemplos negativos

Entre as vantagens de árvores de decisão estão a facilidade de compreensão do método gerada pela série de decisões simples locais e a flexibilidade em termos de auto adaptação ao conjunto de treinamento (FLACH, 2012). Entretanto, como particionam recursivamente os dados, a árvore construída pode ser muito grande e propensa a super adaptação, que ocorre quando o modelo se torna demasiadamente especialista ao invés de generalizar (COPPIN, 2010). Além disso, algumas outras características negativas são observadas tanto pela instabilidade na criação dos modelos com pequenas variações no conjunto de treinamento, uma vez que as inferências feitas próximo as folhas tendem a ser menos confiáveis quanto aquelas feitas na raiz ou também quando é necessário classificar um valor ausente no conjunto de treinamento. Este tipo de ocorrência pode assumir inconsistências na classificação (FLACH, 2012).

Regras de Decisão. Este classificador implementa condições modulares no conjunto de dados, cujas definições se apresentam na relação entre atributos e valores do domínio. Normalmente estas condições, ou regras, podem ser interpretadas isoladamente uma das outras (FACELLI, LORENA, *et al.*, 2011).

As regras de decisão se assemelham a árvores de decisão, entretanto nos classificadores baseados em árvores de decisão, um nó tem o contexto definido por testes anteriores, o que pode gerar inconvenientes de interpretação. Já as regras de decisão são modulares e cada regra tipicamente cobre uma região específica do espaço de instâncias (FACELLI, LORENA, *et al.*, 2011).

Durante a fase de aprendizado o classificador interativamente pode aprender uma regra geral e em seguida, remover as amostras que já são cobertas por aquela regra, ou vice-versa, partindo das regras específicas até encontrar uma regra mais ampla. Também nesta fase de aprendizado, uma função opera em conjunto avaliando as hipóteses das regras, por exemplo, calculando as taxas de erro geradas pelas regras (FLACH, 2012).

De modo abrangente as regras de decisão são mais fáceis de entender o conjunto de dados do que as árvores de decisão. Assim são recomendadas em tarefas que requerem um conhecimento descritivo da aplicação (FLACH, 2012).

Máquina de vetores de suporte. Os algoritmos de classificação por máquinas de vetores de suporte (*Support Vector Machine* - SVM) baseiam-se na teoria de aprendizado estatístico (VAPNIK, 1995). Esta teoria apresenta modelos matemáticos para a escolha de uma função que separe as

classes em um conjunto de dados de treinamento, de forma a maximizar a generalização do classificador obtido. Para isto, busca-se um compromisso entre a minimização do erro de treinamento e a complexidade da função escolhida para separação das classes de dados (VAPNIK, 1995).

As SVMs buscam por hiperplanos no conjunto de dados que delimitem as regiões das classes. Para estimar uma função usando os exemplos de treinamento e utilizar a teoria de aprendizado estatístico para que a margem de separação entre as classes seja maximizada, o algoritmo procura por um hiperplano ótimo que não ocorra o sobre ou sub ajuste, onde a complexidade da função obtida é superior ou inferior a necessidade do problema.

A equação do hiperplano é dada por (FACELLI, LORENA, *et al.*, 2011):

$$h(x) = w \cdot x + b$$

Onde:

' w ' é o vetor normal ao hiperplano

' b ' é um termo compensador

' x ' é o vetor do conjunto de treinamento

O par (w, b) é determinado durante o treinamento do classificador. A estratégia tem a finalidade de separar o espaço de entradas em regiões, as quais irão delimitar as classes. Considerando um conjunto de dados com duas classes, o resultado seria:

$$w \cdot x + b > 0$$

$$w \cdot x + b < 0$$

Entre as vantagens do uso de SVM estão a boa capacidade de generalização, a robustez em grandes dimensões, por exemplo, categorização de imagens e textos (FACELLI, LORENA, *et al.*, 2011).

I. Questionário de pesquisa de opinião sobre aplicativo de sensoriamento participativo

1) Você já usou algum aplicativo para coletar dados sobre a sua cidade? (Ex.: 1746 Rio)

- ☐ Sim ☐ Não ☐ Nunca ouvi falar

2) O que acha da ideia?

- ☐ Parece útil ☐ Parece inútil ☐ Neutro

3) Por que usaria este tipo de aplicativo?

- ☐ Para denunciar o problema (altruísmo)
☐ Para ver onde existem problemas similares perto de mim (Ex. Na rua onde moro)
☐ Não usaria de jeito nenhum
☐ Apenas se estivesse sendo prejudicado no momento com o problema

4) Se não usaria, qual o principal motivo?

- ☐ Consome muitos dados
☐ Você não é recompensado por isto
☐ Falta de segurança para tirar foto
☐ Falta de privacidade
☐ Parece complicado
☐ Não aplicável

- 5) Em uma situação hipotética que estivesse usando um aplicativo similar para denunciar problemas na sua cidade e recebesse um convite da prefeitura pelo celular dizendo que existe um problema similar ao que está reportando próximo a você. Qual a distância que você poderia caminhar para tentar encontrar o problema informado pela prefeitura?
- ☐ Até 5 metros
 - ☐ Até 20 metros
 - ☐ Até 100 metros
 - ☐ De jeito nenhum ajudaria
- 6) Se fosse recompensado com desconto no seu IPTU você participaria com ao menos 1 relatório por semana?
- ☐ Pelo menos 10% de desconto
 - ☐ Pelo menos 50% de desconto
 - ☐ 100% de desconto
 - ☐ De jeito nenhum ajudaria