Rodrigo Aparecido Morbach

Proposta de uma Arquitetura para Monitoramento e Gerenciamento de Redes Sensoras em Redes Metropolitanas de Acesso Aberto

Campinas
2014

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Rodrigo Aparecido Morbach

Proposta de uma Arquitetura para Monitoramento e Gerenciamento de Redes Sensoras em Redes Metropolitanas de Acesso Aberto

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Orientador: Prof. Dr. Leonardo de Souza Mendes.
Coorientador: Prof. Dr. Gean Davis Breda.

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Rodrigo Aparecido Morbach, e orientada pelo Prof. Dr. Leonardo de Souza Mendes.

Campinas
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

Morbach, Rodrigo Aparecido, 1990-
M798p Proposta de uma arquitetura para monitoramento e gerenciamento de redes sensoras em redes metropolitanas de acesso aberto / Rodrigo Aparecido Morbach. – Campinas, SP : [s.n.], 2014.

Orientador: Leonardo de Souza Mendes.
Coorientador: Gean Davis Breda.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Arquitetura de redes. 2. Redes de comunicação. 3. Redes de sensores. 4. Arquitetura orientada a serviços (Computação). I. Mendes, Leonardo de Souza,1961-. II. Breda, Gean Davis,1968-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Proposal of an architecture for sensor networks monitoring and management in open access metropolitan networks
Palavras-chave em inglês:
Network architecture
Communication networks
Sensor networks
Service-oriented architecture (Computation)
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Leonardo de Souza Mendes [Orientador]
André Marcelo Panhan
Tatiana Annoni Pazeto
Data de defesa: 08-08-2014
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Rodrigo Aparecido Morbach

Data da Defesa: 8 de agosto de 2014

Título da Tese: "Proposta de uma Arquitetura para Monitoramento e Gerenciamento de Redes Sensoras em Redes Metropolitanas de Acesso Aberto"

Prof. Dr. Leonardo de Souza Mendes (Presidente): _____

Dr. André Marcelo Panhan: _____

Profa. Dra. Tatiana Annoni Pazeto: _____

Resumo

As Cidades Digitais, viabilizadas pelo grande desenvolvimento alcançado pelas Tecnologias da Informação e das Comunicações nos últimos 30 anos, surgem hoje como um novo paradigma nestas áreas. As Cidades Digitais promovem grandes benefícios para a população e para o governo, através de ambientes virtuais de oferecimento de serviços, por meio de redes de comunicações digitais. Nesse cenário, as redes de sensores, principalmente as redes sem fio, desempenham papel fundamental para permitir a coleta de dados em tempo real no âmbito municipal, produzindo grande quantidade de informações importantes para fins de análise, previsão e tomada de decisão.

Esse cenário permite o monitoramento de uma grande variedade de dados e informações. Sendo assim, são utilizados diferentes tipos de redes de sensores em termos de hardware e software. Essa heterogeneidade de dispositivos e de tecnologia resulta em ilhas de sistemas, restritos à domínios específicos de aplicação, prejudicando o gerenciamento das redes e a interoperabilidade entre as aplicações.

Diante desse cenário, esse trabalho propõe uma arquitetura para monitoramento e gerenciamento de redes sensoras em Cidades Digitais. A arquitetura proposta é baseada no conceito de orientação a serviços, proporcionando independência de plataforma e interoperabilidade no monitoramento de redes heterogêneas, além de promover o compartilhamento de dados com sistemas externos de modo padronizado. A arquitetura explora o potencial de Redes Metropolitanas de Acesso Aberto para constituir um sistema sensor em escala municipal. Como prova de conceito, um estudo de caso foi realizado na Cidade Digital de Pedreira, concebida sobre a infraestrutura de uma Rede Metropolitana de Acesso Aberto. Os resultados mostram a viabilidade da proposta e expõem as oportunidades oferecidas pela arquitetura.

Palavras-chave: Redes de sensores, Redes Metropolitanas de Acesso Aberto, Arquitetura Orientada a Serviços, Serviços Web.

Abstract

Digital Cities, fostered by the great development reached by Information and Communication Technologies in the last 30 years, arise today as a new paradigm in those areas. A Digital City promotes important benefits for both citizens and government by means of virtual environments for providing services through digital communications networks. In this setting, sensor networks play important role in order to gather real time data in urban scale, providing large and important amount of data for analysis, forecasting and decision making.

There are a wide range of variables to be monitored in a Digital City. Thus, heterogeneous sensor networks related to hardware and software have been employed. Such diversity of devices and technologies yields systems islands constrained to specific application domains. This feature impacts on networks management and interoperability between systems.

Given this scenario, this work presents an architecture for monitoring and managing sensor networks in Digital Cities. The proposed architecture is based on the concept of services, providing plataform independence and interoperability, and also promoting data sharing. The architeture takes the advantages of Open Access Metropolitan Networks to set up a urban scale sensor network. As proof of concept a study case took place in the Digital City of Pedreira. The results show the feasibility of the proposal and the opportunities presented by the architecture.

Key-words: Sensor networks, Open Access Metropolitan Networks, Service-oriented Architecture, Web services.

Sumário

Introdução **1**

1 Revisão de conceitos tecnológicos **5**
1.1 Redes de Sensores Sem Fio (RSSF) . . . 5
1.1.1 Características . . . 5
1.1.2 Sensores e atuadores . . . 7
1.1.3 IEEE 802.15.4/ZigBee . . . 10
1.2 Redes Metropolitanas de Acesso Aberto (RMAA) . . . 14
1.2.1 Definição . . . 15
1.2.2 Exemplos de Redes Metropolitanos de Acesso Aberto . . . 15
1.2.3 Serviços . . . 16
1.2.4 Monitoramento em cenários metropolitanos . . . 18
1.3 Trabalhos relacionados . . . 19
1.4 Considerações sobre o capítulo . . . 23

2 Arquitetura Orientada a Serviços (SOA) **25**
2.1 SOA - Visão geral . . . 25
2.2 *Web Services* . . . 26
2.2.1 *eXtensible Markup Language* (XML) . . . 27
2.2.2 *Web Services Description Language* (WSDL) . . . 28
2.2.3 *Universal Description, Discovery and Integration* (UDDI) . . . 28
2.2.4 *Simple Object Access Protocol* (SOAP) . . . 29
2.2.5 *Representational State Transfer* (REST) . . . 30
2.2.5.1 *Web services* RESTful . . . 31
2.2.6 XML-RPC . . . 32
2.3 Considerações sobre o capítulo . . . 33

3 Arquitetura proposta **35**
3.1 Camadas que compõem a Arquitetura . . . 35
3.1.1 Camada de Aplicação . . . 37
3.1.1.1 Desenvolvimento da Camada de Aplicação . . . 38

3.1.1.2 Portal *Web* . . . 39
3.1.1.3 API *Web service* . . . 41
3.1.2 Camada de Comunicação . . . 43
3.1.2.1 RMAA . . . 43
3.1.2.2 *Gateways* . . . 44
3.1.3 Camada de Sensoriamento . . . 47
3.2 Modos de processamento da informação . . . 47
3.3 Segurança . . . 48
3.3.1 Segurança na Camada de Aplicação . . . 49
3.3.2 Segurança na Camada de Comunicação . . . 49
3.3.3 Segurança na Camada de Sensoriamento . . . 51
3.4 Considerações sobre o capítulo . . . 52

4 Estudo de caso **55**
4.1 Cenário de aplicação e organização da arquitetura . . . 55
4.2 Modelagem do sistema . . . 59
4.2.1 Modelo de domínio . . . 60
4.3 Camada de Sensoriamento . . . 61
4.3.1 Terminais da rede ZigBee . . . 61
4.3.2 Telemetric . . . 64
4.4 Camada de comunicação . . . 65
4.4.1 RMAA - Pontos utilizados no estudo de caso . . . 65
4.4.2 *Gateways* . . . 66
4.4.2.1 *Gateway* da rede ZigBee . . . 66
4.4.2.2 *Gateway* da rede IP . . . 71
4.5 Camada de Aplicação . . . 72
4.5.1 Aplicação servidora . . . 72
4.5.1.1 API *Web service* . . . 73
4.5.2 Portal *Web* . . . 77
4.6 Segurança . . . 82
4.7 Resultados do monitoramento . . . 84
4.7.1 Monitoramento da rede ZigBee . . . 84
4.7.1.1 Monitoramento da comunicação sem fio . . . 86
4.7.1.2 Monitoramento da tensão de energia . . . 88
4.7.2 Monitoramento utilizando o *Telemetric* . . . 89
4.8 Gerenciamento . . . 91
4.9 Considerações sobre o capítulo . . . 93

5 Considerações Finais **95**

Bibliografia **98**

Aos meus pais, Ary e Juraci, e às minhas irmãs, Rosângela e Rosane.

Agradecimentos

Agradeço,

acima de tudo a Deus por ter me abençoado durante toda a minha vida.

aos meus pais, Ary e Juraci, por terem me ajudado de todas as formas possíveis, aos quais nunca poderei retribuir a altura.

ao professor Leonardo Mendes por te me aceito no programa de mestrado e por oferecer oportunidades para o meu crescimento profissional e também pessoal.

ao professor Gean Breda, pela disponibilidade para orientações, sugestões e críticas construtivas para a realização deste trabalho.

aos colegas de LarCom pelo ótimo convívio durante esse tempo de mestrado.

aos colegas da FEEC em geral, brasileiros e estrangeiros, pelo companheirismo, pelo futebol semanal e encontros de finais de semana.

aos professores da FEEC (Faculdade de Engenharia Elétrica e de Computação) e do IC (Instituto de Computação) pelas excelentes disciplinas ministradas.

aos membros da banca pelas críticas e sugestões para a melhoria do trabalho.

a prefeitura municipal de Pedreira, em especial a Ricardo Lazarete e Claudinei Gobbis, por permitir a realização do estudo de caso na cidade.

a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro.

a todos que de alguma forma contribuíram com o meu progresso como aluno e como Ser.

Obrigado!

Lista de Figuras

1.1 Exemplos de redes. 6
1.2 Exemplo de rede de sensores. 7
1.3 Módulos básicos de um nó sensor (Adaptada de Wu & Tseng (2007, Capítulo 2)). 7
1.4 Fluxo de controle de uma WSAN. (Adaptada de Xia et al. (2007)). 9
1.5 Pilha de protocolos ZigBee. 11
1.6 Topologias de rede ZigBee. 12
1.7 Exemplo de interconexão em um município. 16
1.8 Arquitetura do sistema proposto por Vikatos et al. (2011). 20
1.9 Arquitetura híbrida e hierárquica proposta por Pires (2010). 21

2.1 SOA como um arquitetura do tipo cliente-servidor. 26
2.2 Arquitetura *Web service* com UDDI. 28
2.3 Estrutura de um envelope SOAP. 29
2.4 Exemplos de requisição com e sem dependência de estado (adaptada de (Rodriguez 2008)). 32

3.1 Arquitetura proposta. 36
3.2 Elementos e protocolos de comunicação que compõem a arquitetura. 37
3.3 Componentes da camada de aplicação. 40
3.4 Exemplo de arquitetura *Web service*. 42
3.5 Arquitetura física de uma RMAA (adaptada de (Breda et al. 2011)). 44
3.6 Comunicação com *gateways* e exemplos de serviços fornecidos. 45
3.7 Troca de mensagens entre cliente e servidor HTTPS (adaptada de Alonso et al. (2004, capítulo 4)). 50
3.8 Interação do método básico de autenticação HTTP. 51

4.1 Prédios interconectados pela Infovia de Pedreira. 58
4.2 Arquitetura aplicada em Pedreira. 59
4.3 Modelo de domínio. 61
4.4 Arduino Uno e *Shield* XBee utilizados no estudo de caso. 62
4.5 Módulo Xbee *Series* 2 utilizado. 63
4.6 Visão sistêmica de um nó sensor ZigBee criado. 63
4.7 Nó sensor ZigBee implantado no CPD da prefeitura. 63

4.8 Dispositivo *Telemetric*. 64
4.9 Centro de gerência da Infovia, prédio da Prefeitura e do Morro do Cristo utilizados no estudo de caso. 65
4.10 *Gateway* ConnectPort X2. 67
4.11 Exemplos de *request-response* RCI. 68
4.12 Interface *Web service* do ConnectPort. 68
4.13 Parte da descrição do serviço *query state*. 69
4.14 Requisição e resposta de estado de um rádio XBee. 70
4.15 XML *Schema* para os dados coletados. 71
4.16 Arquivo XML de um dado coletado. 71
4.17 Comunicação do *Telemetric* com a camada de aplicação através do *Gateway* em Python. 72
4.18 Modelo entidade relacionamento do sistema desenvolvido. 74
4.19 Organização do código no padrão MVC. 75
4.20 URL de compartilhamento de dados via *Web service*. 75
4.21 Dados reportados em formatos distintos. 76
4.22 Autenticação no portal via *smartphone*. 77
4.23 Seção monitorada na prefeitura de Pedreira. 78
4.24 Rádios da rede ZigBee implantada na prefeitura. 79
4.25 Configurações e estado do nó ZigBee XBee CPD implantado. 80
4.26 Configurações do *gateway* ZigBee. 81
4.27 Módulo de visualização de dados. 82
4.28 *Telemetric* monitorado no Morro do Cristo 83
4.29 VLAN utilizada no estudo de caso. 84
4.30 Médias diárias de temperatura coletadas durante duas semanas. 85
4.31 Médias horárias de temperatura coletadas durante o mês de abril. 85
4.32 RSSI e LQI da comunicação entre o coordenador e o rádio do CPD. 87
4.33 RSSI e LQI da comunicação entre o coordenador e o rádio da sala. 87
4.34 Variação de tensão do nó XBee CPD durante um dia. 88
4.35 Valores médios de tensão coletados durante duas semanas. 89
4.36 Monitoramento da tensão AC em um período de cinco horas. 90
4.37 Mudança de estado em uma das tomadas monitoradas. 90
4.38 Médias diárias de temperatura durante duas semanas do mês do maio. 91

Lista de Tabelas

1.1 Evolução dos nós sensores ao longo dos anos (Chong & Kumar 2003). 8
1.2 Endereçamento ZigBee. 13
1.3 Comparativo entre padrões de comunicação sem fio 14
1.4 Exemplos de RMAA (Miani 2009). 17

4.1 Pontos interconectados pela RMAA de Pedreira. 56
4.2 Especificações do Arduino Uno R3. 62
4.3 Especificações do *gateway* ConnectPort X2. 66
4.4 API REST criada. 76
4.5 Informações estatísticas dos dados coletados. 86
4.6 Dados estatísticos da comunicação ZigBee. 88
4.7 Estatísticas da tensão AC monitorada com o *Telemetric*. 90
4.8 Estatísticas da temperatura coletada pelo *Telemetric*. 91

Lista de Acrônimos e Notação

AC	*Alternating Current*
AES	*Advanced Encryption Standard*
API	*Application Programming Interface*
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance*
DARPA	*Defense Advanced Research Project Agency*
DC	*Direct Current*
DHCP	*Dynamic Host Configuration Protocol*
GPRS	*General Packet Radio Service*
HTML	*HyperText Markup Language*
HTTP	*Hypertext Transfer Protocol*
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
IP	*Internet Protocol*
JSON	*JavaScript Object Notation*
LQI	*Link Quality Indicator*
LAN	*Local Area Network*
LaRCom	Laboratório de Redes de Comunicações
MAC	*Medium Access Control*
MAN	*Metropolitan Area Network*
MANET	*Mobile Ad hoc Network*
MEMS	*Microelectromechanical systems*
MVC	*Model-View-Controller*
PHP	*HyperText Preprocessor*
QoS	*Quality of Service*
RCI	*Remote Command Interface*
REST	*Representation State Transfer*
RMAA	Rede Metropolitana de Acesso Aberto
RFC	*Request for Comments*
RPC	*Remote Procedure Call*
RSASF	Rede de Sensores e Atuadores Sem Fio
RSSF	Rede de Sensores Sem Fio
RSSI	*Received Signal Strength Indicator*
SOA	*Service Oriented Architecture*
SOAP	*Simple Object Access Protocol*

TCP	*Transmission Control Protocol*
UDP	*User Datagram Procotol*
VLAN	*Virtual LAN*
VoIP	*Voice over Internet Protocol*
W3C	*World Wide Web Consortium*
WPAN	*Wireless Personal Area Network*
WSDL	*Web Services Description Language*
UDDI	*Universal Description, Discovery and Integration*
UML	*Unified Modeling Language*
XML	*eXtensible Markup Language*

Introdução

A convergência entre redes ópticas e *wireless* de alta velocidade representa um dos principais candidatos para uma única rede convergente universal (Breda et al. 2011). Tal evolução permite a construção de redes de comunicações digitais, que proporcionam o oferecimento de imensa quantidade de serviços, no sentido de atender aos usuários finais. Esses avanços têm impulsionado o desenvolvimento das Cidades Digitais.

Existem diversas definições na literatura para Cidades Digitais, podendo ser classificadas de acordo com o seu objetivo em: comerciais, governamentais, virtuais e multiuso (Panhan 2011). Independente do seu objetivo, uma Cidade Digital é constituída por uma infraestrutura de comunicação digital e promove serviços e benefícios para entidades de governo, empresas e cidadãos.

No Brasil, o projeto de Redes Metropolitanas de Acesso Aberto (RMAA), ou Infovias municipais (Mendes, Botolli & Breda 2010), desenvolvido pelo Laboratório de Redes de Comunicações (LaRCom) da Universidade Estadual de Campinas (UNICAMP) desde 2001, tem por objetivo construir infraestruturas de comunicação para a criação das Cidades Digitais. De acordo com Mendes (2008), uma Infovia municipal é uma rede de alta velocidade, baseada em infraestrutura formada por rede óptica e células de acesso sem fio. Os principais protocolos de comunicação utilizados podem ser encontrados na pilha TCP/IP: 10G Ethernet, IEEE 802.11, IEEE 802.16, Gigabit Ethernet. Esses protocolos formam um ambiente genérico onde a escalabilidade e a flexibilidade são facilmente alcançadas, permitindo a transmissão de diferentes tipos de tráfego em sua rede e consequentemente, a possibilidade de oferecimento de uma ampla quantidade de serviços.

Nesse contexto, as cidades estão optando por implementar soluções tecnológicas com o objetivo de resolver problemas reais dos municípios que podem ser provenientes, por exemplo, do crescimento populacional. Esse crescimento pode afetar a entrega de serviços básicos como eliminação de resíduos, energia elétrica, água potável, transporte e saneamento básico (Bartlett et al. 2011). Questões relacionadas aos níveis de qualidade do ar também são preocupantes em virtude da poluição.

Implementações de tecnologias avançadas são essenciais para lidar com esses desafios. Uma tecnologia cada vez mais fundamental para as Cidades Digitais, no sentido de torná-las ainda mais inteligentes e ajudar a resolver os problemas supracitados, são as redes de sensores (Juraschek et al. 2012, Hernández-Muñoz et al. 2011, Bartlett et al. 2011). As redes de sensores

representam um importante meio para a coleta de informações e atuação em ambientes de interesse. Avanços em comunicações *wireless* e microeletrônica possibilitaram o desenvolvimento de sensores de baixo custo/consumo, com dimensões cada vez menores (Akyildiz et al. 2002). Sua aplicação abrange diversas áreas do conhecimento, incluindo saúde, militar, industrial, tráfego e ambiental. O estado da arte de redes de sensores consiste de redes de área pessoal sem fio para a construção de sistemas sensores de baixo custo, baixo consumo de energia e baixa taxa de transferência de dados. Dentre as especificações existentes destaca-se atualmente o padrão IEEE 802.15.4/ZigBee (Fan & Guo 2011).

A chave para melhorar as operações de uma cidade é a tomada de decisão inteligente. Isso depende da disponibilidade, acurácia e quantidade suficiente de informações. Redes de sensores são empregadas em cenários urbanos para coletar dados, com o objetivo de propor melhorias em termos de transporte, monitoramento ambiental e segurança pública (Hernández-Muñoz et al. 2011). Nesse sentido, vários trabalhos têm proposto sistemas e aplicações para contribuir na melhora de serviços oferecidos, ao mesmo tempo que promove benefícios econômicos para o município através da redução de custos.

Em cenários urbanos, as principais informações coletadas por redes de sensores estão relacionadas às condições ambientais do município, em termos de qualidade do ar, da água potável e do tráfego de veículos. Essas informações são de interesse comum para cidadãos, empresas e órgãos do governo. Nesse sentido, é importante que os sistemas responsáveis pela coleta dessas informações, que em geral são heterogêneos em termos de hardware e software, compartilhem seus dados de maneira padronizada, para que sistemas externos (clientes) sejam capazes de obtê-los de modo unificado.

Além de interfaces genéricas para acesso aos dados coletados, são necessários meios para promover a interoperabilidade entre sistemas sensores em nível municipal. Essa interoperabilidade corresponde a capacidade de acesso unificado aos diferentes sistemas para troca de informações de modo padronizado. Em termos de sistemas distribuídos, *Web services* (W3C 2004) constituem um modelo baseado em protocolos e padrões abertos, independentes de plataforma e de linguagem de programação, que permite fornecer interfaces genéricas para comunicação em rede.

Tendo em vista a diversidade de ambientes passíveis de monitoramento em um município e a heterogeneidade de dispositivos empregados para realizar tal atividade, a proposta dessa dissertação é combinar as vantagens de redes de sensores e as oportunidades fornecidas por Redes Metropolitanas de Acesso Aberto. Com isso, concebemos uma arquitetura baseada no modelo de orientação a serviços para o monitoramento e gerenciamento unificado de redes de sensores nesse cenário. A arquitetura promove a interoperabilidade através da utilização de *Web services* para a criação de interfaces de compartilhamento de dados e de comunicação homogênea entre sistemas sensores heterogêneos.

Objetivos e contribuições

As tecnologias em termos de comunicação *wireless* e capacidade computacional de redes de sensores têm evoluído consideravelmente nos últimos anos. Como consequência disso, aumentou

consideravelmente a complexidade e a heterogeneidade tanto no tocante a hardware quanto a software.

Dessa maneira, o objetivo desse trabalho é propor uma arquitetura baseada na *Web* para o monitoramento de redes sensoras aplicadas em cenários metropolitanos. A arquitetura deve consistir de tecnologias atuais de redes de sensores sem fio com o intuito de criar sistemas em nível municipal. Deve também usufruir da ubiquidade da *Web*, utilizando *Web services* para compartilhamento dos dados coletados pelos sensores e também para estabelecer mecanismos de comunicação unificados e homogêneos entre diferentes sistemas sensores espalhados ao longo de um município, consistindo de equipamentos com diferentes capacidades e funcionalidades. A arquitetura pretende utilizar os recursos existentes em uma Rede Metropolitana de Acesso Aberto, responsável por interconectar ambientes importantes em um município.

Destacam-se como contribuições deste trabalho:

- A proposta de uma arquitetura para monitorar redes sensoras em Redes Metropolitanas de Acesso Aberto, capaz de integrar diferentes sistemas sensores através do uso de padrões abertos e independentes de plataforma;
- A criação de um ambiente integrado para o monitoramento de redes de sensores em escala municipal, capaz de gerenciar múltiplas aplicações e centralizar informações de domínios diferentes, constituindo uma importante base de conhecimento;
- A criação de um ambiente interoperável e escalável que fornece baixo acoplamento entre os componentes e protocolos de comunicação empregados nos sistemas sensores em cenários urbanos;
- Permitir o compartilhamento dos dados coletados com sistemas externos no contexto municipal, através de interfaces genéricas baseadas em *Web services*, para que sistemas inteligentes possam utilizar informações com a finalidade de realizar análises, encontrar padrões, propor melhorias e reagir proativamente a eventos;
- O trabalho contribui também para a evolução das RMAA, acrescentando uma importante funcionalidade ao seu caráter multisserviço.

Organização

Este trabalho encontra-se estruturado em cinco capítulos, da forma como segue:

- O Capítulo 1 apresenta dois dos principais conceitos tecnológicos abordados nessa dissertação: o de redes de sensores, especificamente as sem fio, e também o de Redes Metropolitanas de Acesso Aberto. No que diz respeito as RSSF, são descritas as singularidades desse tipo de rede e é relatado em maiores detalhes a especificação IEEE 802.15.4/ZigBee, que representa um dos padrões de rede pessoal sem fio mais utilizados na atualidade. Com relação às RMAAs, esse capítulo expõe a infraestrutura desse conceito de redes de comunicação, bem como a concepção de soluções para Cidades Digitais, criadas sobre essa

infraestrutura de comunicação. Alguns dos serviços proporcionados nesse cenário também são expostos. Após apresentar as tecnologias que fundamentam o trabalho, trabalhos relacionados são exibidos.

- A arquitetura proposta se enquadra na categoria de Arquitetura Orientada a Serviços (SOA). Assim, o Capítulo 2 apresenta o conceito desse estilo arquitetural e descreve, de maneira breve, as alternativas estratégicas e tecnológicas empregadas na criação de serviços *Web*, com o intuito de proporcionar independência de plataforma e interoperabilidade entre aplicações e sistemas.

- A arquitetura proposta é apresentada detalhadamente no Capítulo 3. Também neste capítulo sãs expostas as camadas que a compõem e as tecnologias empregadas em cada uma, bem como os mecanismos para o estabelecimento da comunicação entre elas. Visto que a segurança da informação é uma questão importante em aplicações que envolvem sensoriamento, são descritas, também, diferentes abordagens que podem ser empregadas nas camadas que compõem arquitetura, de modo a propiciar diferentes níveis de segurança.

- O Capítulo 4 descreve o processo de desenvolvimento de um sistema para monitoramento de redes sensoras nos moldes da arquitetura proposta, e os resultados obtidos a partir de sua aplicação Cidade Digital de Pedreira. O capítulo expõe a modelagem de software realizada e os recursos de hardware utilizados para a concepção de um sistema de monitoramento ambiental em ambientes públicos da cidade.

- Finalmente, no Capítulo 5 estão as principais conclusões e as sugestões para trabalhos futuros.

Capítulo 1

Revisão de conceitos tecnológicos

Este capítulo define alguns dos conceitos abordados no trabalho, considerando aspectos tecnológicos de redes de sensores sem fio e Redes Metropolitanas de Acesso Aberto. O potencial de cada uma dessas tecnologias e os benefícios que sua utilização conjunta podem oferecer ficam evidentes a partir deste capítulo revisório. Também são apresentados trabalhos relacionados que utilizam as tecnologias apresentadas.

1.1 Redes de Sensores Sem Fio (RSSF)

Esta seção descreve as principais especificações de redes de sensores sem fio, apresentando as similaridades e diferenças com as redes tradicionais, bem como as singularidades que as tornam adequadas para aplicações de diversas naturezas. São exibidos os detalhes da especificação IEEE 802.15.4/ZigBee, amplamente utilizada atualmente em RSSF e que foi objeto de estudo neste trabalho.

1.1.1 Características

Assim como várias tecnologias, as redes de sensores surgiram em aplicações desenvolvidas para fins militares. Durante a guerra fria, o *Sound Surveillance System* (SOSUS), um sistema de sensores acústicos (microfones) localizados em posições estratégicas no fundo do oceano, foi desenvolvido pela *Defense Advanced Research Project Agency* (DARPA), com o objetivo de fazer um acompanhamento silencioso de submarinos soviéticos (Chong & Kumar 2003). Pesquisas mais modernas sobre as redes de sensores tiveram início na década de 80 com o programa *Distributed Sensor Networks* (DSN), também da DARPA, sendo voltado principalmente para monitoramento de território em campos de batalha e observação de movimentação inimiga. Com o advento da comunicação sem fio as redes de sensores passaram a ser empregadas em diferentes cenários e para as mais diversas finalidades.

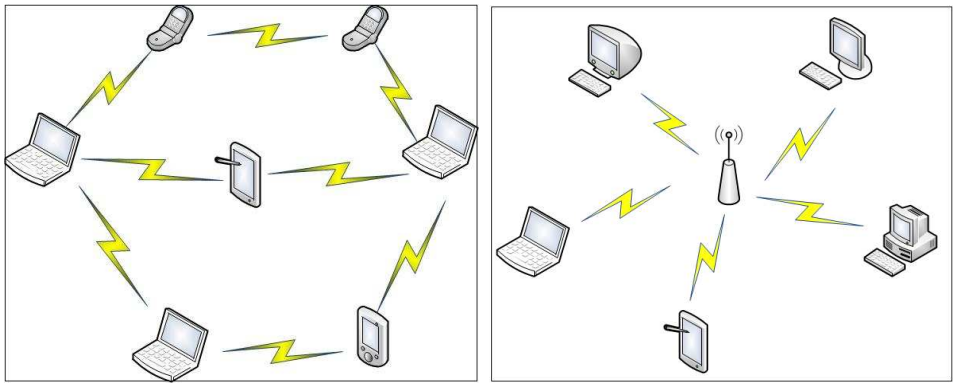
Atualmente, as redes de sensores são usadas em aplicações como: monitoramento de segurança, controle e acompanhamento industrial, monitoramento ambiental, controle e monitoramento de tráfego de veículos, assistência médica, práticas militares e automação doméstica.

"Uma rede de sensores é composta de um grande número de nós densamente implantados

próximos a fenômenos de interesse" (Akyildiz et al. 2002). Em termos de comunicação, as redes de sensores podem ser cabeadas (*wired*) ou sem fio (*wireless*), dependendo do cenário de aplicação. No entanto, a alternativa sem fio é mais amplamente estudada do que a primeira, em virtude de suas características, como flexibilidade de implantação, o que possibilita o seu emprego no desenvolvimento de aplicações para diversas áreas.

RSSF são diferentes de redes de computadores convencionais de diversas maneiras. Podem ser vistas como um tipo especial de Rede Móvel Ad hoc (MANET), ou seja, nesse tipo de rede não há a necessidade de uma rede infraestruturada para a comunicação entre os nós. Normalmente essas redes possuem um grande número de nós distribuídos, apresentam restrições de energia e devem possuir mecanismos para auto-organização e adaptação devido à falhas de comunicação ou escassez de energia (Loureiro et al. 2003).

A Figura 1.1 apresenta exemplos de MANET e de uma rede infraestruturada.



(a) MANET. (b) Rede infraestruturada.

Figura 1.1: Exemplos de redes.

Conforme exposto na Figura 1.1, MANETs estabelecem comunicação entre os seus integrantes sem que haja a necessidade de uma infraestrutura, como acontece em redes tradicionais, onde os dispositivos estabelecem comunicação entre si e com entidades externas por intermédio de pontos de acesso. Desta forma, RSSF são consideradas um tipo de MANET. Porém, o funcionamento e os objetivos das duas são diferentes. Em redes de sensores os nós trabalham colaborativamente de maneira a prover os dados coletados para nós especiais, denominados sorvedouros (*sinks*), para que esses possam fazer o processamento e posterior envio das informações para a aplicação responsável pelo seu tratamento. A Figura 1.2 exibe um exemplo de rede de sensores sem fio.

Como pode-se observar na Figura 1.2, uma RSSF é tipicamente formada pelos seguintes elementos: nós sensores, *gateways* e sorvedouros. Os nós sorvedouros em uma rede de sensores podem desempenhar diferentes funções, uma vez que a arquitetura e implementação dessa é totalmente dependente da aplicação que a utiliza (*application-specific*). Porém, de modo geral, o sorvedouro tende a possuir maior capacidade computacional que os demais, pois esse nó pode realizar o processamento local e transmitir somente informações sumarizadas para a estação responsável pelo tratamento e apresentação dos dados. Em muitos casos o nó sorvedouro é ao mesmo tempo o *gateway* entre a rede de sensores e a estação de processamento, ou pode ser o único elemento a se comunicar diretamente com um *gateway*, conforme a Figura 1.2.

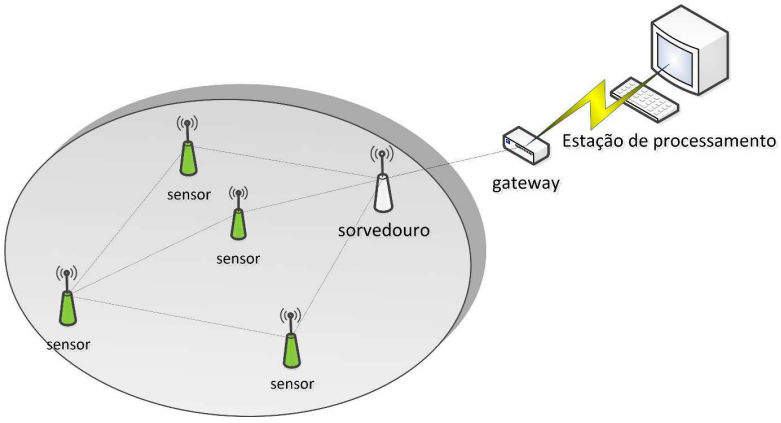


Figura 1.2: Exemplo de rede de sensores.

Além de terminais sensores as RSSF podem conter também nós que são capazes de intervir no ambiente monitorado. São os chamados atuadores, que são descritos na próxima seção.

1.1.2 Sensores e atuadores

Como descrito no seguimento 1.1, as redes de sensores sem fio podem ser vistas como um tipo especial de MANET, com algumas similaridades e também certas diferenças. Uma importante diferença é com relação aos dispositivos que as compõem. Em RSSF podem existir terminais sensores e também atuadores.

Um nó sensor é um dispositivo capaz de perceber fenômenos de interesse no ambiente onde está inserido, coletar e enviar essas informações para entidades responsáveis pelo posterior tratamento dos dados. Em geral, um sensor apresenta baixo poder computacional e possui restrições de energia.

Um dispositivo sensor básico possui quatro módulos, conforme exposto na Figura 1.3.

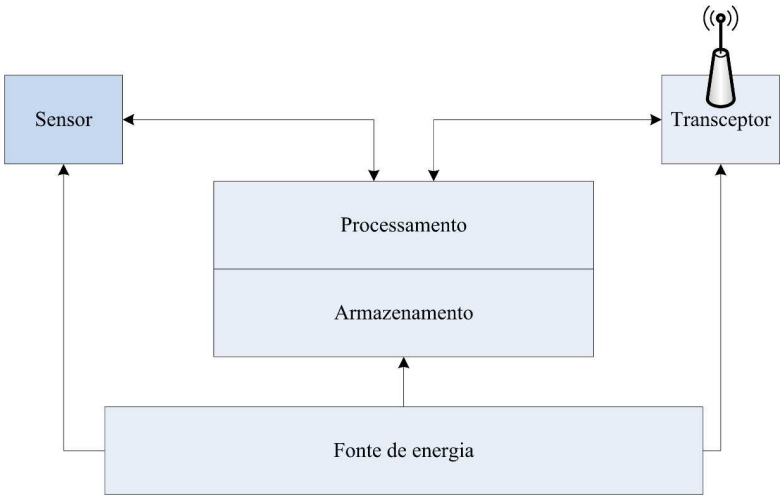


Figura 1.3: Módulos básicos de um nó sensor (Adaptada de Wu & Tseng (2007, Capítulo 2)).

Os módulos expostos na Figura 1.3 são (Wu & Tseng 2007):

- *módulo de energia*: representa a principal preocupação nesses dispositivos. Em geral trata-se de uma fonte de energia finita e é usualmente formado por pequenas baterias. Uma vez que a reposição de baterias em sensores pode não ser possível em virtude das condições de implantação, estratégias para a economia de energia devem ser adotadas, como protocolos leves e modos de gerenciamento eficientes;

- *módulo de processamento*: responde pelo gerenciamento dos procedimentos executados pelo nó, incluindo comunicação e sensoriamento. Possui uma unidade para armazenamento do sistema operacional, algoritmos e protocolos. Por possuir baixo poder de processamento, os algoritmos e protocolos devem ser leves para não causar *overhead* no dispositivo;

- *módulo de sensoriamento*: é responsável pela percepção de fenômenos e geração de sinais a partir desses. Esses sinais são convertidos através de um módulo Analógico-Digital (A/D) e então enviados para a unidade de processamento. Um dispositivo pode conter mais de um módulo de sensoriamento, ou seja, pode possuir vários sensores embarcados;

- *módulo transceptor*: esse módulo é o responsável pela comunicação do nó com os demais dispositivos na rede. A transmissão e recepção de sinais para um nó sorvedouro ou para outros nós na rede é feita através desse módulo. A comunicação pode se dar por meio de infravermelho ou radiofrequência, por exemplo.

A crescente evolução em termos de Sistemas Microeletromecânicos (MEMS) tem tornado os dispositivos sensores e outros módulos para a formação de RSSF cada vez mais compactos, poderosos e também mais baratos, devido a sua produção em larga escala (Chong & Kumar 2003).

A tabela 1.1 apresenta algumas características dessa evolução com o passar dos anos, de acordo com Chong & Kumar (2003).

É evidente que a última coluna da tabela 1.1 trata-se de uma perspectiva feita pelos autores sobre o futuro dos dispositivos sensores de acordo com a sua visão na época. Contudo, nota-se que as previsões feitas por eles correspondem, em boa parte, a realidade atual, exceto que em certas redes de sensores, principalmente em Redes de Sensores Corporais (RSC) sem fio, o emprego da topologia em estrela ainda é predominante.

Em adição as tarefas de sensoriamento alguns nós podem possuir a capacidade de intervir no ambiente onde estão inseridos, para controlar equipamentos. Esses são conhecidos como atuadores, de modo que juntamente com o uso de nós sensores formam uma rede de sensores e atuadores sem fio (RSASF), ou *Wireless Sensor-actuator Network* (WSAN) (Rezgui & Eltoweissy 2007).

Nós sensores e atuadores se comunicam e colaboram com o objetivo de cumprir tarefas específicas. Sensores coletam informações sobre o ambiente onde estão inseridos enquanto que atuadores tomam decisões e realizam ações que mudam o estado do ambiente (Rezgui & Eltoweissy 2007), como temperatura e iluminação, bem como propriedades intrínsecas a ele e a outros nós.

Tabela 1.1: Evolução dos nós sensores ao longo dos anos (Chong & Kumar 2003).

Característica	**1980 - 1990**	**2000 - 2003**	**2010 - hoje**
Tamanho	Dimensões maiores que a de uma caixa de sapatos	Do tamanho de um *deck* de cartas, até menores que uma caixa de sapatos	Pequenas partículas (poeira)
Peso	Quilogramas	Gramas	Desprezível
Arquitetura	Processamento, comunicação e sensoriamento separados	Sensoriamento, processamento e comunicação integrados	Sensoriamento, processamento e comunicação integrados
Fonte de energia/Tempo de vida	Grandes baterias/horas, dias	Baterias AA/dias, semanas	Solar/meses a anos
Topologia	Point-to-point, estrela	Cliente-servidor, peer-to-peer	Peer-to-peer

Como as redes de sensores são de aplicação específica, dependendo do contexto, os nós podem ser estacionários ou móveis.

A Figura 1.4 exibe o fluxo de controle em uma rede composta por atuadores e sensores.

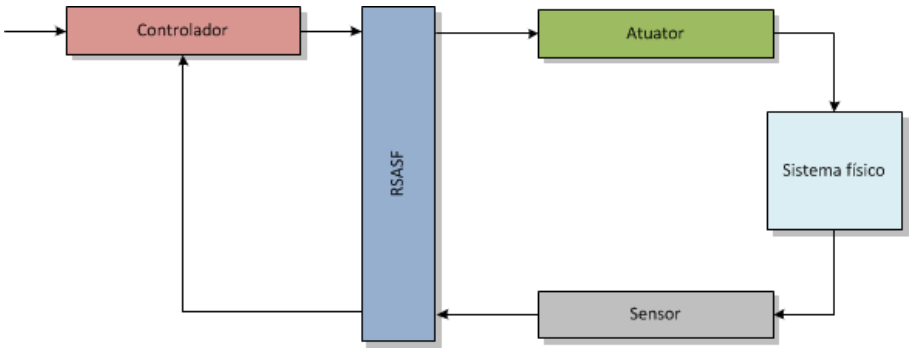


Figura 1.4: Fluxo de controle de uma WSAN. (Adaptada de Xia et al. (2007)).

Observando-se a Figura 1.4 pode-se notar que uma ou mais entidades controlam a WSAN. Os controladores podem ser módulos embarcados nas estações de processamento ou ainda nós separados, equipados com módulos de comunicação e computação suficientes para realizar tal tarefa. Assim, os sensores coletam dados e enviam para as entidades controladoras que podem tomar a decisão de acionar dispositivos capazes de interagir com o ambiente monitorado (Xia et al. 2007).

Conforme as características apresentadas, o uso de terminais sensores e atuadores sustenta grande potencial para automação, seja doméstica, comercial ou industrial. Em termos de comunicação entre as entidades de uma RSSF, diversos padrões podem ser utilizados, inclusive os existentes em redes locais ou metropolitanas. No entanto, as redes de área pessoal sem fio (*Wireless Personal Area Network* (WPAN)) ganharam espaço nos últimos anos, apresentando-se

como uma alternativa de baixo custo, baixo consumo de energia e curto alcance de transmissão. Dentre os padrões existentes nessa categoria destaca-se a especificação ZigBee. O padrão ZigBee trabalha com o protocolo para redes de área pessoal sem fio de baixa potência, IEEE 802.15.4. Ambos são descritos na próxima seção.

1.1.3 IEEE 802.15.4/ZigBee

O IEEE 802.15.4, também conhecido como *Low-rate Wireless Personal Area Network* (LR-WPAN), é um protocolo para redes simples, com comunicação de baixo custo, que permite a conectividade de aplicações que apresentam restrições de energia e baixo requisito de rendimento (IEEE-Standard 2011). O protocolo IEEE 802.15.4 é conhecido por trabalhar com a especificação ZigBee nas camadas superiores, utilizando rádios de curto alcance e baixa capacidade de transmissão. A especificação ZigBee, por sua vez, é mantida pela ZigBee Alliance, um grupo formado por mais de 200 empresas com o intuito de definir padrões para a utilização de redes de sensores sem fio em áreas como saúde, energia inteligente, monitoramento doméstico e industrial e automação. Dentre as empresas que formam o grupo estão Philips, Motorola, Sony, Honeywell e Ericsson (Zigbee-Alliance 2012).

O padrão IEEE 802.15.4 define o gerenciamento de energia, endereçamento, correção de erros, formatos de mensagens e outras especificações necessárias para a comunicação entre dois ou mais rádios (Faludi 2010).

Algumas das capacidades providas pelo padrão IEEE 802.15.4 são (IEEE-Standard 2011):

- operação em estrela e *peer-to-peer*;
- endereço único extendido de 64 bits (MAC) ou endereço alocado de 16 bits;
- Acesso Múltiplo com Verificação de Portadora com Prevenção de Colisão (CSMA/CA);
- protocolo reconhecido pela transferência confiável;
- baixo consumo de energia;
- Detecção de energia (ED);
- Indicação de qualidade de *link* (LQI);

Além disso, o padrão IEEE 802.15.4/ZigBee opera na faixa de frequência Industrial, Científica e Médica (ISM), ou seja, 2.4 GHz com 16 canais, além de 10 canais operando em 915 MHz e outro canal em 868 MHz. As taxas de transferência de dados em cada uma das frequências são 250, 40 e 20 Kbps, respectivamente.

Como mencionado, o IEEE 802.15.4 opera nas camadas inferiores do protocolo ZigBee, definindo as responsabilidades das camadas física e também de Controle de Acesso ao Meio (MAC). O ZigBee é responsável pelas camadas de rede, de segurança e também de aplicação.

A camada de aplicação é definida por perfis, dos quais há dois tipos: públicos, que são certificados pela ZigBee Alliance para propósitos de interoperabilidade; e privados, para uso em sistemas proprietários (Patil et al. 2012).

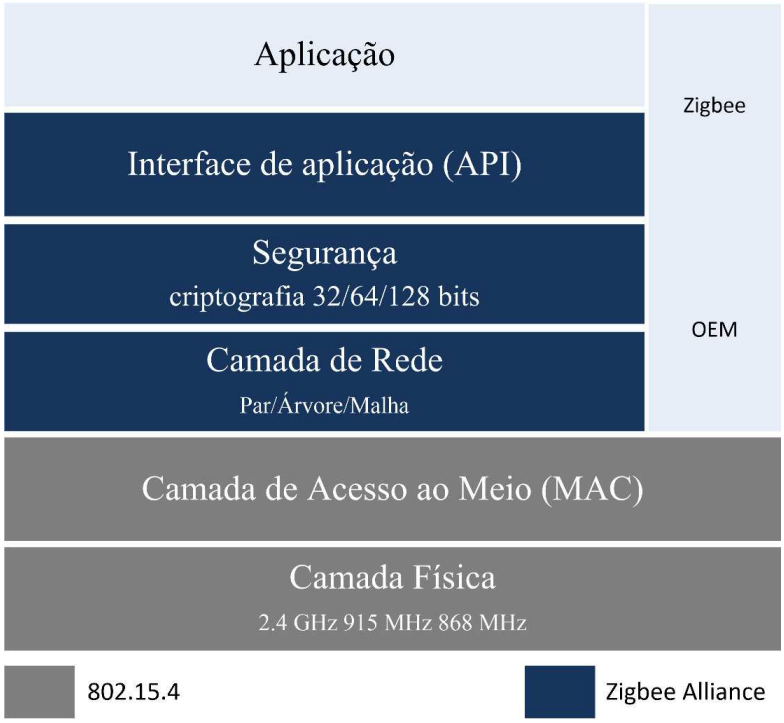


Figura 1.5: Pilha de protocolos ZigBee.

A Figura 1.5 exibe as camadas que compõem a especificação ZigBee.

A camada de aplicação consiste de subcamadas de suporte à aplicação, objeto de dispositivo ZigBee (ZDO) e objetos de aplicação definidos pelo fabricante. Camadas de suporte implementam perfis de aplicação, *clusters* e *endpoints*.

Perfis de aplicação ZigBee são definições comuns e protocolos que permitem que diversos dispositivos operem em conjunto em um domínio particular, tal como monitoramento industrial. Cada perfil define tipos de dispositivos e funcionalidades necessárias. Como mencionado, perfis públicos são mantidos pela ZigBee Alliance e certificam que dispositivos de fornecedores diferentes podem interoperar (Faludi 2010).

Clusters são coleções de funcionalidades que as aplicações podem realizar de acordo com um perfil. Um *cluster* corresponde a um serviço ou ação em particular. Já o papel do *endpoint* é descrever o que determinado dispositivo está executando. É importante para casos onde o dispositivo pertence a diferentes perfis de aplicação.

O fato de o padrão ZigBee ser utilizado em diversas áreas torna necessária a implementação de métodos de criptografia para proteção de dados importantes. Para utilizar criptografia em redes ZigBee é preciso analisar a real necessidade de seu uso, uma vez que isso resulta no aumento do consumo de energia e implica em maior esforço de desenvolvimento (Faludi 2010).

Em nível de segurança, o padrão ZigBee utiliza criptografia de chave simétrica *Advanced Encryption Standard* (AES) de 128 bits, implementando chaves de rede e de *link*. Uma chave de rede é conhecida por todos os nós que fazem parte desta. Ela protege os *frames* de dados que trafegam entre os nós, onde cada nó decodifica e criptografa novamente os dados para envio pela rede. Esse processo adiciona um atraso de transmissão. Já as chaves de *link* oferecem uma camada adicional de proteção fim-a-fim. A chave gerada é conhecida somente pelos dois

nós que estão se comunicando. Dessa maneira, os dados são criptografados pelo remetente e permanecem assim até o seu destino, o qual é o único capaz de descriptografá-los (Faludi 2010). Essa abordagem previne que nós intermediários examinem os dados transmitidos.

Na camada de rede o protocolo ZigBee atua implementando funções de endereçamento e roteamento entre os nós. O padrão suporta três tipos de topologias, conforme exposto na Figura 1.6.

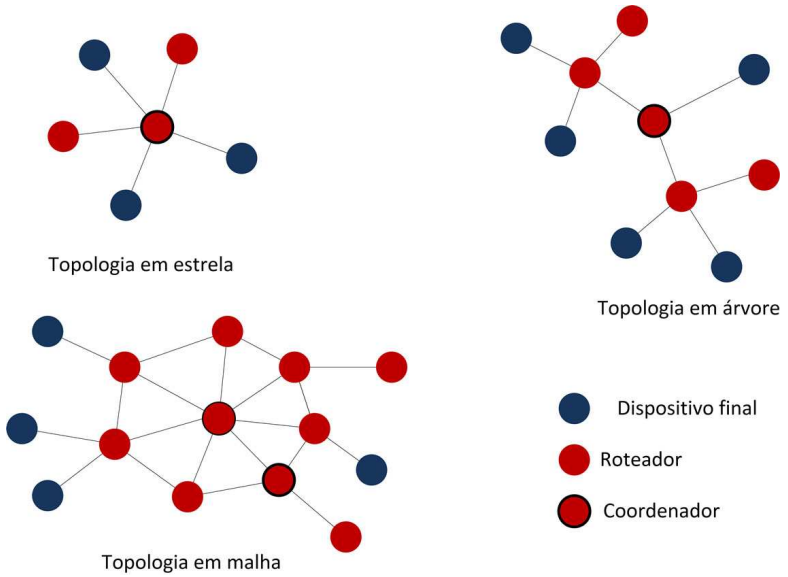


Figura 1.6: Topologias de rede ZigBee.

Como pode ser observado, o padrão ZigBee suporta topologias ponto-a-ponto e multiponto, resultando nas formações apresentadas na Figura 1.6, sendo estas:

- *estrela*: um coordenador central é o responsável por gerenciar o fluxo de mensagens entre os terminais da rede. Todos os dados são direcionados para o coordenador;
- *malha*: tipo mais complexo de rede. Além do coordenador, esse tipo de rede utiliza diversos nós roteadores para a troca de mensagens entre os dispositivos. O coordenador nesse caso atua para gerenciar a rede, mas também pode executar funções de roteamento;
- *árvore*: nesse tipo de rede os roteadores formam uma espécie de *backbone* com terminais conectados a cada um deles. É bastante similar a topologia em malha.

Como pode ser observado na Figura 1.6, os dispositivos em uma rede ZigBee podem assumir diferentes papéis. As funções desempenhadas por um nó são dependentes das características desse, que podem ser de dois tipos: Dispositivo de Função Completa (*Full Function Device* (FFD)) e Dispositivo de Função Reduzida (*Reduced Function Device* (RFD)). Um FFD é um dispositivo com maior capacidade computacional que pode atuar tanto como coordenador da rede quanto como roteador. Já um RFD possui hardware básico e por isso só pode atuar como dispositivo de borda de rede (Terminal).

As funções que um nó ZigBee pode desempenhar são:

- *coordenador*: cada rede ZigBee possui sempre um único coordenador, que é o responsável pela formação, atribuição de endereços aos nós e gerenciamento de outras funções de configuração de rede, como segurança;

- *roteador*: é um dispositivo de função completa responsável pelo roteamento de dados, gerenciando as mensagens trocadas entre outros nós em uma rede;

- *terminal*: em geral é um dispositivo de função reduzida (RFD), mas nada impede que um FFD seja utilizado. Esse dispositivo só pode se comunicar com o coordenador da rede ou com roteadores, nunca entre si. São mais econômicos com relação ao consumo de energia e podem implementar o modo *sleep*, de maneira a ficarem temporariamente indisponíveis com o intuito de economizar energia.

Ainda relacionado a camada de rede, consoante ao endereçamento, os nós em uma rede ZigBee possuem dois tipos de endereços principais. Primeiro, cada nó possui o seu endereço único e permanente, o seu número de série de 64 bits. Esse endereço é globalmente único e dessa forma nenhum rádio possuirá um mesmo endereço de 64 bits que outro. Há também um endereço de 16 bits, análogo ao endereço IP. Esse endereço é dinâmico e atribuído pelo coordenador da rede assim que o dispositivo se junta a ela. Além desses endereços, de maneira opcional, a um nó pode ser atribuído um rótulo, um texto de identificação para o dispositivo. Podem ser utilizados nomes de acordo com as funções desempenhas pelo nó, por exemplo. Esse endereço não tem unicidade garantida.

Para ilustrar o endereçamento de nós em uma rede ZigBee, a tabela 1.2 exibe os endereços de um dos nós utilizados no estudo de caso realizado nesse trabalho.

Tabela 1.2: Endereçamento ZigBee.

Tipo	**Exemplo**	**Único**
64-bit	00:13:a2:00:40:ac:6c:3b	Sim, sempre e em todo lugar
16-bit	0xCB0D	Sim, somente dentro da rede
Rótulo do nó	XBee CPD	Unicidade não garantida

Além dos endereços listados na tabela 1.2, cada nó integrante guarda o endereço da rede a qual ele pertence, o chamado *Personal Area Network ID* (PAN ID). Esse endereço é representado em hexadecimal por valores entre 0x0 e 0xFFFF (Faludi 2010).

O baixo custo para a utilização da especificação ZigBee permite que a tecnologia seja altamente empregada em aplicações de controle e monitoramento sem fio. O baixo consumo de energia garante mais tempo de vida à rede apenas com o uso de pequenas baterias, e as topologias suportadas fornecem confiabilidade em larga escala (Patil et al. 2012).

Embora existam diversos padrões para comunicação sem fio utilizados em redes de sensores, tais como IEEE 802.11 **b** e **g**, HomeRF e Bluetooth, o padrão IEEE 802.15.4/ZigBee destaca-se atualmente para tal finalidade. Isso porquê o objetivo da criação do protocolo IEEE 802.15.4 é justamente prover comunicação de baixa potência, curto alcance e com baixo consumo de energia.

Tabela 1.3: Comparativo entre padrões de comunicação sem fio

	Wi-FI (802.11b)	**Bluetooth** (802.15.1)	**ZigBee** (802.15.4)
Aplicação	Redes corporativas; LANs domésticas	Substituição aos cabos; Redes Ad-hoc para troca de dados e voz	Automação e redes de sensores
Taxa de transmissão	11 Mbps	1 Mbps	250 Kbps
Tempo de vida das baterias	Curto prazo	Médio prazo	Longo prazo
Alcance de transmissão	100 metros	10 metros	$>$ 80 metros

A tabela 1.3 exibe um comparativo entre as características dos principais padrões de comunicação sem fio empregados na concepção de sistemas sensores.

O padrão IEEE 802.11b é utilizado em redes que exigem altas taxas de transferência de dados. É de fácil integração com redes Ethernet e possui longo alcance de transmissão. Em virtude da alta taxa de transferência de dados, esse padrão consome mais energia que os demais apresentados, conforme a tabela 1.3, o que o torna inadequado para boa parte das aplicações de redes de sensores sem fio.

O Bluetooth é uma padrão criado com a finalidade de conectar telefones celulares a outros dispositivos, como notebooks e *Desktops*, sem a necessidade de cabos, utilizando rádios de baixa potência e de baixo custo (Tanenbaum & Wetherall 2011). Uma rede Bluetooth é formada por um mestre, que controla a rede, e por escravos. Cada mestre pode ter no máximo 8 escravos conectados a ele formando uma *piconet*. A conexão de duas ou mais *piconets* forma uma *scatternet*. Esse padrão utiliza a técnica de transmissão *Frequency Hopping Spread Spectrum* (Espalhamento espectral com saltos em frequência) (FHSS), que permite que os dispositivos se comuniquem mesmo em locais com muita interferência eletromagnética. As principais desvantagens do Bluetooth em comparação com o ZigBee são o tempo de vida das baterias e o alcance de transmissão.

Além dos padrões listados na tabela 1.3, há outros, como o HomeRF. Trata-se de uma especificação para redes de voz e dados em residências, também conhecida como *Shared Wireless Access Protocol* (SWAP) (Lansford 2000). A topologia de rede suporta um número ilimitado de nós, podendo operar em modo ad-hoc ou em modo infraestrutura. Muitas de suas características são derivadas do padrão IEEE 802.11. Por se preocupar basicamente com o usuário doméstico, o HomeRF se mostrou um tanto restritivo e por isso não se tornou tão popular, sendo substituído gradativamente por outros, incluindo o ZigBee.

1.2 Redes Metropolitanas de Acesso Aberto (RMAA)

Essa seção apresenta os principais conceitos de Redes Metropolitanas de Acesso Aberto. RMAAs fornecem a base para a criação das Cidades Digitais e com elas a oportunidade de oferecimento de uma enorme gama de serviços aos cidadãos.

1.2.1 Definição

A convergência entre as redes ópticas e sem fio de alta velocidade permite o oferecimento de largura de banda quase que "ilimitada" e proporciona o oferecimento de vários tipos de serviços através de redes de comunicações digitais. No tocante a Redes de Área Metropolitana (MAN), surgem novos modelos para os municípios, as Redes Metropolitanas de Acesso Aberto. Elas fornecem a infraestrutura necessária para a criação das chamadas Cidades Digitais.

Uma Rede Metropolitana de Acesso Aberto, também conhecida como *Open Access* MAN ou Infovia municipal (Mendes 2008), pode ser definida como uma rede multimídia convergente que oferece acesso para toda a população de um município (Mendes, Botolli & Breda 2010). A principal característica de uma RMAA é o seu caráter universalizante, tratando-se de uma rede multisserviço, que possibilita a distribuição de serviços oferecidos separadamente pelos sistemas atuais.

O objetivo da RMAA é a construção de um ambiente de comunicação intramunicipal que leve a modernização dos municípios (Mendes 2008). Uma das grandes vantagens dessa infraestrutura é a convergência e democratização dos meios de comunicação e informação, permitindo o fluxo de dados multimídia, tais como: imagens médicas; videoconferência; educação a distância; bancos de dados educacionais e serviços de Voz sobre IP (VoIP) para a comunicação interna, gerando a universalização da informação.

A construção das RMAA pode estar baseada em diferentes tecnologias de rede, como fibras ópticas, acessos sem fio, cabo coaxial, acessos dedicados e híbrida. Diversos fatores devem ser considerados na escolha de uma delas, incluindo limitação de recursos e as demandas por banda que serão exigidas do sistema. Independente disso, a rede opera sobre o protocolo de comunicação TCP/IP.

Para melhor ilustrar um exemplo de RMAA, a Figura 1.7 exibe a interconexão de diferentes locais em um município.

No exemplo da Figura 1.7, o *backbone* de fibra óptica é responsável por interconectar prédios públicos da cidade, tais como: escolas; prédios do governo e centros de saúde. Os demais ambientes, como empresas e residências dos cidadãos são conectados através de *links* de rádio.

1.2.2 Exemplos de Redes Metropolitanos de Acesso Aberto

A implantação de projetos de RMAA tem sido realizada com sucesso em diferentes cidades pelo mundo (Bauer et al. 2002, Alexiou et al. 2005, Kramer et al. 2006). A tabela 1.4 exibe alguns exemplos de cidades que possuem projetos de RMAA, as tecnologias de rede empregadas e suas contribuições para o município.

Como pode ser observado, duas principais tecnologias de rede são utilizadas na construção de

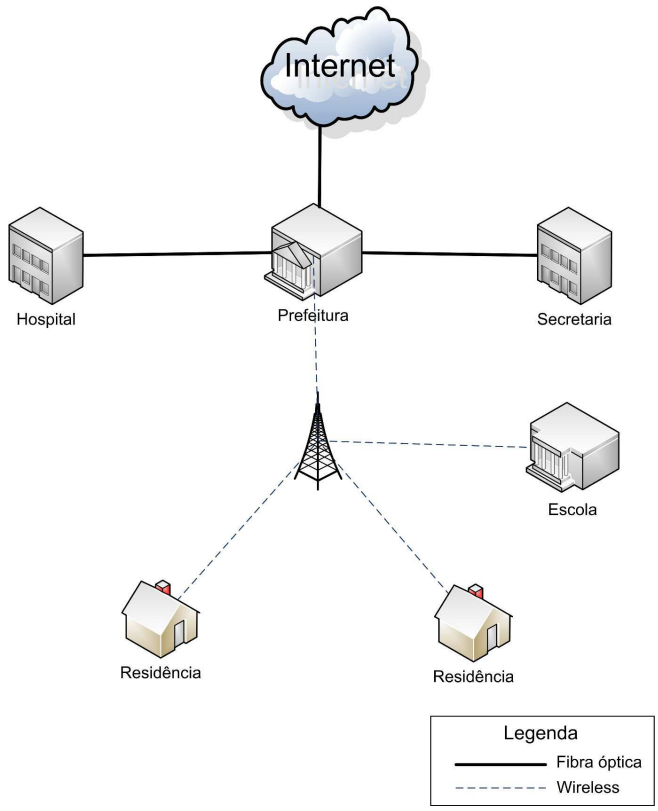


Figura 1.7: Exemplo de interconexão em um município.

RMAA. Por exemplo, a RMAA da cidade de Patras (Alexiou et al. 2005) consiste de rede híbrida, composta de fibra óptica e células de acesso sem fio baseadas em WiMAX (IEEE 802.16) e Wi-Fi, para interconectar três universidades, seis centros de pesquisas, quatro hospitais e 120 escolas. Já a RMAA de Leiden (Kramer, Lopez & Koonen 2006) na Holanda consiste de uma rede em malha comunitária, formada somente por tecnologia *wireless*. Logo, o projeto é conhecido como *Wireless Leiden*. O projeto da *Wireless Leiden* foi iniciado por três moradores da cidade que queriam trocar dados entre si através de conexões IEEE 802.11**b**. A partir daí, novos nós foram adicionados e a rede cresceu, oferecendo conexões internas na cidade e à Internet. Outra característica particular desse projeto é que ele é financiado somente por voluntários, sejam empresas ou cidadãos.

No Brasil, o projeto da cidade de Pedreira é baseado no modelo proposto por Mendes, Botolli & Breda (2010) para Redes Metropolitanas de Acesso Aberto e Cidades Digitais. Esse modelo consiste principalmente na interligação dos órgãos da prefeitura através de fibras ópticas e pontos de acesso sem fio para a distribuição de Internet aos munícipes. O projeto é maduro e foi implantado em outras cidades do interior do estado de São Paulo, como Itatiba, Vinhedo e Guará.

O foco desta dissertação é nesse tipo de RMAA, visto que a arquitetura proposta foi construída considerando sua infraestrutura de comunicação. A interconexão de prédios do setor público em um município, como prefeitura e hospitais, possibilita a criação de diversas soluções de monitoramento e controle para esse setor.

Tabela 1.4: Exemplos de RMAA (Miani 2009).

Cidade	Tecnologia de Rede	Contribuições
Pedreira, Brasil	Fibra óptica e sem fio	Interconexão de prédios públicos, distribuição de Internet para a população e VoIP corporativo
Patras, Grécia	Fibra óptica e sem fio	interconexão de instituições de educação, pesquisa e administração pública.
Leiden, Holanda	Sem fio	Distribuição de vídeo, servidores de jogos e VoIP.
Kutztown, Estados Unidos	Fibra óptica	Fornecimento de conexões de fibra óptica de alta velocidade para residências, escolas, prédios públicos e privados
Granbury, Estados Unidos	Sem fio	Criação de uma rede que une serviços de segurança pública (polícia, bombeiros e serviços de emergência) e da administração municipal.
Cheynne, Estados Unidos	Sem fio	Gerenciamento e controle de tráfego.
Spokane, Estados Unidos	Sem fio	Sistemas governamentais
Shanghai, China	Fibra óptica e sem fio	Sistemas governamentais, informatização dos setores da educação e saúde, implementação de sistema de pagamento eletrônico

1.2.3 Serviços

Como pode ser observado, a concepção de uma RMAA envolve diversos aspectos tecnológicos e consequentemente os custos para tal pode ser considerado elevado para a realidade de algumas prefeituras. É possível, contudo, destacar vários aspectos benéficos que a implantação dessa infraestrutura pode trazer para o município e principalmente para os seus cidadãos, tais como: a infraestrutura composta por fibra óptica permite que a rede transmita dados com maior eficiência e velocidade e seja capaz de suportar serviços em tempo real como VoIP, videoconferência, vídeo sob demanda, *Web TV*, *e-learning*, diagnóstico médico remoto, dentre vários outros (Breda et al. 2011); a transparência na administração pública pode ser melhorada, através da prestação de contas online; maior contato do cidadão com as entidades de governo, e assim por diante.

Disponibilizar o acesso à Internet para toda a população de um município permite que o governo forneça virtualmente muitos serviços que primordialmente eram realizados de maneira presencial. A infraestrutura de uma RMAA oferece condições ideais para serviços que requerem canais de comunicação estáveis e confiáveis entre os participantes.

A seguir são listados alguns serviços que podem ser oferecidos a partir da disponibilização de banda larga (Bauer et al. 2002):

- **Sistema de gestão municipal:**

  Sistemas de gestão municipal, ou *e-Government*, possibilitam a integração de informações de diferentes secretarias/sistemas de um município. Esses sistemas promovem benefícios para setores como: saúde, por meio de sistemas de gerenciamento de consultas e prontuários digitais de atendimento; educação, através de sistemas de administração escolar e bibliotecas virtuais, por exemplo; administração municipal com a utilização de sistemas para controle de frotas, sistemas financeiros e de recursos humanos, dentre outros.

- **Acesso à Internet para a população:**

  Sendo um dos principais canais de comunicação da atualidade, a Internet permite que o cidadão se mantenha atualizado sobre os fatos que ocorrem no seu município e no mundo, além de poder utilizar demais serviços fornecidos por essa plataforma. A Internet também funciona como uma importante fonte para pesquisas, onde as consultas e divulgação de conteúdos são feitas de forma rápida e eficiente. Inserir o cidadão nesse contexto contribui para o desenvolvimento da cidade através da inclusão digital.

- **Telemedicina:**

  Trata-se do provimento de serviços de cuidados médicos, informações ou educação médica, por intermédio de Tecnologias de Informação e Comunicação. Os serviços dessa categoria podem ser em tempo real (síncronos) ou com atraso não previsto para resposta (assíncrono). Serviços de natureza síncrona consistem no exame e diagnóstico de pacientes em tempo real por um canal de comunicação ao vivo, em geral por videoconferência. Já o serviço assíncrono é um pouco mais flexível e não exige análise em tempo real. Alguns serviços médicos que podem ser feitos remotamente incluem: telediagnóstico, telemonitoramento, teleradiologia e teledermatologia.

- **VoIP:**

  O serviço de voz sobre o protocolo da Internet consiste na utilização de redes comutadas trabalhando com o protocolo IP para a transmissão de sinais de voz (Tanenbaum & Wetherall 2011). Trata-se de um serviço de telefonia sobre a Internet. Em uma RMAA o principal benefício do VoIP corresponde a diminuição nos custos com ligações, visto que uma ligação entre dois dispositivos VoIP é gratuita, além da despesa para ligação de um equipamento VoIP para um telefone convencional possuir tarifa menor, se comparado a tarifa cobrada entre dois dispositivos tradicionais. Para um município que possui diversos prédios espalhados geograficamente, essa diferença de valores pode representar uma importante economia.

- ***E-learning*:**

  O *e-learning* corresponde à qualquer tipo de ensino/aprendizagem que é realizado sobre um canal de comunicação digital (Bauer et al. 2002). Podem ser utilizadas ferramentas

computacionais para aprendizagem via Internet, ou aparelhos televisores com conexão com a Internet (*T-learning*). Assim como a telemedicina, o *e-learning* pode ocorrer de forma síncrona, com o professor e o aluno interagindo em tempo real através de canais de áudio e vídeo, ou também de forma assíncrona, com atividades realizadas por meio de portais de gerenciamento de conteúdos, via *e-mail*, ou outra forma de fornecimento de material pelo professor e *feedback* por parte do aluno.

1.2.4 Monitoramento em cenários metropolitanos

Com toda a cidade formando uma rede única de comunicação, o monitoramento de ambientes é uma importante tarefa que se torna factível em virtude da infraestrutura existente e dos avanços na área de microeletromecânica no que se refere a equipamentos sensores.

O monitoramento remoto pode ser utilizado para observar variáveis críticas existentes, tais como:

- *ruas e avenidas*: inspecionar o tráfego é importante para identificar veículos estacionados em locais proibidos, carros trafegando fora do limite de velocidade permitido, identificar colisões e demais acidentes. Para criar sistemas de transporte inteligente através da identificação de congestionamentos, propor rotas alternativas, verificar níveis de poluição causados por gases de escape, com o intuito de fazer um melhor planejamento e propor melhorias principalmente para o transporte público;

- *variáveis ambientais*: mensurar os níveis de chuva em determinadas épocas do ano, bem como a umidade relativa do ar, focos de queimadas. Tudo isso pode ser utilizado para criar situações de alerta ou até mesmo de emergência e assim aplicar ações preventivas e corretivas.

- *serviços básicos*: pode ser utilizado para monitorar variáveis relacionadas aos serviços básicos fornecidos por um município, como os níveis de qualidade da água potável e do ar, monitorar reservatórios que abastecem a cidade, realizar o controle de iluminação pública através do gerenciamento de postes e verificar o seu consumo de energia;

- *acompanhamento de pessoas*: pode ser utilizado para localizar pessoas em um município, por exemplo idosos, através de redes implantadas em postes de iluminação e dispositivos de identificação. A ocorrência de algum acidente pode ser prontamente identificada e, de maneira automática, autoridades competentes podem ser acionadas;

- *gastos públicos*: monitorar o consumo de energia elétrica e água em prédios públicos. Esse monitoramento permite a criação de históricos e torna possível fazer previsões sobre dispêndio e também propor métodos para a redução de gastos.

1.3 Trabalhos relacionados

Conforme exposto na seção 1.1 as redes de sensores sem fio possuem aplicação em diversas áreas do conhecimento. No entanto a sua utilização em cenários de larga escala, como metro-

politano, ainda é pouco difundida principalmente devido as características intrínsecas a essa tecnologia. Nessa seção são expostos alguns trabalhos relacionados a essa dissertação.

Vikatos et al. (2011) apresentam um sistema para monitoramento de gases nocivos no município de Patras, na Grécia. O sistema proposto utiliza nós sensores móveis acoplados à veículos, os quais são responsáveis por realizar a coleta de informações e alimentar uma base central de dados. O sistema idealizado foi batizado de PatrasSense.

Assim como alguns municípios brasileiros, a cidade de Patras também possui projeto de infraestrutura de banda larga para interconectar universidades, centros de pesquisas, hospitais e escolas, conforme descrito em Alexiou et al. (2005). Embora o cenário considerado nos referidos trabalhos seja o mesmo, Vikatos et al. (2011) não fazem qualquer menção a rede metropolitana de Patras em seu trabalho. Os autores expõem a arquitetura do sistema concebido, conforme apresentado na Figura 1.8.

Figura 1.8: Arquitetura do sistema proposto por Vikatos et al. (2011).

De acordo a Figura 1.8, constituem a camada de percepção os dispositivos sensores que podem ser plataformas de prototipagem, tais como Arduino, ou ainda *Smartphones* com sensores incorporados. Acima desses estão os dispositivos responsáveis por transmitir os dados coletados pelos sensores para uma base de armazenamento e compartilhamento. O último componente do sistema é um portal *Web* que fornece a visualização dos dados armazenados pelo sistema.

No sistema, a extração dos dados coletados pelo nível de sensoriamento e posterior transmissão para a camada superior é realizada por mulas (*Mobile Ubiquitous LAN Extension*) de dados. Uma mula de dados é um coletor móvel que consome as leituras dos sensores visitando nó a nó individualmente e os transmite para a camada superior via rede WiFi ou 3G. Dessa maneira, os sensores no sistema proposto devem possuir considerável capacidade de armazenamento para persistir os dados até o momento da coleta por esse intermediário.

Os autores desenvolveram uma aplicação de teste onde sensores foram acoplados a um carro e a uma motocicleta, coletando informações sobre gases de escapamento de veículos no município. Embora o uso de *Smartphones* para monitoramento tenha sido previsto na arquitetura, os autores não fizeram uso desses na aplicação de teste, que foi realizada no período de uma semana.

A proposta dos autores descreve uma importante aplicação de redes de sensores em cenários urbanos. Algumas considerações podem ser feitas, principalmente a respeito do método de interconexão entre os dispositivos sensores e o banco de dados. O uso de mulas de dados se torna inviável à medida que a quantidade de sensores aumenta, inviabilizando a escalabilidade do sistema. O sistema também não prevê dispositivos heterogêneos em nível de sensoriamento.

Pires (2010) descreve uma arquitetura para aplicação de redes de sensores em cenários metropolitanos. A arquitetura proposta é exibida na Figura 1.9.

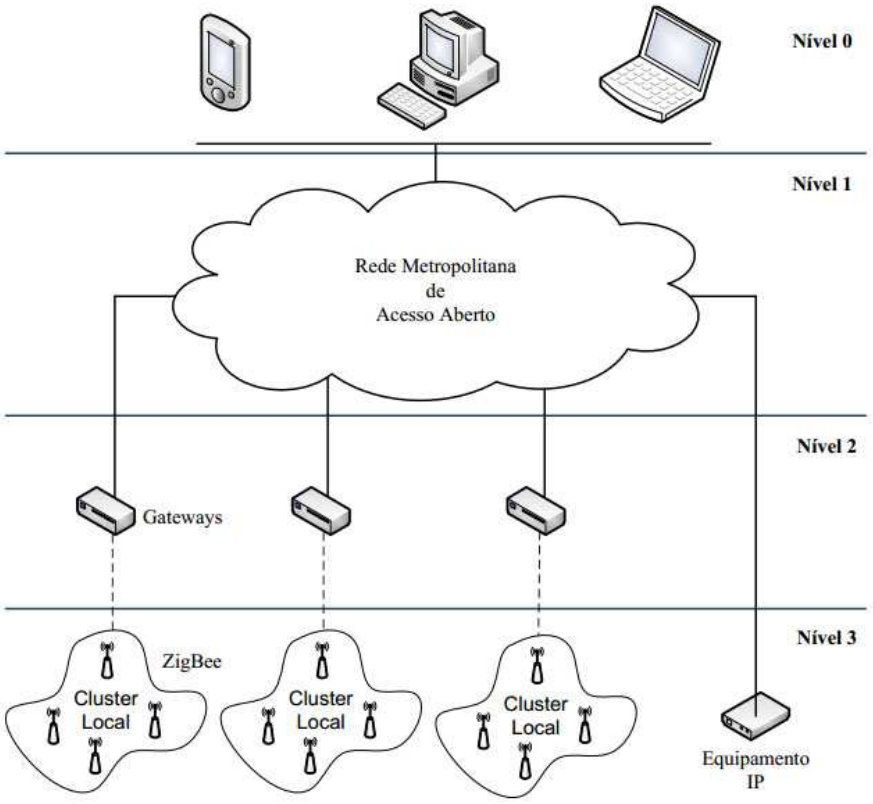


Figura 1.9: Arquitetura híbrida e hierárquica proposta por Pires (2010).

Conforme exposto na Figura 1.9, a proposta é composta de quatro camadas. No nível 0 encontram-se as estações de processamento, que são centrais utilizadas para realizar o monitoramento de redes específicas implantadas em um município. O nível 1 é composto pela infraestrutura de comunicação digital de uma RMAA, responsável por conectar a central de monitoramento no nível 0, a uma rede de sensores no nível 3. Os *gateways* no nível 2 são os responsáveis pela interconexão entre a RMAA e as rede de sensores. No estudo de caso descrito no trabalho, o dispositivo nesse nível se comunicou com a central de monitoramento via *socket* TCP. Por fim, no nível 3 estão as redes de sensores. Pires (2010) considerou principalmente o uso da especificação ZigBee nesse nível.

O trabalho expõe o potencial da utilização de recursos presentes em uma RMAA para a criação de sistemas sensores em nível municipal. Entretanto, a arquitetura proposta apresenta alto acoplamento com os protocolos e tecnologias utilizadas, o que dificulta a expansão do

sistema e o uso de diferentes redes sensoras.

Gniadek et al. (2008) propõem uma infraestrutura para monitoramento de tráfego em ambientes urbanos, baseada em redes ZigBee e *Web services*. A infraestrutura consiste de dispositivos sensores posicionados em rodovias, transmitindo suas leituras para um nó de acesso, que é encarregado de encaminhar as informações para um servidor. Usuários podem acessar o servidor por meio de *Web services* a fim de obter informações sobre as condições de tráfego. O sistema, no entanto, não foi aplicado em um cenário real.

Outro exemplo de aplicação de redes ZigBee em cenários urbanos foi demonstrado por Fan & Guo (2011), no qual os autores apresentam a proposta de um sistema para controle de iluminação pública. Uma arquitetura é apresentada no trabalho. A motivação dos autores está relacionada ao aumento populacional e a expansão das cidades, resultando em maior dispêndio de energia elétrica, agravado pelos problemas existentes em sistemas de controle tradicionais. A proposta consiste de terminais ZigBee instalados nos postes de iluminação, formando redes *mesh*, para controle das lâmpadas. Nós controladores, atuando como *gateways*, são os responsáveis por se comunicar com a rede ZigBee e com o centro de monitoramento e controle de iluminação, através de módulos GPRS. O centro de monitoramento permite o controle remoto das lâmpadas e é responsável também pelo armazenamento dos dados.

Realizar o acompanhamento de pacientes e otimizar o processo de atendimento em situações emergenciais é o objetivo da plataforma *miTag* (*Medical Information Tag*), proposta por Gao et al. (2008). Em situações de acidentes, os pacientes são equipados com nós sensores que formam redes de sensores corporais IEEE 802.15.4. Os nós coletam informações dos pacientes e as enviam para nós denominados repetidores, distribuídos no cenário pelos próprios atendentes, e que formam uma rede em malha capaz de transmitir as leituras dos sensores para uma central local de monitoramento. A falta de infraestruturas de comunicação confiáveis, recursos limitados e informações inadequadas, foram os motivadores do trabalho. Os autores realizaram dois estudos de caso no condado de Montgomery, próximo a cidade de Washington. Em um dos experimentos foi simulada uma situação de acidente automobilístico, onde a abordagem proposta foi aplicada e comparada aos métodos de triagem tradicionais. Resultados mostraram que a plataforma não consumiu mais tempo que a abordagem tradicional e permitiu obter informações mais precisas sobre os pacientes e identificar prioridades de atendimento. O segundo estudo permitiu o monitoramento de pacientes internados em um hospital. O objetivo desse segundo estudo foi demonstrar a capacidade da plataforma de operar em ambientes com elevado nível de interferência eletromagnética. Os resultados desse estudo também foram satisfatórios.

Uma proposta de sistema para monitoramento ambiental e previsão do tempo de curto prazo em cenários urbanos, através do uso de redes ZigBee, é descrita em (Qiu, Xiao & Lv 2012). Os sensores sem fio são responsáveis por coletar dados sobre a qualidade do ar, enquanto que estações meteorológicas coletam informações climáticas em tempo real, em locais específicos de um município, como parques, aeroporto e distrito comercial. Os dados são então enviados para uma central de análise, para que sejam transmitidos à comunidade. Segundo os autores, as informações coletadas podem ser utilizadas para abrandar ou prevenir situações de desastres, promover atividades sociais e melhorar serviços de previsão do tempo e qualidade do ar, através da coleta de dados em tempo real.

Esses são somente alguns exemplos de trabalhos realizados e dos benefícios que a utilização de redes de sensores em cenários urbanos pode trazer para todas as entidades que constituem um município. Entretanto, como se pode constatar na revisão teórica, diferentes abordagens têm sido aplicadas para a concepção desses sistemas. Essa diversidade implica em sistemas heterogêneos em termos de hardware e software e também de informação gerada, resultando em ilhas de sistemas independentes incapazes de interoperar e de serem gerenciados de forma homogênea.

Adicionalmente, verificou-se que a maioria dos sistemas desenvolvidos são de domínios específicos de aplicação e que em geral as informações provenientes do monitoramento em tempo real não são compartilhadas com outras entidades em nível municipal, conforme descrito em Bartlett et al. (2011). Essa característica inibe a cidade de identificar, responder, se adaptar e otimizar de acordo com as condições operacionais.

Com isso, o foco principal no desenvolvimento de aplicações para redes de sensores em cenários metropolitanos deve estar concentrado na utilização de mecanismos para promover a interoperabilidade de sistemas de monitoramento e a integração de dados heterogêneos. Em termos de sistemas distribuídos, o paradigma de Computação Orientada a Serviços (SOC) e sua arquitetura de suporte SOA, são considerados promissores para promover a interoperação entre sistemas com um alto nível de abstração. SOA tem por objetivo melhorar a eficiência, a agilidade e a produtividade, através do uso de serviços para a realização de objetivos da lógica de negócios (Erl 2008). Trata-se de um conceito que prevê o baixo acoplamento entre componentes de software, os quais fornecem funcionalidades na forma de serviços.

É importante notar que SOA consiste de um modelo arquitetural neutro a qualquer plataforma tecnológica. Entretanto, no mercado atual, a tecnologia que está mais fortemente atrelada a sua concepção é a *Web*, por meio da utilização de *Web services* (Erl 2008). *Web services* permitem criar interfaces de comunicação entre diferentes sistemas através da adoção de padrões independentes de plataforma e de linguagem de programação, resultando em ambientes escaláveis, aplicações com baixo acoplamento e capazes de interoperar entre si.

1.4 Considerações sobre o capítulo

Esse capítulo apresentou uma revisão de tecnologias empregadas em redes de sensores sem fio e Redes Metropolitanas de Acesso Aberto. No que tange a redes de sensores, a tecnologia ZigBee teve forte aceitação comercial nos últimos anos. Pesquisas realizadas (Gill et al. 2009) abordam a eficiência da especificação e o esforço de empresas em desenvolver um padrão robusto e confiável. Essas são algumas das justificativas para a utilização dessa tecnologia nesse trabalho, além do baixo custo para aquisição dos rádios transmissores.

Exemplos de soluções alcançadas com a criação de Cidades Digitais também foram apresentadas. Conforme exposto no capítulo, as RMAA beneficiam um extenso número de atores sociais, entre entidades do governo e cidadãos.

Diante das possibilidades apontadas, redes de sensores estão se tornando uma tecnologia ubíqua que permite monitorar extensas regiões geográficas e prontamente detectar eventos particulares. Porém, como pode ser notado na literatura, as redes de sensores, principalmente as

sem fio, são utilizadas mais comumente em aplicações de escalas reduzidas para monitoramento de objetos de interesse. Isso se deve principalmente ao poder computacional empregado pelos dispositivos sensores, além do curto alcance de transmissão dos rádios, características inerentes devido ao limitado recurso energético fornecido pelos dispositivos.

Nota-se também o crescimento de iniciativas para a aplicação de redes de sensores em cenários urbanos. Entretanto, fica claro a rigidez das soluções propostas e a natureza restritiva destas, focadas em domínios específicos de aplicação. A falta de uma infraestrutura de comunicação adequada para cobrir uma grande área, como municipal, também inibe a centralização de informações.

Ainda assim, constata-se a necessidade de criação de uma arquitetura baseada em padrões genéricos e independentes de plataforma para o monitoramento e gerenciamento eficiente das redes de sensores aplicadas em cenários urbanos. Sendo assim, a arquitetura que se propõe considera a infraestrutura de RMAAs como parte formadora da camada de interconexão entre as redes existentes no cenário monitorado. Uma RMAA exibe características que permitem a transmissão de informações de maneira eficiente e confiável.

Com o intuito de criar uma arquitetura independente de plataforma e protocolos de comunicação em nível de sensores, a arquitetura proposta está baseada em *Web services*, conferindo a ela o caráter de Arquitetura Orientada a Serviços (SOA). O capítulo 2 descreve esse conceito, bem como o de *Web services*.

Capítulo 2

Arquitetura Orientada a Serviços (SOA)

A arquitetura proposta nessa dissertação utiliza *Web services* para compartilhamento de dados e para estabelecimento da comunicação entre a RMAA e as redes de sensores implantadas no município.

Esse capítulo apresenta uma visão geral sobre os principais conceitos relacionados a esse paradigma que evoluiu consideravelmente nos últimos anos. Uma das principais tecnologias que impulsionaram esse crescimento foi a de *Web services*.

2.1 SOA - Visão geral

Em termos de sistemas distribuídos a Computação Orientada a Serviços (*Service Oriented Computing* (SOC)) que compreende a Arquitetura Orientada a Serviços, representa um dos paradigmas mais promissores da atualidade. Este tipo de arquitetura procura estabelecer soluções independentes de plataforma, principalmente através da substituição de padrões proprietários por serviços ubíquos, abstratos, bem definidos e interoperáveis. Isso promove o baixo acoplamento entre elementos básicos de software através de interfaces comuns de acesso e protocolos de comunicação padronizados.

De acordo com Erl (2008), SOA estabelece um modelo arquitetural que objetiva melhorar a eficiência, a agilidade e a produtividade, posicionando serviços como meios primários, através dos quais uma solução lógica é representada para suportar a realização de objetivos associados com a Computação Orientada a Serviços. Como o próprio nome sugere, a pedra angular de uma solução orientada a serviços é o serviço. Serviços existem como softwares fisicamente independentes e com características de design distintos, onde cada serviço possui o seu contexto funcional e é formado por um conjunto de recursos relacionados a esse contexto.

Embora a Arquitetura Orientada a Serviços possa ser vista como um paradigma único, ela pode ser concebida de diversas maneiras. Em muitos aspectos SOA pode ser vista como uma arquitetura do tipo cliente-servidor aplicada em um novo contexto (Ma 2005). A Figura 2.1 ilustra esse comportamento.

De maneira simplificada, um requerente de serviço (cliente) comunica o provedor (servidor) sobre um serviço desejado, um recurso ou uma ação a ser realizada, através de uma mensagem, e recebe uma resposta em forma de informação, ou então, de confirmação da execução de uma

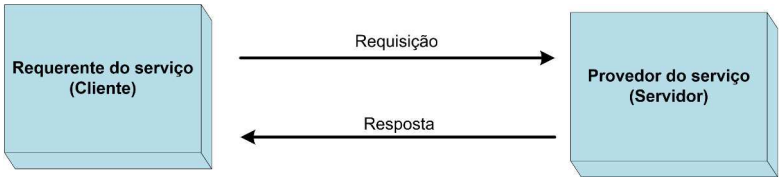


Figura 2.1: SOA como um arquitetura do tipo cliente-servidor.

ação. É importante ter em mente que SOA é um modelo de arquitetura neutro a qualquer plataforma tecnológica. Entretanto, no contexto atual, a plataforma que está mais associada com a concepção de arquiteturas orientadas a serviços é a *Web services* (Erl 2008). *Web services* evoluíram ao ponto de oferecer uma tecnologia rica e flexível o suficiente para tornar SOA uma realidade (Ma 2005).

2.2 *Web Services*

Segundo o *World Wide Web Consortium* (W3C 2004), um *Web service* é um mecanismo de software desenvolvido para suportar as interações máquina-máquina em uma rede de computadores. Um serviço é um recurso dotado de uma semântica específica e caracterizado pelo conjunto abstrato da funcionalidade que é fornecida, a qual é coerente do ponto de vista de provedores e clientes.

Conforme ilustrado na Figura 2.1, *Web services* são compostos por requerentes e provedores de serviços. Uma entidade provedora é uma pessoa ou organização que provê funcionalidades através de um ou mais serviços específicos. Em contrapartida, um requerente é uma pessoa ou uma organização que deseja fazer uso dos *Web services* fornecidos pelo provedor.

Um serviço é uma noção abstrata que precisa ser implementada por agentes concretos. Dessa forma, as entidades provedoras e requerentes implementam agentes, que consistem de unidades de software que enviam e recebem mensagens, tipicamente por meio do protocolo *HyperText Transfer Protocol* (HTTP).

Para ilustrar a diferença entre serviço e agentes, um *Web service* (funcionalidade) pode ser implementado utilizando uma linguagem de programação (agente) específica, como Java. Essa mesma funcionalidade (serviço) pode ser implementada por outro agente provedor, ou seja, uma outra linguagem de programação, como Python. Nesse caso, embora os agentes provedores sejam diferentes, o serviço concebido permanece o mesmo.

Para que um requerente possa utilizar *Web services* é necessário que: (1) ele conheça o provedor do serviço; (2) as duas entidades acordem sobre a descrição do serviço e a semântica que irá governar as interações entre elas; (3) os agentes, que atuam em nome do requerente e do provedor, troquem mensagens em nome das suas entidades (W3C 2004). O passo 1 está relacionado a descoberta de um serviço. Em geral, o processo de descoberta é iniciado pela entidade requerente, que pesquisa em registros ou índices os serviços que atendam as funcionalidades ou outros critérios desejados. Quando um serviço é encontrado, no passo 2, a sua interface é disponibilizada ao requerente. A interface de um *Web service* é o contrato entre os agentes requerente e provedor. Ela é a responsável por definir as mensagens, os tipos de dados

e o mecanismo de transporte para comunicação entre os agentes. Finalmente, no passo 3, os agentes são conscientes um do outro e acordaram sobre a semântica e descrição de um serviço. Portanto, podem realizar a comunicação de acordo com a interface do serviço.

A arquitetura de *Web services* não define meios para a implementação de unidades de software. Somente é necessário que as abordagens utilizadas não prejudiquem a interoperabilidade e mantenham íntegros os conceitos e relacionamentos definidos pela arquitetura.

Desse modo, é possível destarcar as principais vantagens da utilização de *Web services*: baixo acoplamento, *Web services* concentram-se nas interfaces de acesso, não importando-se com detalhes de implementação, o que aumenta também a produtividade no desenvolvimento; alta granularidade, pois o baixo acoplamento permite criar soluções de software independentes para prover funcionalidades significativas; ubiquidade, serviços disponíveis em qualquer lugar e a qualquer momento; e interoperabilidade, pois faz uso de tecnologias padronizadas, neutras e compatíveis com padrões amplamente aceitos, funcionando em uma ampla variedade de plataformas/*frameworks*.

A característica chave de *Web services* é a sua independência de plataforma. Para que isso seja conseguido, padrões abertos e especificações tecnológicas foram criados. Embora existam muitos meios de conceber e usar *Web services*, as demais seções descrevem de maneira superficial os principais padrões e tecnologias utilizadas. Derivadas das tecnologias descritas, diversas outras especificações associadas a *Web services* foram propostas. Essas especificações englobam mecanismos de descoberta, endereçamento e segurança, por exemplo. O prefixo *WS-* é usado para indicar tais especificações, como por exemplo *WS-Addressing*, *WS-Security* e *WS-Discovery.* Esses, no entanto, estão fora do escopo deste trabalho.

2.2.1 *eXtensible Markup Language* (XML)

O W3C declara o XML como parte da solução para a construção de *Web services*, particularmente para a troca de mensagens entre agentes. XML é um formato de texto simples e flexível utilizado para representar informações de modo estruturado (W3C 2003). Desenvolvido inicialmente para atender aos desafios da publicação eletrônica em larga escala, o XML desempenha hoje papel fundamental para troca e armazenamento de informações em uma ampla variedade de aplicações, principalmente de natureza *Web*.

De modo similar ao HTML, o XML também é uma linguagem de marcação que utiliza *tags* para estruturar dados de forma hierárquica. A diferença é que não há *tags* predefinidas em XML. Sendo assim, a responsabilidade pela definição dessas *tags* fica a cargo do criador do documento, o qual deve utilizar nomenclatura que seja significativa para quem irá utilizar o documento. Essa definição é, em geral, feita por meio da utilização de esquemas XML (*XML Schema*) para criar formatos em termos de estrutura e tipos de dados. Assim, XML foca na sintaxe para a representação de dados, ao invés da semântica de documentos.

Essa característica é notável, visto que o XML não se utiliza de protocolos proprietários para a codificação e decodificação de mensagens, uma vez que sua formatação é baseada em texto puro, passível de entendimento por ambos, humanos e máquinas. Além dessa característica, o XML é suportado por diversas linguagens de programação que são capazes de analisar documentos e extrair o conteúdo existente, bem como a sua estrutura.

2.2.2 *Web Services Description Language* (WSDL)

Para que um serviço *Web* seja consumido é preciso que o cliente saiba onde encontrá-lo e também como interagir com ele. O W3C define um modelo em formato XML para a descrição de *Web services*, o *Web Services Description Language* (WSDL) (Christensen et al. 2001). O WSDL fornece uma gramática XML para descrever informações pertinentes a um *Web service*, em conformidade com o *schema* XML criado para WSDL. Isso inclui um nome, mecanismo de transporte e formato de mensagens. Assim, essencialmente o WSDL descreve três propriedades fundamentais de um *Web service*: o que ele faz (suas operações); como o serviço é acessado; e onde ele está localizado. Dessa maneira, agentes requerentes o utilizam para entender a interface particular de um serviço específico.

Uma interface de serviço pode conter qualquer número de operações (Graham et al. 2002). Apesar de descrever a estrutura dos dados envolvidos em uma transação *Web service*, o WSDL não define o método que a máquina utiliza para analisar e interpretar as entradas do usuário (Ma 2005).

2.2.3 *Universal Description, Discovery and Integration* (UDDI)

O UDDI é um serviço de diretório onde *Web services* podem ser registrados e pesquisados. Pode ser um registro de serviços centralmente controlado, ou então um guia, baseado em índices, para serviços distribuídos. Se assemelha a uma lista telefônica. Ele mantém uma lista com os serviços disponíveis na rede, onde os provedores podem registrar os *Web services* em uma base compartilhada para que as entidades requerentes possam pesquisá-los, descobrir suas informações, como utilizar e como acessá-los. A imagem exposta na Figura 2.2 ilustra o comportamento do UDDI entre um cliente e um provedor de serviços.

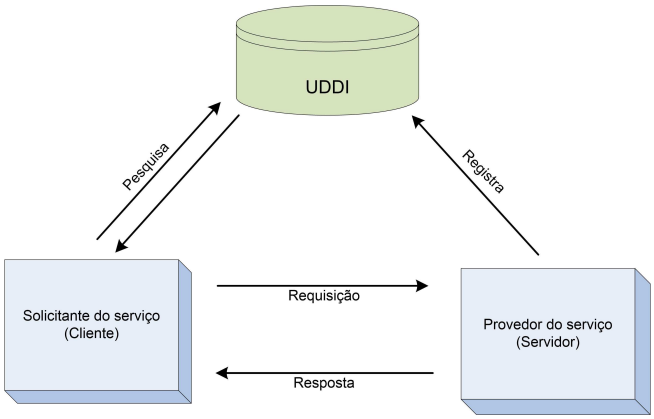


Figura 2.2: Arquitetura *Web service* com UDDI.

Um serviço de descoberta requer a habilidade de consultar *Web services* apropriados, baseado em descrições funcionais ou outros critérios. Nesse sentido, o UDDI é o responsável por encontrar serviços registrados que satisfaçam as informações fornecidas. Caso um serviço seja encontrado, a sua interface de descrição de serviço, em geral o arquivo WSDL, é retornado para o agente requerente que enviou a requisição. A troca de mensagens entre o UDDI e clientes em geral é feita utilizando o protocolo SOAP.

O processo de divulgação e descoberta de serviços pode acontecer de forma automática ou manual. Na forma automática, agentes, atuando em nome do provedor ou do requerente, pesquisam serviços que atendam suas necessidades. Na forma manual, as próprias entidades são as responsáveis por realizar esse processo.

2.2.4 *Simple Object Access Protocol* (SOAP)

Um *Web Service* necessita de um mecanismo padronizado para a troca de dados entre clientes e provedores de serviços. A recomendação do W3C para tal é o SOAP. SOAP é um protocolo de camada de aplicação para a troca de mensagens em um ambiente distribuído e descentralizado (Box et al. 2000). As informações transmitidas entre cliente e servidor são empacotadas em mensagens SOAP baseadas em XML, também conhecidas como *envelopes*. Um envelope é composto por um cabeçalho (*header*) opcional e um corpo (*body*). O corpo representa carga útil do envelope, ou seja, a informação que está sendo transmitida. O protocolo SOAP não especifica a sintaxe nem a codificação do conteúdo localizado no corpo do envelope, ficando a cargo das partes envolvidas na transação concordarem sobre a estrutura dos documentos trocados entre elas. Ambos, o cabeçalho e o corpo, podem conter múltiplas sub-partes na forma de blocos, isto é, blocos de cabeçalho e blocos de corpo. A Figura 2.3 ilustra a estrutura de um envelope SOAP.

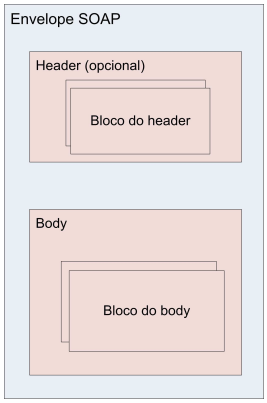


Figura 2.3: Estrutura de um envelope SOAP.

Algumas das especificações do SOAP incluem um conjunto de regras que qualquer entidade que o processe precisa seguir e uma descrição de como as mensagens devem ser transportadas. Essa ideia segue a abordagem de protocolos de comunicação padronizados, onde as informações de controle estão localizadas no cabeçalho do pacote. No entanto, a utilização do cabeçalho só é realmente útil quando há nós intermediários entre o remetente e o receptor final da informação, o que é comum em arquiteturas multicamadas.

As primeiras versões do SOAP foram baseadas somente em HTTP. No entanto a versão atual (1.2) não impõe qualquer protocolo de transporte para a transferência de envelopes. Por exemplo, as mensagens podem ser transportadas também via protocolo SMTP de forma assíncrona. A especificação de qual utilizar é chamada de associação/ligação (*binding*) (Alonso, Casati & Kuno 2004).

O protocolo SOAP pode ser empregado para troca de informações em rede (modelo de documento) e também para a chamada de procedimentos remotos (*Remote Procedure Call* (RPC)) (modelo RPC).

Segundo Ma (2005), uma das desvantagens do SOAP é o *overhead* causado por ele. Isso porque a comunicação entre cliente e servidor é feita por mensagens XML, as quais podem ser extensas e que precisam ainda ser envoltas por pacotes SOAP, aumentando ainda mais o tamanho do pacote.

2.2.5 *Representational State Transfer* (REST)

É comum encontrar referências que descrevem a arquitetura para a criação de *Web services* somente baseada nos três componentes anteriormente descritos: SOAP, para a troca de mensagens; WSDL, para a descrição de serviços; e UDDI, como diretório de nomes e serviços.

Entretanto, o estilo de arquitetura REST tem ganhado ampla aceitação como uma alternativa simplificada em comparação a *Web services* baseados em interfaces WSDL e SOAP (Zou, Mei & Wang 2010). Essa substituição ocorre principalmente devido a sua simplicidade e escalabilidade.

A arquitetura REST foi proposta na tese de doutorado de Fielding (2000), um dos principais autores das especificações do protocolo HTTP e do *Uniform Resource Identifier* (URI).

O estilo arquitetural REST define um conjunto de restrições que, quando aplicadas como um todo, enfatizam a escalabilidade de interações entre componentes, interfaces genéricas e independência entre eles. As restrições especificadas são:

- *Cliente-Servidor*: segue o princípio da separação de interesses. Separar a interface do usuário do armazenamento de dados permite melhor portabilidade de interface entre múltiplas aplicações e beneficia a escalabilidade, simplificando os componentes do servidor;

- *Independência de estado*: corresponde a interação cliente-servidor. A comunicação precisa ser livre de estado, tal que cada solicitação do cliente para o servidor deve conter todas as informações necessárias para que o servidor a entenda e seja capaz de processá-la. Escalabilidade e confiabilidade são melhoradas com as propriedades dessa restrição;

- *Cache de dados*: objetiva melhorar a eficiência da rede. Respostas de um servidor precisam ser rotuladas como armazenáveis (*cacheable*) ou não. Caso uma resposta seja armazenável, o cliente tem o direito de reutilizar os seus dados em requisições equivalentes futuras;

- *Interface uniforme*: é a característica central que distingue REST de outros estilos de arquitetura em rede. As implementações são independentes dos serviços providos por elas, encorajando assim a evolução independente. Uma desvantagem dessa interface uniforme é em termos de eficiência, uma vez que a informação é transmitida de maneira padronizada, em detrimento a meios e formatos específicos que satisfaçam as necessidades da aplicação;

- *Sistemas em camadas*: permite que a arquitetura seja composta de camadas hierárquicas, tal que cada componente não possua conhecimento além da camada imediata com a qual

ele interage. Isso restringe o conhecimento a uma única camada, limitando a complexidade global do sistema;

- *Código sob demanda*: trata-se de uma restrição opcional do REST. Ela permite que a funcionalidade do cliente seja estendida, por meio de *download* e execução de códigos, tais como *applets* Java ou códigos *JavaScript*. Essa característica melhora a extensibilidade do sistema.

As restrições mencionadas foram empregadas para guiar o design e implementação da arquitetura para a *Web* (Fielding 2000). REST descreve como objetos de dados distribuídos, ou recursos, podem ser definidos e abordados. De modo geral, trata-se de uma arquitetura orientada a recursos fundamentada em padrões *Web*. Recursos possuem identificadores únicos globais, como URLs, para endereçamento *Web*.

O termo REST descreve qualquer interface simples para a manipulação de dados (recursos) de domínio específico sobre HTTP, sem a necessidade de camadas de mensagens adicionais. *Web services* que implementam o conceito de REST são conhecidos como RESTful (Rodriguez 2008).

2.2.5.1 *Web services* RESTful

REST não define nenhuma linguagem ou protocolo e se utiliza basicamente de URIs e dos métodos (verbos) fornecidos pelo protocolo HTTP, sendo os mais utilizados:

- *GET*: para a consulta de um recurso específico ou de um grupo de recursos;
- *HEAD*: para retornar informações de estado de um recurso. É similar ao GET, exceto que o servidor não retorna o recurso especificado. Pode ser utilizado para verificar se um recurso existe, se foi alterado, e assim por diante;
- *POST*: para a criação de um novo recurso através do envio de dados para o servidor, enviar parâmetros utilizados em consultas de bancos de dados;
- *PUT*: para atualizar um recurso especificado. Pode também ser usado para criar um recurso específico;
- *DELETE*: para remover um recurso especificado por um identificador;

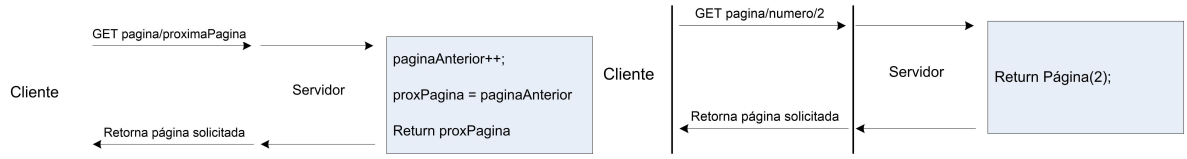
Dessa maneira, cada recurso disponibilizado pelo servidor é identificado através de uma URI única. Esses recursos fornecem interface uniforme de acesso aos métodos e as suas representações.

Embora o REST não seja padronizado, *Web services* baseados nesse modelo de arquitetura utilizam especificações bem definidas. Além disso, um dos diferenciais de *Web services* RESTful é a utilização do formato de dados *JavaScript Object Notation* (JSON) para representação de recursos, uma alternativa ao XML proposto em 1999, útil para representar objetos JavaScript como sequências de caracteres (*strings*) e transmiti-los através da rede. É um formato simples

de se produzir e consumir, além de ser mais rápido para analisar (*parse*) do que o XML (Deitel & Deitel 2008).

REST apresentou forte destaque para a criação de *Web services* nos últimos anos e passou a ser adotado por grandes empresas como Google e Yahoo (Rodriguez 2008), as quais permitem que sistemas externos utilizem os seus serviços através de interfaces de programação de aplicações (APIs) bem definidas.

Uma das principais características das transações baseadas em REST é a sua independência de estado. Isso significa que o servidor não mantém informações de sessão sobre uma requisição realizada previamente. Dessa forma, toda solicitação é tratada como uma nova transação independente. A Figura 2.4 ilustra a diferença entre sincronização e independência de estado.



(a) Dependência de estado (b) Independência de estado

Figura 2.4: Exemplos de requisição com e sem dependência de estado (adaptada de (Rodriguez 2008)).

Em uma requisição independente de estado, cada solicitação do cliente para o servidor precisa conter todas as informações necessárias para que ele a compreenda (Fielding 2000). O controle de sessão é mantido totalmente no cliente. Essa característica é importante quando são utilizados *clusters* de servidores e há alguma espécie de balanceamento de carga para distribuição de tráfego. Com essa independência de estado, as requisições podem ser encaminhadas para qualquer instância do servidor que esteja menos sobrecarregada.

2.2.6 XML-RPC

As abordagens apresentadas anteriormente nesse capítulo são principalmente orientadas à mensagens. Isto é, provedores e clientes de serviços comunicam-se por canais de duas vias (*request-response*), por meio de troca de mensagens. *Web services* podem também ser baseados em chamadas de procedimentos remotos (RPC) (Graham et al. 2002). Refere-se à chamada de operações (métodos) de um cliente para um servidor RPC. Essa chamada é similar a invocação de uma função em um sistema local. Para que ela seja realizada são necessários: um alvo (provedor), ou seja, a máquina que hospeda a operação a ser consumida como serviço; o nome da operação, da mesma forma como implementado no servidor RPC; e um conjunto opcional de parâmetros. A localização do alvo é quase sempre uma URI em forma de URL.

O XML-RPC trata-se de um protocolo para chamada de procedimentos remotos que utiliza o protocolo HTTP como mecanismo de transporte (Winer 1999). Ele permite que sistemas que funcionam em diferentes plataformas e ambientes possam trocar informações por meio de operações e estruturas de dados complexas formatadas em XML (Dissanaike, Wijkman & Wijkman 2004). Mensagens XML-RPC são enviadas via o método POST do HTTP, onde o corpo da mensagem de requisição é um documento XML contendo o nome da operação desejada e

pode conter um conjunto de parâmetros associados. O servidor XML-RPC recebe a requisição, executa o método desejado e pode ou não fornecer uma resposta à solicitação realizada, dependendo do modelo de comunicação implementado. Caso uma resposta seja fornecida, essa é também codificada em XML. Isso vale ainda para os casos onde a operação não tenha sido realizada com sucesso, seja por motivo de método inexistente, formato de parâmetro incorreto, e assim por diante. As principais linguagens de programação, como JAVA, C#(*C-sharp*), PHP e Python, possuem bibliotecas que facilitam a criação de clientes e servidores XML-RPC.

O XML-RPC foi desenvolvido para ser tão simples quanto possível. De acordo com Winer (1999), os tipos de parâmetros suportados por ele são: escalares, números (inteiros, reais), booleanos, *strings*, datas, binários codificados em base64, e ainda tipos complexos, como estruturas e *arrays*. Todas as requisições a procedimentos remotos são enviados a uma mesma URL no servidor XML-RPC.

O uso de XML-RPC evita a necessidade de criação de protocolos e interfaces específicas para que sistemas desenvolvidos para plataformas e em linguagens de programação diferentes, possam estabelecer comunicação entre si. É um formato de troca de mensagens simples, flexível e de fácil implementação.

2.3 Considerações sobre o capítulo

Esse capítulo discorreu brevemente sobre o conceito de Arquitetura Orientada a Serviços e *Web services*, apresentando as principais tecnologias empregadas nas suas concepções.

Como pode ser observado, a utilização de *Web services* resulta em aplicações independentes de plataforma e protocolos de comunicação, culminando em ambientes interoperáveis e altamente escaláveis. Essas são características importantes no que diz respeito ao monitoramento de redes de sensores em cenários metropolitanos. Grande heterogeneidade de dispositivos e tecnologias podem coexistir nesses ambientes, o que resulta em sistemas isolados e difíceis de gerenciar.

Assim, *Web services* são parte fundamental da arquitetura que propomos nesse trabalho, tanto em termos de disponibilização dos dados coletados pelos sensores, quanto para a interligação da aplicação de monitoramento com os sensores a nível de rede.

Consoante as características descritas, considera-se que a arquitetura REST apresenta maior flexibilidade e simplicidade de implementação. Trabalhos existentes na literatura comparam o desempenho de implementações de *Web services* baseados em SOAP e REST, como em (Aihkisalo & Paaso 2012, Mulligan & Gracanin 2009), que comprovam a superior eficiência de interfaces REST em termos de consumo de banda e latência de transmissão. Na abordagem utilizando SOAP, é preciso que ambos, cliente e servidor, possuam suporte a esse protocolo, isto é, bibliotecas cliente SOAP e servidor SOAP precisam ser utilizadas. Em implementações REST, qualquer cliente HTTP pode interagir com um servidor HTTP, somente através do nome do *host* ou endereço de rede.

Dessa forma, esse conceito foi aplicado para prover interfaces de acesso a sistemas externos para obter os dados das redes de sensores implantadas em um município, para fins de processamento e análise.

Para a concepção do nível de comunicação da arquitetura considera-se o uso de *gateways*

provedores de *Web services*, implementando operações para que clientes possam invocar procedimentos remotos (RPC) através de mensagens XML, para obter informações sobre as redes de sensores monitoradas. Embora o protocolo SOAP também possa ser utilizado para chamada de procedimentos remotos, ele apresenta uma maior complexidade e menor flexibilidade se comparado ao XML-RPC.

Capítulo 3

Arquitetura proposta

As Redes Metropolitanas de Acesso Aberto fornecem recursos para a integração de redes de sensores implantadas em uma cidade, possibilitando a criação de sistemas de controle e monitoramento em escala municipal. A aplicação de redes de sensoriamento nas cidades também agrega valor as RMAAs, pois acrescentam uma importante funcionalidade ao seu caráter multisserviço.

Esse capítulo detalha a arquitetura proposta. Ela consiste na utilização de padrões *Web*, como o protocolo HTTP e o XML, para a criação de *Web services*, onde os serviços são utilizados tanto em camada de aplicação quanto de comunicação. Dessa forma, interfaces genéricas e independentes de plataforma são concebidas, o que possibilita a realização do monitoramento de redes heterogêneas implantadas em um município. Outro fator importante que é analisado nesse capítulo são métodos para promover a segurança da informação em termos da arquitetura.

3.1 Camadas que compõem a Arquitetura

A aplicação de redes de sensores em escala urbana é complexa e precisa ser cuidadosamente analisada. Um cenário metropolitano apresenta grande diversidade de ambientes e pontos de monitoramento. Essa diversidade resulta também em uma gama variada de informações coletadas. Todas essas características precisam ser avaliadas para a implantação de redes de sensores em cenários metropolitanos.

Ainda assim, redes de sensores implantadas em cenários urbanos são aplicadas em domínios específicos, e dessa forma, os sistemas desenvolvidos apresentam alto acoplamento com as tecnologias empregadas. Esse acoplamento afeta a escalabilidade e a interoperabilidade com outros sistemas, tendo em vista a aplicação de padrões rígidos e proprietários.

A arquitetura que está sendo proposta tem o objetivo de abstrair a heterogeneidade de dispositivos e tecnologias utilizadas em redes sensoras implantadas em município, com o intuito de conceber um ambiente único e genérico para a realização de tarefas de monitoramento. A arquitetura promove o compartilhamento de informações coletadas por aplicações de domínios distintos, com o intuito de auxiliar gestores e sistemas inteligentes no processo de tomada de decisões. Ela se baseia no conceito de orientação a serviços, o que permite o desacoplamento de componentes e a definição de interfaces padronizadas, independentes de plataforma e de linguagem de programação.

Considerando o potencial apresentando por redes de sensores e RMAA, conforme exposto no capítulo 1, e a natureza independente e flexível de Arquiteturas Orientadas a Serviços, exposta no capítulo 2, está sendo proposta uma arquitetura composta por três camadas, sendo estas: *Aplicação*, *Comunicação* e *Sensoriamento*.

A Figura 3.1 expõe as camadas da arquitetura.

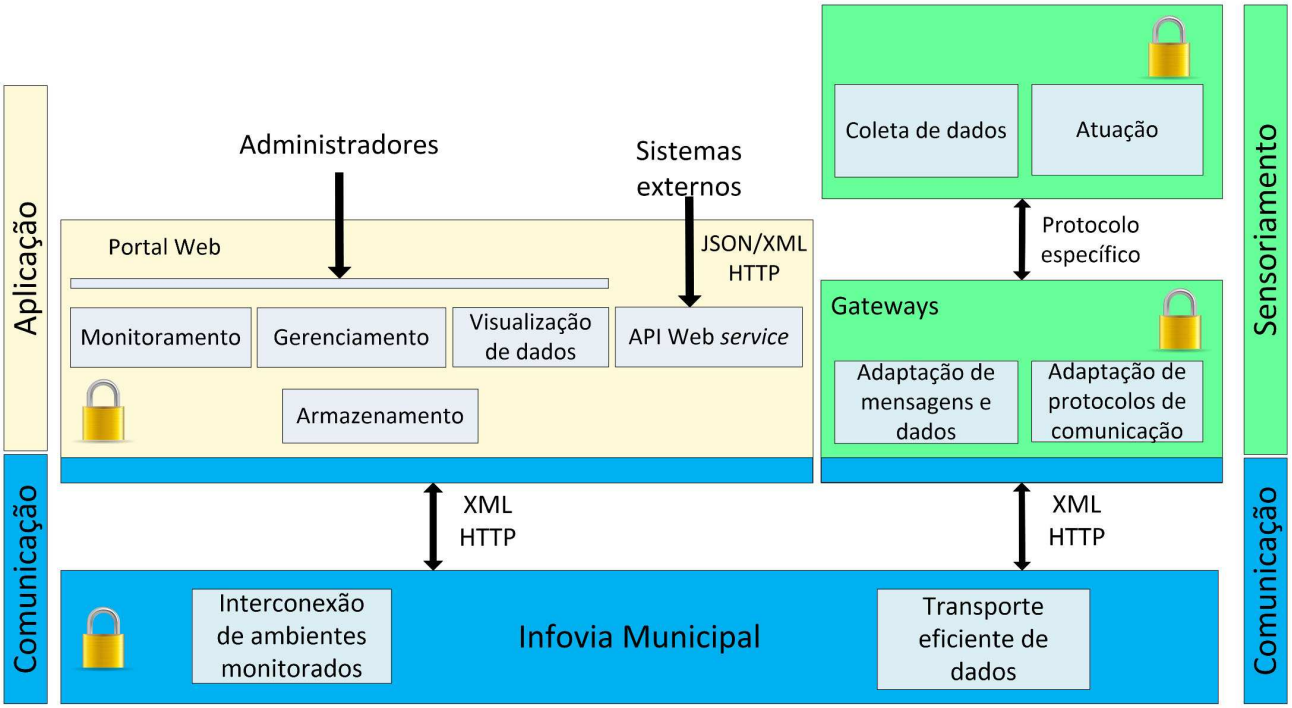


Figura 3.1: Arquitetura proposta.

A Camada de Aplicação, que será descrita na seção 3.1.1, refere-se aos servidores *Web* e de banco de dados. Ela fornece interfaces amigáveis para que os administradores possam monitorar as redes de sensores existentes no município. Todas as informações de usuários, dos ambientes monitorados e das redes de sensores são armazenadas nessa camada da arquitetura em uma base de dados única. Essa camada disponibiliza ainda APIs *Web services* para que sistemas externos possam obter os dados coletados pelas redes sensoras monitoradas.

A Camada de Comunicação, descrita na seção 3.1.2, compreende a infraestrutura (rede de dados) da RMAA, a qual é a responsável por interligar os ambientes monitorados no cenário municipal. Dispositivos do tipo *gateway*, encarregados de tratar as interações entre a camada de aplicação e as redes de sensores, conectam-se diretamente a infraestrutura da RMAA. Os *gateways* atuam como *Web servers*, fornecendo serviços de descoberta, configuração e coleta de dados das redes de sensores subjacentes.

A Camada de Sensoriamento é descrita na seção 3.1.3. Ela é formada por dispositivos sensores responsáveis pela percepção de fenômenos de interesse nos ambientes monitorados. Grande diversidade de dispositivos e padrões de comunicação podem coexistir nessa camada da arquitetura. Os *gateways* também estão presentes nessa camada, estabelecendo comunicação com as redes de sensores por meio de protocolo de comunicação específico.

Para identificar os elementos que constituem cada camada da arquitetura e também os protocolos empregados na comunicação entre elas, deve ser considerada a imagem exibida na Figura 3.2.

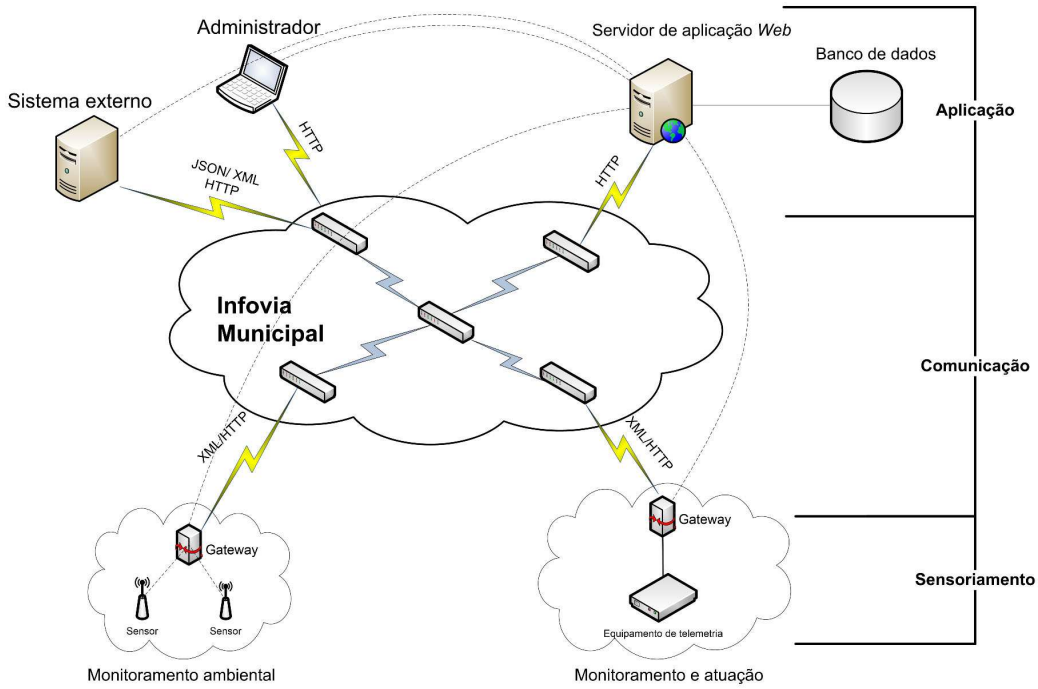


Figura 3.2: Elementos e protocolos de comunicação que compõem a arquitetura.

A RMAA funciona como um *backbone* para interconectar os demais componentes da arquitetura. As linhas pontilhadas na Figura 3.2 representam a comunicação lógica entre os elementos da arquitetura. Como pode ser observado, os administradores monitoram as redes implantadas em um município por meio do servidor *Web* da camada de aplicação, o qual é o responsável por utilizar os serviços fornecidos pelos *gateways* através de procedimentos remotos baseados em XML para manipulação das redes sensoras.

As camadas apresentadas são descritas detalhadamente nas demais seções.

3.1.1 Camada de Aplicação

Conforme exposto na Figura 3.2, na camada de aplicação encontram-se os servidores *Web* e de banco de dados. Os administradores das redes implantadas no cenário monitorado interagem com elas a partir desse nível da arquitetura. Ela funciona como um centralizador de informações, armazenando dados sobre as redes de sensores de diferentes domínios de aplicação em um município. Essa abordagem permite criar uma visão unificada do que acontece na cidade, integrada em uma base central. De um modo geral três componentes formam a camada de aplicação, são eles: um portal *Web*, um banco de dados central e *Web services*.

O portal *Web* existente nessa camada é utilizado pelos administradores para monitorar as redes de sensores. Isto é, após a implantação das redes nos ambientes de interesse, é possível monitorar e configurá-las a partir dessa interface. Entretanto, o portal não interage diretamente

com as redes de sensores. Ele fornece uma interface de alto nível de abstração para a comunicação com os *gateways*, que são os responsáveis por aplicar as operações solicitadas em nível de aplicação. É de responsabilidade dos *gateways* também descrever os serviços disponíveis pelas redes de sensores correspondentes. Dessa forma, o portal *Web* representa uma interface genérica de acesso aos serviços providos por redes de sensores heterogêneas.

O gerenciamento dos ambientes e dos respectivos usuários administradores também é realizado através do portal *Web*. Todas as informações relacionadas ao processo de monitoramento e gerenciamento, incluindo dados de usuários e localidades, configurações das redes de sensores, bem como os dados coletados por essas, são armazenadas em uma base central de conhecimento. Dessa maneira, a segurança da informação é uma questão de extrema importância e que deve apresentar requisitos de confidencialidade, integridade e disponibilidade. Aspectos de segurança na arquitetura proposta são examinados na seção 3.3.

Para promover o compartilhamento de informações, sistemas externos são capazes de obter os dados coletados em tempo real, através de interface *Web service* em nível de aplicação. O serviço de compartilhamento de dados é disponibilizado através de API *Web service*, contendo informações de localização, parâmetros e tipos de dados necessários para a obtenção dos dados desejados.

Para desenvolver e organizar o nível de aplicação da arquitetura proposta,f processos de engenharia de software foram aplicados.

3.1.1.1 Desenvolvimento da Camada de Aplicação

Como mencionado anteriormente, o desenvolvimento de sistemas sensores para cenários metropolitanos é uma tarefa complexa. Para abordar essa etapa, a concepção da camada de aplicação compreendeu fases importantes na criação de um de sistema de software, incluindo: análise, modelagem do banco de dados, programação, *Web design* e testes. Essas etapas são descritas rapidamente a seguir:

- *análise*: essa fase corresponde ao levantamento de especificações do sistema em termos funcionais e não funcionais, para atender as necessidades dos usuários. A análise deve resultar em requisitos mínimos para a criação de um sistema computacional;

- *especificação e modelagem da base de dados*: representa a definição de estruturas de dados de regras de negócio do sistema. Informações de negócio são representadas em termos de objetos de dados, características e relacionamentos;

- *programação*: nessa etapa os requisitos funcionais do sistema e as regras de negócio são traduzidas em código. Compreende a elaboração e documentação dos módulos para a execução do sistema;

- *Web design*: compreende diferentes áreas do desenvolvimento de software. Parte principal de uma aplicação *Web*, o planejamento da interface é importante para que o sistema forneça elementos e informações comuns para o usuário, no caso, o administrador das redes sensoras.

- *testes*: tem o intuito de verificar se os requisitos levantados na etapa de análise foram respeitados e se as funcionalidades atendem as necessidades dos usuários. Problemas e inconsistências do sistema são identificados nessa etapa do processo.

Tendo em vista que a camada de aplicação responde pelo armazenamento de dados e também pelo provimento de interfaces de acesso, é importante separar os componentes da lógica de negócios, de apresentação da informação e de persistência de dados. Essa estratégia fornece segurança e flexibilidade para a arquitetura. Para descrever essa separação a organização do portal é descrita na próxima seção.

3.1.1.2 Portal *Web*

O portal representa a interface gráfica do sistema para o monitoramento das redes de sensores. Além dos dados coletados por elas, ele deve reportar para os administradores informações como: localização dos nós, topologia da rede, configurações dos dispositivos e gráficos estatísticos.

Em cenários urbanos vários ambientes podem ser monitorados simultaneamente, o que implica em diversas redes para monitorar. Os administradores também devem gerenciar as informações dos pontos monitorados, bem como fazer o gerenciamento de usuários e privilégios de acesso. Isso porquê pode haver vários administradores monitorando diversos ambientes, que por sua vez, não devem ser capazes de acessar redes que não estejam sob o seu domínio.

O portal apresenta elevado nível de abstração para fornecer uma interface genérica para os usuários e encapsular a heterogeneidade de tecnologias empregadas na camada de sensoriamento. Para tanto, a organização do portal considerou o padrão de arquitetura *Model-View-Controller* (MVC). Esse padrão permite dividir o sistema em camadas para facilitar a manutenção e a criação de interfaces. O MVC foi proposto por Reenskaug em 1979 (Reenskaug 2007) e é muito utilizado para a concepção de sistemas *Web* (Naruephiphat, Prom-Ya & Chansripinyo 2013).

Essa arquitetura de três camadas permite desacoplar a interface de apresentação da informação, da navegação e também do comportamento da aplicação, isto é, da lógica de negócios. A Figura 3.3 exibe a organização MVC do portal *Web*.

Uma das grandes contribuições da *Web* é o provimento de um cliente universal, o *Web browser* (Erl 2008). Dessa maneira, como pode ser observado na Figura 3.3, a apresentação dos dados está localizada na estação cliente, onde nenhuma aplicação específica precisa ser instalada. Essa abordagem retira o processamento do cliente e o centraliza na estação servidora, de modo que as operações realizadas pelos usuários são registradas diretamente nesse último, de maneira transparente. Sendo assim, a única imposição da arquitetura para acesso ao sistema é a utilização de um navegador *Web*.

A camada de visão (*view*) do MVC responde pela apresentação das informações. Essa camada fornece a interface de interação dos administradores com as diferentes redes sensoras existentes no município. Alguns trabalhos descritos na literatura têm propostos diferentes modelos de interface, como em Naruephiphat, Prom-Ya & Chansripinyo (2013) e Stojkoska & Davcev (2009). No entanto, as abordagens utilizadas estão quase sempre restritas a uma única categoria de dispositivos, como computadores pessoais. Com isso, a utilização do sistema através

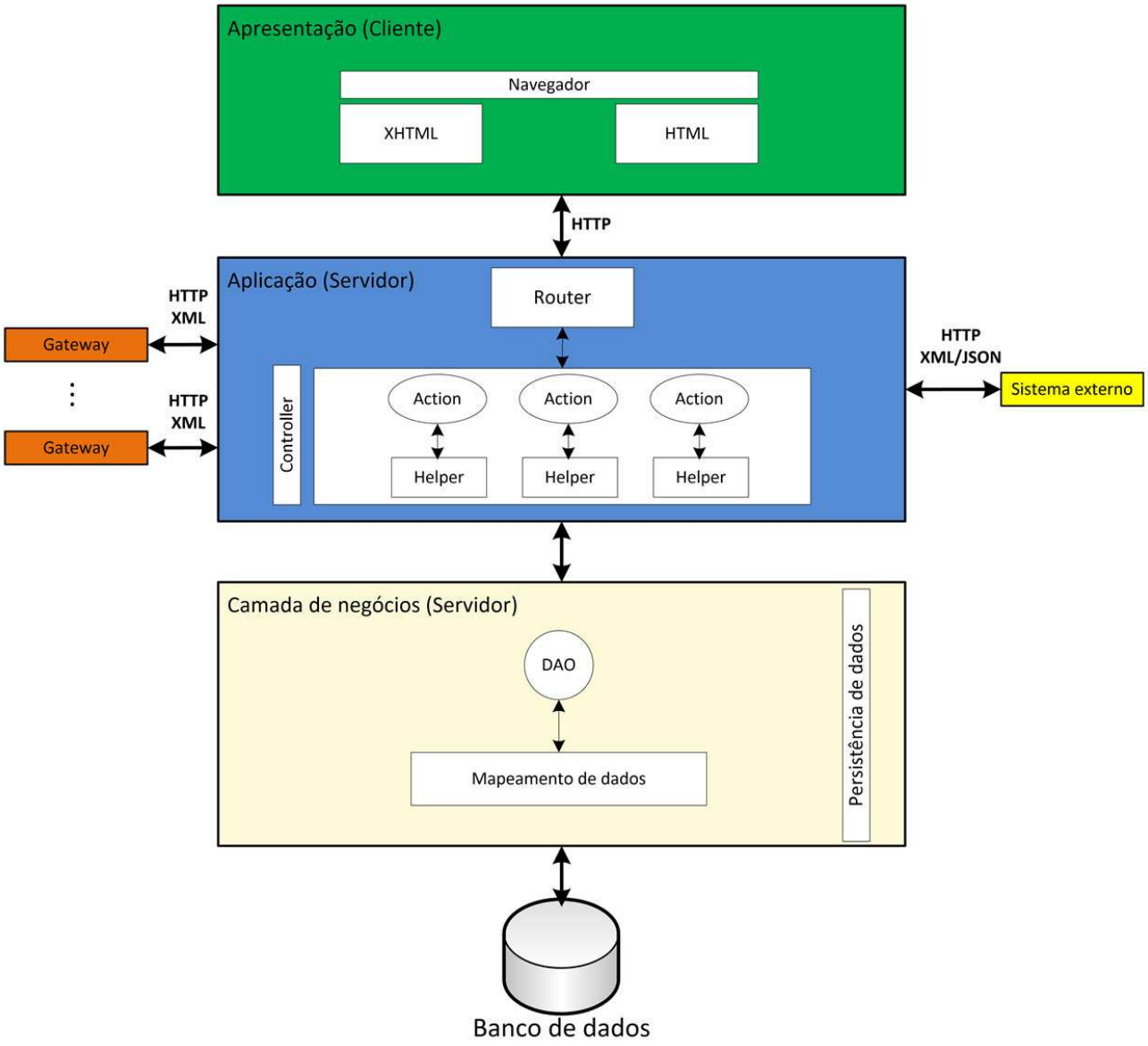


Figura 3.3: Componentes da camada de aplicação.

de outras classes de equipamentos implica no desenvolvimento de aplicações para plataformas específicas.

O acesso a sistemas *Web* está cada vez mais difundido por intermédio de dispositivos portáteis, como *tablets* e *smartphones*. De acordo com o IDC (2013), o mercado de *tablets* deve superar o de *notebooks* no ano de 2014. Dessa forma, aplicações *Web* precisam ser otimizadas para organizar a apresentação do conteúdo de acordo com o equipamento utilizado para acesso. Isso pode ser alcançado através do emprego de *Responsive Web Design*, o qual permite às aplicações *Web* a flexibilidade de se adaptar a diferentes resoluções de telas de dispositivos (Mohorovicic 2013) em tempo de execução, organizando a apresentação das informações. Enquanto algumas tecnologias estão perdendo espaço na *Web*, como o Flash, o W3C está trabalhando na quinta versão do HTML (HTML5), que fornecerá novos recursos suportados pela grande maioria dos navegadores, eliminando assim a necessidade de instalação de *plugins* ou

add-ons.

O nível de aplicação, exposto na Figura 3.3, é responsável por gerenciar as requisições dos clientes e preparar os dados para o nível de apresentação. Ao receber uma requisição, essa camada realiza um processo de roteamento com base na URL especificada para determinar o *controller* e a *action* responsáveis por atender aquela requisição. *Controllers* em MVC comunicam-se com as camadas de negócios e de apresentação, gerenciando o fluxo de informação entre elas. Como pode ser observado na Figura 3.3, os *gateways* e os sistemas externos se comunicam diretamente com essa camada, visto que eles não utilizam interface gráfica para interação com o sistema.

A camada de negócios, o *model* do MVC, corresponde ao código da lógica de negócio que funciona por trás da aplicação. Essa camada é responsável por encapsular os dados e as funções de aplicação do sistema, gerenciando o seu comportamento e os dados no domínio. Obter os dados e armazená-los em bancos de dados é responsabilidade desse nível da arquitetura MVC. Ela deve fornecer interfaces abstratas para o mapeamento de objetos de dados, com a finalidade de tornar a persistência e a recuperação de dados simples.

A arquitetura proposta não impõe nenhuma abordagem específica para o armazenamento de informações de sensoriamento, usuários e configurações, sendo possível utilizar métodos relacionais ou não-relacionais (NoSQL). Entretanto, elas devem ser robustas o suficiente para atender aos requisitos do cenário. É importante que a camada de negócios forneça mecanismos de abstração que desacoplem a lógica da aplicação desenvolvida do serviço de bancos de dados utilizado.

3.1.1.3 API *Web service*

As redes de sensores podem ser utilizadas em vários segmentos de um município. Nesse contexto, as cidades estão optando por implementar soluções tecnológicas para abordar problemas relacionados a determinados serviços, tais como: eletricidade; água potável; saneamento e transporte, agravados pelo aumento populacional (Bartlett et al. 2011). O principal problema com essas soluções é que, em geral, elas estão restritas a domínios de aplicação específicos, não compartilhando informações com demais sistemas para tornar a cidade inteligente no contexto global do município. A utilização de abordagens proprietárias e heterogêneas torna essa rigidez ainda mais evidente. Somente através de interfaces flexíveis, abertas e de fácil utilização, os diferentes agentes envolvidos (administração pública, empresas e cidadãos) serão capazes de conceber novas soluções para interagir e gerenciar todos os aspectos da vida urbana com uma ótima relação custo-benefício (Hernández-Muñoz et al. 2011).

Com o intuito de padronizar o compartilhamento dos dados coletados pelas redes de sensores implantadas em um município, a arquitetura proposta dispõe na camada de aplicação de Interface de Programação de Aplicação (API) para acesso ao *Web service* de consulta de dados. Uma API corresponde a uma interface de serviço estável que aplicações ou outros sistemas podem usar para interagir com o sistema em questão (Alonso, Casati & Kuno 2004). Dessa maneira, sistemas externos podem utilizar essas informações para realizar análises mais robustas, como por exemplo, correlacionar variáveis e eventos. Novas aplicações também podem ser desenvolvidas com os dados obtidos.

Conforme descrito no capítulo 2 existem diversas alternativas para a implementação de *Web*

services, todas se utilizam de formatos flexíveis de dados, principalmente do XML.

Consoante as características das tecnologias empregadas, *Web services* baseados em REST apresentam maior flexibilidade de implementação e melhor desempenho em comparação àqueles baseados em SOAP, conforme exposto no capítulo 2. Interfaces baseadas em REST possibilitam disponibilizar dados em formatos distintos, como XML e JSON, oferecendo aos clientes *Web* alternativas de utilização. Sendo assim, considera-se esse conceito ideal para promover o acesso aos dados coletados para sistemas externos. A imagem exibida na Figura 3.4 ilustra o funcionamento de um *Web service* baseado em requisições HTTP e respostas encapsuladas em JSON e XML.

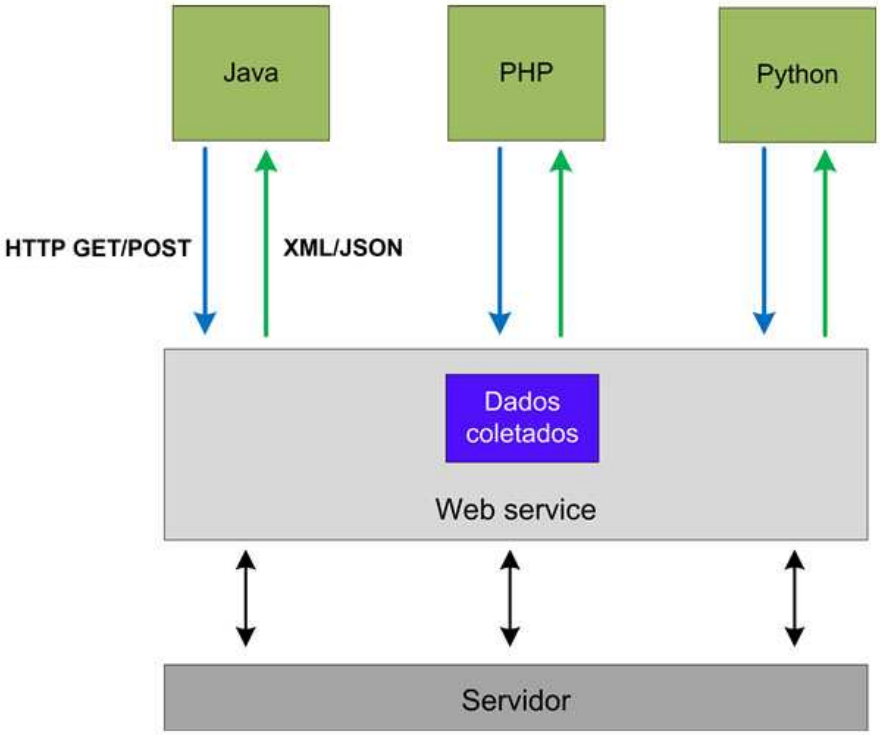


Figura 3.4: Exemplo de arquitetura *Web service*.

A interface uniforme do REST, baseada nos métodos fornecidos pelo protocolo HTTP, torna as interações baseadas em requisição e resposta auto-descritivas. Sendo assim, a lista de métodos e a semântica de *Web services* RESTful é fixa, característica que diferencia esse tipo de arquitetura de outras baseadas em RPC ou SOAP, onde a semântica de cada interação é específica de aplicação.

Utilizando o conceito de REST, cada serviço fornecido pelo sistema, no caso o acesso aos dados coletados pelas redes de sensores em diferentes ambientes, é identificado unicamente por um URI em forma de URL. Sendo assim, faz-se necessário somente a utilização de um cliente HTTP, como o próprio navegador *Web* para a sua utilização. Uma das restrições da arquitetura REST para melhorar a performance da rede é a utilização de *cache* de dados, para tornar as requisições às mesmas operações mais eficientes. No entanto, tratando-se de dados altamente sensíveis, como os provenientes de sensores, em geral, os recursos são rotulados como não-armazenáveis, pois dados obsoletos podem diminuir a confiabilidade das informações.

No contexto de RMAA a descrição da API para acesso aos dados dos sensores pode estar disponível no portal do município, contendo todas as informações necessárias para que clientes utilizem os serviços fornecidos. Na arquitetura, a definição dessa interface e da semântica do serviço por parte da entidade provedora, estabelece um contrato do tipo *take-it-or-leave-it*, que as entidades requerentes precisam aceitar sem modificações (W3C 2004). Nessa dissertação não é proposto nenhum modelo para a definição de APIs *Web service*, visto que cada solução desenvolvida pode possuir requisitos específicos para compartilhamento dos dados. Desse modo, fica a cargo do setor de Tecnologia da Informação do município a criação e a publicação de modelos para acesso aos dados.

3.1.2 Camada de Comunicação

Esta camada é formada pela rede de dados de uma RMAA e também por dispositivos *gateways*. A RMAA disponibiliza a infraestrutura física de comunicação capaz de interconectar os pontos monitorados em um município suportando o tráfego de informações. Já os *gateways* fornecem interfaces comuns para que os administradores possam realizar o monitoramento unificado das diferentes redes. As próximas seções descrevem os papéis exercidos por cada uma delas.

3.1.2.1 RMAA

Conforme exposto no capítulo 1 os dados em uma RMAA são transmitidos através de estruturas ópticas e sem fio baseadas nos protocolos TCP/IP. No entanto, o estado da arte de redes de sensores consiste em redes de área pessoal sem fio, de curto alcance e baixa capacidade de transmissão (Hui, juan & jun 2011), construídas principalmente sobre a família de protocolos IEEE 802.15. Assim, a interconexão direta entre as RSSF e a RMAA necessita de meios para o estabelecimento da comunicação que não degradem as informações e ofereçam a Qualidade de Serviço (QoS) adequada.

A infraestrutura de comunicação de uma RMAA é ideal para a aplicação de redes de sensores, possibilitando a implantação de redes de sensores de área pessoal sem fio, tais como ZigBee, em cenários de escala metropolitana. Ela fornece escala e eficiência para transmitir diferentes tipos de tráfego e interligar pontos remotos em uma cidade, de maneira simples.

A arquitetura que se propõe está baseada no modelo de RMAA proposto por (Mendes, Botolli & Breda 2010). Fisicamente, esse modelo é divido em três camadas: *Núcleo da rede*, *Camada de Distribuição* e *Camada de Acesso*. A Figura 3.5 destaca as camadas mencionadas.

O núcleo é formado tipicamente por um *backbone* de fibra óptica que constitui a parte central da rede que é responsável por suportar as intensas demandas de tráfego. Ele atende aos maiores geradores de tráfego da cidade. O núcleo deve ainda fornecer pontos para a interconexão com a camada de distribuição, conhecidos como Pontos de Presença (POP), os quais estão identificados na Figura 3.5 como *Gigabit* POP (GPOP), justamente devido a demanda de tráfego existente.

A Camada de Distribuição é a responsável por centralizar o fluxo de informações de entrada e saída dos pontos existentes na Camada de Acesso. Há diversos centros de distribuição nessa camada, os quais são conectados diretamente ao núcleo da rede por meio de GPOPs. As taxas

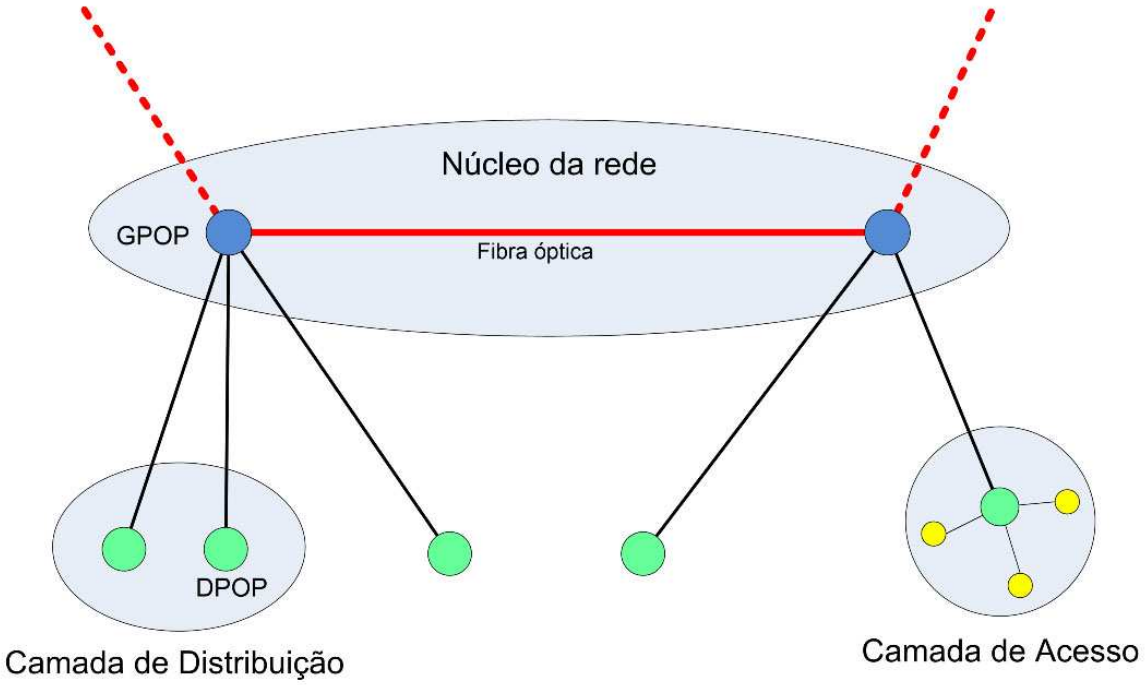


Figura 3.5: Arquitetura física de uma RMAA (adaptada de (Breda et al. 2011)).

de transferência nessa camada são inferiores as existentes no núcleo. No entanto, ela deve ser capaz de manipular algumas dezenas de megabits por segundo e até alguns gigabits. Os pontos nessa camada são identificados como Pontos de Presença de Distribuição (DPOP). Finalmente, a Camada de Acesso fornece pontos para conexão de residências e empresas. Os pontos nessa camada são derivados dos DPOPs. Nessa camada podem ser utilizadas tecnologias como Wi-Fi, par-trançado ou fibra óptica, desde que sejam compatíveis com as empregadas nos DPOPs.

Do ponto de vista lógico, a RMAA é construída sobre os protocolos TCP/IP e padrões Ethernet. A evolução do Ethernet permitiu a sua aplicação na construção de MAN e WAN (Breda et al. 2011). Além disso, outra propriedade importante dessa solução é o seu baixo custo, se comparados a redes SDH, ATM ou WDM (Mendes, Botolli & Breda 2010). Da mesma maneira, em virtude da sua ampla utilização com o advento da Internet, o conjunto de protocolos TCP/IP é o candidato ideal para a troca de dados na RMAA.

Na arquitetura proposta, o fluxo de mensagens trocadas entre o nível de aplicação e os dispositivos *gateways* é feito através da RMAA. Sua responsabilidade é interconectar os ambientes monitorados em um município, principalmente do setor público, conforme exposto no capítulo 1. A RMAA é um dos principais componentes da arquitetura proposta, tendo em vista seu caráter universalizante.

3.1.2.2 *Gateways*

Os dispositivos *gateways* são os responsáveis pela interconexão entre a RMAA e as redes de sensores. Eles funcionam como "pontes' que transmitem os dados das redes de sensores para a camada de aplicação e vice-versa. De acordo com Wang et al. (2009), se comparados a outros métodos de interconexão, como o emprego de TCP/IP a nível de sensores, os *gateways*

apresentam como vantagens a fácil implementação, baixo custo e maior suporte para a integração das redes. Ainda assim, muitos *gateways* pecam em flexibilidade, pois são desenvolvidos para aplicações específicas e apresentam alto acoplamento de protocolos e componentes.

Os *gateways* devem implementar procedimentos de adaptação para integrar logicamente as redes de sensores e a RMAA. Adaptações necessárias para o estabelecimento da comunicação estão relacionadas a:

- *adaptação de protocolo de comunicação*: tratando-se do ponto de interconexão entre a rede de dados TCP/IP e a rede de sensores, o *gateway* deve fazer a interface de tradução entre os paradigmas de comunicação empregados nas diferentes redes;
- *adaptação no formato dos dados transmitidos*: formatar informações de configuração e provenientes de observações e medições em formatos flexíveis, padronizados e independentes de plataforma.

A interoperabilidade de informações em nível de comunicação está baseada no uso do XML. Para estabelecer padrões de mensagens XML transmitidas entre o *gateway* e a camada de aplicação, considera-se o uso de *XML Schema*. Esquemas XML são utilizados para descrever a estrutura, o conteúdo e a semântica de documentos XML (W3C 2001). A abordagem utilizada viabiliza, de maneira rápida e eficiente, a troca e o processamento dos dados e promove a independência de plataforma.

A comunicação entre os equipamentos *gateways* e o servidor de aplicação e armazenamento deve ser realizada conforme a Figura 3.6.

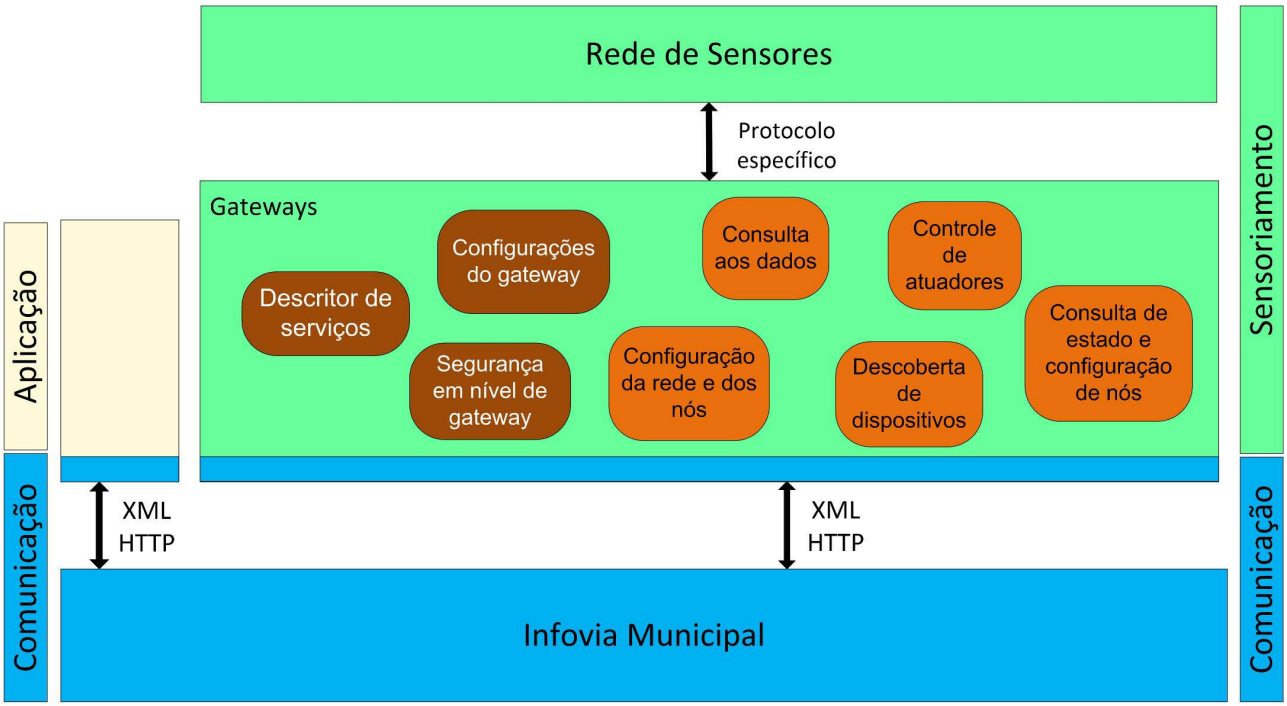


Figura 3.6: Comunicação com *gateways* e exemplos de serviços fornecidos.

.

É importante mencionar, conforme exposto na Figura 3.6, que o servidor na camada de aplicação não tem conhecimento dos protocolos de comunicação utilizados na camada de sensoriamento.

Cada *gateway* na arquitetura assume também o papel de sorvedouro da rede de sensores correspondente. Isso porquê todos os dados a serem reportados para o usuário, por meio de sistemas em nível de processamento de informação, são transmitidos através do *gateway*.

Conforme mencionado anteriormente, *gateways* atuam como provedores de serviços. Isto é, eles fornecem métodos que podem ser invocados através da interface de monitoramento e gerenciamento. O *gateway* deve publicar para a camada de aplicação as suas características e métodos suportados, bem como os parâmetros necessários para sua utilização. Dessa forma, eles atuam como *Web servers*, disponibilizando serviços de descoberta, manipulação e configuração das redes de sensores. A seguir são listados exemplos de serviços providos por um *gateway*, sendo eles:

- *descrição de serviços*: trata-se de um documento processável por máquina, que descreve os serviços oferecidos pelo *gateway* para a manipulação da redes de sensores subjacente. Este documento deve incluir a descrição dos serviços, protocolo de comunicação empregado e os parâmetros necessários para a sua utilização;

- *segurança*: corresponde à segurança em nível de *gateway*, ou seja, estratégias que podem ser empregadas para evitar o acesso e a manipulação indevida dos dados. Por tratar-se de um servidor *Web*, considera-se o emprego de autenticação HTTP baseada em usuário e senha para acesso aos serviços e uso do protocolo SSL para comunicação entre o *gateway* e o servidor na camada de aplicação;

- *monitoramento da RSSF e dados coletados*: dentre os serviços fornecidos pelos *gateways*, os principais correspondem a gerência dos nós que compõem as redes de sensores. Esses serviços devem permitir alterações nos parâmetros de configuração das redes em nível de sensoriamento. Como o *gateway* atua como o sorvedouro da RSSF, ele pode também implementar serviços de tratamento local de dados, como filtragem e agregação.

Para a criação de *Web services* nessa camada considera-se o uso de interfaces no estilo XML-RPC, visto que os recursos providos pelo *gateway* são dinâmicos. As características descritas na seção 2.2.6, principalmente a simplicidade, a possibilidade de criar canais de comunicação síncronos e assíncronos e a facilidade de implementação em rede foram os motivadores para o emprego desse protocolo em nível de comunicação.

Desse modo, os métodos fornecidos pelos *gateways* são invocados através de Chamadas de Procedimentos Remotos baseadas em XML. Embora o SOAP também posso atuar como protocolo para a chamada de procedimentos remotos, sua especificação é mais complexa e menos flexível que o XML-RPC, além deste último ser mais simples, de fácil entendimento e menos oneroso em termos de consumo de banda que o SOAP.

Para promover a independência de plataforma, o *gateway* deve encapsular os dados coletados pelas redes de sensores e possíveis mensagens de notificação, como alerta de perda de conexão com um nó, em mensagens XML e encaminhá-las para o servidor na camada de aplicação.

O formato das informações e dos metadados deve ser padronizado para todos os *gateways* existentes no município. Com isso, a comunicação entre a camada de aplicação e as redes de sensores ocorre de modo transparente, sem a necessidade de adaptações nos dados transmitidos. A criação de modelos semânticos e dicionários de dados pode ser utilizada para mapear informações de diferentes sistemas, de forma a prover uma convenção de nomes para esses sistemas. Nessa dissertação não são propostos modelos para a padronização dessas mensagens, ficando a cargo do setor de TI do município definir esse padrão.

A utilização de padrões *Web* na camada de comunicação permite o monitoramento unificado de diferentes redes sensoras, independente dos protocolos utilizados na camada de sensoriamento. Essa interoperabilidade é conceito chave para cenários urbanos e Cidades Digitais.

3.1.3 Camada de Sensoriamento

Como pode ser visto na imagem da Figura 3.2, na camada de sensoriamento encontram-se as redes de sensores encarregadas da coleta de dados nos ambientes onde estão inseridas. Cada local do cenário metropolitano pode conter uma ou mais redes de sensores, dependendo da área monitorada. Nessa camada da arquitetura podem ser empregados dispositivos com diferentes requisitos em termos de hardware, software e protocolos de comunicação.

Ainda de acordo com a Figura 3.2, pode-se observar que os *gateways* também fazem parte dessa camada, comunicando-se via protocolo específico com as redes de sensores, sendo os responsáveis pela execução de operações solicitadas na camada de aplicação.

Em geral, as iniciativas que empregam redes de sensores sem fio em ambientes de grandes proporções, como urbanos, são baseadas nos padrões de comunicação IEEE 802.11 a/b/g ou n. No entanto, conforme apresentado no capítulo 1, as redes de área pessoal sem fio apresentam características que as tornam adequadas para redes de sensores, tais como: baixa capacidade de transmissão e eficiência energética. A primeira característica dificulta a interceptação dos dados transmitidos pela rede. Com relação à eficiência energética, as WPAN, principalmente a especificação IEEE 802.15.4/ZigBee, possuem autonomia de energia de meses ou até mesmo anos, mantidas apenas por pequenas baterias. Essa característica viabiliza a utilização de dispositivos de tamanhos reduzidos e sua aplicação em locais remotos de difícil acesso, onde a substituição das fontes de energia é inviável ou mesmo impraticável. WPAN apresentam também menor custo de aquisição.

Com base nisso, esse trabalho foca na aplicação de redes IEEE 802.15.4/ZigBee, principalmente pelas características expostas na seção 1.1.3. Entretanto, a arquitetura idealizada considera que diferentes tecnologias de redes de sensores podem ser empregadas, devido a característica modular e a independência de plataforma apresentada por ela.

3.2 Modos de processamento da informação

Os dados coletados pelas redes de sensores são processados e apresentados para o usuário através do portal na camada de aplicação. Com base nessas informações, os administradores podem tomar decisões e promover alterações no ambiente monitorado. Porém, algumas dessas

informações podem ser irrelevantes, e dessa maneira, acabam por consumir, de modo desnecessário, a banda de comunicação.

A organização da arquitetura e os equipamentos que a compõem permitem que diferentes estratégias de gerenciamento e processamento de dados sejam adotadas. O montante de dados pode ser todo transmitido para a camada de aplicação, onde são tratados e reportados para o usuário. De modo alternativo, eles podem sofrer algum tipo de pré-processamento antes de serem transmitidos via RMAA. Estratégias de processamento centralizado e distribuído podem ser empregadas. Os dois modos são:

- *processamento distribuído*: nesse tipo de processamento os dispositivos *gateways*, na camada de comunicação, são responsáveis por tratar os dados antes de enviá-los para a camada de aplicação. O processamento distribuído pode considerar somente valores dentro de limiares definidos para transmissão ao nível de aplicação. *Gateways* podem também realizar operações de filtragem e agregação de dados, eliminando redundância proveniente de observações similares. Esse tipo de atividade pode também ser realizada em nível de rede de sensores, por *clusters head* e dispositivos com maior poder computacional;

- *processamento centralizado*: nessa forma de processamento todos os dados coletados por uma rede de sensores são propagados para a camada de aplicação através da RMAA. O servidor deve ser capaz de manipular grandes quantidades de dados que serão provenientes de diferentes fontes de sensoriamento. O *overhead* de tráfego na RMAA não é um problema, visto que ela é capaz de suportar intensas demandas de tráfego. Essa abordagem apresenta a desvantagem do único ponto de falha, pois se o servidor parar de funcionar os dados podem ser prejudicados. Entretanto, ela elimina a necessidade de utilização de dispositivos sofisticados em nível de *gateway*, tornando a solução mais viável financeiramente.

A escolha por uma ou outra forma de processamento é dependente da aplicação que a rede sensora está executando. Ambas apresentam vantagens e desvantagens, ficando a cargo do projetista da rede optar pelo modo que seja mais adequado às suas necessidades. É importante destacar que a arquitetura é flexível o suficiente para permitir ambas as alternativas.

3.3 Segurança

A natureza multisserviço de RMAAs impõe desafios aos analistas de segurança. Isso porque essa rede é responsável por interconectar prédios públicos de um município, onde dados sigilosos pertencentes a administração pública e aos munícipes são manipulados.

Com base nessa premissa, Miani (2009) aplicou um conjunto de métricas propostas para análise de segurança em RMAAs. O trabalho permitiu identificar vulnerabilidades nos níveis de rede, serviços e pontos de interconexão, e sugerir abordagens para a correção de problemas.

Na arquitetura proposta grande quantidade de informações são trocadas entre diferentes partes, tanto em nível de aplicação quanto de comunicação. Por se utilizar de *Web services*, a segurança da informação na arquitetura está fundamentada em cinco requisitos principais (Graham et al. 2002, capítulo 5):

- *confidencialidade*: garante que as informações estão protegidas contra interceptação;
- *autenticação*: garante que o acesso às informações e aos dados da aplicação é restrito à entidades devidamente reconhecidas;
- *integridade*: refere-se à garantia de que a informação não foi modificada durante a transmissão;
- *autorização*: processo para decidir se determinada entidade pode ou não acessar um recurso em particular.

Dependendo da aplicação, a privacidade e a proteção dos dados em redes de sensores é fator crítico, principalmente se essa conter atuadores. Esse trabalho não tem o objetivo de propor novas técnicas para a segurança da informação nesse cenário. No entanto, as demais seções apresentam algumas abordagens alternativas que podem ser adotadas para promover segurança nas três camadas da arquitetura, atendendo aos requisitos de segurança destacados acima.

3.3.1 Segurança na Camada de Aplicação

O acesso ao portal *Web* de monitoramento das redes de sensores deve ser restrito a usuários autenticados no sistema. Essa autenticação é realizada mediante o uso de credenciais de acesso baseadas em *login* e senha. Tal abordagem permite ao sistema identificar o administrador e definir quais informações e cenários estarão disponíveis para a sua manipulação, ou seja, ele emprega um sistema de controle de acesso baseado em papéis. Além disso, evita que entidades não autorizadas tenham permissão para realizar alterações indevidas.

As credenciais de acesso devem estar armazenadas em banco de dados, que por sua vez, também deve implementar mecanismos de controle de acesso. As senhas registradas no banco de dados podem ser criptografadas utilizando algoritmos de *hash* como MD5 (Rivest 1992) ou SHA1 (Eastlake 2001). Para evitar engenharia reversa, caso as senhas sejam roubadas, pode-se utilizar *salt*. Trata-se de uma cadeia de caracteres privada, somente conhecida pela aplicação, que se funde a senha do usuário para aumentar a segurança, tornando mais difícil, senão impossível, descobrir a senha original. Registros da utilização do sistema, como data e hora de acesso e de término de sessão, também são importantes para fins de auditoria e conferência.

Para que sistemas externos possam ter acesso aos dados coletados por redes de sensores via API, podem ser utilizadas chaves únicas (*keys*) para identificação de sistemas e usuários que estejam consumindo determinado serviço. Essa abordagem é comumente empregada na utilização de *Web service* fornecidos por empresas como Google[1] e Yahoo[2]. Os administradores das redes podem definir se os dados coletados pelas redes de sensores estarão ou não disponíveis via *Web service*.

[1] *Google Developers*: http://developer.google.com
[2] *Yahoo Developer Network*: http://developer.yahoo.com

3.3.2 Segurança na Camada de Comunicação

Questões de segurança na camada de comunicação estão relacionadas ao transporte seguro dos dados através da RMAA e ao acesso aos dispositivos *gateways*.

Uma das métricas de segurança na RMAA proposta por Miani (2009), em nível de rede, é a segmentação do tráfego de dados em sub-redes através do uso de *Virtual Local Area Networks* (VLANs). VLANs são implementadas em *switches* de camada de enlace (*link* de dados) e são utilizadas para separar logicamente o tráfego dos diferentes domínios de interesse (Tanenbaum & Wetherall 2011). Em uma RMAA, podem existir VLANs para escolas, prédios da administração pública, postos de saúde e para acesso da população. Nesse sentido, podem também ser criadas uma ou mais VLANs específicas para o tráfego de dados provenientes das redes de sensores existentes no município.

Visto que os dados coletados pelas redes de sensores podem ser sensíveis ao atraso e latência, é importante criar canais de comunicação que sofram o mínimo de influência de outras fontes de tráfego, pois isso agrega maior confiabilidade às leituras.

Em razão da comunicação entre os *gateways* e o servidor *Web* estar baseada em HTTP, o qual opera sobre o Protocolo de Controle de Transmissão (TCP), mecanismos de segurança *Web* podem ser empregados para o estabelecimento de conexões seguras. Um dos meios mais utilizados para esse fim é o *Secure Socket Layer* (SSL), protocolo criado em 1995 pela *Netscape Communications Corp.*, então dominante no mercado de *browsers* da época, em virtude do crescimento na utilização da *Web* para a realização de operações financeiras e demais envolvendo dados sigilosos (Tanenbaum & Wetherall 2011, Capítulo 8). O SSL pode ser representado como uma camada de segurança que atua entre as camadas de aplicação e de transporte do modelo TCP/IP, recebendo requisições provenientes de um cliente, um navegador HTML por exemplo, e enviando-as a camada de transporte para transmissão ao servidor via protocolo TCP. A Figura 3.7 ilustra a utilização do SSL.

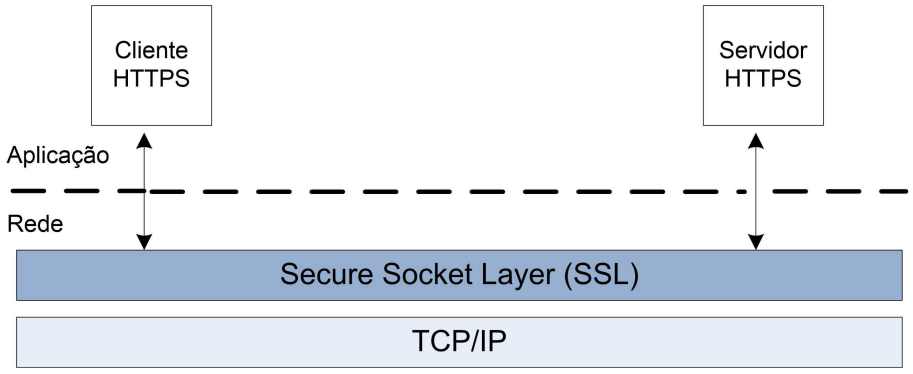


Figura 3.7: Troca de mensagens entre cliente e servidor HTTPS (adaptada de Alonso et al. (2004, capítulo 4)).

Quando o HTTP opera sobre SSL ele é chamado de *Secure HTTP* (HTTPS) e em geral utiliza a porta 443 ao invés da 80. HTTPS permite que clientes e servidores realizem autenticação e estabeleçam conexões seguras (criptografadas) entre eles (Alonso, Casati & Kuno 2004, capítulo 4). Assim, ele requer que ambos possuam suporte a SSL. Entretanto, a utilização de SSL

adiciona *overhead* de comunicação e aumento de recursos de processamento devido aos esquemas de criptografia empregados.

O método de segurança mais popular na Internet é a combinação de autenticação básica (*BASIC-AUTH*) e SSL (Graham et al. 2002). Essa abordagem é amplamente utilizada, principalmente devido à sua simplicidade de configuração. Dessa forma, em adição ao uso do SSL para segurança no transporte dos dados, é indicado também a aplicação de autenticação HTTP para acesso aos métodos disponibilizados pelos *gateways*. Uma interação típica de autenticação básica HTTP é exposta na Figura 3.8.

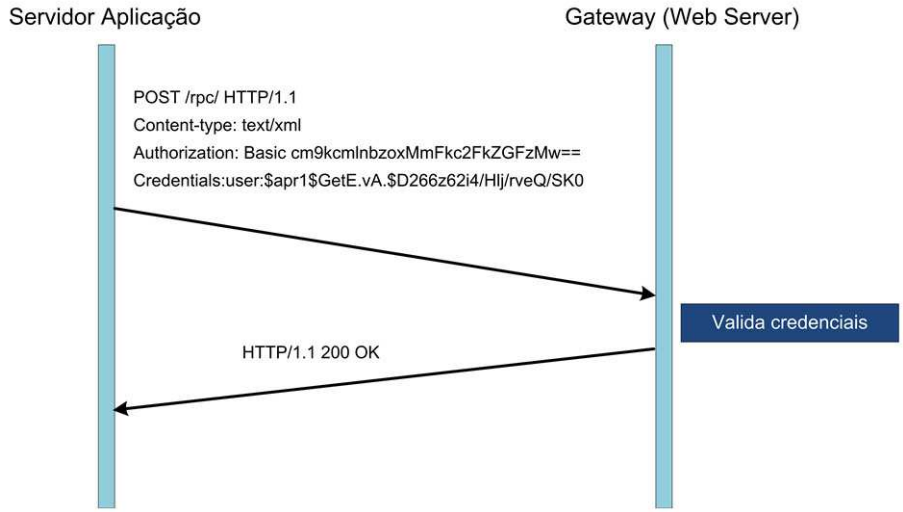


Figura 3.8: Interação do método básico de autenticação HTTP.

Para o caso de utilização do método POST do HTTP, as credenciais de autenticação devem ser enviadas juntamente com o documento XML da chamada RPC. Caso as credencias enviadas estejam corretas o procedimento é executado. Caso contrário, um erro HTTP 401 *Unauthorized* é retornado.

O método *BASIC-AUTH*, definido na RFC 2617, atende aos requisitos de autenticação e autorização previstos para a segurança da informação. Dessa maneira, as chamadas à métodos remotos devem ser realizadas mediante o fornecimento de credenciais de acesso. Somente com esses dados é possível obter informações e manipular as redes de sensores subjacentes.

3.3.3 Segurança na Camada de Sensoriamento

Segurança é uma questão importante em redes de sensores. Especialmente para as redes voltadas para aplicações médicas, automação doméstica ou industrial, militares e que manipulem dados privados. Essas redes podem sofrer ataques que afetam o seu funcionamento e permitam o roubo de dados.

Alguns exemplos de ataques que podem ocorrer em redes de sensores, de acordo com Du & Chen (2008), são: bloqueio; adulteração; colisão; manipulação de informações de roteamento; inundação (*flooding*) e clonagem de nós. Esses ataques ocorrem em diferentes níveis de rede, desde físico até aplicação. Dessa maneira, esquemas de segurança são necessários.

A segurança na camada de sensoriamento é dependente da tecnologia empregada. Conforme exposto na seção 1.1.3, a especificação ZigBee utiliza criptografia de chave simétrica AES através de chaves de *link* e de rede. Além disso, podem ser utilizadas também listas de controle de acesso e temporizadores de pacotes. No entanto, o emprego desse tipo de segurança pode ser uma desvantagem em termos de recursos da rede e esforço de desenvolvimento. Portanto, é importante avaliar a real necessidade de sua utilização (Faludi 2010, Capítulo 8).

A baixa potência de transmissão dos rádios também ajuda para que o sinal não se propague por longas distâncias, reduzindo as chances de interceptação. Além da utilização de criptografia, o próprio ambiente onde a rede está implantada pode funcionar como um componente de segurança, evitando que os seus nós sejam danificados ou mesmo furtados.

Outros aspectos de segurança em redes de sensores sem fio podem ser encontrados na literatura, como em Du & Chen (2008).

3.4 Considerações sobre o capítulo

Com base no estado da arte de redes de sensores sem fio e no desenvolvimento de Redes Metropolitanas de Acesso Aberto, esse capítulo apresentou a proposta de uma arquitetura baseada na *Web* para o monitoramento de redes sensoras aplicadas no cenário em questão. A ubiquidade da *Web* e a familiaridade de usuários com essa tecnologia, favorece a sua aceitação e utilização em larga escala.

O monitoramento de ambientes em um cenário urbano inclui grande diversidade de redes de sensores e de dispositivos heterogêneos em termos de hardware e software. Dessa maneira, é importante separar as especificidades das aplicações, dos protocolos de comunicação de dados, com o objetivo de fornecer uma abordagem genérica para a manipulação dessas redes.

A arquitetura descrita nesse capítulo é flexível por se utilizar de tecnologias *Web*, o que a torna capaz de prover escalabilidade e interoperabilidade entre diferentes sistemas. A arquitetura é baseada no conceito de orientação a serviços, onde *Web services* são utilizados tanto em camada de aplicação quanto de comunicação. Na camada de aplicação, sistemas externos acessam os dados coletados pelas redes de sensores, para promover melhorias para o município. Essa abordagem estimula o compartilhamento de informações de diferentes domínios, fornecendo amplo conjunto de dados para sistemas inteligentes e responsáveis pela tomada de decisões.

Na camada de comunicação, os *gateways* que interligam as redes de sensores à Rede Metropolitana de Acesso Aberto implementam interfaces padronizadas para acesso as configurações e dados providos pelas redes. Dessa forma, o sistema utilizado na camada de aplicação é não possui conhecimento dos protocolos de comunicação de dados empregados na camada de sensoriamento, necessitando apenas conhecer os serviços fornecidos por elas e os meios para acessá-los.

A RMAA fornece a infraestrutura capaz de suportar intensas demandas de tráfego e interligar entidades importantes em um município. Assim, o modelo arquitetural apresentado permite a criação de um ambiente único e integrado para monitoramento das redes de sensores. Adicionalmente ao modelo em camadas pode-se destacar a modularização, o baixo acoplamento entre os componentes e diferentes níveis de abstração. A proposta permite o desacoplamento da lógica de negócios da apresentação das informações, dessa maneira, a responsabilidade de

processamento está concentrada no servidor de aplicação, não compartilhando responsabilidades de processamento com clientes, o que resulta em "clientes-leves", encarregados apenas da apresentação dos dados.

Vale salientar que a arquitetura proposta possibilita a distribuição e hierarquização do processo de tomada de decisão e gerenciamento, que pode ser distribuído entre as três camadas da arquitetura: sensoriamento (rede de sensores); comunicação (*gateway*) e aplicação (servidor).

Embora o foco do trabalho não seja propor novos métodos para garantir a segurança em sistemas sensores, a fim de atingir um maior grau de confiabilidade, o capítulo descreveu também algumas abordagens pertinentes a segurança da informação que podem ser empregadas nas diferentes camadas que compõem a arquitetura.

Aspectos de ordem técnica e não-técnica, bem como eventuais problemas que possam surgir a partir da aplicação da arquitetura, serão avaliados com base no estudo de caso apresentado no capítulo 4.

Capítulo 4

Estudo de caso

O estudo de caso em um ambiente real permite identificar possíveis problemas de ordem técnica e não-técnica que possam surgir. Pode-se observar na literatura que grande parte das propostas relacionadas são testadas em ambientes controlados, não representando de maneira ideal o cenário-alvo de aplicação.

A falta de relatos de experiências no campo de redes de sensores aplicadas em Cidades Digitais pode prejudicar a evolução de Tecnologias de Informação e Comunicação. Nesse sentido, a infraestrutura de comunicação de uma RMAA fornece condições ideais para o desenvolvimento de experimentações dessa natureza.

São inúmeras as soluções que podem ser alcançadas com os recursos da arquitetura proposta nesse trabalho. Com o intuito de avaliar a sua viabilidade, realizamos uma prova de conceito por meio de estudo de caso em uma Cidade Digital brasileira, fundamentada sobre as soluções de uma RMAA. O local de aplicação foi a cidade de Pedreira, situada no interior do estado de São Paulo. Nesse município foram monitoradas redes de sensores heterogêneas, implantadas em dois locais públicos. Para realizar esse monitoramento, desenvolvemos um sistema considerando as tecnologias descritas na arquitetura.

O cenário de aplicação, a modelagem do sistema e os componentes das camadas que compõem a arquitetura são descritos em detalhes nesse capítulo. Além disso, também apresentamos os resultados do monitoramento e, finalmente, fornecemos algumas considerações sobre o estudo realizado.

4.1 Cenário de aplicação e organização da arquitetura

Conforme mencionado, o estudo de caso desenvolvido foi realizado no município de Pedreira, situado no interior do estado de São Paulo. A cidade possui um número aproximado de 45 mil habitantes (Instituto Brasileiro de Geografia e Estatística 2010) e é conhecida como a "capital da porcelana", sendo essa atividade a sua base econômica.

No ano de 2006 Pedreira recebeu o projeto de uma Rede Metropolitana de Acesso Aberto, iniciativa do Laboratório de Redes de Comunicações (LaRCom) da UNICAMP em parceria com a prefeitura municipal. Após a conclusão do projeto, Pedreira tornou-se o primeiro município brasileiro a ser conhecido como uma Cidade Digital (Breda et al. 2011).

A sua infraestrutura de comunicação é híbrida, composta de células de acesso sem fio baseadas em padrões IEEE 802.11 **a** e **g**, cerca de 60 células espalhadas pela cidade, e de um *backbone* Gigabit Ethernet de fibra óptica, em torno de 30 km de fibra óptica monomodo.

O objetivo principal do projeto da RMAA de Pedreira é a interconexão de instituições e prédios públicos, tais como escolas, prefeitura e centros de saúde. A Figura 4.1 exibe o mapa completo com os prédios interconectados pela RMAA e as tecnologias empregadas nas conexões.

A infraestrutura óptica da RMAA de Pedreira é distribuída através dos postes de energia elétrica, interconectando cerca de 13 prédios ao *backbone* principal através de *links* de 1 Gbps. O nó central da rede está localizado na prefeitura municipal, onde todo o tráfego é centralizado. Além de interconectar prédios públicos da cidade, a RMAA conecta também cerca de 4200 residências, oferecendo inúmeros serviços a municipalidade, como por exemplo, o acesso à Internet, sem custos, promovendo a democratização do acesso à informação e a inclusão digital.

O mapa da Figura 4.1 exibe o caminho percorrido pela fibra e os enlaces sem fio que alimentam determinados pontos da cidade. A tabela 4.1 exibe a identificação dos pontos interligados pela RMAA.

Tabela 4.1: Pontos interconectados pela RMAA de Pedreira.

Ponto	**Localização**
P01	Prefeitura municipal
P02	Morro do Cristo
P04	Cemitério municipal
P08	Parque Bela Vista
P09	Estação de Água Vila Santo Antônio
P10	Bairro Jardim Santa Clara
P12	Caixa d'água Limoeiro
P13	Bairro Jardim Alzira
P16	Caixa d'água Jardim Andrade
P18	Bairro Estância Santa Rita
P20	Caixa d'água Xícara V.M Alegre
P26	Arnaldo Rossi
P29	EMEF Maria Helena
P34	EMEI São José
P35	Hospital Consaúde
P36	Caixa d'água Idalina A. Steula
P40	Pronto Socorro São Rafael Arcanjo
P41	Caixa d'água Marajoara
P43	Creche Marajoara
P45	Caixa d'água Rainda da Paz
P46	Beija-Flor
P47	EMEI Jurandir Piva
P48	Posto de saúde Samuca
P49	Posto de saúde Jardim Triunfo
P50	EMEI Gerson Avilla

P52	EMEI Benedita Lago
P54	Bairro Morumbi
P57	SENAI
P58	Bairro Macedo
P59	EMEI Jorge Mari
P61	Central de Saúde
P62	EMEI Jesus Menino
P69	EMEI José Moratori
P70	EMEF Humberto Piva
P72	EMEI São Benedito
P73	Secretaria de Educação
P80	Escola Ayrton Policarpo
P81	Caixa d'água do Panorama LINK NET
P87	Bairro Jardim São Nilo
P90	Escritório da Infovia municipal
P97	Bairro São Kobayashi

Na Tabela 4.1 as siglas EMEI e EMEF correspondem a Escola Municipal de Ensino Infantil e Escola Municipal de Ensino Fundamental, respectivamente.

Os pontos identificados representam escolas, centros de saúde, órgãos do governo e demais prédios que estão interconectados pela RMAA de Pedreira, além de bairros e caixas d'água utilizados como pontos na abertura de células de rádio.

Com isso a cidade de Pedreira se apresentou como candidato ideal para a aplicação e avaliação da arquitetura que estamos propondo. Três pontos dos listados na tabela 4.1 foram utilizados no estudo de caso, sendo eles: P01, P02 e P90. A motivação para a utilização desses locais e a descrição de cada um deles são apresentadas mais adiante, na seção 4.4.1.

Os pontos P01 e P02 foram monitorados entre os meses de março e maio de 2014. Para realizar esse monitoramento, o sistema foi implantado de acordo com a arquitetura que estamos propondo, ganhando a forma apresentada na Figura 4.2.

Para cada uma das camadas expostas na Figura 4.2 se fez necessário ou o desenvolvimento de ferramentas computacionais ou a configuração de dispositivos eletrônicos para a realização do monitoramento. Os recursos fornecidos pela RMAA de Pedreira foram utilizados para fornecer a base da arquitetura, sobre a qual as demais camadas foram estruturadas. Na camada de sensoriamento, redes heterogêneas foram implantadas no município, consistindo de uma rede ZigBee e de um dispositivo de telemetria com sensores e atuadores embarcados, denominado *Telemetric*.

A próxima seção descreve a modelagem realizada para se conceber o sistema de monitoramento ambiental utilizado no estudo de caso. As demais seções descrevem os componentes desenvolvidos e utilizados em cada uma camadas da arquitetura exposta na Figura 4.2.

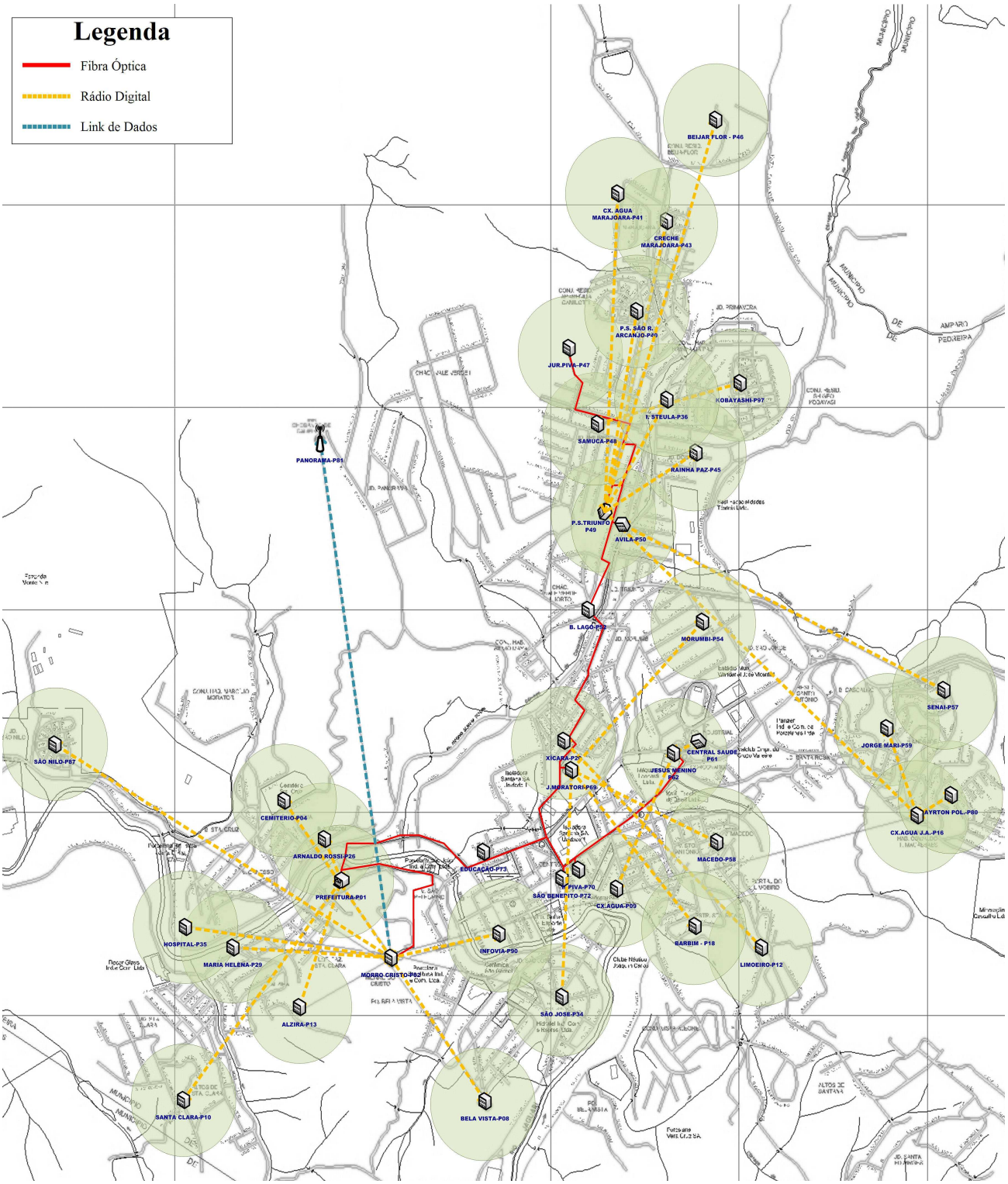


Figura 4.1: Prédios interconectados pela Infovia de Pedreira.

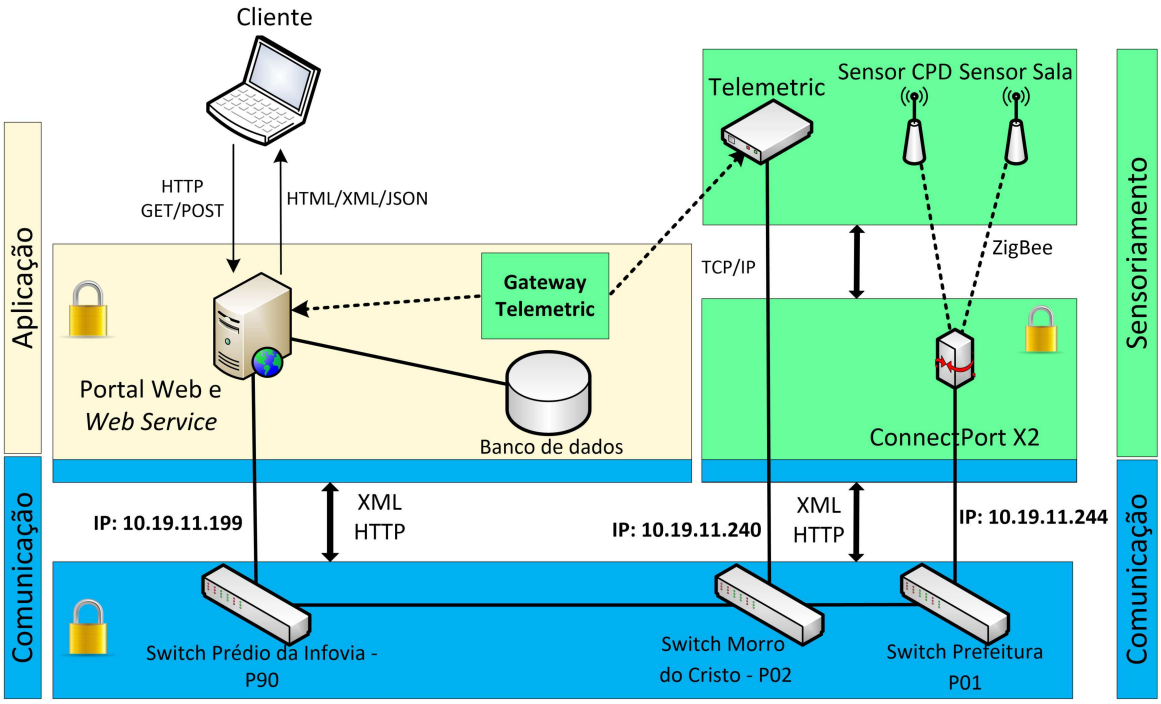


Figura 4.2: Arquitetura aplicada em Pedreira.

4.2 Modelagem do sistema

A elaboração do sistema para monitoramento ambiental utilizado no estudo de caso compreendeu o ciclo descrito na seção 3.1.1.1.

Independente do contexto de aplicação de redes de sensores, o seu principal objetivo é a coleta de dados. Sua implantação em escala urbana apresenta como principal desafio o monitoramento de um número crescente de dispositivos heterogêneos e geograficamente distribuídos no município.

De maneira geral, as principais características identificadas para o sistema de monitoramento desenvolvido foram:

- possibilitar o monitoramento unificado das entidades que compõem as redes de sensores;
- coletar e armazenar em uma base central de conhecimento as leituras dos sensores, informações dos ambientes e dos administradores relacionados;
- permitir o gerenciamento de usuários;
- implementar mecanismos de segurança no acesso ao sistema;
- oferecer diferentes formas de visualização dos dados coletados;
- prover interface de acesso a sistemas externos para o compartilhamento de dados;
- ser capaz de atender a múltiplas aplicações.

Levantadas as características do sistema é possível modelar os seus componentes e o relacionamento entre eles.

4.2.1 Modelo de domínio

O *design* de um sistema de redes de sensores para RMAA deve considerar características específicas desse cenário. Isso porque é necessário utilizar diferentes dispositivos e tecnologias, para permitir a criação de um ambiente sensor em escala municipal.

Sendo assim, para ilustrar as entidades envolvidas no estudo de caso utilizamos a Linguagem de Modelagem Unificada (UML). Esta linguagem permite representar os componentes graficamente, de forma a modelar as interações e o fluxo de informações entre eles.

Uma importante etapa no desenvolvimento do sistema é a criação de um modelo de domínio, o qual é utilizado para representar classes conceituais significativas no contexto do domínio do problema (Larman 2005). Vale salientar que nesse modelo devem ser representadas classes conceituais do mundo real, não de componentes de software.

Existem várias formas de representar um modelo de domínio. Nesse trabalho utilizamos o paradigma de orientação a objetos através da criação de classes conceituais, definição de atributos e relacionamentos (generalização, composição e associação) entre elas. A notação UML de um modelo de domínio é representada na forma de diagrama de classes. Definições de operações (métodos) e responsabilidades são dispensadas nesse modelo.

A Figura 4.3 exibe o modelo de domínio contendo as principais classes conceituais do ambiente em questão.

Embora o contexto de aplicação seja a RMAA, o modelo de domínio apresentado na Figura 4.3 exibe elementos comuns para diversas aplicações de redes de sensores. Entretanto, existem especificidades pertinentes ao cenário metropolitano.

Uma rede de sensores é constituída por um ou mais terminais sensores, que por sua vez são formados pelos módulos de sensoriamento (*sensor*), comunicação (*rádio*) e fonte de energia, os quais possuem seus próprios atributos. Cada ambiente monitorado no cenário metropolitano é dotado de um ou mais dispositivos do tipo *gateway*. Essa entidade possui atributos singulares que a tornam capaz de interconectar uma rede de sensores à infraestrutura da RMAA.

Tendo em vista que um cenário metropolitano pode conter várias redes de sensores em diversas localidades (*local* e *seção*), consequentemente, há também diferentes usuários responsáveis pelo monitoramento dos ambientes. A entidade *usuário* representada no modelo de domínio está relacionada com uma ou mais localidades e possui acesso as informações provenientes das redes de sensores implantadas com base no seu perfil de acesso e nível de privilégio.

Compõe ainda o modelo de domínio a entidade responsável pelo registro de informações no sistema. Essa é especializada em outras duas classes conceituais. Uma delas representa os dados coletados pelas redes de sensores e a outra registra possíveis notificações, como mensagens de alerta (*notificação* e *dados coletados*).

A aplicação do diagrama de classes UML para um modelo de domínio produz um modelo de visão conceitual (Larman 2005, capítulo 9). Identificar um conjunto de classes conceituais que formam um sistema é importante para dar suporte a um melhor entendimento e comunicação.

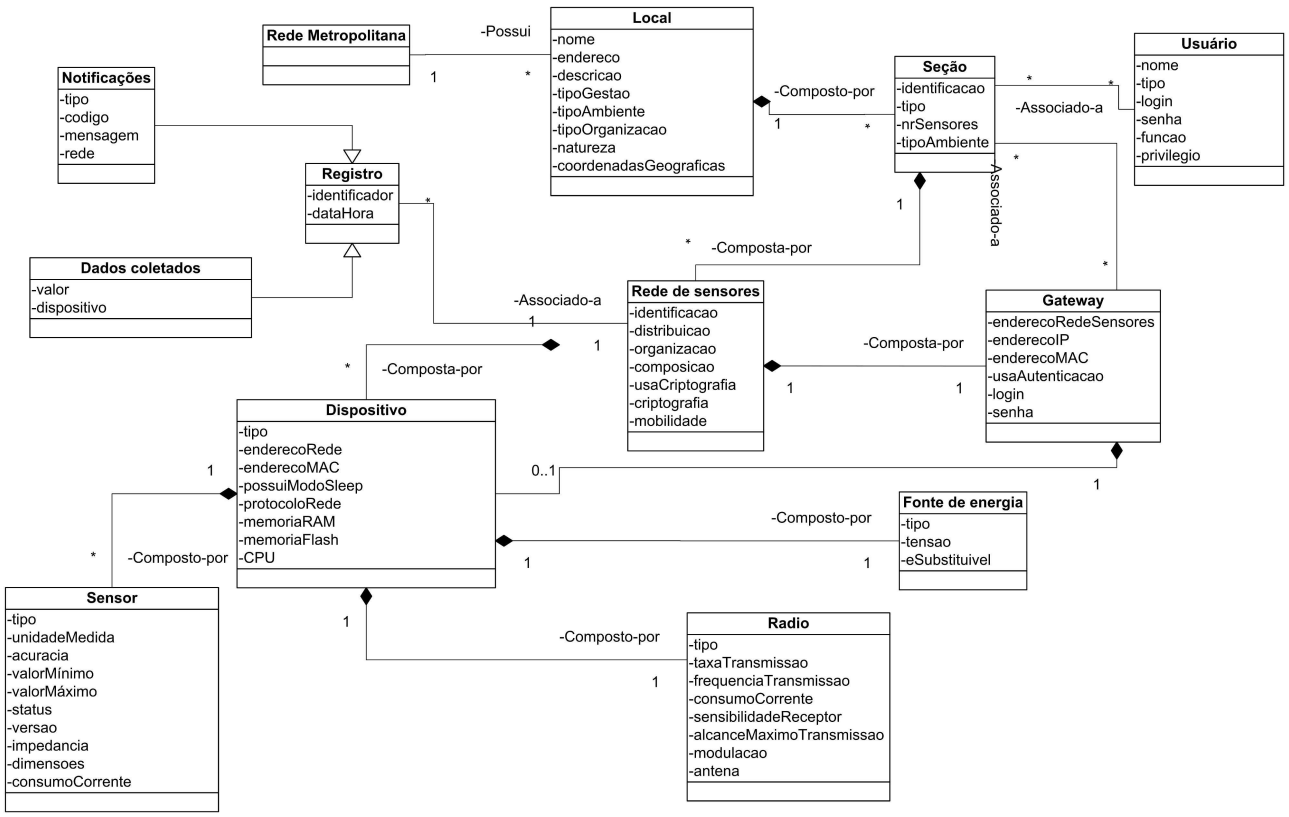


Figura 4.3: Modelo de domínio.

4.3 Camada de Sensoriamento

Duas tecnologias distintas foram empregadas na camada de sensoriamento. Uma pequena rede ZigBee, operando em topologia estrela, composta por dois dispositivos sensores e um dispositivo sorvedouro, foi responsável por coletar dados de temperatura no ambiente da prefeitura. O sorvedouro da rede ZigBee é parte componente do *gateway* da camada de comunicação. No morro do Cristo, um equipamento IP de maior porte denominado *Telemetric* foi empregado para monitorar tensões AC e DC, estado de tomadas (*On/Off*) e temperatura ambiental. Os nós empregados em nível de sensoriamento são descritos em maiores detalhes nas próximas seções.

4.3.1 Terminais da rede ZigBee

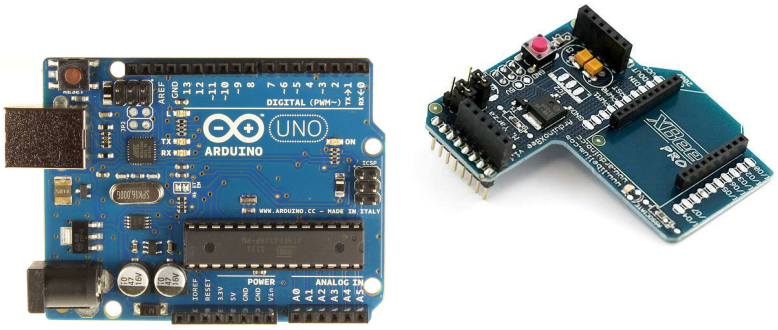
A camada de sensoriamento consiste dos dispositivos responsáveis pela percepção de fenômenos de interesse no ambiente monitorado. Na rede ZigBee, dois nós sensores constituem essa camada da arquitetura proposta. Eles foram responsáveis pela coleta de informações de temperatura no ambiente da prefeitura. Para isso, utilizamos duas placas microcontroladoras Arduino Uno R3 (Arduino 2014) para formar os terminais dessa rede. O Arduino Uno é baseado no microcontrolador ATmega328, possui 14 pinos de entrada/saída digitais, seis entradas analógicas, conexão USB e conector de energia. As principais características do Arduino são expostas na tabela 4.2.

Tabela 4.2: Especificações do Arduino Uno R3.

Microncontrolador	ATmega328
Voltagem operacional	5V
Tensão de entrada	7-12V
Memória Flash	32KB do ATmega328 dos quais 0.5 são usadas para inicialização
SRAM	2KB
EEPROM	1KB

O Arduino foi escolhido por se tratar de uma placa microcontroladora de baixo custo de aquisição e por possuir quantidade satisfatória de pinos digitais e analógicos para a utilização de sensores e atuadores.

Para possibilitar a comunicação ZigBee do *gateway* com os Arduinos, foi necessária a utilização de *shields* XBee. A Figura 4.4 exibe ambos componentes: Arduino Uno; *shield* XBee.



(a) Arduino Uno R3 (b) *Shield Arduino XBee*

Figura 4.4: Arduino Uno e *Shield* XBee utilizados no estudo de caso.

A coleta das informações de temperatura do ambiente foi realizada por dois modelos diferentes de sensores. Um dos modelos utilizados, o LM35, é um sensor de precisão com tensão de saída linearmente proporcional a temperatura do ambiente em graus Celsius. O LM35 não requer calibração externa ou *trimming* para fornecer acurácia sobre uma faixa de temperatura variando entre -55ºC e +150ºC. De acordo com o *datasheet* do sensor, sua precisão é de aproximadamente $\pm\frac{1}{4}$ºC. O outro modelo de sensor utilizado, o LM61, possui tensão de saída linearmente proporcional a temperatura em graus Celsius (+10 mW/ºC), capaz de detectar temperaturas entre -30ºC e +100 ºC, operando com tensão de entrada de 2.7V. A acurácia do LM61 é de ±2.0ºC. Cada terminal da rede foi equipado com um dos sensores mencionados.

Para estabelecer comunicação dos terminais sensores com o *gateway* via protocolo ZigBee, utilizamos dois módulos Digi XBee 802.15.4/ZigBee *Series* 2, comercializado pela *Digi International Inc.*. Os módulos possuem antena *wire*, com frequência de transmissão de 2.4 GHz, potência de transmissão de 1.25 mW e alimentação de 3.3V. Seu alcance de transmissão é de até 40 metros em ambientes internos e até 120 metros em ambientes externos. A Figura 4.5 exibe um dos módulos XBee utilizados.

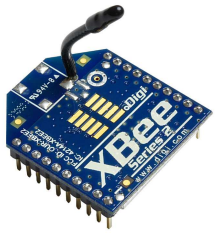


Figura 4.5: Módulo Xbee *Series* 2 utilizado.

O XBee foi desenvolvido para trabalhar com o padrão IEEE 802.15.4, provendo uma solução confiável, de baixo custo e baixo consumo para redes de sensores sem fio. Ele promove a interoperabilidade com dispositivos ZigBee de outros fabricantes, apresentando características de autocura e descoberta para a estabilidade da rede. O XBee corresponde ao módulo de comunicação do nó sensor.

A imagem da Figura 4.6 exibe a visão sistêmica dos nós criados.

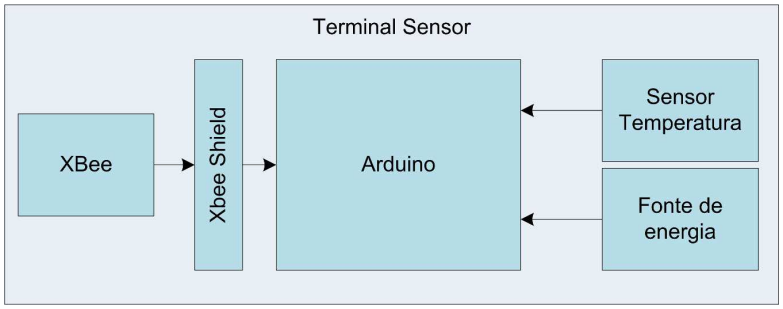


Figura 4.6: Visão sistêmica de um nó sensor ZigBee criado.

Os componentes dos nós desenvolvidos compreendem os descritos na seção 1.1.2. Cada terminal sensor é constituído por um transceptor, no caso um XBee, um módulo sensor de temperatura, LM61 ou LM35, processamento no microcontrolador do Arduino e uma fonte de energia. Como módulo de energia optamos por utilizar fontes de alimentação de 9V ligadas a rede elétrica do prédio, de modo que o gerenciamento de energia não foi uma preocupação na rede.

A Figura 4.7 exibe o módulo equipado com o sensor LM61 implantado no CPD da prefeitura.

Os módulos XBee foram estaticamente configurados antes da implantação da rede. Suas configurações de *firmware* foram modificadas com a aplicação *X-CTU*, fornecida pelo fabricante. Essa fase de configuração é necessária para a definição de funções executadas pelos nós e inicialização da rede. Alterações dinâmicas podem ser realizadas após a implantação da rede por meio do sistema desenvolvido.

O terceiro componente da rede sensora ZigBee é parte integrante do dispositivo *gateway*. Trata-se de um módulo XBee 802.15.4/ZigBee com antena RPSMA 2.4GHz, configurado para atuar como coordenador da rede. Ele é o responsável por inicializar e gerenciar a rede. No estudo de caso, ele também desempenhou o papel de sorvedouro, recebendo diretamente os dados coletados pelos terminais sensores.

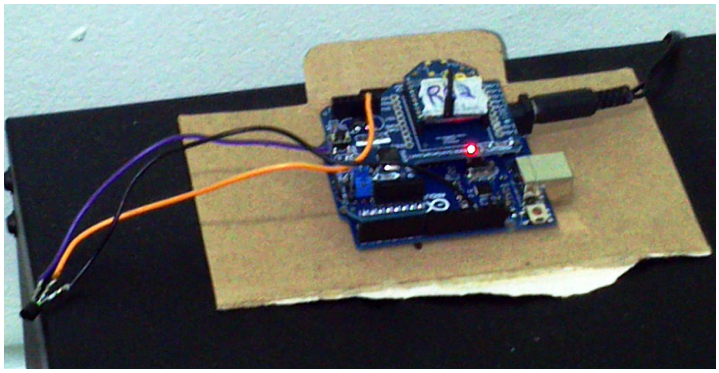
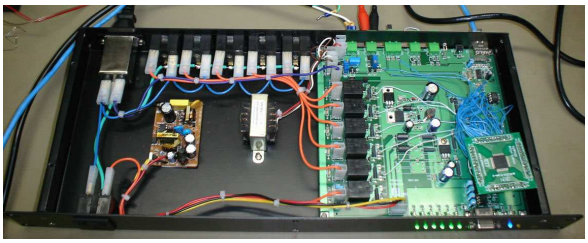


Figura 4.7: Nó sensor ZigBee implantado no CPD da prefeitura.

4.3.2 Telemetric

O *Telemetric* é um equipamento desenvolvido pela empresa IgnisCom para realizar o monitoramento e controle de dispositivos remotamente. Sua dimensão de tamanho 1 "U" (*Rack unit*) é adequada para a criação de um sistema de supervisão.

Ele possui sensores e atuadores embarcados e, assim como a rede ZigBee, compõe a camada de sensoriamento da arquitetura. Sua utilização no estudo de caso tem por objetivo avaliar a capacidade da arquitetura em abstrair a heterogeneidade de tecnologias utilizadas em nível de sensoriamento. A Figura 4.8 exibe imagens de um protótipo de um *Telemetric*, o qual corresponde ao modelo existente no prédio do morro do Cristo em Pedreira..



(a) Visão do circuito (protótipo)

(b) Vista traseira (protótipo)

Figura 4.8: Dispositivo *Telemetric*.

O *Telemetric* possui sensores e atuadores que permitem monitorar e controlar equipamentos.

Na Infovia municipal de Pedreira existem diversos *Telemetrics* implantados. No estudo de caso utilizamos o *Telemetric* do morro do Cristo (P02) para monitorar o estado (*On/Off*) de três tomadas, a temperatura do ambiente em graus Celsius, a tensão da rede elétrica e a tensão DC gerada por uma bateria.

O *Telemetric* pode ser gerenciado através do protocolo SNMP. Ele também possui uma interface *Web* para monitoramento em tempo real de sensores e controle de dispositivos atuadores. Para promover a independência de plataforma, proposta pela arquitetura, a comunicação com o *gateway* ocorreu através da troca de mensagens em arquivos XML. O *gateway* foi o responsável por permitir a comunicação uniforme do *Telemetric* com o nível de aplicação.

4.4 Camada de comunicação

A camada de comunicação da arquitetura proposta compreende a infraestrutura da RMAA de Pedreira e os dispositivos *gateways*, responsáveis por interconectar os diferentes paradigmas de comunicação utilizados na rede ZigBee e no *Telemetric*. Essa seção descreve a RMAA, destacando os pontos utilizados no estudo de caso, e os *gateways* empregados nessa camada da arquitetura.

4.4.1 RMAA - Pontos utilizados no estudo de caso

Conforme mencionado no início do capítulo, foram escolhidos três locais distintos conectados pela RMAA de Pedreira para a aplicação do estudo de caso. Os locais escolhidos correspondem ao prédio da prefeitura municipal (P01), o morro do Cristo (P02) e o centro de gerência da Infovia (P90).

A imagem da Figura 4.9 destaca os pontos utilizados no estudo de caso.

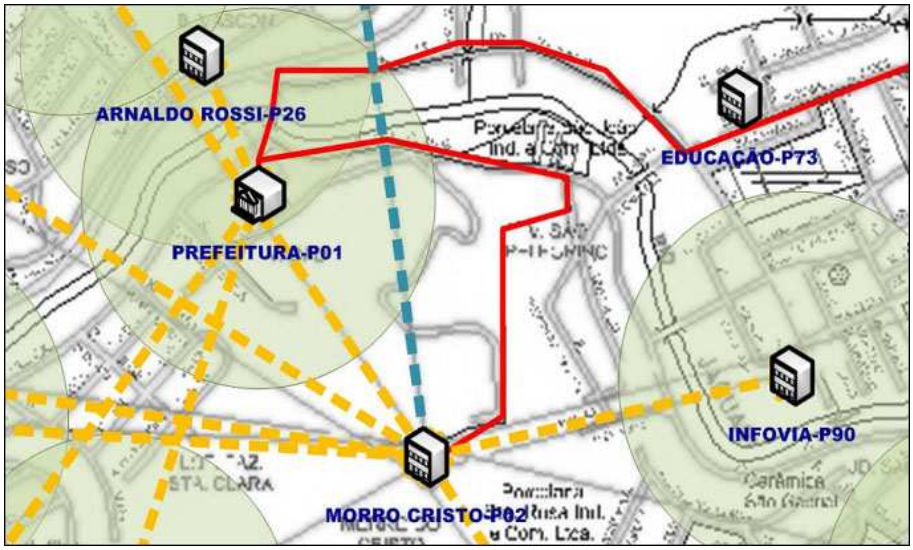


Figura 4.9: Centro de gerência da Infovia, prédio da Prefeitura e do Morro do Cristo utilizados no estudo de caso.

Como pode ser observado na Figura 4.9, a rede de fibra óptica termina no morro do Cristo, a partir do qual a Infovia é distribuída para locais mais distantes por meio de *links* de rádio.

Escolhemos o prédio de gerência da Infovia (P90) para a implantação da camada de aplicação da arquitetura, ou seja, os servidores *Web* e de banco de dados. Nesse ambiente encontra-se a equipe encarregada pela manutenção da RMAA, responsável por monitorar os pontos conectados para a identificação e correção de falhas. Em virtude dessas características e também da disponibilidade para uso de equipamentos, esse local se mostrou propício para a implantação da camada de aplicação da arquitetura.

O outro ponto utilizado no estudo foi a prefeitura municipal (P90). Esse ponto possui um Centro de Processamento de Dados (CPD), local onde o tráfego da RMAA é centralizado. Nesse ambiente nós implantamos a rede ZigBee, composta por dois terminais sensores de temperatura

e por um nó sorvedouro. Os sensores monitoraram a temperatura ambiental do CPD e também de uma sala próxima, possibilitando assim a comparação entre os valores coletados.

No morro do Cristo encontra-se o equipamento *Telemetric*, responsável por realizar o monitoramento e controle em *hacks* de equipamentos e que compõe a camada de sensoriamento da arquitetura. Esse é um importante local a ser monitorado pois ele é o responsável por interconectar pontos distantes a Infovia por meio de *links* de rádio.

4.4.2 *Gateways*

Os *gateways* na camada de comunicação são as entidades responsáveis pela interconexão entre os equipamentos em camada de sensoriamento e a RMAA. Cada *gateway* utilizado deve atender aos requisitos da rede de sensores correspondente e prover interfaces padronizadas para acesso aos serviços fornecidos pela rede.

Dois tipos diferentes de *gateways* foram utilizados no estudo de caso. Um *gateway* ZigBee/Ethernet e outro para comunicação homogênea entre a aplicação e o *Telemetric*. A próxima seção descreve os dois tipos utilizados.

4.4.2.1 *Gateway* da rede ZigBee

Um dos *gateways* utilizados para compor a camada de comunicação da arquitetura proposta foi o dispositivo *gateway* ZigBee/Ethernet ConnectPort X2[1] versão comercial. Trata-se de um dispositivo que permite a comunicação IP para redes de sensores ZigBee. Ele possui conexão Ethernet e é composto também por um rádio XBee para a comunicação sem fio via protocolo IEEE 802.15.4. Algumas das especificações do dispositivo são exibidas na tabela 4.3.

Tabela 4.3: Especificações do *gateway* ConnectPort X2.

Requisito	**Características**
Protocolos suportados	HTTP(S), TCP, UDP, DHCP, Telnet
Memória	8 MB de RAM e 890 KB de Flash
Segurança	Tunelamento SSL e autenticação HTTP, além das funções de segurança providas pela especificação ZigBee
Rádio	Módulo XBee/ZigBee com antena RPSMA, pré-configurado para atuar como coordenador da rede
Taxa de dados e modo de comunicação	Ethernet 10/100 Mbps (detecção automática); *Full* ou *Half duplex* (detecção automática)
Fonte de alimentação	12 VDC com entrada entre 9 e 20 VDC
Dimensões (C x L x A)	13.0 cm x 7.0 cm x 2.9 cm

O ConnectPort X2 é exibido na Figura 4.10.

O ConnectPort X2 possui interpretador da linguagem de programação Python (Python 2014) versão 2.4 embarcado. Dessa forma, é possível criar agentes capazes de manipular os

[1]Detalhes do produto podem ser encontrados em http://www.digi.com/products/wireless-routers-gateways/gateways/connectportx2gateways

Figura 4.10: *Gateway* ConnectPort X2.

dados provenientes das redes de sensores para tratamento no próprio *gateway*, sem que haja a necessidade de enviá-los para processamento no servidor na camada de aplicação, conservando recursos da rede. Essa característica permite definir políticas de tratamento de dados e propicia maior controle sobre o gerenciamento das informações.

Por se tratar de um equipamento de dimensões reduzidas, conforme exposto na Figura 4.10, este *gateway* pode ser facilmente posicionado em diversas localidades do ambiente. Apesar de possuir baixo poder computacional, ele é capaz de realizar, de maneira eficiente, suas funções como *gateway*, sendo uma alternativa mais barata se comparado à computadores pessoais dedicados à realização dessa função.

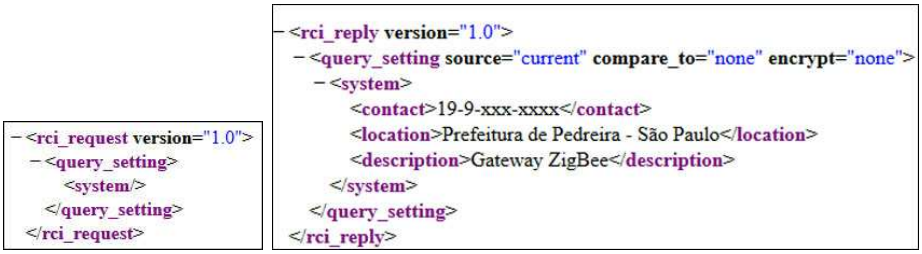
O ConnectPort X2 possui ainda um servidor HTTP embarcado e fornece *Web services* que podem ser invocados por clientes via protocolo específico para RPC baseado em XML. Conforme estudo realizado por Dissanaike, Wijkman & Wijkman (2004), XML-RPC se mostrou adequado para implementação em sistemas embarcados, comparado a abordagem utilizando SOAP, principalmente devido a complexidade e *overhead* causados por esse último.

Nesse sentido, o mecanismo utilizado para configurar, controlar e trocar informações com o ConnectPort X2, é o *Remote Command Interface* (RCI). Trata-se de um protocolo criado pela *Digi International Inc.* baseado em XML, para comunicação com seus dispositivos. Os métodos fornecidos pela interface RCI são utilizados para:

- inspecionar e alterar configurações de dispositivos;
- controlar e alterar o estado de dispositivos;
- reinicializar um equipamento;
- configurar redes ZigBee;
- manipular arquivos do sistema através de métodos PUT, GET e DELETE;
- enviar requisições e receber respostas de agentes registrados no dispositivo, tal como um programa em Python.

Como mencionado, as chamadas aos serviços fornecidos pelo *gateway* são realizadas via protocolo HTTP. Dessa maneira, a comunicação consiste de pares de solicitação e resposta baseados na troca de documentos XML. O método POST do HTTP é empregado para o encaminhamento das requisições. O processamento e a resposta do *gateway* são executados com base no corpo do arquivo de requisição, que contém a operação desejada e pode compreender parâmetros associados a ela.

As imagens apresentadas na Figura 4.11 são exemplos reais de requisição e resposta RCI manipuladas pelo ConnectPort X2 utilizado.



(a) Requisição RCI (b) Resposta RCI

Figura 4.11: Exemplos de *request-response* RCI.

Uma mensagem de requisição tem como nó raiz a *tag* <*rci_request*>. Em contrapartida, a resposta possui como nó raiz o elemento <*rci_reply*>. No exemplo da Figura 4.11 é requisitado ao *gateway* as informações do sistema, obtendo-se como resposta o local onde o *gateway* está implantando e o seu rótulo de identificação (*Gateway* ZigBee).

Conforme mencionado no capítulo 3, os *gateways* devem publicar para o nível de aplicação os serviços por ele disponibilizados e os parâmetros necessários, caso existam, para a manipulação das redes de sensores. No ConnectPort X2 essa interface é disponibilizada mediante a chamada RCI do método *query_descriptor*. A resposta à essa chamada pode ser vista na Figura 4.12.

O documento exposto na Figura 4.12 exibe todos os métodos suportados pelo dispositivo para a manipulação da rede ZigBee. Alguns dos métodos listados são:

- *query_setting*: para obtenção de informações do sistema (Figura 4.11);

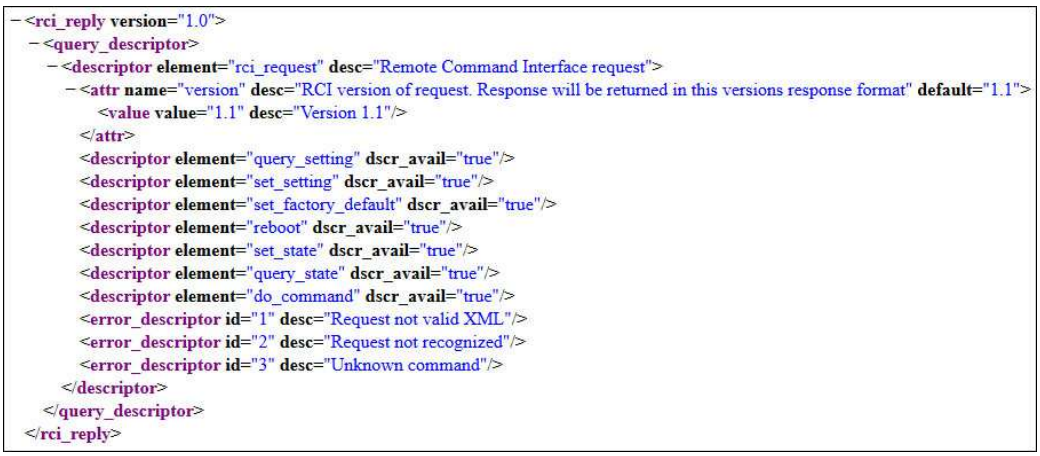


Figura 4.12: Interface *Web service* do ConnectPort.

- *set_setting*: para aplicação de alterações nas informações de sistema do *gateway*;
- *reboot*: para reinicialização do dispositivo *gateway*;
- *query_state*: para consulta de estado do nó XBee embarcado no *gateway*;
- *do_command*: inclui uma série de subcomandos para a manipulação dos nós na rede ZigBee.

Há também os códigos de erro e suas respectivas descrições. Essas informações são utilizadas para a criação dos serviços de manipulação da rede ZigBee na camada de aplicação da arquitetura.

Informações adicionais sobre cada um dos métodos suportados podem ser consultadas através da chamada ao método *query_descriptor*, juntamente com o(s) nome(s) do(s) método(s) desejado(s).

A Figura 4.13 exibe a descrição do serviço *query_state*.

```
- <rci_reply version="1.0">
  - <query_descriptor>
    - <descriptor element="query_state" desc="Request device current state">
      - <descriptor element="device_info" desc="Device info" access="read_only">
          <element name="mac" desc="MAC address" type="mac_addr"/>
          <element name="product" desc="Product name" type="string"/>
          <element name="model" desc="Model" type="string"/>
          <element name="productid" desc="Product ID" type="hex32"/>
          <element name="hardwarestrapping" desc="Hardware strapping" type="hex32"/>
          <element name="company" desc="Company" type="string"/>
          <element name="vendorid" desc="Vendor ID" type="hex32"/>
          <element name="boot" desc="Boot version" type="string"/>
          <element name="post" desc="POST version" type="string"/>
          <element name="firmware" desc="Firmware version" type="string"/>
          <element name="manufvpd" desc="Manufacturing VPD version" type="string"/>
          <element name="bootsoftrevstr" desc="Boot version number" type="string"/>
          <element name="postsoftrevstr" desc="POST version number" type="string"/>
          <element name="firmwaresoftrevstr" desc="Firmware version number" type="string"/>
          <element name="bootsoftrev" desc="Internal boot version" type="uint32"/>
          <element name="postsoftrev" desc="Internal post version" type="uint32"/>
          <element name="firmwaresoftrev" desc="Internal firmware version" type="uint32"/>
          <element name="firmwareid" desc="Firmware ID" type="uint32"/>
          <element name="digiproduct" desc="Digi product name" type="string"/>
        </descriptor>
```

Figura 4.13: Parte da descrição do serviço *query state*.

Em virtude da extensão do documento XML contendo a descrição do serviço *query_state*, somente parte dele é exposto na Figura 4.13. O documento contém informações sobre os parâmetros retornados, uma descrição textual sobre ele e também o tipo de dado retornado (inteiro, *string*, etc). Essas informações estão relacionadas ao *gateway*, incluindo o rádio XBee existente internamente.

Algumas das informações obtidas através da chamada ao serviço de consulta de estado são o endereço MAC do dispositivo, versão de *firmware* utilizado, o modelo e o nome do produto.

Devido a limitação de memória do *gateway* ConnectPort X2, as mensagens RCI de requisição devem conter no máximo 32KB de tamanho. Qualquer mensagem que exceda esse limiar deve ser dividida em múltiplas requisições RCI. No entanto, as respostas do ConnectPort não estão sujeitas a esse limite.

O *gateway* também disponibiliza informações específicas sobre os nós sensores da rede, necessitando apenas do endereço físico (MAC) do dispositivo ZigBee desejado. A Figura 4.14 mostra um exemplo de requisição de estado do nó ZigBee implantado no CPD e parte da resposta correspondente.

```
- <rci_request version="1.1">
  - <do_command target="zigbee">
      <query_state addr="00:13:a2:00:40:ac:6c:3b!"/>
    </do_command>
  </rci_request>
```

(a) Requisição de estado RCI.

```
- <rci_reply version="1.0">
  - <do_command target="zigbee">
    - <query_state addr="00:13:a2:00:40:ac:6c:3b!">
      - <radio>
          <pan_id>0xc634</pan_id>
          <ext_pan_id>0x000000000000aaaa</ext_pan_id>
          <channel>0x16</channel>
          <net_addr>0xcb0d</net_addr>
          <parent_addr>0xfffe</parent_addr>
          <association>0x0</association>
          <firmware_version>0x22a7</firmware_version>
          <hardware_version>0x194b</hardware_version>
          <device_type>0x30000</device_type>
          <children>12</children>
          <max_payload>84</max_payload>
          <supply_voltage>3258</supply_voltage>
          <rssi>44</rssi>
          <tx_power>3</tx_power>
          <ack_failures>0</ack_failures>
        </radio>
      </query_state>
    </do_command>
  </rci_reply>
```

(b) Resposta de estado RCI.

Figura 4.14: Requisição e resposta de estado de um rádio XBee.

O documento de resposta à consulta de estado do nó compreende informações da rede, como o endereço à qual ele pertence e o identificador (*net_addr*) do nó na rede. Estão presentes também informações de associação, sobre a tensão de alimentação, erros de reconhecimento e indicador de potência do sinal recebido (RSSI). Essas informações correspondem ao estado do nó no momento da consulta. Dessa maneira, elas podem variar a cada nova requisição.

O estado dos dispositivos sensores, bem como as suas configurações são armazenadas em banco de dados no servidor da camada de aplicação. Assim é possível montar um histórico de movimentação de nós, registrar períodos de inatividade, verificar horários de estabelecimento e perda de conexão, e assim por diante.

Embora o ConnectPort X2 forneça diversos serviços para monitorar e manipular os nós da rede ZigBee, não há um serviço específico para coletar os dados dos terminais sensores. Assim sendo, com o intuito de padronizar o formato de envio dos dados coletados pela rede ZigBee para a camada de aplicação, um agente foi desenvolvido em Python e embarcado no ConnectPort. À medida que os sensores transmitem os dados coletados para o *sink* da rede, no caso o próprio *gateway*, o agente é o responsável por ler essas informações e encapsular os dados em um pacote XML para envio ao servidor de aplicação. Para definir o formato das mensagens enviadas, um esquema XML foi criado. O formato do arquivo é apresentado na Figura 4.15.

```
- <sensor:schema attributeFormDefault="unqualified" elementFormDefault="qualified">
  - <sensor:element name="dadoColetado">
    - <sensor:complexType>
      - <sensor:sequence>
          <sensor:element type="sensor:string" name="dataHora"/>
          <sensor:element type="sensor:string" name="macDispositivo"/>
          <sensor:element type="sensor:string" name="sensor"/>
          <sensor:element type="sensor:string" name="valor"/>
        </sensor:sequence>
      </sensor:complexType>
    </sensor:element>
  </sensor:schema>
```

Figura 4.15: XML *Schema* para os dados coletados.

O uso do XML *Schema* é importante para padronizar o formato dos dados coletados para envio à camada de Aplicação. O arquivo exposto na Figura 4.15 define a hierarquia das informações, os campos que o arquivo XML deve conter, bem como a sua ordenação, e também os tipos de dados de cada um deles. Um exemplo de dado coletado encapsulado em XML é exposto na imagem 4.16.

```
- <dadoColetado>
    <dataHora>2014-04-20 12:00:22</dataHora>
    <macDispositivo>00:13:a2:00:40:ac:6c:3b!</macDispositivo>
    <sensor>Temperatura</sensor>
    <valor>16.00</valor>
  </dadoColetado>
```

Figura 4.16: Arquivo XML de um dado coletado.

Os dados coletados pelos terminais sensores, ambos ZigBee e Telemetric, são empacotados segundo o formato da Figura 4.16 e enviados para o servidor de aplicação em intervalos de cinco minutos. O arquivo contém o nome do sensor e o endereço único (MAC) do dispositivo responsável pela coleta. Um campo importante no arquivo criado é o de data e hora da coleta. Ele funciona como um carimbo de tempo, correspondente ao momento em que a informação foi recebida pelo agente no *gateway* via comunicação ZigBee. Isso porque podem ocorrer atrasos na transmissão da informação para o servidor de aplicação via RMAA. Como os dados provenientes de sensoriamento podem ser sensíveis ao atraso e latência, a informação contida nesse campo garante maior precisão e confiabilidade e permite registrar os dados de acordo com o horário da coleta.

4.4.2.2 *Gateway* da rede IP

Para o *gateway* correspondente ao sensoriamento com o *Telemetric*, desenvolvemos um componente de software no próprio servidor de aplicação, conforme exposto na Figura 4.2.

O *Telemetric*, descrito na seção 4.3.2, é um equipamento que pode ser gerenciado via protocolo *Simple Network Management Protocol* (SNMP) ou através de sua interface *Web*. Entretanto, ele também gera dados em formato XML conforme padrão proprietário do dispositivo.

Dessa maneira, desenvolvemos um módulo de software em linguagem Python para funcionar como *gateway* para o *Telemetric*. Um *script Commom Gateway Interface* (CGI) foi desenvolvido para processar as requisições HTTP provenientes de clientes em nível de aplicação e também para

manipular os dados coletados pelo *Telemetric*. O *script* Python se comunica com o *Telemetric* via HTTP, trocando mensagens formatadas em XML com o dispositivo. Assim, ele encapsula os dados do *Telemetric* em um formato compreensível para a camada de aplicação.

A Figura 4.17 exibe a comunicação do *gateway* em Python com o *Telemetric* e com a camada de aplicação.

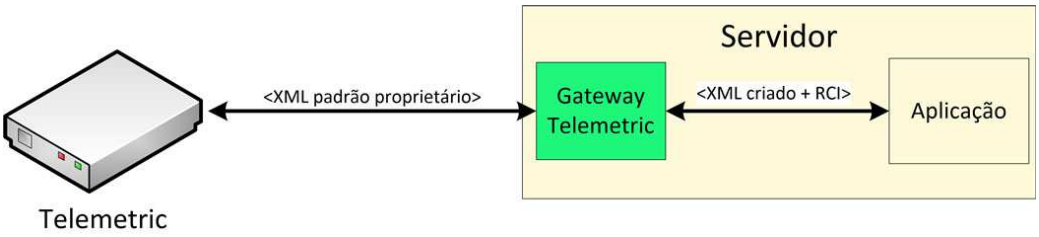


Figura 4.17: Comunicação do *Telemetric* com a camada de aplicação através do *Gateway* em Python.

Como pode ser visto na Figura 4.17 o agente Python fornece interface XML-RPC baseada no protocolo RCI. Escolhemos essa interface pelo fato de ela estar disponível no *gateway* ZigBee utilizado. Desse modo, a comunicação com os diferentes *gateways* ocorre de maneira homogênea.

Os dados coletados pelo dispositivo são enviados para o nível de aplicação seguindo o padrão definido no esquema XML da Figura 4.15.

Sendo assim, o módulo Python foi instalado no próprio servidor de aplicação, para diminuir os custos do emprego de um computador dedicado somente à realização dessa tarefa. A interface criada baseada em XML-RPC permitiu monitorar as diferentes redes, ZigBee e IP (Telemetric), de maneira homogênea, sem a necessidade de adaptação de código em nível de aplicação.

4.5 Camada de Aplicação

Essa seção descreve os detalhes do desenvolvimento dos componentes pertinentes a camada de Aplicação da arquitetura. O *back-end* do sistema, o portal de monitoramento e a modelagem do banco de dados são expostos nesta seção.

4.5.1 Aplicação servidora

Conforme mencionado na seção 3.1.1.2, o desenvolvimento do portal, que compreende o servidor de aplicação e a interface do sistema, considerou o emprego do padrão MVC para separar a lógica de negócio do nível de apresentação dos dados.

Optamos pela arquitetura MVC por esta permitir a modularidade de código, o que torna a aplicação mais flexível e escalável. Essa separação da interface de usuário do controle de fluxo de dados e da lógica de negócios, simplifica a implementação e aumenta a reutilização de código.

O servidor corresponde a camada de aplicação da arquitetura. Diferentes linguagens de programação podem ser utilizadas para conceber esse nível. No estudo de caso, desenvolvemos esse componente em linguagem *HyperText Preprocessor* (PHP) versão 5.4 (PHP 2014). O PHP foi utilizado para estabelecer conexão com o serviço de banco de dados e para gerar

informações de apresentação dinamicamente. Como servidor HTTP foi utilizada a versão 2.4.4 do *Apache*. O *Apache* é um projeto de código aberto, sendo um dos mais populares e mais utilizados mundialmente para servir páginas *Web* (Apache 2014). Vale mencionar que ambos, PHP e Apache, operam em sistemas operacionais Linux e Windows, além de outros baseados em UNIX, como o OS X.

Para constituir o módulo de armazenamento e centralização de informações, o MySQL (MySQL 2014) foi empregado como Sistema de Gerenciamento de Banco de Dados (SGBD). O MySQL trata-se de um SGBD bastante difundido e utilizado para suportar aplicações de pequeno e médio porte, sendo capaz de persistir milhões de registros em bancos de dados relacionais. O banco de dados criado na camada de aplicação representa a base de conhecimento da arquitetura, responsável por armazenar os dados coletados pelas redes de sensores, as informações de configuração dessas, dos locais monitorados e também dos administradores cadastrados. É importante mencionar que todas as ferramentas utilizadas no estudo de caso possuem licença comercial gratuita.

Persistir as informações em banco de dados permite realizar análises mais rebuscadas e sob demanda, além de possibilitar a criação de históricos para fins de comparações entre diferentes intervalos de tempo.

A Figura 4.18 exibe o diagrama entidade relacionamento (DER) do banco de dados desenvolvido para o sistema de monitoramento.

Os atributos e os relacionamentos definidos no modelo de domínio da Figura 4.3 são uteis para modelar parte das entidades que compõem o banco de dados, como pode ser visto na Figura 4.18. Os registros armazenados em banco de dados são transformados em informação para análise dos usuários por meio do PHP.

A aplicação servidora foi construída a partir do emprego do *framework* PHP Zend. Trata-se de um *framework* orientado a objetos, de código aberto, para o desenvolvimento de aplicações *Web* e serviços usando PHP versão 5.3 ou superior (Zend 2014). A arquitetura do Zend possui baixo acoplamento, permitindo a utilização dos seus componentes de maneira independente. Ele permite também organizar o código da aplicação de acordo com o padrão MVC, separando códigos de lógica de negócio do nível de apresentação da informação. O Zend fornece também mecanismos de abstração de banco de dados, tal que a aplicação suporte diferentes SGBD.

A Figura 4.19 exibe a organização MVC do projeto criado com o Zend *framework*.

A estrutura de um projeto criado com o Zend permite melhorar a organização do código. Sua natureza modular também possibilita a utilização de componentes de maneira independente.

Além de promover melhorias na qualidade do código, o emprego de um *framework* fornece métodos pré-construídos para a criação de *Web services*, agilizando o processo de desenvolvimento.

4.5.1.1 API *Web service*

Uma das principais características da arquitetura é a criação de uma base central de conhecimento e o compartilhamento de informações de sensoriamento. Esse aspecto permite que sistemas externos obtenham dados de diferentes domínios de modo padronizado, possibilitando a criação de novas aplicações. Assim, esse módulo consiste de um *Web service* baseado em

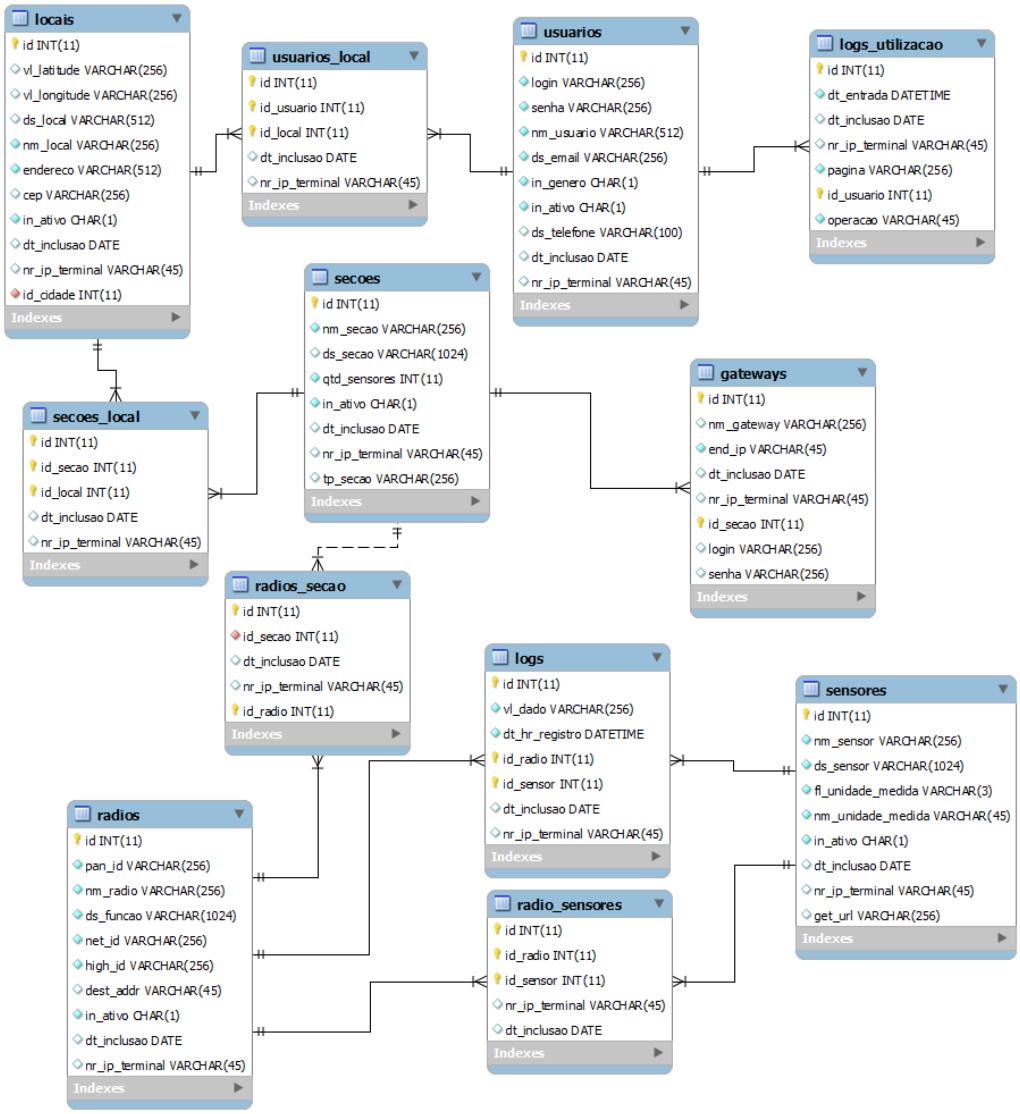


Figura 4.18: Modelo entidade relacionamento do sistema desenvolvido.

REST para acesso aos dados coletados pelas redes de sensores.

O *framework* Zend utilizado possui classes pré-construídas que facilitam a criação de *Web services*. Essas classes foram empregadas para fornecer dados aos sistemas clientes em formatos JSON e XML. Uma interface REST foi criada para padronizar o acesso as informações. A Tabela 4.4 exibe os parâmetros e as respectivas descrições da API desenvolvida.

A interface criada é bastante simples e não tem por objetivo definir um padrão a ser seguido. Fica à cargo do setor de tecnologia do município a definição dessa interface.

Conforme já mencionado, a API REST, juntamente com os locais que disponibilizam seus dados via *Web service*, podem ser publicadas no portal *Web* do município, para divulgação aos sistemas externos.

Os dados das redes de sensores podem ser obtidos em formatos JSON e XML através da interface uniforme do REST, por meio dos métodos GET e POST do HTTP. O serviço de compartilhamento é identificado unicamente por uma URL, que retorna diferentes recursos de

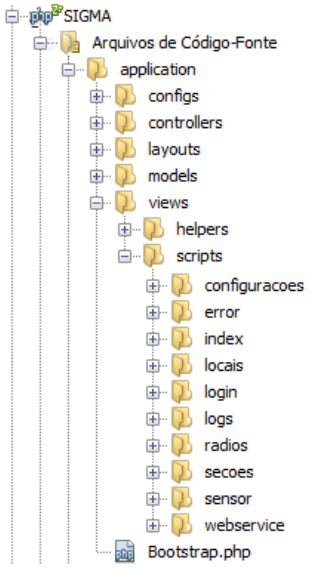


Figura 4.19: Organização do código no padrão MVC.

acordo com o recebimento dos parâmetros descritos na tabela 4.4.

A Figura 4.20 exibe o formato da URL para acesso ao serviço de compartilhamento de dados.

```
http://10.19.11.199/rest/webservices/servico/buscardados/local/{nome_local}/secao/{nome_secao}/sensor/{nome_sensor}/dtInicial/{ano-mes-dia}/dtFinal/{ano-mes-dia}/formato/{xml ou json}
```

Figura 4.20: URL de compartilhamento de dados via *Web service*.

A URL exibida na Figura 4.20 corresponde ao endereço do servidor em Pedreira a partir do qual os dados coletados pelos sensores poderiam ser obtidos por aplicações de software cliente HTTP, como um navegador *Web*.

Uma das vantagens da aplicação do estilo REST para a criação de *Web services*, conforme descrito no capítulo 2, é o fornecimento de recursos (dados) em diferentes formatos para os clientes, enquanto que serviços *Web* tradicionais baseados em SOAP consistem somente do uso de XML.

A Figura 4.21 exibe a parte da resposta do *Web service* de consulta de dados coletados pelos sensores de temperatura implantados na prefeitura no dia 01 de abril de 2014, formatados em XML e em JSON. Essa consulta foi realizada através do navegador pelo endereço especificado na Figura 4.20.

Tabela 4.4: API REST criada.

Parâmetro	**Tipo**	**Descrição**
local	*string*	O nome do local de onde se deseja obter os dados. Pode ser informado somente parte do nome.
secao	*string*	A seção do local especificado, de onde se deseja obter os dados. Pode ser informado somente parte do nome.
sensor	*string*	Sensor responsável pela coleta dos dados de interesse. Pode ser informado somente parte do nome.
dtInicial	*date*	Data inicial do período de tempo desejado. O formato desse parâmetro é o americano, ou seja, ano-mês-dia (yyyy-mm-dd) separados por hífen (-).
dtFinal	*date*	Data final do período de tempo desejado. O formato desse parâmetro é o americano, ou seja, ano-mês-dia (yyyy-mm-dd) separados por hífen (-).
formato	*string*	Corresponde ao formato de dados que se deseja obter como resposta. Estão disponíveis o XML e o JSON.

(a) Resposta em formato XML (b) Resposta em formato JSON

Figura 4.21: Dados reportados em formatos distintos.

A resposta do *Web service* inclui a hora exata em que o dado foi coletado e recebido pelo *gateway*, para transmissão à de aplicação.

Compartilhar dados em diferentes formatos permite que sistemas externos optem pela alternativa que seja mais adequada às suas necessidades. É importante também definir um modelo de dados para estruturar as informações compartilhadas com sistemas externos. Vale salientar que os formatos disponibilizados são suportados pela grande maioria das linguagens de programação.

4.5.2 Portal *Web*

O portal *Web* fornece a interface gráfica de interação do usuário com o sistema. Conforme mencionado no capítulo 3, a única imposição da arquitetura para acesso ao sistema por parte dos administradores é a utilização de um navegador HTML. Atualmente, diferentes categorias de dispositivos podem ser utilizados para navegar na *Web*, tais como: computadores pessoais; *tablets* e *smartphones*. Pensando nisso, para atender as diferentes categorias sem a necessidade de desenvolver aplicações específicas de plataforma, aplicamos o conceito de *Responsive Web Design* na concepção do portal.

Trata-se de um conceito relativamente novo mas que está sendo amplamente utilizado para atingir um número cada vez maior de dispositivos. As informações são estruturadas através da versão cinco do HTML (HTML5) e formatadas por arquivos *Cascading Style Sheets*, no caso o CSS3. Segundo Mohorovicic (2013), os recursos-chave do *Responsive Web Design* são: grade flexível, consistindo de múltiplas colunas expressas em tamanhos relativos (por exemplo, em porcentagem); imagens flexíveis, que escalam de acordo com o tamanho do *container*; e *media queries*, para construir múltiplos *layouts* utilizando o mesmo documento HTML. *Media queries* são declarações condicionais que podem identificar os tipos de *media* (tela) e navegadores e permitir que a melhor configuração seja aplicada de acordo com essas informações.

O portal desenvolvido é a interface central para monitoramento das redes de sensores do município. Para ter acesso a elas os administradores devem se autenticar no sistema mediante o uso de credenciais baseadas em *login* e senha.

A Figura 4.22 exibe a tela de autenticação do usuário otimizada para apresentação em um *smartphone*.

Figura 4.22: Autenticação no portal via *smartphone*.

O formulário de autenticação se adapta a resolução de tela do dispositivo, no caso de 320x568 *pixels*, distribuindo as informações de modo eficiente.

O acesso ao portal desenvolvido é realizado via Intranet, somente dentro da rede metropo-

litana de Pedreira. Esse requisito foi atendido para prover maior segurança no acesso as redes. No entanto, uma Rede Privada Virtual (VPN) pode ser criada para permitir o acesso externo via Internet.

Após realizar a autenticação no portal, o administrador é direcionado para uma tela contendo os últimos dados coletados pelas redes de sensores.

Na área restrita do portal são listados os locais aos quais o administrador possui privilégios de monitoramento.

A Figura 4.23 exibe informações sobre a prefeitura de Pedreira.

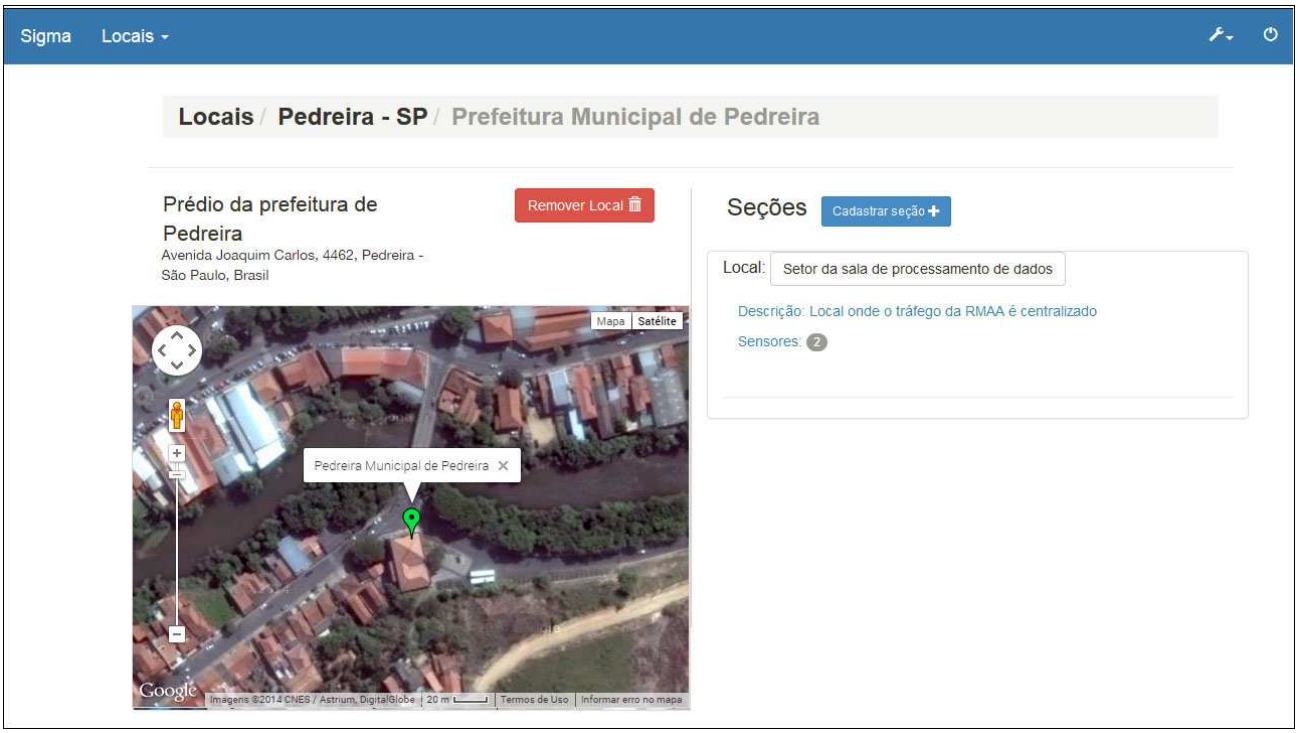


Figura 4.23: Seção monitorada na prefeitura de Pedreira.

A interface exposta na Figura 4.23 contém dados de localização do ambiente selecionado, no caso a prefeitura de Pedreira, bem como quais seções estão sendo monitoradas e a quantidade de sensores existentes. Como podemos observar, o administrador possui acesso a seção do Centro de Processamento de Dados (CPD). Por meio dessa interface ele pode obter informações detalhadas sobre a rede implantada, gerenciar demais localidades, isto é, editar informações, excluir ou cadastrar um novo local e também novas seções. O menu contém ainda opções para alterar as credenciais de acesso.

Ao selecionar uma das seções monitoradas na prefeitura o administrador é direcionado para a interface exposta na Figura 4.24.

A interface da Figura 4.24 exibe informações sobre a rede ZigBee implantada no prédio da prefeitura. O administrador pode visualizar os nós existentes na rede bem como o estado de cada um deles, ou seja, se estão ativos ou não. Visto que o *gateway* ZigBee fornece serviço de descoberta de nós, o administrador pode executar esta ação por meio do botão "Pesquisar rádios". Com isso ele obtém informações sobre novos nós que tenham se juntado a rede e que

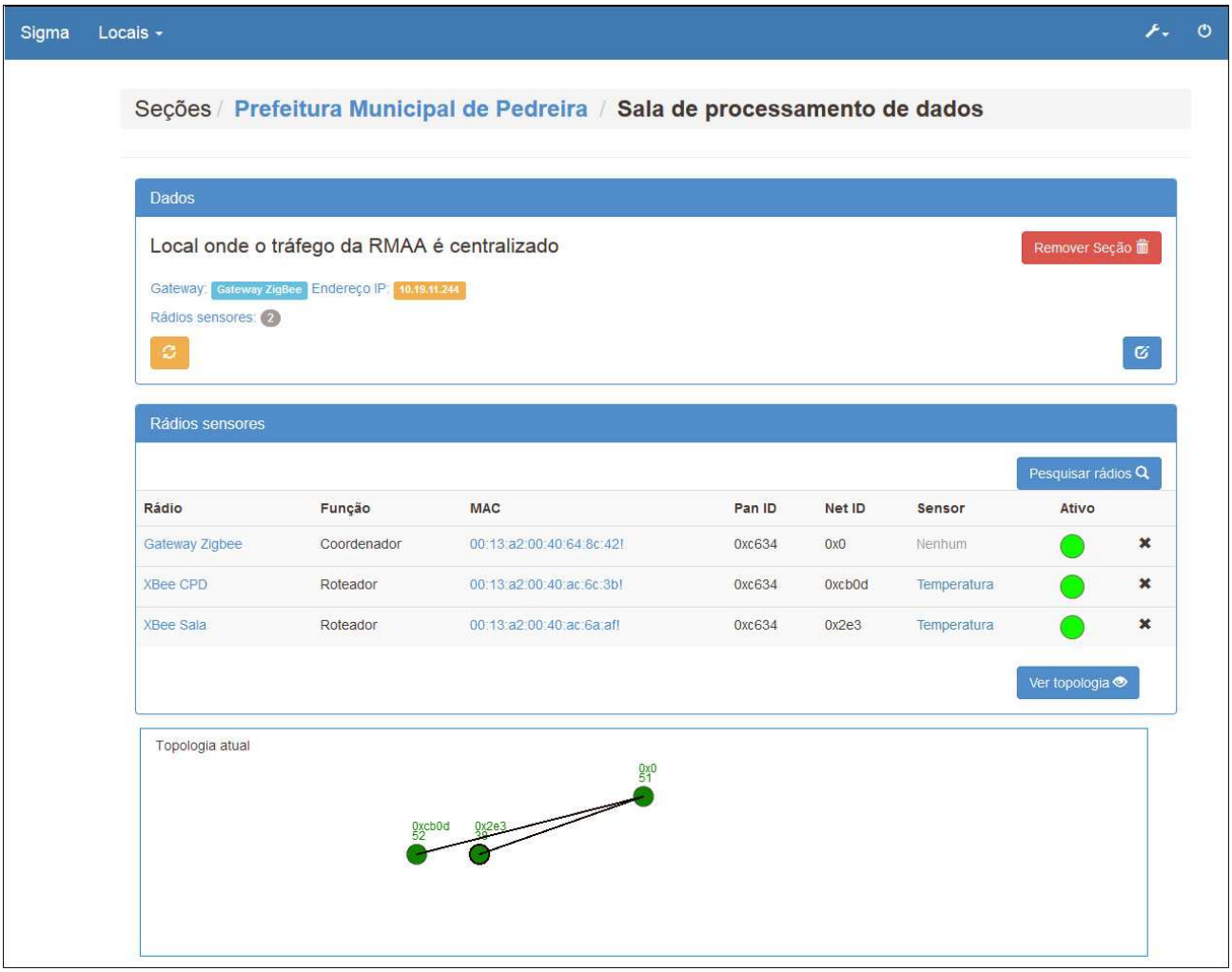


Figura 4.24: Rádios da rede ZigBee implantada na prefeitura.

não foram cadastrados no sistema.

A topologia da rede também é apresentada em tempo real. Na Figura 4.24 podemos observar a topologia em estrela formada por dois terminais sensores e pelo nó sorvedouro. Com a visualização topológica da rede os administradores podem identificar o caminho percorrido pelos dados para transmissão ao nó sorvedouro e verificar se há sobrecarga de tráfego direcionado a um único nó.

Certas funcionalidades dessa interface estão acopladas ao *gateway* em nível de comunicação, tais como: descoberta de novos nós; alteração de configurações; e consulta de estado de dispositivos.

Ao selecionar um dos nós componentes da rede, por exemplo o XBee CPD, a aplicação realiza uma chamada RPC codificada em XML ao *gateway* para obter informações de configuração e estado do nó selecionado.

Na Figura 4.25 é apresentada parte dos dados de configuração e estado do nó ZigBee XBee CPD, reportada pelo portal *Web*.

As informações de configuração do nó sensor XBee CPD, bem como o seu estado atual, são transmitidas pelo *gateway* em nível de comunicação para o servidor na camada de aplicação em

Figura 4.25: Configurações e estado do nó ZigBee XBee CPD implantado.

formato de arquivo XML. A aplicação então analisa o arquivo e o converte para apresentação em formato HTML. Dessa maneira os dados expostos na interface da Figura 4.25 variam de acordo com o dispositivo utilizado na camada de sensoriamento. Para o nó ZigBee selecionado as informações obtidas incluem a configuração dos pinos digitais e endereço do nó de destino dos dados, ou seja, para qual nó os dados coletados devem ser enviados. Com relação as informações de estado do dispositivo no momento da consulta (coluna direta da Figura 4.25), estas incluem o endereço da rede a qual ele pertence, o seu endereço na rede, a versão do *firmware* do dispositivo e tensão de alimentação.

Ainda a partir da interface da Figura 4.25 é possível promover modificações nas configurações do nó. Os parâmetros alteráveis incluem endereçamento do nó na rede, endereço de nó destino, para o caso de uma rede multi-salto, rótulo de identificação do nó e assim por diante. Ao solicitar essas modificações a aplicação empacota as novas configurações em formato XML e envia para o serviço de alteração de configuração fornecido pelo *gateway*. Essas alterações são então registradas em banco de dados na camada de aplicação visando a criação de um histórico de modificações para fins de controle e auditoria.

Por meio da interface exposta na Figura 4.24 é possível ainda realizar modificações nas configurações do *gateway*.

A Figura 4.26 expõe a tela de edição de configurações do *gateway* ConnectPort da rede ZigBee.

A identificação do *gateway* e o seu endereçamento na RMAA, bem como as credenciais utilizadas para autenticação HTTP, são configuradas a partir da interface da Figura 4.26.

Um dos módulos principais do portal *Web* é o de apresentação dos dados coletados pelas

Figura 4.26: Configurações do *gateway* ZigBee.

redes de sensores. No portal desenvolvido esta apresentação é realizada através de gráficos estatísticos, a partir de um intervalo de tempo definido pelo administrador da rede.

A Figura 4.27 exibe a interface de apresentação dos dados coletados por um sensor de temperatura.

O administrador pode realizar alterações nas informações do sensor, como variável monitorada e unidade de medida, a partir da interface da Figura 4.27. Os valores coletados e as informações de data e hora da coleta são expostos graficamente. A visualização pode ser alternada entre os gráficos de linhas e de colunas. Essas informações podem ainda ser exportadas em formato *Comma-separated Values* (CSV), para que sejam realizadas análises mais rebuscadas em softwares específicos.

O compartilhamento dos dados é habilitado a partir da interface da Figura 4.27. Os administradores podem permitir ou negar o acesso de sistemas externos aos dados de determinados sensores.

Para a geração dos gráficos estatísticos dos dados coletados pela rede de sensores utilizamos uma biblioteca JavaScript. Com isso a criação de gráficos é feita de maneira dinâmica a partir das informações fornecidas pelo usuário. Em virtude da natureza modular do portal, novas formas de visualização dos dados, como em tabelas, podem ser facilmente adicionadas.

A partir do portal desenvolvido o administrador é capaz de monitorar tanto a rede ZigBee quanto o *Telemetric* de modo transparente, ou seja, utilizando a mesma interface de interação. A Figura 4.28 exibe informações do *Telemetric* monitorado no morro do Cristo.

Como pode ser observado, diferente dos nós da rede ZigBee que possuíam somente um sensor cada um, o *Telemetric* possui seis sensores embarcados, sendo eles: temperatura, estado de três tomadas, tensão AC e tensão DC.

Em virtude dos meios padronizados para comunicação entre a camada de aplicação e de sensoriamento por meio dos dispositivos *gateways*, as duas redes podem ser monitoradas de

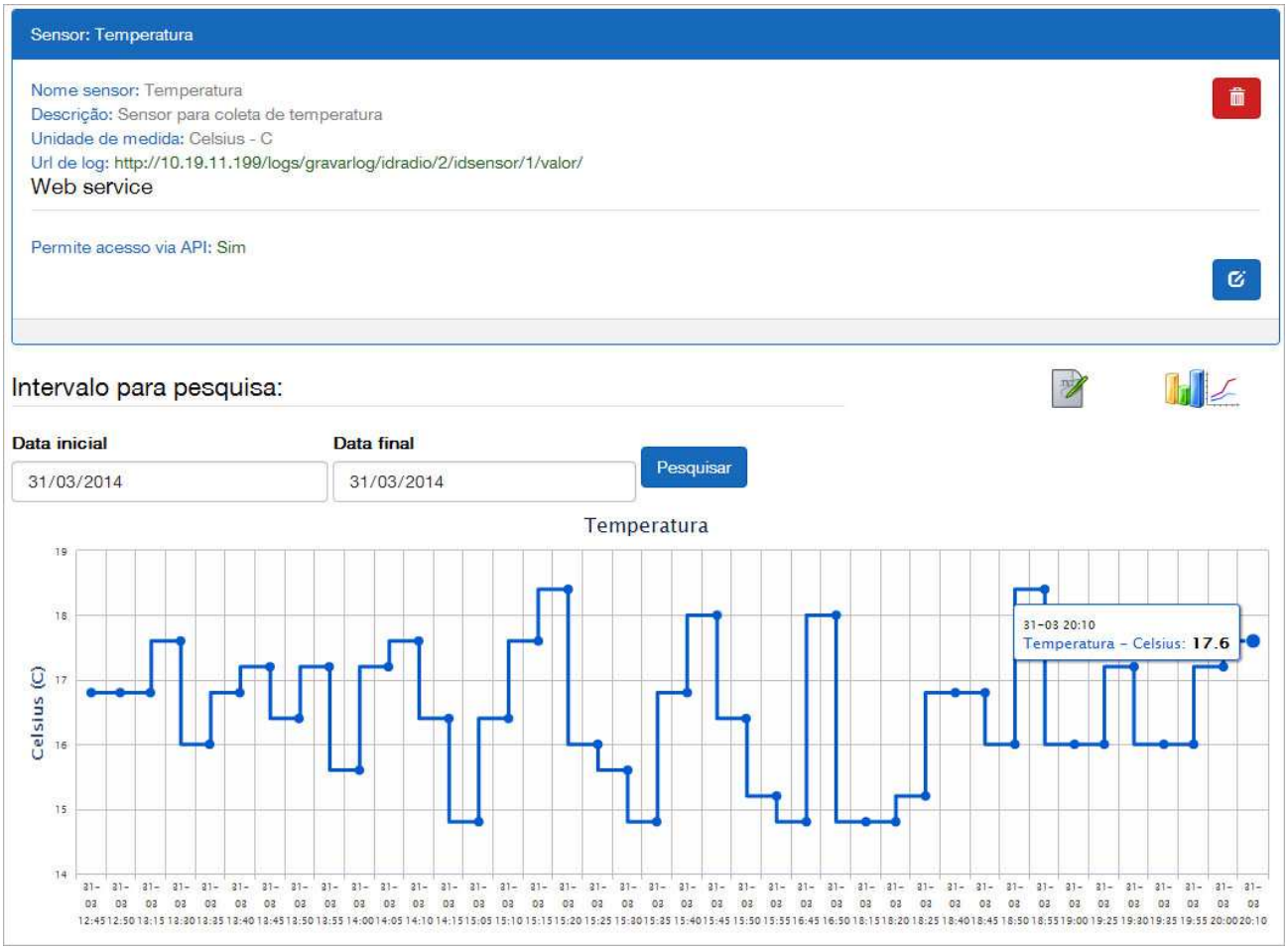


Figura 4.27: Módulo de visualização de dados.

modo unificado. Entretanto, as ações que podem ser realizadas na camada de aplicação variam de acordo com o dispositivo monitorado. Por exemplo, o *gateway* do *Telemetric* não fornece o serviço de descoberta de nós, portanto essa função é desabilitada nessa interface.

4.6 Segurança

A segurança da informação na RMAA é uma questão que merece substancial atenção. Conforme exposto na seção 3.3, os dados que trafegam na Infovia possuem informações sigilosas da administração pública e dos munícipes. Sendo assim, diferentes estratégias foram apresentadas para promover a segurança nas três camadas da arquitetura proposta.

Algumas das abordagens descritas no capítulo 3 foram empregadas no estudo de caso. Uma delas, descrita em Miani (2009) para promover a segurança da informação em RMAA, é a criação de sub-redes para segregar o tráfego de diferentes domínios. Essa abordagem foi aplicada no estudo de caso com o auxílio de VLAN's em *switches* da rede, conforme exposto na Figura 4.29.

Uma VLAN específica foi utilizada para segregar o tráfego das redes de sensores implantadas, isto é, da rede ZigBee e do *Telemetric*. Dessa maneira, as informações não sofreram influências de outras fontes de tráfego, agregando maior confiabilidade e garantindo a integridade dos dados

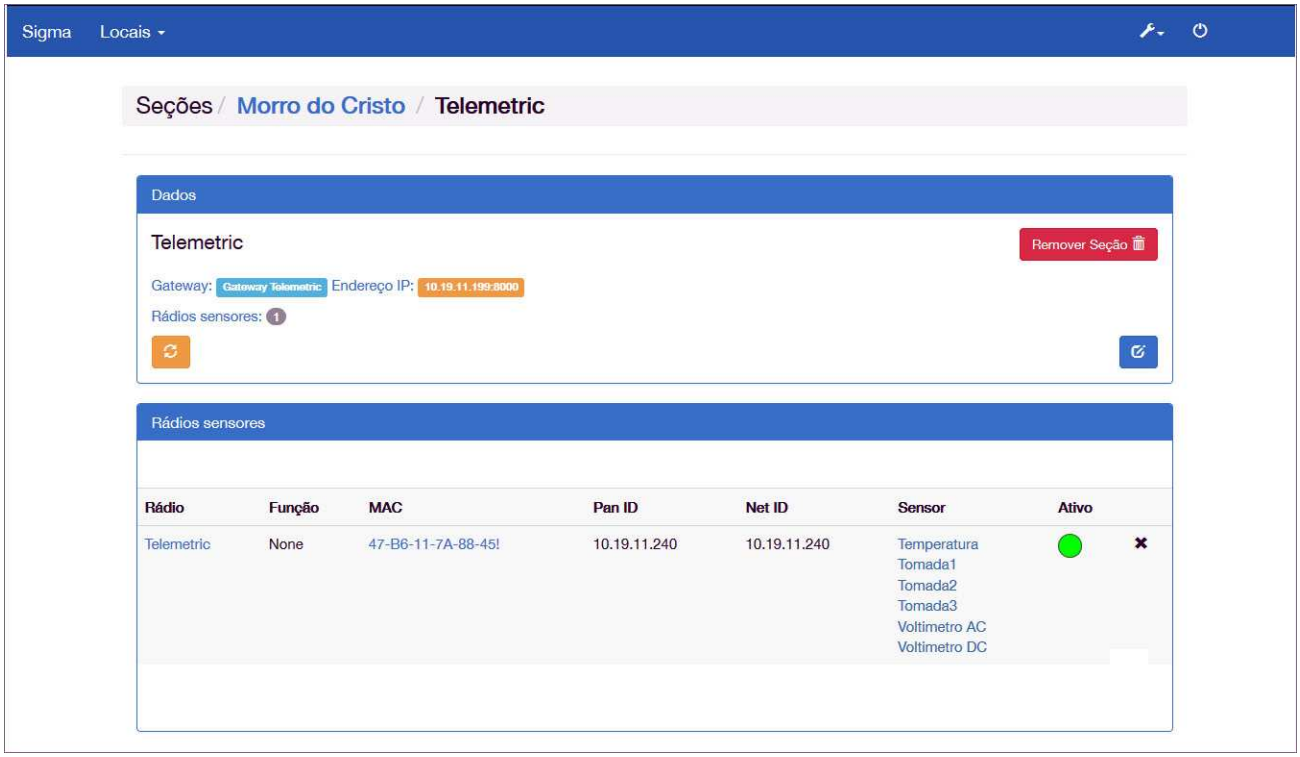


Figura 4.28: *Telemetric* monitorado no Morro do Cristo

coletados.

Na camada de aplicação a estratégia de segurança aplicada foi a de acesso ao portal de monitoramento limitado aos administradores autenticados no sistema mediante o fornecimento de credenciais baseadas em usuário e senha. Para aumentar o nível de segurança os dados de acesso foram criptografados utilizando o algoritmo de *hash* MD5 e armazenados na base central de dados.

Na camada de comunicação empregamos a abordagem de autenticação HTTP (*BASIC-AUTH*) para acesso aos *gateways*, descrita na seção 3.3.2. Ambos os *gateways* utilizados implementaram esse método de autenticação e autorização para acesso aos serviços fornecidos pelas redes de sensores. Esse método evita a conexão e manipulação indevida das informações providas pela camada de sensoriamento. Uma das principais vantagens da utilização de autenticação HTTP é a sua simplicidade de configuração e implementação.

Na camada de sensoriamento a abordagem de segurança empregada consistiu da proteção dos nós ZigBee contra danos físicos causados pelo meio. Dessa forma, os dispositivos foram posicionados em pontos que nós consideramos seguros e menos suscetíveis à danos naturais, como quedas e contato com a água.

Não foram aplicados métodos para a segurança da informação em nível de sensoriamento em virtude de os dados coletados não apresentarem características de sigilo e confidencialidade.

Em sistemas sensores de escala urbana é importante identificar vulnerabilidades de segurança da rede e propor métodos para o controle de falhas. Como pode ser observado, o caráter multisserviço da RMAA exibe numerosos desafios para os analistas de segurança.

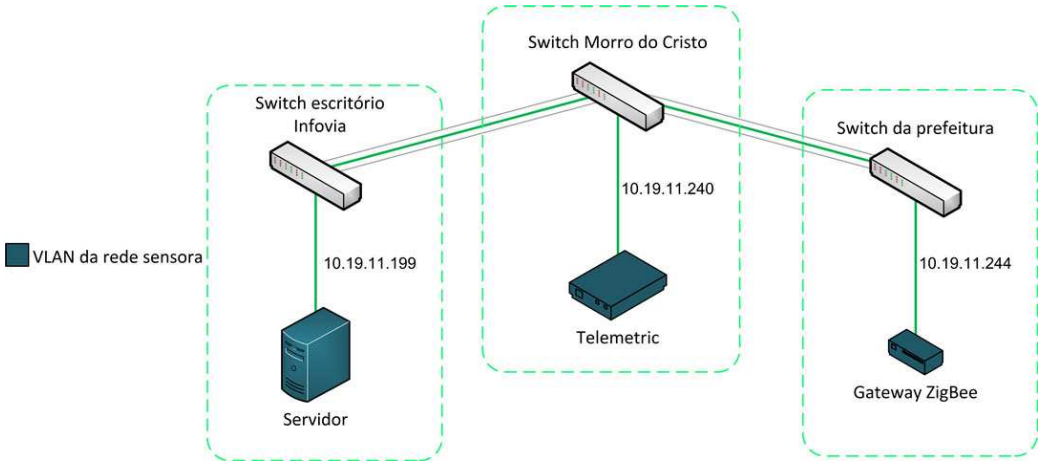


Figura 4.29: VLAN utilizada no estudo de caso.

4.7 Resultados do monitoramento

O sistema de monitoramento desenvolvido nos moldes da arquitetura proposta foi implantado na cidade de Pedreira no dia 27 de março de 2014 e permaneceu em funcionamento até o final de maio do mesmo ano. Durante esse período os dados coletados pela camada de sensoriamento alimentaram a base central de conhecimento na camada de aplicação. Os resultados desse monitoramento divididos de acordo com a tecnologia empregada são apresentados nas próximas seções. Os gráficos e as estatísticas expostas foram gerados manualmente com as informações armazenadas no banco de dados utilizando-se a ferramenta de análise Microsoft Office Excel 2007.

Além de realizar a coleta de dados nos ambientes de interesse, monitoramos também a comunicação *wireless* da rede ZigBee bem como a tensão de alimentação dos nós sensores, com o intuito de identificar níveis de interferência eletromagnética na rede e variações na voltagem de energia. Os resultados do monitoramento com o *Telemetric* também são expostos.

4.7.1 Monitoramento da rede ZigBee

Os sensores da rede ZigBee descritos na seção 4.3.1 foram dispostos em locais diferentes na prefeitura municipal de Pedreira. O nó equipado com o sensor de temperatura LM61 foi posicionado no Centro de Processamento de Dados (CPD), onde o tráfego da RMAA é centralizado e onde encontram-se os equipamentos servidores, *switches* e roteadores. Nessa sala também foi instalado o *gateway* ConnectPort X2, conectando-se a um dos *switches* da rede. O dispositivo equivalente ao sensor LM35 foi disposto em uma sala comum do prédio próxima ao CPD.

Após a instalação da rede de sensores na prefeitura e da implantação da aplicação servidora, os administradores puderam fazer o acompanhamento em tempo real das informações coletadas pelos sensores e monitorar a rede ZigBee. O mesmo ocorreu com sistemas externos, capazes de obter os dados também em tempo real.

Por não se tratar de dados altamente dinâmicos, isto é, que apresentam grandes variações frequentemente, definimos um intervalo de cinco minutos para o envio de amostras de tempera-

tura dos ambientes monitorados para a camada de aplicação. Durante o período de realização do estudo de caso, esses dados foram coletados e armazenados no banco de dados criado na camada de aplicação.

A Figura 4.30 exibe as médias diárias de temperatura dos dois sensores utilizados durante aproximadamente duas semanas entre os meses de março e abril.

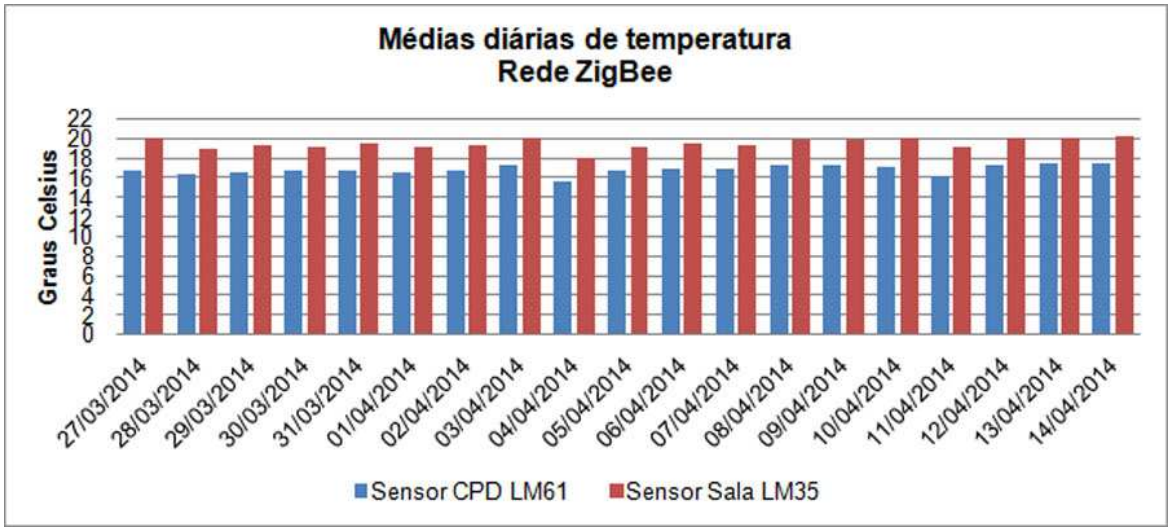


Figura 4.30: Médias diárias de temperatura coletadas durante duas semanas.

Observando a Figura 4.30 é possível notar que o nó equivalente ao sensor LM35 coletou temperaturas mais elevadas que o sensor LM61. Isso pode ser pelo fato de a sala do CPD ser equipada com aparelhos de ar-condicionado operando em temperaturas mais baixas em virtude dos equipamentos existentes no local.

A Figura 4.31 apresenta os valores médios de temperatura em intervalos de quatro horas.

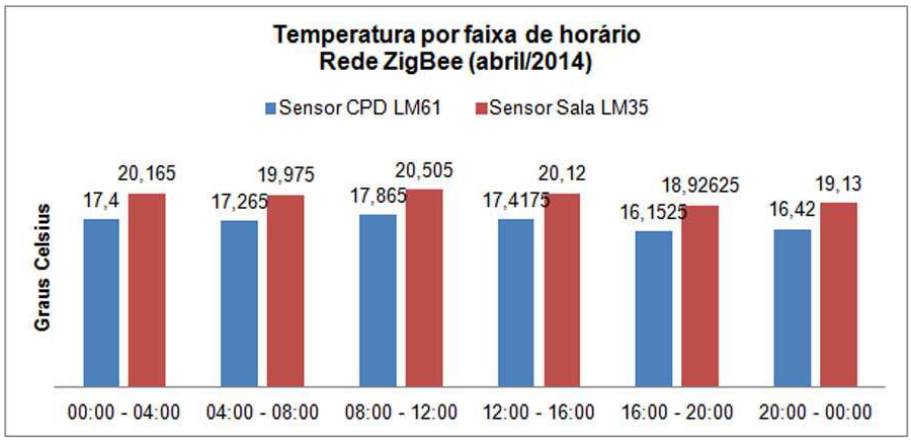


Figura 4.31: Médias horárias de temperatura coletadas durante o mês de abril.

Os valores exibidos na Figura 4.31 correspondem a todo o mês de abril de 2014. As médias foram divididas em intervalos de quatro horas. Podemos notar que as médias mais baixas de temperatura foram coletadas entre as 16 e 20 horas, e também entre as 20h e meia-noite.

A tabela 4.5 sumariza algumas informações estatísticas das temperaturas coletadas pelos diferentes sensores.

Tabela 4.5: Informações estatísticas dos dados coletados.

	LM35	**LM61**
Média	19,54	16,92
Desvio Padrão	1,26	1,38
Variância	1,59	1,92
Moda	20	17,2

As informações expostas na tabela 4.5 representam todos os dados coletados durante o período de monitoramento. Pode-se notar que não houve grande variação nas leituras considerando o desvio padrão calculado. Os valores de temperatura que mais se repetiram, isto é, as modas das populações, se aproximam das médias apresentadas nas Figuras 4.30 e 4.31.

No entanto, em certos momentos, notamos alterações consideráveis nos valores coletados, principalmente pelo sensor LM61. Podemos considerar que houve grande movimentação no ambiente do CPD nesses momentos e também há de se ponderar sobre o sensor utilizado, visto que, conforme exposto na seção 4.3.1, o LM61 apresenta elevado grau de variação.

4.7.1.1 Monitoramento da comunicação sem fio

Um dos serviços oferecidos pelo *gateway* da rede ZigBee é o fornecimento de informações sobre a comunicação sem. Assim, com o intuito de verificar a qualidade da comunicação sem fio a cada amostragem de temperatura foram coletadas também informações sobre a intensidade do sinal recebido (*Received Signal Strengh Indicator* (RSSI)) pela antena do coordenador e sobre a qualidade do *link* de comunicação (*Link Quality Indicator* (LQI)) entre ele e os demais nós da rede. Essas informações são importantes para os administradores das RSSF, pois possibilita analisar e identificar interferências eletromagnéticas e realizar alterações na organização dos nós e na topologia da rede, no sentido de promover melhorias.

O RSSI é a indicação do nível de potência do sinal recebido pela antena. Quanto maior for o valor do RSSI coletado mais forte é o sinal de rádio recebido. De acordo com o *datasheet* dos módulos XBee o valor RSSI desses dispositivos é mensurado em -*dBm*, indicando a potência do sinal do último salto de comunicação. Sendo assim, esse valor precisa ser analisado com cautela em redes *mesh*. Entretanto, a rede ZigBee implantada não utilizou comunicação multi-salto, logo, considera-se as leituras desses valores como confiáveis. Valores de RSSI acima de -80 indicam boa potência de sinal recebido enquanto que valores inferiores a -100 representam baixa potência do sinal.

O LQI é utilizado em redes IEEE 802.15.4 tratando-se de uma caracterização da potência e/ou a qualidade de um pacote recebido (IEEE-Standard 2011). Ele considera a potência do sinal e também o número de erros recebidos. Os valores mínimos e máximos de LQI são 0 (0x00) e 255 (0xFF), respectivamente, os quais estão associados com os níveis menos e mais elevados de qualidade do sinal.

Dessa maneira coletamos ambos os valores de RSSI e LQI da rede ZigBee. O gráfico apresentado na Figura 4.32 exibe os pares de dados coletados na comunicação entre o coordenador

(*gateway*) e o terminal XBee CPD.

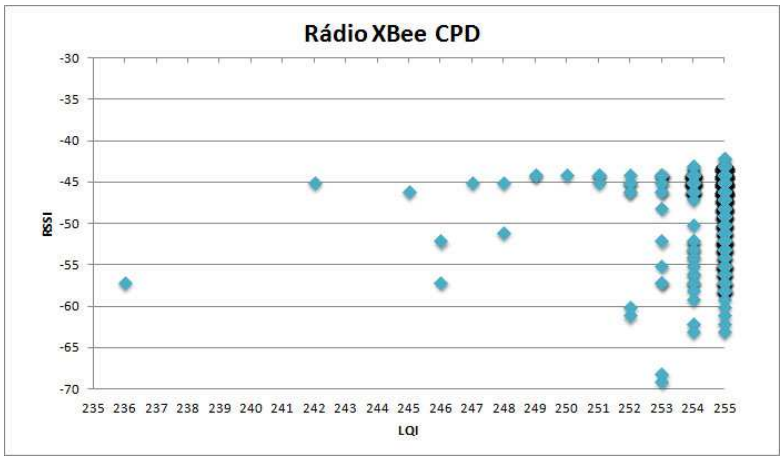


Figura 4.32: RSSI e LQI da comunicação entre o coordenador e o rádio do CPD.

Os dados coletados apresentados na Figura 4.32 apresentaram variações principalmente em termos de LQI, porém, nada que degradasse a comunicação entre o coordenador e o terminal XBee CPD.

A Figura 4.33 exibe as informações da comunicação sem fio entre o coordenador ZigBee e o nó XBee Sala.

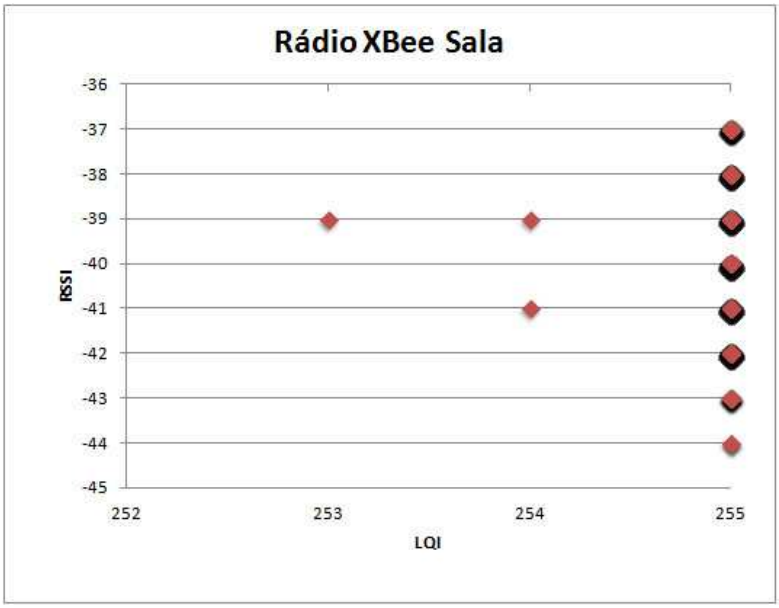


Figura 4.33: RSSI e LQI da comunicação entre o coordenador e o rádio da sala.

Analisando a Figura 4.33 é possível notar que a comunicação sem fio estabelecida entre esse terminal e o coordenador apresentou melhor qualidade e menor variação se comparado ao nó XBee CPD, apresentando maior concentração de amostras de altos valores de RSSI e de LQI.

A tabela 4.6 sumariza informações estatísticas da comunicação sem fio com os dois terminais da rede.

Tabela 4.6: Dados estatísticos da comunicação ZigBee.

	Rádio CPD		Rádio Sala	
	RSSI	**LQI**	**RSSI**	**LQI**
Média	-44,93	254,95	-38,55	254,99
Desvio Padrão	1,94	0,44	1,01	0,02
Variância	3,78	0,19	1,02	0,00
Moda	-44	255	-38	255

Os dados apresentados na tabela 4.6 e nas Figuras 4.33 e 4.32, correspondem a todo o período de monitoramento. Os dados da tabela confirmam a boa qualidade da comunicação na rede. Podemos concluir que não houve considerável interferência eletromagnética nos ambientes monitorados, embora a rede ZigBee opere na banda de frequência ISM, utilizada por diversos outros dispositivos, tais como: telefones sem fio; fornos micro-ondas; e rádios Wi-Fi.

De posse das informações sobre a comunicação sem fio os administradores são capazes de identificar interferências eletromagnéticas e realizar alterações na organização e topologia da rede.

4.7.1.2 Monitoramento da tensão de energia

Assim como a comunicação sem fio da rede ZigBee monitoramos a tensão de alimentação dos rádios XBee. Os rádios utilizados devem receber entre 2.6 e 3.6 Volts de alimentação para poder operar normalmente.

A Figura 4.34 exibe as variações de tensão do nó XBee CPD durante o dia 28 de abril.

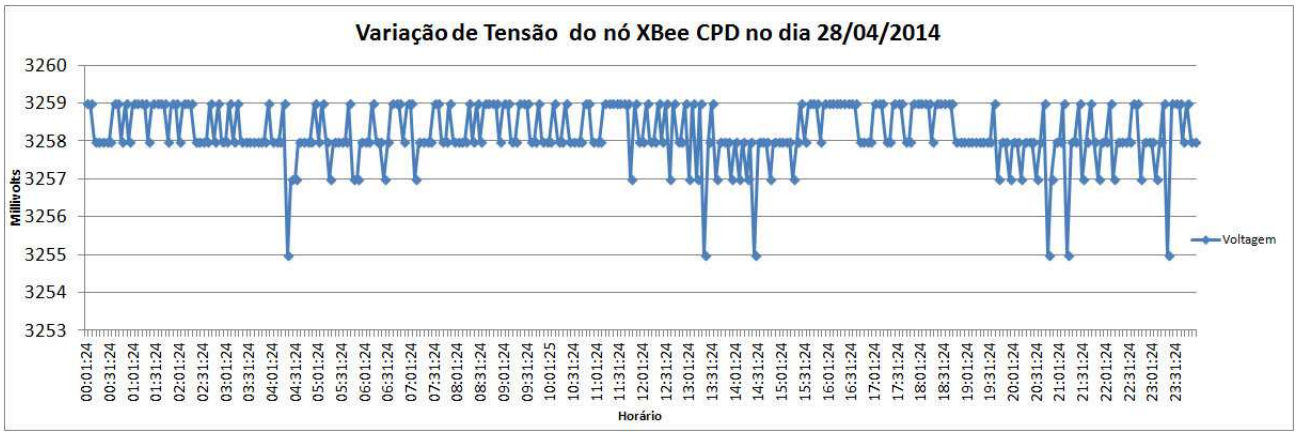
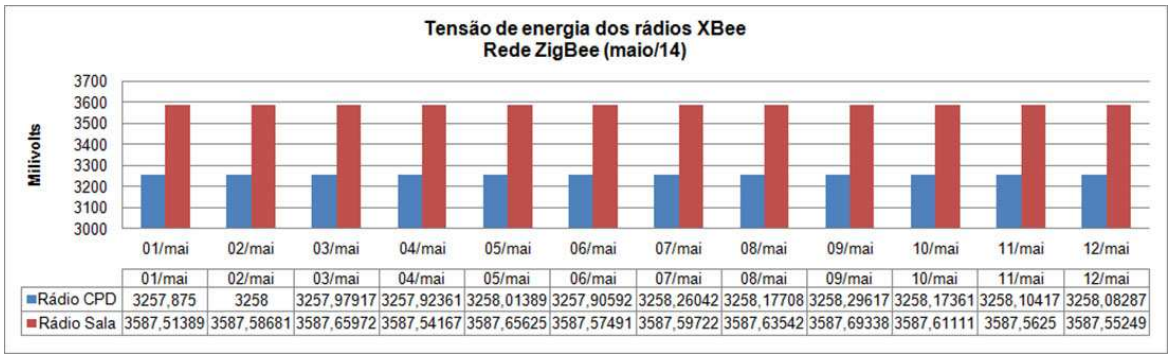


Figura 4.34: Variação de tensão do nó XBee CPD durante um dia.

Como mencionado, os dados sobre a tensão de alimentação foram coletados juntamente com a amostragem de temperatura enviada pelo nó, ou seja, em intervalos de cinco minutos. Pode-se observar também que a tensão de alimentação do nó se manteve estável na casa dos 3.2 Volts.

A Figura 4.35 exibe as variações de tensão apresentadas pelos dois nós da rede ZigBee entre os dias 01 e 12 do mês de maio de 2014.

Podemos notar na Figura 4.35 que as variações foram praticamente insignificantes. O rádio do CPD apresentou média de 3258 milivolts de alimentação (3.2V), enquanto que o outro



	01/mai	02/mai	03/mai	04/mai	05/mai	06/mai	07/mai	08/mai	09/mai	10/mai	11/mai	12/mai
Rádio CPD	3257,875	3258	3257,97917	3257,92361	3258,01389	3257,90592	3258,26042	3258,17708	3258,29617	3258,17361	3258,10417	3258,08287
Rádio Sala	3587,51389	3587,58681	3587,65972	3587,54167	3587,65625	3587,57491	3587,59722	3587,63542	3587,69338	3587,61111	3587,5625	3587,55249

Figura 4.35: Valores médios de tensão coletados durante duas semanas.

XBee teve 3588 (3.5V) de média. O monitoramento da fonte de energia permite verificar se há algum tipo de falha no módulo de alimentação do nó caso grande variabilidade de valores seja identificada.

Como mencionado na seção 4.3.1, utilizamos fontes de energia ligadas diretamente a rede elétrica do prédio para alimentar os nós sensores. Durante o monitoramento não constatamos problemas de queda ou sobrecarga de energia na rede. No caso de redes com nós alimentados por baterias, esse parâmetro monitorado permite a criação de um mapa de energia, onde é possível identificar os nós que possuem menor quantidade de energia restante e promover mudanças na rede para maximizar e prolongar o tempo de vida dos nós.

4.7.2 Monitoramento utilizando o *Telemetric*

Por se tratar de um equipamento de maior porte, o *Telemetric* apresenta recursos mais robustos que os nós ZigBee utilizados. Ele possui sensores e atuadores embarcados capazes de monitorar variáveis e controlar tomadas de energia.

Monitoramos seis diferentes variáveis com esse dispositivo na sala do morro do Cristo (P02) em Pedreira, sendo estas: tensão AC, tensão DC, temperatura ambiental e o estado de três tomadas. Embora o *Telemetric* possua seis tomadas para acionamento de dispositivos para fins de estudo de caso somente três foram monitoradas.

O voltímetro DC existente no *Telemetric* é responsável por monitorar a tensão gerada por uma bateria de 12V. Entretanto, não havia bateria no dispositivo monitorado. Portanto, esse sensor apresentou sempre o valor zero em suas leituras.

A Figura 4.36 exibe o monitoramento da tensão AC em um período de aproximadamente cinco horas no dia 10 de maio.

Na Figura 4.36 é possível visualizar as variações da tensão AC em função do tempo. Pode-se notar que os valores coletados variaram entre 112 e 113 Volts. Esse padrão se repetiu durante todo o período de monitoramento. O gráfico exposto na Figura 4.36 foi extraído da interface do portal *Web* da camada de aplicação.

Durante o período de monitoramento não foram identificadas grandes oscilações de energia, conforme exposto na tabela 4.7.

As tomadas monitoradas apresentam apenas dois estados, *On* e *Off*. O arquivo XML gerado pelo dispositivo *Telemetric* reporta tal estado em formato descritivo, ou seja *On* e *Off*. Sendo

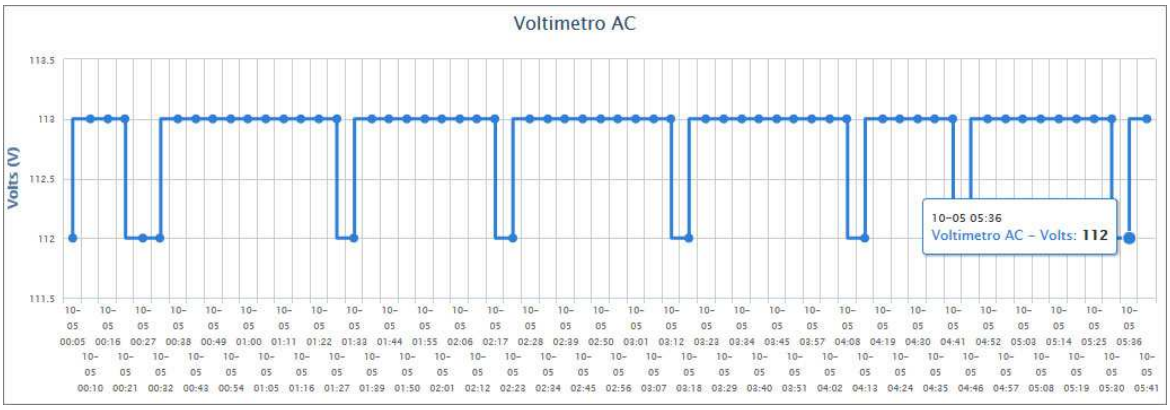


Figura 4.36: Monitoramento da tensão AC em um período de cinco horas.

Tabela 4.7: Estatísticas da tensão AC monitorada com o *Telemetric*.

Voltagem AC	
Média	112,96
Variância	0,03
Desvio Padrão	0,17
Moda	113

assim, o *gateway* do *Telemetric* desenvolvido em Python foi o responsável por converter esse estado em formato numérico, sendo 0 para *Off* e 1 para *On*, para permitir a realização de análises quantitativas.

A Figura 4.37 exibe o acompanhamento de uma das tomadas monitoradas por um período de cinco horas no dia 12 de maio.

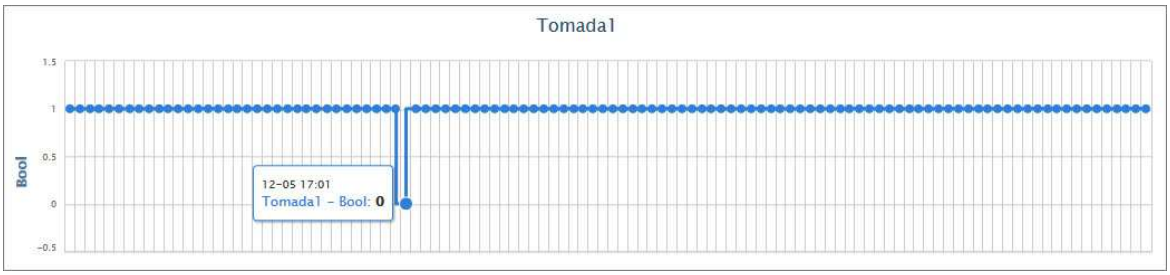


Figura 4.37: Mudança de estado em uma das tomadas monitoradas.

O gráfico apresentado na Figura 4.37 também foi retirado da interface de monitoramento *Web*. Nele é possível observar a desativação da tomada 1 e o seu acionamento logo em seguida. As demais tomadas monitoradas no *Telemetric* permaneceram em estado *On* durante todo o período de estudo de caso.

O sensor de temperatura embarcado no *Telemetric* coletou dados na sala do morro do Cristo.

A Figura 4.38 exibe as médias diárias de temperatura por um período de duas semanas do mês de maio.

É notável que as leituras de temperatura são muito mais elevadas que as coletadas pela rede ZigBee no prédio da prefeitura. Essa diferença é devido ao fato de não haver aparelho de

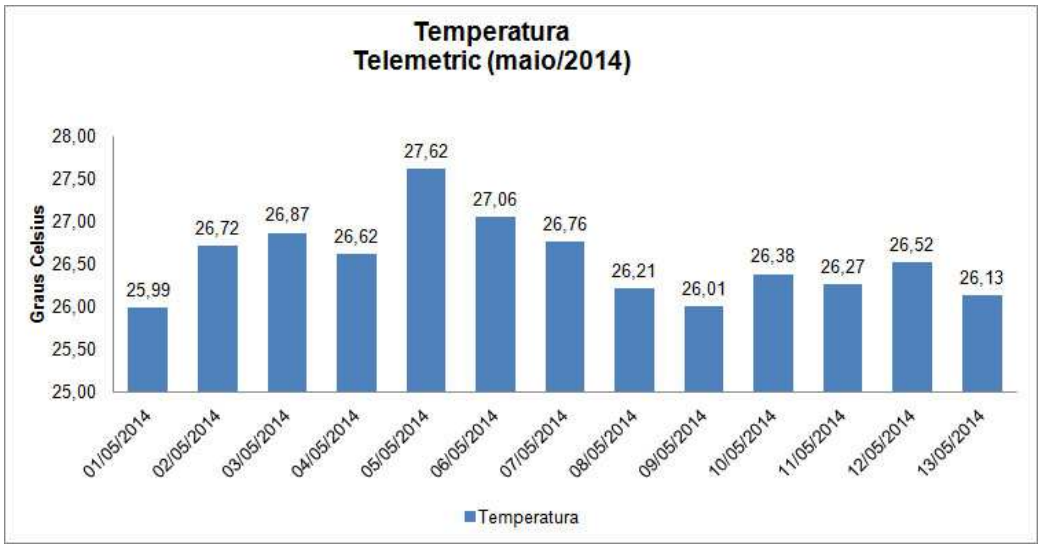


Figura 4.38: Médias diárias de temperatura durante duas semanas do mês do maio.

ar-condicionado na sala monitorada no morro do Cristo.

Podemos observar na Figura 4.38 que não houveram variações consideráveis nos valores coletados. Essa característica prevaleceu durante todo o tempo de estudo de caso, conforme exposto na tabela 4.8 que apresenta estatísticas referentes a toda população de dados coletados.

Tabela 4.8: Estatísticas da temperatura coletada pelo *Telemetric*.

Temperatura	
Média	26,59
Variância	0,63
Desvio Padrão	0,79
Moda	26

4.8 Gerenciamento

A função de um sistema de gerenciamento é monitorar e controlar a rede. Isso implica em um amplo conjunto de atividades que correspondem as cinco áreas funcionais do modelo criado pela Organização Internacional para Padronização (OSI) que estão descritas abaixo, de acordo com Kurose & Ross (2010):

- Gerenciamento de performance: tem por objetivo quantificar, mensurar e analisar a performance, por exemplo, em termos de utilização e rendimento, de diferentes componentes da rede.
- Gerenciamento de falhas: objetiva a criação de registros e resposta a condições anormais na rede.
- Gerenciamento de configuração: permite monitorar e alterar configurações de dispositivos na rede gerenciada, considerando as suas configurações de hardware e software.

- Gerenciamento de contabilização: refere-se ao controle e monitoramento de uso de recursos da rede, por um usuário, por exemplo. Esse tipo de gerenciamento inclui a definição de quotas e alocação de recursos baseada no perfil do usuário.

- Gerenciamento de segurança: tem por objetivo controlar o acesso aos recursos da rede.

Em termos de redes sensoras não há um protocolo que permita gerenciar de modo padronizado redes heterogêneas. Em redes TCP/IP um dos protocolos mais utilizados para realizar o gerenciamento é o SNMP, definido pelo IETF (Kurose & Ross 2010).

O processo de gerenciamento utilizando o SNMP envolve três componentes: o dispositivo gerenciado, o agente e o gerente. O dispositivo é um nó da rede, o qual hospeda um agente SNMP. O agente é o responsável por reunir informações do dispositivo e disponibilizá-las para o gerente. O gerente se comunica com diversos agentes para obter dados dos dispositivos. Dados sobre o dispositivo gerenciado são armazenados em bases de dados denominadas Base de Informações de Gerenciamento (*Management Information Base* (MIB)), das quais os agentes extraem as informações para encaminhamento ao gerente.

A utilização do protocolo SNMP para gerenciamento de redes IP e altamente explorada. Entretanto, no que diz respeito a redes sensoras, essa abordagem não se aplica para todas as redes, que em geral não operam sobre os protocolos TCP/IP.

Uma das primeiras e principais propostas para gerenciamento de redes de sensores sem fio é a MANNA (Ruiz et al. 2003). Ela compreende arquitetura de informação e arquitetura funcional, considerando especificidades desse tipo de rede. Conforme mencionado pelos autores, as redes de sensores sem fio apresentam características distintas de redes de computadores convencionais, que precisam ser levadas em consideração no momento do gerenciamento. Por exemplo, a ocorrência de falhas em RSSF é algo comum em virtude de escassez de energia, perda de comunicação com um nó, dentre outros, o que não é esperado em redes tradicionais. Ainda diferente de redes convencionais, as RSSF podem possuir comportamento dinâmico.

A arquitetura funcional da MANNA compreende a utilização de gerentes e agentes, e como essas entidades estão distribuídas na rede, isto é, se são centralizados, distribuídos ou hierárquico, considerando as características da rede como homogeneidade e organização. Essa arquitetura também prevê a utilização de MIBs para armazenamento das informações dos nós.

A arquitetura MANNA não define um protocolo para comunicação entre as entidades envolvidas no gerenciamento. Os autores consideram protocolos tradicionais, como SNMP, muito rígidos, sugerindo o uso de abordagens mais leves e que considerem a inteligência localizada em diferentes entidades, não somente no gerente, como ocorre no SNMP.

A arquitetura de informação é baseada no modelo de informação orientado a objetos. Esse modelo contém classes de objetos que representam recursos gerenciáveis em uma RSSF.

A heterogeneidade de redes sensoras que podem ser implantadas em um município dificulta a adoção de padrões para a gerência dessas. A arquitetura que estamos propondo permite obter dados das redes heterogêneas de modo padronizado, o que possibilita que grande volume de informações sejam analisadas e que alterações possam ser promovidas com base em tais análises.

Conforme apresentado nesse capítulo, na arquitetura proposta, funções de gerenciamento podem ser disponibilizadas na forma de serviços. Tais serviços podem incluir informações esta-

tísticas sobre o funcionamento de um determinado terminal sensor. O diferencial na utilização de serviços para fornecer informações pertinentes ao gerenciamento das redes sensoras, é que esses podem apresentar informações específicas que variam de rede para rede gerenciada. No entanto, as informações são obtidas sempre de modo padronizado, por meio de interfaces genéricas. De modo similar ao protocolo SNMP, na arquitetura proposta é possível utilizar gerentes e agentes, localizados em diferentes camadas. Pode haver somente um gerente, localizado no servidor da camada de aplicação se comunicando com agentes na camada de comunicação, localizados nos *gateways*. Da mesma maneira, é possível que haja vários gerentes, localizados nos dispositivos *gateways* dotados com tal capacidade, os quais se comunicam com agentes localizadas nas redes de sensores. As diferentes formas de organização dependem da quantidade de redes monitoradas e também dos dispositivos empregados.

Tendo em vista a amplitude de funções de gerenciamento que podem ser realizadas, nos atemos, nesse trabalho, a analisar algumas atividades simples, sempre com base nos dados coletados, com por exemplo: verificação de falhas na comunicação com os nós; configuração e contabilização. As demais atividades serão desenvolvidas em trabalhos futuros.

4.9 Considerações sobre o capítulo

Esse capítulo apresentou o estudo de caso realizado na Cidade Digital de Pedreira, no estado de São Paulo. Como prova de conceito, desenvolvemos e implantamos um sistema de monitoramento de redes de sensores em escala municipal nos moldes da arquitetura proposta.

O capítulo apresentou o desenvolvimento das soluções empregadas nas camadas componentes da arquitetura, desde a modelagem de software até a organização e montagem dos componentes de hardware.

Empregamos tecnologias distintas na camada de sensoriamento. Uma rede ZigBee em topologia estrela foi utilizada para monitorar a temperatura ambiental de duas diferentes salas do prédio da prefeitura. Para avaliar a capacidade da arquitetura em abstrair a heterogeneidade de dispositivos em nível de sensoriamento, um equipamento IP de maior porte, denominado *Telemetric* também foi empregado. Ele foi o responsável por coletar dados referentes ao controle de tomadas, voltagem AC, voltagem DC gerada por uma bateria e também temperatura ambiental.

Na camada de Comunicação, os *gateways* utilizados foram fundamentais para permitir o monitoramento homogêneo das redes de sensores. Eles foram responsáveis por abstrair as tecnologias existentes em nível de sensoriamento, provendo interfaces uniformes de acesso aos dados.

O portal *Web* apresentado representa a interface central para monitorar as redes implantadas. Como mencionado no capítulo 3, a sua utilização impôs apenas o uso de um navegador *Web* para acesso as informações. Aplicamos o conceito de *Responsive Web Design*, o que permitiu otimizar a organização das informações do portal para diferentes categorias de dispositivos.

Os dados coletados das diferentes redes foram armazenados na base central em nível de aplicação. Com o histórico de dados, realizamos e apresentamos algumas análises estatísticas das variáveis mensuradas. Também exibimos, de forma superficial, o monitoramento da comunica-

ção sem fio ZigBee, com o intuito de identificar interferências eletromagnéticas no ambiente. Isso é devido a existência de outras tecnologias de comunicação sem fio no local. Os resultados mostraram que a qualidade da comunicação foi ótima, com algumas oscilações em termos de RSSI, mas nada que degradasse de modo significativo a comunicação. Resultados do monitoramento da tensão de energia dos nós sensores ZigBee também foram expostos.

Com relação a RMAA, é importante destacar que durante o estudo de caso realizado não registramos falhas de comunicação na rede. Isto é, a comunicação do servidor com o *gateway* e com o *Telemetric* ocorreu de forma consistente.

É importante destacar, também, que o objetivo do estudo de caso não foi o de realizar estudos aprofundados nos dados coletados dos ambientes monitorados. As análises apresentadas serviram apenas para expor o potencial da arquitetura em prover dados para auxiliar sistemas inteligentes.

Por fim, a finalidade do sistema de monitoramento ambiental em escala municipal desenvolvido foi o de validar a arquitetura proposta nesse trabalho. Conforme mencionado, o estudo de caso em um cenário real permite identificar problemas não previstos na fase de elaboração da arquitetura e propor melhorias para aplicações futuras.

Capítulo 5

Considerações Finais

As redes de sensores desempenham importante papel na concepção das Cidades Digitais. A implantação de sensores permite monitorar inúmeras variáveis em um município, produzindo imensa e valiosa quantidade de informações para contribuir com a evolução e melhoria da cidade.

Neste trabalho propomos uma arquitetura baseada no modelo de orientação a serviços para o monitoramento e gerenciamento de redes de sensores em Redes Metropolitanas de Acesso Aberto. Constatou-se na literatura o crescimento de iniciativas, ainda que em baixo número, voltadas para a aplicação dessas redes em cenários urbanos. Podemos notar também que uma das principais dificuldades para criar sistemas de larga escala é a falta de uma infraestrutura de comunicação capaz de prover conectividade em um município. Adicionalmente, a heterogeneidade de tecnologias empregadas na concepção das soluções de sensores inviabiliza a interoperação entre sistemas.

Assim sendo, nossa arquitetura constitui um ambiente de interoperabilidade e de integração considerando diferentes redes de sensores existentes em um cenário de Cidade Digital. Para isso utilizamos *Web services* tanto para compartilhamento de dados quanto para comunicação, interconectando diferentes sistemas.

A utilização dos recursos disponíveis na RMAA representa uma importante característica da arquitetura proposta. Além de promover o acesso à informação e beneficiar o governo e seus munícipes através de soluções em forma de serviços, principalmente de modo *Online*, as RMAA permitem o desenvolvimento de diversas outras soluções para uma cidade. Na arquitetura ela é responsável pela interconexão de pontos distantes e centralização de informações no cenário metropolitano, sem a necessidade de recursos como a Internet.

Nos baseamos na mudança de paradigma de sistemas *Desktop* para plataformas *Web*, aproveitando os benefícios desse ambiente para propor uma arquitetura independente de plataforma, flexível e interoperável. Nossa proposta considerou tecnologias de ponta aplicadas nas diferentes camadas da arquitetura. Na camada de aplicação, pensando no crescimento e extensibilidade de sistemas sensores em RMAA, modelamos os seus componentes de acordo o padrão MVC, para separar a apresentação das informações da lógica de negócios da aplicação, permitindo que ambas escalem de modo independente e eficiente. Considerando aspectos tecnológicos atuais, o portal desenvolvido leva em conta diferentes categorias de dispositivos, como *tablets* e *smartphones*, para o monitoramento das redes. A única imposição da arquitetura para acesso ao sistema

de monitoramento é a utilização de um navegador *Web*.

Ainda na camada de aplicação, uma base central de conhecimento concentra as informações de diferentes domínios de aplicação de sensores no município. Essa base representa uma fonte de dados importante para sistemas inteligentes, formuladores de políticas e responsáveis por tomada de decisões. Para compartilhamento dos dados utilizamos *Web services* baseados em REST.

O desenvolvimento de RMAA é a base para constituir a camada de comunicação da arquitetura. Os *gateways*, provedores de serviço para a manipulação das redes de sensores, são fundamentais para promover a interoperabilidade nessa camada de arquitetura.

A camada de sensoriamento da arquitetura considerou o estado da arte de redes de sensores, principalmente as redes de área pessoal sem fio, como o padrão ZigBee. Mais do que isso, nossa arquitetura permite que diferentes tecnologias de sensores sejam empregadas, uma vez que a camada de comunicação é a responsável por abstrair essa heterogeneidade.

Realizamos um estudo de caso para avaliar a viabilidade da proposta. Desenvolvemos e implantamos nos moldes da arquitetura, um sistema sensor de escala municipal na Cidade Digital de Pedreira, concebida sobre uma RMAA. Esse sistema nos permitiu monitorar de modo eficiente e transparente uma rede ZigBee e um *Telemetric* utilizados em nível de sensoriamento. Após a coleta dos dados, realizamos simples análises estatísticas nas informações armazenadas na base central de conhecimento.

A aplicação do sistema de monitoramento ambiental na cidade de Pedreira permitiu avaliar a proposta do trabalho. O sistema implantado mostrou-se capaz de coletar dados de diferentes redes e reportar dados para o usuário de maneira eficiente, além de permitir a gerência de ambiente e dos componentes da rede, igualmente, fornecer interface de acesso para sistemas externos.

O compartilhamento dos dados sensores permite que novas aplicações possam ser desenvolvidas. Alguns exemplos de serviços incluem informações atualizadas sobre condições climáticas em determinados pontos da cidade, condições do tráfego também em regiões específicas do município, localização de pessoas e objetos, e assim por diante.

Podemos então concluir afirmando que os objetivos definidos nesse trabalho foram atingidos. A aplicação do conceito de orientação a serviços se mostrou eficiente para a realização do monitoramento de redes heterogêneas em cenários urbanos. Nossos resultados demonstram que esse conceito se mostra promissor, pois a utilização de formatos flexíveis de dados e protocolos independentes de plataforma permite abstrair a complexidade em termos de sistemas sensores. O potencial das RMAA também ficou evidente nesse trabalho, pois, além da imensa gama de serviços que ela permite oferecer, sua infraestrutura de comunicação permite a experimentação de propostas como a nossa.

Observamos ainda que o modelo arquitetural proposto neste trabalho vai além do contexto de Cidades Digitais, podendo ser empregado em situações onde exista diversidade de aplicações sendo monitoradas, bem como, cenários de larga escala.

Por fim, não podemos desvincular as contribuições elencadas na introdução dessa dissertação.

Trabalhos Futuros

O objetivo da arquitetura proposta é ambicioso, e assim, trabalhos adicionais precisam ser desenvolvidos no sentido de aprimorá-la. A seguir listamos algumas direções para trabalhos futuros, a citar:

- Analisar a aplicação da proposta em outros modelos de RMAA.
- Criar padrões de representação de informação que suportem uma sintaxe comum, para a troca de dados entre sistemas sensores no âmbito de uma Cidade Digital.
- Verificar meios para a integração de sistemas já existentes em um município, através da arquitetura.
- Propor *gateways* mais robustos e funcionais, capazes de manipular grandes quantidades de dados e armazenar localmente as leituras realizadas pelas redes de sensores.
- Avaliar a real necessidade dos municípios para aplicar a arquitetura em sistemas que de fato produzam benefícios para o governo e população.
- Analisar com mais profundidade as áreas funcionais de gerenciamento de redes para que sejam contempladas em sua totalidade pela arquitetura.
- Explorar mais a presença de dispositivos atuadores em nível de sensoriamento.

Publicações

Nessa seção são apresentados alguns trabalhos publicados, resultantes das pesquisas realizadas nessa dissertação.

- Morbach, Rodrigo A.; Breda, Gean D.; Mendes, Leonardo de S. "Proposta de uma arquitetura para monitoramento e gerenciamento de redes de sensores em Redes Metropolitanas de Acesso Aberto". VI Simpósio Brasileiro de Computação Ubíqua e Pervasiva, XXXIV Congresso da Sociedade Brasileira de Computação. Brasília - DF, 2014. (Trabalho aceito para publicação).
- Morbach, Rodrigo A.; Pazeto, Tatiana A.; Mendes, Leonardo de S. "Ferramenta para gestão de custo utilizando redes de sensores". In Anais do XII Simpósio de Informática, de 24 a 22 de outubro de 2013, Santa Maria, Rio Grande do Sul, Brasil.
- Morbach, Rodrigo A.; Pazeto, Tatiana A.; Mendes, Leonardo de S. "Protótipo para gerenciamento de contabilização em espaços públicos". In Anais do V Congresso de Computação do Sul de Mato Grosso, de 20 a 23 de agosto de 2013, Rondonópolis, Mato Grosso, Brasil.

Bibliografia

Aihkisalo, T. & Paaso, T. (2012), Latencies of Service Invocation and Processing of the REST and SOAP Web Service Interfaces, *in* 'Services (SERVICES), 2012 IEEE Eighth World Congress on', pp. 100–107.

Akyildiz, I., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002), 'A survey on Sensor Networks', *Communications Magazine, IEEE* **40**(8), 102–114.

Alexiou, A., Bouras, C., Igglesis, V., Kapoulas, V., Paraskeuas, M., Scopoulis, I. & Papagiannopoulos, J. (2005), Deployment of broadband infrastructure in the region of Western Greece, *in* 'Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on', pp. 1510–1515 Vol. 2.

Alonso, G., Casati, F. & Kuno, H. (2004), *Web Services: Concepts, Architectures and Applications*, 1th edn, Springer.

Apache (2014), 'Apache HTTP Server Project'. Disponível em: https://httpd.apache.org/. Acesso em 10 de fevereiro de 2014.

Arduino (2014), 'Arduino Uno'. Disponível em: http://arduino.cc/en/Main/arduinoBoardUno. Acesso em 12 de março de 2014.

Bartlett, D., Harthoorn, W., Hogan, J., Kehoe, M. & Schloss, R. (2011), 'Enabling integrated city operations', *IBM Journal of Research and Development* **55**(1.2), 15:1–15:10.

Bauer, J., Gai, P., Kim, J., Muth, T. & Wildman, S. (2002), Broadband: Benefits and Policy Challenges, Technical report, Michigan State University.

Box, D., Ehnebuske, D., Kakivaya, G., Andrew, L., Mendelsohn, N., Nielsen, H. F., Thatte, S. & Winer, D. (2000), 'Simple Object Access Protocol (SOAP) 1.1'. Disponível em: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/. Acesso em 14 de fevereiro de 2014.

Breda, G. D., Capucin, T. W., Mendes, L. D. S. & Zarpelão, B. B. (2011), 'Applied Architectures to Brazilian ´ s Open Access MAN s', *International Journal of Computer Science and Network Security* **11**(9), 29–34.

Chong, C.-Y. & Kumar, S. P. (2003), Sensor Networks: Evolution, Opportunities, and Challenges, *in* 'Proceedings of IEEE', Vol. 91, pp. 1247–1256.

Christensen, E., Curbera, F., Meredith, G. & Weerawarana, S. (2001), 'Web Services Description Language (WSDL) 1.1'. Disponível em: http://www.w3.org/TR/wsdl. Acesso em 01 de abril de 2014.

Deitel, P. J. & Deitel, H. M. (2008), *AJAX, Rich Internet Applications and Web Development for Programmers*, 1th edn, Pearson Education Inc., Boston, MA, USA.

Dissanaike, S., Wijkman, P. & Wijkman, M. (2004), Utilizing XML-RPC or SOAP on an embedded system, *in* 'Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on', pp. 438–440.

Du, X. & Chen, H.-H. (2008), 'Security in Wireless Sensor Networks', *Wireless Communications, IEEE* **15**(4), 60–66.

Eastlake, D. (2001), 'US Secure Hash Algorithm 1 (SHA1)'. Disponível em: http://tools.ietf.org/html/rfc3174. Acesso em 20 de fevereiro de 2014.

Erl, T. (2008), *SOA: Principles of service design*, 1th edn, Pearson Education Inc.

Faludi, R. (2010), *Building Wireless Sensor Networks*, 1th edn, O'REILLY, Sebastopol, CA, USA.

Fan, C.-l. & Guo, Y. (2011), The application of a ZigBee based Wireless Sensor Network in the LED street lamp control system, *in* 'Image Analysis and Signal Processing (IASP), 2011 International Conference on', pp. 501–504.

Fielding, R. T. (2000), Architectural Styles and the Design of Network-based Software Architectures, PhD thesis, University of California, Irvine - CA. Disponível em: http://www.ics.uci.edu/˜fielding/pubs/dissertation/top.htm. Acesso em 14 de fevereiro de 2014.

Gao, T., Pesto, C., Selavo, L., Chen, Y., Ko, J., Lim, J. H., Terzis, A., Watt, A., Jeng, J., rong Chen, B., Lorincz, K. & Welsh, M. (2008), Wireless Medical Sensor Networks in Emergency Response: Implementation and Pilot Results, *in* 'Technologies for Homeland Security, 2008 IEEE Conference on', pp. 187–192.

Gill, K., Yang, S.-h., Yao, F. & Lu, X. (2009), A ZigBee-Based Home Automation System, *in* 'Computer Eletronics', Vol. 55, IEEE, pp. 422–430.

Gniadek, A., Li, Y., Lung, C.-H. & Wei, Q. (2008), A Web services-based Infrastructure for traffic monitoring using ZigBee, *in* 'Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on', pp. 562–567.

Graham, S., Simeonov, S., Boubez, T., Davis, D., Daniels, G., Nakamura, Y. & Neyama, R. (2002), *Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI*, 1th edn, Sams.

Hernández-Muñoz, J. M., Vercher, J. B., Muñoz, L., Galache, J. A., Presser, M., Gómez, L. A. H. & Pettersson, J. (2011), The future internet, *in* J. Domingue, A. Galis, A. Gavras, T. Zahariadis, D. Lambert, F. Cleary, P. Daras, S. Krco, H. Muller, M.-S. Li, H. Schaffers, V. Lotz, F. Alvarez, B. Stiller, S. Karnouskos, S. Avessta & M. Nilsson, eds, 'The Future Internet', Vol. 6656 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, chapter Smart Cities at the Forefront of the Future Internet, pp. 447–462.
URL: *http://dl.acm.org/citation.cfm?id=1983741.1983773*

Hui, W. Q., juan, B. L. & jun, P. Y. (2011), Design of the temperature monitoring system based on ZigBee Wireless Sensor Network, *in* 'Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on', pp. 4612–4615.

IDC, I. D. C. (2013), 'Worldwide smart connected device market crossed 1 billion shipments in 2012, Apple pulls near Samsung in fourth quarter'. Disponível em: http://www.idc.com/getdoc.jsp?containerId=prUS24037713. Acesso em 18 de março de 2014.

IEEE-Standard, I. C. S. (2011), IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low-Rate Wireless Personal Area Network (LR-WPANs), Technical report, IEEE-SA Standards Board.

Instituto Brasileiro de Geografia e Estatística, I. (2010), 'Informações sobre a cidade de Pedreira - SP'. Disponível em: http://cidades.ibge.gov.br/xtras/perfil.php?codmun=353710. Acesso em 14 de agosta de 2014.

Juraschek, F., Zubow, A., Hahm, O., Scheidgen, M., Blywis, B., Sombrutzki, R., Gunes, M. & Fischer, J. (2012), Towards Smart Berlin - an experimental facility for heterogeneous Smart City infrastructures, *in* 'Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on', pp. 886–892.

Kramer, R. D. J., Lopez, A. & Koonen, A. M. J. (2006), Municipal Broadband Access Networks in the Netherlands - Three Successful Cases, and How New Europe May Benefit, *in* 'Proceedings of the 1st International Conference on Access Networks', AcessNets '06, ACM, New York, NY, USA.
URL: *http://doi.acm.org/10.1145/1189355.1189367*

Kurose, J. F. & Ross, K. W. (2010), *Computer Networkking: A top-down approach*, 5th edn, Pearson Education.

Lansford, J. (2000), HomeRF(TM)/SWAP: a wireless voice and data system for the home, *in* 'Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on', Vol. 6, pp. 3718–3721 vol.6.

Larman, C. (2005), *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*, 3rd edn, Person Education.

Loureiro, A., Nogueira, J. M. S., Mini, R. A. F., Ruiz, L. B., Nakamura, E. F. & Figueiredo, C. M. S. (2003), Redes de Sensores sem Fio, *in* 'XXI Simpósio Brasileiro de Redes de Computadores', Natal, Rio Grande do Norte, pp. 179–226.

Ma, K. (2005), 'Web services: what's real and what's not?', *IT Professional* **7**(2), 14–21.

Mendes, L. d. S., Botolli, M. L. & Breda, G. (2010), 'Digital Cities and Open MANs: A New Communications Paradigm', *IEEE Latin America Transactions* **8**(4), 394–402.

Mendes, L. S. (2008), 'Infovia Municipal - Um novo paradigma em comunicações'. Disponível em: http://www.politics.org.br/edicao_01/downloads/poliTICS_LeonardoMendes.pdf. Acesso em 10 de fevereiro de 2014.

Miani, R. S. (2009), Aplicação de métricas à análise de segurança em Redes Metropolitanas de Acesso Aberto, Master's thesis, Universidade Estadual de Campinas, Campinas - SP.

Mohorovicic, S. (2013), Implementing Responsive Web Design for enhanced Web presence, *in* 'Information Communication Technology Electronics Microelectronics (MIPRO), 2013 36th International Convention on', pp. 1206–1210.

Mulligan, G. & Gracanin, D. (2009), A comparison of SOAP and REST implementations of a service based interaction independence middleware framework, *in* 'Simulation Conference (WSC), Proceedings of the 2009 Winter', pp. 1423–1432.

MySQL (2014), 'MySQL'. Disponível em: http://www.mysql.com. Acesso em 10 de fevereiro de 2014.

Naruephiphat, W., Prom-Ya, R. & Chansripinyo, C. (2013), A Web-based management system design for Wireless Sensor Ntwork monitoring, *in* 'Computer Science and Engineering Conference (ICSEC), 2013 International', pp. 281–285.

Panhan, A. M. (2011), Proposta de uma Arquitetura para Cidades Digitais baseada em um Middleware Peer-to-Peer, PhD thesis, Universidade Estadual de Campinas, Campinas - SP.

Patil, S. D., Aurangbad, J., Prof, A. & Aurangabad, J. (2012), 'Wireless AMR System using ZigBee', **1**(4), 1–6.

PHP (2014), 'Hypertext Preprocessor'. Disponível em: http://www.php.net. Acesso em 10 de fevereiro de 2014.

Pires, F. M. (2010), Proposta de uma Arquitetura Híbrida e Hierárquica de Redes de Sensores/Atuadores para Aplicação em Cenários Metropolitanos, Master's thesis, Universidade Estadual de Campinas, Campinas - SP.

Python (2014), 'Python Programming Language'. Disponível em: https://www.python.org/. Acesso em 10 de fevereiro de 2014.

Qiu, J., Xiao, B. & Lv, K. (2012), Design and implementation of the short-term forecasting and environmental monitoring system over WSN for urban special site, *in* 'Advanced Computational Intelligence (ICACI), 2012 IEEE Fifth International Conference on', pp. 756–759.

Reenskaug, T. (2007), 'The Original MVC reports'. Disponível em: http://heim.ifi.uio.no/~trygver/2007/MVC_Originals.pdf. Acesso em 17 de janeiro de 2014.

Rezgui, A. & Eltoweissy, M. (2007), 'Service-oriented sensor-actuator networks: Promises, challenges, and the road ahead', *Computer Communications* **30**(13), 2627–2648.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0140366407002472*

Rivest, R. (1992), 'The MD5 Message-Digest Algorithm'. Disponível em: http://www.ietf.org/rfc/rfc1321.txt. Acesso em 20 de fevereiro de 2014.

Rodriguez, A. (2008), 'RESTful Web services: The basics'. Disponível em: http://www.ibm.com/developerworks/webservices/library/ws-restful/. Acesso em 28 de fevereiro de 2014.

Ruiz, L., Nogueira, J. & Loureiro, A. A. F. (2003), 'Manna: a management architecture for wireless sensor networks', *Communications Magazine, IEEE* **41**(2), 116–125.

Stojkoska, B. & Davcev, D. (2009), Web interface for habitat monitoring using wireless sensor network, *in* 'Wireless and Mobile Communications, 2009. ICWMC '09. Fifth International Conference on', pp. 157–162.

Tanenbaum, A. S. & Wetherall, D. J. (2011), *Computer Networks*, 5th edn, Prentice Hall, New York, NY, USA.

Vikatos, P., Theodoridis, E., Mylonas, G. & Tsakalidis, A. (2011), PatrasSense: Participatory Monitoring of Environmental Conditions in Urban Areas Using Sensor Networks and Smartphones, *in* 'Informatics (PCI), 2011 15th Panhellenic Conference on', pp. 392–396.

W3C, W. W. W. C. (2003), 'Extensible Markup Language'. Disponível em: http://www.w3.org/XML/. Acesso em 13 de fevereiro de 2014.

W3C, W. W. W. C. (2004), 'Web Services Architecture'. Disponível em: http://www.w3.org/TR/ws-arch/. Acesso em 14 de fevereiro de 2014.

W3C, W. W. W. C. U. W. D. (2001), 'XML Schema'. Disponível em: http://www.w3.org/XML/Schema. Acesso em 02 de março de 2014.

Wang, W., Zou, Y.-X., Shi, G. & Zhu, Y. (2009), A Web service based gateway architecture for wireless Sensor Networks, *in* 'Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on', Vol. 02, pp. 1160–1163.

Winer, D. (1999), 'XML-RPC Specification'. Disponível em: http://xmlrpc.scripting.com/spec.html. Acesso em 03 de março de 2014.

Wu, S.-L. & Tseng, Y.-C. (2007), *Wireless Ad Hoc Networking: Personal-Area, Local-Area and the Sensory-Area Networks*, 1th edn, Auerbach Publications, Boca Rator, FL, USA.

Xia, F., Tian, Y.-c., Li, Y. & Sun, Y. (2007), 'Wireless Sensor/Actuator Network Design for Mobile Control Applications', *Sensors* pp. 2157–2173.

Zend (2014), 'Zend framework'. Disponível em: http://www.framework.zend.com. Acesso em 10 de fevereiro de 2014.

Zigbee-Alliance (2012), 'The Zigbee Alliance Website'. Disponível em: http://zigbee.org/Specifications/ZigBee/Overview.aspx. Acesso em 04 de agosto de 2013.

Zou, J., Mei, J. & Wang, Y. (2010), From Representational State Transfer to Accountable State Transfer Architecture, *in* 'Web Services (ICWS), 2010 IEEE International Conference on', pp. 299–306.