UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

INTERAÇÃO DINÂMICA DE DISPOSITIVOS RESIDENCIAIS ATRAVÉS DE UM GATEWAY ORIENTADO A SERVIÇOS

RUBENS ROCHA VALENTE JUNIOR

MANAUS-AM

2013

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RUBENS ROCHA VALENTE JUNIOR

INTERAÇÃO DINÂMICA DE DISPOSITIVOS RESIDENCIAIS ATRAVÉS DE UM GATEWAY ORIENTADO A SERVIÇOS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica, área de concentração Automação de Sistemas.

Orientador: Prof. Dr.–Ing. Vicente Ferreira de Lucena Júnior

MANAUS-AM

2013

Ficha Catalográfica
(Catalogação realizada pela Biblioteca Central da UFAM)

Valente Junior, Rubens Rocha

V154i Integração dinâmica de dispositivos residenciais através de um Gateway orientado a serviços / Rubens Rocha Valente Junior - Manaus, 2013.

92f. il. p&b.

Dissertação (mestrado em Engenharia Elétrica) – Universidade Federal do Amazonas.

Orientador: Prof. Dr. Ing. Vicente Ferreira de Lucena Júnior

1. Gateway 2. Zigbee 3. Web Services I. Lucena Júnior, Vicente Ferreira II. Universidade Federal do Amazonas III. Título

CDU 2007 62: 004.41(043.3)

RUBENS ROCHA VALENTE JUNIOR

INTERAÇÃO DINÂMICA DE DISPOSITIVOS RESIDENCIAIS ATRAVÉS DE UM GATEWAY ORIENTADO A SERVIÇOS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 20 de Junho de 2013.

BANCA EXAMINADORA

Prof. Dr.- Ing Vicente Ferreira de Lucena Junior, Presidente

Universidade Federal do Amazonas- UFAM

Prof. Dr. Waldir Sabino da Silva Junior, Membro

Universidade Federal do Amazonas- UFAM

Prof. Dr. José Pinheiro de Queiroz Neto, Membro

Instituto Federal de Educação, Ciência e Tecnologia do Amazonas- IFAM

AGRADECIMENTO

A Deus por ter me dado saúde e força para superar as dificuldades.

A esta universidade, seu corpo docente, direção e administração pelo carinho, dedicação e entusiasmo demonstrado ao longo do curso.

Ao professor e orientador Vicente Ferreira de Lucena Junior por seu apoio e inspiração no amadurecimento dos meus conhecimentos e conceitos que me levaram a execução e conclusão desta tese.

A minha família pelo amor, incentivo e apoio incondicional.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

As pesquisas relacionadas aos ambientes inteligentes têm enfatizado os aspectos técnicos associados ao controle de dispositivos e à interligação desses dispositivos em rede, com sistemas de controles domésticos envolvendo questões relativas ao gerenciamento de dispositivos eletroeletrônicos no ambiente doméstico, iniciando com a aquisição e o tratamento de dados provenientes de sensores, até a modificação do estado dos atuadores. Esse processo envolve o estudo das características desses transdutores, o desenvolvimento de protocolos e mecanismos de transmissão de dados, a criação de métodos de análise e gerenciamento das informações, a criação de interfaces, etc. Para que isso seja possível, é necessário prover meios que forneçam a interoperabilidade entre os eletrodomésticos e as aplicações de controle.

Este trabalho propõe um sistema *gateway* baseado em serviços para monitoramento e controle de eletrodomésticos em uma rede de comunicação sem fio. O *gateway* proposto fornece uma camada de abstração entre aplicações e a infraestrutura de rede subjacente, bem como serviços especialmente designados para atender às necessidades específicas das aplicações. O *gateway* provê uma interface padrão para o acesso às funcionalidades dos eletrodomésticos, o que confere alto grau de flexibilidade e interoperabilidade ao sistema. O projeto da interface disponibilizada pelo *gateway* foi inspirado na área de serviços Web, e utiliza tecnologias correlatas à área, como o estilo de arquitetura de software REST, considerado padrão ubíquo na Web. Ao esconder detalhes quanto à configuração e adaptação da rede dos desenvolvedores, o *gateway* minimiza os esforços de desenvolver aplicações para automação residencial.

Palavras-chave: ZigBee, Gateway, serviços Web.

ABSTRACT

The research related to smart environments have emphasized the technical aspects associated with the control of these devices and the interconnection network devices, home control systems involving issues relating to the management of electronic devices in the home, starting with the acquisition and processing of data from sensing, by modifying the state of the actuators. This process involves the study of the characteristics of these transducers, the development of protocols and mechanisms for data transmission, the creation of methods of analysis and information management, the creation of interfaces, etc. For this to be possible, it is necessary to provide means which provide interoperability between appliances and control applications.

This paper proposes a system based gateway services for monitoring and control of home appliances in a network of wireless communication. The proposed gateway provides an abstraction layer between applications and the underlying network infrastructure and provides services specially designed to meet the needs of specific applications. The gateway provides a standard interface to access the functionality of the appliance, which provides a high degree of flexibility and interoperability system. The design of the interface provided by the gateway was inspired in the area of Web services and related technologies used to the area, as the style of software architecture REST, considered standard ubiquitous on the Web By hiding details of setting up and adapting the network of developers, gateway minimizes the efforts of developing applications for home automation.

Keywords: ZigBee, Gateway, Web Services.

Lista de figuras

Lista de tabelas

Sumário

Capítulo 1 - Introdução

O desenvolvimento dos sistemas de Automação Residencial está cada vez mais presente nos projetos de empreendimentos residenciais, deixando de ser algo futurista, e se transformando em uma realidade, ocorrendo cronologicamente depois de seus similares nas áreas industrial e comercial.

Nas últimas duas décadas, a popularização do computador pessoal, da Internet e, mais recentemente, dos dispositivos móveis e redes sem fio têm alterado substancialmente o modo como as pessoas se comunicam, trabalham e se entretêm. Essas tecnologias vêm familiarizando as pessoas com novos equipamentos eletrônicos, novas interfaces e com o conceito de interligação de dispositivos em rede, potencializando o papel das residências, que deixam de ser uma estrutura inerte para se tornarem ativas e dinâmicas, com a possibilidade de interagirem com seus moradores e com o mundo externo. Assim, é natural se esperar mudanças também nas rotinas domésticas.

Segundo Bonzani(2004), os avanços da microeletrônica e a redução dos custos associados ao processo de implementação de programas em minúsculos chips de silício têm permitido a aplicação de sistemas de controle de processos em situações que, há alguns anos, tinham custo proibitivo. A miniaturização e o crescente acesso às redes de comunicação contribuíram para o surgimento de uma classe de equipamentos mais avançados, como os dispositivos inteligentes e, com eles, diversas outras aplicações de controle e monitoramento para as residências, motivando pesquisas sobre temas até então nunca abordados.

Pesquisas relacionadas aos ambientes inteligentes (BONZANI, 2004b, 2009) têm enfatizado os aspectos técnicos associados ao controle de dispositivos e à interligação desses dispositivos em rede, com sistemas de controle domésticos envolvendo questões relativas ao gerenciamento de dispositivos eletroeletrônicos no ambiente doméstico, iniciando com a aquisição e o tratamento de dados provenientes de sensores, até a modificação do estado dos atuadores. Esse processo envolve o estudo das características desses transdutores, o desenvolvimento de protocolos e mecanismos de transmissão de dados, a criação de métodos de análise e gerenciamento das informações, a criação de interfaces, etc.

Embora os dispositivos eletroeletrônicos sejam susceptíveis de serem amplamente distribuídos em um futuro próximo, a integração direta de dispositivos do mundo real com a Internet ainda é uma tarefa bastante complexa, principalmente nos casos em que os dispositivos não suportam o protocolo IP, como ocorre normalmente no contexto dos sensores e atuadores. Nessas situações, nas quais os dispositivos não são capazes de se comunicar via IP, é possível utilizar um dispositivo intermediário chamado *gateway*.

1.2 Desafios

As pesquisas sobre a integração de dispositivos domésticos evoluíram bastante ao longo dos últimos anos com aplicações potenciais em diversos domínios. No entanto, a diversidade de tecnologias e protocolos de comunicação tem sido um impasse na integração e no desenvolvimento de aplicações para gerenciamento e monitoramento desses eletroeletrônicos.

Na integração de dispositivos domésticos (sensor, atuador), as dificuldades mais significativas estão relacionadas ao desenvolvimento, comunicação e tratamento dos dados coletados. Isso se deve à heterogeneidade existente entre diferentes plataformas, pois são necessários conhecimentos específicos de diferentes plataformas de hardware, linguagens de programação e protocolos de comunicação, o que demanda muito esforço e consome grande parte dos recursos de um projeto (YICK, 2008).

Nesse contexto, os desenvolvedores de aplicações para dispositivos eletroeletrônicos precisam conhecer tanto as especificidades das redes, incluindo protocolos e configurações de baixo nível, quanto os requisitos de alto nível da aplicação de controle. Por exemplo, se um programador desejar desenvolver uma aplicação para monitoramento ou interação com soluções de automação existentes, talvez seja necessário aprender uma nova linguagem de programação (ex. *NesC*), um novo paradigma de programação (ex. programação baseada em componentes), um novo sistema operacional embarcado (ex. *TinyOS*), e, provavelmente, tomar conhecimento de alguns detalhes do hardware da plataforma. O alto acoplamento entre a lógica da aplicação e a plataforma base do sensor resulta em projetos com códigos dependentes e difíceis de manter, modificar e reutilizar.

Essa situação está longe de ser a ideal, desenvolvedores de aplicações deveriam se concentrar em questões de nível do aplicativo, como o desenvolvimento da interface com o usuário e, idealmente, usar as linguagens de programação e ferramentas que eles estão acostumados. Fornecendo assim aos programadores abstrações adequadas e instrumentos para o desenvolvimento de aplicações que incorporam recursos disponibilizados pelos dispositivos físicos.

O segundo fator que limita a conexão de dispositivos domésticos é a existência de vários protocolos de comunicação que não são compatíveis entre si (SAUDATE, 2012), o que dificulta a integração com outras redes, principalmente as redes baseadas em IP, em especial a Internet. O monitoramento de sistemas residenciais via Internet é uma necessidade bastante presente na contemporaneidade, principalmente com o surgimento de aparelhos com, cada vez mais, poder de processamento, tais como *smartphones*, *tablets*, entre outros.

Dispositivos residenciais nem sempre possuem recursos computacionais suficientes para suportar um servidor embarcado. Portanto, a integração direta de dispositivos do mundo real com a Web ainda é uma tarefa bastante complexa; principalmente nos casos de dispositivos extremamente restritos em recursos como muitas plataformas de sensores utilizados em uma rede doméstica. Nesses casos, é necessário utilizar um padrão de integração diferente baseado na adoção de um dispositivo intermediário chamado *gateway*.

A interconexão entre dispositivos físicos e a Internet tem sido estudada por diversos pesquisadores, e existem principalmente três métodos para realizá-la: Gateway, DTN overlay e TCP/IP [(HUI; HAN, 2004), (DUNKELS; ALONSO; VOIGT, 2004)]. O estudo de comparação destes métodos foi proposto em trabalhos anteriores (DUNKELS, 2004). O gateway apresenta as vantagens de facilidade de implementação, baixo custo e uma conectividade, permitindo manter as estruturas sem necessidade de alterações em nenhuma das redes, de modo a respeitar a integridade de ambos os protocolos (JEONG-HEE, 2007).

Neste trabalho será descrito um sistema de automação residencial para integração de dispositivos residenciais com aplicações de controle e monitoramento, tanto local quanto remotamente através da Internet, com o uso de tecnologias e padrões de comunicação sem fio. A proposta tem como base a implementação de um sistema *gateway* que

disponibilizará as funcionalidades dos dispositivos na Internet através de serviços Web, que poderão ser acessados por aplicações de automação.

O serviço Web foi escolhido para implementar a integração das diferentes formas de serviços de rede, conforme pesquisa de Perumal et al (2008). Além da integração entre diferentes tipos de rede, os serviços Web serão usados para oferecer uma abstração ao desenvolvimento de aplicações para dispositivos físicos, de modo que esses serviços não tenham que se preocupar com o funcionamento e nem com as tecnologias envolvidas em cada sensor da rede, cada dispositivo será tratado como um provedor de serviços, por exemplo, uma lâmpada terá os serviços de ligar e desligar disponibilizados para outros dispositivos de uma rede IP através dos serviços Web gerados pelo *gateway*. Com o sistema *gateway* proposto, o desenvolvedor de aplicações de controle precisará apenas acessar os serviços Web, na sua linguagem de domínio, para se conectar com os dispositivos domésticos, restando assim mais tempo para o desenvolvimento de interface para o usuário final.

1.2 Motivação

Nos últimos anos, a introdução de entidades físicas (sensores, atuadores) em ambientes domésticos tem prosseguido a um ritmo sem precedentes. Além disso, com a rápida expansão da Internet, existe um potencial para o monitoramento remoto e o controle desses dispositivos em rede. No entanto, a oportunidade para aumentar a interligação de dispositivos em redes de protocolos diferentes em ambientes residenciais permanece um desafio (GILL, 2009).

A heterogeneidade entre redes e o alto acoplamento das aplicações fazem com que o acesso às informações seja uma tarefa difícil. Desta forma, este trabalho se propõe a apresentar um projeto no sentido da promoção da flexibilidade e a interoperabilidade dos dispositivos residenciais, utilizando uma arquitetura orientada a serviços. Com essa proposta, os dados gerados por um determinado dispositivo residencial serão disponibilizados por meio de serviços Web para qualquer aplicação que esteja interessada nesses dados.

Diante desse cenário, uma arquitetura simples e flexível para monitoramento e controle, voltado para dispositivos residenciais, é formada pelas premissas que motivam o trabalho proposto.

1.3 Objetivos

Nesta seção, serão apresentados os objetivos deste trabalho, os quais se apresentam divididos em objetivo geral e objetivos específicos.

1.3.1 Objetivo Geral

Propor e desenvolver um sistema de automação residencial para integração de dispositivos residenciais com aplicações de controle e monitoramento, tanto local quanto remotamente através da Internet, com o uso de tecnologias e padrões de comunicação sem fio através de um sistema *gateway* orientado a serviços.

1.3.2 Objetivos Específicos

- Identificar tecnologias de comunicação de dispositivos domésticos existentes na literatura e no mercado atual, elaborando uma análise comparativa de suas principais características;
- Identificar e avaliar as tecnologias de desenvolvimento de *Web Services*, apropriadas para o domínio de estudo;
- Definir e desenvolver um *gateway* capaz de integrar dispositivos residenciais a outros dispositivos de uma rede IP através de uma solução baseada em serviços Web;
- Desenvolver uma aplicação que ajude a provar a validade do conceito proposto, capaz de acessar os dispositivos residenciais conectados através do *gateway*.

1.4 Delimitações do Escopo

O escopo deste trabalho será focado na formulação e apresentação de um sistema *gateway* orientada a serviços para interconexão de dispositivos residenciais, não abordando tópicos como segurança dos serviços Web e dos dados transmitidos na rede doméstica e a conservação de energia.

1.5 Contribuições Esperadas

A principal contribuição desse trabalho será a proposição de um sistema *gateway* para integração de dispositivos residenciais com outras redes IP. Além da proposta do *gateway*, algumas outras contribuições secundárias serão feitas por esse trabalho:

- A disponibilidade das funcionalidades dos dispositivos residenciais através de serviços Web, facilitando a implementação de aplicações de controle de forma mais simples e abstrata;
- A definição de um modelo de protocolo que possa ser compartilhado pelos fornecedores de tecnologia residencial para integração dos dispositivos.

1.6 Organização do Trabalho

As demais partes desta dissertação estão organizadas como segue:

O Capítulo 2 apresenta os conceitos teóricos necessários à compreensão do trabalho, tais como as características de uma arquitetura orientada a serviços, serviços Web e as tecnologias de implementação utilizadas, bem como as tecnologias de comunicação, dando ênfase à comunicação sem fio baseada em rádio frequência como Bluetooth, Wi-Fi e ZigBee.

O Capítulo 3 apresenta os trabalhos relacionados que abordaram em seu contexto assuntos relacionados à integração de dispositivos físicos através de *middleware/gateway*, orientação a serviços e serviços Web, classificando-os quanto aos requisitos relacionados, caracterizando as tecnologias abordadas nos trabalhos pesquisados.

O Capítulo 4 apresenta um sistema *gateway* para automação de dispositivos residenciais, abordando o cenário de atuação do sistema, os requisitos de projeto do sistema de automação, os componentes que compõem o sistema e suas funcionalidades.

O Capítulo 5 descreve informações sobre os hardwares/softwares utilizados para montagem do ambiente de teste, descrevendo os protótipos desenvolvimentos, a plataforma de desenvolvimento e a linguagem de programação. Os componentes apresentados neste capítulo serão utilizados nos experimentos descritos no capítulo seguinte.

O Capítulo 6 apresenta os experimentos funcionais realizados no sistema de automação proposto, considerando as funcionalidades essências ao bom funcionamento do sistema, tais como comunicação, desempenho, acesso aos serviços Web.

O Capítulo 7 traz as considerações finais em relação ao trabalho desenvolvido e algumas sugestões para trabalhos futuros. Apresentando de forma analítica os resultados obtidos durante o desenvolvimento deste projeto, dificuldades e sugestões de melhorias.

Capítulo 2 – Fundamentos Teóricos

Este capítulo tem como objetivo apresentar a conceituação teórica de tópicos abordados no sistema *gateway* proposto. A seção inicial conceitua e caracteriza a arquitetura orientada a serviços (SOA). Esse tópico é importante para o entendimento da arquitetura usada pelo *gateway* para disponibilizar as funcionalidades dos dispositivos residenciais às aplicações de controle e monitoramento.

A seção seguinte apresenta as características dos serviços Web, abordando as tecnologias WS-* e REST usadas no desenvolvimento de serviços Web. Ao final, é realizada uma comparação entre essas tecnologias. Este tópico está relacionado à escolha da tecnologia de serviços Web que melhor se adequada à dispositibilização dos serviços oferecidos pelos dispositivos residenciais.

Na última seção, serão apresentadas as tecnologias de comunicação sem fio, devido à flexibilidade, mobilidade e escalabilidade, abordando as principais características técnicas de padrões de comunicação sem fio presentes em um ambiente residencial, tais como Wi-Fi, Bluetooth e ZigBee, ao final, é feita uma análise comparativa dos padrões. Em relação à solução proposta, este tópico é importante para a escolha de um padrão de comunicação único para interconectar os dispositivos residenciais.

2.1 SOA – Service Oriented Architecture

A arquitetura orientada a serviços (SOA) possui uma interessante abordagem para construção de sistemas distribuídos que atendem às novas tendências de um mercado ágil e flexível. SOA foi definido como um paradigma para organização e utilização de competências distribuídas que estão sob controle de diferentes domínios proprietários (MACKENZIE, 2008).

O paradigma SOA soluciona a lacuna existente entre o negócio e a tecnologia de informação (TI), principalmente, no sentido semântico, no qual as pessoas de negócio e de TI aparentemente falam e pensam em linguagens totalmente diferentes. O objetivo do SOA é estruturar grandes sistemas distribuídos baseados nas abstrações de regras e funções de

negócios. A proposta SOA aceita, de uma forma única, manter a flexibilidade em grandes sistemas distribuídos, suportar a heterogeneidade, a descentralização e a tolerância à falha. A arquitetura orientada a serviços é um paradigma, é uma nova forma de pensar e desenvolver sistemas (JOSUTTIS, 2007).

Serviços são módulos de negócio ou funcionalidades das aplicações que possuem interfaces expostas e que são invocados via mensagens (MARKS; BELL, 2006). Em SOA, recursos de software são empacotados como serviços bem definidos, autocontidos, provendo funcionalidades padrões do negócio, independentes do estado ou contexto de outros serviços.

Os serviços são expostos via suas interfaces e o projetista do cliente não tem acesso direto ao desenvolvimento do serviço, apenas à sua interface, mantendo uma camada de abstração clara que encapsula detalhes técnicos (MARZULLO, 2009).

Para que seja possível a interação entre os serviços que implementam as competências e as necessidades das entidades, é necessária a colaboração dos serviços através de papéis, conforme apresentado na Figura 1. Segundo Endrei (2004), essa arquitetura possui três papéis de serviços: o Consumidor, o Provedor e o Registro.

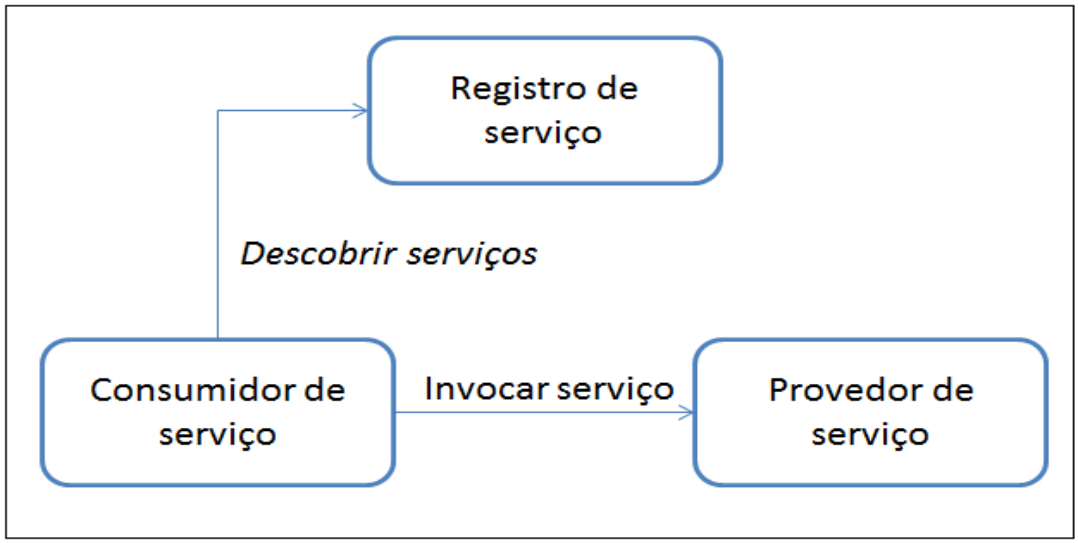

Figura 1 – Papéis SOA.
Fonte: Adaptado de (ENDREI, 2004).

- Consumidor: aplicação ou outro serviço que necessita de um serviço e o executa de acordo com o contrato de interface;

- Provedor: entidade que aceita e executa as requisições dos consumidores. Ele publica seus serviços e os registra por meio de um contrato de interface;
- Registro: contém um repositório de serviço em que o consumidor realiza uma localização dinâmica do serviço requisitado, publicado interiormente por um provedor. Esta localização é realizada por meio de consulta no serviço de registro e, depois de encontrado, o consumidor conecta e invoca o serviço solicitado. A colaboração entre esses serviços segue o paradigma de encontrar, conectar e invocar.

A arquitetura orientada a serviços pode ser representada através do processo chamado paradigma de "procura-consolida-executa", ilustrado na Figura 2. O processo preconiza que os provedores de serviços registrem informações em um registro central com suas características. O registro é utilizado pelo cliente para determinar as características dos serviços necessários, caso esteja disponível no registro central, como, por exemplo, por um catálogo de serviços, o cliente poderá utilizá-lo, sendo este oficializado através de um contrato que enderece este serviço.

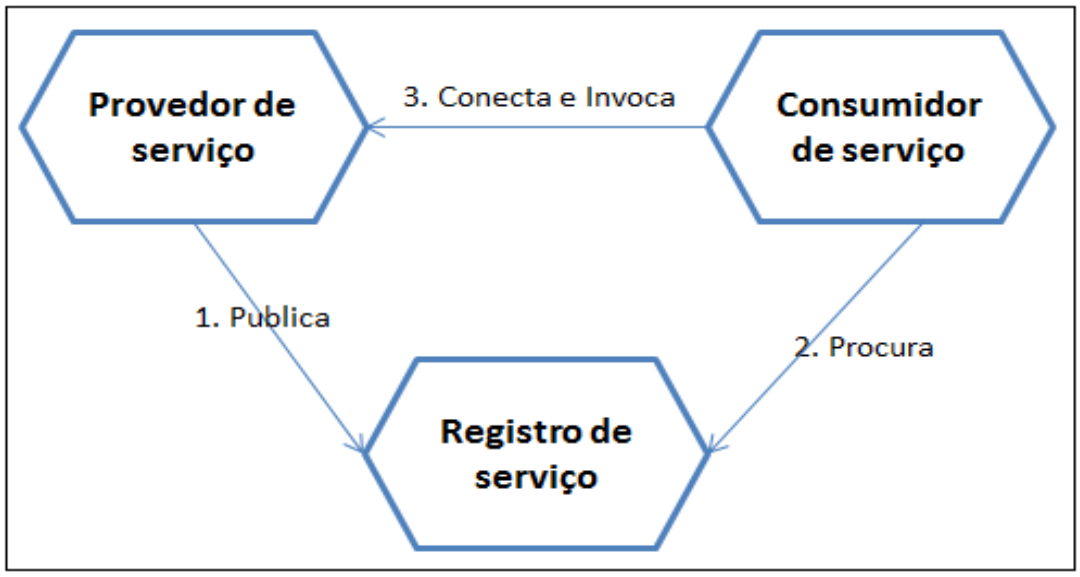

Figura 2 – Processo de Integração SOA.
Fonte: Adaptado de (ENDREI, 2004).

O registro de serviços tem o papel de fazer o intermédio entre o consumidor (cliente) e o provedor de serviços. Ele identifica os serviços requisitados pelo cliente. Dessa

maneira o cliente não precisa saber sempre qual é a localidade do serviço desejado, o que diminui a dependência entre o consumidor e o provedor de serviço.

2.2 Web Services

A definição de serviços Web (*Web Services*), de acordo com a W3C, é um sistema de software responsável por proporcionar a interação entre duas máquinas através de uma rede. Para possibilitar essa interação uma interface descrita em um formato específico, WSDL (*Web Services Description Language*), permite que sistemas interajam com um *Web Service* usando essa interface e enviando mensagens SOAP (*Simple Object Access Protocol*) ou utilizando outros protocolos. As mensagens SOAP basicamente são documentos XML serializados, seguindo o padrão W3C, enviados em cima de um protocolo de rede.

Com esta tecnologia, torna-se possível que aplicações diferentes interajam entre si e sistemas desenvolvidos em plataformas diferentes se tornem compatíveis. Os *Web Services* são componentes que permitem que aplicações enviem e recebam dados em formatos variados. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, como é o caso do formato XML.

Uma característica fundamental dos *Web Services* diz respeito à possibilidade de utilização de diferentes formas de transmissão de dados pela rede. Logo, a arquitetura de *Web Services* pode trabalhar com diversos protocolos, tais como: HTTP, SMTP, FTP, RMI/IIOP ou protocolos de mensagem proprietários. Segundo Potts e Stephen (2003), as primeiras versões das especificações de *Web Services* ofereciam apenas o HTTP como meio de transporte de dados (mensagens SOAP XML) e comunicação entre clientes e serviços, sendo ainda o mais utilizado atualmente.

2.2.1 *Web Services* WS-*

No final da década de 90, milhares de pessoas estavam desenvolvendo sistemas de e-commerce baseados em HTTP e XML. Cada qual com sua própria maneira de implementar segurança, confiabilidade, gerenciamento de transações. Com isso, um caos começou a ser formado: incompatibilidades entre sistemas, dificuldade na manutenção e,

em contrapartida, a necessidade de se criar uma maneira comum de realizar essas tarefas. Tais fatos impulsionaram o surgimento dos padrões WS-*. Hoje, mantidos pela W3C e OASIS (WEERAWARANA, 2005).

Atualmente, a arquitetura WS-* é composta por mais de 20 especificações, sendo as denominadas bases desse conjunto: SOAP, WSDL e UDDI. Além dos citados, existem também os padrões *WS-Notification, WS-Addressing, WS-Transfer, WS-Policy, WS-Security, WS-Trust, WS-ReliableMessaging, WS-Transfer, WS-I Basic Profile, WS-Transaction* entre outros. Uma das principais reclamações diz respeito a esse grande número de especificações que torna a arquitetura WS-* complexa e burocrática, difícil de ser dominada por uma só pessoa. Nos padrões WS-*, as mensagens trocadas entre serviço e cliente consumidor devem ser armazenadas em envelopes SOAP. Esse protocolo de comunicação dita um formato de envio de mensagens entre aplicações (NEWCOMER, 2002).

Para descrever e localizar *Web Services*, é utilizada a linguagem WSDL. Considerada um vocabulário XML, permite que desenvolvedores de serviços disponibilizem informações importantes para a utilização dessas ferramentas. WSDL é, altamente, adaptável e extensível, o que permite a descrição de serviços que se comunicam por diferentes meios, tais como SOAP, RMI/IIOP. UDDI *(Description, Discovery, and Integratio*n) é um protocolo que disponibiliza métodos padrões para publicação e localização de informações sobre *Web Services*, funcionando como um repositório de metadados com informações úteis para a utilização de serviços (CERAMI, 2002).

2.2.1 *Web Services REST*

Na era da Web 2.0, a integração entre aplicações se faz um requisito básico. É cada vez mais freqüente o aparecimento de aplicações que usam conteúdos de mais de uma fonte para criar um novo serviço completo. Esse tipo de sistema é, em parte, proveniente da *Service Oriented Architecture* (SOA), estilo de arquitetura orientado a serviços que possui aspectos fortes de reutilização de código e desacoplamento.

Pensando nesse contexto, os sistemas devem ser capazes de disponibilizar diferentes formatos de dados para uma mesma fonte. Para uma aplicação Web, o conveniente seria retornar HTML ou ainda *JavaScript Object Notation* (JSON), diferentemente de uma

aplicação desktop, na qual o mais indicado seria o retorno de um formato XML. Logo, enfrenta-se a necessidade de disponibilizar serviços de maneira fácil e em diferentes formatos adequados para integração (HTML, XML, JSON, TEXT PLAIN), características estas que podem ser atendidas por sistemas desenvolvidos utilizando o estilo de arquitetura REST, formalizado por Roy Fielding (DIAS NETO; MACHADO, 2007).

2.2.3 *REST e WS-**

Os Serviços Web REST e os Serviços Web WS-* (SOAP, WSDL, etc.) são técnicas de integração de aplicações distribuídas que visam manter o baixo acoplamento entre as partes envolvidas. REST têm sido amplamente aplicados na integração dos dispositivos inteligentes a Web, porque esses princípios parecem ser mais adequados para dispositivos com poucos recursos de hardware (GUINARD; TRIFA, 2009). Porém, ainda é necessária uma comparação mais detalhada entre essas técnicas de integração. Para facilitar a compreensão de tal escolha e obter mais detalhes sobre essa comparação, podem ser vistos os argumentos em (PAUTASSO, 2008).

A primeira diferença entre o REST e o WS-* SOAP está na forma como o protocolo HTTP é empregado. Com REST, o HTTP é utilizado para definir a interação entre o cliente da aplicação e o provedor do recurso. Nesse caso, toda a semântica presente nos quatro métodos (GET, POST, PUT, DELETE) do protocolo HTTP é utilizada na definição da interface do sistema (RUBY; RICHARDSON, 2007). Os WS-* SOAP, por outro lado, utilizam o protocolo HTTP para transportar as mensagens no formato SOAP com o objetivo de integrar aplicações. Nessa abordagem, as mensagens SOAP são adicionadas ao corpo do HTTP para comunicação remota através de firewalls [(GUINARD; TRIFA, 2009), (PAUTASSO, 2008)]. Nos WS-* SOAP, o método POST do HTTP é utilizado na troca de mensagens entre clientes e servidores e a informação sobre qual funcionalidade deve ser executada está presente na mensagem SOAP e não na requisição HTTP. É por esse motivo que se diz que, para os serviços Web SOAP, o protocolo HTTP desempenha funcionalidade de transporte, mesmo ele sendo um protocolo do nível de aplicação. Enquanto as aplicações WS-* SOAP utilizam a Web como meio de troca de mensagens, as aplicações Web *RESTful* são parte da Web a qual é vista como um terreno comum para as aplicações (PAUTASSO, 2008).

Além do HTTP, as mensagens SOAP também podem ser encapsuladas e transportadas dentro de outros protocolos (por exemplo, os protocolos TCP e SMTP podem ser utilizados para esse propósito). Isso é possível porque o SOAP possui um formato próprio de mensagem (baseado em XML), porém ele requer que outro protocolo seja utilizado para transferir essa mensagem. O formato da mensagem SOAP ocasiona um maior consumo de banda do que o ocasionado pelo protocolo HTTP, devido ao tamanho da mensagem SOAP. Esse é um dos motivos para o REST apresentar vantagens para ser utilizado em dispositivos com pouca capacidade de hardware ou restrição de banda de rede disponível (RUBY; RICHARDSON, 2007). Além disso, um formato pré-definido de mensagem força o cliente a tratar aquele tipo de mensagem, caso ele deseje utilizar um serviço. Com REST é possível oferecer diferentes tipos de representações. Assim, um cliente pode, por exemplo, escolher se para ele é mais adequado receber uma representação JSON ou um XML. Porém, fornecer vários formatos exige mais esforço no processo de desenvolvimento, pois diferentes tipos de mensagens precisam ser providos.

Do ponto de vista do acoplamento, tanto REST quanto o SOAP fomentam o desenvolvimento de sistemas distribuídos com acoplamento fraco entre as partes. Porém, avaliar qual das duas abordagens atinge esse objetivo, conseguindo um menor acoplamento, é uma tarefa subjetiva, pois para definir o acoplamento vários aspectos devem ser observados, a saber: tempo/disponibilidade, clareza de localização e evolução do serviço. Geralmente, o termo “fraco acoplamento” é relacionado à capacidade de fazer modificações no provedor do serviço sem afetar o cliente.

Nesse caso, os serviços *Web RESTful* são menos acoplados, pois as operações sobre os recursos não mudam, já que tais operações são baseadas nos métodos do HTTP, os quais se mantêm inalterados, mesmo quando ocorre alguma mudança no serviço. Contudo, quando acontecem alterações nos parâmetros passados nas mensagens, ambos (SOAP e REST) compartilham o mesmo nível de acoplamento. Outra diferença entre REST e SOAP está na forma como essas abordagens utilizam URIs.

Com REST, a URI não é utilizada apenas para identificar o recurso, mas também para encapsular toda a informação necessária para identificar e localizar os recursos sem a necessidade de um registro centralizado. Entre outros benefícios, empregar URIs dessa forma permite que os recursos sejam marcados e que links hipermídia sejam fornecidos

para o cliente de modo que ele interaja com os recursos do sistema. O WS-* também utiliza URI, mas não da mesma forma que em uma abordagem REST, o que acarreta a necessidade de utilização de outros mecanismos para agregar informação ao serviço (PAUTASSO, 2008).

Prosseguindo com as comparações entre os serviços Web SOAP e as aplicações *RESTful*, é possível observar diferenças entre a forma como os clientes consomem os serviços. Com SOAP, os clientes utilizam um documento de descrição formal dos serviços disponíveis, esse documento é o WSDL (W3C). O uso de descritores fornece aos clientes meios que permitem a geração automática de código para consumir esses serviços. Porém, sua utilização pode ocasionar falhas no cliente, caso ocorram modificações no servidor.

As aplicações *RESTful* não precisam utilizar um contrato formal entre o cliente e o servidor. Freqüentemente, os recursos de uma aplicação *RESTful* são descritos de forma textual ou por meio da documentação da API da aplicação. Além dessas abordagens, também surgiram alguns provedores de serviços *RESTful* que oferecem bibliotecas (em diferentes linguagens de programação) para serem usadas pelos clientes que desejam consumir os recursos que o servidor oferece (FERREIRA FILHO, 2013).

Por serem mais simples, por tornarem as aplicações criadas com base em seus princípios parte da Web, permitindo que todos os recursos disponíveis na Web possam ser utilizados para o mundo físico e ainda em razão do menor tamanho das suas mensagens, os princípios REST são considerados mais adequados para serem utilizados na integração de dispositivos com baixa capacidade de hardware na Web.

2.3 Tecnologias de Comunicação Utilizadas em Redes Domésticas

Segundo Callaway (2010) e Watanabe (2010), os projetos de Automação Residencial e redes domésticas têm buscado cada vez mais utilizar tecnologia de comunicação sem fio, devido à instalação, redução de custo final, flexibilidade, mobilidade e escalabilidade.

As várias aplicações residenciais exigem diferentes requisitos como largura de banda, instalação, velocidade e, conseqüentemente, refletem em custos distintos. Para

interagir com todos esses dispositivos remotamente, é necessário colocá-los sob uma simples interface de controle padronizada que interconecta todos numa rede, especialmente numa Rede de Área Doméstica, HAN (*Home Area Network).*

Dentre uma grande variedade de tecnologias de comunicação que podem ser encontradas no mercado, optou-se neste trabalho por analisar a categoria de redes sem fios. As redes sem-fios, como IEEE 802.15 (Bluetooth), IEEE 802.15.4 (ZigBee) e IEEE 802.11x (Wi-Fi), têm crescido significativamente, sobretudo, devido à conveniência de conexão com um ou mais dispositivos usados simultaneamente sem custo adicional, ao baixo custo de produção, à fácil integração entre vários tipos de dispositivos, como telefones celulares, ventiladores e *home theaters*, e ao baixo consumo de energia.

2.3.1 Wi-Fi

Uma das primeiras tecnologias desenvolvidas para redes sem fio foi o Wi-Fi, que corresponde à especificação 802.11x do IEEE e que inclui o protocolo de segurança WEP (*Wired Equivalency Protocol*). Esse protocolo visa proporcionar uma segurança equivalente à da rede com cabeamento. Para se conectar à Internet, utilizando uma rede Wi-Fi, deve-se estar perto de um ponto de acesso (*access point*) ou local público onde a rede seja disponibilizada a todos que utilizem dispositivo móvel, tais como computadores portáteis, *tablet*s, dentre outros.

O Wi-Fi possui várias taxas de transmissão de dados: 11 Mbps (802.11b), 54 Mbps (802.11a e 802.11g). Essa velocidade não garante qualidade de serviço (QoS) e o alcance de sinal varia desde 50 m (ambientes fechados) até 100 m (ambientes abertos) (JARDIM, 2007). A compara os padrões do *Wi-Fi* quanto à velocidade e faixa de freqüência.

Padrão	Velocidade máxima	Faixa de frequência
802.11 b	11 Mbps	2.4 GHz
802.11 a	54 Mbps	5 GHz
802.11 g	54 Mbps	2.4 GHz

Tabela 1 - Tabela comparativa dos padrões Wi-Fi.
Fonte: (JARDIM, 2007).

2.3.2 Bluetooth

Como grande parte das tecnologias *wireless*, o Bluetooth utiliza a banda ISM de 2,4 GHz. Também utiliza a técnica de saltos de freqüência (FHSS), como o 802.11, mas realiza essa operação mais rapidamente, saltando até 1600 vezes por segundo (JARDIM, 2007).

Os dispositivos com Bluetooth criam automaticamente conexões independentes entre outros dispositivos e o usuário não precisa configurar a rede. O Bluetooth permite que até 8 dispositivos sejam conectados diretamente entre si, formando uma *piconet* (pequena rede). Redes maiores podem ser criadas, mas os dispositivos só podem visualizar diretamente outros 7 dispositivos. Essas redes são chamadas *scatternets* (redes de difusão – redes ad-hoc) e é necessário dividi-las em *piconets*. A tecnologia Bluetooth possibilita a transmissão em *half duplex* e *full duplex*. A taxa de saltos de freqüência de 1600 saltos/s possibilita que um dispositivo realize uma transmissão a cada 625µs denominado time-slot. Para implementar o modo *full duplex*, o Bluetooth utiliza a técnica TDD (*Time Division Duplex*) em que o dispositivo envia em um time-slot e recebe no seguinte. Sendo assim, um aparelho Bluetooth envia e recebe pacotes quase que simultaneamente (MILLER, 2001).

Os equipamentos Bluetooth têm capacidade de localizar outros equipamentos próximos, formando as *piconets*. Dentro de uma *piconet*, o dispositivo que transmitiu primeiro é considerado mestre e os outros são os escravos. Essa característica permite que as conexões sejam feitas automaticamente. Os dispositivos que formam as *piconets* determinam um padrão de transmissão que é conhecido apenas pelos componentes da rede (sequência dos saltos de frequências). Isso garante certo grau de segurança nas transmissões e também permite que diferentes redes *wireless* compartilhem o mesmo espaço físico, com menor probabilidade de interferências.

A técnica de saltos de frequências dificulta as escutas clandestinas, mas não as impede. Assim, os idealizadores do Bluetooth acrescentaram três níveis de segurança à tecnologia:

- Sem segurança (*Non-Secure*) – a transmissão de dados é feita sem nenhum tipo de segurança. Normalmente é utilizada em transmissões de dados irrelevantes.

- Segurança estabelecida no nível do serviço (*Service-Level Enforced Security*) – o dispositivo Bluetooth inicia os procedimentos de segurança depois de estabelecida a conexão.

- Segurança no nível de link (*Link-Level Enforced Security*) – esse modo é o mais seguro. Os procedimentos de segurança são iniciados antes do estabelecimento da conexão. Os procedimentos de segurança incluem as chaves de códigos, a autenticação dos dispositivos e a criptografia dos dados transmitidos.

2.3.3 ZigBee

A tecnologia ZigBee foi desenvolvida como solução em redes sem fio com baixo consumo de energia, baixa taxa de transmissão, segurança e confiabilidade. Os equipamentos ZigBee utilizam as faixas de freqüência não licenciadas (faixas ISM), possuem taxas de transmissão de até 250 kbps e alcance de até 80m (FARAHANI, 2008).

A principal característica da tecnologia ZigBee é o baixo consumo. O ZigBee foi desenvolvido para permitir um longo tempo de uso de baterias. O baixo consumo permite o uso do ZigBee em sistemas remotos de medição, sensores sem fio e câmeras de segurança remotas, além de sistemas portáteis (CHENGBO, Y. et al, 2009). O consumo é reduzido porque os dispositivos ZigBee podem ser colocados em modo de espera (*stand-by*) por horas e os pacotes de dados da comunicação são pequenos. O uso de pacotes de dados pequeno também resulta em menor probabilidade de erros de transmissão, tornando a rede com dispositivos ZigBee mais confiável (SHIZHUANG; JINGYU.; YANJUN, 2007).

O ZigBee utiliza as mesmas freqüências de outros dispositivos *wireless* (Bluetooth, 802.11b etc.) e isso pode acarretar problemas de interferência. Na verdade, os dispositivos ZigBee têm menor taxa de erros de transmissão em locais onde existem outras redes, porque a transmissão de dados ZigBee é feita em pequenos pacotes. Como existe a possibilidade de se utilizar várias tecnologias *wireless* no mesmo espaço físico, os comitês IEEE 802.10 e IEEE 802.15.2 estão empenhados em corrigir problemas da coexistência (TANENBAUM, 2003).

As principais áreas de aplicação para o ZigBee são na indústria, comércio, automações residencial e predial, monitoramento médico de pacientes, controle de acesso para veículos. Uma rede ZigBee pode conter até 255 dispositivos ativos (1 coordenador e

254 dispositivos) e cada coordenador de rede pode se conectar a outros coordenadores visando a criar redes maiores (por exemplo, uma rede com 255 dispositivos ativos e suporte para 16 canais na banda pode chegar a mais de 4000 nós em uma única rede com alta confiabilidade) (FARAHANI, 2008).

As topologias de rede suportadas pelo ZigBee são: ad hoc, mestre-escravo e *peer-to-peer* (ponto a ponto). Qualquer dispositivo na rede pode assumir o papel de coordenador (mestre). A topologia ad hoc permite que dispositivos se conectem à rede por meio do dispositivo mais próximo, como se esse dispositivo fosse um coordenador da rede. Em relação à segurança, o ZigBee utiliza o mecanismo de criptografia *Advanced Encryption Standard* (AES128) (JARDIM, 2007). As principais características do ZigBee são:

- Baixo consumo de energia e baixo custo;
- Menor taxa de transmissão, se comparado às outras tecnologias – 250 Kbps;
- Redes maiores, com até 255 elementos ativos;
- Alcance de até 30 m (típico).

2.3.4 Estudo Comparativo das Tecnologias de Comunicação

Na Tabela 2, é mostrada uma comparação entre as tecnologias de comunicação que podem ser usadas em uma rede doméstica, levando em conta a aplicação, frequência, modulação, taxa de transmissão, alcance, consumo, custo estimado e vida útil de baterias.

O Bluetooth é indicado em aplicações cujo número de equipamentos é pequeno, mas o tráfego de dados pode ser grande (com taxas de até 1 Mbps). As redes com dispositivos Bluetooth podem ser facilmente criadas, permitindo, assim, que a tecnologia seja utilizada em escritórios em que um *notebook* ou um celular podem acessar diretamente a rede. (MILLER, 2001).

O Wi-Fi também pode ser utilizado em aplicações de escritório. Possui alta taxa de transmissão (11 Mbps), com conexões confiáveis, com longo alcance (100 metros) e pode ser integrada facilmente com uma rede Ethernet. As desvantagens são o custo dos equipamentos e o elevado consumo de energia, o que poderia ser um problema para os dispositivos residências mais simples, movidos à bateria.

O protocolo ZigBee é um padrão de comunicação sem fio de muito baixo custo, baixo consumo de energia e pilha protocolar de implementação simplificada

[(FARAHANI, 2008)]. Assim, o ZigBee tem aplicação direta em equipamentos utilizados nas automações residenciais, prediais e industriais. Uma grande rede ZigBee pode ser criada com sensores remotos, controladores de ar condicionado e iluminação, câmeras de segurança e outros dispositivos, sem necessidade de utilizar cabos para as conexões de sinais.

	Bluetooth	**IEEE 802.11b**	**ZigBee**
Aplicação	Substituição de cabos	Redes corporativas ou universitárias	Redes de sensores, automação
Freqüência e Modulação	2,4 GHz FHSS	2,4 GHz DSSS	2,4 GHz DSSS
Taxa de transmissão	1 Mbps	11 Mbps	250 kbps
Alcance	10 metros	100 metros	80 m
Consumo	Baixo	Alto	Baixo
Custo estimado	$5	$30	$5
Vida útil de baterias	5 dias	< 1 dia	>1000 dias

Tabela 2 - Comparação entre as tecnologias Bluetooth, IEEE 802.11b e ZigBee.
Fonte: (BRAY; STURMAN, 2001).

Os tipos de dados que circulam dentro de uma rede de sensores e atuadores que incluem automação residencial são pequenos pacotes que controlam dispositivos ou que obtêm seu *status*. Na maior parte do tempo, para muitas aplicações como detector de fumaça ou dispositivos sem fio de segurança de casas, os dispositivos estão em modo de espera dormindo (*sleeping*) e somente enviam uma curta informação se um evento ocorrer. Os principais requisitos, para os dispositivos em cada tipo de rede, são: consumo extremamente baixo de energia, habilidade de permanecer inativo por um longo período, simplicidade e baixo custo (DING, 2006).

Diante desse cenário, o protocolo ZigBee apresenta-se como uma tecnologia baseada em padrões que endereça as necessidades da maioria das aplicações de redes domésticas. A aplicação deste padrão tem crescido em diversas áreas, havendo realizações de pesquisas até mesmo na área de comunicação de voz sobre IP (VoIP - *Voice Over IP*), bem como sobre o protocolo ZigBee (WANG; SOHRABY, 2008).

As áreas principais nas quais o padrão ZigBee pode ser aplicado são: medicina, residencial, controle e monitoramento industrial. Alguns exemplos de aplicação incluem iluminação residencial e comercial, detector de fumaça e monóxido de carbono, aquecimento, ventilação e ar condicionado (HVAC - *Heating, Ventilation and Air Conditioner*), controle ambiental, segurança residencial, sensoriamento e controle médico, controle remoto em geral e automação industrial (EGAN, 2005).

2.4 Resumo

Este capítulo teve como objetivo introduzir conceitos da arquitetura orientada a serviços (SOA) e tecnologias de comunicação utilizadas em redes domésticas. No que tange à arquitetura, abordou-se alguns conceitos sobre as tendências tecnológicas desenvolvidas para suportar interoperabilidade entre máquinas, fazendo com que aplicações, desenvolvidas nas mais diversas plataformas, interajam entre si. Sob essa perspectiva, abordaram-se dois tipos de serviços Web: WS-* e REST, ressaltando os respectivos conceitos e características, e posteriormente, apresentou-se um estudo comparativo dos padrões.

Após o estudo das tecnologias de interoperabilidade, apresentaram-se as tecnologias de comunicação utilizadas em redes domésticas e, tendo em vista a variedade de tecnologias de comunicação que se encontra no mercado, foi dado um enfoque, apenas, nas redes sem fio e nos padrões IEEE 802.15 (Bluetooth), IEEE 802.15.4 (ZigBee) e IEEE 802.11x (Wi-Fi). Para cada padrão apresentado, foi dada uma abordagem sucinta sobre os conceitos, topologias, qualidades de serviço (QoS) e segurança. Além disso, mostrou-se uma comparação entre as tecnologias, levando em consideração a aplicação, frequência, modulação, taxa de transmissão, alcance, consumo, custo estimado e vida útil de baterias.

No próximo capítulo, serão analisados alguns trabalhos referentes à solução *gateway* proposta, cuja seleção foi realizada segundo os tópicos relacionados à arquitetura orientada a serviços, integração de dispositivos físicos e serviços Web. Durante a exposição dos trabalhos, serão apresentadas as semelhanças e divergências da proposta *gateway* desta monografia.

Capítulo 3 – Trabalhos Relacionados

Serão apresentados neste tópico alguns trabalhos que abordaram em seu contexto assuntos relacionados à integração de dispositivos físicos através de *middleware/gateway*, orientação a serviços e serviços Web. Durante a descrição dos trabalhos, serão destacados os pontos em comum com a proposta desta dissertação. A seção seguinte aborda as tecnologias utilizadas nos trabalhos apresentados, descrevendo suas vantagens e desvantagens.

Na seção a seguir, alguns requisitos referentes à integração de sistemas heterogêneos são descritos e posteriormente os trabalhos relacionados são caracterizados, resultando em um quadro comparativo.

3.1 Análises Existentes

Durante a pesquisa, identificaram-se diferentes abordagens feitas na integração de dispositivos com internet através de uma arquitetura orientada a serviços. Foram selecionados alguns trabalhos que apresentam semelhança como integração através de *middleware/gateway*, a orientação a serviços e a utilização de serviços Web. A seguir, será apresentada uma breve visão sobre as abordagens selecionadas e uma comparação das características e desafios comuns envolvidos com a proposta desta dissertação, que será referenciada como BeeHome.

BeeHome, solução *gateway* orientada a serviços, proposta deste trabalho cujo objetivo principal é apresentar um sistema para integração entre os dispositivos residenciais e a Internet, através de uma interface baseada em serviços Web e implementada utilizando tecnologias *RESTful*. O *gateway* funciona como uma ponte entre as aplicações de controle e os dispositivos residenciais. Uma característica comum aos dispositivos físicos é o perfil. Cada dispositivo possui um perfil que descreve as funcionalidades ao *gateway* o qual faz uso de um repositório de perfil, localizado na Web para acessar a estrutura das funcionalidades dos eletroeletrônicos.

Othman et al.(2007) propuseram uma solução utilizando o paradigma SOA, a fim de tornar transparente e flexível a comunicação entre os consumidores de serviços e os dados

disponibilizados por dispositivos físicos. Entre as principais características, destaca-se o fato de a comunicação ser realizada diretamente com cada dispositivo da rede, através de serviços Web. Dessa forma, cada dispositivo traz internamente a descrição de suas funcionalidades em um arquivo WSDL, não fazendo uso de *middleware* para a comunicação. Vale resslatar que, nesta obra, não foram implementados os serviços de registro e de publicação.

Apesar de objetivo semelhante à integração de dispositivos físicos, a proposta desta dissertação se diferencia por não permitir o acesso direto aos dispositivos da rede. Para tanto, um *gateway* é utilizado com o objetivo de disponibilizar as aplicações clientes através de serviços Web referentes aos dispositivos, ilustrado na Figura 3. Outra diferença está no fato da proposta BeeHome não fazer uso de tecnologias tradicionais de implementação SOA (WSDL, SOAP, UDDI), optando pelo desenvolvimento de interfaces *RESTful*.

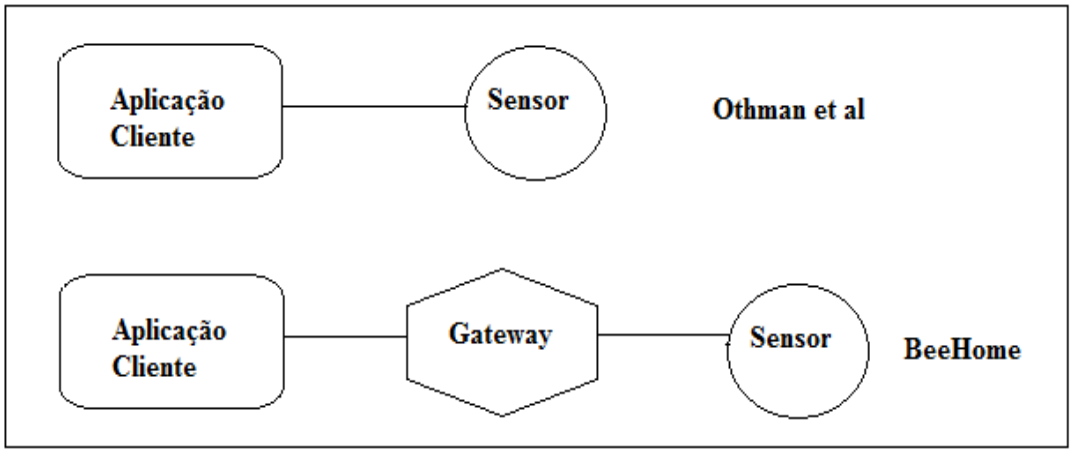

Figura 3 – Diferenças entre o trabalho de Othman e a proposta BeeHome
Fonte: o autor.

Priyantha et al (2008) investigaram o uso de serviços Web em sensores a fim de organizar de forma estruturada suas funcionalidades e seus dados, tornando-os acessíveis. Além disso, os autores avaliaram o *overhead* e o consumo de recursos dos sensores ocasionados pela adoção WS-* e por uma implementação adaptada da pilha TCP/IP utilizada nos sensores para que fossem acessados através da internet. Os serviços dos sensores foram descritos utilizando WSDL, e o acesso a esses serviços foi realizado

utilizando o protocolo HTTP, bem como os métodos *URL encoding* e *URL replacement* seguindo uma abordagem WS-*.

Quando fosse necessário passar uma grande quantidade de parâmetros ou quando esses parâmetros fossem complexos, uma estrutura XML era adotada. Os autores utilizaram um servidor HTTP embarcado e demonstraram que é possível utilizar esse tipo de servidor mesmo em dispositivos com poucos recursos de hardware. Apesar de este trabalho demonstrar o interesse em integrar sensores à Web, ele se diferencia da já citada proposta BeeHome, ao permitir o acesso direto aos sensores e ao utilizar protocolos WS-* na disponibilização de serviços Web.

Na proposta BeeHome, as aplicações de controle não possuem acesso direto aos dispositivos físicos, o que é somente possível através do sistema *gateway*. Outro ponto é a utilização do padrão *RESTful* para disponibilização dos serviços Web. A interface *RESTful* possui características mais adequadas para sistemas que irão executar em dispositivos com recursos limitados (RUBY; RICHARDSON, 2007). Os eletrodomésticos atuam como provedores de serviços Web, podendo ter seus recursos utilizados da mesma forma que qualquer outro recurso Web. Além disso, esta dissertação foca na integração de dispositivos físicos à Web e não apenas em utilizar uma técnica de integração com um sensor isolado em que o acesso é realizado diretamente ao sensor.

Glombitza et al (2010) propuseram a utilização de um método autodescritivo e automático, baseado em tecnologias de serviços Web para descoberta de serviços de sensores. Eles buscaram adequar esses protocolos aos recursos dos sensores, utilizando compressão do descritor WSDL e um protocolo de descoberta de serviços a fim de permitir que sistemas back-end de TI se comunicassem com os serviços Web presentes nos sensores. Os autores ainda apresentaram uma estrutura para conversão entre os protocolos dos sistemas finais e os protocolos utilizados nos sensores. Essa conversão foi realizada com base nas descrições dos serviços Web desses dispositivos. Esses autores utilizaram tecnologias de serviço Web e uma arquitetura SOA. O sistema ficou acoplado ao protocolo desenvolvido, não permitindo a integração com outros dispositivos com diferentes protocolos.

Apesar de ter um objetivo semelhante ao trabalho (GLOMBITZA, 2010), a presente proposta diferencia-se por utilizar uma interface *RESTful* para os serviços Web na

arquitetura SOA. Além disso, a comunicação com os dispositivos físicos são baseados em Perfil, o que permite uma automatização no reconhecimento dos dispositivos.

Delicato (2010) propôs um *middleware* genérico para integração de dispositivos de uma rede de sensores sem fio (RSSF) adotando tecnologias de serviços Web, SOAP e SOA. Nesse trabalho, foi feito um levantamento das características dos dispositivos (sensores) e foram listados os requisitos de um *middleware*, cujo proposto ofereceu abstrações para serviços internos da rede e para serviços da RSSF no *middleware*. Ele oferecia a interface de acesso aos serviços da RSSF e tais serviços foram descritos para os usuários finais por meio de arquivos WSDL. Em seu trabalho, Delicato (2010), o middleware precisa fazer parte da estrutura dos sensores, conforme ilustrado na Figura 4.

Esse trabalho se assemelha com a presente proposta em relação à observação dos requisitos para integração de dispositivos físicos e dos serviços básicos expostos pelo *gateway*. A diferença está nos meios adotados para que as redes fossem utilizadas em diferentes aplicações. Enquanto a autora utilizou SOAP, este trabalho utiliza REST para disponibilizar as funcionalidades das RSSF aos clientes finais na forma de serviços Web. Dessa forma, os recursos das redes podem ser beneficiados por todas as tecnologias e padrões da Web, facilitando o uso, visto que a Web é amplamente utilizada e oferece meios que permitem que até usuários com pouco conhecimento construam suas próprias aplicações.

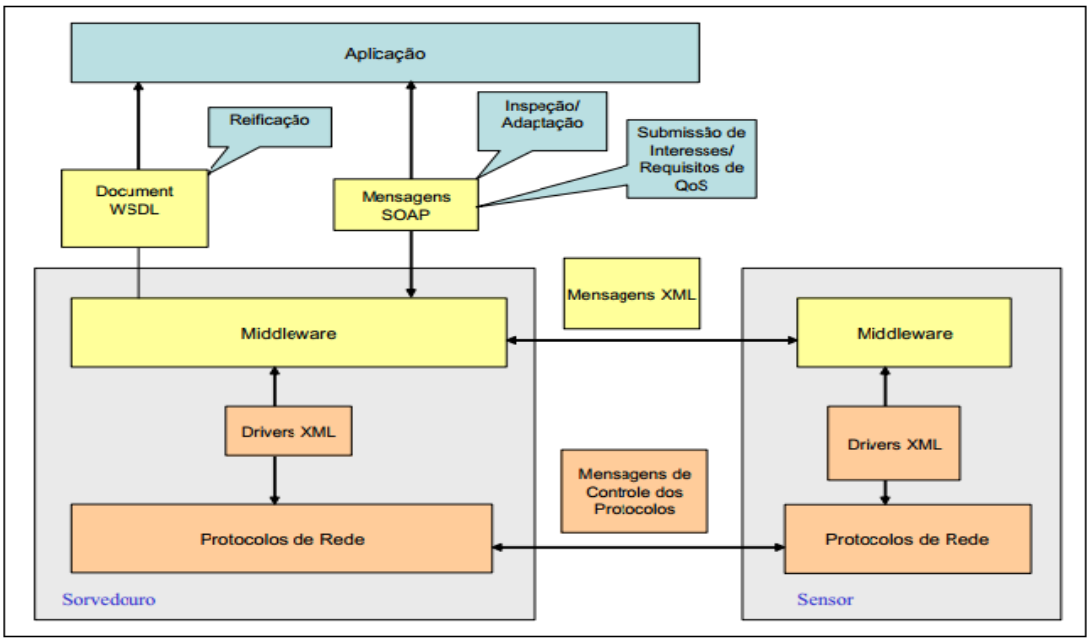

Figura 4 – Middleware proposto por Delicato
Fonte: (DELICATO, 2010)

A estrutura dos sensores é outro ponto que se diferencia. Em (DELICATO, 2010), o sensor deve ter em sua estrutura o *middleware* e os *drivers* XML, enquanto nesta proposta (BeeHome), o sensor não precisa ter outros softwares estranhos instalados na sua estrutura. O protocolo de comunicação é baseado em Perfil que possui a estrutura armazenada em Repositório na Web. Essa necessidade da presença do middleware nas estruturas dos sensores torna essa solução fortemente acoplada ao middleware.

Delicato et al. (2006) propuseram uma arquitetura para RSSF utilizando SOA e *mashups*. Os *mashups* são sites personalizados ou aplicações web que usam conteúdos de mais de uma fonte para criar um novo serviço completo. Tal arquitetura possuía três camadas. A primeira camada é a camada de provisão de dados. Nela, os sensores atuam como provedores de dados brutos enviando os dados dos sensores para o *middleware*. A segunda camada é a camada de extração e interoperabilidade de dados. Nessa camada**,** o middleware atua como provedor de serviço para as aplicações dos usuários fornecendo uma interface Web segundo a tecnologia SOAP. A terceira camada é a camada de composição. Nesta última os autores propõem o uso de *mashups* Web para integrar informação de diversas RSSFs. Esse trabalho não apresentou um estudo de caso, porém a arquitetura proposta se assemelha com a que é proposta nesta dissertação, que tem o gateway como provedor dos serviços dos sensores.

Apesar de não fazer parte do escopo deste trabalho (BeeHome), a arquitetura proposta oferece suporte através de serviços Web às aplicações para implementação de mashups. Entre as diferenças, em (DELICATO, 2006), seguiu uma concepção orientada a serviços com a utilização do protocolo SOAP, enquanto que a presente proposta adota uma implementação *RESTful* SOA, integrando os princípios REST com a estrutura SOA, oferecendo mais vantagens para dispositivos embarcados, permitindo que os serviços das redes sejam mais facilmente utilizados em aplicações Web, diferenciando-se pela forma como as funcionalidades dos sensores são disponibilizadas na Web (GUINARD, 2009).

Avilés e García (2009) propuseram uma arquitetura orientada a serviços que oferecesse uma abstração de alto nível para o desenvolvimento de aplicações de controle. Isso permitiu que desenvolvedores de aplicativos pudessem acessar dispositivos com sensores através de seus aplicativos, usando uma API orientada a serviços, em uma linguagem de programação de sua escolha. A principal vantagem foi evitar que os desenvolvedores de aplicativos lidem com os detalhes técnicos de baixo nível de hardware e de comunicação para obter os dados dos sensores. A proposta (AVILÉS; GARCÍA, 2009) pode ser usada para integrar aplicativos da Internet com dispositivos físicos e permitir às aplicações de controle coletar informações dos sensores, possuindo mecanismos de registro e descoberta dos componentes da rede. Além disso, há um componente que age como se fosse um *gateway*, uma ponte entre os dispositivos e as aplicações externas. Serviços Web foram utilizados para permitir o acesso aos dispositivos.

A proposta BeeHome compartilha pontos em comum, como, por exemplo, o uso de serviços Web para oferecer uma abstração da complexidade dos dispositivos residenciais aos desenvolvedores de aplicações clientes (controle e monitoramento). Diferenças, entretanto, estão presentes nas funcionalidades do *gateway*. Na proposta desta dissertação, o *gateway* oferece serviços de configuração e controle dos dispositivos residenciais, além de disponibilizar esses serviços de forma dinâmica às aplicações clientes, funcionalidades essas não encontradas no trabalho (AVILÉS; GARCÍA, 2009). Outra diferença é a implementação da comunicação através de serviços baseados em Perfil.

Barnaghi et al. (2010) propõem um mecanismo para mediar a troca de dados entre plataformas de sensores heterogêneas e aplicações web e serviços através de um *gateway* que liga diferentes RSSF com aplicações web. Esta troca é feita de forma seria de forma

única? (o que você quer dizer mesmo) por meio de serviços Web. Este *middleware* foi projetado com base no perfil de dispositivos para Web Services (DPWS), que é um subconjunto da tecnologia de *Web Services*; DPWS foi concebido para orientar os dispositivos com recursos limitados e é constituído por componentes, tais como *WS-Discovery e WS-Eventing*. *WS-Discovery* permite recursos *plug-and-play* para os sensores. *WS-Eventing* permite a troca de mensagens assíncronas entre nós sensores através de um mecanismo de publicação. DPWS é usado no *middleware* como base para reduzir a sobrecarga imposta por tecnologias tradicionais de serviço Web. Por exemplo, em vez de grandes mensagens XML, são usadas técnicas de codificação binária.

As mensagens são codificadas, antes de serem transferidas para reduzir a largura de banda. Além disso, o *WS-Eventing* é usado para reduzir a carga na rede de comunicação, na qual apenas os eventos ou as informações necessárias são enviados. O acesso aos dispositivos com IP é feito através do protocolo 6loWPAN.

Este trabalho se diferencia da presente proposta (BeeHome) pelo fato de utilizar outros protocolos (6loWPAN) e componentes WS-* para realizar a comunicação com os sensores e aplicações clientes. O trabalho desta dissertação faz uso de interfaces *RESTful* e protocolos nativos dos sensores para comunicação, o que torna possível que dispositivos com sensores equipados com pouca memória e poder de processamento possam participar da interação dos serviços.

3.2 Características das tecnologias abordadas

Entre os trabalhos relacionados, algumas características mostraram-se comuns, como o uso de serviços Web no acesso às funcionalidades dos dispositivos físicos, facilitando o processo de desenvolvimento de aplicações clientes. Outra característica é o uso da linguagem WSDL para descrever os serviços dos dispositivos, normalmente atrelado à estrutura interna dos referidos serviços. Tais características permitem-lhes oferecer serviços de composição, padronização, registro e publicação; heterogeneidade do ambiente e integração transparente de aplicações clientes.

No entanto, o uso de WSDL não oferece aos consumidores de serviços uma forma eficaz dos serviços publicados, pois com a adição de novos dispositivos à rede doméstica,

faz-se necessário a reformulação da aplicação cliente para adesão aos novos serviços oferecidos pelo dispositivo adicionado. O principal desafio desta abordagem é que ela introduz um nível de acoplamento entre o cliente e o serviço. Em cenários de SOA, em que centenas de serviços e clientes estão envolvidos, esse nível de dependência serviço-cliente costuma ser a causa de sérios desafios de gerenciamento de serviços e controle de versões (RODRIGUEZ; DEMSAK, 2012).

Na a Figura 5, é ilustrado um cenário no qual há incompatibilidade de versão do WSDL, ocasionando, portanto, que alguns clientes tivessem os seus serviços desconectados.Conseqüentemente, o uso de WSDL compromete os serviços de descoberta e mecanismos de integração.

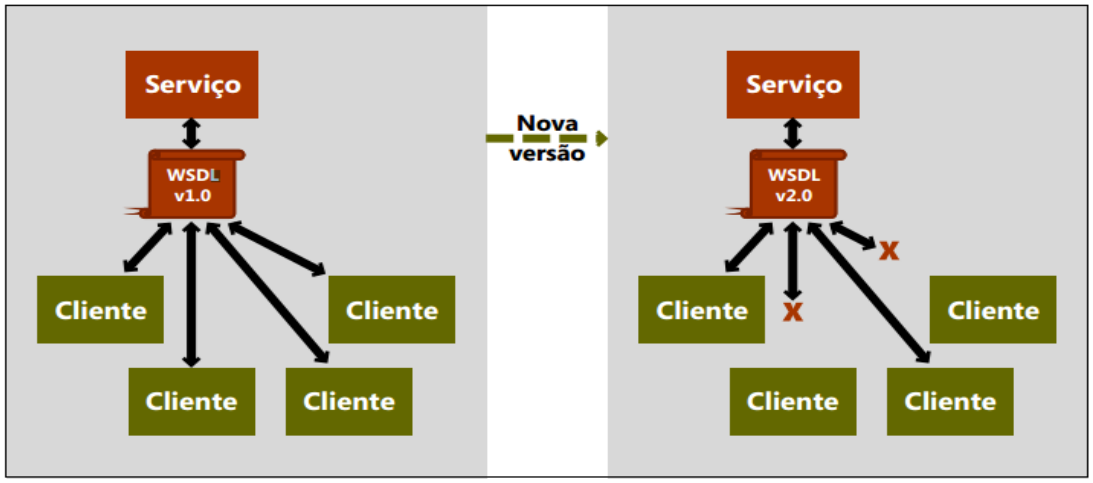

Figura 5 – Dependência de WSDL.
Fonte: (RODRIGUEZ; DEMSAK, 2012).

Em relação à utilização da tecnologia WS-*, a interoperabilidade é um aspecto desafiador, já que diferentes conjuntos de ferramentas de *Web Service* implementam diferentes protocolos WS-*, diferentes versões dos mesmos protocolos, ou mesmo diferentes aspectos da mesma especificação. Além disso, a adoção de WS-* foi fundamentalmente reduzida para os ecossistemas .Net e Java, tornando completamente impossível aproveitar os paradigmas emergentes de programação como linguagens dinâmicas ou funcionais em soluções de SOA(RODRIGUEZ; DEMSAK, 2012).

Alguns trabalhos [(DELICATO, 2010), (DELICATO, 2006), (AVILÉS; GARCÍA, 2009)] optaram por ter previamente armazenados a estrutura dos *middlewares drives* correspondentes ao funcionamento dos sensores, sem mecanismos de atualização. Tal prática compromete o requisito de escalabilidade, pois não oferece suporte a novas

estruturas de sensores. Quanto ao requisito eficiência, o uso do protocolo SOAP consome mais energia que protocolos implementados utilizando os paradigmas de REST (ROUACHED; BACCAR; ABID, 2012).

Os trabalhos se assemelham ao deixar para aplicações clientes a implementação dos requisitos de segurança e recursos de QoS. Parte dessa atuação é explicável pelo fato de tais requisitos serem estritamente relacionados com as regras implementadas nas aplicações clientes. O uso de serviços Web oferece suporte aos desenvolvedores no gerenciamento dos recursos de QoS.

A presente proposta, por utilizar uma interface *RESTful*, não faz uso da linguagem WSDL para descrever os seus serviços. Para tanto, utiliza-se do recurso de serviços Web, como: serviço de listagem dos dispositivos (listDevices); serviço de listagem das funcionalidade de um dispositivo específico (listServicesDevice); serviço de execução de uma funcionalidade de um dispositivo (executeServiceDevice). A utilização de um repositório Web com a estrutura dos dispositivos propõe uma solução escalável com suporte a dispositivos construídos posteriormente que tenham a estrutura de seus perfis cadastrados no Repositório. Na atual proposta, não foram considerados os requisitos de segurança e recursos de QoS.

A maioria dos trabalhos pesquisados tiveram seus serviços baseados em SOAP dependendo do HTTP como protocolo de transporte. Contudo, a vinculação SOA HTTP usa apenas um pequeno subconjunto da especificação HTTP, restrito ao método POST e alguns cabeçalhos. Assim sendo, os serviços baseados em SOA HTTP não aproveitam as vantagens oferecidas por muitos dos recursos que fizeram do HTTP um protocolo de transporte comum e escalonável. Embora os Web Services sejam independentes do ponto de vista de transporte, muitas das implementações do mundo real aproveitam o HTTP como protocolo subjacente (RODRIGUEZ; DEMSAK, 2012). Esses serviços hospedados em HTTP trabalham de forma semelhante aos sistemas baseados na Web. Contudo, as diferentes camadas de abstrações que são construídas sobre o protocolo HTTP limitam esses serviços e impedem que façam uso total das capacidades da Web. Este trabalho aborda esse desafio, adotando estilos de arquiteturas como o REST nas tecnologias SOA. Seguindo os princípios do REST**,** foram arquitetados serviços escalonáveis, aproveitando os princípios da Web.

3.3 Requisitos de Classificação

A abordagem SOA cobre uma ampla variedade de domínios de aplicação e, geralmente, cada um é projetado para servir a um propósito específico. Conseqüentemente, cada solução SOA tem a sua própria construção e conjunto de recursos oferecidos. No entanto, alguns requisitos têm se apresentado comuns. Através de estudos anteriores [(AL-JAROODI; MOHAMED, 2011), (AL-JAROODI; MOHAMED, 2012), (AKYILDIZ, 2012)], selecionaram-se alguns requisitos para caracterização dos trabalhos a serem abordados, tais como:

a) Serviço de composição e padronização. Suporte para que os provedores de serviços possam construir serviços seguindo um modelo padronizado. O sistema deve oferecer APIs de desenvolvimento utilizáveis que irão facilitar o processo de desenvolvimento e assegurar o cumprimento de normas. Como resultado, os desenvolvedores não precisarão estar envolvidos com problemas de normalização e conformidade, mas apenas focar nas características dos serviços. Além disso, um conjunto maior de aplicações uniformes será criado, mesmo que por grupos diferentes.

b) Serviço de registro e publicação. O sistema deve fornecer aos desenvolvedores ferramentas que os ajude na divulgação de seus serviços, tornando-os disponíveis para reutilização. Além disso, deve oferecer algumas ferramentas de apoio na disponibilidade e continuidade dos serviços para evitar problemas com aplicações que consomem os serviços.

c) Serviços de descoberta e integração. Suporte para as aplicações consumidoras de serviços na descoberta e utilização de forma eficaz dos serviços publicados. É necessário fornecer ferramentas de descoberta de serviços e um modelo que facilite a integração e utilização dos serviços descobertos. Além de ser capaz de lidar com diferentes necessidades dos consumidores de serviços e capacidade de integração.

d) Heterogeneidade. Abstração dos ambientes. Essa é uma questão importante, pois as aplicações e serviços são desenvolvidos independentes um do outro, em diferentes ambientes de desenvolvimento e sistemas operacionais. Além disso, alguns serviços podem ser utilizados em equipamentos e infraestrutura de hardware. Portando, simplesmente cumprir com um determinado padrão não é suficiente para garantir a integração harmoniosa e a interoperabilidade nos serviços, aplicações e plataformas. O sistema deve fornecer APIs

e protocolos que permitam a integração, sem considerar os diferentes ambientes e ferramentas de desenvolvimento.

e) Integração transparente para aplicações clientes. Idealmente, as aplicações clientes não deviam se preocupar com o funcionamento interno dos serviços. As interfaces de comunicação devem ser suficientes para realizar as comunicações. Os serviços devem ser integrados com as aplicações clientes sem interferir em suas funcionalidades operacionais.

f) Escalabilidade e Eficiência. Capacidade de lidar com altas taxas de acesso, grandes volumes de dados e cargas elevadas de comunicação. Muitos serviços são amplamente utilizados e têm alta demanda de acesso sobre eles. Além disso, muitos serviços devem gerar grandes volumes de dados, impondo assim cargas elevadas no sistema e na infraestrutura de comunicação. O sistema deve fornecer mecanismos bem desenhados para apoiar o desenvolvimento de um serviço eficiente e funções de gerenciamento de dados, além de oferecer suporte ao surgimento de novos sensores na rede.

g) Segurança e confiabilidade. O sistema deve fornecer as aplicações clientes, serviços seguros e confiáveis, tanto na comunicação, na qual a infraestrutura e a rede são os alvos principais, quanto nas aplicações, que precisam encontrar formas seguras e confiáveis para integração e utilização dos serviços.

h) Suporte a requisitos de QoS. Tópico importante para aplicações distribuídas. O sistema deve oferecer serviços com suporte aos desenvolvedores na implementação de aplicações com QoS.

i) Suporte para integração com outros sistemas: o sistema proporciona a integração dos dispositivos físicos com outros sistemas, tais como sistemas web e empresariais.

A seguir, a Tabela 3 apresenta a sumarização de cada trabalho aos requisitos analisados, com as seguintes identificações numéricas: (1) Serviço de composição e padronização; (2) Serviço de registro e publicação; (3) Serviços de descoberta e integração; (4) Heterogeneidade; (5) Integração transparente para aplicações clientes; (6) Escalabilidade; (7) Eficiência; (8) Segurança e confiabilidade; (9) Suporte a requisitos de QoS; (10) Integração com outros sistemas.

Trabalhos	Principais Características	Tecnologias Utilizadas.	Requisitos suportados
Othman et al. (2007)	Permite o acesso direto aos dispositivos, não fazendo uso de middlewares.	SOAP, WSDL	1; 3; 4; 5; 6; 10;
Priyantha et al.(2008)	Cada sensor contém dois WSDL descrevendo as suas funcionalidades. O acesso aos sensores é feito por um componente de controle	SOAP, WSDL, XML, WS-*.	1; 3; 4; 5; 6; 7; 10
Glombitza et al. (2010)	Integração entre os protocolos dos sistemas finais e os protocolos utilizados nos sensores. Uma solução específica para um determinado problema.	SOAP, WSDL.	1; 2; 3; 5; 6; 8;
Delicato (2010)	Implementação de algoritmos de roteamento.	SOAP, WSDL, XML, UDDI.	1; 2; 3; 5; 6; 9; 10.
Delicato et al.(2006)	Integração de RSSF através de serviços Web e Mashups.	WSDL, HTML	1; 2; 4; 5; 10.
Avilés e García (2009)	Suporte a múltiplas linguagens de programação	WSDL, SOAP,	1; 2; 4; 5; 10.
Abangar et al (2010)	O mapeamento do dispositivo com IP é feito através do protocolo 6loWPAN	DPWS, WS-*	1; 2; 4; 5; 6;10.
BeeHome (Proposta desta tese)	Utiliza um *gateway* para ter acesso a dispositivos residenciais.	*Web Services*, REST.	1;2;3;4;5;6;7;10

Tabela 3 – Sumarização dos trabalhos relacionados.

3.5 Resumo

Nesse capítulo, foram abordados alguns trabalhos que fazem integração através de *middleware/gateway* orientado a serviços e a utilização de serviços Web. Apresentou-se o

estudo de Othman et al, que propuseram uma solução utilizando o paradigma SOA, objetivando tornar transparente e flexível a comunicação entre os consumidores de serviços e os dados disponibilizados por dispositivos físicos; Priyantha et al. investigaram o uso de serviços Web em sensores a fim de organizar de forma estruturada suas funcionalidades e seus dados, tornando-os acessíveis. Além disso, os autores avaliaram o *overhead* e o consumo de recursos dos sensores ocasionados pela adoção WS-*; Glombitza *et al* argumentaram que os *middlewares* para os sensores existentes não são adequados para serem utilizados pelas indústrias e propuseram a utilização de um método auto-descritivo e automático, baseado em tecnologias de serviços Web para descoberta de serviços dos dispositivos físicos.

Avilés e Garcia adotaram uma arquitetura orientada a serviços que oferecesse uma abstração de alto nível para o desenvolvimento de aplicações de controle; Abangar *et al.* propõem um mecanismo para mediar a troca de dados entre plataformas de sensores heterogêneas e aplicações web e serviços através de um *gateway* que liga diferentes dispositivos com aplicações web; O trabalho de Delicato versa sobre um *middleware* genérico para integração de dispositivos sem fio adotando tecnologias de serviços Web SOAP e SOA. O *middleware* proposto ofereceu abstrações para serviços internos da rede e para serviços da RSSF no *middleware*.

Diante disso, o trabalho proposto (BeeHome) se assemelha aos trabalhos selecionados na integração de dispositivos físicos à Internet. A proposta BeeHome se diferencia ao propor um *gateway* como ponte entre os dispositivos físicos e as aplicações de controle, não precisando aos dispositivos ter parte do *gateway* em sua estrutura e nem carregar arquivos XML com informações de sua estrutura, pois o *gateway* irá fazer uso de um repositório para consultar essas informações. Outra diferença está nos meios adotados para que os dispositivos fossem utilizados em diferentes aplicações, enquanto a maioria utilizou SOAP, BeeHome propôs a utilização do *RESTful* para disponibilizar as funcionalidades dos dispositivos aos clientes finais na forma de serviços Web.

No próximo capítulo, será descrito sistema *gateway* orientado a serviços, seu cenário de atuação, seus requisitos funcionais, componentes da estrutura, tecnologia de comunicação e as funcionalidades apresentadas para integração de dispositivos residenciais com a Internet.

Capítulo 4 – Arquitetura do *gateway* residencial

Este capítulo tem por objetivo descrever o sistema *gateway* orientado a serviços desenvolvidos neste trabalho. O principal objetivo desse sistema consiste na integração de dispositivos residenciais com dispositivos de controle, como *smartphones*, *tablets*, *notebooks*, *etc.* Para tal, propõe-se o uso do padrão de comunicação sem fio, ZigBee, e de serviços Web como interfaces *RESTful*, como tecnologias intermediárias na comunicação do *gateway* com os dispositivos residenciais e os dispositivos de controle. Tópicos que tiveram suas funcionalidades abordadas anteriormente no capítulo 2, fundamentos teóricos.

Este capítulo apresenta inicialmente o cenário residencial considerado para elaboração do sistema *gateway*, com a presença de dispositivos residenciais que não se comunicam entre si, nem com outros dispositivos móveis. Fato esse incrementado pela diferença de protocolos entre eles. Em seguida, é apresentado um cenário modificado com a inclusão de um sistema *gateway* residencial, que funcionará como ponto para conexão com os dispositivos residenciais. Além do *gateway*, dispositivos de comunicação sem fio são adicionados aos dispositivos físicos para a troca de dados com o *gateway*.

Após a ilustração dos cenários, serão apresentados os componentes da estrutura do *gateway*, suas funcionalidades, características e o papel exercido dentro do cenário proposto. Em seguida, serão descritas as funcionalidades do sistema *gateway*, descrevendo serviços como o reconhecimento de novos eletrodomésticos, os serviços Web de listagem de eletrodomésticos, listagem das funcionalidades de um eletrodoméstico específico e o serviço Web de solicitação de execução de uma funcionalidade.

4.1 Cenário

A Figura 6 ilustra o cenário residencial considerado para elaboração desta tese. Este cenário é formado pelos dispositivos residenciais e pelos dispositivos de controle. Os dispositivos residenciais fazem parte de sistemas de segurança, entretenimento, controle de iluminação, controle de temperatura, eletrodomésticos, controladores e centrais de automação. Cada um desses dispositivos exige diferentes requisitos, como: instalação, velocidade, consequentemente, refletem em custos distintos. Os dispositivos de controle são os *smartphones, tablets, Ipod, Ipad, notebooks, dentre outros.* Dispositivos que são usados para enviar e receber dados, também considerados como opção de acesso à internet. Com o advento das tecnologias sem fios, os aparelhos móveis podem se comunicar com outros dispositivos a seu redor.

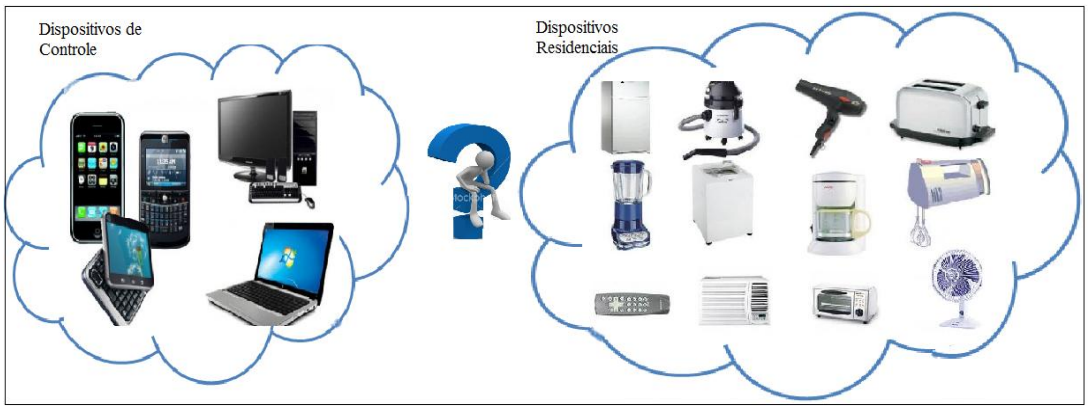

Figura 6 - Cenário residencial sem integralização dos dispositivos

Para interagir os dispositivos residenciais com outros dispositivos remotamente, é necessário colocá-los sob uma simples interface de controle padronizado que interconecte todos numa rede, especificamente em uma rede de área doméstica (HAN – *Home Area Network*). Os tipos de dados que circulam dentro de um sistema de automação residencial são de sensores e atuadores com pequenos pacotes que controlam dispositivos ou que obtêm seus *status*. Um grande desafio é que os dispositivos residenciais não possuem uma interface/comunicação com outros dispositivos.

A Figura 7 ilustra a proposta desta tese, na qual um *gateway* é utilizado como ponte para integração entre os dispositivos residenciais e os dispositivos de controle. Para realizar a comunicação dos dispositivos residenciais com o *gateway*, após estudo realizado na seção

2.3, optou-se por adicionar aos dispositivos físicos um módulo ZigBee, para o envio e recebimento de dados ao *gateway*, formando juntamente com outros eletrodomésticos uma rede do tipo estrela, em que o *gateway* exerce o papel de coordenador. Cada dispositivo residencial terá suas funcionalidades mapeadas em perfil que serão cadastrados em um repositório na Web.

O *gateway* disponibilizará as funcionalidades dos dispositivos residenciais aos dispositivos de controle através de serviços Web, desenvolvidos em uma interface *RESTful*, conforme estudo apresentado na seção 2.2.

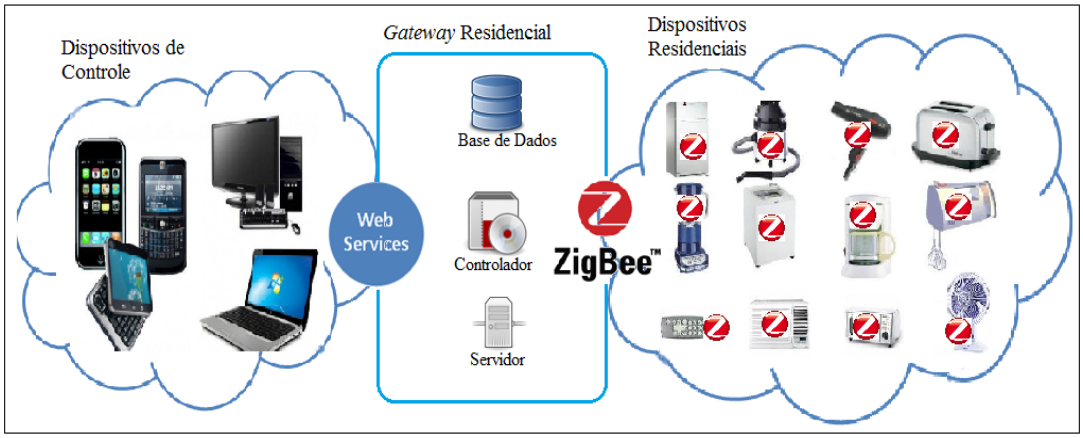

Figura 7 - Sistema *Gateway* proposto.

O *gateway* residencial é o responsável por receber as requisições dos dispositivos de controle, processá-las e encaminhá-las aos dispositivos residenciais específicos. O *gateway* é formado pelos componentes da Base de Dados, responsável pelo armazenamento das informações; o componente Controlador, responsável pelo tráfego de rede de dados com os dispositivos residenciais; o componente Servidor, responsável por disponibilizar os serviços Web.

4.2 Requisitos do sistema *gateway*

O sistema *gateway* proposto deve atender a alguns requisitos para que seja realizada a integração dos eletrodomésticos com outros dispositivos de controle. São eles:

- Abstração. O sistema *gateway* deve disponibilizar ferramentas para permitir o acesso às funcionalidades dos eletrodomésticos sem precisar especificar informações do *hardware/software* dos itens.
- Adaptável. O sistema deve ser adaptável ao adicionamento de novos eletrodomésticos sem que seja necessária uma reprogramação nas aplicações de controle.
- Escalável. O sistema deve estar preparado para o crescimento do número de eletrodomésticos.
- Integração. O sistema deve permitir a integração com dispositivos de outras redes.

Esses requisitos serão levados em consideração durante a fase de experimentos na qualificação dos resultados.

4.3 Componentes do *gateway*

A estrutura do sistema de *gateway* proposta consiste em vários módulos, em que cada módulo implementa um conjunto de funcionalidades do *gateway* e corresponde a um subsistema ou componente. Entre os componentes que compõem o *gateway*, há o Coordenador, Repositório, Base de Dados, Controlador e Servidor. A Figura 8 ilustra esses componentes que serão descritos nas seções a seguir.

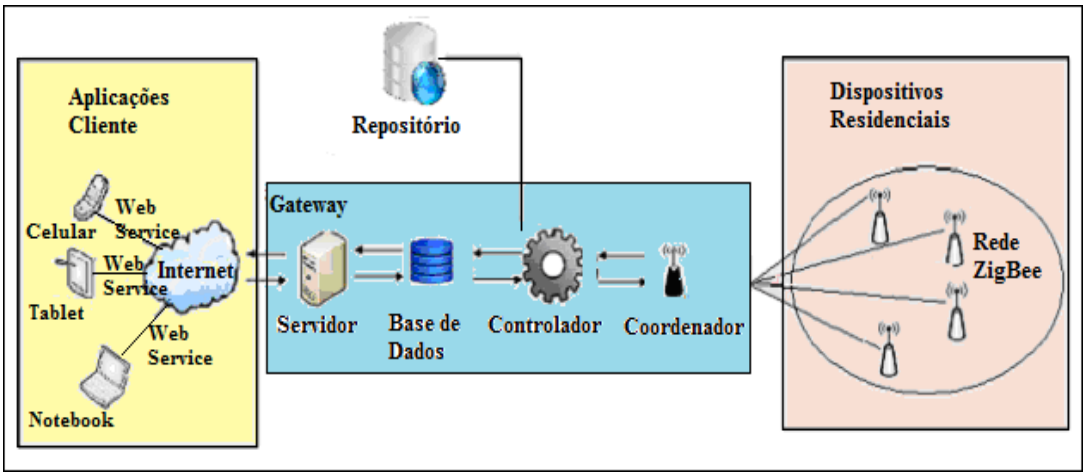

Figura 8- Componentes da Arquitetura

Coordenador é o componente acoplado através da UART (*Universal Asynchronous Receiver-Transmitte*), responsável pela parte física de envio e recebimento de dados pelo *gateway* aos eletrodomésticos. Esse elemento será o coordenador da rede ZigBee.

Repositório é o local de armazenamento das estruturas dos Perfis. Todos os dispositivos deverão ter seus perfis cadastrados no Repositório para consultas pelo *gateway* residencial. Assim, na fabricação de um novo dispositivo, a estrutura de seu Perfil de funcionamento deverá ser cadastrada no Repositório que irá disponibilizá-los ao *gateway* por meio de serviços Web. Estes perfis englobam comandos práticos que envolvem uma rotina de atividades, tais como: ligar/desligar luzes, controlar a temperatura de ambientes, medidores de energia, medidores de gás, controles de bombas de água, detectores de incêndios, dentre outros.

Controlador é o componente responsável pela parte lógica de envio e recebimento de dados pelo *gateway* a rede de sensores. Toda a comunicação com os dispositivos residenciais passa por esse componente. Entre suas funções, há a identificação e registro dos dispositivos residenciais disponíveis na rede, envio de comandos para execução de serviços solicitados por aplicações externas, recebimento do retorno dos eletrodomésticos às solicitações, consulta ao Repositório por informação de Perfis de dispositivos não registrados.

Base de Dados é o componente de armazenamento da rede que guardará as informações referentes aos eletrodomésticos, seus perfis, as solicitações realizadas pelas

aplicações externas e o retorno dos dispositivos internos. Os componentes que acessam a Base de Dados são o Servidor e o Controlador.

Servidor é o componente que tem como funcionalidade principal trabalhar como provedor de serviços Web. O componente deverá disponibilizar três serviços principais para integração com aplicações externas como a listagem de todos os dispositivos disponíveis, a listagem dos serviços suportados por cada dispositivo e um serviço para executar uma funcionalidade específica. Os dados para a geração dos serviços serão obtidos da Base de Dados.

4.4 Perfil dos Dispositivos

Cada dispositivo residencial possui um Perfil que caracteriza o seu funcionamento e os serviços suportados por ele. Exemplo: O perfil de uma lâmpada possui os serviços de ligar e desligar. Outras informações sobre os serviços são encontrados nos perfis, tais como: se o serviço possui parâmetro, o tipo de parâmetro, as informações das mensagens de retorno, etc.

O Repositório Web armazena todos os perfis e os disponibiliza às aplicações através de serviços Web. Em relação ao *gateway*, possui uma base de dados interna na qual armazena as informações dos perfis dos dispositivos residenciais, usados na comunicação interna do *gateway* com os dispositivos e na geração dos serviços Web disponibilizados às aplicações clientes.

No surgimento de um dispositivo cujo perfil não esteja armazenado na base de dados do *gateway*, ele acessa o serviço Web do repositório e baixa as informações, estruturas do novo perfil. Dessa forma, os perfis na base de dados podem ser atualizados, e os dispositivos não precisam ter sua memória interna consumida com arquivo contendo sua estrutura de funcionamento.

4.5 *Gateway* segundo o padrão SOA

Como visto anteriormente, na abordagem proposta, os dispositivos residenciais são vistos como um fornecedor de serviços para as aplicações clientes. O objetivo desta seção é

mostrar como os componentes do sistema *gateway* aderem ao padrão de uma arquitetura orientada a serviços. Demonstrando como as operações e os papéis definidos no padrão SOA são mapeados para a arquitetura de rede residencial utilizada no presente trabalho.

Uma aplicação cliente consultando dados de um eletrodoméstico, desempenha o papel de um solicitante de serviços. Dispositivos residenciais são fornecedores de serviços, fornecendo o serviço básico de entrega de dados e serviços específicos.

Os dispositivos físicos enviam sua identificação de Perfil ao *gateway*, que através de um repositório externo ao *gateway* consulta a estrutura de funcionamento dos eletrodomésticos, executando dessa forma uma operação de publicação. O *gateway* mantém uma base de dados com descrição dos serviços de cada tipo de eletrodoméstico existente na rede ZigBee.

O *gateway* funciona também como agregador de serviços, combinando as funcionalidades de provedor e requisitante, trabalhando como uma camada intermediária entre serviços consumidores (aplicações de controle) e os serviços provedores (dispositivos residenciais), encapsulando o acesso às funcionalidades providas por diferentes eletrodomésticos. Dessa forma, o *gateway* atua como provedor ao disponibilizar funcionalidades dos eletrodomésticos na forma de serviços às aplicações de controle. O *gateway* atua como requisitante na medida em que ele requisita a execução de funcionalidades providas pelos eletrodomésticos.

4.6 Funcionalidades do *gateway* residencial

O serviço básico fornecido do *gateway* consiste na integração dos eletrodomésticos com os dispositivos de controle. Essa integração depende da descoberta das capacidades de cada dispositivo residencial, das solicitações de dados pelas aplicações de controle e da forma como se dá a comunicação entre os dispositivos físicos produtores de dados e as aplicações de controle (consumidora de dados).

A construção do *gateway* proposto é influenciada pelos requisitos das aplicações e por questões relacionadas à infraestrutura de comunicação sobre a qual o *gateway* irá executar. A descrição do sistema de *gateway* proposto inicia com a definição dos serviços que devem ser fornecidos para atender às necessidades das aplicações e para se adequar ao

ambiente residencial. A seguir, será descrito como os dispositivos residenciais interagem com o *gateway* e como suas funcionalidades são disponibilizadas às aplicações de controle.

4.6.1 Registro de dispositivos

Antes que o sistema possa entrar em operação, é necessário que os dispositivos residenciais sejam conhecidos. Para possibilitar a execução desses serviços, uma fase inicial de descoberta de dispositivos internos é necessária.

A descoberta de dispositivos começa com uma fase inicial de registro, durante a qual o *gateway* envia uma mensagem para os dispositivos residenciais solicitando as suas confirmações. A mensagem de retorno dos dispositivos inclui a identificação do eletrodoméstico e o seu perfil. O *gateway* armazena o conteúdo das mensagens de configuração recebidas em uma base de dados local. Esse procedimento de verificação dos dispositivos residenciais é feito periodicamente. A Figura 9 mostra a troca de mensagens entre o *gateway* e os dispositivos físicos.

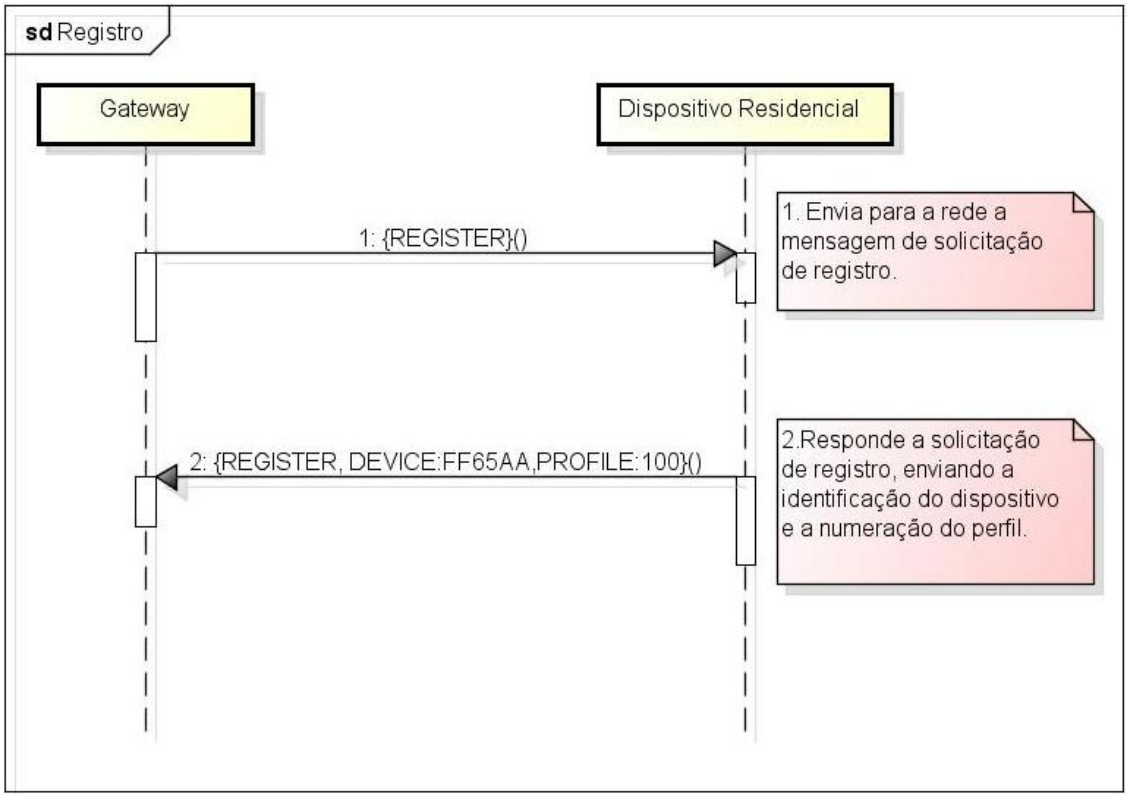

Figura 9 – Fluxo de Mensagens para registro de dispositivo

A Figura 10 ilustra a seqüência de atividades juntamente com os componentes envolvidos, cujo Controlador, Coordenador, Base de Dados e Repositório são da estrutura do *gateway*. O Coordenador recebe a mensagem enviada pelo Dispositivo Residencial e encaminha-a ao componente Controlador que faz uma busca na Base de Dados por tal perfil. O Perfil contém informações do funcionamento do dispositivo, seus serviços e comando de execução. Se o perfil não for encontrado, o Controlador acessará o Repositório, lugar onde estão localizadas as informações de todos os Perfis suportados pela aplicação, para fazer download do novo perfil e registrá-lo na Base de Dados. Com o novo perfil cadastrado, o Controlador incluirá o dispositivo na lista do *gateway*.

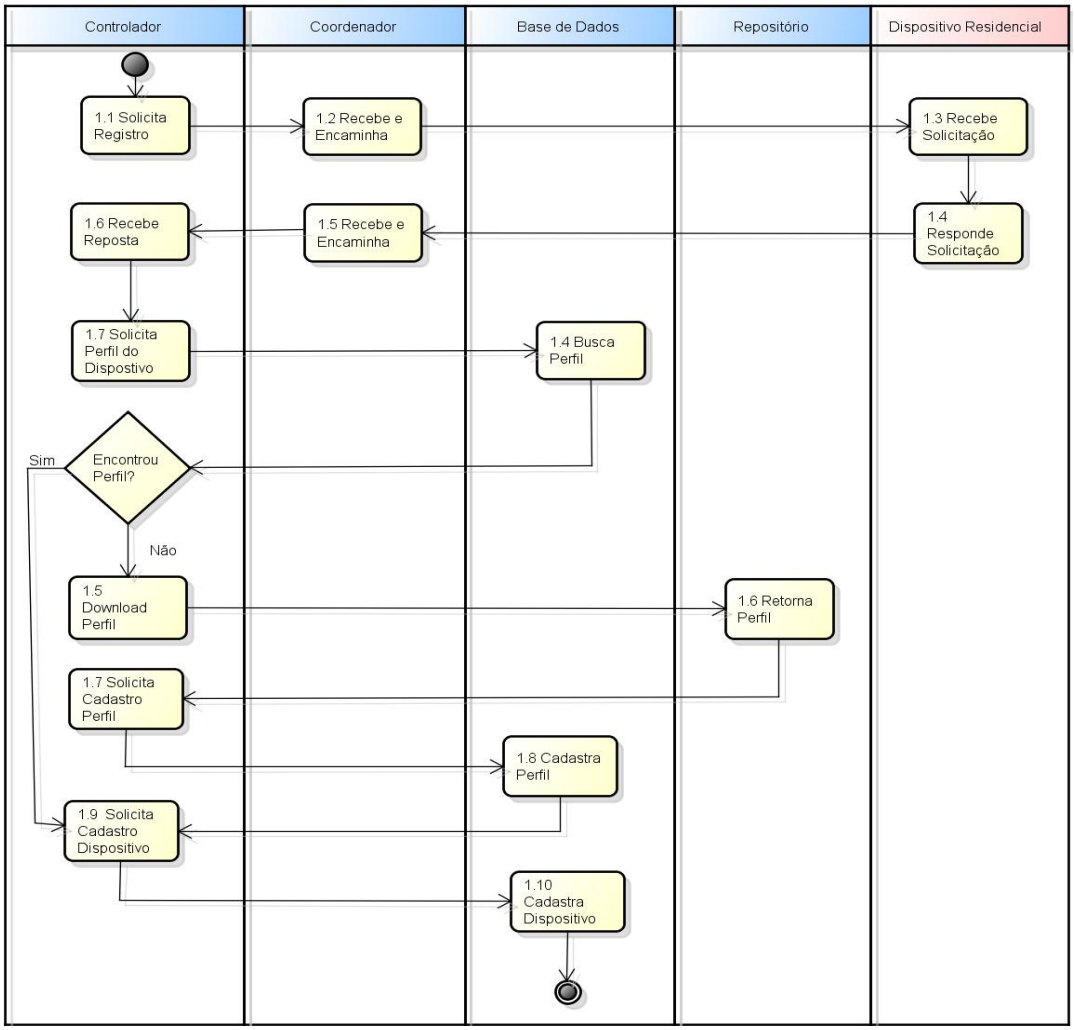

Figura 10 - Atividades Registro de Dispositivos

Esse procedimento de solicitação de registro é feito regulamente em um determinado período de tempo. Caso um dispositivo registrado fique por certo tempo sem

confirmar a mensagem de solicitação de registro, ficará desabilitado até nova confirmação e não aparecerá na lista de dispositivos do *gateway*.

4.6.2 Acesso à lista de dispositivos disponíveis

Na fase inicial, a aplicação passa a conhecer o endereço do *gateway*, os serviços disponíveis nos dispositivos residenciais e o correspondente formato das mensagens para solicitá-los. A primeira operação solicitada pela aplicação tem por objetivo disponibilizar a listagem dos dispositivos residenciais. Essa operação é realizada através da execução ao serviço Web denominado *AllDevices.*

A requisição ao serviço Web é formada pelo endereço do servidor Web e a porta de acesso (*http://127.0.0.1:8080*), o nome da aplicação (*/homegateway*), o ponto de acesso (*/zigbee*) e o serviço Web (*/alldevices*). A mensagem de retorno será enviada no formato *RESTful*, organizada em vetores e com informações de todos os dispositivos, seu perfil e os serviços suportados por eles. A Tabela 4 exemplifica uma requisição e o retorno feito ao serviço Web *AllDevices*.

URL	http://localhost:8080/homegateway/zigbee/alldevices
Retorno	{"device":{"id":"1","netAddr":"XYZ123","profile":{"id":"1","name":"TempSensor","services":{"command":"readTempCelsius","description":"Reads the celsius temperature value receive from the sensor node","id":"1","inputData":"none","inputUnit":"none","method":"readTempCelsius","name":"readStatus","outputData":"number","outputUnit":"C","param":"0"}}}, {"device":{"id":"2","netAddr":"ABC123","profile":{"id":"2","name":"MagnetSensor","services":{"command":"readStatus","description":"Reads the door/window status received from the sensor node (1: open; 0: closed)","id":"2","inputData":"none","inputUnit":"none","method":"readStatus","name":"readStatus","outputData":"number","outputUnit":"none","param":"0"}}}, {"device":{"id":"3","netAddr":"DEF456","profile":{"id":"3","name":"ElectricFan","services":[{"command":"turnOn","description":"Turns on the eletricfan","id":"3","inputData":"none","inputUnit":"none","method":"turnOn","name":"turnOn","outputData":"none","outputUnit":"none","param":"0"}}}

Tabela 4 – Solicitação e retorno do serviço Web *AllDevices*

Ao receber a solicitação, o *gateway*, através do componente Servidor, consulta a Base de Dados por todos os dispositivos residenciais disponíveis, montando uma listagem com informações dos mesmos, seus perfis e serviços suportados; retornando com esses dados à aplicação cliente, conforme ilustrado na Figura 11.

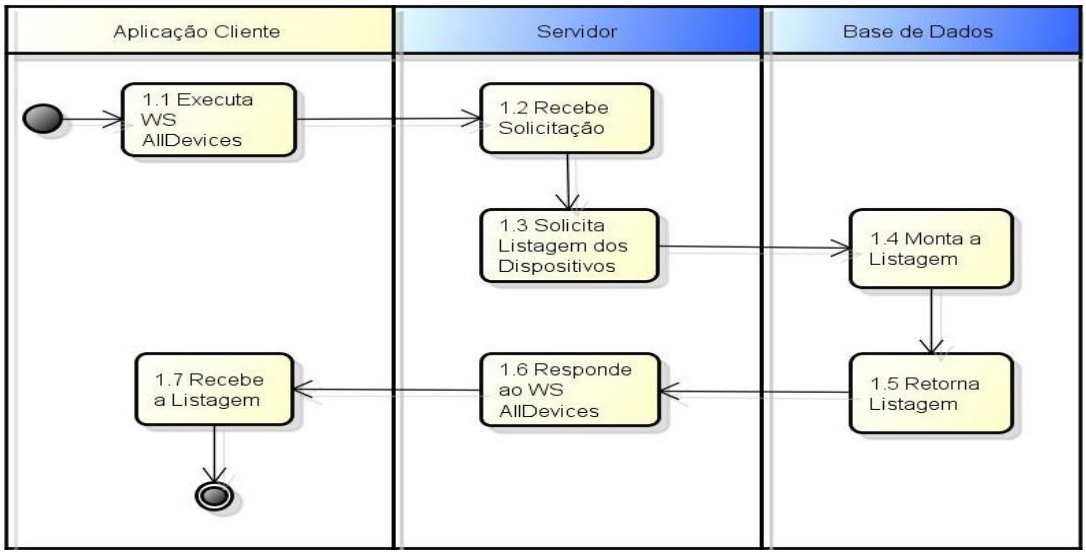

Figura 11 - Atividade Acessando a lista dos dispositivos

4.6.3 Acesso às informações de um dispositivo específico.

Após conhecer os dispositivos físicos existentes, a aplicação cliente pode solicitar o serviço Web denominado *Device* para conhecer as características de um dispositivo residencial específico. Na solicitação do dispositivo de interesse, as aplicações clientes devem informar valores mínimos de parâmetros a serem respeitados, como a identificação do dispositivo residencial.

A Figura 12 ilustra a seqüência de atividades para acessar as informações de um dispositivo entre as aplicações cliente e o *gateway* (Servidor e Base de Dados), que ao receber a solicitação, consulta a Base de Dados e monta um pacote com todas as informações do dispositivo requerido e retorna a aplicação cliente, conforme demonstrado na Tabela 5.

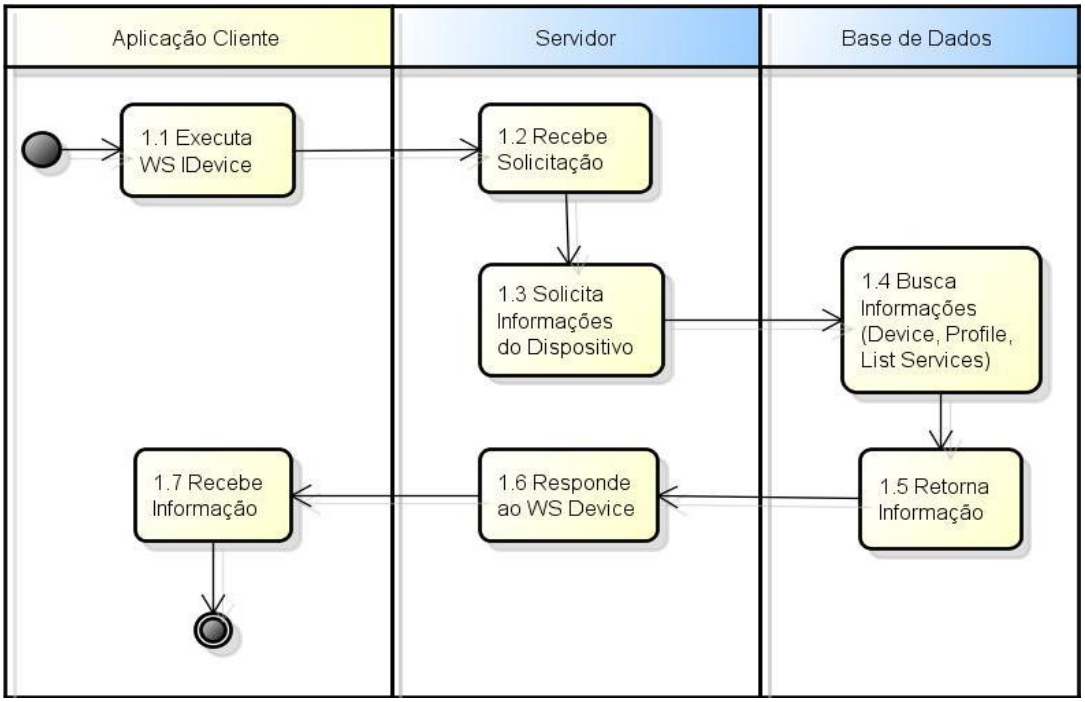

Figura 12 - Atividade Acessando Informações do dispositivo

URL	http://localhost:8080/homegateway/zigbee/device/ABC123
Retorno	{"device":{"id":"3","netAddr":"DEF456","profile":{"id":"3","name":"ElectricFan","services":[{"command":"turnOn","description":"Turns on the eletricfan","id":"3","inputData":"none","inputUnit":"none","method":"turnOn","name":"turnOn","outputData":"none","outputUnit":"none","param":"0"}]}

Tabela 5 – Solicitação e retorno do serviço Web *Device.*

4.6.4 Solicitação de um serviço.

O contexto de execução de serviço é representado pelo serviço Web denominado *ExecuteService*, que compreende um conjunto de parâmetros que são definidos em termos dos valores de dados dos sensores.

A execução do serviço Web *ExecuteService* contém como parâmetros o identificador do dispositivo residencial, o nome serviço solicitado e os parâmetros de controle do serviço. Por exemplo, um dispositivo de perfil Lâmpada, contém os serviços de ligar, desligar, luminosidade. Os serviços de ligar e desligar não necessitam de parâmetros, por serem comandos diretos, no entanto o serviço luminosidade necessita de um parâmetro

numérico, fazendo referência à intensidade da luz que se deseja controlar. A Figura 13 ilustra a seqüência de atividades necessárias à execução de um serviço.

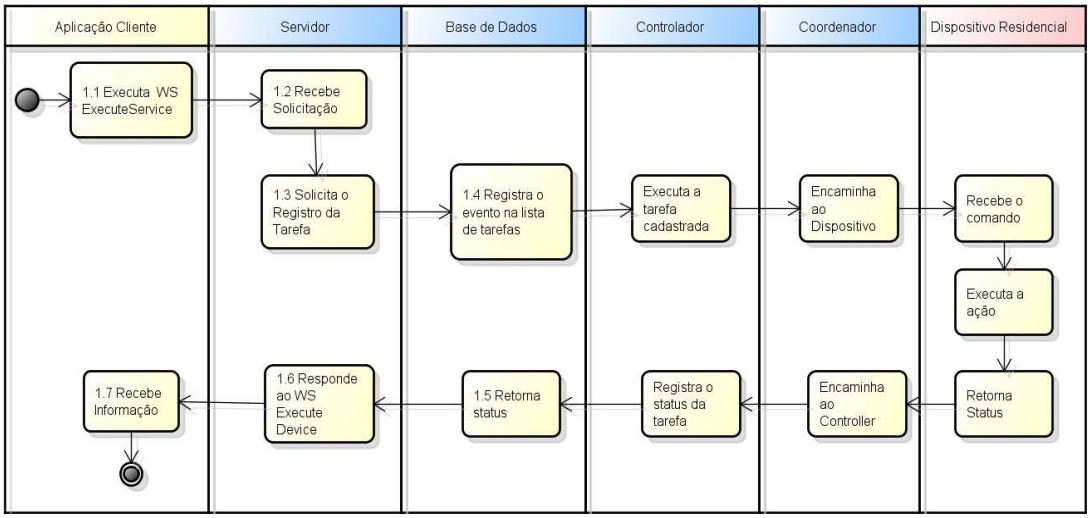

Figura 13 - Execução do *WS ExecuteService*

Ao receber a requisição do serviço Web, o *gateway*, através do componente Servidor, registra na Base de Dados uma tarefa com situação pendente de execução e aguarda por resposta do dispositivo. Posteriormente, o Controlador verificará a listagem das tarefas pendentes de execução e enviará uma solicitação de execução das tarefas aos dispositivos residenciais de destino, com informações do dispositivo destino, o serviço, os parâmetros e o registro da tarefa na Base de Dados, que será usado como localizador da tarefa na Base de Dados em posterior atualização do registro.

O dispositivo destino recebe a mensagem, executa a tarefa e retorna ao Coordenador outra mensagem informando o resultado da execução da solicitação, juntamente com identificador da tarefa na Base de Dados. Com a chegada da mensagem, o Controlador interpreta e registra a mensagem de retorno na Base de Dados. Com o retorno do dispositivo, o registro da tarefa na Base de Dados passará da situação de pendente para executado. Nesse momento, o componente Servidor, acessa a Base de Dados e envia o retorno do dispositivo residencial à aplicação cliente solicitante.

Se, após certo tempo, o dispositivo não enviar a mensagem de execução da tarefa, o *gateway* cancela a tarefa e envia à Aplicação Cliente uma mensagem informando que não houve resposta. A mensagem enviada aos dispositivos residenciais é formada por um

identificador da mensagem ("*TASK*"), o identificador do dispositivo destino, o serviço, os parâmetros do serviço e a identificação da tarefa registrada na Base de Dados.

A Tabela 6 demonstra uma solicitação de serviço de controle de temperatura, enviada a um Dispositivo Residencial de identificação XYZ123, solicitando a execução de um serviço de controle de temperatura chamado *setTemperatura* e tendo como parâmetro uma temperatura de 18º C. Após a execução, o *gateway* retorna à Aplicação Cliente uma mensagem de confirmação da execução ou uma mensagem de erro.

URL	http://localhost:8080/homegateway/zigbee/execute/XYZ123/setTemperatura/18
Retorno	{“task”,{“id”:”100”,“device”:”XYZ123”, ”service”:”SETTEMPERATURE”,“result”:”OK”}}

Tabela 6 – Solicitação e retorno do serviço Web *ExecuteService*

A Figura 14 mostra o fluxo de atividades do *gateway* com o dispositivo, em que o Controlador envia uma mensagem de controle a uma lâmpada, solicitando o desligamento desta. O identificador da lâmpada é XYZ123, o serviço correspondente é o *TURNOFF* e a tarefa é a de número 100.

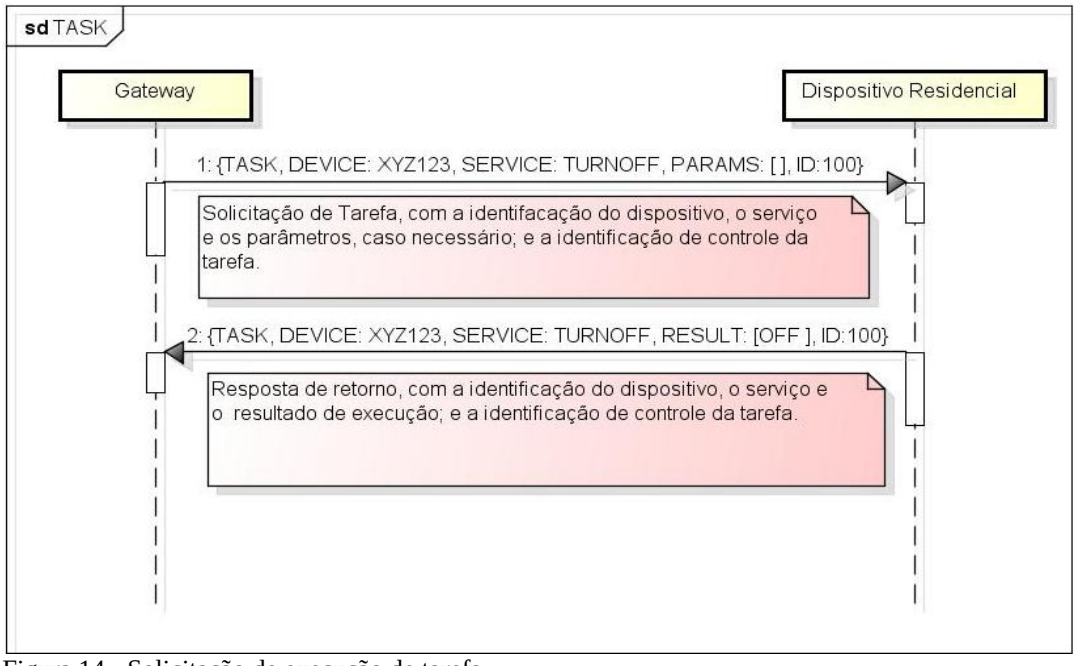

Figura 14 - Solicitação de execução de tarefa

4.7 Resumo

Neste capítulo, foi apresentado o sistema *gateway* proposto para integração entre os dispositivos residenciais e dispositivos móveis (*smartphones, tablets, notebooks, etc)*, que poderão interagir com os eletrodomésticos através de aplicações de controle e monitoramento.

Inicialmente, foi apresentado o cenário sem o *gateway*, cenário esse no qual os dispositivos residenciais não interagem entre si, nem com outros dispositivos. Em seguida, foi apresentado o cenário com a implementação da proposta deste trabalho. Nesse novo cenário, os dispositivos residenciais têm acoplado a sua estrutura um módulo ZigBee, para a troca de dados com o *gateway* residencial. Ainda em relação à comunicação entre os eletrodomésticos e o *gateway*, cada eletrodoméstico possui um perfil que informa a sua estrutura funcional ao *gateway*. O *gateway* funciona como uma ponte entre os dispositivos residenciais e os de controle. Este deverá disponibilizar as funcionalidades dos eletrodomésticos às aplicações de controle através de uma interface baseada em serviços Web.

A seção seguinte apresenta os requisitos a serem considerados durante o desenvolvimento do sistema proposto. Esses requisitos serão como índices qualitativos na avaliação dos experimentos. Em seguida, os componentes do *gateway* (Coordenador, Base de Dados, Controlador, Repositório, Servidor) são apresentados, suas funcionalidades, características e o papel exercido dentro da estrutura do *gateway*. O componente Coordenador exerce a função de coordenador da rede ZigBee, responsável por enviar e receber os dados dos eletrodomésticos. O componente Base de Dados é o responsável pelo armazenamento das informações referentes às solicitações das aplicações de controle, às respostas dos eletrodomésticos, à estrutura funcional dos perfis dos dispositivos residenciais, incluindo a listagem dos serviços suportados.

O Componente Controlador é o responsável pelo processamento interno das mensagens entre as aplicações de controle e os eletrodomésticos. Entre as suas funcionalidades, pode-se citar o gerenciamento das requisições das aplicações cliente; as mensagens de retorno dos dispositivos residenciais; o reconhecimento periódico dos eletrodomésticos.

O componente Servidor é o responsável por manter os serviços Web às aplicações clientes, recebendo as requisições e armazenando-as na Base de Dados para posterior processamento pelo componente Controlador. O componente Repositório mantém as informações das estruturas de todos os perfis. Caso um novo eletrodoméstico seja adicionado com um perfil desconhecido pelo *gateway,* este deverá acessar o Repositório para atualizar a sua base de perfil. Esse componente não faz parte da estrutura interna do *gateway*, ele fica externo, na Web, e é acessado através de serviços Web.

Após a descrição dos componentes, a próxima seção é dedicada às funcionalidades do sistema *gateway*, descrevendo a forma como o sistema *gateway* irá interagir com os eletrodomésticos e as aplicações de controle. A seção descreve as funcionalidades de reconhecimento de eletrodoméstico, cujo sistema deve registrar na sua listagem de dispositivos novos eletrodomésticos que venham a fazer parte da rede ZigBee. Em relação à interação com as aplicações de controle, foram descritas as funcionalidades de listagem de dispositivos, listagem de serviços e execução de um serviço específico.

O próximo capítulo abordará a estrutura (*hardware/software)* utilizada para a implementação do sistema *gateway*, o ambiente de desenvolvimento, as ferramentas utilizadas na construção dos protótipos.

Capítulo 5 – Implementação do *gateway* residencial

Este capítulo tem o propósito de descrever a implementação de um *gateway* residencial e de seus componentes. Devido à diversidade de dispositivos domésticos, optou-se por fazer uso de protótipos para simulação das funcionalidades de um aparelho real. Na construção dos protótipos, adotou-se o paradigma de orientação a objetos e um processo de desenvolvimento interativo e incremental, em que a cada etapa novas funcionalidades foram adicionadas ao sistema em construção. Como as características de *hardware* diferem significativamente entre os dispositivos domésticos e o *gateway*, diferentes plataformas de desenvolvimento foram utilizadas para implementar cada tipo de simulador dos aparelhos domésticos.

Além dos simuladores, foi montada uma lâmpada acoplada a um módulo ZigBee através de placas Arduino. A tecnologia ZigBee foi escolhida para comunicação entre o *gateway* e os dispositivos residenciais, por possuir características de baixo consumo de energia e baixo custo, menor taxa de transmissão e rede com maior número de elementos. O comparativo do padrão ZigBee com outras tecnologias sem fio foi abordado no capítulo 2.

A seguir serão apresentados os ambientes (hardware e software) dos simuladores utilizados na concepção dos dispositivos domésticos. A implementação do sistema *gateway*, seus componentes e as configurações necessárias para a utilização desses componentes.

5.1 Protótipos dos dispositivos residenciais

Para simular o comportamento de dispositivos domésticos, tais como ventiladores elétricos, condicionadores de ar, persianas, entre outros, foram desenvolvidos alguns simuladores através da plataforma de desenvolvimento Delphi. A escolha dessa plataforma justifica-se em razão da variedade de componentes disponibilizados em seu ambiente de desenvolvimento que facilitaram a construção dos simuladores.

Durante o processo de construção, outros componentes externos à plataforma Delphi foram utilizados em implementações específicas, tais como o *Async Professional* (componentes que fornecem serviços de comunicação serial) e o *InstrumentLab* (componentes visuais de instrumentos de Laboratório).

O simulador foi executado em um computador com sistema operacional Windows 7 (64bits), com 4Gb de memória interna. Acoplado à porta USB estava uma placa CON-USBBEE, ilustrado na Figura 13, proporcionando uma conexão estilo *Pen drive*, que após a instalação do driver USB e conexão da placa ao computador, o Windows cria automaticamente uma porta COMx virtual, possibilitando, assim, desenvolver um programa que acesse a COMx como se fosse uma porta serial padrão RS232. O ZigBee utilizado é o modelo XBee-Pro, com antena de fio.

A Figura 14 ilustra um simulador de ventilador elétrico com as funcionalidades de ligar, desligar e ajuste no nível de velocidade e a Figura 15 ilustra o simulador de termômetro. A tela em branco presente nas aplicações de simulação foi utilizada para monitoramento das mensagens enviadas pelo gateway.

Figura 15 - Placa CON-USBBEE com módulo Xbee Pro.

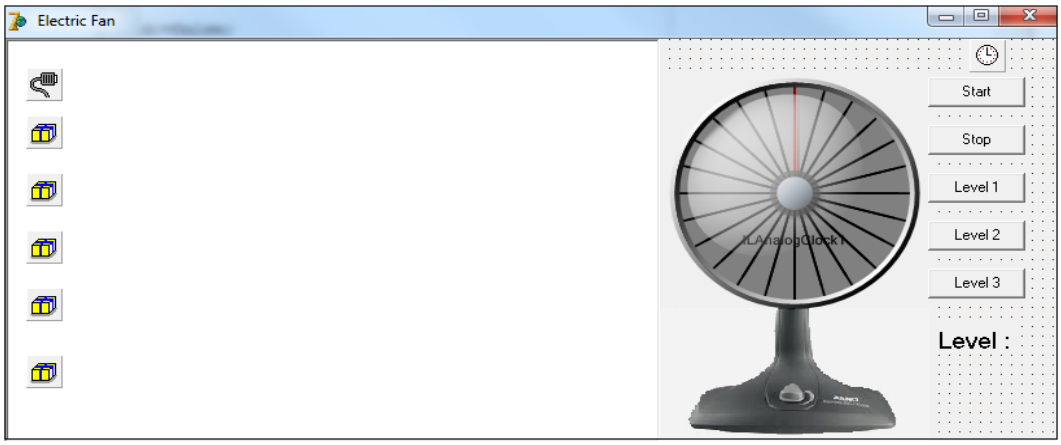

Figura 16 - Simulador Ventilador Elétrico

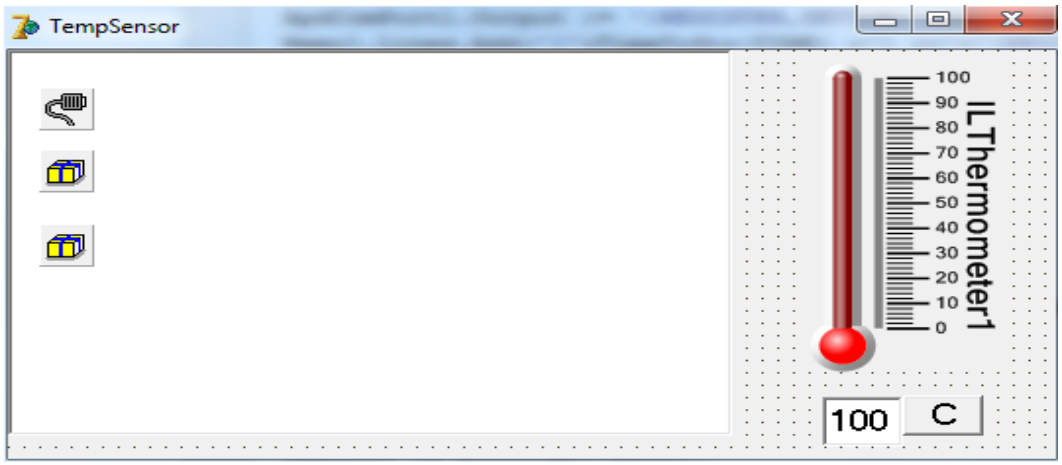

Figura 17 - Simulador Sensor de Temperatura

5.2 Protótipo do *gateway*

A construção do gateway foi dividida em cinco partes, referentes aos componentes Coordenador, Base de Dados, Controlador, Servidor e Repositório. Todos implementados em um único computador com sistema operacional Windows 7, processador Intel Xeon 2.67 GHz, 8Gb de memória.

O componente Coordenador constitui-se de um dispositivo ZigBee, modelo *XBee-Pro*, acoplado a uma placa CON-USBBEE, com o *firmware* de *ZigBee Coordenador* AT instalado em suas configurações. Conforme ilustração da Figura 18.

Figura 18 - Zigbee Coordenador acoplado ao computador.

A Base de Dados de dados foi desenvolvida utilizando o Banco de Dados *Postgres* 9, a modelagem do sistema foi construída seguindo o conceito de que cada dispositivo residencial tenha um Perfil associado que descreve as suas funcionalidades e o conjunto de serviços fornecidos por esses aparelhos.

A Figura 19 apresenta as classes que compõem o sistema e os seus relacionamentos. O Diagrama é composto de 4 classes: a Classe *Profile* está responsável por armazenar as informações dos Perfis dos dispositivos (identificação, nome e descrição); a Classe *Service* que, associada à classe *Profile*, armazena as informações dos serviços disponibilizados pelos Perfis (identificação, nome, descrição, parâmetros de entrada, parâmetros de saída, comando de execução, entre outras); a Classe *Device* contém informações dos dispositivos da rede de sensores (identificador, o Perfil associado, o endereço de rede e a localização); e a Classe *Task* que registra as solicitações efetuadas pelos dispositivos clientes, identificando o serviço e o dispositivo da rede de sensores, registra o início, término e o retorno do dispositivo. Esta classe está associada à classe *Device* e à classe *Service*.

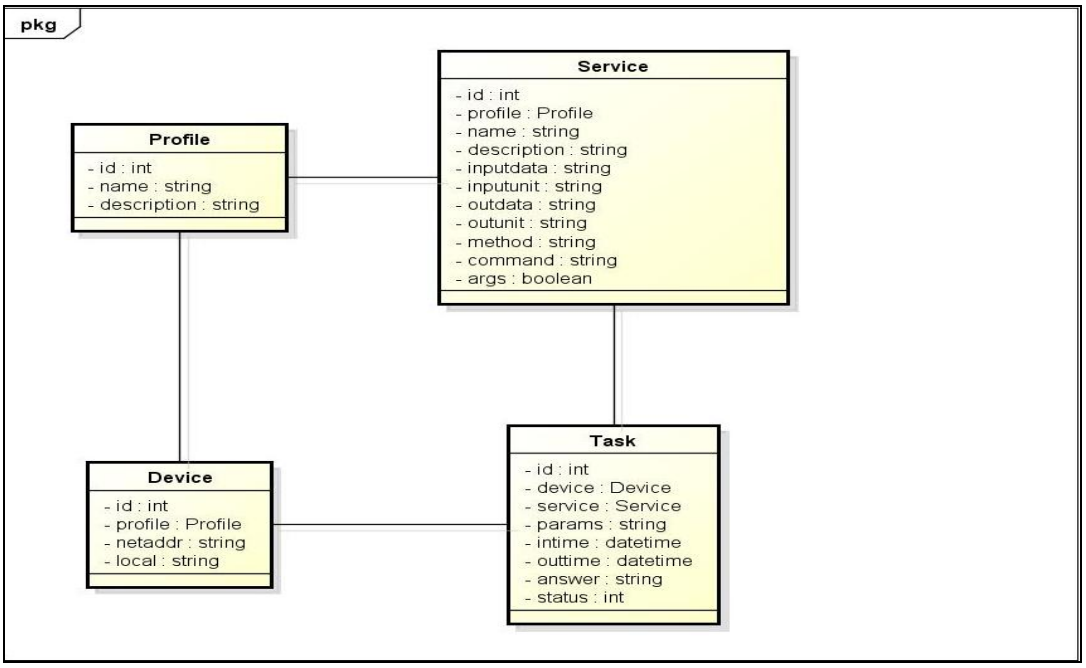

Figura 19– Diagrama de Classe da Base de Dados

O componente Servidor foi desenvolvido utilizando o servidor de aplicações *TomCat* 7 em uma plataforma de desenvolvimento Java, tendo como editor o eclipse versão Juno. A Figura 20 mostra o ambiente de desenvolvimento do Eclipse e o projeto *ZigBee Gateway* que desempenha o papel do componente Servidor.

Na construção do componente Controlador, utilizou-se a linguagem de programação Delphi 7, que consiste de elementos, ferramentas de design e de banco de dados que auxiliaram no desenvolvimento e testes da aplicação de forma rápida e intuitiva.

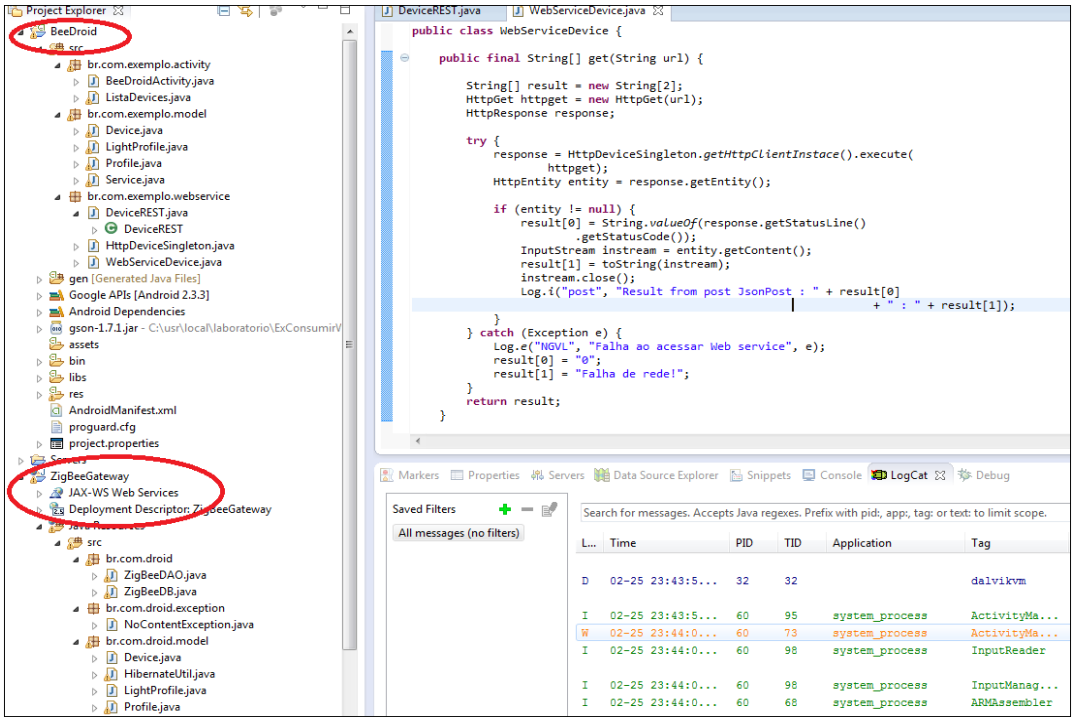

Figura 20 - Ambiente de Programação Java

A Figura 21 mostra o componente Controlador em ambiente de desenvolvimento. Os componentes utilizados foram o *Async Professional* para realização da comunicação serial; componentes ADO na comunicação com Base de Dados *Postgres*; componentes SOAP na comunicação de *Web Services*; *Timer* na sincronização das tarefas.

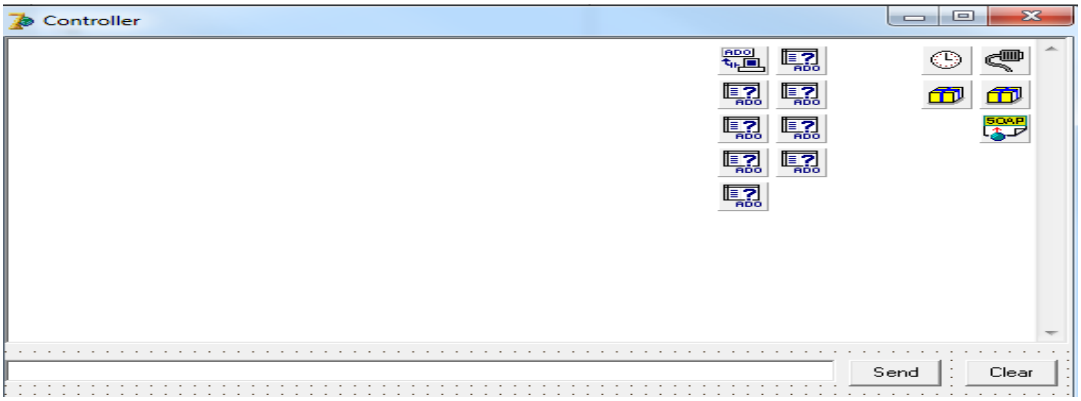

Figura 21 - Aplicação que exerce a função do componente Controle

Para o Repositório, criou-se uma aplicação que disponibiliza serviços Web para retornar a estrutura dos perfis dos dispositivos residenciais ao *gateway*, Figura 22. No desenvolvimento da Web Service, foi utilizada a tecnologia SOAP, que gerou um arquivo WSDL, Figura 23, com a descrição de funcionamento do serviço web. Através desse arquivo, o Controlador pode ter acesso aos serviços dos perfis consultados.

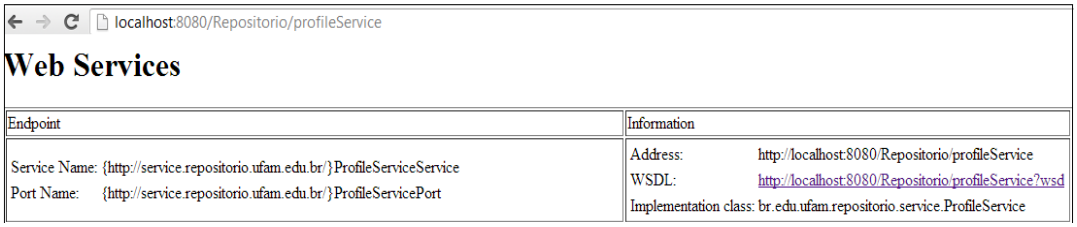

Figura 22 – Tela de informações de acesso do *Web Service*

Figura 23 – Arquivo WSDL gerado para o serviço web de consulta de Perfil

5.3 Aplicação Cliente

Para testar as Aplicações Cliente, desenvolveu-se uma aplicação para celular utilizando um simulador de *smartphones* Android. O aplicativo foi desenvolvido unicamente com o intuito de testar o acesso aos Dispositivos Residenciais através dos *Web*

Services disponibilizados. A tela principal foi desenvolvida com três opções de entrada: a identificação do dispositivo residencial, o serviço a ser executado e os parâmetros de entrada. Logo em seguida, colocaram-se três botões que executam os serviços web de Listagem (*AllDevices*) de todos os dispositivos registrados no *gateway*, o serviço de consulta à informação de um dispositivo específico (*Device*) e o botão de execução de serviço (Execute), conforme ilustrado na Figura 24.

O ambiente de desenvolvimento utilizou-se da máquina virtual *Android SDK Tooks* 2.2; simulador smartphone Nexus; eclipse Juno.

Figura 24 - Aplicação Cliente no simulador *Android*.

5.4 Dispositivo residencial de teste

Além do ambiente de simulação, montou-se uma lâmpada para uso nos testes com dispositivos físicos reais. A Figura 25 ilustra a lâmpada com o módulo ZigBee acoplado à plataforma Arduino. Durante os testes, essa lâmpada terá as funcionalidades de ligar e desligar, funcionalidades essas que o gateway irá disponibilizar através de serviços Web.

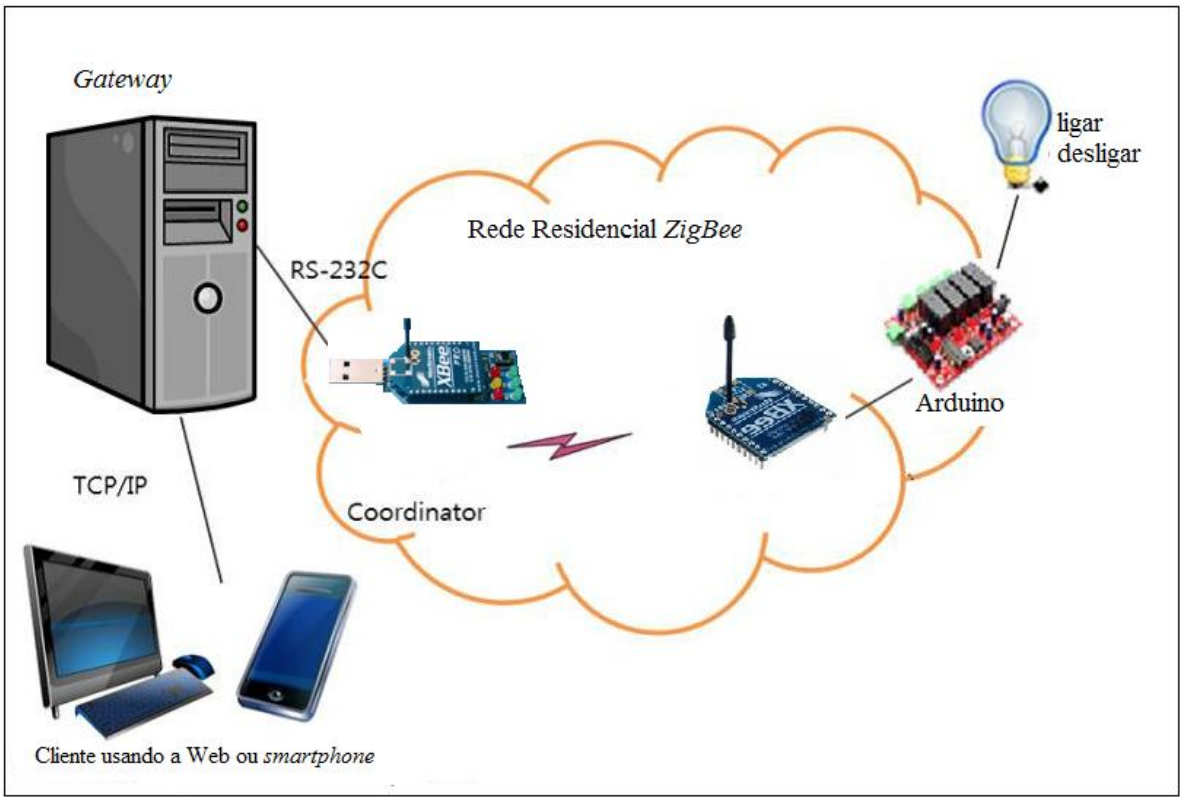

Figura 25 – Lâmpada de teste acoplada a módulo ZigBee através do Arduino.

Arduino consiste num kit que permite, com facilidade, a implementação de projetos que envolvem hardware e software. Essa plataforma é aberta e, por isso, qualquer fabricante pode desenvolver kits baseados no original. Esse kit Arduino é composto de uma placa na qual estão conectados um microcontrolador da Atmel, responsável pela execução da lógica programável, e diversos LEDs e botões conectados as suas portas analógicas e digitais, além de portas seriais, podendo estas configurações variar de acordo com o modelo do Kit. A indicação das portas do microcontrolador pode ser vista na Figura 26.

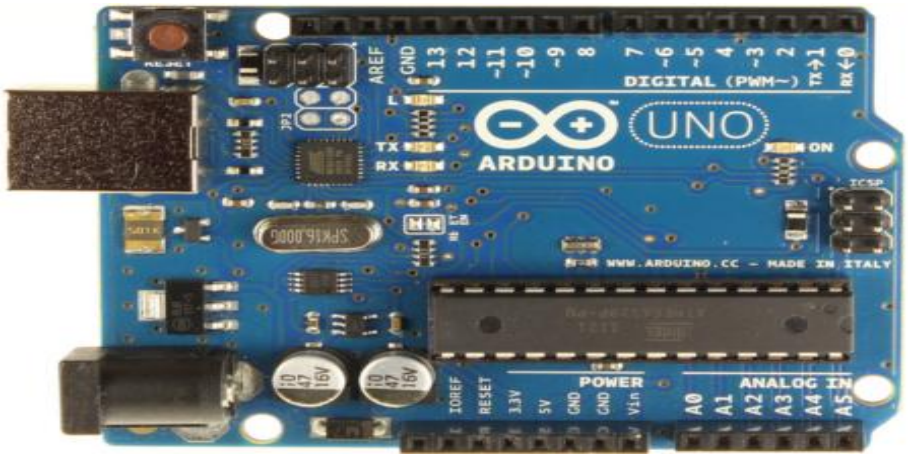

Figura 26 - Elementos de um Kit Arduino.

Para programação do microcontrolador presente no Arduino, é utilizada a linguagem C, além de bibliotecas em C++. Já quanto à escrita dos códigos, pode ser utilizado o ambiente de desenvolvimento integrado (IDE) do Arduino. Essa IDE possui recursos para facilitar a implementação dos projetos no Arduino, tais como diversos exemplos de códigos e uma ferramenta denominada Serial Monitor, possibilitando visualizar os dados recebidos e, também, os dados enviados pela porta serial.

Além da conexão serial, os Kits Arduino ainda possuem suporte a ouros tipos de conexão como Ethernet, Wi-Fi e diversas outras. A expansão das funcionalidades do Arduino, através de conexões de rede, pode ser auxiliada com a utilização de *shields*. Esses *shields* são placas especificamente projetadas para encaixar nas portas do Arduino. No mercado, existem *shields* para ethernet, 802.11, 802.15.4, Bluetooth, controladores de motores de passo, etc.

5.4 Resumo

Este capítulo descreveu a implementação de um *gateway* residencial e de seus componentes, optando-se por fazer uso de protótipos para simulação das funcionalidades de um aparelho real. Na construção dos protótipos, adotou-se o paradigma de orientação a objetos e um processo de desenvolvimento interativo e incremental, de forma que, a cada etapa, novas funcionalidades foram adicionadas ao sistema em construção. Adotaram-se diferentes plataformas de desenvolvimento para implementar cada tipo de simulador dos

aparelhos domésticos, visto que as características de hardware diferem demasiadamente entre os dispositivos domésticos e o *gateway*.

A construção do *gateway* foi dividida em 5 partes, referentes aos componentes Coordenador, Base de Dados, Controlador, Servidor e Repositório. Todos esses componentes foram implementados em um único computador. Para cada componente, foi descrito o ambiente de desenvolvimentos, os *softwares* e *hardwares* utilizados.

Além dos protótipos, construiu-se um ambiente de teste prático envolvendo um dispositivos real (uma lâmpada) acoplado a um módulo ZigBee através da plataforma Arduino.

Na próxima seção, serão avaliadas as funcionalidades do *gateway*, através de testes que enfatizaram a comunicação do *gateway* com as aplicações de controle, a comunicação do *gateway* com os dispositivos residenciais, o acesso aos serviços Web e teste de desempenho.

Capítulo 6 - Experimentos e resultados

Este capítulo tem como propósito a descrição do processo de experimentação do sistema *gateway*, que tem como proposta a disponibilização de eletrodomésticos para aplicações de automação residencial. O capítulo está dividido em testes de integração do *gateway* com dispositivos domésticos e com as aplicações clientes; testes de acesso aos serviços Web disponibilizados; testes de desempenho do sistema. Os resultados devem fazer referências aos requisitos listados no capítulo anterior, na seção de requisitos do sistema.

Durante os testes, o ambiente doméstico, teve os dispositivos organizados em forma de estrela, tendo o *gateway* como elemento central, conforme ilustrado na Figura 27.

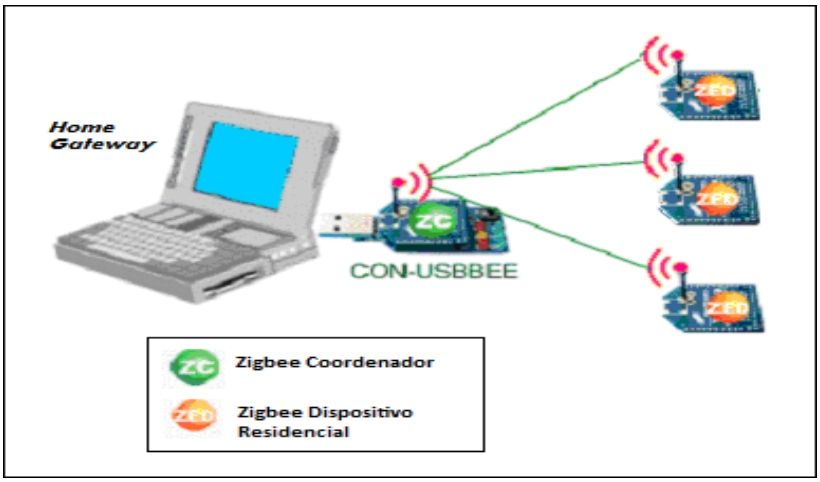

Figura 27 - Rede ZigBee com topologia em estrela

6.1 Testes de comunicação entre o *gateway* e os dispositivos residenciais

A integração entre os dispositivos residenciais e o *gateway* ocorreu de forma bastante coordenada. O *gateway*, através do Controlador, enviava a cada 30 segundos uma mensagem de registro para os dispositivos residenciais que, em uma fração de tempo inferior a 1 segundo, respondiam à solicitação.

Os dispositivos inicialmente não tinham a estrutura de seus perfis cadastrados na base de dados do *gateway*, portanto, durante o primeiro registro, fez-se uso do repositório para localizar, e posteriormente, cadastrá-la na base de dados.

Devido a diversidades de aplicações residenciais possíveis, utilizaram-se aplicações para simular as funcionalidades de aparelhos domésticos, tais como ventilador elétrico, controlador de temperatura. Para não restringir os testes a apenas simuladores, uma plataforma Arduino foi utilizada para testar o funcionamento de uma lâmpada que respondia aos serviços de ligar/desligar, ver Figura 28.

Figura 28 – Experimento com Arduino.

A Figura 29 ilustra o esquema montado para testar o dispositivo residencial de forma prática, havendo um computador pessoal que funciona como *gateway*. Com uma placa CON-USBBEE como coordenador da rede ZigBee, a lâmpada tem acoplada a sua estrutura um placa Arduino integrada com um dispositivos ZigBee. O *gateway* envia os comandos à plataforma ZigBee que processa e executa as funcionalidades liga/desliga.

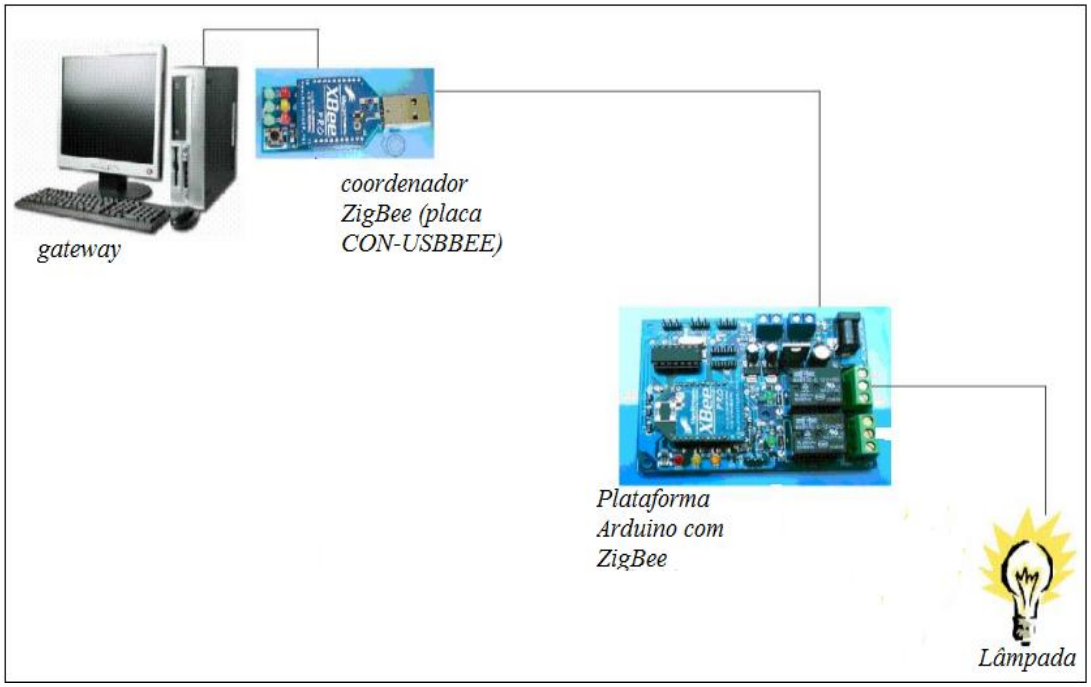

Figura 29 – Esquema do teste prático

Com o resultado deste teste, o sistema mostrou-se adaptável e escalável, já que os adicionamentos de novos eletrodomésticos não influenciaram na programação interna do *gateway*. Com a opção de utilizar o padrão ZigBee, o sistema passou a suportar um número de aproximadamente 65.000 dispositivos, característica das redes ZigBee. Outra característica apresentada foi o suporte à padronização, através da estruturação baseada em Perfil. Com essa estrutura, os desenvolvedores não precisaram estar envolvidos com problemas de normalização e conformidade, apenas precisaram focar nas características dos serviços, proporcionando um conjunto maior de aplicações uniformes criadas, mesmo que por grupos diferentes.

6.2 Testes de acesso aos serviços Web

Para testar os serviços Web disponibilizados, criou-se um painel de controle utilizando a linguagem HTML com a listagem de todos os dispositivos residenciais e seus serviços web disponibilizados pelo *gateway* através de seu componente interno, o Servidor.

Com o painel de controle, pode-se testar cada funcionalidade dos serviços e a integração com os aparelhos da rede de sensores através das mensagens de retorno. O painel está ilustrado na Figura 30.

Com esses resultados, o sistema mostrou-se abstrato, proporcionando uma ferramenta de utilização em que o conhecimento prévio da complexidade estrutural dos dispositivos residenciais não foi necessário. Essa é uma questão importante, pois as aplicações e serviços são desenvolvidos independentes um do outro, em diferentes ambientes de desenvolvimento e sistemas operacionais. A utilização por serviços Web proporciona uma ferramenta de descoberta de serviços e um modelo que facilita a integração e utilização dos serviços descobertos.

Device	Service
Device: XYZ123 Profile: EletricFan	23. turnOn http://localhost:8080/WebServiceREST/zigbee/execute/XYZ123/TURNON/null 24. turnOff http://localhost:8080/WebServiceREST/zigbee/execute/XYZ123/TURNOFF/null 25. setSpeedLevel http://localhost:8080/WebServiceREST/zigbee/execute/XYZ123/SETSPEEDLEVEL/null 26. readStatus http://localhost:8080/WebServiceREST/zigbee/execute/XYZ123/READSTATUS/null 27. readStatus http://localhost:8080/WebServiceREST/zigbee/execute/XYZ123/READSTATUS/null
Device: AB12CD Profile: Lamp	29. TURNOFF http://localhost:8080/WebServiceREST/zigbee/execute/AB12CD/TURNOFF/null 28. TURNON http://localhost:8080/WebServiceREST/zigbee/execute/AB12CD/TURNON/null
Device: 10AB01 Profile: TempSensor	1. readStatus http://localhost:8080/WebServiceREST/zigbee/execute/10AB01/READSTATUS/null

Figura 30– Experimento com Painel de Controle

6.3 Testes de comunicação entre o *gateway* e as aplicações clientes

A aplicação cliente foi criada com o objetivo de testar o acesso aos dispositivos residenciais através dos serviços Web disponibilizados pelo *gateway*. Na aplicação cliente, utilizou-se um simulador de *smartphnone Android* que acessou os serviços e obteve as respostas dos dispositivos da rede sem mais complexidade.

Durante o desenvolvimento da aplicação cliente, fez-se necessário o conhecimento unicamente do funcionamento dos serviços Web, não necessitando conhecer o modo de como as aplicações residenciais trabalham para responder as requisições. Na comunicação entre a aplicação cliente e o *gateway*, utilizou-se a interface *RESTful*. A Figura 31 ilustra o simulador da aplicação cliente utilizado.

Os resultados apresentados demonstraram que as interfaces de comunicação foram suficientes para realizar a integração entre as aplicações clientes e os eletrodomésticos, demonstrando ser um sistema de fácil integração.

Figura 31 – Experimento com simulador *Android*.

6.4 Testes de desempenho

Para teste de desempenho da solução proposta, utilizou-se a ferramenta JMeter (JMETER), programa que serve para fazer testes de performance, carga e *stress*. Este é um software livre, sendo parte do projeto Jakarta (JAKARTA) da Apache Software Foundation.

O serviço Web requisitado durante todo o teste foi o de leitura de temperatura ambiente, por possuir um funcionamento simples de leitura e controle, características encontradas nos dispositivos residenciais (DELICATO, 2006).

O ambiente de teste é composto de um grupo de usuários, que servem para configurar quantas pessoas (virtuais) serão utilizadas no teste e a quantidade de interações dos processos (quantidade de vezes que será executado o teste).

A Figura 32 ilustra os tempos calculados no sistema, sendo o T1 correspondente ao tempo entre a solicitação da requisição e o recebimento dela pelo *gateway*. T2 corresponde ao intervalo em que as tarefas solicitadas aguardam na fila para serem processadas (T2') e o tempo de processamento da solicitação (T2''). T3 corresponde ao intervalo da solicitação

da tarefa pelo *gateway* e o recebimento pelo dispositivo residencial. T4 corresponde ao tempo da mensagem de retorno do dispositivo residencial ao *gateway* e T5 corresponde ao recebimento da mensagem pela aplicação cliente.

Na tabela 9, encontram-se registrados todos os tempos levantados de cada requisição, em que o tempo total (T1+T2+T3+T4+T5) corresponde ao tempo total que as aplicações clientes deverão aguardar para receber o resultado da requisição e o tempo efetivo (T2+T3+T4–T2') corresponde ao tempo de processamento da tarefa pelo *gateway*, e a resposta dos dispositivos residenciais, desconsiderando o tempo de espera na fila para processamento. Todos os tempos estão calculados em milissegundos (ms).

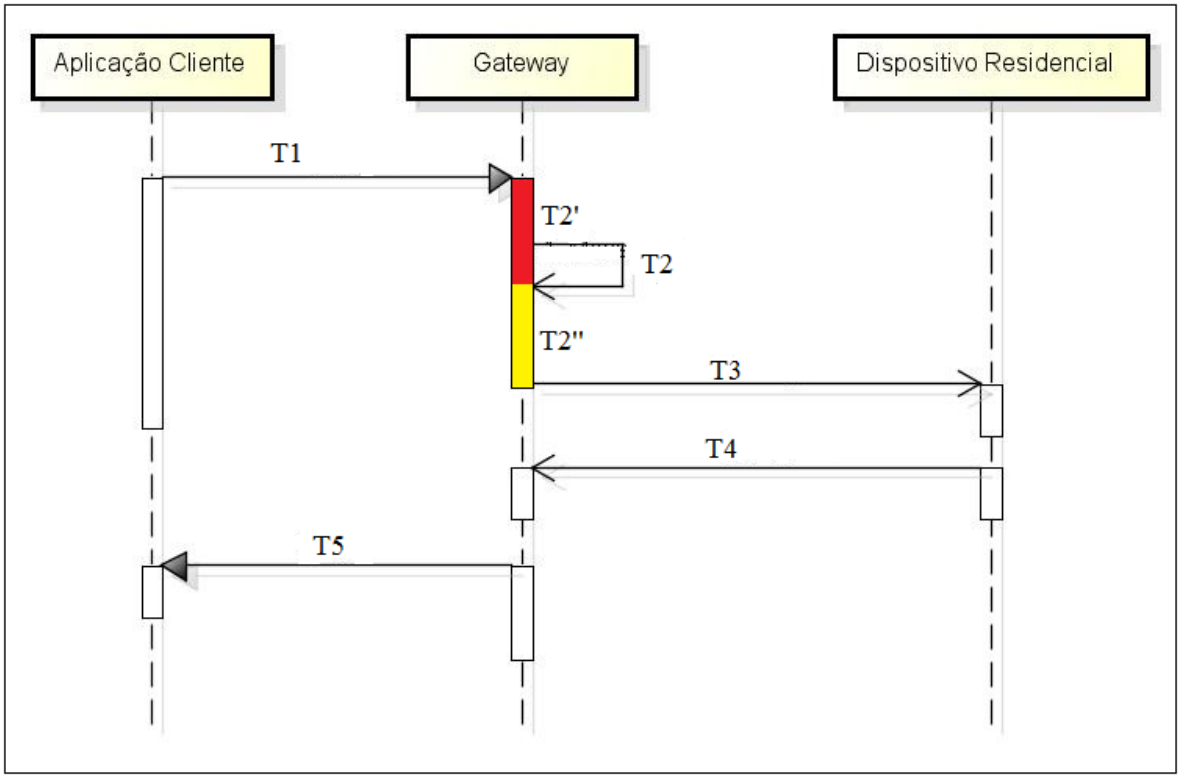

Figura 32 –Seqüência dos tempos calculados

Nas primeiras amostragens, foram realizadas requisições seqüenciais, uma após a outra, mantendo o número de usuários fixo em um e variando o número de requisições, Tabela 9 (cenários 1, 2, 3, 4). O tempo médio de processamento e retorno para as aplicações clientes foi de aproximadamente um segundo. Como as requisições eram realizadas após o término anterior, não houve interferência no tempo de resposta gerado pela espera de processamento da requisição anterior.

O gerenciamento das requisições pelo *gateway* segue o conceito de fila. A primeira que chega será a primeira a sair. Dessa forma, se um mesmo serviço Web for requisitado por várias aplicações clientes ao mesmo instante, essas requisições farão parte de uma fila de espera e o tempo de retorno às aplicações clientes variarão de acordo com a posição na fila de espera. Para ilustrar esse cenário, realizaram-se testes, Tabela 9 (cenários 5, 6, 7), simulando vários usuários acessando um único serviço ao mesmo instante. Com base nos resultados da Tabela 9, verificou-se que o tempo de processamento manteve a média de aproximadamente um segundo, no entanto, o tempo de resposta ao usuário aumentou conforme o tamanho da lista de espera e a posição da requisição na lista. A Tabela 10 sumariza os valores amostrados nas experiências e apresenta outros valores como a média do tempo de resposta às aplicações clientes, o total de tempo de processamento consumido no gateway para processamento de todas as requisições, o tempo mínimo e máximo de resposta apresentado.

Com esses resultados, verificou-se que o sistema possui um tempo de processamento de aproximadamente um segundo, no entanto, o tempo de resposta ao cliente é influenciado pelo número de requisições a sua frente na lista de espera. O sistema também se mostrou escalável, com capacidade de lidar com altas taxas de acesso, grandes volumes de dados e cargas elevadas de comunicação.

Resultado (ms)					
Cenário	Usuário Virtual	Requisição	Amostra	Tempo Total	Tempo Efetivo
1	1	1	#1	1204	1024
2	1	1ª	#1	546	546
	1	2ª	#2	1086	1086
3	1	1ª	#1	1076	1076
	1	2ª	#2	1027	1027
	1	3ª	#3	988	988
	1	4ª	#4	1001	1001
4	1	1ª	#1	347	347
	1	2ª	#2	1086	1086
	1	3ª	#3	1008	1008
	1	4ª	#4	895	895
	1	5ª	#5	968	968
	1	6ª	#6	975	975
	1	7ª	#7	947	947
	1	8ª	#8	989	989
5	1	1ª	#1	546	546
	2	1ª	#2	1102	556
6	1	1ª	#1	434	434
	2	1ª	#2	1184	750
	3	1ª	#3	1930	746
	4	1ª	#4	2704	774
7	1	1ª	#1	1290	1290
	2	1ª	#2	2144	854
	3	1ª	#3	2990	846
	4	1ª	#4	3937	947
	5	1ª	#5	4777	840
	6	1ª	#6	5636	859
	7	1ª	#7	6530	894
	8	1ª	#8	7487	957

Tabela 7 – Teste de Desempenho – Resultado das Amostragens

Resultado Agregado (ms)						
Cenário	Usuário Virtual	Requisição	Média	Total	Mínimo	Máximo
1	1	1	1024	1024	-	-
2	1	2	816	1632	546	1086
3	1	4	1023	4092	988	1076
4	1	8	901,875	7215	347	1086
5	2	1	901,875	7215	347	1086
6	4	1	824	1102	546	1102
7	8	1	4348,875	7487	1290	7487

Tabela 8 – Teste de Desempenho – Resultado Agregado

6.5 Resultados

Nesta seção, foram realizados testes para verificar a adequabilidade do emprego de um *gateway* orientado a serviços através de serviços Web *RESTful* para integração de dispositivos residenciais. Os testes foram realizados utilizando o padrão de comunicação ZigBee junto a um simulador de controle de temperatura. O uso do *gateway* permitiu que esses dispositivos fornecessem suas funcionalidades através de uma interface *RESTFul*, permitindo que a integração com os dispositivos residenciais ocorressem no nível de aplicação. O *gateway* atuou como um registrador, mantendo uma base de dados com a descrição dos serviços de cada dispositivo da rede. Diferentemente dos trabalhos relacionados, serviços Web *RESTful* foram adotados em preferência a outras abordagens de serviços Web (SOAP, XML, WSDL), permitindo a construção de aplicações com baixo acoplamento, com arquiteturas escaláveis e mais simples.

Com esses resultados, verificou-se que o sistema possui um tempo de processamento de aproximadamente um segundo, tempo total que as aplicações clientes deverão aguardar para receber o resultado da requisição. O tempo de resposta ao cliente é influenciado pelo número de requisições presentes na lista de espera. Além dos resultados de desempenho, alguns requisitos apresentados no capítulo dedicado à arquitetura do sistema *gateway* puderam ser observados. O sistema mostrou-se abstrato, adaptável, escalável e de integração com outros sistemas.

A utilização de interfaces micro processadas, como a plataforma Arduino, responsável por receber as mensagens enviadas pelo *gateway* e a execução das funcionalidades dos dispositivos domésticos, mostrou-se eficaz, no entanto, a depender das

funcionalidades e da estrutura dos dispositivos domésticos, essa pode se tornar uma tarefa complexa e intrusiva na estrutura do dispositivo.

Por fim, verificou-se que a utilização de um único *gateway* como ponte de interligação de eletrodomésticos com as aplicações clientes apresenta uma limitação, a desconexão do *gateway*, que não possibilitaria a comunicação entre os dispositivos residenciais e as aplicações clientes. Tal limitação foi considerada aceitável em um ambiente de automação doméstico no qual o monitoramento dos dispositivos tem como objetivo principal propiciar conforto e economia aos seus proprietários não se tratando de automação de tempo real.

6.6 Resumo

O principal objetivo deste capítulo foi avaliar as funcionalidades do *gateway* residencial descritas nos capítulos anteriores. Os testes descritos enfatizaram as comunicações do *gateway* com as aplicações clientes, *gateway* com os dispositivos domésticos e o acesso aos serviços Web disponibilizados às aplicações clientes. Outro tipo de teste realizado foi o de desempenho, no qual se analisaram alguns parâmetros, como o tempo médio de troca de informações entre os aplicativos interativos e os dispositivos domésticos.

No teste de comunicação com os dispositivos residenciais, o sistema mostrou-se adaptável e escalável, já que os adicionamentos de novos eletrodomésticos não influenciaram na programação interna do *gateway*. Outra característica apresentada foi o suporte à padronização, através da estruturação baseada em Perfil, proporcionando um conjunto maior de aplicações uniformes criadas, mesmo que por grupos diferentes.

No teste de acesso aos serviços Web, o sistema mostrou-se abstrato, proporcionando uma ferramenta de utilização em que o conhecimento prévio da complexidade estrutural dos dispositivos residenciais não foi necessário.

No teste de comunicação com as aplicações clientes, os resultados apresentados demonstraram que as interfaces de comunicação foram suficientes para realizar a integração entre aquelas e os eletrodomésticos, demonstrando ser um sistema de fácil integração.

Em relação ao teste de desempenho, o sistema apresentou um tempo de processamento de aproximadamente um segundo, no entanto, o tempo de resposta ao cliente é influenciado pelo número de requisições a sua frente na lista de espera. O sistema também se mostrou escalável, com capacidade de lidar com altas taxas de acesso, grandes volumes de dados e cargas elevadas de comunicação.

Para automação das funcionalidades dos dispositivos domésticos, foi utilizada a plataforma Arduino, que executou as tarefas de recebimento e envio de mensagens *gateway*, além da execução das funcionalidades correspondentes nos dispositivos domésticos.

Por fim, verificou-se que a utilização de um único *gateway*, como ponte de interligação de eletrodomésticos com as aplicações clientes, apresenta uma limitação: a desconexão do *gateway*, que não possibilitaria a comunicação entre os dispositivos residenciais e as aplicações clientes. Tal limitação foi considerada aceitável em um ambiente de automação doméstico, cujo monitoramento dos dispositivos tem como objetivo principal propiciar conforto e economia aos seus proprietários, não se tratando de automação de tempo real.

No próximo capítulo, serão realizadas as considerações finais sobre este trabalho, destacando a sua importância e contribuições, bem como identificando pontos de melhorias e sugestões para trabalhos futuros.

Capítulo 7 – Considerações Finais

Este trabalho apresentou um sistema *gateway* orientado a serviços para dispositivos residenciais. Tal sistema permite que as funcionalidades dos eletrodomésticos (sensores e atuadores) sejam disponibilizadas na forma de serviços Web que foram utilizados na definição da interface da aplicação e como base de interação entre os dispositivos. A adoção dos princípios REST permite que a integração dos dispositivos ocorra no nível de aplicação, acima da conectividade de rede (DELICATO, 2006).

Neste trabalho, os dispositivos residenciais desempenham o papel de provedores de dados brutos, enquanto que o *gateway* atua como provedor de serviços Web para as aplicações clientes. No *gateway*, foi fornecida a interface *RESTful* de acesso a todos os serviços disponibilizados. O *gateway* também é responsável por acessar os serviços dos dispositivos residenciais e por receber as mensagens enviadas por esses dispositivos.

Na implementação da arquitetura orientada a serviços (SOA), serviços Web *RESTful* foram adotados em preferência a outras abordagens de serviços Web (SOAP, XML, WSDL) por permitir que sejam criadas aplicações com baixo acoplamento, com arquiteturas escaláveis e mais simples. Além disso, essa abordagem está diretamente relacionada ao uso do HTTP e existência de diversas bibliotecas e clientes HTTP disponíveis para diferentes linguagens.

Arquiteturas SOA tradicionais, que utilizam o protocolo SOAP, implementam diferentes camadas de abstrações sobre o protocolo HTTP, limitam seus serviços e impedem que façam uso total das capacidades da Web. Por isso, serviços Web *RESTful* estão se tornando uma das mais onipresentes e leves tecnologias de integração de dispositivos embarcados (MAYER, 2010). O uso de padrões Web para dar suporte à integração entre com dispositivos físicos tem sido considerado uma solução promissora. Embora o uso do HTTP como protocolo de transporte introduza uma sobrecarga de comunicação e aumente a latência média, os parâmetros de qualidade providos em geral são suficientes, de forma que tais atrasos não prejudiquem a comunicação entre dispositivos (HARRISON; UCKELMANN, 2011).

Com os resultados obtidos nos experimentos realizados, foi observado que o atraso gerado pela adoção de serviços *RESTful* é aceitável para situações nas quais as informações

dos dispositivos residenciais não precisam de uma comunicação em tempo real (aplicações com atrasos de milissegundos não são tolerados). Situações que toleram pequenos atrasos até que a informação dos dispositivos seja recebida pelos clientes, o emprego dos serviços *RESTful* e a integração de dispositivos residenciais à Web se apresentam como uma solução promissora.

As vantagens oferecidas pela introdução de suporte aos serviços Web diretamente no nível de dispositivos são benéficas para o desenvolvimento de uma nova geração de aplicações que conterão informações do mundo real. O emprego dos serviços Web como base comum para diferentes dispositivos torna muito mais simples a programação dos mesmos. Além disso, a maioria das linguagens de programação suportam serviços Web.

A integração na Web aumenta as possibilidades de aplicações que podem ser criadas no topo das redes domésticas e permite que essas redes sejam amplamente utilizadas. Integrar dispositivos residenciais à Web alavanca a concretização da Internet das Coisas por permitir que esses dispositivos possam ser vistos como fornecedores de serviços do mundo real para o mundo digital (MAYER, 2010). Ao serem vistos e estarem disponíveis como serviços Web, os dispositivos residenciais podem ser utilizados como provedores de serviços para diferentes aplicações clientes.

Para comunicação do *gateway* com os dispositivos domésticos, foi utilizado o padrão ZigBee, que após uma análise comparativa com outros padrões *wireless* como Wi-Fi e Bluetooth, mostrou-se ser a solução adequada para utilização em sistemas de monitoramento, tais como sistemas de automação doméstica, segurança, controle de iluminação, acesso e etc. Principalmente pelas características de baixo consumo de energia, habilidade de permanecer inativo por um longo período, possibilidade de implementação de redes com elevado número de dispositivos, simplicidade e baixo custo.

Por outro lado, alguns pontos requerem melhorias, como o uso de interfaces micro processadas em dispositivos domésticos, apesar das experiências terem mostrado ser uma solução viável para comunicação de dispositivos com funcionalidades predominantemente mecânicas, o desenvolvimento dessas interfaces, a depender das funcionalidades e da estrutura dos dispositivos domésticos, pode se tornar uma tarefa complexa e intrusiva.

A conclusão final é que esse trabalho obteve êxito em relação aos objetivos definidos no início desta pesquisa. As principais tecnologias de comunicação foram

avaliadas de modo a escolher as mais adequadas ao ambiente doméstico. Foram estudadas as tecnologias de construção de serviços Web a fim de criar uma estrutura mais adequada à disponibilização dos serviços às aplicações clientes. Foram criados casos de testes para validar esse ambiente de integração entre a rede sensorial sem fio e aplicações de automação residencial.

Ficam para trabalhos futuros a adoção e avaliação de outros modelos de comunicação baseados em push. Tais modelos de comunicação são importantes no âmbito das redes domésticas, pois para aplicações de monitoração é comum a ocorrência imprevista de eventos. Um modelo de comunicação assíncrona que pode ser adotado é o PubSubHuhhuh(PUBSUBHUBBUB) o qual utiliza nós intermediários para envio de notificações aos clientes que tenham anteriormente assinado feeds associados ao evento. Outro modelo que vem emergindo está presente no HTML5 (FRANÇA, 2011) o qual fornece meios que permitem que o servidor notifique o cliente da ocorrência de um evento. Além desses, um serviço simples disponível da própria Web que pode ser adotado é o twitter, visto que seria relativamente simples enviar mensagens de publicação para as contas dos clientes que desejassem ser notificados da ocorrência de um evento.

Referências Bibliográficas

AKYILDIZ, F., “Wireless Sensor Networks: A Survey”, Computer Networks, vol. 38, no. 4, pp. 393-422, 15 Mar. 2012.

AL-JAROODI, J.; MOHAMED, N. A Survey on Service-Oriented Middleware for Wireless Sensor Networks, Journal of Service Oriented Computing and Applications on, vol. 5, no. 2, pp. 71-85. 01 Jun 2011.

AL-JAROODI, J.; MOHAMED, N., “Service-Oriented Middleware: A Survey”, Journal of Network and Computer Applications on, vol. 35, no. 1, pp. 211-220. Jan. 2012.

AVILÉS, E.; GARCÍA, J. TinySOA: a service-oriented architecture for wireless sensor networks, Journal Service Oriented Computing and Applications, vol. 3, no., 2, pp. 99-108, 01 Jun. 2009.

BARNAGHI, P. et al. A Service Oriented Middleware Architecture for Wireless Sensor Networks. In: Future Network Summit 2010, Florence, Italy. 16-18 Jun. 2010.

BONZANI, C. Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes: Uma introdução aos sistemas domóticos. 2004. Universidade de São Paulo, Dissertação de Mestrado.

BONZANI, C. Residências Inteligentes. Editora Livraria da Física, 2004b.

BONZANI, C.; NETTO, M. The engineering of micro agents in smarts environments. International Journal of Knowledge-based and Intelligent Engineering Systems, v. 13, n. 1, p. 31-38, 2009.

BRAY, J.; STURMAN, C. Bluetooth: connect without cables. 2ª ed., New Jersey, EUA: Prentice Hall PTR, 2001.

CALLAWAY, E. et al. Home Networking with IEEE 802.15.4: A Developing Standard for Low Rate Wireless Personal Area Networks. IEEE Communications Magazine, pp 70-77, Agosto 2010.

CERAMI, E. Web Services Essentials - Distributed Applications with XML-RPC, SOAP,UDDI & WSDL. 1º ed., Sebastopol, EUA: O’Reilly, 2002.

CHENGBO, Y. et al. ZigBee Wireless Sensor Network in Environmental Monitoring Applications, Wireless Communications, Networking and Mobile Computing (WiCom 2009). 5th International Conference on, vol., no., pp.1, 5; 24-26 Set. 2009.

DELICATO, F. et al., Wireless Sensor Networks as a Service, Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on , vol., no., pp.410-417, 22-26 Mar. 2010.

DELICATO, F. Middleware Baseado em Serviços para Redes de Sensores sem Fio, Tese Doutorado, Universidade Federal do Rio de Janeiro, Brazil on, Anais do XXVi Congresso da SBC. XIX Concurso de Teses e Dissertações, Campo Grande MS, Jun. de 2006.

DIAS NETO, A.; MACHADO, L. REST com Struts 2. 48ª ed., São Paulo: Java Magazine, 2007. Pag. 58-66.

DING, G. et al. Tree-Based Data Broadcast in IEEE 802.15.4 and ZigBee Networks, Mobile Computing, IEEE Transactions on , vol.5, no.11, pp.1561,1574; Nov. 2006.

DUNKELS, A. et al. Connecting Wireless Sensornets with TCP/IP Networks, Wired/Wireless Internet Communications (WWIC 2004) on, vol., no., pp.143-152, Feb. 2004.

DUNKELS, A.; ALONSO, J.; VOIGT, T. Making TCP/IP Viable for Wireless Sensor Networks", European Conference on Wireless Sensor Networks (EWSN 2004), Jan. 2004.

EGAN, D. The emergence of ZigBee in building automation and Industrial control. The IEE Computing and Control Engineering, vol. 16, issue 2, p. 14-19, maio 2005.

ENDREI, M., et al. Patterns: Service-Oriented Architecture and Web Services. 1ª ed., IBM Redbooks publication, 2004.

FARAHANI, S. “Zigbee Wireless Networks and Transceivers”. 1ª ed., Burlington, EUA: Elsivier, 2008.

FERREIRA FILHO, O. Serviços Semânticos: Uma abordagem RESTful. Dissertação de Mestrado, USP, Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde11082010-120409/pt-br.php/>. Acessado em 15 de julho de 2013.

FRANÇA, T. et al. C.M. Web das Coisas: Conectando Dispositivos Físicos ao Mundo Digital. IN: Fabíola Gançalves Pereira Greve. Ronaldo Alves. (Org). Minicursos / XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, v.1, p. 103-150, Porto Alegre: Sociedade Brasileira de Computação – SBC, 2011.

GILL, K. at al., A zigbee-based home automation system, Consumer Electronics, IEEE Transactions on , vol.55, no.2, pp.422,430, Mai. 2009.

GLOMBITZA, N. et al., Self-Description and Protocol Conversion for a Web of Things, Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on , vol., no., pp.229-236, 7-9 Jun. 2010.

GUINARD, D. et al. Towards Physical Mashups in the Web of Things. In Proceedings of IEEE Sixth International Conference on Networked Sensing Systems, Pittsburgh, USA, on , vol., no., pp.196 - 199, 7-9 Jun. 2009.

GUINARD, D.; TRIFA, V. Towards the Web of Things: Web Mashups for Embedded Devices". In Proceedings of Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), International World Wide Web Conferences, Abr. 2009.

HARRISON, M.; UCKELMANN D., Architecting the Internet of Things. 1ªed., New York, EUA: Spinger, 2011.

HICKSON, I. HTML5 - A vocabulary and associated APIs for HTML and XHTML. W3C Working Draft 25 May 2011, World Wide Web Consortium. Disponível em: http://www.w3.org/TR/html5/. Acessado em agosto de 2013.

HUI, D.; HAN, R. Unifying micro sensor networks with the Internet via overlay networking [wireless networks], Local Computer Networks, 2004. 29th Annual IEEE International Conference on, vol., no., pp.571, 572, 16-18 Nov. 2004.

JAKARTA. Disponível em: http://jakarta.apache.org/. Acessado em 04 de setembro de 2013.

JARDIM, F. Treinameno Avançado em Redes Wireless. 1ª ed. São Paulo: Digerati Books, 2007.

JEONG-HEE, K. et al. Address Internetworking between WSNs and Internet supporting Web Services, Multimedia and Ubiquitous Engineering (MUE 2007). International Conference on, vol., no., pp.232, 240, 26-28 Abr. 2007.

JMETER. Disponível em: http://jmeter.apache.org/. Acessado em 04 de setembro de 2013.

JOSUTTIS, N. M. SOA in Practice - The Art of Distributed Systems Design. 1ª ed., Sebastopol, EUA: O'Reilly Media, 2007.

MACKENZIE, C. et al. OASIS Reference Model for Service Oriented Architecture 1.0", SOA Reference Model TC, OASIS Standard, 2008. Disponível em: <http://docs.oasis-open.org/soa-rm/v1.0/>. Acessado em 20 de julho de 2013.

MARKS, E.; BELL, M. "Service-Oriented Architecture: a planning and implementation guide for business and technology", 1ª ed., New Jersey, EUA: John Willey & Sons Inc., 2006.

MARZULLO, F.P. SOA na Prática. 1ª ed., Rio de Janeiro: Novatec, 2009.

MAYER, S. Deployment and Mashup Creation Support for Smart Things, Institute for Pervasive Computing Department of Computer Science ETH Zurich, 2010. Disponível em <http://www.vladtrifa.com/files/publications/Mayer10.pdf>. Acessado em 02 de Agosto de 2013.

MILLER, M. Descobrindo Bluetooth. 1ª ed., Rio de Janeiro: Editora Campus, 2001.

NEWCOMER, E. Understanding Web Services XML, WSDL, SOAP, and UDDI. 1ª ed., Boston, EUA: Addison-Wesley, 2002.

OTHMAN, Y.; et al. The Design and Implementation of a Web Service Framework for Individual Nodes in Sinkless Wireless Sensor Networks, Computers and Communications, 2007 (ISCC 2007). 12th IEEE Symposium on, vol., no., pp.941-947, 1-4 Jul. 2007.

PAUTASSO, C. et al. RESTFul Web Services vs. Big Web Services: Making the Right Architectural decision. 17th International Conference on World Wide Web on, vol., no., pp.805-814, 21-25 Abr. 2008.

PERUMAL, T. et al, Interoperability for Smart Home Environment Using Web Services, Internacional Journal of Smart Home, vol. 2, no. 4, Out. 2008.

POTTS, S.; STEPHEN, M. Aprenda em 24 Horas Web Services. 1ª ed. Rio de Janeiro: Editora Elsevier, 2003.

PRIYANTHA, N. et al. Tiny web services: design and implementation of interoperable and evolvable sensor networks, In the Proceedings of the 6th ACM conference on Embedded network sensor systems on, vol., no., pp. 253-266, Raleigh, NC, EUA. 2008.

PUBSUBHUBBUB. Disponível em: http://code.google.com/p/pubsubhubbub/. Acessado em agosto de 2013.

RODRIGUEZ, J.; DEMSAK, D. Lightweight SOAs: Exploring Patterns and Principles of a New Generation of SOA Solutions. Journal The Architecture. 22ª ed., 2013.

ROUACHED, M.; BACCAR, S.; ABID, M. RESTful Sensor Web Enablement Services for Wireless Sensor Networks, Services (SERVICES), 2012 IEEE Eighth World Congress on , vol., no., pp.65,72, 24-29 Jun. 2012

RUBY, S.; RICHARDSON, L. RESTful Web Services. 1ª ed., Sebastopol, EUA: O'Reilly, Mai. 2007

SAUDATE, A. SOA aplicado: Integrando com web services e além, 1ª ed., São Paulo: Casa do Código, 2012.

SHIZHUANG, L.; JINGYU, L.; YANJUN, F. ZigBee Based Wireless Sensor Networks and Its Applications in Industrial, Automation and Logistics, 2007 IEEE International Conference on , vol., no., pp.1979,1983; 18-21 Ago. 2007.

TANENBAUM, A. Redes de Computadores. 3º ed., Rio de Janeiro: Editora Campus, 2003.

W3C. Web Services Architecture: W3C Working Group 2004. Disponível em: <http://www.w3.org/TR/ws-arch/>. Acessado em 20 de agosto de 2013.

WANG, C.; SOHRABY, K. Voice Communications over ZigBee Networks. IEEE Communications Magazine, pp 121-127, Janeiro 2008.

WATANABE, C. Comparação do Desempenho de Técnicas CDMA para Comunicação sem Fio em Ambientes Interiores, Dissertação (mestrado em Engenharia Elétrica) – Universidade de Campinas, 2010.

WEERAWARANA, S. et al. Web Services Platform Architecture: SOAP, WSDL, Ws-Policy, WS-Adressing, WS-BPEL, WS-Reliable, Messaging, and more. 1º ed., New Jersey, EUA: Prentice Hall PTR, 2005.

YICK, J. et al. Wireless sensor network survey. Computer Networks, Vol. 52, No. 12, p. 2292-2330, 2008.