

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RODRIGO HOMMERDING ESCHNER

**Sistema de Automação Residencial Baseado  
em Sensores ZigBee**

Trabalho de Diplomação.

Prof. Dr. João Cesar Netto  
Orientador

Porto Alegre, 5 de dezembro de 2011.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Prof<sup>a</sup>. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*Dinheiro até pode trazer felicidade mas não satisfação! A  
satisfação vem quando ele é conquistado honestamente!*

Israel Ziller

# SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>6</b>
<b>LISTA DE TABELAS .....</b>	<b>7</b>
<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>8</b>
<b>RESUMO.....</b>	<b>9</b>
<b>ABSTRACT.....</b>	<b>10</b>
<b>1 INTRODUÇÃO .....</b>	<b>11</b>
1.1 Contextualização.....	11
1.2 Motivação .....	11
1.3 Objetivo Principal do Trabalho .....	12
<b>2 PRINCIPAIS CONCEITOS RELACIONADOS .....</b>	<b>13</b>
2.1 O Padrão ZigBee.....	13
2.1.1 Camada Física .....	14
2.1.2 Camada MAC.....	14
2.1.3 Camada de Rede .....	15
2.1.4 Camada de Aplicação .....	15
2.1.5 Características.....	16
2.1.6 Funcionamento .....	16
2.1.7 Topologias de Rede .....	17
2.2 Módulo XBee.....	18
<b>3 IMPLEMENTAÇÃO DO PROJETO.....</b>	<b>21</b>
3.1 Funcionamento Geral do Sistema .....	21
3.2 Sensores e Dispositivos ZigBee .....	22
3.2.1 Sensor de Movimento.....	22
3.2.2 Sensor de Chuva .....	23
3.2.3 Sensor de Temperatura .....	23
3.2.4 Dispositivos .....	24
3.3 Arquitetura do Sistema .....	25
3.4 Sistema de Automação .....	26
3.4.1 Configuração dos Módulos XBee .....	26
3.4.2 Cadastro dos Componentes .....	26
3.4.3 Configuração do Cenário.....	27

3.4.4	Sensores .....	29
3.4.5	Acionamento.....	29
3.4.6	Estrutura do Software .....	30
<b>4</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>32</b>
4.1	Conclusões .....	32
4.2	Trabalhos Futuros .....	32
	<b>REFERÊNCIAS.....</b>	<b>34</b>
	<b>APÊNDICE 1 - DIAGRAMA DE CASOS DE USO.....</b>	<b>35</b>
	<b>APÊNDICE 2 - DIAGRAMA ER .....</b>	<b>38</b>

## LISTA DE FIGURAS

Figura 2.1: Camadas do padrão ZigBee (KAUR, AHUJA, 2011) .....	13
Figura 2.2: Topologia estrela.....	17
Figura 2.3: Topologia em árvore .....	17
Figura 2.4: Topologia em malha.....	18
Figura 2.5: Módulo XBee-PRO.....	18
Figura 2.6: Estrutura do frame XBee .....	19
Figura 3.1: Visão geral do funcionamento do sistema .....	22
Figura 3.2: Sensor de movimento com o módulo XBee .....	23
Figura 3.3: Sensor de chuva junto ao módulo XBee .....	23
Figura 3.4: Componente LM34 conectado ao módulo XBee.....	24
Figura 3.5: Protótipo do dispositivo acionador .....	25
Figura 3.6: XBee Dongle.....	25
Figura 3.7: Cadastro dos sensores e dispositivos .....	27
Figura 3.8: Configuração do cenário por sensor.....	28
Figura 3.9: Adição de regras e de dispositivos .....	29
Figura 3.10: Threads do módulo de controle.....	30
Figura A1.1: Diagrama de casos de uso do sistema .....	35
Figura A2.1: Diagrama ER do sistema.....	38

## LISTA DE TABELAS

Tabela 2.1: Faixas de frequência do padrão IEEE 802.15.4.....	14
Tabela 2.2: Principais comandos XBee .....	20

## LISTA DE ABREVIATURAS E SIGLAS

AES	<i>Advanced Encryption Standard</i>
API	<i>Application Program Interface</i>
APS	<i>Application Support Sublayer</i>
CSMA-CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
GTS	<i>Guaranteed Time Slot</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
JSP	<i>Java Server Pages</i>
LED	<i>Light Emitting Diode</i>
MAC	<i>Medium Access Control</i>
OSI	<i>Open Systems Interconnection</i>
PAN	<i>Personal Area Network</i>
PDU	<i>Protocol Data Unit</i>
PHY	<i>Physical Layer</i>
RF	<i>Radio Frequency</i>
USB	<i>Universal Serial Bus</i>
VCC	Voltagem Corrente Contínua
ZDO	<i>ZigBee Device Object</i>

## RESUMO

Após presenciar o avanço da automação industrial e predial, o Brasil está buscando, cada vez mais, os serviços de automação residencial, um mercado emergente no país. Porém, a maioria dos sistemas atuais se limita à possibilidade de controlar a residência com a intervenção do usuário, utilizando poucos recursos que atuam de forma realmente automática.

Este trabalho consiste na implementação de um sistema de automação residencial baseado em diversos tipos de sensores sem fio. A atuação do sistema ocorre de forma automática, somente com as informações adquiridas desses sensores (movimento, temperatura, umidade, luminosidade, entre outros). Eles utilizam o protocolo ZigBee como meio de comunicação, formando uma rede de sensores sem fio.

**Palavras-Chave:** Automação Residencial, ZigBee, Sensores.

## **ABSTRACT**

After witnessing the advance of industrial automation, Brazil is seeking the services of home automation, an emerging market in the country. However, most current systems are limited to controlling the residence with user intervention, with few resources that works automatically.

This work consists in the implementation of a home automation system based on various types of wireless sensors. The operation of the system occurs automatically, only with the information acquired from these sensors (motion, temperature, humidity, luminosity, etc.). They use the ZigBee protocol to communicate, forming a network of wireless sensors.

**Keywords:** Home Automation, ZigBee, Sensors.

# 1 INTRODUÇÃO

## 1.1 Contextualização

As áreas de automação, tanto residencial como industrial, vem buscando cada vez mais as novas tecnologias de comunicação sem fio. Em função disso, o mercado de redes de transmissão de dados também está se expandindo, principalmente em redes sem fio para ambientes locais. Nesse contexto, foi criado o protocolo ZigBee, com o objetivo de padronizar o desenvolvimento de produtos nessa área.

O mercado de automação residencial está em um grande momento, impulsionado por essas novas tecnologias sem fio, que proporcionam uma grande facilidade, principalmente na instalação, pois evita a necessidade de reformas na residência. Outro grande motivo dessa expansão é a redução dos custos dessas tecnologias, o que está tornando o mercado de automação cada vez mais popular e acessível.

O protocolo ZigBee é um dos fatores que influenciou essa redução do custo de desenvolvimento de sistemas de automação sem fio, pois até então só existiam sistemas que utilizavam protocolos fechados e específicos para uma determinada solução.

## 1.2 Motivação

Apesar da evolução da tecnologia ter impulsionado o mercado de automação residencial, a maioria dos sistemas existentes se limita no controle da residência pela intervenção do usuário, seja pelo celular, internet ou algum tipo de painel de controle. Poucas são as soluções que atuam de forma realmente automática, e quando existem, elas atuam de forma local, por exemplo, um sensor de luminosidade em uma lâmpada específica que ligará quando o nível de luz for baixo.

Além disso, muitas soluções de automação residencial oferecem as possibilidades de configuração e agendamento bastante “engessadas”, em alguns casos obrigando o usuário a acionar a empresa responsável pelo sistema para alterar alguma função programada.

Levando em consideração esse contexto, o presente trabalho propõe desenvolver uma solução que atua de forma automática e inteligente, limitando ao máximo a interferência do usuário.

### **1.3 Objetivo Principal do Trabalho**

O objetivo do trabalho é desenvolver um sistema de automação baseado em sensores sem fio, que tem como base na sua comunicação o protocolo ZigBee. O diferencial desse sistema é a forma automática de sua atuação, pois todas as funções são executadas de acordo com as informações de sensores espalhados pela residência. O único momento de interação direta do usuário com o sistema será no momento de configurar as regras dos sensores e suas ações para serem aplicadas automaticamente.

Para o desenvolvimento dessa solução, é necessário adaptar diferentes tipos de sensores para que eles se comuniquem com uma central de controle via protocolo ZigBee. Nessa central ficará instalado o software de automação, que realizará o processamento de todas as informações e o gerenciamento das funções da residência.

## 2 PRINCIPAIS CONCEITOS RELACIONADOS

### 2.1 O Padrão ZigBee

O ZigBee é um padrão desenvolvido pela ZigBee Alliance, uma associação formada por grandes empresas do ramo de tecnologia, que trabalham em conjunto no desenvolvimento dessa alternativa de comunicação sem fio. Conforme a própria ZigBee Alliance (2008) essa tecnologia é fruto de esforços de empresas com necessidades em comum em torno de um padrão de comunicação simples e robusto.

De acordo com Farahani (2008) o principal alvo do padrão ZigBee é aplicações não muito complexas que necessitam de baixo consumo de energia, baixas taxas de transferências e baixo custo.

O ZigBee utiliza a norma IEEE 802.15.4, homologada em maio de 2003, como camada física e camada de acesso ao meio, onde é proposta uma comunicação de dispositivos sem fio com baixa taxa de transferência em uma PAN (*Personal Area Network*).

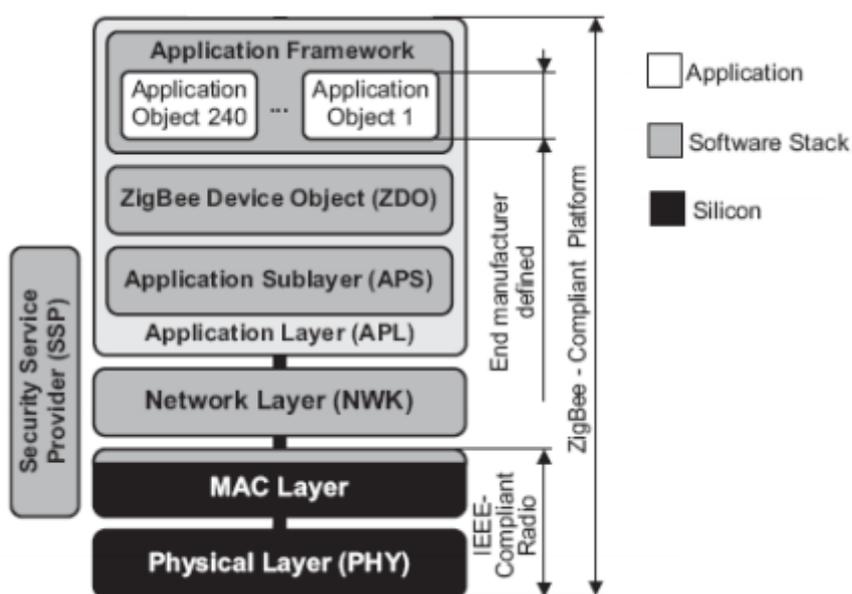


Figura 2.1: Camadas do padrão ZigBee (KAUR, AHUJA, 2011).

### 2.1.1 Camada Física

Segundo o modelo OSI, a camada física, também conhecida como PHY (*Physical Layer*), tem a responsabilidade de tratar a transmissão dos dados brutos por meio de um canal de comunicação. Para isso, é preciso definir uma série de parâmetros, como, por exemplo, modulação e taxa e transferência, necessários para estabelecer esse vínculo.

O padrão IEEE 802.15.4 oferece diferentes faixas de frequência, que são aplicadas de acordo com região. Nos Estados Unidos, a frequência utilizada é de 902–928 MHz, possibilitando uma taxa de 40kbps. Na Europa, usa-se a frequência 868–868.6 MHz, que atinge uma taxa de 20kbps. Já a frequência 2400–2483.5 MHz é usada em todo o mundo, a taxa nesse caso é de 250kbps. A modulação também varia de acordo com a região, conforme a tabela abaixo.

Tabela 2.1: Faixas de frequência do padrão IEEE 802.15.4

Camada Física	Frequência	Canais	Taxa de bit (Kbits/s)	Modulação
868/915 MHz	868-870 MHz	0	20	BPSK
	902-928 MHz	1-10	40	BPSK
2.4 GHz	2.4-2.4835 GHz	11-26	250	O-QPSK

### 2.1.2 Camada MAC

A camada de controle de acesso ao meio (MAC) tem as seguintes responsabilidades: gerenciar o acesso ao canal, sincronização de *beacons* (baliza), validação/reconhecimento do quadro e de associação/desassociação de dispositivos na rede.

Segundo Farahani (2008), existem dois métodos de acesso ao canal: baseado em contenção e livre de contenção.

- No modelo baseado em contenção, o dispositivo começa a transmitir imediatamente se verificar que o meio está livre, utilizando o mecanismo CSMA-CA, que possui, em alguns casos, um tempo de contenção para iniciar a transmissão efetiva dos dados.
- No método livre de contenção, o coordenador da PAN define um intervalo de tempo para cada dispositivo. Esse intervalo, chamado de *guaranteed time slot* (GTS), possibilita ao dispositivo realizar a transmissão sem a necessidade do protocolo CSMA-CA, logo sem necessidade de contenção. Porém, é necessário que todos os dispositivos da rede estejam sincronizados para que esse método funcione. Isso é feito através de um pacote especial e controle, chamado de *beacon*, que é enviado pelo coordenador.

Para o ZigBee, a grande vantagem do modelo livre de contenção, também conhecido como *beaconned*, é a economia de energia, pois o dispositivo pode ficar no estado dormiente (*sleep*) enquanto não estiver no seu determinado *slot*. Já o modo que utiliza CSMA-CA, ou modo *non-beaconned*, é necessário que os dispositivos do tipo roteador estejam sempre ativos, impossibilitando poupar energia com a funcionalidade *sleep*.

### 2.1.3 Camada de Rede

A camada de rede é responsável por duas funções, o transporte de dados e também o suporte as aplicações dos dispositivos ZigBee. Segundo TEIXEIRA, 2006, algumas funcionalidades lhe foram concedidas para desempenhar esse papel:

- Network Scan: Capacidade de um dispositivo de detectar um ou mais canais ativos em sua faixa de comunicação;
- Creating/Joining a PAN: Criar uma rede local e ingressar em uma já existente;
- Device Discovery: Capacidade de encontrar dispositivos sobre o canal ativo na PAN;
- Service Discovery: Descoberta de um serviço e a capacidade de determinar quais os serviços são suportados pelos dispositivos dentro de uma rede;
- Binding: Capacidade de se comunicar no nível da aplicação com outro dispositivo da rede.

### 2.1.4 Camada de Aplicação

A camada de aplicação é a camada de mais alto nível definida pela especificação ZigBee. Ela pode ser dividida em três elementos: subcamada de suporte a aplicação (APS), Application Framework e ZigBee Device Object.

- Subcamada de Suporte a Aplicação (APS): fornece uma interface entre a camada de rede e aplicação. É responsável por gerenciar a tabela de ligação, mantendo uma base de dados com os dispositivos conectados na rede. Também é sua função prover a transmissão de dados (PDUs) entre dois dispositivos na mesma rede. Para isso ele realiza a fragmentação e remontagem das PDUs, além dos serviços de segurança.
- Application Framework: é o ambiente onde são alocados os objetos de aplicação ZigBee (*Application Objects*). Esses objetos possuem funções definidas pelos fabricantes, incluindo primitivas de serviço de dados como *request*, *response*, *confirm*. Cada objeto representa um tipo (ou *profile*) de aplicação diferente, que é definido para um único dispositivo ZigBee, é possível armazenar até 240 objetos de aplicação distintos em um dispositivo.
- ZigBee Device Object (ZDO): implementa o *profile* do dispositivo proporcionando as funcionalidades básicas que permitem a comunicação entre a APS e os objetos de aplicação. Também é responsável por descobrir e identificar os serviços oferecidos por novos dispositivos na rede.

*Application Profiles* são definições de formatos de mensagens e processamento de ações que possibilitam a interoperabilidade entre dispositivos. É obrigatório que qualquer dispositivo na rede ZigBee implemente um *profile*.

Já os *Application Objects* encapsulam um conjunto de atributos que representam o estado do dispositivo e também funcionalidades (serviços) para ler/escrever nesses atributos.

### 2.1.5 Características

O padrão ZigBee possui diversas funcionalidades que foram desenvolvidas para atender as necessidades do mercado, principalmente da área de automação. Algumas das principais características do ZigBee são:

- Baixo consumo de energia: possui a habilidade de operar no modo sleep, podendo reduzir o consumo de energia de 70 mili amperes para 6 micro amperes;
- Baixo custo: possui uma pilha de protocolo de fácil implementação;
- Grande quantidade de nós: é possível ter 65535 nós em uma única rede;
- Diferentes topologias de rede: estrela, árvore ou malha;
- Baixa latência: tempo pequeno de ligação a rede e rapidez na transição do modo de espera (sleep) para o modo ativo (30 milissegundos ou menos);
- Dois modos de operação da rede: beaconing e non-beaconing;
- Segurança e confiabilidade: possui recursos de encriptação com a implementação do padrão AES (*Advanced Encryption Standard*) de 128 bits;

### 2.1.6 Funcionamento

Para entender o funcionamento de uma rede ZigBee, se faz necessário conhecer os três tipos de nós que a rede pode possuir. Os três tipos de nó são: roteador, dispositivo final ou coordenador.

O dispositivo coordenador (*coordinator*) é o agente central da rede, o nó que cria e gerencia uma rede ZigBee. Existe apenas um único coordenador por rede, sendo ele o responsável por concentrar as informações de interesse da aplicação, por isso ele possui uma maior capacidade computacional. O coordenador é quem determina o identificador da rede (PAN ID) e este valor deverá ser utilizado por todos os equipamentos que desejam fazer parte da mesma rede.

Os dispositivos roteadores (*router*) possuem o importante papel de redirecionar os pacotes entre os nós da rede que não conseguem se comunicar diretamente. Isso permite a expansão da rede do ponto de vista físico.

Dispositivos finais (*end devices*) são aqueles que desempenham as funções de sensores ou atuadores da rede. Esses dispositivos podem se comunicar somente com os roteadores e o coordenador.

Uma das principais funcionalidades dos dispositivos finais é a possibilidade de operar no modo *sleep*, o que reduz de forma significativa o consumo de energia elétrica. Esse tipo de dispositivo é projetado para ficar a maior parte do tempo nesse modo, sendo que a transição para o modo ativo é extremamente rápida, cerca de 30 milissegundos.

### 2.1.7 Topologias de rede

De acordo com Farahani (2008), as aplicações que utilizam como base o IEEE 802.15.4 podem suportar as topologias estrela e ponto-a-ponto, sendo que nesta última é possível formar redes mais complexas como árvores e malha.

Na topologia estrela (*star*) diversos dispositivos se comunicam apenas com o coordenador da rede. Nesta topologia os dispositivos finais não conseguem se comunicar entre si, somente por intermédio do coordenador, por esse motivo esta topologia é utilizada geralmente em aplicações de baixa complexidade.

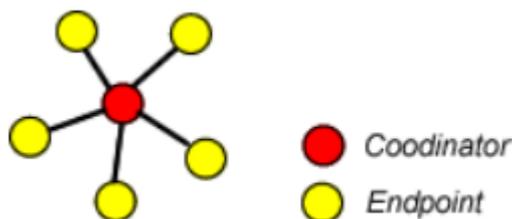


Figura 2.2: Topologia estrela.

Na topologia do tipo ponto-a-ponto, os dispositivos de uma mesma rede podem se comunicar entre si diretamente, sem a necessidade do coordenador. Nesse caso, é possível organizar a rede em duas outras topologias: árvore ou malha.

Nas redes em árvore (*cluster tree*) é utilizada uma estratégia de roteamento hierárquico, onde o coordenador se comunica com os módulos roteadores e esses com os dispositivos finais. Com a utilização dos roteadores é possível expandir a rede geograficamente, porém os roteadores não se comunicam entre si nesse modelo, apenas com o coordenador.

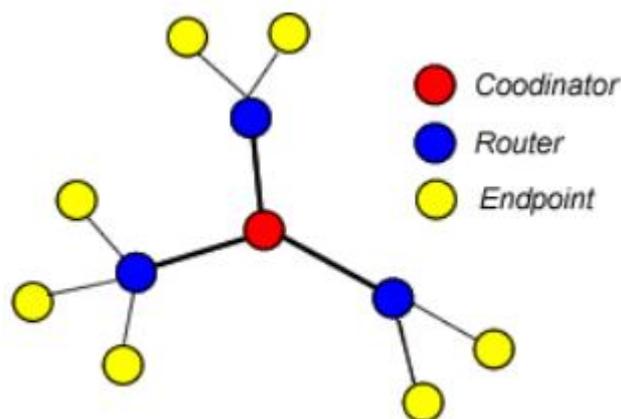


Figura 2.3: Topologia em árvore.

A rede em malha (*mesh*) é considerada uma extensão da topologia em árvore. Nela os roteadores podem se comunicar diretamente, sem a necessidade de passar pelo coordenador. Com isso é possível expandir a rede indefinidamente, apenas acrescentando mais roteadores. Outra vantagem é a redundância na comunicação, caso um roteador perca a conexão, é possível redirecionar as mensagens por uma rota alternativa.

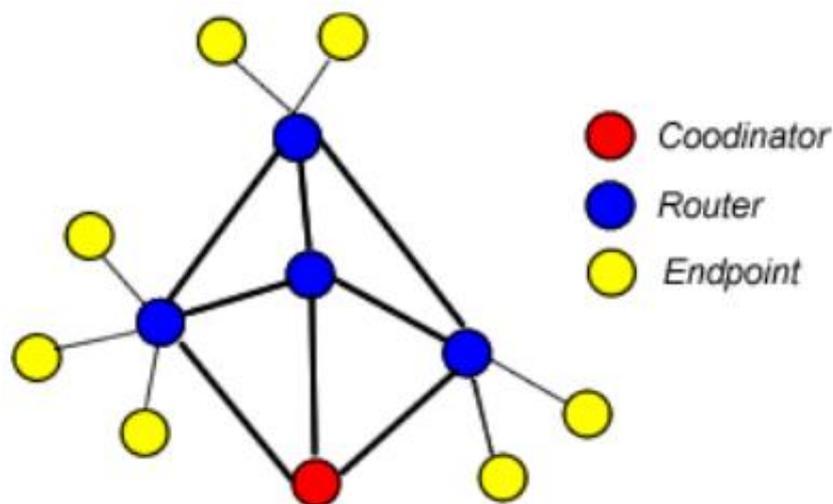


Figura 2.4: Topologia em malha.

## 2.2 Módulo XBee

Para implementação do sistema de automação, foi utilizado um módulo de rádio frequência XBee. Esse dispositivo foi desenvolvido em 2005 pela empresa Digi International, fabricante de dispositivos de comunicação sem fio.

O XBee é um módulo que possui o protocolo ZigBee embarcado no próprio microcontrolador. Esse módulo é disponibilizado em dois modelos, sendo eles XBee e XBee-PRO, onde ambos possuem as mesmas funcionalidades. A diferença entre eles está apenas na potência de transmissão.



Figura 2.5: Módulo XBee-PRO.

Cada módulo possui diversos pinos de I/O que podem ser configurados para atuar como leitores ou acionadores (*input* ou *output* respectivamente). Esses pinos são o meio com que o módulo se comunica com o ambiente externo, no caso desse projeto, os sensores e os dispositivos atuadores.

Além de poder operar em diferentes topologias, o módulo também implementa a funcionalidade de *sleep mode*, o que permite que ele seja alimentado por pilhas ou baterias, obtendo uma autonomia de meses ou até anos.

O formato dos dados enviados na comunicação entre os XBees é especificado por uma API (*Application Program Interface*), o que facilita o gerenciamento dos módulos por outros programas. Cada mensagem trafegada na rede ZigBee é estruturada em um *frame*, contendo informações pertinentes para uma comunicação efetiva na rede.

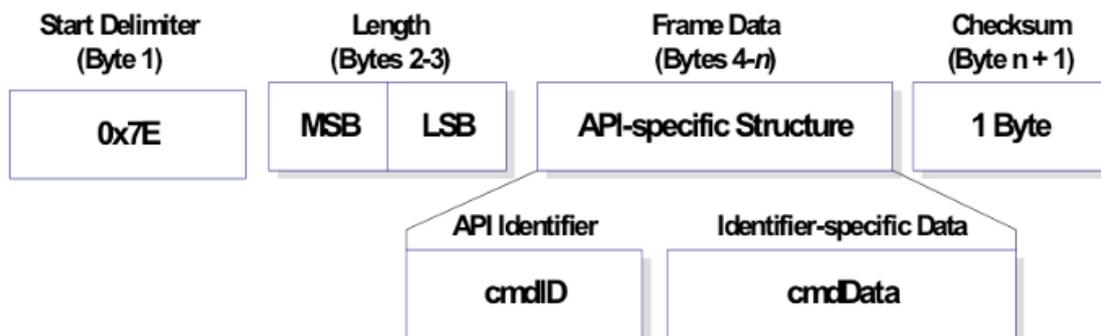


Figura 2.6: Estrutura do frame XBee.

De acordo com a figura 2.6, o conjunto de dados enviados ou recebidos é composto pelos seguintes campos:

- *Start Delimiter*: byte indicador do início do *frame*;
- *Length*: dois bytes que informam o tamanho do campo "*Frame Data*";
- *Frame Data*: estrutura composta por um identificador do tipo de mensagem e pelo seu conteúdo;
- *API Identifier*: define o tipo de comando (*AT*, *Remote Request*, *Remote Response*, entre outros);
- *Identifier-specific Data*: conteúdo do *frame* com a estrutura interna de dados que varia de acordo com o tipo de comando que é utilizado;
- *Checksum*: byte que valida a integridade do frame.

De acordo com o manual disponibilizado pela Digi International, existem dois tipos de mensagens que podem ser enviadas. Cada tipo possui diferentes comandos que podem ser usados para interagir com os módulos.

O *AT Command* é utilizado quando se deseja enviar uma mensagem diretamente para o coordenador da rede, ou seja, é uma mensagem local, que não será retransmitida para outro nó na rede. Como esse módulo XBee estará conectado fisicamente no dispositivo onde o programa usuário está em execução, não é necessário especificar o endereço do módulo de destino (sempre será o próprio coordenador), tornando o *frame* dessa mensagem bastante simples.

Para enviar um comando para um módulo remotamente, é necessário utilizar a mensagem do tipo *Remote Command Request*. Nela é preciso informar o endereço do

módulo de destino podendo ser inclusive um endereço de *broadcast*, enviando o comando para todos os módulos da rede. As mensagens serão sempre enviadas ao coordenador, que por sua vez é o encarregado de retransmitir até o módulo de destino.

Os XBees podem ser configurados para atuar como roteadores, formando assim uma topologia em malha (*mesh*). Assim é possível expandir a rede ZigBee de acordo com a necessidade, como foi descrito anteriormente.

O próprio *firmware* do XBee realiza os tratamentos necessários para que as mensagens sejam transmitidas entre os módulos com sucesso, de acordo com as regras especificadas pelo padrão ZigBee. Caso o coordenador não consiga enviar a mensagem até o módulo de destino, então ele enviará uma mensagem sinalizando que não foi possível executar o comando desejado. Essa funcionalidade é bastante útil para a questão de confiabilidade de um sistema, onde é possível que o programa usuário trate os casos onde ocorreram erros na transmissão de comandos.

Toda a comunicação necessária entre os módulos para o controle dos dispositivos e o gerenciamento dos sensores é feita através dos comandos disponibilizados pela API do XBee. A tabela a seguir apresenta os comandos mais importantes que foram utilizados nesse projeto.

Tabela 2.2: Principais comandos XBee

Comando	Nome	Descrição
ND	Node Discover	Busca todos os módulos presentes na rede.
ID	PAN ID	Busca ou configura o identificador da rede.
D0	AD0 / DIO0 Configuration	Busca ou configura as funções do pino 0. Pode ser configurado como entrada analógica, entrada digital, saída em baixa ou saída em alta.
IS	Force Sample	Força a leitura ( <i>sample</i> ) de todos os pinos habilitados do módulo.
SM	Sleep Mode	Busca ou configura o <i>sleep mode</i> .

A lista completa de comandos pode ser encontrada no manual do produto disponibilizado pela Digi International, informado nas referências deste trabalho.

## **3 IMPLEMENTAÇÃO DO PROJETO**

O sistema de automação foi desenvolvido para possibilitar a consolidação dos dados adquiridos dos sensores sem fio e para coordenar as ações que devem ser realizadas com base nessas informações. Dessa forma, o controle de dispositivos da residência como luzes, persianas, aparelhos eletrônicos, entre outros, ocorrerá de forma automática, sem a necessidade de intervenção do usuário.

A troca de dados entre os sensores, a central de comandos e os dispositivos que serão acionados é feita através dos módulos de rádio frequência XBee, que implementam o protocolo de comunicação ZigBee, introduzido anteriormente.

### **3.1 Funcionamento Geral do Sistema**

A solução de automação residencial desenvolvida nesse projeto é composta por um conjunto de componentes de hardware e software que interagem por meio de uma rede sem fio, utilizando o protocolo ZigBee.

Os sensores são responsáveis por informar à central de controle sobre qualquer mudança de estado, como alteração na temperatura ou acionamento de um sensor de movimento, por exemplo. Já os dispositivos como luzes e persianas são elementos passivos, pois são acionados por comandos vindos da central.

Quem realiza comunicação entre os sensores e os dispositivos é o componente denominado de central de controle. Essa central é onde ficará instalado o software de automação, responsável por processar os dados adquiridos dos sensores e aplicar as ações nos atuadores, caso seja necessário.

A decisão de acionar ou não os dispositivos é feita com base em regras definidas pelo usuário. Sendo assim, a única intervenção do usuário no funcionamento do sistema é no momento de definir as regras para os sensores e quais dispositivos devem ser acionados caso aconteça a situação definida nas regras.



Figura 3.1: Visão geral do funcionamento do sistema.

## 3.2 Sensores e Dispositivos ZigBee

Os sensores utilizados no sistema são uma adaptação de sensores comerciais com a inclusão dos módulos XBee. Algumas funcionalidades disponibilizadas pela API de programação do XBee são usadas para tornar possível essa união do módulo com o sensor.

Como descrito anteriormente, esses módulos possuem pinos de I/O para se comunicarem com o meio externo. Esses pinos são a ligação entre os sensores/dispositivos e o XBee, onde os pinos que conectam os sensores serão do tipo *input* e os pinos dos dispositivos do tipo *output*.

O desenvolvimento dos sensores e a adaptação com o XBee foi feita com o auxílio do engenheiro Bernardo Pianta, formado em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS).

### 3.2.1 Sensor de Movimento

Para implementar esse sensor, foi utilizado o sensor de movimento comercial da marca Fairchild e nele foi adaptado um módulo XBee para possibilitar a interação com a central de controle.

Esse sensor funciona pela detecção da radiação infravermelha, ou seja, ele é acionado quando existe uma variação das ondas eletromagnéticas emitidas por fontes de calor que estão dentro da área de alcance do sensor.

Para adaptar o módulo XBee, foi necessário conectar um dos pinos de I/O na saída do sensor que indica se houve movimento (*output* 1 do amplificador operacional da placa do sensor). Também foi necessário conectar o pino de terra do XBee no sensor, pois ambos precisam ter o terra comum.

Como o XBee e o sensor de movimento trabalham em tensões diferentes, fez-se necessário que o XBee ter a sua própria fonte de alimentação de 3,3 VCC, sem poder aproveitar a alimentação do sensor.

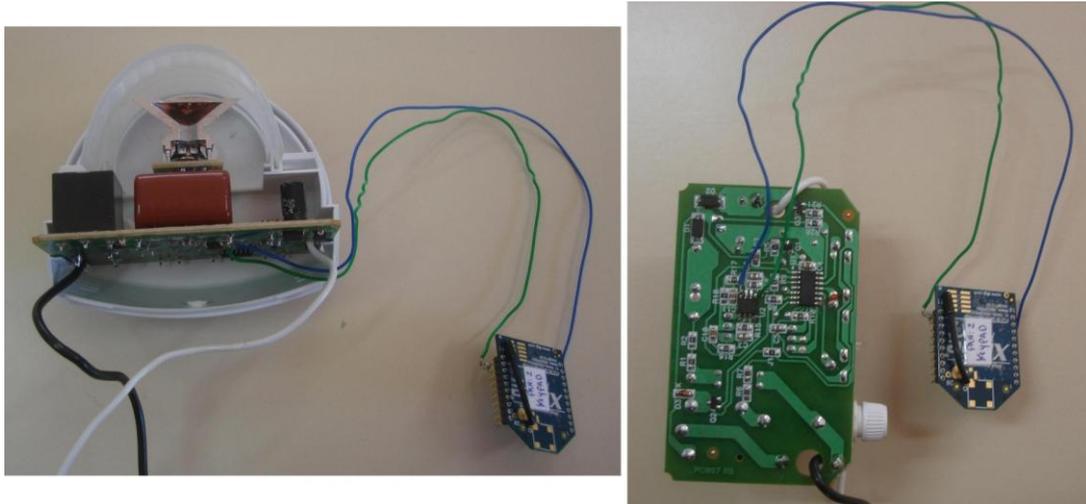


Figura 3.2: Sensor de movimento com o módulo XBee.

### 3.2.2 Sensor de Chuva

Para o sensor de chuva, não foi utilizado nenhum sensor comercial com a finalidade de detecção da chuva. Ao invés disso, foi desenvolvida uma placa com algumas trilhas para fazer a condução da corrente elétrica em caso de contato com a água da chuva.

O funcionamento desse sensor é bastante simples: as trilhas na placa foram projetadas de forma que, ao entrar em contato com a água, ocorrerá o contato elétrico e acionará o pino do XBee. Além dessa conexão da placa nos pinos do XBee (pino DIO 0 e terra), também foi necessário somente alimentar o módulo na tensão 3,3 VCC.

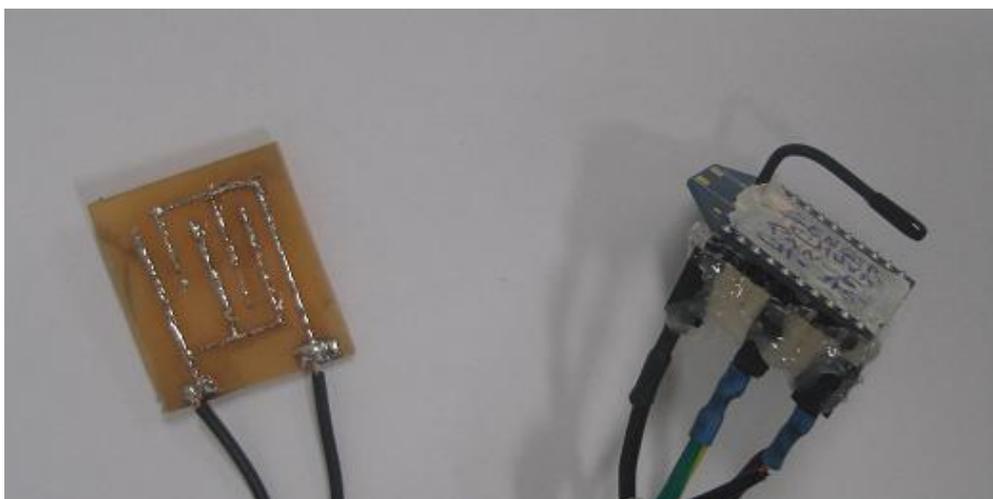


Figura 3.3: Sensor de chuva junto ao módulo XBee.

### 3.2.3 Sensor de Temperatura

Para implementar o sensor de temperatura ZigBee, foi utilizado o componente LM34. Esse circuito é um sensor que tem como saída uma tensão proporcional à temperatura ambiente na escala Fahrenheit.

A saída desse sensor é ligada diretamente no pino DIO 0 do XBee, pois esse é um dos pinos que pode ser configurado como uma entrada analógica, possibilitando que o valor da temperatura seja lido desse pino.

Como esse componente suporta a tensão que o XBee trabalha, não existe a necessidade do componente ter sua própria fonte de energia. O sensor pode ser conectado nos pinos de alimentação do XBee, que por sua vez é alimentado por uma fonte de 3,3 VCC.

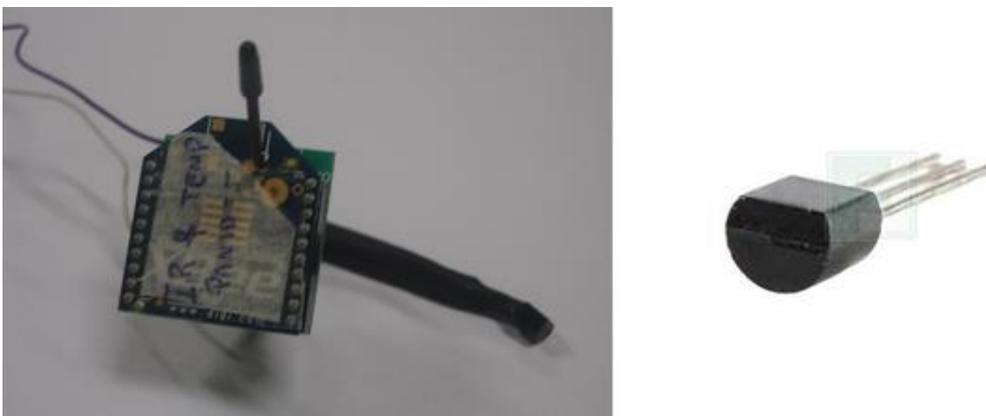


Figura 3.4: Componente LM34 conectado ao módulo XBee.

### 3.2.4 Dispositivos

O módulo XBee pode controlar qualquer tipo de dispositivo que esteja conectado em seus pinos de I/O, como, por exemplo, luzes, cortinas e persianas automatizadas, motores, tomadas, entre outros.

Como o foco do presente trabalho é a interação entre os sensores, não será implementado um circuito para acionar esses diferentes dispositivos. Ao invés disso, será utilizado o próprio módulo XBee alimentado por uma fonte de energia 3,3 VCC e com um componente LED conectado diretamente nos pinos, para detectar quando o dispositivo está ligado ou desligado.

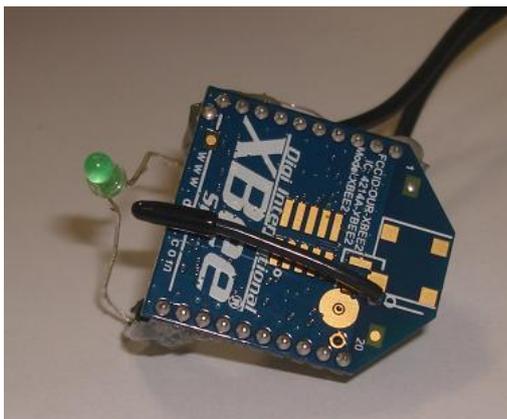


Figura 3.5: Protótipo do dispositivo acionador.

### 3.3 Arquitetura do Sistema

O software de automação desenvolvido ficará instalado no componente denominado de central de controle. Essa central é formada por um computador conectado a um módulo XBee, configurado como sendo o coordenador da rede ZigBee.

O computador poderá rodar qualquer sistema operacional (nesse projeto foi utilizado o Ubuntu 10.10), pois o software desenvolvido utiliza a linguagem JAVA, que é independente de plataforma.

Para armazenar os dados dos dispositivos e sensores cadastrados, assim como as regras de acionamento definidas pelo usuário, foi utilizado o banco de dados MySQL mantido pela empresa Oracle.

A comunicação física entre a central e o XBee é feita por meio de um adaptador denominado XBee Dongle, que conecta o módulo XBee com a central por uma porta USB. Na parte de software, foi utilizado a biblioteca XBee-API versão 0.9 para auxiliar a montagem dos pacotes enviados à rede ZigBee, essa biblioteca também foi desenvolvida na linguagem JAVA.



Figura 3.6: XBee Dongle.

O usuário acessa o sistema por meio de uma interface WEB, implementada com a tecnologia JSP (*Java Server Pages*). Isso permite que o sistema esteja disponível pela da internet, funcionalidade muito importante em um sistema desse tipo.

### 3.4 Sistema de Automação

O software desenvolvido fica instalado na central de controle, sendo responsável por coordenar todos os componentes que estão na rede ZigBee. O sistema possui uma interface WEB bastante simples e intuitiva, oferecendo as funcionalidades de cadastrar os sensores e dispositivos e também possibilita a configuração das regras de acionamento dos dispositivos.

#### 3.4.1 Configuração dos Módulos XBee

Cada módulo XBee deve ser configurado com um série de parâmetros antes de ser utilizado no sistema de automação. Para executar essa configuração foi utilizado o programa X-CTU da própria empresa Digi International. Para configurar cada módulo, ele deve ser conectado no adaptador XBee Dongle que se comunica com o computador por uma porta USB, da mesma maneira que é feita com o módulo coordenador.

Para todos os módulos, tanto sensores quanto dispositivos atuadores, eles devem possuir o mesmo PAN ID. Esse parâmetro é um valor de 4 bytes que representa o identificador da rede, por isso todos os módulos que pretendem se comunicar, devem obrigatoriamente ter o mesmo valor de PAN ID.

Outra configuração comum a todos os módulos é o modo de operação API. Com esse parâmetro, os dados são transmitidos com base em quadros (frames). Além disso, a biblioteca Java XBee-API que é usada nesse projeto só é compatível com os XBees configurados desse modo.

O módulo XBee que ficará conectado junto a central de controle deve ser do tipo *Coordenador*, sendo ele o responsável por criar a rede ZigBee. Já os módulos de sensores e dispositivos devem ser configurados como sendo *End Devices*. Para permitir a ampliação da rede ZigBee, poderia ser utilizado módulos configurados como *Routers*, porém eles não serão utilizados nesse projeto.

Nos módulos acoplados nos sensores de movimento e de chuva, eles devem ser configurados com o pino DIO 0 para operar como uma entrada (*input*). Além disso, é importante que se habilite a opção de *Digital IO Change Detection* nesse pino. Com essa opção, o módulo enviará uma mensagem (*sample*) ao coordenador sempre que ocorrer uma mudança de estado no pino. Assim será possível detectar imediatamente sempre que ocorrer um evento nesses sensores (chuva ou movimento).

Para o sensor de temperatura é necessário configurar o pino DIO 0 como uma entrada analógica (*analog input*). Assim a temperatura poderá ser lida do pino como um valor analógico que representará a temperatura na escala Fahrenheit.

Nos dispositivos acionados, a principal configuração é colocar o pino DIO 0 como uma saída em baixa (*output low*), com isso o estado inicial do LED conectado ao XBee será sempre desligado. Na execução do sistema de automação, o LED será ligado por um comando remoto que colocará esse pino no estado *output high*.

#### 3.4.2 Cadastro dos Componentes

O cadastro dos sensores e dos dispositivos a serem acionados deve ser feito por um usuário com a função de administrador do sistema.

Pela interface WEB o administrador poderá adicionar ou remover os dispositivos controlados pelo sistema, informando o endereço de cada módulo dentro da rede

ZigBee. É necessário informar um nome para o dispositivo e também qual o pino de I/O que será utilizado, pois em um mesmo módulo XBee poderemos ter diversos dispositivos, limitados pelo número de pinos de I/O. Porém nesse trabalho cada módulo estará relacionado a apenas um dispositivo atuador, que é cadastrado pelo tipo “Liga e Desliga” no software de automação.

De forma similar, o administrador poderá cadastrar os sensores ZigBee que ficarão instalados na residência. Nesse caso, além do nome e endereço do módulo XBee, será necessário informar também qual o tipo de sensor, que poderá ser de movimento, chuva ou temperatura. Nesse caso o pino será preenchido automaticamente de acordo com o sensor selecionado.



O formulário apresenta os seguintes campos e elementos:

- ID:
- Nome do Dispositivo:
- Endereço:
- Pino:
- Status:
- Tipo:  (menu suspenso aberto com opções: Seleccione..., Liga e Desliga, Sensor Movimento, Sensor Temperatura, Sensor Chuva)
- Botão Criar (azul)
- Botão Cancelar (azul)

Figura 3.7: Cadastro dos sensores e dispositivos.

### 3.4.3 Configuração do Cenário

A configuração do cenário por sensores é uma das funcionalidades mais importantes do sistema. Ela é a única parte do sistema que o usuário final poderá interagir diretamente pela interface.

Nessa tela, o usuário poderá adicionar elementos em duas listas diferentes, uma de regras e outra de dispositivos que serão acionados. Além disso, também é preciso informar um nome para esse cenário, conforme a figura a seguir.

Cenário por Sensores

Nome:

Regras de Ativação

Adicionar
Remover

Tipo/Dispositivo	Regra
Sensor Temperatura	Acima de 20°C

Dispositivos Acionados

Adicionar
Remover

Dispositivo	Ação
Dispositivo 1	Ligar

Voltar
Salvar

Figura 3.8: Configuração do cenário por sensor.

É necessário definir uma ou mais regras de ativação como sendo os critérios para a execução desse cenário. Existem diferentes tipos de regras que podem ser aplicadas, sendo elas:

- Sensor de movimento foi acionado
- Temperatura maior/menor que uma temperatura definida
- Sensor de chuva com/sem precipitação
- Horário atual do dia maior/menor que um horário definido

Após definir a lista de regras, o usuário deverá informar também a lista de dispositivos que serão acionados. Será possível escolher a opção de ligar ou desligar qualquer dispositivo que esteja conectado em um pino de I/O de um módulo XBee, formando assim um cenário que será executado de acordo com as regras definidas anteriormente.

Figura 3.9: Adição de regras e de dispositivos.

#### 3.4.4 Sensores

Os sensores utilizados no projeto são adaptados com um módulo XBee para poderem interagir com a central de controle. Dependendo do tipo de sensor, ele irá se comunicar de forma diferente com o XBee e a central.

Os sensores de movimento e de chuva utilizam a funcionalidade *Digital IO Change Detection*. Assim o coordenador recebe a mensagem (*sample*) no exato momento em que ocorre um evento no sensor (movimento ou chuva).

Já para o sensor de temperatura, a técnica de comunicação é outra. Nesse caso, a central de controle envia um comando requisitando as informações necessárias do sensor. O comando XBee utilizado é o “IS”, com isso é forçado um *sample* no nodo desejado e as informações podem ser adquiridas com a mensagem de resposta. Essa requisição é enviada a cada minuto e o resultado é usado para verificar as regras de acionamento cadastradas e, caso seja necessário, executar as ações definidas nos cenários.

#### 3.4.5 Acionamento

A verificação das regras de acionamento ocorre em dois momentos: quando uma mensagem de algum dos sensores é recebida (*sample*) ou a cada minuto, quando é feita a leitura dos sensores de temperatura. Nesses dois casos, as regras de todos os cenários são analisadas, serão executados os cenários que possuem todas suas regras em um estado verdadeiro.

Na execução de um cenário, uma mensagem é enviada a cada um de seus dispositivos cadastrados. Esses dispositivos serão ligados ou desligados, de acordo com a configuração feita pelo usuário.

Nos dispositivos desenvolvidos nesse trabalho, foi utilizado o pino DIO 0 (*digital input/output 0*), o valor desse pino é alterado para alto ou baixo pelo comando “D0” da API de programação do XBee, possibilitando ligar ou desligar o LED do circuito.

O Apêndice 1 contém maiores informações sobre o processo de configuração e acionamento dos dispositivos, detalhando todos os casos de uso do sistema.

### 3.4.6 Estrutura do Software

O software de automação foi desenvolvido em dois módulos: módulo WEB e módulo de controle.

- **Módulo WEB:** é responsável por gerar a interface para o usuário (telas de cadastro de cenários, regras e ações). Todos os dados informados pelo usuário são armazenados em um banco de dados (MySQL) que será interpretado pelo módulo de controle.
- **Módulo de Controle:** é responsável por interagir com a rede ZigBee e executar os cenários contidos no banco de dados.

No módulo de controle, existem duas *threads* executando em paralelo que coordenam a validação e execução de todos os cenários.

Uma das *threads* realiza a leitura dos sensores de temperatura a cada minuto, mandando uma mensagem que força um *sample* dos módulos XBee. Além disso, após enviar o comando aos sensores de temperatura, essa *thread* faz a leitura de todos os cenários cadastrados e a execução dos cenários que forem necessários. Essa verificação a cada minuto também é necessária para que sejam executados os cenários que possuem o horário como uma das regras.

A outra *thread* faz a leitura das mensagens que chegam ao coordenador, o tratamento de cada mensagem depende do seu tipo:

- *Sample* do sensor de temperatura: é feita a leitura do pacote e o valor da temperatura é atualizado no banco de dados. Nesse caso a verificação dos cenários é feita pela *thread* que envia a mensagem aos módulos de temperatura.
- *Sample* do sensor de movimento ou chuva: faz a verificação de todos os cenários e executa os cenários que forem necessários. O método de verificação/execução é o mesmo utilizado pela *thread* descrita anteriormente.
- Resposta OK de um dispositivo atuador: atualiza no banco de dados o estado desse dispositivo (ligado ou desligado). Isso é feito na resposta ao invés do envio para garantir que o XBee está funcionando corretamente, evitando que o estado do dispositivo seja alterado indevidamente (se o XBee estivesse indisponível, por exemplo).

Qualquer acionamento de dispositivo, independente da *thread* que executa, utiliza uma estrutura de dados para indicar qual o comando foi enviado para o módulo (ação de liga ou desliga), assim é possível atualizar o banco de dados com o valor correto do dispositivo. As duas *threads* do sistema tem a visibilidade dessa variável. Além disso, o próprio tipo da variável (ConcurrentHashMap do Java) possui o controle de concorrência na sua implementação, evitando problemas de acesso pelas duas *threads* simultaneamente.

A figura a seguir resume o funcionamento das duas *threads* do módulo de controle, onde a Thread 1 é a que realiza a leitura dos sensores de temperatura e a Thread 2 a leitura das mensagens do módulo coordenador.

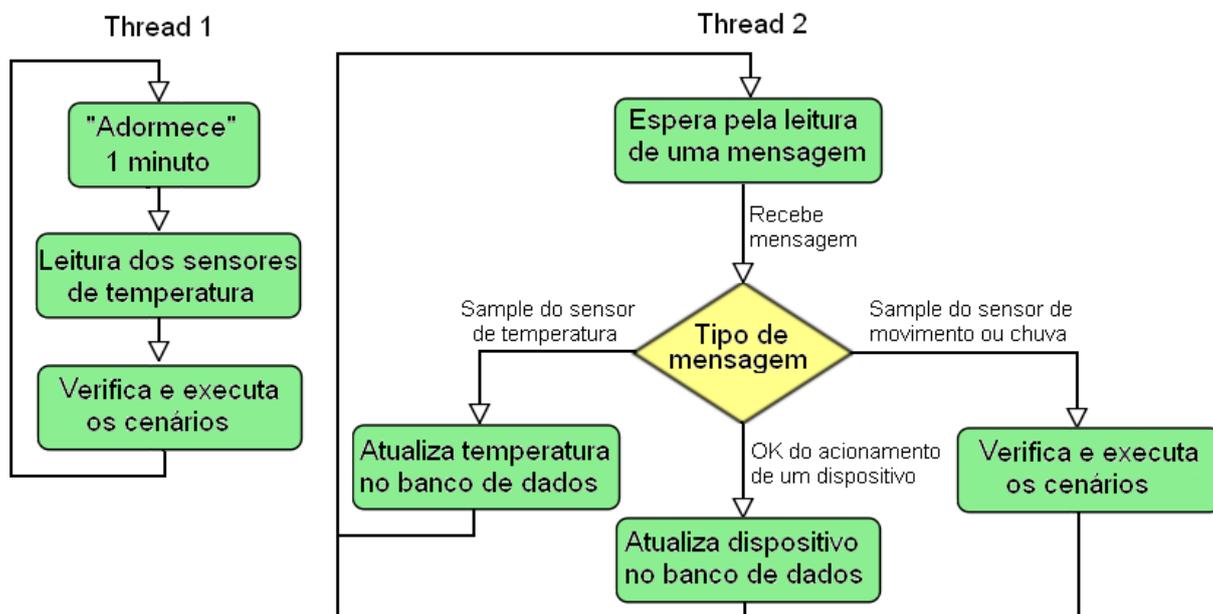


Figura 3.10: Threads do módulo de controle.

## 4 CONCLUSÕES E TRABALHOS FUTUROS

### 4.1 Conclusões

Com a popularização e o desenvolvimento de novas tecnologias sem fio, o mercado de automação residencial está crescendo significativamente, inclusive no Brasil. Esse trabalho propôs uma alternativa de sistema que utiliza apenas sensores *wireless* para automatizar as tarefas rotineiras em uma residência.

O software desenvolvido será incluído no sistema de automação residencial da empresa Home Manager, criada pelo autor desse trabalho em conjunto com o sócio Bernardo Pianta. Atualmente a empresa encontra-se incubada na RAIAR, incubadora de empresas da PUCRS, de Viamão.

A utilização do protocolo ZigBee na implementação desse trabalho agrega várias vantagens para o sistema de automação, como a economia de energia, baixo custo e grande facilidade na instalação do sistema, evitando reformas na residência. Além disso, a capacidade do ZigBee de formar uma rede *mesh* garante uma ampla cobertura de sinal e também agrega confiabilidade na comunicação entre os nodos da rede.

O modelo de configuração do sistema definido por uma lista de regras e outra lista de ações se mostrou bastante flexível, assim é sempre possível criar os cenários de acordo com a necessidade do usuário.

Como esse modelo foi desenvolvido para ser o mais genérico possível, visando à abrangência do maior número de casos, em algumas ocasiões a configuração não é tão intuitiva quanto se desejava. Em alguns casos é necessária a criação de mais de um cenário para o mesmo propósito, um para ligar o dispositivo e outro para desligar após um período de tempo, por exemplo.

### 4.2 Trabalhos Futuros

O presente trabalho possibilita a expansão do sistema desenvolvido com a inclusão de um maior número de sensores ZigBee e também o desenvolvimento dos dispositivos que podem ser controlados. Com isso é possível criar um sistema completo de automação, englobando a parte de controle pelo usuário e também às ações automáticas baseadas em sensores, desenvolvidas nesse trabalho.

Outra melhoria muito interessante nesse projeto seria a utilização de inteligência artificial no sistema de automação. Nessa abordagem o sistema poderia “aprender” a

rotina do usuário em sua residência, baseada nos sensores, possibilitando se antecipar as ações que seriam feitas manualmente. Por exemplo, próximo ao horário do meio dia, logo após o sensor de movimento da sala ser acionado, a televisão é ligada pelo usuário. Um sistema que implemente o mecanismo de aprendizado poderia detectar esse padrão e ligar a televisão assim que o sensor de movimento for acionado nesse intervalo de horário. Com isso não seria mais necessária a configuração das regras pelo usuário, o próprio sistema poderia tentar defini-las de acordo com a rotina dos moradores da residência.

Atualmente não existe nenhum sistema comercial que aplique essa proposta de aprendizado com inteligência artificial, em alguns casos essa técnica é aplicada em projetos experimentais na área de automação. Mesmo assim, existem diversas pesquisas e projetos nesse assunto, buscando sempre uma forma ideal de automatizar uma residência sem produzir regras e ações indesejadas para o usuário.

## REFERÊNCIAS

FARAHANI, Shahin. **ZigBee Wireless Network and Transceivers**. Burlington: Newnes, 2008.

LABIOD, Houda; AFIFI, Hossan; DE SANTIS, Costantino. **Wi-fi, Bluetooth, ZigBee and WiMax**. Paris: Springer, 2007.

KAUR, Gurjit; AHUJA, Kiran. **QoS measurement of Zigbee home automation network using various modulation schemes**. International Journal of Engineering Science and Technology (IJEST), 2011.

William C Craig. ZigBee: “Wireless Control That Simply Works”. <http://www.zigbee.org>

TEIXEIRA, Luis Marcelo. **Desenvolvimento de uma Aplicação com o Protocolo ZigBee aplicado em Instrumentação de Ensaio de Vôo**. 2006. 162f. Tese de Mestrado em Engenharia Eletrônica e Computação – Instituto Tecnológico de Aeronáutica, São José dos Campos.

ZigBee Alliance, “ZigBee Specifications”, version 1.0 r13, December 2006. <http://www.zigbee.org/>

IEEE Computer Society. **IEEE Standard 802.15.4**: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Computer Society. [S.l.]: Disponível em: <<http://standards.ieee.org/getieee802/download/802.15.4a-2007.pdf>>. Acesso em: 21 abr. 2011.

XBee™ / XBee-Pro™ ZB RF Modules. **Manual do Produto**. [S.l.]: Disponível em: <[http://ftp1.digi.com/support/documentation/90000982\\_A.pdf](http://ftp1.digi.com/support/documentation/90000982_A.pdf)>. Acesso em: 14 set. 2011.

## APÊNDICE 1 – DIAGRAMA DE CASOS DE USO

Com o intuito de facilitar o entendimento da solução utilizada, foi desenvolvido o diagrama e a descrição dos casos de uso do sistema.

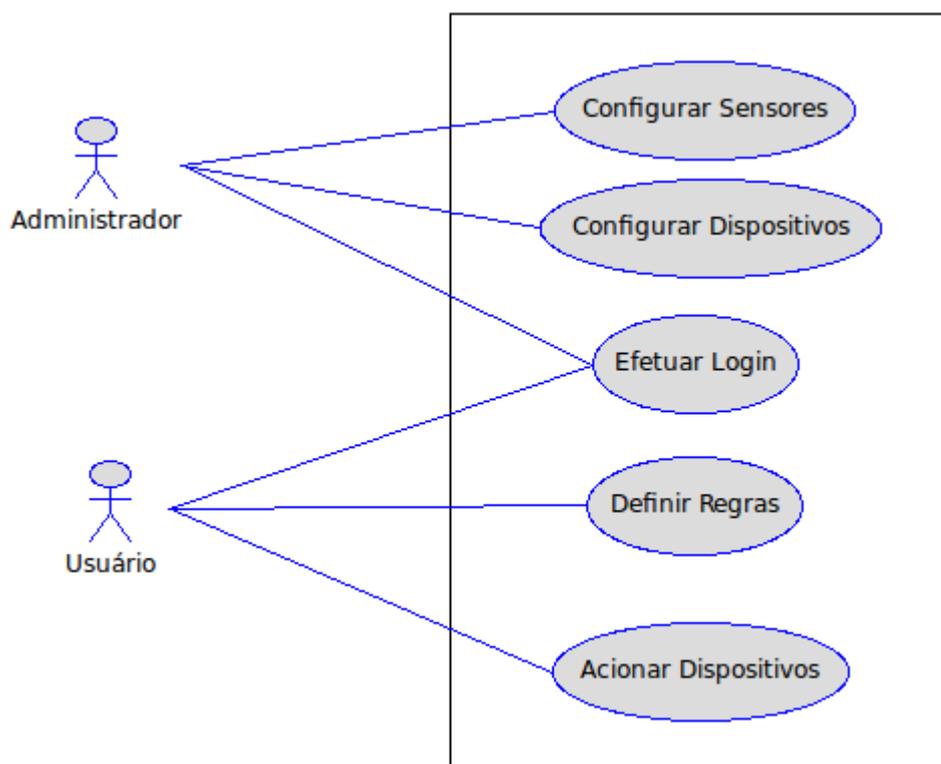


Figura A1.1: Diagrama de casos de uso do sistema.

<b>Caso de Uso:</b>	Efetuar Login
<b>Atores:</b>	Usuário
<b>Pré-condições:</b>	Nenhuma
<b>Descrição:</b>	Processo que autentica o usuário para entrar no sistema
Sequência de Eventos	
Ator	Sistema
1. Entra com o login e senha	2. Valida usuário
	3. Libera o acesso ao sistema de acordo com a permissão definida para o usuário

<b>Caso de Uso:</b>	Configurar Sensores
<b>Atores:</b>	Administrador
<b>Pré-condições:</b>	Ter feito o login com um usuário com a permissão de administrador
<b>Descrição:</b>	Permite gerenciar o cadastro de sensores do sistema
Sequência de Eventos	
Ator	Sistema
1. Acessa a área de cadastro dos sensores	2. Exibe os sensores cadastrados
3. Seleciona um sensor para edição ou seleciona a criação de um novo sensor	4. Exibe tela de cadastro
5. Insere as informações referentes ao sensor	6. Salva as informações no banco de dados

<b>Caso de Uso:</b>	Configurar Dispositivos
<b>Atores:</b>	Administrador
<b>Pré-condições:</b>	Ter feito o login com um usuário com a permissão de administrador
<b>Descrição:</b>	Permite gerenciar o cadastro de dispositivos do sistema
Sequência de Eventos	
Ator	Sistema
1. Acessa a área de cadastro dos dispositivos	2. Exibe os dispositivos cadastrados
3. Seleciona um dispositivo para edição ou seleciona a criação de um novo dispositivo	4. Exibe tela de cadastro
5. Insere as informações referentes ao dispositivo	6. Salva as informações no banco de dados

<b>Caso de Uso:</b>	Definir Regras
<b>Atores:</b>	Usuário
<b>Pré-condições:</b>	Ter feito o login com a permissão de usuário
<b>Descrição:</b>	Permite gerenciar o cadastro de regras de sensores e as ações dos dispositivos
Sequência de Eventos	
Ator	Sistema
1. Acessa a área de configuração de cenários por sensores	2. Exibe todos cenários cadastrados
3. Seleciona um cenário para edição ou seleciona a criação de um novo cenário	4. Exibe a tela de cadastro do cenário, contendo a lista de regras e ações dos dispositivos
5. Adiciona ou remove regras de sensores na lista de regras	6. Atualiza a lista de regras na tela de cadastro
7. Adiciona ou remove ações para os dispositivos na lista de ações	8. Atualiza a lista de ações na tela de cadastro
9. Confirma a alteração ou criação do cenário	10. Salva as informações no banco de dados

<b>Caso de Uso:</b>	Acionar Dispositivos
<b>Atores:</b>	Usuário
<b>Pré-condições:</b>	Ter configurado no sistema o sensor que será disparado
<b>Descrição:</b>	Aciona os dispositivos configurados de acordo com a informação dos sensores
<b>Sequência de Eventos</b>	
<b>Ator</b>	<b>Sistema</b>
1. Dispara algum dos sensores da residência	2. Busca todos os cenários que possuem como regra o sensor disparado
	3. Para cada cenário, valida se todas as regras do cenário são verdadeiras
	4. Aciona os dispositivos dos cenários que possuem todas as regras verdadeiras

## APÊNDICE 2 – DIAGRAMA ER

Este apêndice possui o diagrama ER (Entidade e Relacionamento) do sistema de automação. O diagrama representa o modelo de dados do sistema e foi desenvolvido utilizando a ferramenta MySQL Workbench da Oracle.

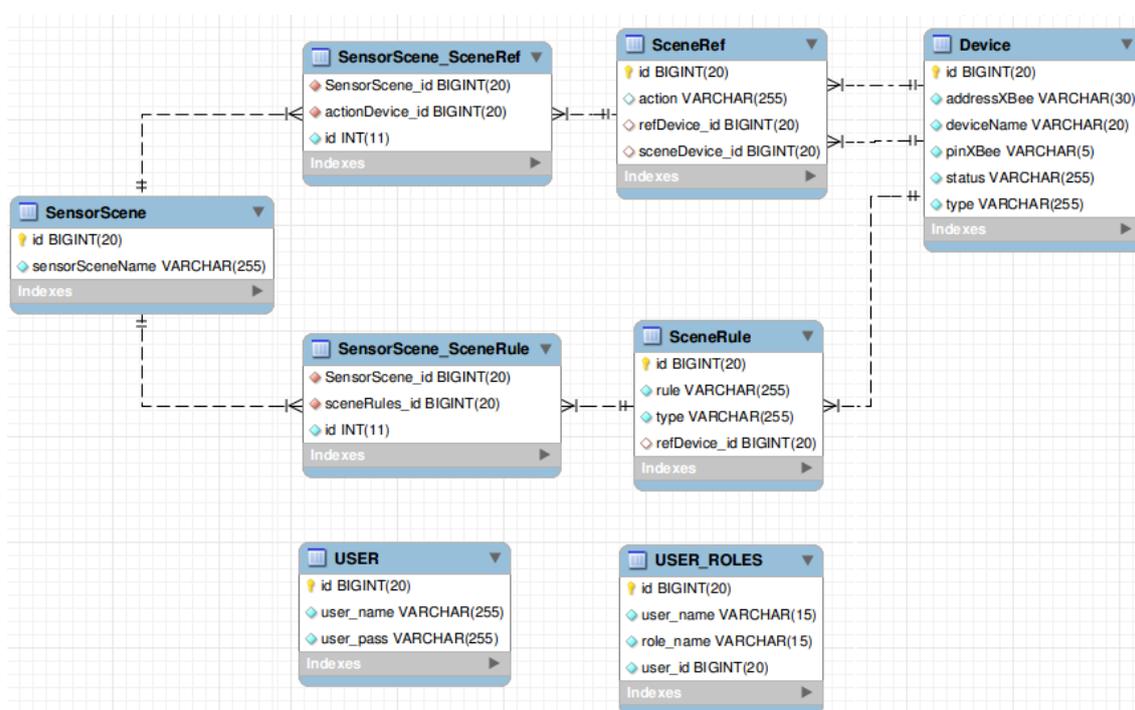


Figura A2.1: Diagrama ER do sistema.