



UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
TECNÓLOGO EM REDES DE COMPUTADORES

**MATHEUS MEDEIROS DA CRUZ**

**CLIMADUINO: SISTEMA EMBARCADO DE CLIMATIZAÇÃO PARA  
EDIFÍCIOS INTELIGENTES**

**QUIXADÁ  
2013**

**MATHEUS MEDEIROS DA CRUZ**

**CLIMADUINO: SISTEMA EMBARCADO DE CLIMATIZAÇÃO PARA  
EDIFÍCIOS INTELIGENTES**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Tecnólogo em Redes de Computadores da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Tecnólogo.

Área de concentração: computação

Orientadora Prof<sup>a</sup>. Atslands Rego da Rocha

**QUIXADÁ  
2013**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

---

C963c Cruz, Matheus Medeiros da  
Climaduino: sistema embarcado de climatização para edifício inteligente / Matheus Medeiros da Cruz. – 2013.  
66 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso Superior de Tecnologia em Redes de Computadores, Quixadá, 2013.  
Orientação: Profa. Dra. Atslands Rego da Rocha  
Área de concentração: Computação

1. Sistemas de comunicação sem fio 2. Edifícios inteligentes 3. Sistemas embarcados (computadores) I. Título.

---

CDD 621.3810285

**MATHEUS MEDEIROS DA CRUZ**

**CLIMADUINO: SISTEMA EMBARCADO DE CLIMATIZAÇÃO PARA EDIFÍCIOS  
INTELIGENTES**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Tecnólogo em Redes de Computadores da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Tecnólogo.

Área de concentração: computação

Aprovado em: \_\_\_\_\_ / dezembro / 2013.

**BANCA EXAMINADORA**

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Atslands Rego da Rocha (Orientadora)  
Universidade Federal do Ceará-UFC

---

Prof. Dr. David Sena Oliveira  
Universidade Federal do Ceará-UFC

---

Prof. Msc. Marcos Dantas Ortiz  
Universidade Federal do Ceará-UFC

## **AGRADECIMENTOS**

À Prof<sup>ª</sup>. Dr<sup>ª</sup>. Atslands Rego da Rocha, pela dedicada orientação e por sempre mostrar soluções para os inúmeros desafios encontrados durante a criação desse trabalho, sendo para mim um referência de profissionalismo e comprometimento.

Ao bolsista do Núcleo de Capacitação em Desenvolvimento de Sistemas Embarcados – NDSE, José Carlos Carvalho, por compartilhar seus conhecimentos e ajudar na execução do projeto.

Aos professores membros da banca de defesa, Prof. Dr. David Sena Oliveira, Prof. Dr. Marcos Dantas Ortiz, pelas valiosíssimas sugestões que serviram para amadurecimento desse trabalho.

À minha filha Isadora, pelos sorrisos e abraços que alegravam intermináveis e cansativos dias de estudo.

Ao meu pai, Antonio Carlos, por inúmeras vezes que sentou do meu lado me ajudando a escrever esse trabalho, além de tê-lo como um exemplo de intelectualidade.

À minha mãe, Eleuda, por ter sempre me apoiado e nesse tempo todo cuidando de mim com carinho e dedicação.

Às minhas irmãs Camila e Sofia, pelas palavras de apoio e momentos de descontração que me fazem bem.

Aos colegas de faculdade por compartilharem as alegrias e dificuldades desses anos de convívio, além de ajudar na construção do conhecimento que resultaram nessas páginas.

## RESUMO

Em virtude do desperdício no consumo de energia em aparelhos de ar-condicionado, ligados desnecessariamente nos ambientes de universidades, e tendo como estudo de caso o campus da UFC de Quixadá, surgiu a ideia de desenvolver um sistema, baseado nos conceitos das Redes de Sensores Sem Fio (RSSF), para o controle dos aparelhos de ar-condicionado, podendo ser aplicado em outros ambientes públicos ou residenciais. Todos os equipamentos utilizados na RSSF são baseados em hardware e software Open Source, tendo como plataforma de prototipação dos sensores o Arduino e a comunicação sendo realizada através de dispositivos Xbee. O processo de automação é realizado com o envio de comandos infravermelho para aumentar/diminuir a temperatura da sala e também, ligar/desligar aparelho de ar-condicionado de forma automática através de dados coletados pela própria RSSF. O sistema de validação foi construída a partir da norma técnica NR-17, que define o conforto térmico em um laboratório. O desempenho da rede dentro dessa norma e a capacidade de economia de energia após a implantação da rede foram avaliados.

Palavras chave: RSSF. Arduino. Xbee. Automação.

## **ABSTRACT**

Due the wasteful consumption of energy of air conditioning devices, unnecessarily turned on the university environments, and taking as a case of study the UFC (Quixadá/CE), an idea arose to develop a system based on the concept of Wireless Sensor Networks (WSN) in order to the control of air-conditioning devices. Our system can be applied to both others public or residential environments. All software and hardware used in WSN are based on Open Source license. We use the Arduino platform and the Xbee communication. The automation process is performed by sending infrared commands in order to (i) increase/decrease the temperature of the room and also (ii) on/off the air-conditioner device automatically. This automation process is based on data collected by the WSN itself. The network was built from the NR-17 technical standard that defines thermal comfort in a laboratory. The performance of the network and the energy saving capacity after the system deployment were evaluated.

Keywords: Automation, energy consumption, WSN, Arduino, Xbee.

## LISTA DE FIGURAS

Figura 1 – Rede de sensores de Queiroz e Azevedo (2009).....	18
Figura 2 – Rede de sensores de Cunha (2013) .....	18
Figura 3 – Elementos de um nó sensor.....	21
Figura 4 – Placa Arduino UNO .....	28
Figura 5 – IDE de programação do Arduino .....	29
Figura 6 – Módulo Xbee.....	29
Figura 7 – Redes Xbee .....	30
Figura 8 – Arquitetura protocolo ZigBee .....	31
Figura 9 – Software X-CTU .....	32
Figura 10 – Topologia da RSSF .....	36
Figura 11 – Diagrama de Atividades - Nó Sensor.....	37
Figura 12 – Diagrama de Atividades – Nó atuador .....	42
Figura 13 – Nó Sensor .....	43
Figura 14 – Sensor de Temperatura/Umidade –DHT11.....	44
Figura 15 – Sensor de Presença - PIR .....	44
Figura 16 – Nó Atuador.....	45
Figura 17 – LED IR .....	46
Figura 18 – Esquema do circuito com o transistor .....	46

## LISTA DE TABELAS

Tabela 1 - Caracterização das RSSF segundo a Configuração.....	22
Tabela 2 - Caracterização RSSF segundo o sensoriamento .....	23
Tabela 3 - das RSSF segundo a comunicação (Parte A). .....	23
Tabela 4 - Caracterização das RSSF segundo a comunicação (Parte B).....	24
Tabela 5 - - Caracterização das Redes de Sensores Sem Fio segundo o processamento .....	24
Tabela 6 - Consumo energia aparelhos 15 e 30 minutos. ....	48

## LISTA DE GRÁFICOS

Gráfico 1 - Média de Temperatura no Lab04. ....	50
Gráfico 2 - Valores de presença no Lab04. ....	51

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>15</b>
1.1	<i>Trabalhos relacionados.....</i>	16
1.2	<i>Motivação.....</i>	19
1.3	<i>Objetivos.....</i>	19
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>20</b>
2.1	<i>Redes de Sensores Sem Fio .....</i>	20
2.2	<i>Sistemas Embarcados.....</i>	25
2.2.1	<i>Microcontroladores .....</i>	27
2.3	<i>Tecnologias Utilizadas.....</i>	27
2.3.1	<i>Arduino.....</i>	27
2.3.2	<i>Xbee.....</i>	29
2.4	<i>Conforto Térmico.....</i>	32
2.4.1	<i>Norma Regulamentadora N° 17 (NR-17) .....</i>	33
<b>3</b>	<b>DESENVOLVIMENTO .....</b>	<b>34</b>
3.1	<i>Requisitos da Automação .....</i>	34
3.2	<i>Clonagem do Controle Remoto.....</i>	34
3.3	<i>Topologia da Rede.....</i>	36
3.4	<i>O Software Embarcado.....</i>	37
3.4.1	<i>Código dos Sensores.....</i>	37
3.4.2	<i>Código dos Atuadores.....</i>	39
3.5	<i>O Hardware.....</i>	43
3.5.1	<i>Nó Sensor .....</i>	43
3.5.2	<i>Nó Atuador .....</i>	45
<b>4</b>	<b>ESTUDO DE CASO .....</b>	<b>47</b>
4.1	<i>Ambientes Monitorados.....</i>	47
4.2	<i>Cálculos de Consumo.....</i>	47
4.3	<i>Testes.....</i>	49
4.4	<i>Limitações.....</i>	51
<b>5</b>	<b>CONCLUSÕES.....</b>	<b>52</b>
5.1	<i>Trabalhos Futuros.....</i>	52
	<b>REFERÊNCIAS.....</b>	<b>54</b>
	<i>ANEXO A – Tutorial Configurações Básicas XBee – CHAT .....</i>	58
	<i>ANEXO B – Tutorial Clonagem Controle Remoto Com Arduino .....</i>	61
	<i>ANEXO B – Tutorial Sensor PIR.....</i>	68
	<i>ANEXO C – Tutorial Sensor DHT11.....</i>	70

## 1 INTRODUÇÃO

Na década de 70, o uso dos computadores nas indústrias passou a ser imprescindível, substituindo algumas atividades de maior precisão e de maiores riscos à mão-de-obra humana. Surgiu assim a ideia de automação. Entende-se por automação a capacidade de se executar comandos, obter medidas, regular parâmetros e controlar funções automaticamente, sem a intervenção humana (PINHEIRO, 2004).

Uma década depois, esse processo que antes era industrial passou a atender as exigências do comércio automatizando os sistemas de iluminação, condicionadores de ar, segurança patrimonial, entre outros, surgindo os chamados “edifícios inteligentes”. Os edifícios inteligentes são definidos pelo “Intelligent Building Institute” (INSTITUTE, 2013) como “um edifício que integra vários sistemas para controlar de forma coordenada e eficaz os recursos visando maximizar o desempenho do usuário, economia nos custos de investimentos e flexibilidade.” (DEFINITIONS, 2013), possibilitando maior conforto, segurança e o consumo racional de energia.

Com a expansão das tecnologias específicas e para atender a um público mais sofisticado, os sistemas automatizados foram incorporados às residências criando um novo conceito, a Domótica. A palavra domótica deriva das palavras Domus (casa) e Robótica (controle automatizado de algo). A domótica define-se como a possibilidade de controle de forma automática das nossas casas tornando-as o que vulgarmente se costuma designar por “casas inteligentes” (ALVES, 2003). Quis o autor na citação, quando coloca o termo “vulgarmente”, afirmar que só um sistema isolado não configura uma casa ou um edifício como inteligente. Ao contrário, para ter essa denominação a residência deve ser formada por uma rede interligando vários sistemas de modo que um parâmetro específico para uma aplicação poderá resultar numa ação em outra. Assim, se os sensores de temperatura de um sistema de climatização perceber altos valores, podem acionar um sistema de combate a incêndio, por exemplo.

Para ter um ambiente domótico devemos observar dois fatores fundamentais: o custo e a interferência nas instalações do ambiente. Segundo Braga (2009, p. 02),

[...] para que a domótica seja utilizada para fins de uso racional, suas aplicações requerem equipamentos de baixo custo por ambiente monitorado, assim como de reduzidas dimensões, a fim de propiciar instalações camufláveis, que não interfiram na arquitetura do ambiente a ser automatizado.

Uma solução que atende a esses requisitos é a Rede de Sensores Sem Fio (RSSF). As RSSFs são utilizadas em praticamente todas as áreas, como: operações militares, meteorologia, monitoramento de veículos, agricultura de precisão, medicina, e inclusive na automação predial. Uma das vantagens do uso de RSSFs em automação predial é devido a sua instalação rápida e de fácil execução. Pelo fato da tecnologia não usar fios, a RSSF dispensa a implantação de uma infraestrutura de grande custo, sendo ideal para casas ou prédios onde não existe mais espaços ou que não sejam permitidas alterações bruscas na sua estrutura, ou como por exemplo em prédios históricos.

Sendo assim, este trabalho tem como alvo implementar um membro de um Sistema Domótico, no caso, uma automação dos aparelhos condicionadores de ar, utilizando algumas características das RSSFs.

## 1.1 Trabalhos relacionados

O presente trabalho apresenta uma similaridade em relação aos trabalhos de *Queiroz e Azevedo (2009)* e *Cunha (2013)*. Nesse projeto (assim como no de *Queiroz e Azevedo (2009)*), o ambiente monitorado será composto por vários nós, estrategicamente dispostos, para medir a temperatura do ambiente e um nó atuador para cada ar condicionado que realizará ações de acordo com os dados recebidos pelos nós sensores (Figura 1). No nosso trabalho acreditamos, que a presença de vários nós sensores em um cômodo realizando o monitoramento da temperatura ambiental, poderá retornar uma medição com maior acurácia.

No trabalho de *Cunha (2013)*, para cada ambiente com um aparelho de ar-condicionado, há apenas um nó com responsabilidades de sensoriamento e atuador (Figura 2). Uma das diferenças entre essa proposta e a de *Cunha (2013)* é que a quantidade de nós instalados no ambiente vai ser proporcional à quantidade de aparelhos de ar-condicionado e ao tamanho do ambiente.

Além das diferenças já citadas, o principal diferencial desse trabalho em relação aos outros projetos referenciados é a implementação de uma RSSF autônoma, ou seja, alguns nós da rede serão inteligentes quanto à análise e tomada de decisão. Nos demais trabalhos

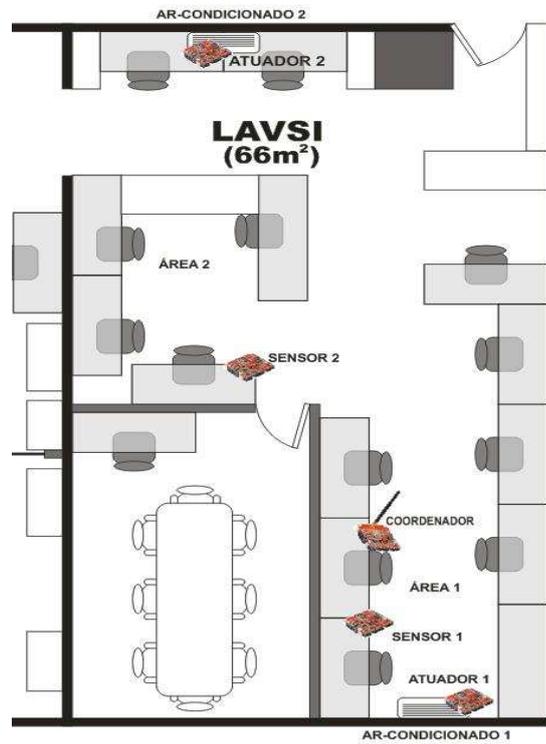
relacionados, todos os nós apenas coletam os dados e os enviam a um servidor que realiza, de forma centralizada, a análise dos dados e a tomada de decisão com base nos resultados obtidos. Uma desvantagem dessa abordagem é que em caso de falha do servidor, o sistema é interrompido.

Nesse projeto, os dados serão analisados e processados por nós da rede, que realizarão também as tomadas de decisão. Essa abordagem reduz a comunicação entre os nós e o servidor, tornando a tomada de decisão mais rápida. Além disso, opcionalmente, o sistema poderá contar com uma plataforma do tipo Web para o controle da rede pelo usuário administrador.

No presente projeto, além da temperatura, os nós terão sensores de presença que irão permitir o ligar e desligar do condicionador de ar de forma automática, quando existir a presença ou a ausência de pessoas no recinto respectivamente, com o intuito do uso sustentável de energia. Outro ponto diferencial entre os trabalhos relacionados, é a forma de controle dos aparelhos condicionadores de ar. Nos trabalhos citados, o controle de temperatura é realizado através do hardware do aparelho. Como exemplo: ligam e desligam o equipamento através de relés. Esse trabalho visa prover esse serviço através de códigos IR (Infravermelho), preservando ao máximo o aparelho.

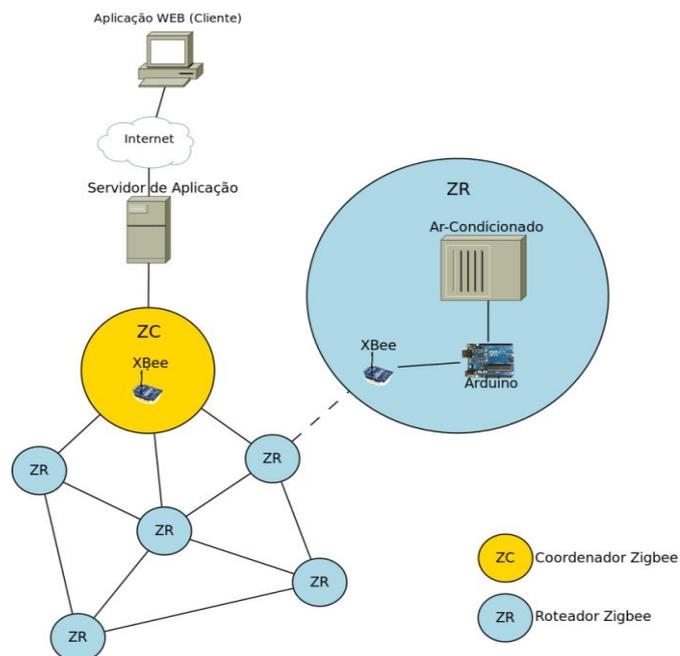
A escolha das plataformas para este trabalho não foi baseada na comparação entre as tecnologias para identificar apenas qual é a melhor, mas principalmente pelos recursos disponíveis e pelo custo/benefício do projeto.

Figura 1 – Rede de sensores de Queiroz e Azevedo (2009)



Fonte: Queiroz e Azevedo (2009)

Figura 2 – Rede de sensores de Cunha (2013)



Fonte: Cunha (2013)

## 1.2 Motivação

A razão motivacional desse trabalho é a busca da comodidade do usuário e o uso da energia de forma sustentável, que são os principais objetivos da automação. Um dos aspectos que foram observados é que com o crescimento de sua estrutura física e o número de aparelhos do Campus da UFC de Quixadá (local de aplicação do projeto), o responsável por desligar os aparelhos terá um maior trabalho de executar essa tarefa.

Outro fator imprescindível é a possibilidade de desperdício de energia. Nos períodos de intervalo de aulas e final de turnos de expediente, qualquer demora existente no desligamento dos aparelhos pode acarretar em uso dos recursos de forma inadequada. Portanto, a aplicação proposta neste trabalho pretende ser uma solução para os problemas de gerenciamento de uso dos aparelhos (liga/desliga) e do consumo desnecessário de energia, que são comuns em ambientes onde o contexto seja similar ao contexto das universidades.

## 1.3 Objetivos

O objetivo geral do presente trabalho é desenvolver uma aplicação RSSF de automação de condicionadores de ar para utilização nos laboratórios e salas de universidades e outros ambientes públicos afins, baseado em seu contexto visando comodidade e a economia de energia. Com os seguintes objetivos específicos que dão escopo ao trabalho são:

- Desenvolvimento de uma aplicação embarcada que irá controlar os aparelhos de condicionadores de ar;
- Desenvolvimento do protótipo de nós sensores e do nó atuador;
- Estudo do comportamento de utilização dos aparelhos e;
- Verificação do consumo de energia após a implantação do sistema.

## 2 REVISÃO BIBLIOGRÁFICA

Nesta seção serão apresentados os principais conceitos que fundamentam esse trabalho, que são: redes de sensores sem fio e suas características (2.1), a definição de sistemas embarcados (2.2), as tecnologia utilizadas (2.3) e a resolução do ministério da saúde sobre conforto térmico (2.4).

### 2.1 Redes de Sensores Sem Fio

Redes de Sensores Sem Fio (RSSFs) são formadas por um grande número de dispositivos autônomos chamados nós sensores. Caracterizam-se por ser um tipo especial de rede sem fio ad-hoc e tem um papel importante na computação ubíqua. (RUIZ et al, 2004, p.01).

Os nós sensores são dispositivos com capacidade de sensoriamento, processamento, armazenamento e comunicação e têm como característica, a limitação de recursos como, largura de banda, processamento, memória e energia. Este último é um dos desafios da área. Segundo Oliveira (2012, p 01), “A limitação de energia é o fator mais crítico, pois pode levar à inoperabilidade do nó sensor de forma prematura e comprometer o desempenho da rede”.

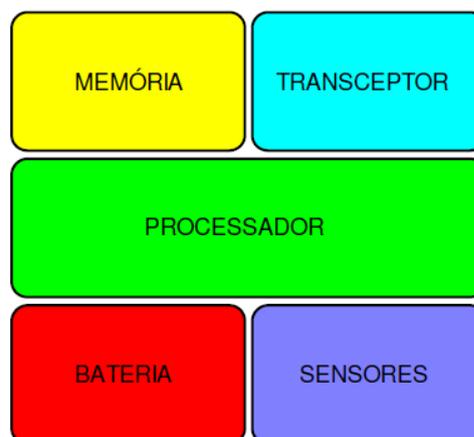
As Redes de Sensores Sem Fio são redes *ad-hoc*. A principal característica de uma rede ad-hoc é o fato de não existir uma infraestrutura fixa, como existe nas redes tradicionais, possibilitando que a informação possa ser trocada entre os elementos da rede (LOUREIRO et al, 2003). Essa característica da rede *ad-hoc* é muito importante em relação à área de cobertura da rede, pois quando os nós sensores estiverem em uma posição em que não podem se comunicar com os sorvedouros, não implica, obrigatoriamente, que os mesmos não poderão se comunicar, pois a existência de sensores intermediários poderá viabilizar a comunicação (SEIDEL et al, 2007).

Diferentemente das redes *ad-hoc* tradicionais, as RSSF procuram exercer um papel colaborativo cujos elementos (sensores) prove em dados que serão processados pelos nós sorvedouros (nós mais robustos responsáveis da comunicação da rede com ambientes externos), ou por dispositivos que podem controlar parâmetros do ambiente controlado, chamados de nós atuadores (LOUREIRO et al, 2003).

Os nós sensores apresentam os seguintes componentes básicos: transceptor, memória, processador, sensor e bateria. Por apresentar pequenas dimensões em seu tamanho,

tornam os recursos dos componentes limitados. Embora essas limitações ainda sejam um grande desafio pra as RSSFs, avanços tecnológicos e pesquisas na área mostram que são restrições temporárias (ROCHA et al, 2012). A Figura 3 faz a representação desses elementos.

Figura 3 – Elementos de um nó sensor



Fonte: PEREIRA et al, 2003

Além dos nós sensores, os outros principais elementos de uma RSSF são o observador e o fenômeno observado (PEREIRA et al, 2003). O observador é o usuário interessado em obter as informações coletadas pela rede de sensores em relação a um fenômeno. A rede pode operar de modo ativo ou reativo em relação ao observador. Quando envia respostas a consultas feitas pelo usuário, a rede está em modo reativo. Em contraposição, a rede pode enviar dados periodicamente sem que o usuário solicite, ou seja, de modo ativo. Além disso, podem existir, simultaneamente, múltiplos observadores numa rede de sensores (PEREIRA et al, 2003).

“O fenômeno é a entidade de interesse do observador, que está sendo monitorada e cuja informação potencialmente será analisada/filtrada pela rede de sensores. Além disso, múltiplos fenômenos podem ser observados concorrentemente numa rede.” (PEREIRA et al, 2003).

Em alguns casos, os nós sensores não são dispostos de forma determinada na área onde se deseja coletar dados, podem ser simplesmente dispostos de forma aleatória. Antes de iniciarem as atividades de sensoriamento, os nós despertam para estabelecer a rede. No estabelecimento da rede, existem algumas formas do nó sensor descobrir outro nó vizinho, podendo usar, por exemplo, medidas de intensidade de sinal ou algoritmos de descoberta de localização (LOUREIRO et al, 2003). Em seguida, a rede torna-se um sistema auto-

organizado (*self-organizing*), ou seja, os nós podem se organizar da maneira que acharem melhor e se por uma eventual falha ou degradação de algum membro do sistema, ser capaz de achar uma rota alternativa para não deixa a área descoberta, e mais importante sem a necessidade de intervenção humana (LOUREIRO et al, 2003).

As características das RSSFs dependerão exclusivamente da aplicação, os protocolos de comunicação, a quantidade de nós que vão compor a rede, o tipo de dado que será tratado, o tipo de serviço que será provido pela rede e o tempo de vida da rede (RUIZ et al, 2004).

De acordo com Ruiz (2004), as RSSFs podem ser classificadas conforme o sensoriamento (Tabela 1), a configuração (Tabela 2) e segundo o tipo de comunicação (Tabelas 3 e 4). Uma RSSF também pode ser distinta segundo o tipo de processamento que executa (Tabelas 5). Estas tabelas poderão ser visualizadas no anexo (ANEXO D).

Tabela 1 - Caracterização das RSSF segundo a Configuração

<b>Configuração</b>		
<b>Composição</b>	Homogênea	Rede composta de nós que apresentam a mesma capacidade de hardware. Eventualmente os nós podem executar software diferente.
	Heterogênea	Rede composta por nós com diferentes capacidades de hardware.
<b>Organização</b>	Hierárquica	RSSF em que os nós estão organizados em grupos (clusters). Cada grupo terá um líder (cluster-head) que poderá ser eleito pelos nós comuns. Os grupos podem organizar hierarquias entre si.
	Plana	Rede em que os nós não estão organizados em grupos
<b>Mobilidade</b>	Estacionária	Todos os nós sensores permanecem no local onde foram depositados durante todo o tempo de vida da rede.
	Móvel	Rede em que os nós sensores podem ser deslocados do local onde inicialmente foram depositados.
<b>Densidade</b>	Balanceada	Rede que apresenta uma concentração e distribuição de nós por unidade de área considerada ideal segundo a função objetivo da rede.
	Densa	Rede que apresenta uma alta concentração de nós por unidade de área.
	Esparsa	Rede que apresenta uma baixa concentração de nós por unidade de área.
<b>Distribuição</b>	Irregular	Rede que apresenta uma distribuição não uniforme dos nós na área monitorada.
	Regular	Rede que apresenta uma distribuição uniforme de nós sobre a área monitorada.

Fonte: RUIZ (2004, p. 172).

Tabela 2 - Caracterização RSSF segundo o sensoriamento

<b>Sensoriamento</b>		
<b>Coleta</b>	Periódica	Os nós sensores coletam dados sobre o(s) fenômeno(s) em intervalos regulares. Um exemplo são as aplicações que monitoram o canto dos pássaros. Os sensores farão a coleta durante o dia e permaneceram desligados durante a noite.
	Contínua	Os nós sensores coletam os dados continuamente. Um exemplo são as aplicações de exploração interplanetária que coletam dados continuamente para a formação de base de dados para pesquisas
	Reativa	Os nós sensores coletam dados quando ocorrem eventos de interesse ou quando solicitado pelo observador. Um exemplo são as aplicações que detectam a presença de objetos na área monitorada.
	Tempo Real	Os nós sensores coletam a maior quantidade de dados possível no menor intervalo de tempo. Um exemplo são aplicações que envolvem risco para vidas humanas tais como aplicações em escombros ou áreas de desastres. Um outro exemplo são as aplicações militares onde o dado coletado é importante na tomada de decisão e definição de estratégias.

Fonte: RUIZ (2004, p. 171).

Tabela 3 - das RSSF segundo a comunicação (Parte A).

<b>Classificação Segundo a Comunicação</b>		
<b>Disseminação</b>	Programada	Os nós disseminam em intervalos regulares.
	Contínua	Os nós disseminam os dados continuamente.
	Sob Demanda	Os nós disseminam os dados em resposta a consulta do observador e a ocorrência de eventos.
<b>Tipo Conexão</b>	Simétrica	Todas as conexões existentes entre os nós sensores, com exceção do nó sorvedouro têm o mesmo alcance.
	Assimétrica	As conexões entre os nós comuns têm alcance diferente.
<b>Transmissão</b>	Simplex	Os nós sensores possuem transceptor que permite apenas transmissão da informação.
	Half-duplex	Os nós sensores possuem transceptor que permite transmitir ou receber em um determinado instante.
	Full-duplex	Os nós sensores possuem transceptor que permite transmitir ou receber dados ao mesmo tempo.

Fonte: RUIZ (2004, p. 172).

Tabela 4 - Caracterização das RSSF segundo a comunicação (Parte B).

<b>Classificação Segundo a Comunicação</b>		
<b>Alocação de Canal</b>	Estática	Neste tipo de rede se existirem "n" nós, a largura de banda é dividida em "n" partes iguais na frequência (FDMA - Frequency Division Multiple Access), no tempo (TDMA - Time Division Multiple Access), no código (CDMA - Code Division Multiple Access), no espaço (SDMA - Space Division Multiple Access) ou ortogonal (OFDM – Orthogonal Frequency Division Multiplexing). A cada nó é atribuída uma parte privada da comunicação, minimizando interferência.
	Dinâmica	Neste tipo de rede não existe atribuição fixa de largura de banda. Os nós disputam o canal para comunicação dos dados.
<b>Fluxo de Informação</b>	Flooding	Neste tipo de rede, os nós sensores fazem broadcast de suas informações para seus vizinhos que fazem broadcast desses dados para outros até alcançar o ponto de acesso. Esta abordagem promove um alto overhead mas está imune às mudanças dinâmicas de topologia e a alguns ataques de impedimento de serviço (DoS - Denial of Service). e a ocorrência de eventos.
	Multicast	Neste tipo de rede os nós formam grupos e usam o multicast para comunicação entre os membros do grupo. do nó sorvedouro têm o mesmo alcance.
	Unicast	Neste tipo de rede, os nós sensores podem se comunicar diretamente com o ponto de acesso usando protocolos de roteamento multi-saltos.
	Gossiping	Neste tipo de rede, os nós sensores selecionam os nós para os quais enviam os dados.
	Bargaining	Neste tipo de rede, os nós enviam os dados somente se o nó destino manifestar interesse, isto é, existe um processo de negociação. manifestar interesse, isto é, existe um processo de negociação em um determinado instante.

Fonte: RUIZ (2004, p. 172).

Tabela 5 - - Caracterização das Redes de Sensores Sem Fio segundo o processamento

<b>Classificação Segundo o Processamento</b>		
<b>Cooperação</b>	Infra-estrutura	Os nós sensores executam procedimentos relacionados a infraestrutura da rede como por exemplo, algoritmos de controle de acesso ao meio, roteamento, eleição de líderes, descoberta de localização e criptografia.
	Localizada	Os nós sensores executam além dos procedimentos de infraestrutura, algum tipo de processamento local básico como por exemplo, tradução os dados coletado pelos sensores baseado na calibração e a ocorrência de eventos.
	Correlação	Os nós estão envolvidos em procedimentos de correlação de dados como fusão, supressão seletiva, contagem, compressão, multi-resolução agregação.

Fonte: RUIZ (2004, p. 173).

Em relação essas classificações, a RSSF proposta por esse trabalho tem as seguintes características:

- A composição é homogênea, pois todos apresentam a mesma capacidade de hardware;
- A organização é plana pois os nós não estão agrupados em grupos;
- A mobilidade é estacionária. Os sensores são estrategicamente posicionados;
- A densidade é balanceada;
- Distribuição regular;
- O sensoriamento é periódico;
- Na comunicação a disseminação é contínua, o tipo de conexão é simétrica e a transmissão é entre os sensores e o atuador é half-duplex e entre o atuador e o nó *sink* full-duplex;
- Segundo o processamento é de cooperação localizada, ou seja, existe processamento em um nó da rede.

## 2.2 Sistemas Embarcados

Um sistema embarcado é a combinação entre software, hardware e outras partes adicionais, assim como um PC. Por exemplo, ele possui: memória, processador, dispositivos de armazenamento, com a diferença de ser responsável por executar bem uma única tarefa, de maneira contínua e em muitas vezes, sem falhas ou panes, podendo ter alguma rede de comunicação com o exterior (MORIMOTO, 2007).

Hoje em dia, os sistemas embarcados estão presentes em todos os tipos de equipamentos como celulares, forno micro-ondas, caixa eletrônico, geladeiras, televisores, entre outros equipamentos. Ainda mais nos últimos anos, motivados por um novo paradigma da computação denominado Internet das Coisas – *Internet of Things (IoT)*. A *IoT*, permitirá a comunicação de qualquer coisa do mundo físico ao ambiente virtual, possibilitando que máquinas, dados e pessoas interajam de forma transparente.

Os sistemas embarcados podem utilizar-se de sensores para obter parâmetros e com eles controlar, ajustar e configurar aplicações ou dispositivos para melhor satisfazer a necessidade das pessoas (ARAUJO, 2003). Os softwares desenvolvidos para sistemas embarcados, também chamados de firmware, são armazenados em uma memória ROM ou memória flash, e não em discos rígidos. Uma vez que o software é embarcado no dispositivo,

difícilmente poderão ser realizadas modificações no código para que o dispositivo execute outra tarefa. A única maneira será a realização de um upgrade para uma nova versão do firmware, fazendo com que o sistema seja reprogramado.

Segundo Cunha (2008), além da sua capacidade computacional e da sua independência de operações, os sistemas embarcados apresentam outras características: os tipos de aplicações, modos de funcionamento e itens desejados em aplicações embarcadas.

Os tipos de aplicações podem ser de propósito geral. Essas são aplicações mais parecidas com as dos computadores tradicionais, tendo maior interação com o usuário através de terminais de vídeo ou monitores, como por exemplo, videogames, caixas de autoatendimento, etc.

Outro tipo de aplicação é o sistema de controle, são aplicações mais complexas, com hardware dedicado e inúmeros sensores, pouca interação com o usuário sendo bastante autônomo. Usado em automação de indústria química, controle de voos, usinas nucleares.

Por fim, um terceiro tipo de sistema é o de comunicação e redes, utilizado em equipamentos de redes como roteadores e switches. Utilizados por empresas de telefonia, telecomunicações e internet.

Os modos de funcionamento são muito importantes para saber como programar o dispositivo e como ele irá se comportar com a aplicação. Os modos podem ser reativos, ou seja, o funcionamento se dá por uma resposta a um evento externo, por intervenção humana ou verificações periódicas do ambiente monitorado. Ou podem ser em modo de controle em tempo real, executando certa atividade numa faixa de tempo limite.

Reforça o autor (CUNHA, 2008), que alguns requisitos são essenciais aos sistemas embarcados. Um deles é o tamanho e o peso. Com a miniaturização dos equipamentos eletrônicos é esperado que os dispositivos dos sistemas embarcados sejam os menores e mais discretos possíveis. Outro fator é o consumo de energia. Espera-se do sistema que ele seja autônomo e consuma menos energia, sem a necessidade de ficar recarregando ou trocar sistemas de alimentação tornando o produto mais competitivo no mercado. Por último, robustez do equipamento. Muitos sistemas embarcados são desenvolvidos para resistir a situações adversas, como por exemplo, altas temperaturas, variação de tensão, umidade, interferência eletromagnética.

### 2.2.1 Microcontroladores

O embarque em chips está cada vez mais comum, fazendo com que sistemas inteiros fiquem disponíveis numa só pastilha também conhecida como circuitos integrados. Existem muitas variedades de circuitos integrados e os microcontroladores são os mais utilizados em ambientes de automação. O microcontrolador foi especificamente desenvolvido para o uso em aplicações embarcadas (MICHAEL BARR, 1999) e é um circuito integrado programável que contém todos os componentes de um computador e alguns outros, como sensores, no mesmo chip (USATEGUI; MARTÍNEZ, 2003).

Eles estão cada vez mais sendo evoluídos e utilizados para substituir diversas atividades manuais. Essa evolução acontece por dois fatores. Uma é pela procura por meios mais baratos de colocar inteligência nos produtos utilizados no dia-a-dia, como microondas, celulares, GPSs entre outros e a outra, motivada pela indústria, para desenvolver soluções para aplicações que envolvam processamentos em intervalos críticos de tempo, como em usinas nucleares, estimulando o desenvolvimento de microcontroladores mais sofisticados e rápidos.

## 2.3 Tecnologias Utilizadas

Nessa seção são apresentadas as principais tecnologias utilizadas na prototipação, programação e comunicação dos equipamentos utilizados neste trabalho.

### 2.3.1 Arduino

É um projeto iniciado em 2005, em Ivrea na Itália. A ideia é criar uma plataforma de prototipagem de baixo custo, dando a possibilidade a pessoas sem muito conhecimento técnico, de desenvolver seus próprios projetos.

O Arduino foi desenvolvido em hardware e software de código aberto, composta de uma placa microcontroladora, e um ambiente de desenvolvimento (*Integrated Development Environment* – IDE) para escrever o código para a placa. Tendo o seu microcontrolador, da família ATmega, permite a automação de inúmeras equipamentos, desde ligar lâmpada, obter informações de sensores, até controles de motores (MELLIS, 2009).

O Hardware (Figura 4) pode ser comprado pronto ou reproduzido artesanalmente. O esquema de confecção da placa está disponível no site oficial da empresa.

Caso o protótipo se torne um produto comercializado, não é obrigatório a divulgação de qualquer informação do projeto, mas requer que você disponibilize arquivos que permitam a religação do firmware com versões atualizadas do núcleo e bibliotecas do Arduino (MELLIS, 2009).

O microcontrolador utilizado na placa possui portas analógicas, portas para comunicação serial e entradas e saídas digitais (algumas destas saídas podem produzir sinais modulados por largura de pulso (Pulse Width Modulation –PWM)). Existem também módulos que aumentam as funcionalidades da placa do Arduino, chamado de *Shields*. Estes se comunicam com o microcontrolador através de pinos ou via barramento serial, permitindo a expansão das funcionalidades, pois possuem outras interfaces, como Ethernet ou Wi-fi (CUNHA, 2013).

Figura 4 – Placa Arduino UNO

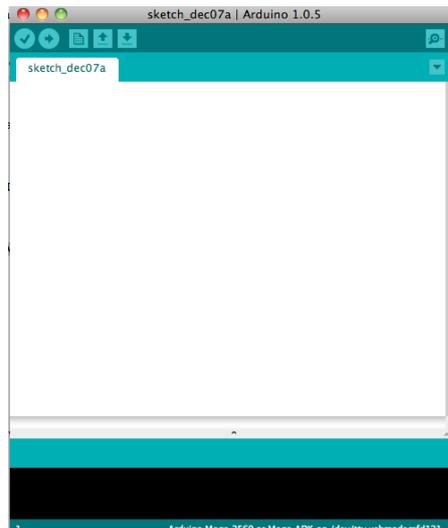


Fonte: MELLIS, 2009

O software responsável por passar instruções ao microcontrolador é escrito na linguagem C/C++ e torna transparente diversos aspectos da programação de baixo nível, tornando a programação mais amigável.

O projeto também disponibiliza uma IDE (Figura 3.2) para facilitar a programação que pode ser obtida na página oficial (MELLIS, 2009) da comunidade. Essa IDE além de apresentar um ambiente para a escrita do código, verifica erros no código e importa os comandos ao microcontrolador (MELLIS, 2009).

Figura 5 – IDE de programação do Arduino



Fonte: elaborado pelo autor

### 2.3.2 Xbee

O módulo de comunicação XBee (Figura 6), desenvolvida pela Digi International, oferece conectividade sem fio para as placas Arduino e possui um microcontrolador com o protocolo ZigBee embarcado. Esse módulo é disponibilizado em duas versões, sendo eles XBee e o XBee-PRO. Ambos possuem as mesmas funcionalidades e a diferença está na potencia de transmissão (ESCHNER, 2011).

Figura 6 – Módulo Xbee



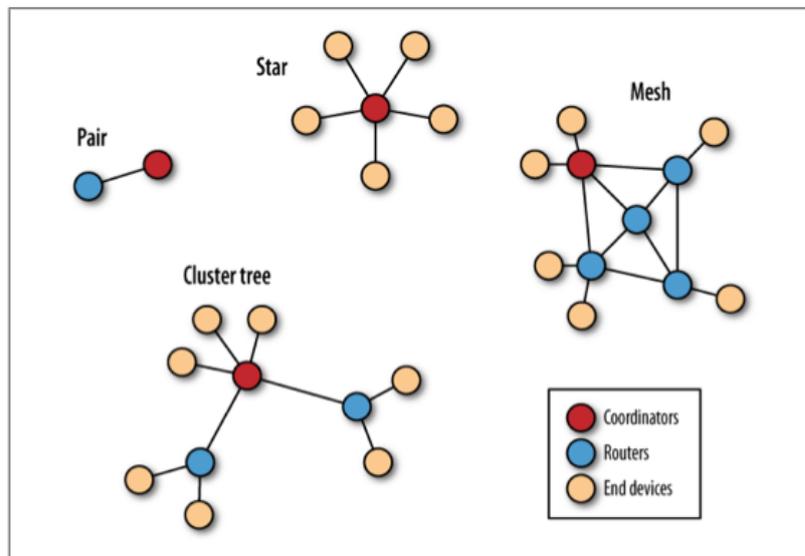
Fonte: ARDUINO, 2013

O protocolo ZigBee utilizado pelo XBee é desenvolvido pela ZigBee Alliance. Ele é um protocolo destinado a aplicações embarcadas que visa o baixo consumo de energia e baixa transferências de dados (CUNHA, 2013).

Originalmente, a tecnologia ZigBee foi desenvolvida com o objetivo de criar redes em malha (*mesh*), mas suporta topologias em estrela (*star*) e em árvore (*cluster tree*), admitindo a formação de redes *ad-hoc* (Figura 7), explicando a sua grande utilização em rede de sensores sem fio, ainda mais por fornecer confiabilidade e maior alcance (SALEIRO, 2013).

No protocolo ZigBee são definidos três tipos de dispositivos: o coordenador, roteador e dispositivos finais. O coordenador (*ZC - ZigBee Coordinator*) é o responsável pelo estabelecimento da rede ZigBee e pela interligação de entre redes, tornando obrigatória a sua presença. O coordenador constrói um canal de operação e um número lógico para formar a rede. Após selecionado o endereço de rede (*PAN ID*) o coordenador permitirá a comunicação entre os dispositivos finais e roteadores (SOLDA et al, 2013). O roteador (*ZR - ZigBee Router*), tem a função de armazenar informações dos nós da rede e com elas determinar as melhores rotas para os pacotes de dados. O dispositivo final sempre deve está vinculado a um roteador ou diretamente ao coordenador, não possuem a capacidade de se comunicarem diretamente. É recomendado que sejam utilizados como nós sensores pois permitem a capacidade de trabalharem no modo adormecido (*sleep*), obtendo uma maior economia de baterias.

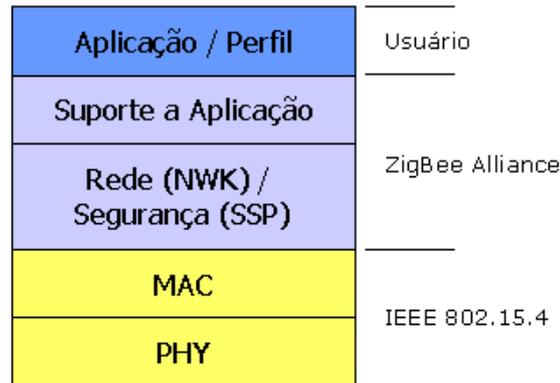
Figura 7 – Redes Xbee



Fonte: FALUDI, 2011

O ZigBee, é um padrão que complementa o padrão IEEE 802.15.4. Define as camadas superiores, de rede e suporte/aplicação (Figura 8). O padrão IEEE 802.15.4 só define a camada física (PHY) e de transmissão (MAC) (CUNHA, 2013).

Figura 8 – Arquitetura protocolo ZigBee

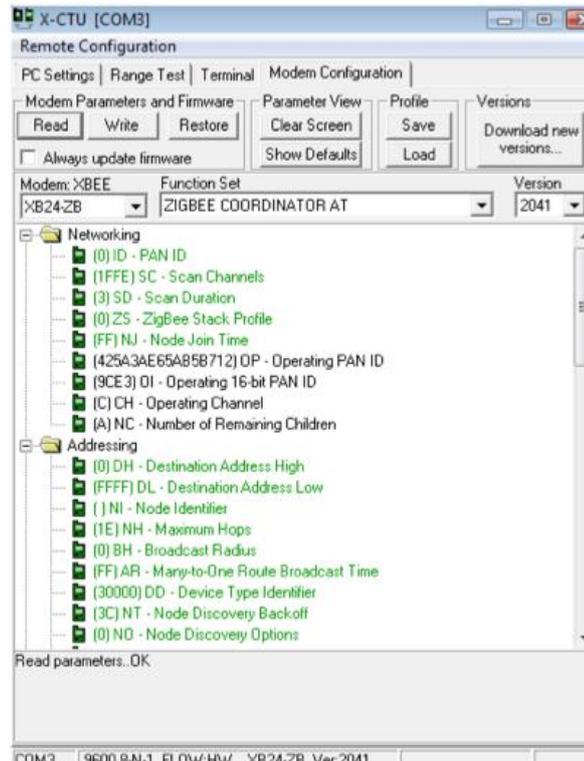


Fonte: SALEIRO; EY, 2009

Os modos de comunicação dos dispositivos XBee são divididos de duas maneiras: modo de comando (API) e modo transparente (AT). De forma simplória, no modo AT os rádios XBee servem simplesmente de interface de envio e recebimento de informações pela porta serial. Já no modo API, os dados são empacotados numa estrutura de comunicação definida pelo próprio XBee. O modo AT é mais simples de usar pois foi projetado para ter uma interação humana mais direta. Entretanto, o modo API é mais completo com funcionalidades extras, porém as suas configurações são bem mais complicadas.

Para configurar os dispositivos XBee, é utilizado o software X-CTU (Figura 9). Encontra-se em anexo um modelo de configuração dos dispositivos Xbee (Anexo A).

Figura 9 – Software X-CTU



Fonte: elaborado pelo autor

## 2.4 Conforto Térmico

Conforto térmico é definido pela norma técnica ASHARE 55 (ASHRAE, 2004), uma norma que determina condições térmicas de um ambiente para ocupação humana, como: “a condição da mente que expressa satisfação com o ambiente térmico”. Do ponto de vista físico, conforto térmico é quando o corpo humano consegue manter a temperatura interna sem utilizar os seus mecanismos reguladores e isso ocorre quando existe um equilíbrio entre o calor produzido pelo organismo e o calor perdido para o ambiente (QUEIROZ; AZEVEDO, 2009).

O conforto térmico proporciona um bem-estar, além do aumento de rendimento das pessoas. A sensação de conforto é subjetiva, estando ligada a questões tanto ambientais como pessoais. Quando duas ou mais pessoas são submetidas às mesmas condições de temperatura, podem ter impressões diferentes em relação ao estado térmico do local (QUEIROZ; AZEVEDO, 2009).

Devida a essa subjetividade, foram criadas normas técnicas que através de estudos estatísticos conseguem estabelecer padrões de conforto que agradam o maior número de pessoas. Os padrões mais populares são: ASHRAE 55 (Condições térmicas de um ambiente para ocupação humana), ASHRAE 62 (ASHARE, 2003) (Ventilação e qualidade do ar aceitável em ambientes internos), ISO 7933 (ERGONOMICS..., 2013)(Ambientes Quentes – e terminação analítica e interpretação do stress térmico a partir do cálculo da taxa de suor requerida), NR-17 . (DISPOSIÇÕES..., 2013).

#### **2.4.1 Norma Regulamentadora Nº 17 (NR-17)**

As Normas Regulamentadoras, conhecidas também como NRs, regulamentam e orientam sobre os procedimentos obrigatórios à segurança e a medicina do trabalho, e devem ser obrigatoriamente observados pelas empresas privadas e públicas e pelos órgãos públicos da administração direta e indireta. (DISPOSIÇÕES..., 2013)

A NR – 17 trata da ergonomia, ou seja, das condições ideais de trabalho às características psicofisiológicas dos trabalhadores, proporcionando um máximo de conforto, segurança e desempenho eficiente.

No texto publicado no site do Ministério do Trabalho e Emprego (ERGONOMIA..., 2013), no item 17.5.2 fala:

17.5.2. Nos locais de trabalho onde são executadas atividades que exijam solicitação intelectual e atenção constantes, tais como: salas de controle, laboratórios, escritórios, salas de desenvolvimento ou análise de projetos, dentre outros, são recomendadas as seguintes condições de conforto:

- a) níveis de ruído de acordo com o estabelecido na NBR 10152, norma brasileira registrada no INMETRO;
- b) índice de temperatura efetiva entre 20°C (vinte) e 23°C (vinte e três graus centígrados);**
- c) velocidade do ar não superior a 0,75m/s;
- d) umidade relativa do ar não inferior a 40 (quarenta) por cento.

### **3 DESENVOLVIMENTO**

Essa seção tem o objetivo de mostrar como foram desenvolvidos os protótipos utilizados nos experimentos. Será detalhado como está configurada a topologia da rede (3.2), os softwares utilizados para programar os nós sensores e atuadores (3.3) e o hardware dos protótipos (3.4).

#### **3.1 Requisitos da Automação**

A automação proposta neste trabalho é descrita da seguinte forma: os protótipos de sensores sem fio serão distribuídos no ambiente a ser monitorado. Os nós coletarão informações de temperatura e presença e enviarão os valores para um nó atuador que terá o código do controle remoto do condicionador de ar embarcado. Esse nó terá uma calibração com parâmetros de temperatura baseado no padrão NR-17 (temperatura entre 20°C a 23°C), ficando em uma posição estratégica e passará comandos automaticamente para o aparelho similar a um controle remoto.

O outro contexto é o equipamento ligar e desligar automaticamente na presença/ausência de pessoas no ambiente, através das informações enviadas dos sensores de presença. Apesar de toda autonomia, a rede poderá ser comandada por um sistema centralizado do tipo Web para o possível gerenciamento da aplicação pelo usuário administrador, se ele assim desejar.

#### **3.2 Clonagem do Controle Remoto**

Até chegarmos no processo de envios de comandos infravermelho utilizado nesse trabalho, foram testadas algumas formas de clonagem de controles. Duas formas foram as mais estudadas, resultando no método atual.

A primeira foi utilizando um método, disponível blog Ken Shirriff's<sup>1</sup>. Nesse método é utilizado uma biblioteca desenvolvida para a plataforma arduino chamada IRremote.h. Essa biblioteca apresenta por padrão alguns protocolos de comunicação que utilizam a tecnologia infra vermelho. Podemos citar os protocolos NEC, Sony SIRC, RC5 Philips, RC6 Philips .

---

<sup>1</sup> <http://www.righto.com/2009/08/multi-protocol-infrared-remote-library.html>

Com a biblioteca IRremote é fácil de clonar os botões de controles. Basta embarcar no arduino códigos padrões que já veem preestabelecidos na biblioteca e com poucos ajustes teremos mapeados os botões do controle. Através desses processos foi possível clonar os controles de aparelhos de DVD, Televisor e de som.

O problema desse método de clonagem foi não ser possível clonar controles de aparelhos de ar-condicionado. Uma das hipóteses para que esse método não ter funcionado é que ele consegue mapear por inteiro o código do controle. Possivelmente o tamanho da informação de um botão de um controle de ar-condicionado seja maior que de uma TV, por exemplo.

O método de clonagem que funcionou corretamente está mostrada no “ANEXO A”. Após muitas pesquisas, essa forma de clonar foi a que funcionou de forma efetiva. Nesse tutorial são apresentados dois códigos: um utilizado para clonar, outro para enviar ao aparelho de ar-condicionado os comandos do controle.

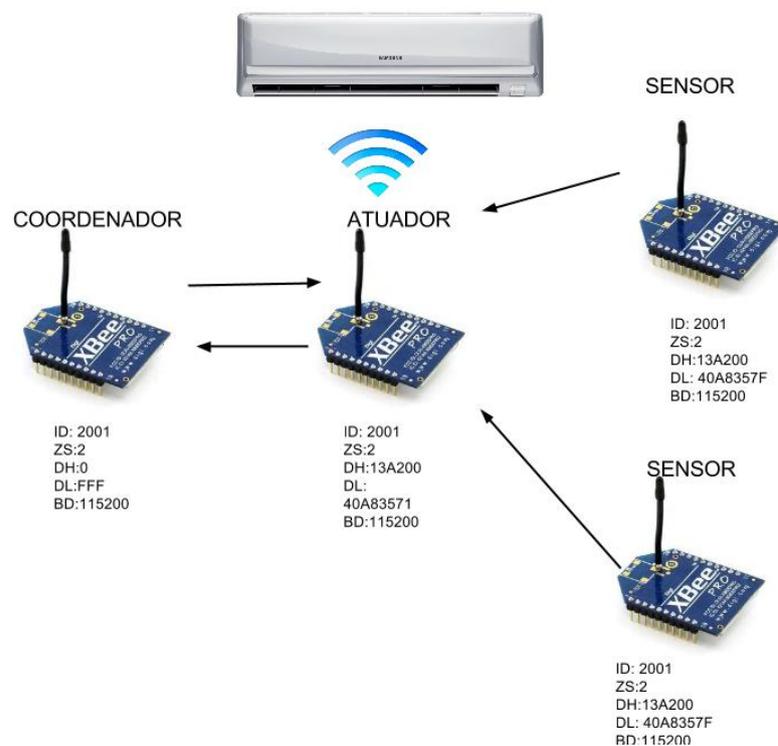
O que torna esse código efetivo, diferente do outro método mencionado anteriormente, é a capacidade de entender por completo os comandos do controle remoto, gerando uma estrutura de mapeamento de cada botão do aparelho. Com isso, torna-se possível o controle de qualquer funcionalidade do controle do ar-condicionado.

Além disso, esse segundo tutorial apresenta um esquema elétrico da utilização de um importante componente, o transistor. Durante esses os testes observamos que apesar de clonado o código, quando enviávamos um comando ao aparelho de condicionador de ar ele não funcionava. A partir da utilização desse componente foi possível controlar o equipamento. Acreditamos que o transistor, devido suas características de funcionamento, consiga deixar tanto o nó atuador como o aparelho de ar numa mesma frequência de comunicação.

### 3.3 Topologia da Rede

Como é mostrada na figura 10 a topologia da RSSF desse trabalho foi configurada da seguinte maneira:

Figura 10 – Topologia da RSSF



Fonte: Elaborado pelo autor

- Cada nó sensor tem os seus pacotes contendo os valores de temperatura/presença destinados ao nó atuador;
- O nó atuador recebe esses pacotes, faz cálculos de média de temperatura e verifica se existe movimento de pessoas na sala. Após essas verificações, ele envia comandos de ajuste de temperatura ou liga/desliga o aparelho condicionador de ar.
- Além de controlar o ar-condicionado, o atuador envia dados contendo informações de cada nó sensor para um servidor, através do nó coordenador (*sink*) possibilitando que a rede seja comandada externamente por um usuário.

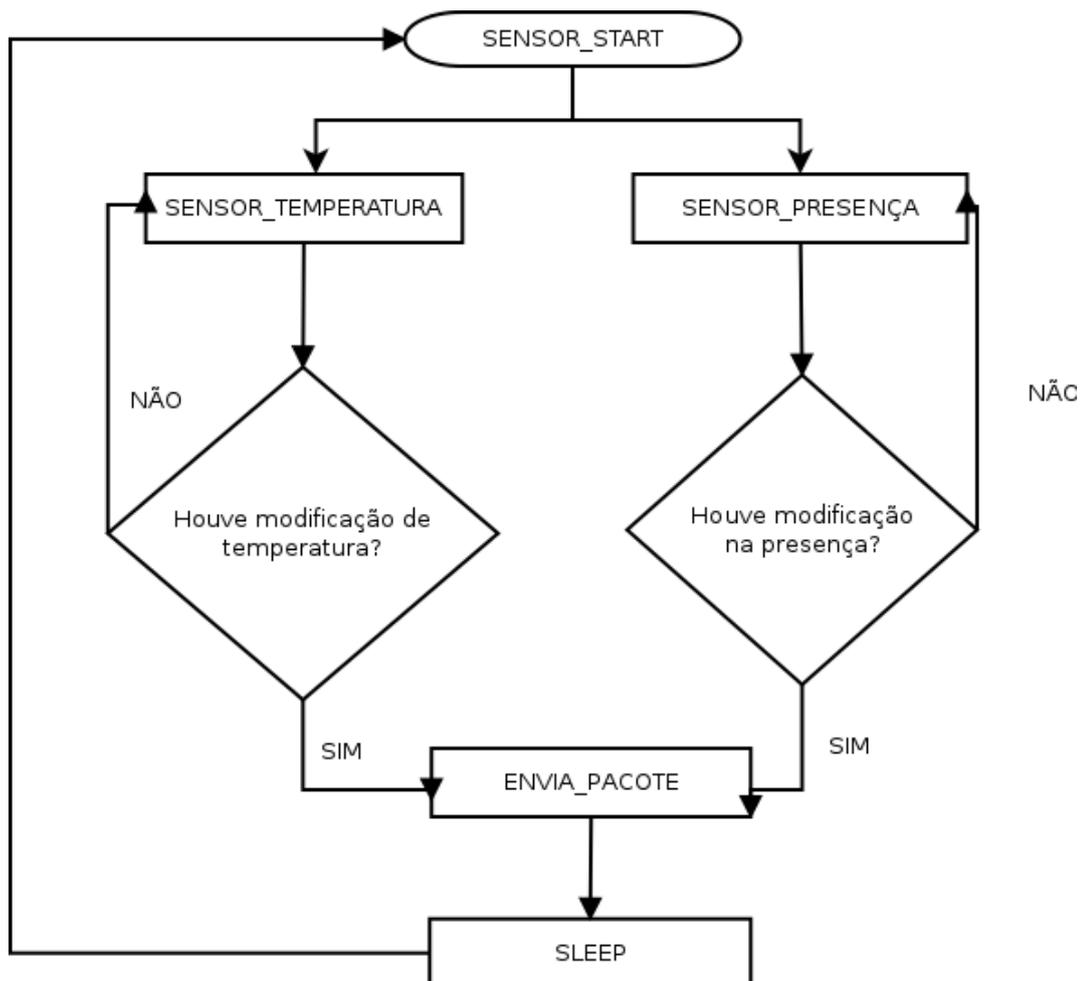
### 3.4 O Software Embarcado

As seguintes seções apresentam os softwares utilizados para programar os nós sensores e nó atuador. A ideia dessas aplicações é testar a utilização dos equipamentos dentro dos requisitos levantados. Acreditamos que muitos ajustes poderão ser realizados para maximizar a performance da rede. A aplicação embarcada é formada pelo código dos nós sensores (3.3.1) e dos nós atuadores (3.3.2).

#### 3.4.1 Código dos Sensores

A rotina de execução do código dos nós sensores (enviaSensor.c) é representada pela figura abaixo (Figura 11).

Figura 11 – Diagrama de Atividades - Nó Sensor



Fonte: Elaborado pelo autor

O nó sensor inicia a sua execução (SENSOR\_START) coletando informações do ambiente e armazenando os dados de cada sensor numa estrutura de dados, como observado no trecho do código abaixo:

```
struct DATA_STRUCTURE{  
    char nome; //o nome identificado do nó  
    char sensor; //o tipo de sensor do nó  
    int valor; // o valor do sensoriamento  
};  
DATA_STRUCTURE pacote;
```

Como o nó sensor é formado por um componente que mede a temperatura e outro que averigua a presença, ele verifica se houve alguma alteração nos valores coletados. Caso exista, é enviado a estrutura para o atuador contendo os dados com informações do sensor: o nome que identifica do nó (identificado com uma letra), o sensor (se é de temperatura ou presença “T” ou “P” respectivamente) e os valores desse sensor. No código representado por:

```
ET.sendData(); //envia os dados dos sensor ao atuador
```

É importante salientar que os valores enviados são do tipo inteiro, inclusive o valor do sensor de presença. No início para detectar o movimento foi pensado em enviar os valores do tipo booleano. Porém, como o atuador funciona calculando médias de valores (será mostrado na próxima seção), quando é detectado um movimento pelo sensor de presença a variável receberá o valor 1000. Com isso, qualquer valor acima de zero, será considerado movimento.

Como observado, a rotina de execução do nó sensor é bem simples e deverá ser, pois num sistema embarcado quanto maior o código maior será o processamento e conseqüentemente o gasto da energia das baterias.

### 3.4.2 Código dos Atuadores

O código do nó atuador é um pouco mais complexo, pois ele fica responsável por tomar decisões como se fosse um controle remoto, baseado nos valores coletados dos nós sensores. A figura 12 representa a rotina de execução do código.

Para iniciar a rede o aparelho de ar-condicionado deverá estar desligado. Quando iniciamos o nó atuador pela primeira vez, ele imediatamente liga o ar-condicionado e atribui a velocidade de resfriamento para o número 16 (dezesseis).

Foram criadas duas variáveis, uma guarda o estado atual de funcionamento e a outra a velocidade de resfriamento do aparelho:

```
boolean ligDes
//variável estado do aparelho ligado ou desligado
int numTemp = 0
// variável que informa o comando atual da temperatura.
```

No atuador também é mantida a mesma estrutura de dados para acessar os valores que são enviados pelos sensores. Os pacotes chegam no nó atuador através do código:

```
ET.receiveData()
```

É verificado dentro de uma janela de tempo qual o tipo de sensoriamento que está chegando no atuador.

```
while( (millis() - tempoMillis) < intervalo) { //janela de tempo
  if(ET.receiveData()) //verifica se chega informação da serial de comunicação
  {
    if(pacote.sensor == 'T' && pacote.valor >= 0 ){// verificação se é de temperatura
      //Serial.println("Chegou temperatura");
      temp = pacote.valor;
      soma = soma + temp; //soma os valores recebidos
      cnt++; //contador de valores recebidos
    }
    else if(pacote.sensor == 'P' && pacote.valor >= 0 ) { } // verificação se é de presença
    //Serial.println("Chegou se tem alguém");
```

```

tempP = (float) pacote.valor;
somaP = somaP + tempP;
cntP++;
}
}

```

De posse dos valores de temperatura dos sensores, o atuador calcula a média desses valores. Com essa média o algoritmo determinar qual o comando de resfriamento certo deverá ser enviado para mudar a temperatura:

```

if (media <= 20 && numTemp == 16) {

    vinteTres(); //comando de número 23
    numTemp = 23;

} else if (media > 23 && numTemp > 16){
    dezesseis();
    numTemp = 16; //comando de número 16
}

```

O código acima tem o seguinte comportamento: se a média da temperatura for menor que 20 °C e o comando que regula a temperatura for igual a 16, então o aparelho começa a resfriar o ambiente a 23 °C. Ao contrário, se a temperatura for maior que 23 °C e o comando for maior que 16 o aparelho volta a resfriar a sala à 16 °C.

Para verificar se existe ou não movimento no local monitorado, como foi dito na seção anterior (3.3.1), o atuador calcula a média dos valores de presença, e então é feita a seguinte lógica:

```

if(mediaP == 0.0 && ligDes == true){
    tempoMillis = millis();// recebe o tempo que a função millis(); foi chamada
    ligDes = false;

    Serial.print("O aparelho esta desligado");

    desligando(); //comando desliga
    delay(500);

}

if((mediaP > 0.0 && ligDes == false) && millis() - tempoMillis > intervalo ){

//if(mediaP>0 && millis() - lowIn > intervalo){ }&&millis() - lowIn > intervalo

```

```

Serial.print("O aparelho esta ligado");
  ligDes = true;
  liga();
  delay(500);

}

```

Se a média dos valores de presença for igual a zero e se o aparelho estiver ligado, é enviado um comando desligando o ar-condicionado. Caso contrário se a média for maior que zero e o aparelho desligado, é enviado um comando para ligar o condicionador de ar.

Após feitas essas verificações os dados são enviados ao nó *sink* da rede que repassa essas informações ao servidor no formato JSON:

```

void sendJSON(char nome, char sensor, int valor){

  Serial1.print("{\"i\": ");
  Serial1.print("\");
  Serial1.print(nome);
  Serial1.print("\");
  Serial1.print(",\"s\": ");
  Serial1.print("\");
  Serial1.print(sensor);
  Serial1.print("\");
  Serial1.print(",\"d\": ");
  Serial1.print(valor);
  Serial1.print("}\n");
}

```

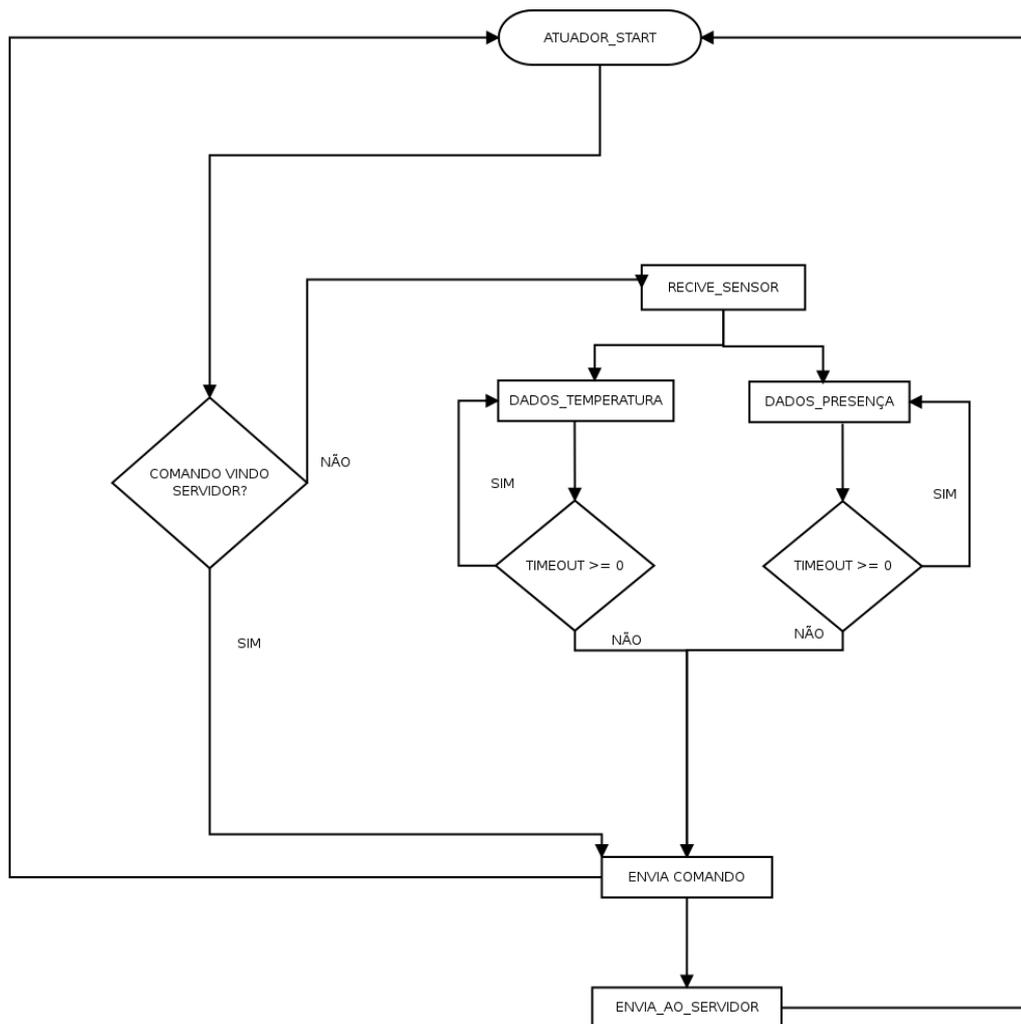
As funções que representam os comandos passados ao ar-condicionado guardam os códigos do controle do ar-condicionado, que foram obtidos através do processo de clonagem. Esse procedimento é detalhado no “Anexo B” desse trabalho.

A figura abaixo (Figura 12) ilustra a rotina de execução do código do atuador. No início do loop é verificado se os pedidos de manipulação do ar-condicionado são enviados do servidor, se essa condição for atendida o nodo atuador envia comando de ajustes ao aparelho de ar-condicionado.

Caso isso não aconteça, o atuador verifica os pacotes que estão chegando na sua porta serial e faz os cálculos de presença e temperatura. A partir desses dados, são tomadas as decisões de quais comandos enviar ao ar-condicionado.

Nesse trabalho, foram mapiados somente alguns comandos: de temperatura e de ligar/desligar. Entretanto com o método de clonagem é possível mapiar qualquer botão do controle remoto, como por exemplo velocidade do vento ou ativação e desativação do swap

Figura 12 – Diagrama de Atividades – Nó atuador



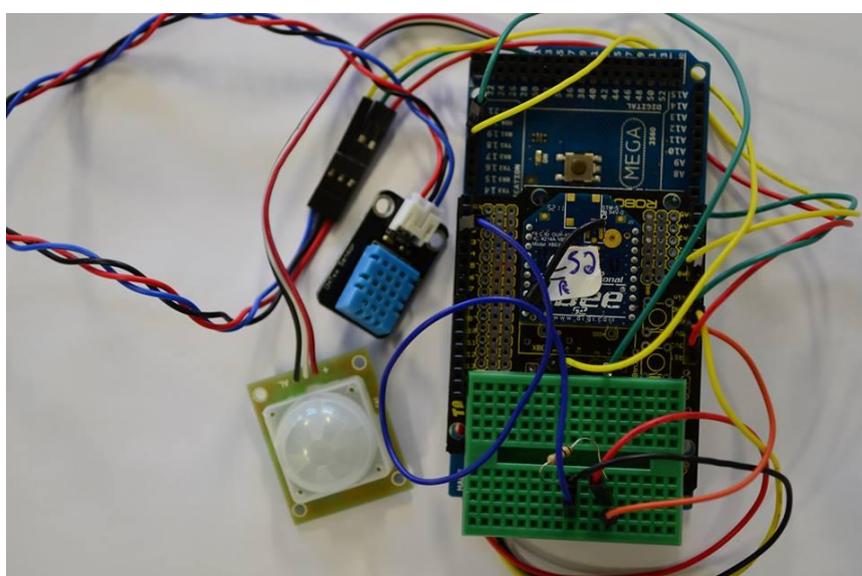
Fonte: Elaborado pelo autor

## 3.5 O Hardware

### 3.5.1 Nó Sensor

Como já citado nesse trabalho, o nó sensor tem a capacidade de medir a temperatura em graus Celsius ( $^{\circ}\text{C}$ ) e a presença de movimento. Abaixo, temos a imagem do sensor.

Figura 13 – Nó Sensor



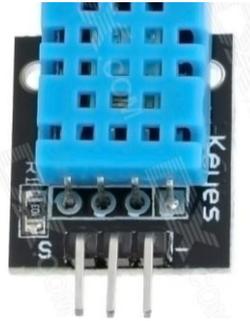
Fonte: Elaborado pelo autor

Este sensor é formado pelos seguintes componentes:

- 1 Arduino Mega 2560
- 1 XBee Series 2
- 1 sensor de temperatura/umidade DHT11
- 1 senso de presença PIR

O DHT11 (figura 14) é um sensor de umidade relativa e temperatura, com saída digital calibrada. Possui uma exclusiva tecnologia para medir a umidade, garantindo a confiabilidade e estabilidade. Possui internamente um microcontrolador de 8 bits para tratar o sinal. Possui tamanho compacto, baixo consumo, encapsulamento simples com apenas quatro terminais, podendo transmitir o sinal por até 20 metros. Todas essas características permitem utilizar esse sensor nas mais diversas aplicações. (D-ROBOTICS, 2010)

Figura 14 – Sensor de Temperatura/Umidade –DHT11



Fonte: D-ROBOTICS, 2010

Os sensor PIR (figura 15) permitem detectar o movimento, quase sempre usado para detectar se um ser humano se moveu dentro ou fora da faixa de sensores. Eles são pequenos, de baixo custo, baixo consumo de energia, fácil de usar. Eles são muitas vezes referidos como PIR, "infravermelho passivo", ou sensores de movimento "IR" . (PARALLAX, 2007)

Figura 15 – Sensor de Presença - PIR

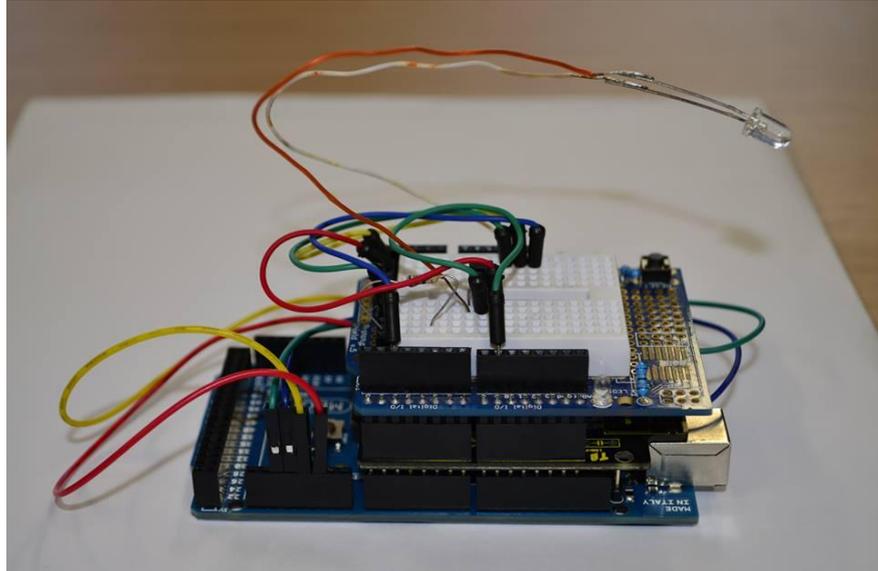


Fonte: PARALLAX, 2007

### 3.5.2 Nó Atuador

Na figura a seguir (figura 16) é mostrado o nó atuador da nossa rede:

Figura 16 – Nó Atuador



Fonte: Elaborado pelo autor

Ele é formado basicamente por:

- 1 Arduino Mega 2560
- 1 XBee Series 2
- LED infravermelho
- Transistor BC818

O emissor de infravermelho (figura 17) é usado para transmitir sinais de infravermelhos. Um LED infravermelho é como qualquer outro LED. Podemos usar o emissor não só para transmitir dados ou comandos, mas também para emular controles remotos para controlar qualquer aparelho que tenha um receptor infravermelho.

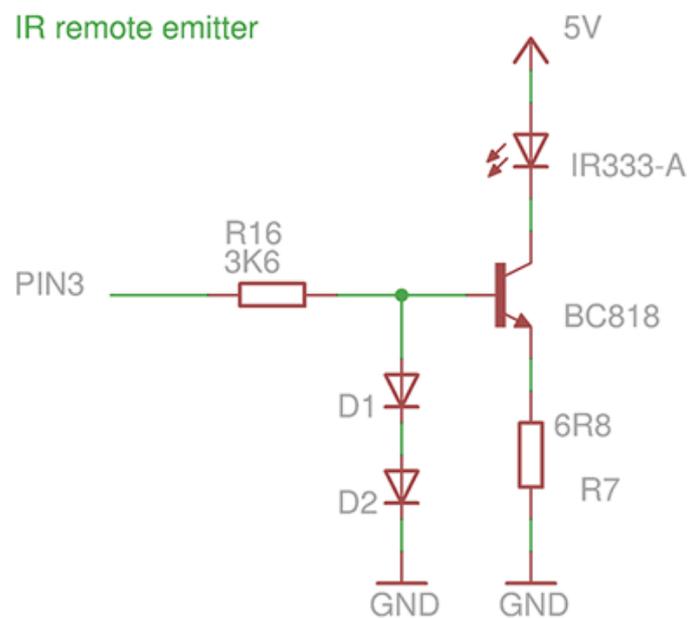
Figura 17 – LED IR



Fonte: HACKS, 2013

Outro componente importante para o funcionamento do atuador é o transistor do tipo BC818 ele é responsável por deixar a frequência do sinal infravermelho perceptível pelo ar-condicionado. Abaixo é mostrado o esquema de montagem do circuito.

Figura 18 – Esquema do circuito com o transistor



Fonte: HACKS, 2013

## 4 ESTUDO DE CASO

### 4.1 Ambientes Monitorados

Os ambientes monitorados foram os laboratórios do bloco 01 (um) da UFC do Campus Quixadá. Todos os quatro laboratórios são equipados com o mesmo modelo, ar-condicionado da marca GREE Split, do modelo GTH36D1B1, de 36000BTU/h, com a potência de 4000W, no total de oito aparelhos.

A implantação do sistema foi feita no laboratório 04, onde tínhamos o maior controle da segurança dos equipamentos e poderíamos fazer ajustes nos dispositivos sem atrapalhar o fluxo normal das aulas. Nos demais laboratórios (01, 02 e 03), durante o período de desenvolvimento, foram feitas verificações da rotina de uso dos condicionadores de ar, ou seja, em que horas eram ligados e em quanto tempo eles demoravam a ser desligados.

Os testes realizados no laboratório 04 foram realizados em dois dias, após verificarmos o funcionamento mais efetivo da rede.

### 4.2 Cálculos de Consumo

Durante a verificação de rotina realizada na UFC em Quixadá, foi observado que os aparelhos ficam ligados desnecessariamente, em média, de 15 a 30 minutos, entre os intervalos de aulas, os horários de refeições e o final das aulas.

Foi utilizado um cálculo simples para estimação do consumo médio dos aparelhos elétricos. O valor a ser pago às empresas distribuidoras de energia elétrica corresponde a quantidade de energia elétrica consumida, definida em kWh multiplicada por um valor unitário denominado tarifa, medida em R\$/kWh (reais por quilowatt-hora), que corresponde ao preço de um quilowatt consumido em uma hora. A fórmula (COELCE, 2012) é descrita a seguir (Equação 01 e 02):

$$\text{kWh} = (\text{Potência do Equipamento} * \text{N}^\circ \text{ horas utilizadas} * \text{N}^\circ \text{ uso por mês}) / 1000 \quad (01)$$

$$\text{Total a pagar} = \text{kWh} * \text{tarifa} \quad (02)$$

Utilizando a Equação 1, um aparelho do modelo utilizado nos testes, ligado em média por 12 horas diárias (correspondente aos horários de aulas nos três turnos na

universidade), pode-se estimar um consumo total de 1056,0 kWh/mês, resultando em um gasto mensal de R\$ 517,44, levando em consideração uma tarifa de R\$ 0,49302/kWh (COELCE, 2012).

Se for levado em consideração que os aparelhos fiquem ligados 15 minutos a mais após o final das aulas antes de serem desligados, pode-se estimar que o consumo aumenta para 1122 kWh/mês. Usando a mesma tarifa, o valor pago no final de um mês é de R\$ 553,16, uma diferença de R\$ 35,72 reais a mais por aparelho na conta de luz.

Se for utilizado o tempo de 30 minutos (em vez de 15 minutos) por dia, os gastos com energia sobem significativamente, pois o consumo por mês torna-se 1188 kWh/mês com um gasto mensal de R\$ 585,70 gerando uma diferença de 68,26 a mais na conta de luz.

Multiplicamos os valores de consumo dos períodos que ficam ligados em excesso pelo número de ar-condicionado dos laboratórios e salas de aulas dos dois blocos do campus que são no total de trinta e dois (32). Verificamos, portanto, que são gastos respectivamente no tempo de 15 min o valor de R\$ 1.143,04 e com o tempo de 30 min o preço pago com energia salta para R\$ 2.184,32 a mais na conta de energia.

Com o sistema implantado em todos os laboratórios e salas de aula existentes hoje no campus, poderemos chegar o mais próximo de um comportamento onde os equipamentos condicionadores de ar seriam desligados quase que imediatamente, ou seja, eles consumiriam aproximadamente 1056 kWh/mês (por aparelho). Com isso, reduziria o gasto excedente podendo em um ano, a instituição ter uma economia estimada de R\$ 26.211, 84 que poderia ser gasto em outras atividades até mesmo na compra dos equipamentos para a instalação do sistema.

Tabela 6 - Consumo energia aparelhos 15 e 30 minutos.

Tempo excedente em minutos (min)	Consumo kWh/mês por aparelho	Valor gasto/Aparelho excedente em reais (R\$)	Gasto de desperdício de 32 aparelhos em reais (R\$)	Gasto excedente por ano com desperdício de energia dos 32 aparelhos

15	1.122 kWh/mês	35,72	1.143,04	13.716,48
30	1188 kWh/mês	68,26	2.184,32	26.211,84

Fonte: Elaborado pelo autor

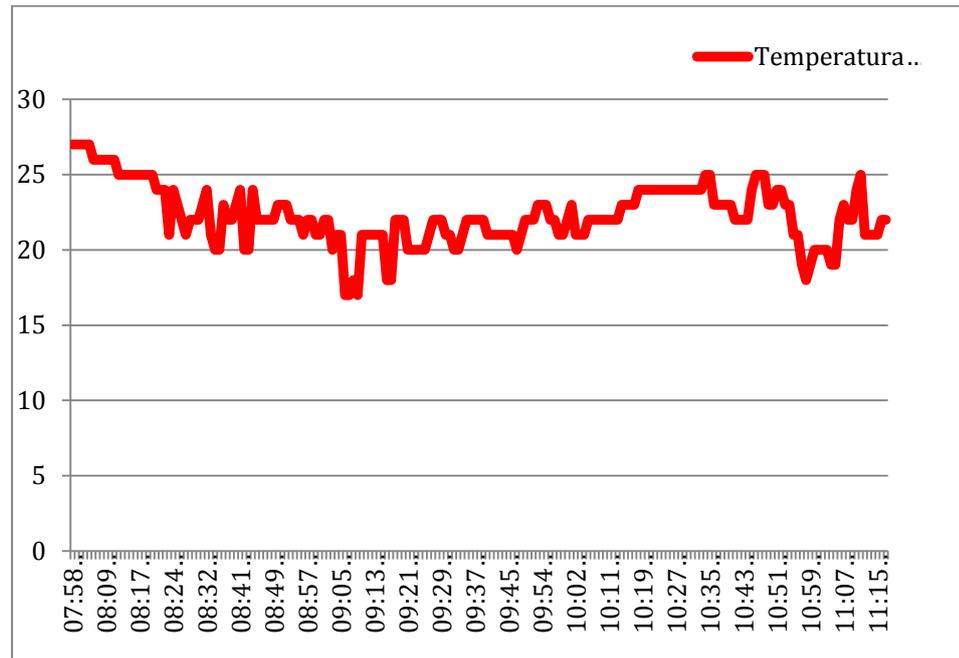
Deve-se observar que os cálculos são estimativas, pois o comportamento do uso dos aparelhos de ar-condicionado no campus é variável, pois há dias que não há aulas de determinadas disciplinas ou os equipamentos são desligados imediatamente após as atividades nas salas. Porém, com o crescimento previsto para o campus, será mais difícil manter o uso adequado dos recursos de laboratórios e salas de aulas, tornando primordial a automação do gerenciamento dos recursos.

### 4.3 Testes

Antes de deixar a rede em pleno funcionamento foram feitos alguns testes para aperfeiçoar a execução do sistema, tais como: a verificação da calibragem desejada dos sensores, as janelas de tempo em que os pacotes deveria chegar no nó atuador, o tempo de envio entre um comando e outro. Verificou-se que a rede funcionou de forma satisfatória, ligando/desligando, aumentando ou diminuindo a velocidade, atendendo as expectativas, mas dentro de limitações que eram previstas.

O teste final foi iniciado às 7 horas do dia 18 de dezembro e finalizados às 22 horas do dia seguinte, perfazendo o total de 39 horas. No gráfico abaixo (gráfico 01) é mostrado um trecho dos dados coletados da porta serial do nó atuador, que corresponde a média de temperatura dos sensores no período de 8 à 11 horas e 15 minutos do dia 19 de dezembro de 2013.

Gráfico 1 - Média de Temperatura no Lab04.

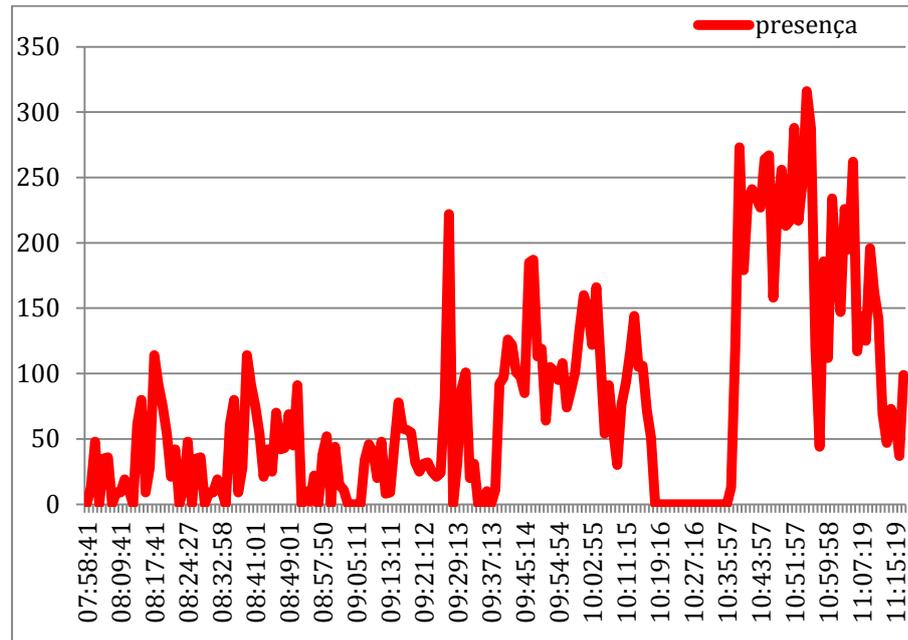


Fonte: Elaborado pelo autor

Como podemos observar, a rede conseguiu manter o padrão de temperatura que foi estabelecido (entre 20 e 23°C) através das trocas automáticas do modo de resfriamento do ar-condicionado. Ainda podemos verificar um período de aumento da temperatura por volta de 10:15 h até 10:35 h ficando numa média de 24 a 26 °C sendo um momento em que o aparelho de ar-condicionado manteve-se desligado, justificando esse aumento.

O próximo gráfico (gráfico 02) representa os dados de presença. Como explicado na seção 3.3 (O Software Embarcado) desse trabalho, qualquer valor maior que zero significa que houve movimento no ambiente. Podemos observar no gráfico que sensor de presença funcionou, no entanto como será melhor detalhado, ele apresenta limitações na distância de sensoriamento.

Gráfico 2 - Valores de presença no Lab04.



Fonte: Elaborado pelo autor

Nesse gráfico (02) o período de monitoramento é o mesmo do gráfico anterior (01). Podemos observar que na mesma faixa de tempo em que houve um aumento de temperatura (de 10:15 às 10:35 h) temos o valor de zero de presença, significando que nesse momento não existia movimento no laboratório e os aparelhos de ar-condicionado estava desligado.

#### 4.4 Limitações

Apesar de apresentar um funcionamento satisfatório a rede apresenta suas limitações que serão elencadas:

1. **O sensor de presença:** os componentes comprados para verificar a presença funcionaram razoavelmente bem. Entretanto, não apresentaram um bom alcance e assim, para garantir uma boa cobertura, com esse modelo de sensor, deverá ter um número maior de sensores.

Para resolver esse problema, existe outra versão semelhante desse componente que promete ter um alcance de 7m, sendo uma possível solução.

2. **Baterias:** As baterias disponíveis para os experimento eram baterias de 9V tensão e com 400 mAh de corrente. Quando conectadas ao sensores a corrente elétrica das baterias eram muito baixas para alimentar os dispositivos.

**3. Comunicação atuador/ar-condicionado:** não é possível garantir que os comando enviados pelo atuador foram recebidos ou não pelo ar-condicionado, pois não existe uma forma de confirmação por parte do aparelho condicionador de ar.

## **5 CONCLUSÕES**

Nesse trabalho verificamos o problema de desperdício de energia em função dos aparelhos de ar condicionado ficarem ligados sem necessidade, nos ambientes de contexto de universidades (e semelhantes) utilizando o campus da UFC de Quixadá como estudo de caso.

Propomos então desenvolver um sistema baseado nos princípios das Redes de Sensores Sem Fio, para controle desses equipamentos de forma autônoma e eficaz. Os objetivos levantados por esse trabalho, foram concluídos satisfatoriamente, mesmo com a suas limitações.

A prototipação dos nós sensores e atuadores foi desenvolvida conforme o planejado, através da plataforma Arduino tendo a sua comunicação entre os dispositivos da rede com os módulos XBee. Os firmwares dos protótipos da rede foram implementados com êxito, possibilitando os nós sensores coletar parâmetros de temperatura e presença do ambiente. Além disso, o atuador pôde manipular os aparelhos de ar-condicionado, trocando a velocidade de resfriamento, ligado ou desligando, através do infravermelho, sendo essa a grande novidade em relação aos trabalhos relacionados. Algumas ideias para melhorar o projeto serão introduzidas a seguir.

### **5.1 Trabalhos Futuros**

A partir do estudo realizado nesse trabalho outras possibilidades podem ser exploradas. O projeto permite a construção de outras ideias devido ao uso de códigos e hardware abertos.

Alguns trabalhos futuros serão importantes para incrementar o trabalho. O primeiro será realiza estimativas em mais salas por mais tempo, para usar cálculos estáticos, a partir de uma base maior de dados. Outro ponto será melhorar a detecção de presença com troca de equipamento da mesma espécie de melhor qualidade, ou a utilização de uma nova forma de detecção como a utilização de câmeras por exemplo.

Um trabalho futuro de grande importância será implementação de um servidor do tipo web. Essa aplicação irá permitir que os usuários credenciando controlem a rede e

modifiquem parâmetro de resfriamento. Além disso, verificar se os aparelhos de ar-condicionado estão ligados ou desligados, dando maior confiabilidade à rede.

Mais uma atividade de importante será a comunicação entre várias redes de sensores. Com essa comunicação será possível enviar um comando para uma determinada rede utilizando outras redes intermediárias.

Além desses trabalho outro de importante valor será a utilização da rede no controle de vários aparelhos de ar-condicionado. Em muitos ambientes é normal a utilização de mais de um desses equipamentos para o resfriamento do ambiente.

Também será posteriormente necessário fazer um modelo de energia do consumo de energia individualmente por sensor da rede. A partir da verificação da quantidade de pacotes que a rede envia, será possível determinar quando o quanto de energia o rádio dos sensores gastam para enviar e receber informações.

Outra possibilidade futura, será a inclusão de novas funcionalidades de automação aproveitando a tecnologia de infravermelho, como por exemplo controlar num ambiente de ensino datashow, aparelhos de som, televisores, entre outros.

## REFERÊNCIAS

ALVES, J. A.; MOTA, J.. **Casas Inteligentes**. 1ª Edição Lisboa: Centro Atlântico, 2003.

AMERICAN SOCIETY OF HEATING, REFRIGERATING AND AIR-CONDITIONING ENGINEERS. **ASHRAE 55: Thermal Environmental Conditions for Human Occupancy**. Atlanta: American Society Of Heating, 2004.

AMERICAN SOCIETY OF HEATING, REFRIGERATING AND AIR-CONDITIONING ENGINEERS. **ASHRAE 62: Ventilation for Acceptable Indoor Air Quality**. Atlanta: American Society Of Heating, 2003.

ARAÚJO, R. B. de. **Computação Ubíqua: Princípios, Tecnologias e Desafios**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003, Natal. Anais... Brasília: Sociedade Brasileira de Computação (sbc), 2003. v. 8, p. 45 – 115

ARDUINO. Disponível em: < <http://arduino.cc/> >. Acesso em: 06 jul. 2013.

ATEMEL. Disponível em: < <http://www.atmel.com/> >. Acesso em: 06 jul. de 2013

BRAGA, Anísio R. et al. **Aplicação de redes de sensores sem fio para o monitoramento de variáveis ambientais**. Disponível em:

<[http://www.cpdee.ufmg.br/documentos/PublicacoesDefesas/904/CBEE\\_2009\\_final.pdf](http://www.cpdee.ufmg.br/documentos/PublicacoesDefesas/904/CBEE_2009_final.pdf)>.

Acesso em: 08 jul. de 2013.

COELCE. **Por dentro da conta de luz da COELCE. 2007**. Disponível em:

<[http://www.aneel.gov.br/arquivos/PDF/Cartilha\\_COELCE2.pdf](http://www.aneel.gov.br/arquivos/PDF/Cartilha_COELCE2.pdf)>. Acesso em: 13 nov. 2013.

CUNHA, A. F.. **O que são sistemas embarcados?** Disponível em:

<<http://www.sabereletronica.com.br/secoes/leitura/274>>. Acesso em: 04 jan. 2008.

CUNHA, F. D. et al. **Uma Nova Abordagem para Acesso ao Meio em Redes de Sensores Sem Fio**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 31., 2013, Brasília. Anais Trilha Principal e Salão de Ferramentas. Porto Alegre: Sociedade Brasileira de Computação (sbc), 2013. p. 01 - 16.

CUNHA, T. F., **Controle centralizado de equipamentos de ar condicionado via rede sem fio ZigBee**, 2013, Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Instituto Federal de Santa Catarina, São José

D-ROBOTICS. **DHT11 Humidity & Temperature Sensor**. 2010. Disponível em: <<http://www.micro4you.com/files/sensor/DHT11.pdf>>. Acesso em: 12 nov. 2013.

DEFINITIONS. Disponível em: <<http://www.ibuilding.gr/definitions.html>>. Acesso em: 06 jul. de 2013.

DISPOSIÇÕES gerais. Disponível em: <[http://portal.mte.gov.br/data/files/FF8080812BE914E6012BEF0F7810232C/nr\\_01\\_at.pdf](http://portal.mte.gov.br/data/files/FF8080812BE914E6012BEF0F7810232C/nr_01_at.pdf)>. Acesso em: 07 jul. 2013

ERGONOMIA. Disponível em: <[http://portal.mte.gov.br/data/files/FF8080812BE914E6012BEFBAD7064803/nr\\_17.pdf](http://portal.mte.gov.br/data/files/FF8080812BE914E6012BEFBAD7064803/nr_17.pdf)>. Acesso em: 07 de jul. 2013

ESCHNER, R. H; **Sistema de Automação Residencial Baseado em Sensores ZigBee**, 2011, Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

FALUDI, Robert. **Building Wireless Sensor Networks**. Gravenstein Highway North: O'reilly Media, 2011. 300 p

HACKS, Cooking. **Control your Infrared Devices from the Internet with IR Remote (Arduino / Raspberry Pi Compatible)**. 2013. Disponível em: <<http://www.cooking-hacks.com/documentation/tutorials/control-hvac-infrared-devices-from-the-internet-with-ir-remote/#ir-remote-arduino>>. Acesso em: 12 nov. 2013.

INSTITUTE, **Intelligent Building. Intelligent Buildings Definitions**. Disponível em: <<http://www.ibuilding.gr/definitions.html>>. Acesso em: 10 maio 2013.

MELLIS, D. A. **Arduino - Introduction**. Arduino, 2009. Disponível em: <<http://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 07 jul. 2013.

MESHNETCS. Disponível em: <<http://www.meshnetics.com/zigbee-modules/>>. Acesso em: 06 jul. de 2013.

MORIMOTO, P. C. E.. **Entendendo os sistemas embarcados**. Disponível em: <<http://www.hardware.com.br/artigos/entendendo-sistemas-embarcados/>>. Acesso em: 29 de maio 2013.

PARALLAX. Datasheet PIR. 2007. Disponível em: <<http://content.solarbotics.com/products/datasheets/pirsensor-v1.2.pdf>>. Acesso em: 12 nov. 2013.

PINHEIRO, J. M. S. **Sistemas de Automação**. Disponível em: <[http://www.projetoderedes.com.br/artigos/artigo\\_sistemas\\_automacao.php](http://www.projetoderedes.com.br/artigos/artigo_sistemas_automacao.php)>. Acesso em: 19 maio 2013.

QUEIROZ, R. B.; AZEVEDO, R. C. A. **Rede de Sensores Sem Fio Para Automação Predial com Módulos Meshbean**. 2009. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Faculdade de Tecnologia, Universidade de Brasília, Brasília.

ROCHA, A. R.; PIRMEZ, L.; DELICATO, F. C.; LEMOS, E.; SANTOS, I.; GOMES, D. G. SOUZA, J. N. S., **WSNs clustering based on semantic neighborhood relationships**, Elsevier: Computer Networks, v. 56, n.5, p. 1627-1645, março, 2012

ROCHA, J. W. V. **Rede de Sensores Sem Fio**. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialrssf/default.asp>>. Acesso em: 11 jul. 2013.

RUIZ, L. B.; CORREIA, L. H. A.; VIEIRA, L. F.; MACEDO, D. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S.; VIEIRA, M. A.; BECHELANE, E. H.; LOUREIRO, A. F.; NOGUEIRA, J. M. **Arquiteturas para Redes de Sensores Sem Fio**. In: Minicursos do 22º Simpósio Brasileiro de Redes de Computadores. 1 ed. Gramado, RS : SBC, 2004, v.1, p.167-218.

SALEIRO, M.; EY, E. **ZigBee: Uma Abordagem Prática**. Disponível em <[http://lusorobotica.com/ficheiros/Introducao\\_ao\\_Zigbee\\_-\\_por\\_msaleiro.pdf](http://lusorobotica.com/ficheiros/Introducao_ao_Zigbee_-_por_msaleiro.pdf)> Acesso em: 11 de jul. de 2013.

SALEIRO, Mário; EY, Emanuel. **ZigBee: uma abordagem prática**. 2009. Disponível em: <[http://lusorobotica.com/ficheiros/Introducao\\_ao\\_Zigbee\\_-\\_por\\_msaleiro.pdf](http://lusorobotica.com/ficheiros/Introducao_ao_Zigbee_-_por_msaleiro.pdf)>. Acesso em: 12 nov. 2013

SOLDA, Rogério; CASTRO, César Ortolani Marcondes de; ASSIS, Wânderson de Oliveira. **SIMPLIFICANDO O USO DE MÓDULOS COM TECNOLOGIA ZIGBEETM EM APLICAÇÕES DE COMUNICAÇÃO WIRELESS E NA LOCALIZAÇÃO DE DISPOSITIVOS MÓVEIS**. Disponível em: <<http://www.maua.br/pesquisas/simplificando-o-uso-de-modulos-com-tecnologia-zigbeetm>>. Acesso em: 15 out. 2013.

SEIDEL, C.; FERREIRA, F.; OLIVEIRA, E. **Aplicação de Redes de Sensores Sem Fio(RSSF) para Engenharia Ambiental**. International Conference on Engineering and Computer Education – ICECE, Santos, SP, Brasil, 11 a 14 de março de 2007. Disponível em:<[http://www.ic.uff.br/~eoliveira/Publicacoes/ICECE\\_07b.pdf](http://www.ic.uff.br/~eoliveira/Publicacoes/ICECE_07b.pdf)>. Acesso em: 28 de julho de 2009. SIEMEN

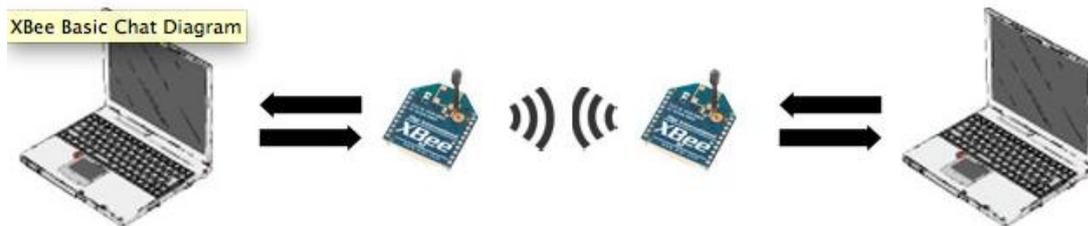
## ANEXOS

**ANEXO A – Tutorial Configurações Básicas XBee – CHAT**

Fonte: <http://examples.digi.com/get-started/basic-xbee-zb-zigbee-chat/>

**1. Introdução**

Esse tutorial tem o objetivo de mostrar as configurações básicas dos dispositivos da rede ZigBee mostrando como exemplo um bate-papo entre os dois computadores onde mandam mensagem de um para o outro utilizando conexão sem fio usando comunicação serial.

**2. Componentes**

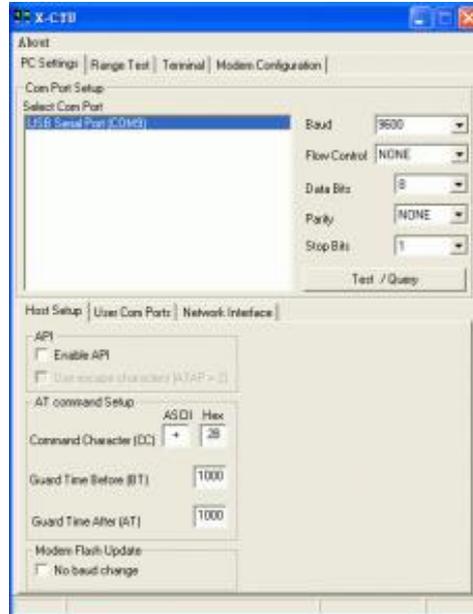
- 1 XBee ZB ZigBee Radio (Series 2);
- 1 XBee ZB ZigBee Radio (Series 2);
- 2 XBee Explorer USB
- 2 Computadores executando programa de terminal serial

Atenção: em cada XBee Series 2 tem um endereço de número de série de 64 bits impresso na parte de trás. O início ou "alto" parte do endereço será 0013A200, intervalo pré-definido de Digi do espaço de endereços. O último ou "baixa" parte do endereço será diferente para cada rádio. Para o rádio, à direita, é 403B9E21. Anote esses números, indicando que é o coordenador e que é o roteador para que você possa consultá-las enquanto você estiver trabalhando com eles.

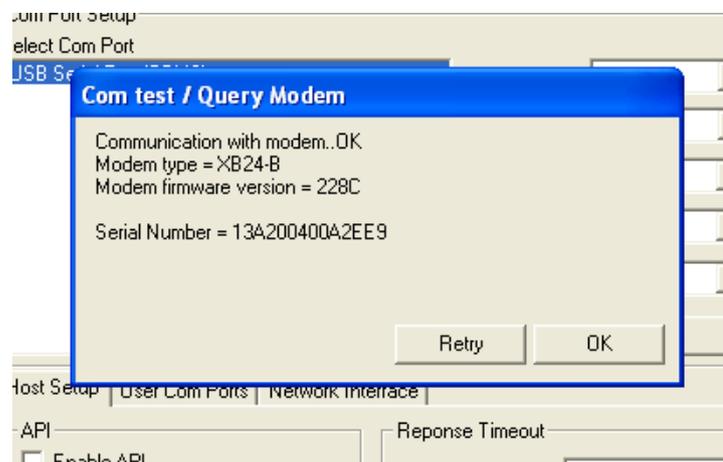


### 3. Configurando o coordenador

- Para configurar o dispositivo como coordenador abra o programa X-CTU insira no XBee Explorer e conecte-o no computador.



- Abriu X-CTU. Selecione uma porta COM e clique em "Teste / Query" para ver se X-CTU pode se comunicar com o rádio.



- Se o teste der certo, clique na aba Modem Configuration. Caso contrário, verifique a velocidade da conexão serial, por padrão a velocidade é 9600 (Baud).
- No Mode Configuraton, clique no botão READ para carregar as configurações atuais do modem.
- Em Function Set escolha a opção: Coordenador AT.
- Configure o Coordenador com os comando abaixo:

Função	Comando	Parâmetro
PAN ID	ATID	2001 (qualquer endereço de 0 a FFFE vai fazer)
Endereço de destino alta	ATDH	0013A200
Endereço de destino baixo	ATDL	[Ver o seu endereço registrado Router]

- Após configurar clique em WRITE para carregar essas configurações no dispositivo.

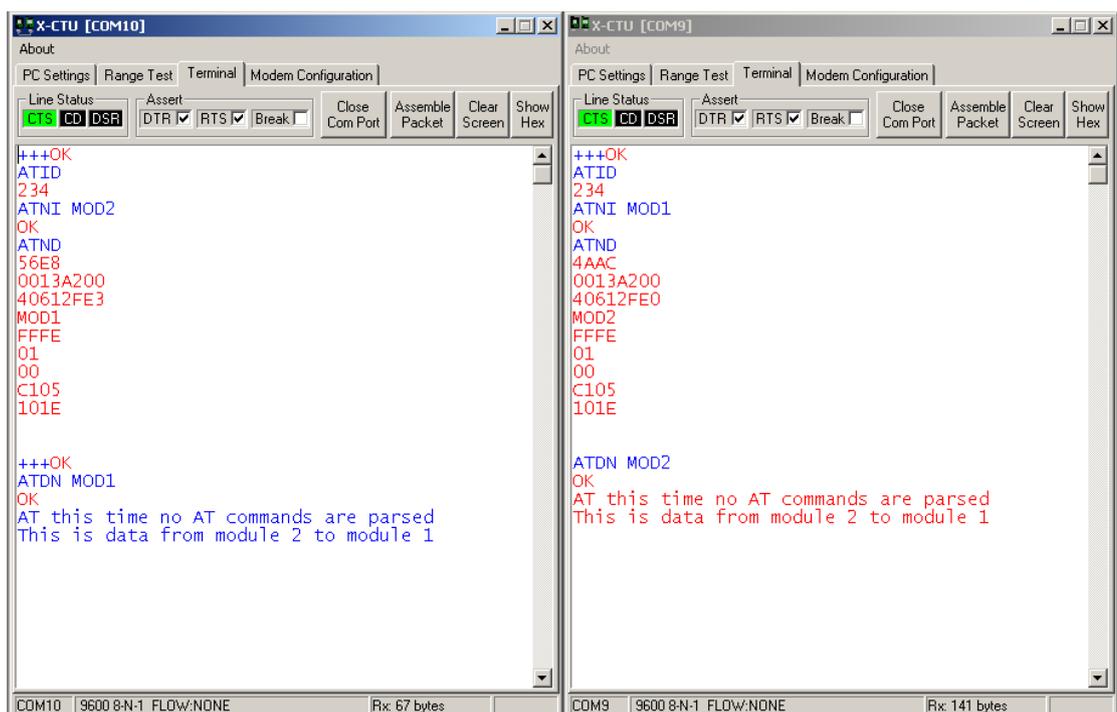
#### 4. Configurando o Roteador

- Faça o mesmo procedimento de configuração que foi utilizado no coordenador, com a diferença de escolher em Function Set a opção: Router AT. Além disso, o endereço de destino para o rádio coordenador.

Função	Comando	Parâmetro
PAN ID	ATID	2001 (qualquer endereço de 0 a FFFE vai fazer)
Endereço de destino alta	ATDH	0013A200
Endereço de destino baixo	ATDL	[Ver o seu endereço registrado Router]

#### 5. Testando

- Ligue cada rádio em um XBee Explorer, e conecte-os nas USBs dos computadores em lugares diferentes na sala. Em cada computador execute o X-CTU e na aba Terminal, digite e perceba o funcionamento do bate-papo.



## ANEXO B – Tutorial Clonagem Controle Remoto Com Arduino

Fonte: <https://arduinolivres.wordpress.com/2012/07/31/clonando-qualquer-controle-remoto/>

Componentes utilizados.

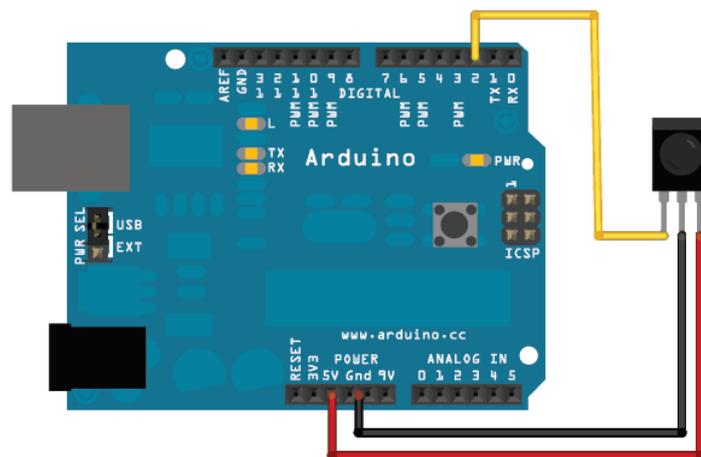
1-Receptor Infravermelho:



2-Led Infravermelho:



Vamos dividir em duas etapas. Na primeira etapa, vamos receber o código do Controle Remoto que você quer clonar. Para isso monte o seguinte esquema com o Receptor Infravermelho:



Depois de montado, abra o IDE do Arduino e cole o seguinte código:

```
/* Raw IR decoder sketch!
```

This sketch/program uses the Arduino and a PNA4602 to decode IR received. This can be used to make a IR receiver (by looking for a particular code) or transmitter (by pulsing an IR LED at ~38KHz for the durations detected

Code is public domain, check out [www.ladyada.net](http://www.ladyada.net) and [adafruit.com](http://adafruit.com) for more tutorials!

```
*/
```

```
// We need to use the 'raw' pin reading methods
// because timing is very important here and the digitalWrite()
// procedure is slower!
//uint8_t IRpin = 2;
// Digital pin #2 is the same as Pin D2 see
// http://arduino.cc/en/Hacking/PinMapping168 for the 'raw' pin mapping
#define IRpin_PIN PIND
#define IRpin 2

// the maximum pulse we'll listen for - 65 milliseconds is a long time
#define MAXPULSE 65000

// what our timing resolution should be, larger is better
// as its more 'precise' - but too large and you wont get
// accurate timing
#define RESOLUTION 20

// we will store up to 100 pulse pairs (this is -a lot-)
uint16_t pulses[100][2]; // pair is high and low pulse
uint8_t currentpulse = 0; // index for pulses we're storing

voidsetup(void) {
  Serial.begin(9600);
  Serial.println("Ready to decode IR!");
}

voidloop(void) {
  uint16_t highpulse, lowpulse; // temporary storage timing
  highpulse = lowpulse = 0; // start out with no pulse length

  // while (digitalRead(IRpin)) { // this is too slow!
  while(IRpin_PIN & _BV(IRpin)) {
    // pin is still HIGH
```

```

// count off another few microseconds
highpulse++;
delayMicroseconds(RESOLUTION);

// If the pulse is too long, we 'timed out' - either nothing
// was received or the code is finished, so print what
// we've grabbed so far, and then reset
if((highpulse >= MAXPULSE) && (currentpulse != 0)) {
  printpulses();
  currentpulse=0;
  return;
}
// we didn't time out so lets stash the reading
pulses[currentpulse][0] = highpulse;

// same as above
while(! (IRpin_PIN & _BV(IRpin))) {
  // pin is still LOW
  lowpulse++;
  delayMicroseconds(RESOLUTION);
  if((lowpulse >= MAXPULSE) && (currentpulse != 0)) {
    printpulses();
    currentpulse=0;
    return;
  }
  pulses[currentpulse][1] = lowpulse;

  // we read one high-low pulse successfully, continue!
  currentpulse++;
}

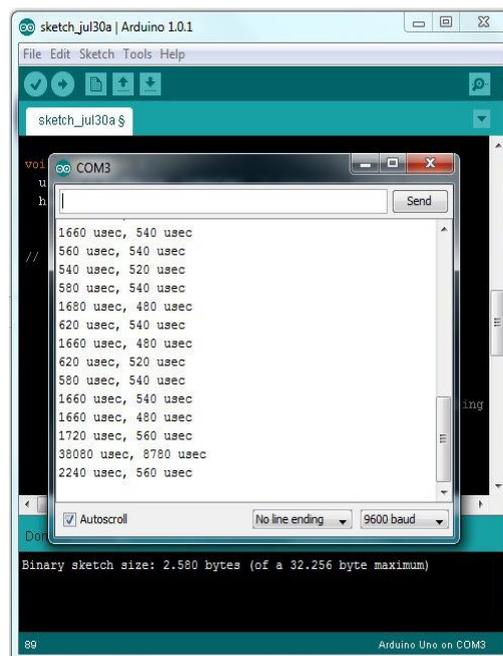
void printpulses(void) {
  Serial.println("\n\rReceived: \n\rOFF \tON");
  for(uint8_t i = 0; i < currentpulse; i++) {
    Serial.print(pulses[i][0] * RESOLUTION, DEC);
    Serial.print(" usec, ");
    Serial.print(pulses[i][1] * RESOLUTION, DEC);
    Serial.println(" usec");
  }
}

```

Feito isso, faça o upload e clique em Serial monitor. Você verá como na figura abaixo:

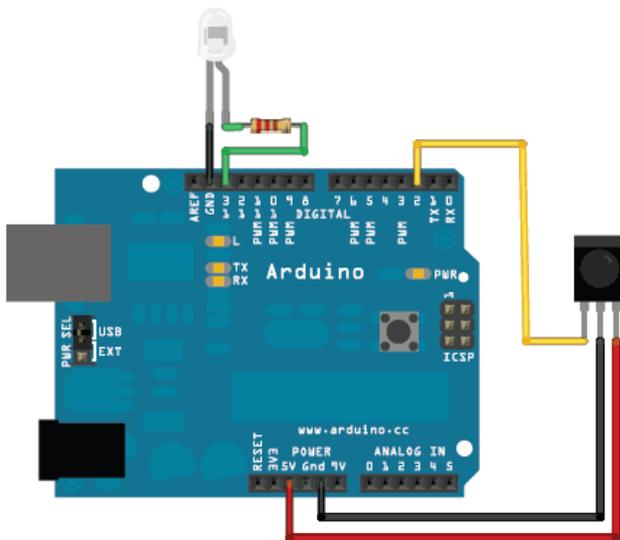


O Arduino está pronto para receber o código. Pegue o controle e aperte a tecla que você deseja clonar. Atenção: aperte apenas uma vez.



Agora você tem que copiar estes números um por um. A quantidade de número varia de controle para controle.

Vamos para segunda parte. Agora que temos os números não precisamos mais do Receptor Infravermelho ao nosso projeto. Fique a vontade de deixar ou retirar. No entanto, o Receptor Infravermelho estará no esquema caso queira clonar mais teclas. Lembrando que para clonar mais, repita novamente o tutorial.



Cole no IDE o seguinte código:

```
// This sketch will send out a Nikon D50 trigger signal (probably works with most Nikons)
// See the full tutorial at http://www.ladyada.net/learn/sensors/ir.html
// this code is public domain, please enjoy!
```

```
intIRledPin = 13; // LED connected to digital pin 13
```

```
// The setup() method runs once, when the sketch starts
```

```
voidsetup() {
// initialize the IR digital pin as an output:
pinMode(IRledPin, OUTPUT);
```

```
Serial.begin(9600);
}
```

```
voidloop()
{
Serial.println("Sending IR signal");
```

```
SendChannelUpCode();
delay(500);
}
```

```
// This procedure sends a 38KHZ pulse to the IRledPin
// for a certain # of microseconds. We'll use this whenever we need to send codes
voidpulseIR(longmicrosecs) {
// we'll count down from the number of microseconds we are told to wait
```

```
cli(); // this turns off any background interrupts
```

```

while(microsecs > 0) {
// 38 kHz is about 13 microseconds high and 13 microseconds low
digitalWrite(IRledPin, HIGH); // this takes about 3 microseconds to happen
delayMicroseconds(10);      // hang out for 10 microseconds
digitalWrite(IRledPin, LOW); // this also takes about 3 microseconds
delayMicroseconds(10);      // hang out for 10 microseconds

// so 26 microseconds altogether
microsecs -= 26;
}

sei(); // this turns them back on
}

voidSendChannelUpCode() {
// This is the code for the CHANNEL + for the downstairs TV COMCAST
delayMicroseconds(); //Time off (Left Column on serial monitor)
pulseIR();           //Time on (Right Column on serial monitor)
delayMicroseconds();
pulseIR();
}

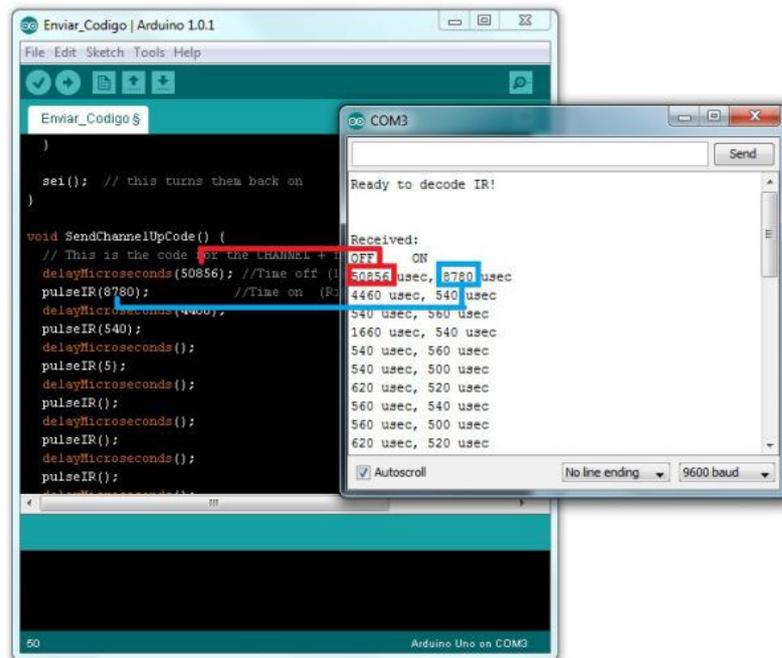
```

É agora que vamos usar aqueles números.

Observe um trecho do código:

```
void SendChannelUpCode() {
  delayMicroseconds();
  pulseIR();
}
```

No “delayMicroseconds(;)” você vai colocar o primeiro número dentro dos parênteses e no “pulseIR(;)” você vai colocar o segundo número também dentro do parênteses, observe com atenção a imagem:



Talvez você precise de mais ou menos delayMicroseconds() e pulseIR(), faça suas devidas mudanças, até que todos os números acabem, e deixe na ordem certa.

Feito isso, faça o upload e se você fez tudo certo, ao apontar o Led IR para a televisão ou para o aparelho que você escolheu, ele vai receber o código clonado, como se fosse o próprio Controle Remoto.

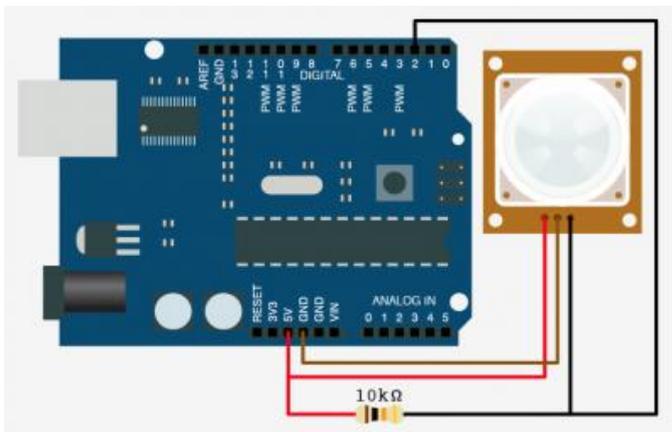
## ANEXO B – Tutorial Sensor PIR

**Fonte:** <http://labdegaragem.com/profiles/blogs/tutorial-como-utilizar-o-sensor-pir-passive-infrared-com-arduino>

Neste tutorial vamos mostrar como utilizar o sensor PIR (Passive Infrared) com Arduino. O sensor PIR é um sensor de movimento por calor (infravermelho). O sensor ativa o alarme assim que detecta uma mudança no infravermelho em um determinado lugar. Portanto, se uma pessoa se movimentar na frente do sensor PIR, este irá detectar a mudança e ativará o alarme.

OBS: A pinagem abaixo está incorreta! Assim como no datasheet! Siga a seguinte pinagem em vermelho: Marrom - GND, Preto - VCC (5V), Vermelho - Saída Digital.

O sensor PIR tem três fios: GND (marrom), VCC (vermelho) e Saída (preto). A figura abaixo mostra a ligação a ser feita no Arduino:



Para que o sensor PIR funcione corretamente é necessário um resistor de 10Kohm como Pull-Up. Para saber mais sobre resistores de Pull-Up, clique aqui.

A programação está mostrada abaixo:

```
int pirPin = 2; //digital 2
intLedPin = 13;

void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(LedPin,OUTPUT);
}

void loop(){
  int pirVal = digitalRead(pirPin);

  if(pirVal == LOW){ //was motion detected
    digitalWrite(LedPin,HIGH);
  }
}
```

```
delay(2000);  
}  
else  
{  
digitalWrite(LedPin,LOW);  
}  
}
```

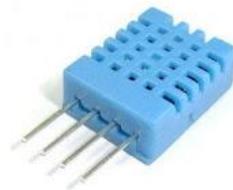
Abra a IDE do Arduino e passe a programação acima. Selecione a versão do seu Arduino (UNO, Duemilanove, etc) e a porta (COMx, ttyUSBx, ttyACMx, etc) e clique em UPLOAD.

Ao passar a mão na frente do sensor PIR, o Led da placa Arduino acenderá. Caso contrário ficará apagado.

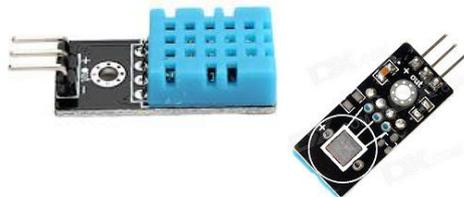
## ANEXO C – Tutorial Sensor DHT11

O sensor DHT11 é um sensor que mede a temperatura e umidade. Ele permite medir temperaturas de 0 a 50 Graus, e a umidade entre 20 a 90 %. Por não ser um sensor muito rápido e preciso, ele não é recomendada para projetos em ambientes que demandem alto risco no controle da temperatura e umidade.

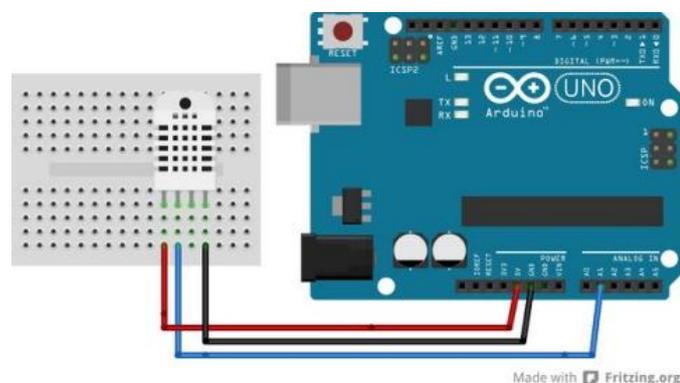
A temperatura tem uma faixa de precisão em 2 graus, e a de umidade na faixa de 5%. O sensor DHT11 é uma solução inteligente e barata devido a instalação dos sensores de temperatura e umidade em um único invólucro, o que permite diminuir os custos adicionais de produção. O sensor possui 4 pinos, mas apenas 3 são utilizados, sendo que o pino 3 não é utilizado.



O mais comum é encontrá-lo em forma de módulo ou brick, onde temos apenas 3 pinos (Vcc, Data e Gnd) como nas figuras abaixo:



Abaixo você pode ver a ligação do sensor a uma das portas do Arduino.



O programa abaixo lê as informações do sensor, e apresenta essas informações no serial monitor do Arduino.

#### 5.1.1.1 Código de Exemplo:

```
#include <dht.h>
#define dht_dpin A1 //Pino 2 Data do sensor vai ligado a porta A1 do Arduino
dht DHT; //Inicialização do sensor
void setup() //configurações
{
  Serial.begin(9600);
  delay(1000); //Aguarda 1 segundo antes de acessar as informações do dht11
}
void loop() //programa principal
{
  DHT.read11(dht_dpin); //Faz leitura das informações do sensor
  Serial.print("Umidade = ");
  Serial.print(DHT.humidity);
  Serial.print(" % ");
  Serial.print("Temperatura = ");
  Serial.print(DHT.temperature);
  Serial.println(" Celsius ");
  delay(2000); //Faz a leitura a cada 2 segundos, conselho a não diminuir muito o valor
}
```