

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS GUARAPUAVA
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

ANDRÉ GOMES

**PROJETO DE UMA REDE SENSOR SEM FIO PARA
MONITORAMENTO AMBIENTAL UTILIZANDO PROTOCOLO
ZIGBEE**

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA

2016

ANDRÉ GOMES

**PROJETO DE UMA REDE SENSOR SEM FIO PARA
MONITORAMENTO AMBIENTAL UTILIZANDO PROTOCOLO
ZIGBEE**

Trabalho de Conclusão de Curso apresentado à disciplina Trabalho de Conclusão de Curso II do Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação.

Orientador: Prof. Me. Paulo Henrique Soares

Co-orientador: Prof. Esp. Maurício Barfknecht

GUARAPUAVA

2016

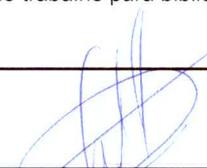
ATA DE DEFESA DE MONOGRAFIA DE TRABALHO DE CONCLUSÃO DE CURSO DO CURSO DE TSI

No dia 29 de novembro de 2016, às 15:00 horas, nas dependências da Universidade Tecnológica Federal do Paraná Câmpus Guarapuava, ocorreu a banca de **defesa da monografia** de Trabalho de Conclusão de Curso intitulada: “**Sistema de Apoio à Gestão de Comércio Eletrônico por Inteligência de Negócios**” da acadêmico **Felipe Gomes** sob orientação do professor **Prof. Me. Emerson Andre Fedechen** do Curso de Tecnologia em Sistemas para Internet.

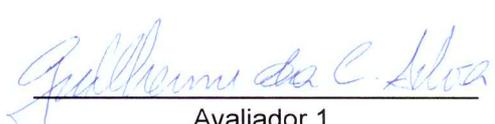
Banca Avaliadora	
Membro	Nome
Orientador	Prof. Me. Emerson Andre Fedechen
Avaliador 1	Prof. Me. Guilherme da Costa Silva
Avaliador 2	Prof. Esp. Fábio Leandro Janiszewski

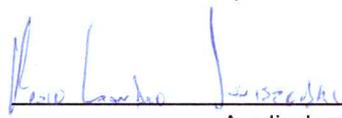
Situação do Trabalho

Situação	<input type="checkbox"/> Aprovado <input checked="" type="checkbox"/> Aprovado com ressalvas <input type="checkbox"/> Reprovado <input type="checkbox"/> Não Compareceu
Encaminhamento do trabalho para biblioteca	<input checked="" type="checkbox"/> Pode ser encaminhado para biblioteca. <input type="checkbox"/> Manter sigilo para publicação ou geração de patente.


 Orientador.
 Prof. Me. Emerson Andre Fedechen


 Orientando
 Felipe Gomes


 Avaliador 1
 Prof. Me. Guilherme da Costa Silva


 Avaliador 2
 Prof. Esp. Fábio Leandro Janiszewski

Guarapuava, 29 de novembro de 2016.

RESUMO

GOMES, André. Projeto de uma rede sensor sem fio para monitoramento ambiental utilizando protocolo Zigbee. 43 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

O corpo humano é uma máquina térmica que necessita estar equilibrada para bom funcionamento. Para manter o equilíbrio o nível de temperatura ideal gira em torno de 36 a 37 °C. Deste modo, saber se os níveis de temperatura são seguros para permanência pode ajudar a evitar problemas de perda de rendimento. Pensando nisso o foco deste trabalho é desenvolver uma rede de sensores sem fio para monitoramento da temperatura em vários pontos de um determinado ambiente, com o objetivo de verificar quais lugares estão mais seguros para permanência. Para o desenvolvimento do projeto será utilizado o protocolo *Zigbee*, através de um módulo *Xbee*. Também serão utilizados sensores de temperatura do tipo LM35 e a plataforma Arduino. Ao final do projeto espera-se que a ferramenta contribua evitando desconforto e de queda de desempenho no trabalho causados por níveis inadequados de temperatura.

Palavras-chave: Rede de Sensores, Zigbee, Xbee, Monitoramento Ambiental, Redes sem Fio

ABSTRACT

GOMES, André. Design of a wireless sensor network for environmental monitoring using Zigbee protocol. 43 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

The human body is a thermal machine that needs to be balanced for proper operation. To maintain equilibrium the ideal temperature level is around 36 to 37 °C. In this way, knowing whether temperature levels are safe for permanence can help avoid problems of loss of yield. Thinking about it, the focus of this work is to develop a network of wireless sensors to monitor the temperature at various points in a given environment, in order to verify which places are safest to stay in. For the development of the project will be used the protocol *Zigbee*, through a module *Xbee*. Also used temperature sensors of type LM35 and the platform Arduino. At the end of the project, the tool is expected to contribute to avoiding physical discomforts and falling work performance caused by inadequate temperature levels.

Keywords: Sensor Networks, Zigbee, Xbee, Environmental monitoring, Wireless Networks

LISTA DE FIGURAS

Figura 9	– Fluxo de dados da plataforma	11
Figura 10	– Adaptador USB-XBee	12
Figura 11	– Esquema geral da RSSF	12
Figura 13	– Placas Arduino UNO	14
Figura 14	– Placa Genuino Uno Rev 3	15
Figura 15	– Exemplo de topologia estrela	16
Figura 16	– Exemplo de topologia em malha	17
Figura 17	– Exemplo de uma RSSF	18
Figura 18	– Sensor de temperatura LM35	19
Figura 19	– Sensor de temperatura e umidade DHT11	19
Figura 20	– Módulo XBee Antenna 2mW Serie 2	20
Figura 21	– XBee Explorer USB Adapter	21
Figura 22	– Software Fritzing	22
Figura 25	– Nó sensor	27
Figura 26	– Cálculo de conversão leitura do pino ADC	29
Figura 27	– Código: Conexão e envio de dados	30
Figura 28	– Página php: Conexão com banco de dados	31
Figura 29	– Página php: Inserção de dados	31
Figura 30	– Página web: Layout responsivo	32
Figura 31	– Nó central da rede feito no Fritzing	32
Figura 32	– Nó sensor da rede feito no Fritzing	33
Figura 33	– Esquema geral da RSSF	33

LISTA DE TABELAS

TABELA 9	–	Comparação entre as plataformas	10
TABELA 10	–	Comparação entre os trabalhos	13
TABELA 12	–	Comparativo Zigbee e outras tecnologias sem fio	17
TABELA 16	–	Comparação em Ambiente Interno	35
TABELA 17	–	Comparação Ambiente Interno x Ambiente Externo	36

LISTA DE SIGLAS

RSSF	Rede de sensores sem fio
OMS	Organização Mundial de Saúde
CO2	Dióxido de Carbono
DHT11	Sensor de Temperatura e Umidade
MG811	Sensor de Dióxido de Carbono
SGBD	Sistema Gerenciador de Banco de Dados
PAN	Personal Area Network
MY	Source Address
DH	Destination High
DL	Destination Low
SH	Serial Number High
SL	Serial Number Low
ADC	Conversor analógico-digital
TX	Transmissor
RX	Receptor
LSB	Least Significant Byte
MSB	Most Significant Byte
MAC	Media Access Control

SUMÁRIO

1 INTRODUÇÃO	8
1.1 OBJETIVOS	9
1.1.1 Objetivo Geral	9
1.1.2 Objetivos Específicos	9
1.2 DIFERENCIAL TECNOLÓGICO	9
1.3 ESTRUTURA DO TRABALHO	9
2 ESTADO DA ARTE	10
3 REFERENCIAL TEÓRICO	14
3.1 ARDUINO	14
3.1.1 Arduino Uno Rev 3	15
3.2 PROTOCOLO ZIGBEE	16
3.3 REDE DE SENSORES SEM FIO	17
3.3.1 Sensor LM35	18
3.3.2 Sensor DHT11	19
3.4 XBEE	20
3.4.1 Módulo Xbee	20
3.4.2 XBee Explorer	20
3.4.3 XCTU	21
3.5 FRITZING	21
4 METODOLOGIA	23
5 DESENVOLVIMENTO	25
5.1 RSSF	25
5.1.1 Módulos Xbee	25
5.1.1.1 Configuração dos Módulos Xbee	26
5.1.2 Nós sensores	26
5.1.3 Nó Central	27
5.2 ENVIO DE DADOS	29
5.2.1 Conexão e envio de dados para servidor	29
5.3 SERVIDOR E APLICAÇÃO WEB	30
5.3.1 Conexão com a base de dados	30
5.3.2 Aplicação Web	31
5.4 FLUXO DE DADOS	32
6 TESTES E RESULTADOS	34
6.1 TESTES	34
7 CONSIDERAÇÕES FINAIS	37
7.1 DIFICULDADES ENCONTRADAS	37
7.2 REAFIRMAÇÃO DA CONTRIBUIÇÃO	37
7.3 TRABALHOS FUTUROS	37
REFERÊNCIAS	39
Anexo A – CÓDIGO FONTE ARDUINO (NÓ CENTRAL)	41

1 INTRODUÇÃO

As Redes de Sensores Sem Fio RSSF, tem sido tema recorrente em diversas pesquisas. Estas redes possuem diversas aplicações que vão desde ferramentas militares, controle e monitoramento até aplicações na área de saúde Ian et al. (2002). Estas RSSF são uma área de pesquisa relativamente nova e que trazem diversos benefícios. Entre eles está a possibilidade de desenvolver ferramentas capazes de concentrar e avaliar dados de diversos pontos de um determinado ambiente. Pensando nisso, lembra-se de um perigo ao qual diariamente estamos expostos: a temperatura do ar. Este mesmo ar que torna possível a vida pode tirá-las se não estiver em condições favoráveis. Condições estas que ao contrário do que pensam não estão somente ligadas a quantidade de poluentes presentes no ar, pois temperaturas muito altas ou baixas já apresentam graves riscos à saúde.

Segundo recomendações da Organização Mundial da Saúde OMS, altas temperaturas não são adequadas para saúde humana e podem causar problemas Cepagri (2009). Conforme Silva e Almeida (2010), altas temperaturas reduzem a capacidade do corpo humano de manter sua temperatura interna correta. Além de desconforto térmico, em um ambiente de trabalho essas condições podem causar ainda mais problemas. Iida (2005) afirma que quando uma pessoa é obrigada a suportar altas temperaturas, seu rendimento no trabalho diminui drasticamente. A velocidade do trabalho diminui, as pausas se tornam recorrentes e mais longas e a propensão a acidentes aumenta.

Por conta disso, o foco deste trabalho é projetar e implementar uma rede de sensores sem fio utilizando protocolo *Zigbee* e tecnologias livres. Espera-se que a plataforma seja capaz de medir se os níveis de temperatura em diversos pontos de um ambiente estão ou não seguros para presença de seres humanos e desta maneira contribuindo para evitar problemas de desconforto térmico e diminuição de desempenho no trabalho.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Desenvolver uma rede sensor sem fio capaz de aferir os níveis de temperatura em diversos pontos de um ambiente utilizando tecnologias livres.

1.1.2 OBJETIVOS ESPECÍFICOS

São objetivos específicos deste trabalho:

- Fazer o levantamento teórico das tecnologias envolvidas.
- Desenvolver os algoritmos para estabelecer comunicação entre os sensores e o Arduíno.
- Implementar a rede de sensores sem fio.
- Realizar testes na medição dos níveis de temperatura.

1.2 DIFERENCIAL TECNOLÓGICO

O principal diferencial tecnológico é a possibilidade de medir os níveis de temperatura em diversos pontos de um mesmo local, sem a dependência de cabos, enquanto a maioria das tecnologias presentes no mercado hoje realizam essa medição em apenas um determinado ponto. Isso permite avaliar quais lugares de uma mesma área estão adequadas para permanência. Além disso, por utilizar tecnologias livres, a RSSF terá baixo custo, sendo um ótimo diferencial em relação as tecnologias atuais que são muito caras.

1.3 ESTRUTURA DO TRABALHO

Essa monografia está dividida da seguinte forma: no capítulo dois é apresentado o estado da arte do projeto, comparando com outros trabalhos na mesma área de pesquisa. No terceiro capítulo é descrito o referencial teórico do projeto, bem como as tecnologias estudadas e quais foram utilizadas. No capítulo quatro é a metodologia seguida para desenvolver todas as etapas do projeto é exposta. O quinto capítulo é sobre o desenvolvimento, descrevendo detalhadamente como o projeto foi feito. E no sexto capítulo são descritas as considerações finais sobre o projeto.

2 ESTADO DA ARTE

Nesta seção serão descritos dois trabalhos correlatos, um na área de monitoramento da qualidade do ar e outro na área de redes de sensores sem fio. Serão apresentadas as principais características de cada um, ressaltando os pontos que podem ser aproveitados no projeto.

O primeiro é o projeto de trabalho de conclusão de curso do autor Almeida (2015), que propõe a construção de uma plataforma para monitorar a qualidade do ar. Este trabalho irá contribuir em toda parte sensorial do projeto, por utilizar os mesmos sensores que serão utilizados na RSSF. O foco foi desenvolver uma ferramenta capaz de monitorar em um ambiente interno fechado a presença de CO₂ e os níveis de umidade e temperatura (Almeida, 2015). Para construção da ferramenta, será utilizada a plataforma de prototipação eletrônica Arduino, sensor de temperatura DHT11 e ainda o sensor de dióxido de carbono MG811, para medir os níveis de CO₂ no ambiente. Os dados coletados são importantes para saber se o ambiente é ou não adequado para presença de seres humanos e serão disponibilizados para visualização através de um painel de LED na própria plataforma e também em um servidor em nuvem.

Conforme Almeida (2015), o grande diferencial tecnológico da plataforma é a capacidade de disponibilizar os dados em nuvem, possibilitando a visualização por qualquer interessado em avaliar a qualidade do ar no ambiente onde a plataforma se encontra. Além disso, outra vantagem é o custo da ferramenta, que mesmo oferecendo vantagens em relação a algumas ferramentas no mercado pode ser desenvolvida com investimento menor, dados que podem ser observados na Tabela 9.

Tabela 9: Comparação entre as plataformas

Fonte: Almeida (2015)

Plataforma	CO ₂ Lim	Temp. Lim	Umid. Lim	Preço
Autopilot Desktop CO ₂ Monitor	0 - 3,000 ppm	0 C - 50 C	20% - 90%	\$ 134
cSense CO ₂ + RH/T Monitor	0 - 9,999 ppm	-10 C - 60 C	0.1% - 99.9%	\$ 249
Mini CO ₂ Monitor	0 - 3,000 ppm	0 C - 50 C -	nao registra	\$ 109
Plataforma proposta	0 a 10,000 ppm	0 C - 50 C	20% - 90%	\$ 96

Segundo Almeida (2015) a plataforma Arduino irá capturar os dados coletados pelos

sensores e enviá-los para o servidor, onde serão armazenados em uma base dados. A base utilizada será controlada pelo Sistema Gerenciador de Banco de Dados SGBD MySQL. Depois que estes dados estiverem salvos o servidor já poderá fazer consultas e enviá-los para um servidor em nuvem. Assim, a plataforma proporciona um meio de visualizar e monitorar a emissão de CO₂ no ambiente. O fluxo do projeto será semelhante ao da Figura 9.

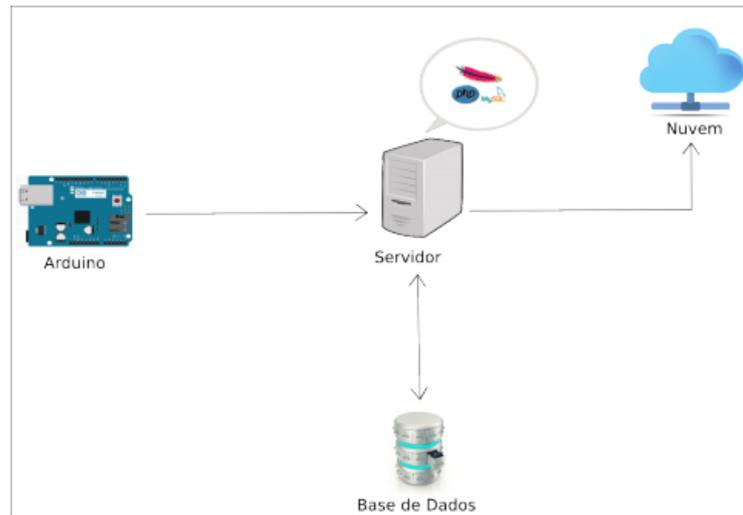


Figura 9: Fluxo de dados da plataforma

Fonte: Almeida (2015)

O segundo trabalho é a monografia de Souza Filho (2011) que propõe o desenvolvimento de uma rede de sensores sem fio utilizando Zigbee para aplicações diversas. Neste trabalho, o autor desenvolveu um sistema completo (plataformas de *hardware*, *software* e *firmware*) que permitem montar uma RSSF utilizando a tecnologia Zigbee, com a finalidade de ser utilizado em uma ampla variedade de aplicações. Conforme Souza Filho (2011), por conta das características favoráveis do Zigbee, como baixa taxa de transmissão por exemplo, fazem com que ele seja o mais utilizado em RSSF, além do fato de que utilizando Zigbee, pode se utilizar a mesma RSSF desenvolvida para abstrair uma nova aplicação com objetivos diferentes somente se preocupando com a seleção dos sensores e análise dos dados coletados. O módulo Zigbee utilizado foi XbeePRO Znet 2.5 OEM, produzido pela Digi. A RSSF desenvolvida é composta por quatro elementos, sendo estes plataformas de hardware, módulo coordenador da rede, módulo concentrador e um computador com um SGBD instalado. Estas plataformas de *hardware* são os nós da rede, que podem ser tanto roteadores quanto dispositivos finais, além de terem a capacidade de sensoriamento. São sistemas microcontrolados que suportam o módulo Zigbee (Souza Filho, 2011).

Souza Filho (2015) explica que o módulo coordenador e o concentrador não possuem

capacidade de sensoriamento, mas se houver redundância de alimentação, então pode-se utilizar a mesma plataforma dos nós sensores com esse fim. A conexão do concentrador com o computador se dá através de um adaptador USB para módulos *XBee*, o qual pode ser visualizado na Figura 10. O nó coordenador utilizado pode ser qualquer um que não esteja sendo utilizado com concentrador. Embora seja possível utilizar o mesmo nó para ambas as tarefas, não é indicado para não ficar sobrecarregado com funções de gerenciamento de rede. Além disso, utilizando um nó que não é o coordenador como concentrador torna mais fácil estender a RSSF para um cenário de múltiplos concentradores.



Figura 10: Adaptador USB-XBee

Fonte: Souza Filho (2011)

O último elemento utilizado é um computador com o SGBD MySQL instalado. A função deste elemento na rede é armazenar os dados coletados pela RSSF. Um esquema geral da rede pode ser visualizado na figura 11. A RSSF foi desenvolvida com sucesso e segundo Souza Filho (2011) por conta da natureza desse tipo de aplicações pode-se prever que em breve elas farão parte integral das nossas vidas.

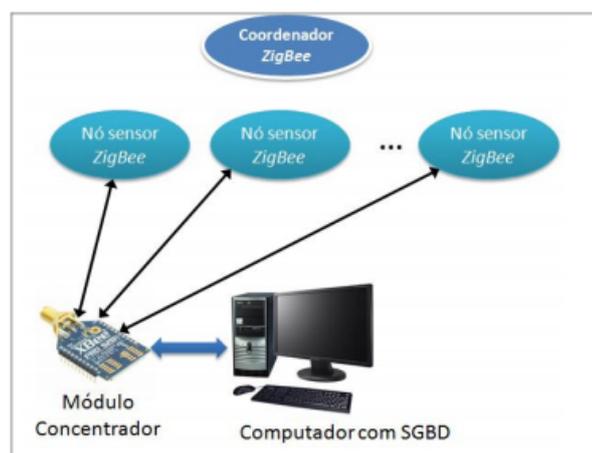


Figura 11: Esquema geral da RSSF

Fonte: Souza Filho (2011)

Ambos os trabalhos representam grande contribuição para este projeto, por utilizarem tecnologias semelhantes e alguns casos as mesmas que serão utilizadas. Na tabela 10 está um comparativo entre os trabalhos, destacando as particularidades do projeto, também citando as

principais características e também criticando os pontos negativos dos trabalhos descritos nesta seção.

Tabela 10: Comparação entre os trabalhos

Fonte: O autor

	Trabalho de Almeida	Trabalho de Souza Filho	Projeto proposto
Características	O projeto será de grande contribuição pelo fato de utilizar o sensor DHT11 e Arduino para o monitoramento da temperatura do ar.	O trabalho servirá como uma boa base para a RSSF por utilizar o protocolo Zigbee, precisamente através de módulos <i>Xbee</i> . Tecnologias estas que serão utilizadas neste projeto.	O projeto vai de certa forma unir os dois trabalhos, uma RSSF com foco em monitoramento da temperatura do ar. Buscando obter as principais qualidades de cada trabalho para esse, além de novas funcionalidades.
Críticas	Apesar de ser um ótimo trabalho na questão do monitoramento, o ponto fraco deste trabalho é a limitação de poder aferir os níveis apenas em um ponto de um determinado ambiente interno fechado.	O trabalho de Souza Filho é uma boa fonte de material na área de RSSF. O que faltou foi aplicar de fato a tecnologia, visto que o projeto é apenas de uma RSSF genérica.	O projeto leva vantagem em relação aos outros dois. A respeito do trabalho de Almeida, pelo fato de poder monitorar a temperatura do ar em diversos pontos de um ambiente, seja este interno ou externo. Quanto ao trabalho de Souza Filho, a vantagem é estar de fato aplicando uma RSSF para contribuir com a sociedade.

3 REFERENCIAL TEÓRICO

Nesta seção serão descritos os principais conceitos e tecnologias que serão utilizados no trabalho, destacando as características de cada um.

3.1 ARDUINO

Com o passar dos anos o Arduino tem se tornado o cérebro de milhares de projetos que vão desde aplicações simples para uso diário até instrumentos científicos complexos. Com uma comunidade de alcance mundial, formada por estudantes, programadores e profissionais, uma incrível quantidade de conhecimento acessível a respeito da plataforma é hoje facilmente encontrado, seja para iniciantes ou profissionais da área (Arduino, 2015).



Figura 13: Placas Arduino UNO

Fonte: Arduino (2015)

Arduino é uma plataforma de prototipação de código aberto baseada nos princípios de facilidade para utilização de hardware e software. As placas de Arduino são capazes de ler entradas de dados como luz em um sensor, um botão pressionado e até mesmo uma mensagem no *Twitter*. Depois processar essa entrada em uma saída que vai desde ativar um motor, ligar um LED até aplicações mais complexas. Deve ser definido o que a placa irá fazer através de instruções que são enviadas a um microcontrolador, instruções estas que são escritas utilizando

a linguagem de programação Arduino e a Arduino Software IDE. Na Figura 13 pode se observar duas placas Arduino, uma Arduino UNO e outra Genuino UNO, Arduino e Genuino são marcas oficiais (Arduino, 2015).

A escolha pelo Arduino como tecnologia utilizada neste projeto se deu por conta da grande quantidade de material acessível na área, mesmo para iniciantes. Além disso, utilizando tecnologias livres conseguimos manter o diferencial no custo baixo da plataforma, pois as placas exigem baixo investimento. Neste projeto serão utilizados vários componentes e eles estão descritos nas subseções a seguir.

3.1.1 ARDUINO UNO REV 3

O Arduino Uno Rev 3 é uma placa microcontroladora baseada no chip *ATmega328*. Possui 14 pinos de entrada e saída digital, dentre as quais 6 podem ser utilizados como saída PWM e 6 são entradas analógicas. Também conta com uma conexão USB, entrada de alimentação, cristal oscilador de 16 MHz e um botão de resetar (Arduino, 2015). Ele contém tudo que é necessário para o microcontrolador, para utilizar basta conectá-lo a um computador através de um cabo USB ou ligar através de uma fonte de energia.

O UNO difere de todas as placas anteriores por não utilizar o chip FTDI de conversão de USB para serial. Na figura 14 está a imagem de um Genuino UNO Rev 3, que possui as mesmas especificações do Arduino, difere apenas o nome da marca oficial de trabalho fora dos Estados Unidos (Arduino, 2015).



Figura 14: Placa Genuino Uno Rev 3

Fonte: Arduino (2015)

A placa será utilizada neste projeto como nó central da RSSF, através de um *Xbee Shield* um módulo *Xbee* será ligado ao Arduino. Nesta mesma placa é que posteriormente serão salvos os dados coletados pelos sensores da rede, em um cartão SD ou em uma base de dados utilizando o SGBD MySQL.

3.2 PROTOCOLO ZIGBEE

A tecnologia de rede sem fio *Zigbee* é a principal norma global para implementação de redes de curto alcance com baixo custo, baixa taxa de dados e com vida útil estendida da bateria (Farahani, 2008). Conforme Souza Filho (2011), por possuir as características favoráveis, como baixa taxa de transmissão e curto alcance por exemplo, o *Zigbee* tem sido amplamente utilizado em várias aplicações que necessitam de redes de sensores sem fio.

O *Zigbee* é uma norma que define um conjunto de protocolos de comunicação para redes sem fio que operam com baixa taxa de dados e curto alcance. Os dispositivos baseados em *Zigbee* operam nas frequências de 868 MHz, 915 MHz e 2.4 GHz, sendo que taxa máxima de dados é de 250 K bits por segundo. *Zigbee* é destinado principalmente para aplicações alimentadas por bateria onde a baixa taxa de dados, baixo custo e bateria de longa vida são requisitos principais. Em grande parte das aplicações o tempo total que o dispositivo sem fio está engajado em qualquer tipo de atividade é muito limitado, pois o dispositivo passa a maior parte do tempo em modo de economia de energia. Como resultado disso dispositivos que trabalham com *Zigbee* são capazes de ficar operacionais por vários anos antes de suas baterias precisarem ser substituídas (Farahani, 2008).

A norma *Zigbee* é desenvolvida pela *Zigbee Alliance* que possui como membros centenas de companhias. Ela foi formada em 2002 como uma organização sem fins lucrativos aberta para todos que quiserem participar. A norma *Zigbee* adotou a IEEE 802.15.4 como sua camada física e de controle de acesso ao meio. Além disso, o *Zigbee* ajuda a reduzir o custo de implementação por simplificar os protocolos de comunicação e reduzir a taxa de dados. Os requisitos mínimos para aprender sobre as especificações do *Zigbee* e IEEE 802.15.4 são relativamente mais tranquilos quando comparados a outras normas como por exemplo o IEEE 802.11, o que reduz a complexidade e custo de implementação (Farahani, 2008).



Figura 15: Exemplo de topologia estrela

Fonte: Farahani (2008)

No quesito topologia, a formação da rede é gerenciada pelo camada de rede do *Zigbee*. A rede precisa estar em uma das duas topologias especificadas na IEEE 802.15.4: estrela ou

peer-to-peer. Na topologia estrela, como pode ser visto na Figura 15, cada dispositivo pode se comunicar apenas com o coordenador PAN.

Em uma topologia *peer-to-peer* cada dispositivo pode se comunicar diretamente com qualquer outro desde que estejam próximos o suficiente para estabelecer um link de comunicação. Uma topologia *peer-to-peer* pode assumir diferentes formatos bastando para isto definir restrições a respeito de quais dispositivos podem se comunicar. Se não houver nenhuma restrição, a rede *peer-to-peer* é conhecida com topologia *mesh* ou malha. A Figura 16 mostra o exemplo de uma topologia em malha.

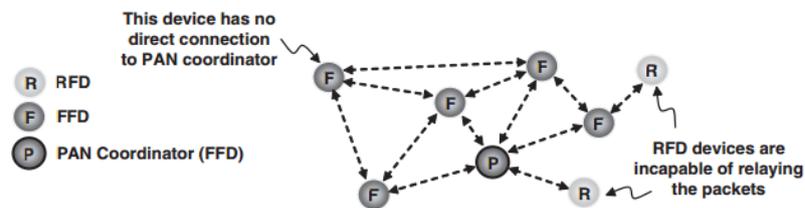


Figura 16: Exemplo de topologia em malha

Fonte: Farahani (2008)

As características do *Zigbee* como foi observado são favoráveis para serem utilizadas em projetos como redes de sensores sem fio. A Tabela 12 reforça que o *Zigbee* é o protocolo mais adequado para aplicações de RSSF, mostrando com detalhes uma comparação entre as características do *Zigbee*, *Bluetooth* e a norma IEEE 802.11b que é o padrão *wireless*. Estudar os protocolos e entender o seu funcionamento é um passo fundamental para o bom andamento do projeto. No projeto será utilizado um módulo *Zigbee Xbee*, para ser usado em conjunto com o Arduino.

Tabela 12: Comparativo Zigbee e outras tecnologias sem fio

Fonte: Ramos (2012)

	Taxa de transmissão	Raio de alcance	Aplicação
Zigbee	20 a 250 kbps	10 a 100m	Redes de sensores sem fio
Bluetooth	1 a 3 Mbps	2 a 10m	Fones e mouses sem fio
IEEE 802.11b	1 a 11 Mbps	10 a 100m	Conexão de internet sem fio

3.3 REDE DE SENSORES SEM FIO

Com o passar dos anos os avanços nas área de comunicação sem fio e sistemas embarcados têm proporcionado uma nova geração de redes de sensores. As aplicações vão desde as mais simples até experimentos complexos, sendo alguns exemplos de utilização as áreas de

monitoramento de trânsito, monitoramento de animais, detecção de incêndios e monitoramento ambiental (Souza Filho, 2011).

Segundo Akyildiz e Kasimoglu (2004), redes de sensores sem fio são um grupo de sensores e atores ligados por um meio sem fio para realizar sensoriamento distribuído e tarefas agindo de uma rede desse tipo. O papel dos sensores é de reunir informações sobre o mundo físico, enquanto o dos atores é tomar decisões e executar ações apropriadas sobre o meio ambiente. Assim, um dos principais objetivos na WSNs é diminuir o atraso de comunicação entre o sentir e agir nas tomadas de decisão sobre um determinado meio.

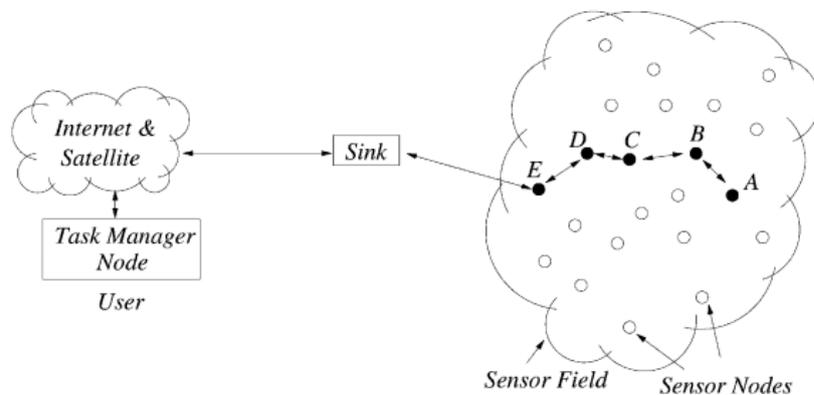


Figura 17: Exemplo de uma RSSF

Fonte: Ian et al. (2002)

Como o foco deste trabalho é desenvolver uma RSSF, os conceitos e definições serão utilizados em todo andamento do projeto. Seguindo as principais recomendações e respeitando as boas práticas e definições deste tipo de rede. A seguir estão descritas as tecnologias relacionadas a RSSF.

3.3.1 SENSOR LM35

Segundo Shah (2015) o LM35 é um sensor de temperatura com base de semicondutores. É um sensor analógico integrado cuja potência elétrica é proporcional a temperatura. Não requer qualquer calibração externa para fornecer dados precisos. Possui baixa impedância, saída linear e de fácil utilização.

O LM35, ainda afirma Shah (2015), possui um custo muito baixo e é facilmente encontrado. A principal vantagem deste sensor é o fato de que ele é linear, ou seja, 10 mV correspondem a 1 grau Celsius. Isso significa que para cada aumento de grau na temperatura

a saída do sensor vai aumentar em 10 mV. Por exemplo, se a saída do LM35 é 220mV ou 0.22V a temperatura será de 22 °C. Isso facilita muito a conversão da saída para informações de temperatura. Um sensor LM35 pode ser visto na figura 18.

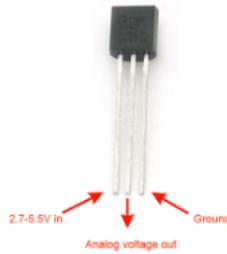


Figura 18: Sensor de temperatura LM35

Fonte: Shah (2015)

3.3.2 SENSOR DHT11

Os sensores de temperatura de umidade DHT11 são básicos e de baixo custo. São construídos em 2 partes para poder medir níveis de umidade e temperatura. O sensor conta com um pequeno e básico chip interno que faz uma conversão de um sinal analógico para digital e então o retorna com a temperatura e a umidade. Este sinal emitido é facilmente lido com um microcontrolador (ADAFRUIT, 2012).

Conforme Adafruit (2012), o sensor DHT11 é bom para leituras de umidade de 20% e 80%, com precisão podendo variar em 5%. No caso da temperatura, são bons para leituras entre 0°C e 50°C, com precisão de 2°C para mais ou menos. O sensor foi escolhido para ser utilizado neste projeto para servir de comparação com os dados de temperatura medidos pelo sensor LM35. O sensor pode ser visto na figura 19.

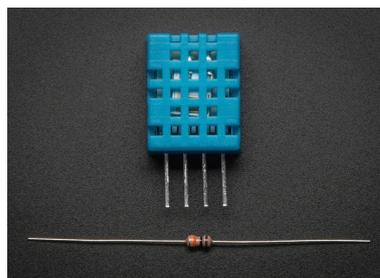


Figura 19: Sensor de temperatura e umidade DHT11

Fonte: Adafruit (2012)

3.4 XBEE

3.4.1 MÓDULO XBEE

O *Xbee* é um dispositivo que foi desenvolvido pela empresa *Digi International*, uma reconhecida empresa na área de dispositivos de comunicação sem fio. O *Xbee* então 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 no alcance e potência de transmissão (Eschner, 2011). A Figura 20 mostra um módulo XBee Antenna.



Figura 20: Módulo XBee Antenna 2mW Serie 2

Fonte: <https://www.adafruit.com/products/968>

Conforme Eschner (2011) além de poder operar em diferentes topologias, outra vantagem do módulo é o fato de que ele implementa a tecnologia de *sleep mode*, dando maior autonomia as baterias ou pilha de alimentação do dispositivo. Autonomia essa que em alguns casos pode chegar a alguns anos sem precisar substituir baterias. Os *XBees* também podem ser configurados para atuar como roteadores, formando uma topologia de rede *mesh*, ou malha. Desta maneira é possível expandir a rede conforme a necessidade. A responsabilidade pelos tratamentos e controle para garantir que as mensagens estão sendo transmitidas com sucesso entre os módulos é do próprio *firmware* do *XBee*, que respeita as regras e normas especificadas pelo padrão *Zigbee*.

O módulo *Xbee* é um dos principais componentes que serão utilizados no projeto e só o fato de ter o protocolo *zigbee* já embarcado em seu microcontrolador é uma grande vantagem. Os módulos *Xbee* serão os nós da rede e será de responsabilidade deles a comunicação entre cada nó sensor e a central.

3.4.2 XBEE EXPLORER

O *XBee Explorer USB Adapter* é utilizado para a configuração de parâmetros do módulo *XBee*, facilitando a transmissão sem fio de dados. Pode ser facilmente conectado a um computador via cabo *USB mini* Arduino (2015). Será utilizado para configurar todos os módulos *XBee* utilizados na construção da RSSF. A Figura 21 mostra um adaptador *XBee Explorer*.



Figura 21: XBee Explorer USB Adapter
Fonte: <https://www.adafruit.com/products/126>

3.4.3 XCTU

O *XCTU* é um sistema multi-plataforma livre, projetado para permitir aos desenvolvedores interagir com módulos de radiofrequência *Digi* através de uma interface gráfica simples de usar. Ele inclui diversas ferramentas que tornam mais fácil a conexão, utilização, configuração e teste de módulos *XBee* (DIGI, 2015).

Segundo *Digi* (2015) o sistema inclui todas as ferramentas que um desenvolvedor precisa para rapidamente trabalhar com *XBee*. Possui características únicas como visualização de rede, funcionalidade que representa graficamente a rede *XBee* juntamente com a força do sinal de cada conexão. Além uma interface intuitiva que permite um modo prático de configurar os módulos.

3.5 FRITZING

Segundo o site oficial da plataforma, *Fritzing* (2016) é uma iniciativa de hardware de fonte aberta que torna a eletrônica acessível como um material criativo para qualquer um. Além de oferecer uma ferramenta de software e um site da comunidade e serviços, promovendo um

ecossistema criativo que permite aos usuários documentar seus protótipos, compartilhá-los com outros e ensinar eletrônica em uma sala de aula.

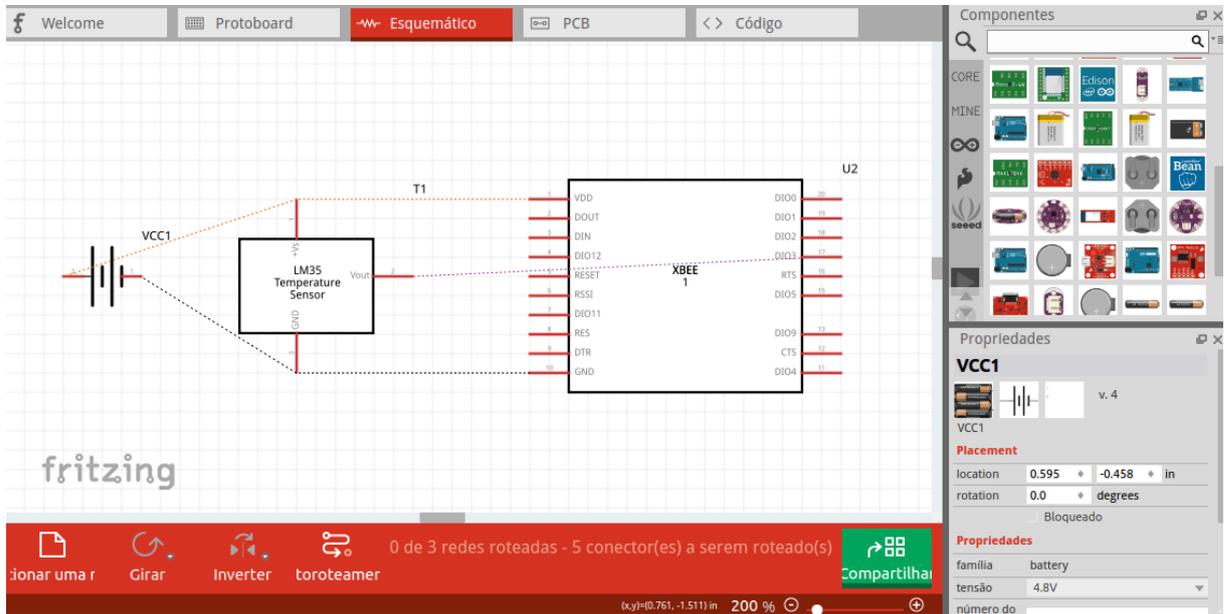


Figura 22: Software Fritzing

Fonte: O autor

4 METODOLOGIA

Esta seção descreve os procedimentos metodológicos utilizados para o desenvolvimento do projeto. Os passos descritos a seguir descrevem sucintamente as etapas e como o projeto foi elaborado.

1. Levantamento teórico das tecnologias

O primeiro passo do projeto foi compreender o funcionamento das tecnologias empregadas. Para isso foi realizado um estudo sobre o protocolo *Zigbee*, suas principais características e vantagens de sua utilização em relação a outras tecnologias sem fio, como *Bluetooth* e IEEE 802.11b, o *Wireless*. Estudar sobre os componentes que foram utilizados, como o módulo *Xbee*, compreendendo suas particularidades e funcionamento foi essencial. Da mesma forma, que foi necessário estudar a respeito da plataforma de prototipagem eletrônica Arduino e sobre a utilização do sensor LM35 de temperatura.

2. Desenvolvimento dos algoritmos de comunicação

Após realizado o levantamento teórico e avaliado como cada tecnologia funciona, foi necessário desenvolver os algoritmos para estabelecer a comunicação entre o Arduino e os sensores.

3. Desenvolvimento dos algoritmos de captura

Nesta etapa, foram desenvolvidos os algoritmos para realizar a captura e medir os níveis de temperatura do ar. Foi utilizado o sensor LM35.

4. Configuração da rede de sensores sem fio

Nesta fase do projeto foi configurada a rede de sensores sem fio e estabelecida a comunicação entre o Arduino e cada módulo *Xbee*, previamente configurado utilizando o *XBee Explorer* e o software *XCTU*. Em cada nó da rede juntamente com um *Xbee* antena foi conectado um sensor LM35. Os dados coletados foram armazenados em uma base de dados utilizando o SGBD MySQL.

5. Desenvolvimento da página web

A página web foi desenvolvida utilizando o framework *Laravel*, nesta foram disponibilizadas para visualização as informações a respeito da temperatura do ar previamente armazenadas na base de dados.

6. Testes

Os testes da RSSF foram realizados para conferir se a comunicação com todos os componentes está correta. A precisão dos dados coletados foi conferida através de comparação com dados coletados com o sensor DHT11 e de sites de previsão do tempo.

7. Avaliação dos resultados

Por fim, os resultados obtidos foram avaliados e descritos de forma clara, também foram descritos os cenários e ambientes em que a ferramenta foi testada e como se comportou.

5 DESENVOLVIMENTO

Nesta seção será descrito todo projeto, dividindo e descrevendo cada fase bem como suas principais características em etapas. O projeto todo foi desenvolvido com recursos próprios, todos os materiais utilizados foram adquiridos pelo autor.

5.1 RSSF

A RSSF é o foco principal do projeto e ao mesmo tempo que representa grande importância para o trabalho foi um dos maiores desafios no desenvolvimento. A rede foi configurada e formada por dois nós sensores e um nó central, os nós sensores são responsáveis por medir os níveis de temperatura, enquanto ao nó central cabe receber estes dados e enviar para um banco de dados em um servidor. Configurar toda a rede comunicando cada um dos nós entre si e enviar os dados coletados para a base foi uma tarefa complexa. A seguir estão descritos os principais aspectos da RSSF bem como componentes utilizados.

5.1.1 MÓDULOS XBEE

Os módulos *Xbee* operam como uma rede do tipo Personal Area Network PAN, redes de área pessoal, que segundo Hackmann (2006) são de baixo custo e compostas por dispositivos que operam em baixa frequência, localizados dentro de 1m a 100m entre si. Cada módulo pode operar basicamente em três configurações: *Coordinator*, *Router* ou *End-device*, que significam respectivamente Coordenador, Roteador e dispositivo final.

Os módulos ainda possuem dois tipos distintos de *firmware*, sendo eles API ou AT. Conforme Faludi (2011) os pacotes enviados no *firmware* API são mais indicados para "conversa" entre computadores pois é menos legível para humanos. Por outro lado dispositivos configurados como AT são mais preparados para interação humana, quando configurados desta maneira ainda existem mais duas possibilidades: o modo transparente e o modo de comando. O modo transparente é a configuração padrão de todos os módulos *Xbee* que utilizam *firmware*

AT. Esse modo é chamado de transparente pelo fato de apenas enviar a informação exatamente como recebe e é utilizado para enviar dados através dele para outro módulo remoto. O modo de comando é utilizado para situações opostas, quando não há interesse em enviar dados mas sim de se conectar diretamente com um módulo, seja para descobrir uma configuração ou para alterar o comportamento do módulo (FALUDI, 2011).

5.1.1.1 CONFIGURAÇÃO DOS MÓDULOS XBEE

Para comunicação o primeiro passo de configuração foi setar todos os módulos, inclusive o coordenador, no mesmo identificador de rede. A identificação de rede é um valor hexadecimal entre 0 e 0xFFFF e os *XBees* só podem se comunicar uns com os outros quando têm o mesmo identificador de rede. O segundo parâmetro é o endereço fonte MY, composto por 16 bits é o endereço do rádio. Outro detalhe importante de configuração é endereço de destino, composto pelos parâmetros Destination High DH e Destination Low DL, que são fundamentais para configuração pois determinam com quem cada módulo vai se comunicar. Essa configuração pode ser feita de dois jeitos:

- Definir o DH como 0 e o DL como o endereço MY do *Xbee* destino.
- Definir o DH como o Serial Number High SH e o DL como Serial Number Low SL do *Xbee* destino.

Os dois métodos funcionam bem, mas geralmente o mais utilizado é o primeiro por ser mais simples. Outro ponto importante é que quando DH e DL forem configurados como 0 o módulo só vai se comunicar com o coordenador da rede.

5.1.2 NÓS SENSORES

Em um primeiro momento os dois nós sensores da rede foram configurados como roteadores com *firmware* AT, visto que neste caso os módulos precisam apenas enviar a informação lida pelo sensor de temperatura LM35 para o nó central, configurá-los no modo transparente é o ideal. Além disso para que ambos roteadores estivessem enviando dados apenas para o nó central os endereços DH e DL foram setados como 0.

Cada nó sensor da rede é composto por um módulo *XBee*, responsável pela comunicação, ligado a um sensor LM35 que por sua vez faz a medição dos níveis de temperatura. Esta ligação

é feita em uma *protoboard*, utilizando fios *jumper* e uma fonte de alimentação de 5V. Na figura 25 pode ser observado um dos nós sensores montados para o projeto e em seguida uma explicação detalhada da configuração.

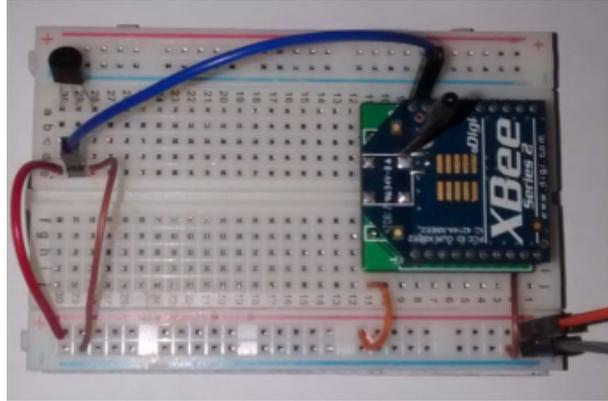


Figura 25: Nó sensor

Fonte: O Autor

Dois fios *jumper* conectados no pino de 5V do Arduino levam energia aos barramentos de alimentação da *protoboard*. O interessante de utilizar uma placa com esse tipo de barramentos é que ao invés de ter que conectar fios da fonte de alimentação até cada componente ligado, os *jumpers* podem ser conectados da própria *protoboard* até o componente através dos barramentos de alimentação.

O módulo *Xbee* está ligado através de um adaptador específico. Ele é alimentado com 5V e para receber os dados do sensor de temperatura o pino digital 3 do *Xbee* deve ser setado como conversor analógico-digital ADC, isto serve para que seja possível converter uma entrada analógica em um sinal digital de modo que seja fácil transmitir. Este é o pino responsável por receber a saída analógica do sensor LM35. Por fim é necessário definir o intervalo de tempo em que as informações de temperatura serão transmitidas, o parâmetro de configuração é o *IR, IO Sampling Rate*, que aceita valores hexadecimais, neste caso foi definido um intervalo de 30 segundos, em hexadecimal o valor equivalente é 7530.

5.1.3 NÓ CENTRAL

O nó central é composto por um *Xbee* ligado ao Arduino Uno conectado em um *Ethernet Shield* através de fios *jumper*. A ligação é feita de maneira simples, conectando um fio da porta TX do *Ethernet Shield*, conectado ao Arduino no pino RX do módulo *Xbee*. Da mesma forma outro fio ligando a porta RX no pino TX. Neste módulo foi instalado e configurado o *firmware Coordinator API*, que é mais indicado para facilitar a leitura dos dados recebidos pelo

coordenador na saída serial do Arduino.

Com todos os módulos conectados já é possível monitorar e trabalhar com os pacotes recebidos pelo coordenador através do Arduino. Foi desenvolvido um código para trabalhar com estes dados para que o pacote recebido possa ser lido pela conexão serial do Arduino, o que possibilita a análise e operação com essa informação. O algoritmo primeiro descarta as informações que não serão utilizadas e em seguida faz a leitura e envia para base de dados duas informações: a temperatura medida pelo sensor LM35 ligado ao módulo e também o endereço MY do nó sensor de onde os dados estão sendo recebidos. Conforme Rhysider (2012) o pacote *Xbee* quando trabalha com entrada e saída de dados utilizando *firmware* API possui 24 *bytes* de tamanho, começando do endereço 0x7e até o 0XF5. Este pacote trás diversas informações, como tamanho, tipo de dado, número serial do módulo que esta enviando o pacote para o coordenador entre outros.

A primeira informação obtida é o endereço MY que é utilizado para identificar de qual nó sensor o pacote está sendo enviado. Conforme Rhysider (2012), os bytes que trazem esse dado estão alocados nos endereços 0x36 e 0x6A. Para extrair essa informação o código é relativamente simples sendo necessário apenas percorrer o pacote até o endereço desejado, ler o dado fazendo uma conversão para hexadecimal e em seguida para texto e então concatenar fazendo a mesma conversão com o dado do endereço seguinte. Como o MY é único para cada nó da rede, obtendo o valor no pacote é possível utilizar uma estrutura condicional e então fazer a verificação de qual módulo está vindo o pacote.

Para obter valores de temperatura detectados pelo sensor LM35 e enviados do nó sensor para o nó central coordenador, são provenientes do pino configurado como ADC. Os dados ADC são armazenados no byte 22, sendo o endereço deste o 0x25. Este endereço sempre trás dois bytes de dados para cada pino do módulo que estiver configurado como ADC. A informação analógica é entregue dividida entre o LSB e MSB, sendo em uma tradução literal LSB o byte menos significativo e MSB o byte mais significativo. Com estes dois bytes devidamente lidos e armazenados, é necessário fazer uma conversão através de um cálculo matemático para obter o valor completo da leitura analógica. O cálculo utilizado foi retirado do guia escrito por Rhysider (2012).

Após obter a leitura analógica correta são necessários outros dois cálculos, sendo um para converter essa leitura em Volts e o outro para converter Volts em graus Celsius, sabendo que para isso é necessário saber que em uma saída analógica de sensores LM35 dez milivolts representam um grau Celsius. Uma vez que os dados são obtidos é necessário enviar para uma base de dados em um servidor. Para essa conexão, no Arduino é conectado um módulo *Ethernet*

Shield e a próxima seção trata sobre a ligação e envio para o servidor, onde os dados enviados serão armazenados. A leitura e os cálculos de conversão podem ser visualizados na figura 26.

```

if (Serial.available() >= 21) {
  if (Serial.read() == 0x7E) {
    for(int i = 1; i < 19; i++) {

      if(i == 12){ // informação do endereço MY
        myAddress = String(Serial.read(), HEX); // conversão
        myAddress = myAddress + String(Serial.read(), HEX);
        i = i+2;
      }

      byte discardByte = Serial.read(); // bytes descartados
    }

    // leitura da informação analógica do pino setado como ADC
    int analogMSB = Serial.read(); //byte mais significativa
    int analogLSB = Serial.read(); //byte menos significativa
    int analogReading = analogLSB + (analogMSB * 256); // cálculo

    // conversão para volts
    temp = analogReading / 1023.0 * 1.23;
    // conversão para temperatura, sabendo, 10mV = 1 C, 1mV = 0.01v
    temp = temp / 0.01;
  }
}

```

Figura 26: Cálculo de conversão leitura do pino ADC

Fonte: O autor

5.2 ENVIO DE DADOS

O envio de dados para o servidor é feito através de um módulo *Ethernet Shield* que é conectado ao Arduino e possui uma interface de rede cabeada. Conforme Arduino (2015) o *Ethernet Shield* conecta seu Arduino à internet em apenas alguns minutos e para isso basta ligar este módulo à placa Arduino e ligá-lo a rede com um cabo RJ45. A configuração inicial da do módulo é feita no código e está descrita a seguir.

Inicialmente foi importada a biblioteca *Ethernet.h* necessária para o funcionamento do *Ethernet Shield*. Para que a plataforma seja identificada na rede é necessário atribuir um endereço físico, também conhecido como MAC. Nenhum endereço vem configurado por padrão, portanto é preciso a declaração que pode ser de qualquer endereço válido que não esteja em uso na rede. Outra declaração necessária é do endereço IP do servidor.

5.2.1 CONEXÃO E ENVIO DE DADOS PARA SERVIDOR

Com a configuração do módulo *Ethernet Shield* somada a configuração de todas as tecnologias previamente explicadas, a plataforma está pronta para ser conectada ao servidor. Nessa etapa, foi utilizado como base o código apresentado no trabalho de Almeida, que trabalha com método *GET* para envio de dados para o servidor. Conforme Almeida (2015), é necessária uma verificação no estado da conexão antes do envio de dados. Caso a plataforma ainda não

esteja conectada essa conexão é feita através da porta 80 do servidor. Uma vez conectado os dados coletados podem ser enviados para o servidor via método *GET*. O autor ainda afirma que como a plataforma Arduino ainda é limitada nesse quesito é preciso escrever passo a passo cada parâmetro na URL. O trecho de código na figura 27 mostra como foi feita a verificação do estado do servidor e o envio dos dados coletados para o servidor.

```
// verificação da conexão com o servidor
if (client.connect(server, 80)) {
  Serial.println("Connected");

  // envio via get
  client.print("GET /tcc2/add_data.php?");
  Serial.print("GET /tcc2/add_data.php?");

  client.print("temperatura=");
  Serial.print("temperatura=");
  client.print(temp);
  Serial.print(temp);
  client.print("&");
  Serial.print("&");
  client.print("sensor=");
  Serial.print("sensor=");
  client.print(sensor);
  Serial.print(sensor);

  client.println(" HTTP/1.0");
  Serial.println(" HTTP/1.0");

  client.println();
  client.stop();
} else {
  Serial.println("Cannot connect to server");
}
```

Figura 27: Código: Conexão e envio de dados

Fonte: O autor

5.3 SERVIDOR E APLICAÇÃO WEB

Nesta seção serão apresentados os métodos de conexão e inserção de dados na base e também a aplicação web desenvolvida para facilitar a visualização das informações de temperatura. O servidor em questão tem instalado o sistema operacional Ubuntu, na versão 14.04, o SGBD MySQL e outras ferramentas necessárias que já o sistema já tem na instalação padrão.

5.3.1 CONEXÃO COM A BASE DE DADOS

No lado servidor foi configurada uma base de dados utilizando o SGBD MySQL e desenvolvidas duas páginas *php*, sendo uma responsável pela conexão com o banco de dados e outra pela execução da *query* de inserção dos dados no banco. O código é simples e utiliza as próprias funções do *php* para as operações, na página de conexão com o banco são passados como parâmetro para função *mysqli_connect()* os dados de acesso do banco de dados, sendo

estes respectivamente o endereço, usuário, senha e por último o banco de dados desejado. Na figura 28 pode ser visualizado o código da página de conexão.

```
<?php
$conn = mysqli_connect("localhost", "root", "_secret", "tcc2")
or die(mysqli_error("Error connecting"));
?>
```

Figura 28: Página php: Conexão com banco de dados

Fonte: O autor

Caso a conexão não seja estabelecida irá retornar um erro. A página de inserção de dados é igualmente simples, foi utilizada a função *include()* para reaproveitar o código da página de conexão e também a função *mysqli_query()* para executar a *query* definida com os parâmetros recebidos pelo método *GET*. O código de inserção pode ser observado na figura 29.

```
<?php
include("dbconnect.php");

$sql = "INSERT INTO tcc2.leituras (temperatura ,sensor) VALUES ('".$_GET["
temperatura"]."', '".$_GET["sensor"]."')";

mysqli_query($conn, $sql);

header("Location: index.php");
?>
```

Figura 29: Página php: Inserção de dados

Fonte: O autor

5.3.2 APLICAÇÃO WEB

Para facilitar a visualização das informações lidas pelos sensores, foi desenvolvida uma aplicação web utilizando o *framework* Laravel. O Laravel é um Framework PHP utilizado para o desenvolvimento web, que utiliza a arquitetura MVC e tem como principal característica ajudar a desenvolver aplicações seguras e performáticas de forma rápida, com código limpo e simples, já que ele incentiva o uso de boas práticas de programação e utiliza o padrão PSR-2 como guia para estilo de escrita do código (ADRIEL, 2015). Além disso, para o desenvolvimento da interface gráfica foi utilizado o Bootstrap, um *framework* que facilita a criação de aplicações responsivas e possui diversos componentes gráficos (BOOTSTRAP, 2016).

A página foi desenvolvida com foco em responsividade e simplicidade, de modo que seja fácil visualizar os dados de temperatura bem como em qual sensor a informação foi coletada. A aplicação possui apenas um página o que torna possível visualizar os dados de todos os sensores no mesmo lugar e foi desenvolvida de modo que caso mais nós sensores forem adicio-

dados serão disponibilizados na mesma página. A Figura 30 mostra o resultado final da página web.

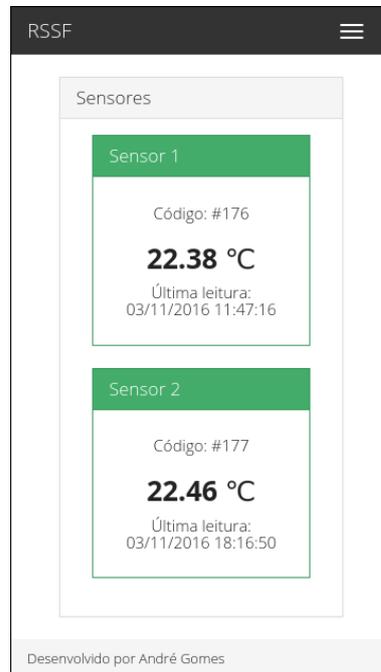


Figura 30: Página web: Layout responsivo

Fonte: O autor

5.4 FLUXO DE DADOS

Nesta seção serão apresentados o fluxo de dados do projeto como um todo e os esquemas desenvolvidos utilizando a ferramenta *Fritzing* que representam visualmente a configuração de ligações entre fios *jumper* e módulos descrita nos capítulos anteriores. A imagem 31 representa o nó central da rede e pode ser vista a seguir.

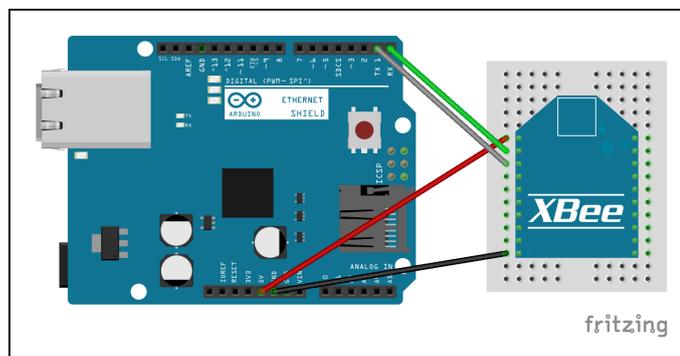


Figura 31: Nó central da rede feito no Fritzing

Fonte: O autor

Já o esquema mostrado na figura 32 representa os nós sensores, demonstrando características como fonte de alimentação utilizada e o sensor LM35, além das ligações via *jumper*. Tanto as ligações feitas na figura 31 quanto na 32 a seguir são exatamente as mesmas utilizadas no projeto. Os pinos utilizados foram configurados previamente via *software* XCTU e ou já possuem configuração padrão, como alimentação, TX e RX por exemplo.

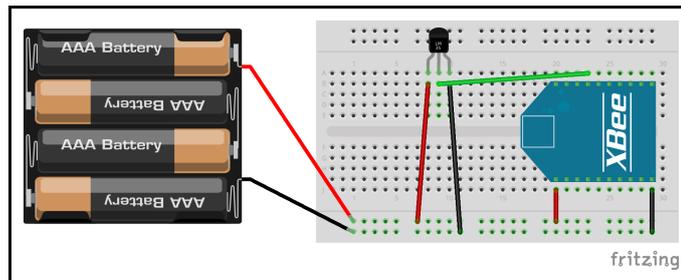


Figura 32: Nó sensor da rede feito no Fritzing

Fonte: O autor

O fluxo de dados do projeto funciona da seguinte forma: Os nós sensores são responsáveis pela medição dos níveis de temperatura do ar, capturados pelo sensor LM35 conectado a cada *XBee Antenna*. Assim que os dados forem captados serão enviados ao nó central, se comunicando através do módulo *XBee Antenna*. Quando nó central receber estes dados irá enviá-los ao servidor, onde serão armazenados na base de dados com o SGBD MySQL. Feito isso, os dados estarão prontos para serem utilizados pela página web, onde as informações sobre temperatura poderão ser visualizadas. A Figura 33 mostra um diagrama geral do funcionamento da RSSF.

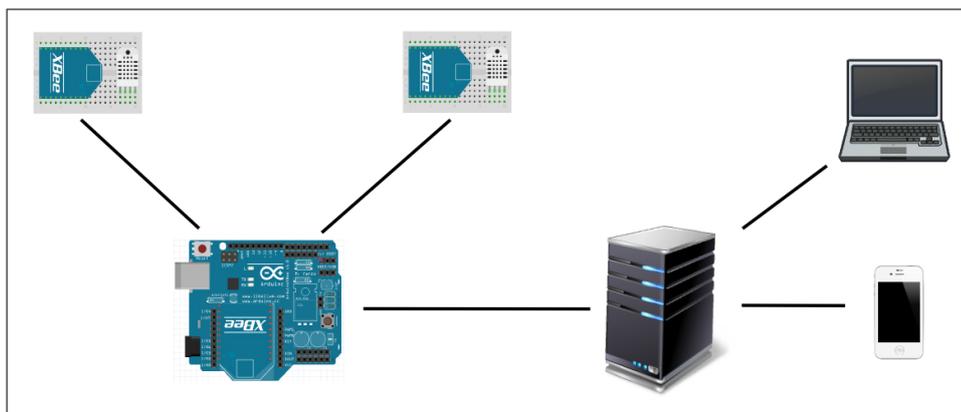


Figura 33: Esquema geral da RSSF

Fonte: O autor

6 TESTES E RESULTADOS

Neste capítulo serão descritos os testes realizados para validação da plataforma e os resultados obtidos.

6.1 TESTES

Para testar a plataforma foram definidos dois cenários diferentes para poder visualizar o comportamento dos sensores e comparar as temperaturas obtidas e ambos estão descritos a seguir:

- Dois nós sensores em pontos diferentes de um ambiente interno fechado.
- Dois nós sensores em ambientes separados, sendo um ambiente interno fechado e outro em área externa.

O primeiro caso de testes foi configurado para verificar se os dois nós sensores estão calibrados corretamente, pois se os dados obtidos dentro de um mesmo ambiente com a temperatura equilibrada (sem interferência externa ou de equipamentos como ar-condicionado e ventiladores) forem muito distoantes, isso pode significar que os sensores não estão medindo a temperatura com precisão.

No dia 14 de Novembro a plataforma foi preparada, com os nós sensores distantes 5 metros entre si e a 3 metros do nó central da rede dentro de um ambiente e ficou coletando dados das 20:50 às 21:55. Nesse período foi possível salvar os dados de temperatura em graus Celsius a cada 5 minutos para fazer uma análise a respeito do funcionamento da plataforma e verificar se os sensores estão calibrados corretamente. Os resultados foram avaliados e de modo geral a RSSF funcionou sem problemas, passando as informações a partir dos nós sensores para o nó central da rede e posteriormente enviando e salvando na base de dados, além de mostrar na aplicação web. Quanto a comprovação de que os sensores LM35 estão medindo a temperatura corretamente, os resultados obtidos foram comparados com a temperatura lida pelo DHT11 e o

resultado foi positivo, visto que tanto os dados enviados pelos dois sensores LM35 quanto pelo DHT11 foram muito próximos. Os dados obtidos podem ser observados na tabela a seguir.

Tabela 16: Comparação em Ambiente Interno

Sensor 1		Sensor 2		DHT 11
Temperatura	Horário	Temperatura	Horário	Temperatura
18.97	20:50:06	19.02	20:50:07	19.16
18.85	20:55:06	19.04	20:55:07	19.11
18.79	21:00:06	19.09	21:00:07	19.05
18.81	21:05:06	19.12	21:05:09	19.14
18.92	21:10:09	19.16	21:10:09	19.21
18.95	21:15:09	19.18	21:15:09	19.17
18.97	21:20:11	19.14	21:20:09	19.25
18.99	21:25:11	19.18	21:25:09	19.34
19.05	21:30:11	19.02	21:30:12	19.38
19.12	21:35:13	19.14	21:35:12	19.28
19.16	21:40:13	19.19	21:40:12	19.25
19.16	21:45:13	19.05	21:45:14	19.22
19.09	21:50:13	19.08	21:50:14	19.19
19.04	21:55:16	19.11	21:55:14	19.15

O segundo caso de teste foi pensado para verificar se a plataforma está trabalhando corretamente fazendo uma análise da diferença dos resultados obtidos em ambiente interno e externo. O teste foi realizado também no dia 14 de Novembro com a diferença que o nó sensor identificado como S2 no teste anterior foi movido para um ambiente externo. O sensor DHT11 também foi movido, colocado próximo ao nó sensor no ambiente externo e foi utilizado para proporcionar um parâmetro de comparação dos resultados.

A plataforma coletou dados das 22:15 às 23:10 e a RSSF operou sem nenhum tipo de problema. O nó sensor S1 foi mantido a 3 metros e o nó S2 foi colocado a aproximadamente 7 metros da central. Como esperado, o sensor S1 manteve uma leitura próxima do teste anterior. Já a média de temperatura obtida pelo sensor S2 foi 1.61 graus Celsius menor do que a leitura do sensor S1, assim como a média de temperatura lida pelo sensor DHT11 foi 1,81 menor que o S1. Estes resultados mostram que os sensores estão medindo níveis de temperatura com precisão mesmo em ambientes diferentes. Os dados medidos podem ser visualizados na tabela a seguir.

Tabela 17: Comparação Ambiente Interno x Ambiente Externo

Sensor 1		Sensor 2		DHT 11
Temperatura	Horário	Temperatura	Horário	Temperatura
19.02	22:15:06	17.65	22:15:08	17.36
19.11	22:20:06	17.68	22:20:08	17.41
19.14	22:25:06	17.62	22:25:08	17.55
19.08	22:30:06	17.51	22:30:08	17.42
19.01	22:35:07	17.45	22:35:10	17.48
19.38	22:40:07	17.69	22:40:10	17.35
19.17	22:45:07	17.44	22:45:10	17.23
19.05	22:50:07	17.55	22:50:10	17.44
19.28	22:55:07	17.61	22:55:10	17.06
19.44	23:00:07	17.77	23:00:11	17.24
19.02	23:05:07	17.18	23:05:11	17.08
19.04	23:10:07	17.21	23:10:11	17.44

7 CONSIDERAÇÕES FINAIS

Nesta seção serão descritas as principais dificuldades encontradas durante o desenvolvimento do projeto, as contribuições do projeto para sociedade e para comunidade acadêmica e sobre os trabalhos que podem ser feitos a partir deste.

7.1 DIFICULDADES ENCONTRADAS

As principais dificuldades encontradas foram na configuração da rede de sensores sem fio, precisamente, na configuração dos módulos *Xbee*. Por ser uma tecnologia relativamente nova existem poucos trabalhos semelhantes para utilizar como referencial. Além disso, o principal motivo deste desafio ter sido grande é a falta de material que trate sobre a parte prática de configuração dos módulos *Xbee*, que foi a etapa mais extensa do projeto. A maior parte dos livros e artigos encontrados ainda não são traduzidos para o Português, sendo em sua maioria escritos na língua inglesa. Contudo, mesmo os bons materiais encontrados escritos em Inglês tratam na maior parte da teoria, deixando a desejar no que diz respeito a como utilizar e configurar estes módulos.

7.2 REAFIRMAÇÃO DA CONTRIBUIÇÃO

A grande contribuição deste trabalho é disponibilizar um referencial teórico e principalmente prático de configuração de redes de sensores sem fio que utilizem o protocolo *Zigbee* através de módulos *Xbee*. Além disso, o monitoramento de temperaturas pode ajudar a evitar problemas de desconforto térmico e queda de desempenho no ambiente de trabalho.

7.3 TRABALHOS FUTUROS

Esta seção tem por objetivo apresentar melhorias futuras para a plataforma.

- **Utilização de mais sensores:** atualmente a plataforma possui apenas um tipo de sensor aferindo informações nos nós sensores, principalmente pela questão de custo visto que o projeto foi desenvolvido com recursos próprios. Ligar mais sensores para aferir diferentes informações como umidade e presença de gases seria um grande diferencial para o projeto.
- **Mineração de dados climáticos:** a plataforma já está pronta para ser ligada por longos períodos. Diante disso, quando a quantidade de dados climáticos for grande o suficiente, pode ser adicionada a plataforma uma ferramenta de mineração de dados para extrair informações relevantes e padrões climáticos.

REFERÊNCIAS

- ADAFRUIT. **DHT temperature and humidity sensors**. 2012. Acesso em: 11 Nov. de 2015. Disponível em: <<https://learn.adafruit.com/dht>>.
- ADRIEL, W. **Introdução ao Laravel Framework PHP**. 2015. Acesso em: 29 Out. de 2016. Disponível em: <<http://www.devmedia.com.br/introducao-ao-laravel-framework-php/33173>>.
- AKYILDIZ, I. F.; KASIMOGLU, I. H. Wireless sensor and actor networks: research challenges. **Science Direct**, Elsevier, v. 74, n. 4, p. 23, 2004.
- ALMEIDA, T. F. de. Dioxiduino: Uma plataforma para o monitoramento da emissão de co2 em ambientes internos. Universidade Tecnológica Federal do Paraná, p. 46, 2015.
- ARDUINO. **What is Arduino?** 2015. Acesso em: 11 Nov. de 2015. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>.
- BOOTSTRAP. **Bootstrap**. 2016. Acesso em: 03 Nov. de 2016. Disponível em: <<http://getbootstrap.com/>>.
- CEPAGRI. **Escala psicrométrica Unicamp para indicação de níveis de umidade relativa do ar prejudiciais à saúde humana**. 2009. Acesso em: 14 Set. 2015. Disponível em: <<http://www.cpa.unicamp.br/artigos-especiais/umidade-do-ar-saude-no-inverno.html>>.
- DIGI. **XCTU**. 2015. Acesso em: 28 Out. de 2016. Disponível em: <<https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>>.
- ESCHNER, R. H. Sistema de automação residencial baseado em sensores zigbee. Universidade Federal do Rio Grande do Sul, p. 38, 2011.
- FALUDI, R. **Building Wireless Sensor Networks**. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'REILLY, 2011.
- FARAHANI, S. **Zigbee Wireless Networks and Transceivers**. Burlington, MA 01803, USA: Elsevier, 2008.
- FILHO, L. H. S. Desenvolvimento de uma rede de sensores sem fio para aplicacoes diversas. Trabalho de conclusão de curso Universidade de Pernambuco, p. 80, 2011.
- FRITZING. **Fritzing**. 2016. Acesso em: 03 Nov. de 2016. Disponível em: <<http://fritzing.org/home/>>.
- HACKMANN, G. 802.15 personal area networks. Washington University in St. Louis, p. 8, 2006.
- IAN, F. A. et al. A survey on sensor networks. **IEEE Communications Magazine**, Georgia Institute Technology, v. 9, n. 4, p. 102–114, 2002.

- IIDA, I. **Ergonomia Projeto e produção**. São Paulo, BR: Edgard Blucher, 2005.
- RAMOS, J. de S. B. **Instrumentação Eletrônica Sem Fio**. [S.l.]: Editora Érica, 2012.
- RHYSIDER, J. **XBee S2 Quick Reference Guide**. 2012. Acesso em: 28 Out. de 2016. Disponível em: <<https://www.tunnelsup.com/xbee-guide/>>.
- SHAH, A. **LM35 Temperature Sensor**. 2015. Acesso em: 28 Out. de 2016. Disponível em: <<http://www.dnatechindia.com/LM35-Temperature-Sensor-Basics.html>>.
- SILVA, T. L.; ALMEIDA, V. d. C. **Influência do calor sobre a saúde e desempenho dos trabalhadores**. 2010. In: SIMEPRO, IV., 2010, Maringá. **Anais eletrônicos...** Maringá: SIMEPRO, 2010. Acesso em: 9 set. 2015. Disponível em: <<http://www.dep.uem.br/simepro/4/files/artigos/1283095871.pdf>>.

ANEXO A – CÓDIGO FONTE ARDUINO (NÓ CENTRAL)

```

#include <Ethernet.h>

byte mac[] = { 0x54, 0x34, 0x41, 0x30, 0x30, 0x31 };

IPAddress server(192,168,1,5);
EthernetClient client;

float temp;
String myAddress;
String sensor;

void setup() {
  Serial.begin(9600);
  Ethernet.begin(mac);
  Serial.print("IP Address: ");
  Serial.println(Ethernet.localIP());
}

void loop() {

  if (Serial.available() >= 21) {

    if (Serial.read() == 0x7E) {
      for(int i = 1; i < 19; i++) {

        if(i == 12){
          myAddress = String(Serial.read(), HEX);
          myAddress = myAddress + String(Serial.read(), HEX);

```

```
        i = i+2;
    }

    byte discardByte = Serial.read();
}

int analogMSB = Serial.read();
int analogLSB = Serial.read();
int analogReading = analogLSB + (analogMSB * 256);

temp = analogReading / 1023.0 * 1.23;
temp = temp / 0.01;

if(myAddress == "5a4a")
    sensor = "S1";
else
    sensor = "S2";

if (client.connect(server , 80)) {
    Serial.println("Connected");

    client.print("GET_/tcc2/add_data.php?");
    Serial.print("GET_/tcc2/add_data.php?");

    client.print("temperatura=");
    Serial.print("temperatura=");
    client.print(temp);
    Serial.print(temp);
    client.print("&");
    Serial.print("&");
    client.print("sensor=");
    Serial.print("sensor=");
    client.print(sensor);
    Serial.print(sensor);
}
```

```
client.println(" HTTP/1.0");  
Serial.println(" HTTP/1.0");  
  
client.println();  
client.stop();  
} else {  
  Serial.println("Cannot connect to server");  
}  
  
delay(5000);  
}  
}  
}
```