

FACULDADE DE TECNOLOGIA DE OSASCO - PREFEITO HIRANT SANAZAR



TCC53

Faculdade de Tecnologia de Osasco
Biblioteca

ISAQUE TEIXEIRA LIMA SILVA
MATHEUS PIRES DE OLIVEIRA FERREIRA

SISTEMA DE MONITORAMENTO DE TEMPERATURA DO AMBIENTE ESCOLAR
POR MEIO DA REDE DE COMPUTADORES

OSASCO

2022

ISAQUE TEIXEIRA LIMA SILVA
MATHEUS PIRES DE OLIVEIRA FERREIRA

SISTEMA DE MONITORAMENTO DE TEMPERATURA DO AMBIENTE ESCOLAR
POR MEIODA REDE DE COMPUTADORES

Trabalho de Graduação como um dos requisitos
para a obtenção do título de Tecnólogo em Redes
de Computadores na Faculdade de Tecnologia
Prefeito Hirant Sanazar(Fatec Osasco).

Orientador: Fabio Brussolo de Oliveira

Dados Internacionais de Catalogação na Publicação (CIP)

SILVA, Isaque Teixeira Lima e FERREIRA, Matheus Pires de Oliveira

Sistema de Monitoramento de Temperatura do Ambiente Escolar Por Meio da Rede de Computadores. – Osasco, 2022
81f.

Orientador: Prof. Fabio Brussolo

Trabalho de Graduação – Curso de Tecnologia em Redes de Computadores -
FATEC Osasco: Prefeito Hirant Sanazar, 2022.

Sistema de Monitoramento de Temperatura do Ambiente Escolar Por Meio da Rede de Computadores I. Isaque Teixeira Lima Silva e Matheus Pires de Oliveira Ferreira. I. BRUSSOLO, F. II. Monitoramento, temperatura de ambientes, diminuição do calor.

ISAQUE TEIXEIRA LIMA SILVA
MATHEUS PIRES DE OLIVEIRA FERREIRA

SISTEMA DE MONITORAMENTO DE TEMPERATURA DO AMBIENTE ESCOLAR
POR MEIO DA REDE DE COMPUTADORES

Trabalho de Graduação como um dos requisitos
para a obtenção do título de Tecnólogo em Redes
de Computadores na Faculdade de Tecnologia
Prefeito Hiran Sanazar(Fatec Osasco).

Cidade, __ de _____ de ____

BANCA EXAMINADORA

Prof. Dr.

Universidade

Prof. Dr.

Universidade

Prof. Dr.

Universidade

Dedico este trabalho aos meus pais e amigos que
sempre me incentivaram através desta jornada.

AGRADECIMENTOS

Agradeço aos meus professores e colegas por me ajudarem a desenvolver este trabalho.

"Nenhuma grana no mundo compra um segundo de tempo." (Tony Stark)

RESUMO

O objetivo deste projeto, é mostrar como a rede de computadores pode ser utilizada para que os dados sejam enviados por meio de um dispositivo com Internet para o computador com o sistema de monitoramento. Para isso, alguns dispositivos são necessários para criar este sistema. O Arduino Uno é a peça principal para o sistema, pois se trata de um microcontrolador que será capaz tanto de fazer com que o ventilador seja acionado, quanto enviar os dados pela rede. A ferramenta Zabbix será utilizada também para que o monitoramento da temperatura do ambiente seja feito a partir de um sensor de temperatura que estará junto do Arduino. Outras peças também serão necessárias para a aplicação do projeto, que nos próximos capítulos serão mostradas.

Palavras-chave: Redes de Computadores; IOT; Monitoramento; Arduino; Tinkercad.

ABSTRACT

The objective of this work, is to show how the computer network can be used so that the data is sent through a device with Internet to the computer with the monitoring system. For that, some devices are needed to create this system. The Arduino Uno is the main part of the system, as it is a microcontroller that will be able to both make the fan turn on and send data over the network. The Zabbix tool will also be used to monitor the temperature of the environment, from a temperature sensor that will be next to the Arduino. Other parts will also be necessary for the application of the work, which in the next chapters will be shown.

Keywords: Computer Network; IOT; Monitoring; Arduino; Tinkercad.

LISTA DE ILUSTRAÇÕES

Figura 1 — Arduino Uno	14
Figura 2 — Conectores de Alimentação do Arduino	15
Figura 3 — Pinos de Entradas Analógicas	17
Figura 4 — Pinos de Entradas e Saídas Digitais	17
Figura 5 — Arduino Shield Internet.....	17
Figura 6 —Zabbix	20
Figura 7 — Simulação Tinkercad.....	20
Figura 8 — Simulação Tinkercad.....	21
Figura 9 — Temperatura alta, LED vermelho se acende.....	21
Figura 10 — Temperatura baixa, LED azul se acende	22
Figura 11 — Cooler Fan	22

LISTA DE ABREVIATURAS E SIGLAS

TCC	Trabalho de Conclusão de Curso
LED	Light Emitting Diode - (Diodo Emissor de Luz)
OHM	Unidade de Medida da Resistência Elétrica
TCP	Transmission Control Protocol (Protocolo de Controle de
Transmissão)	

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Hipótese de pesquisa	12
1.2	OBJETIVO GERAL	12
1.3	OBJETIVOS ESPECÍFICOS	13
2	REFERENCIAL TEÓRICO	14
2.1	COMPONENTES	14
3	ZABBIX	18
4	TINKERCAD	20
5	O QUE É A REDE DE COMPUTADORES ?	23
6	MODELO TCP/IP	24
6.1	SNMP	26
6.2	C++	27
7	METODOLOGIA	28
8	CONSIDERAÇÕES FINAIS	30
9	REFERÊNCIAS	32
10	APÊNDICE	36

1 INTRODUÇÃO

A Internet das Coisas, ou IoT, é um sistema de dispositivos de computação inter-relacionados, máquinas mecânicas e digitais, objetos, animais ou pessoas que recebem identificadores exclusivos Unique Identifier (UIDs) e a capacidade de transferir dados através de uma rede, sem a necessidade de humanos para interação humano ou humano para computador. Uma “coisa” na internet das coisas, pode ser uma pessoa com um implante de monitor cardíaco, um animal de fazenda com um transponder de biochip, um automóvel que possui sensores embutidos para alertar o motorista quando a pressão dos pneus estiver baixa ou qualquer outro produto natural ou artificial objeto que pode receber um endereço Internet Protocol e que pode transferir dados pela rede.

O termo Internet das Coisas, ou IOT (Internet of Things) em inglês, foi primeiramente apresentado por Kevin Ashton da Massachusetts Institute of Technology, em uma apresentação sobre Radio Frequency Identification (RFID) para a Procter & Gamble (P & G) em 1999. (Ashton, 2009). Mesmo que este termo tenha aparecido em 1999, existe uma ideia parecida do conceito de Internet das Coisas mesmo não sendo chamado desta forma. Para Venkatesh (1996, p. 47), que estudou a mudança do uso da computação para fins de trabalho, para uma computação que seria utilizada em casa, com serviços mais “fáceis” de se usar, com aplicativos com interfaces menos poluídas que não necessitavam muito conhecimento técnico para se usar, previu que no futuro poderiam existir casas que fariam as tarefas automaticamente, como por exemplo, fazer as compras para manter os suprimentos ou fazer uma refeição. De forma geral, ainda de acordo com Kevin Ashton, a Internet das Coisas busca diminuir a interação humana fazendo com que sejamos capazes de monitorar e mensurar tudo, reduzindo o desperdício, as perdas e o custo.

Como por exemplo, hoje em dia existem geladeiras que já possuem conexão com a internet, possibilitando que ela seja uma “coisa” da Internet das Coisas, podendo mostrar para o usuário por meio de uma tela embutida nela, quais são os mantimentos que estão na geladeira etc. O futuro da Internet das Coisas parece

bem promissor com tudo o que ela pode oferecer para casas, empresas, cotidiano etc.

Tendo isso em mente, de acordo com o hospital Albert Einstein, hospital das clínicas e outros hospitais da área da saúde por exemplo, os institutos tem apostado em soluções tecnológicas que ajudam no monitoramento de temperatura, e outras grandezas também. A IoT pode ajudar os profissionais da saúde por exemplo, a manter o controle da temperatura. Sejam elas nas geladeiras, câmara, caixas térmicas, ambientes etc. Imagine receber um alerta pelo celular quando a porta de uma câmara estiver aberta ou disparar alarmes de baixa e/ou alta temperatura para sinalizar a ocorrência de falta de energia elétrica? Acredite, isso é possível usando soluções com IoT.(Sensorweb, 2015). Resumindo, a Internet das Coisas otimiza a atuação humana, associando as tecnologias ao treinamento da equipe que trabalha com o monitoramento de temperaturas (e de outras grandezas também). Mas a questão é, como a IOT pode auxiliar no monitoramento de temperatura de ambientes?

1.1 HIPÓTESE DE PESQUISA

Em salas de aula nos períodos de calor, os ventiladores precisam ser acionados manualmente pelos alunos. E se ao invés disso, os próprios ventiladores ligassem sozinhos pois sabem que naquele ambiente está com uma temperatura mais alta do que o normal ? Por meio do conhecimento adquirido, iremos desenvolver uma solução para o monitoramento e controle de temperatura em sala de aula utilizando redes de computadores. Com as pesquisas iniciais, encontramos dois projetos para ter em mente como iniciar esse atual trabalho, ambos com controle de temperatura e internet. A ideia é utilizar um Arduino, junto com um sensor de temperatura e um fan, para monitorar a temperatura do ambiente e enviar esses dados através de um aplicativo chamado Zabbix, e a partir desses dados, fazer o controle da temperatura.

1.2 OBJETIVO GERAL

Um projeto de redes de computadores que incorpora um dispositivo IOT, que monitora a temperatura de um ambiente educacional em períodos de calor para que o ventilador seja acionado automaticamente, e que os dados da temperatura sejam enviados através da rede com uma ferramenta de monitoramento que será capaz de mostrar como estava a temperatura quando o ventilador foi acionado.

1.3 OBJETIVOS ESPECÍFICOS

Existem três tópicos importantes para o desenvolvimento desse projeto, os objetivos específicos são:

- Usar a Rede de Computadores para enviar os dados para a ferramenta de monitoramento;
- Configurar o Arduino junto do sensor de temperatura para que ele acione o ventilador automaticamente sem a necessidade manual;
- Fazer o monitoramento da temperatura do ambiente educacional em períodos de calor.

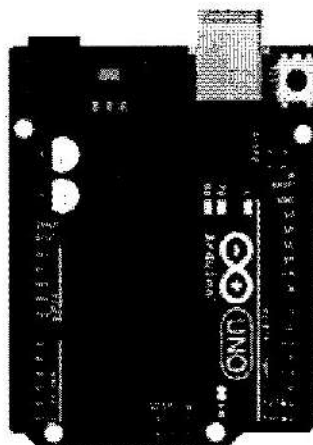
2 REFERENCIAL TEÓRICO

Os seguintes itens serão utilizados no nesse trabalho, o software Arduino Uno Zabbix para comunicação entre o Arduino Shield e Zabbix, a ferramenta Zabbix para monitoramento dos recursos. Na parte de hardware será utilizado um Arduino uno R3 para controle dos componentes, um Arduino Shield para realizar a conexão com a internet, um sensor de temperatura TMP36 para fazer o controle de um ventilador, 3 resistores de 220 Ω para gerenciamento do recurso, uma ventoinha de baixa potência para simular um ventilador, um protoboard para acoplar o circuito, uma fonte de alimentação para o projeto e cabos jumpers para protoboard e ligação dos itens.

2.1 COMPONENTES

De acordo com Michael McRoberts que escreveu o livro Arduino básico, foi utilizado nesse trabalho para principal item para fazer o projeto,o Arduino UNO, ele irá fazer o controle de todos os outros componentes.O Arduino é um microcontrolador com entradas e saídas digitais, programado na linguagem C++, conforme mostrado na Figura 01.

Figura 1 — Arduino Uno



Fonte: Página ARDUINO - ENTRADAS E SAÍDAS DIGITAIS - PINOS DIGITAIS

Disponível em: <https://portal.vidadesilicio.com.br/entradas-e-saidas-digitais/>

Conforme exibido na figura, a placa Arduino UNO possui 14 pinos que podem ser usados como entrada ou saída digitais. Estes Pinos operam em 5 V, onde cada pino pode fornecer ou receber uma corrente máxima de 40 mA. Cada pino possui resistor de pull-up interno que pode ser habilitado por software.

O Arduino possui entradas e saídas digitais, e os pinos para alimentação conforme mostrado na Figura 2, 3 e 4.

Figura 2 — Conectores de Alimentação do Arduino



Fonte: Página Arduino UNO

Disponível em: <https://www.embarcados.com.br/arduino-uno/>.

O Ethernet Shield W5100 que será mostrado mais abaixo, é compatível tanto com o Arduino Uno e Arduino Mega e possui um slot para cartão micro-SD. Usar o Ethernet Shield é uma forma simples de disponibilizar online, dados referentes a sensores ou sistemas que você controle usando um Arduino. A biblioteca Ethernet da IDE oferece todos os recursos necessários para que dados lidos com o Arduino possam ser acessados online.

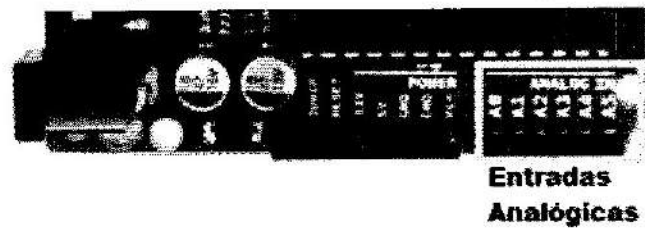
Do ponto de vista de hardware, trata-se de uma conexão simples. Basta encaixar o Shield no seu Arduino UNO ou Mega e ter em mãos um cabo ethernet com conetores RJ45 para ligar no seu modem/roteador. Primeiro, serão mostradas as características do Arduino Uno para que depois sejam mostradas as características do Shield.

O Arduino apresenta as seguintes características:

- IOREF – O primeiro pino, fornece uma tensão de referência para que Shields possam selecionar o tipo de interface apropriada, dessa forma Shields que funcionam com a placas Arduino que são alimentadas com 3,3V. podem se adaptar para ser utilizados em 5V. e vice-versa;
- RESET - pino conectado a pino de RESET do microcontrolador. Pode ser utilizado para um reset externo da placa Arduino;
- 3,3 V. - Fornece tensão de 3,3V. para alimentação de Shields e módulos externos. Corrente máxima de 50 mA;
 - 5 V - Fornece tensão de 5 V para alimentação de Shields e circuitos externos;
 - GND - pinos de referência, terra;
 - VIN - pino para alimentar a placa através de Shield ou bateria externa. Quando a placa é alimentada através do conector Jack, a tensão da fonte estará nesse pino.

O Arduino possui entradas analógicas que vão do pino A0 ao A5, onde cada uma tem a resolução de 10 bits. Por padrão a referência do conversor AD está ligada internamente a 5V, ou seja, quando a entrada estiver com 5V o valor da conversão analógica digital será 1023. A Figura 3 mostra este detalhe.

Figura 3 — Pinos de Entradas Analógicas



Fonte: Página Arduino UNO

Disponível em: <https://www.embarcados.com.br/arduino-uno/>.

O Arduino também possui entradas digitais, conforme mostra a Figura 4.

Figura 4 — Pinos de Entradas e Saídas Digitais



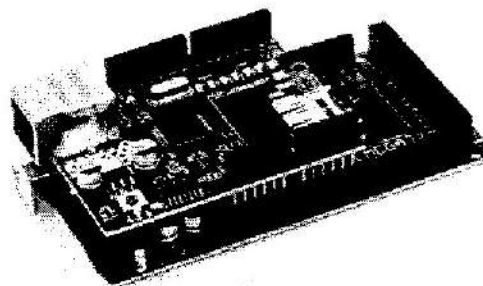
Fonte: Página Arduino UNO

Disponível em: <https://www.embarcados.com.br/arduino-uno/>.

PWM: 3,5,6,9,10 e 11 podem ser usados como saídas PWM de 8 bits através da função `analogWrite()`.

Outro item que será utilizado é o Arduino Shield, que como foi dito anteriormente, é uma extensão do Arduino para que seja utilizada mais funções neles, e sem descartar as pinagens do microcontrolador conforme mostra a Figura 5.

Figura 5 — Arduino Shield Internet



Fonte: Página Arduino UNO

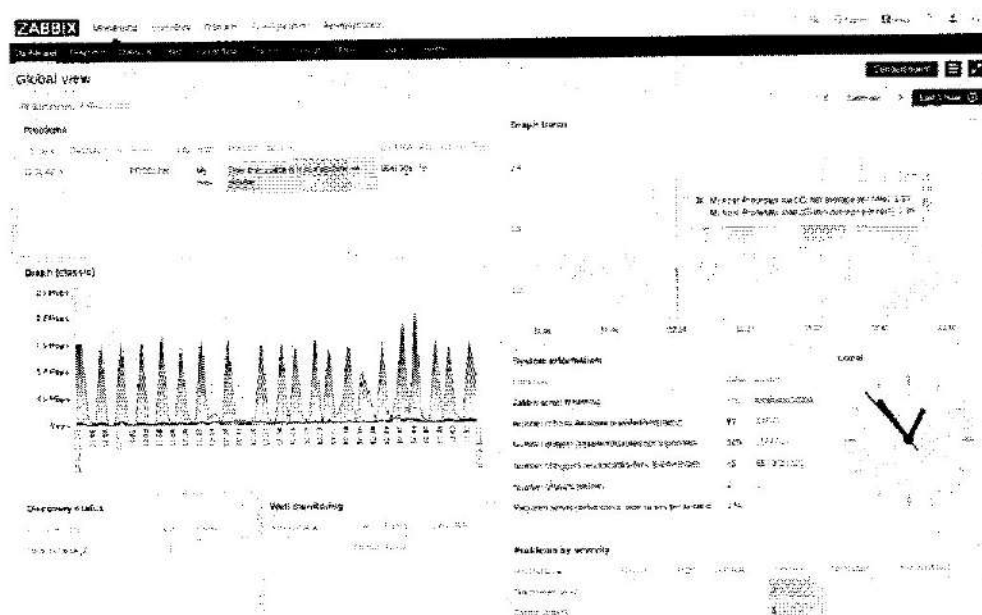
Disponível em: <https://www.embarcados.com.br/arduino-uno/>.

3 ZABBIX

De acordo com Janssen dos Reis Lima no livro Monitoramento de Redes com Zabbix, o Zabbix é uma ferramenta que pode ser utilizada para monitorar desde uma infraestrutura de rede até aplicações diversas, é uma das ferramentas mais utilizadas nesse meio, os dois integrantes do grupo já trabalharam com essa ferramenta graças a um trabalho realizado no terceiro semestre, e pela afinidade com essa ferramenta, foi decidido que usaríamos ela, o funcionamento dessa ferramenta é bem simples, em um sistema Linux fica o servidor Zabbix, aonde ele será instalado, e em outro computador ou aplicação, ficará o sistema que será monitorado.

O Zabbix nesse projeto terá a função importar os dados do monitoramento de temperatura, e avisar o operador via SMS, que a temperatura aumentou e o ventilador foi ligado, ou quando a temperatura diminuir, o ventilador será desligado. Essa integração irá acontecer com o Arduino Shield Ethernet, que permite que o sistema tenha acesso a internet, e possa enviar os dados para o a ferramenta Zabbix.

Figura 6 – Interface Zabbix



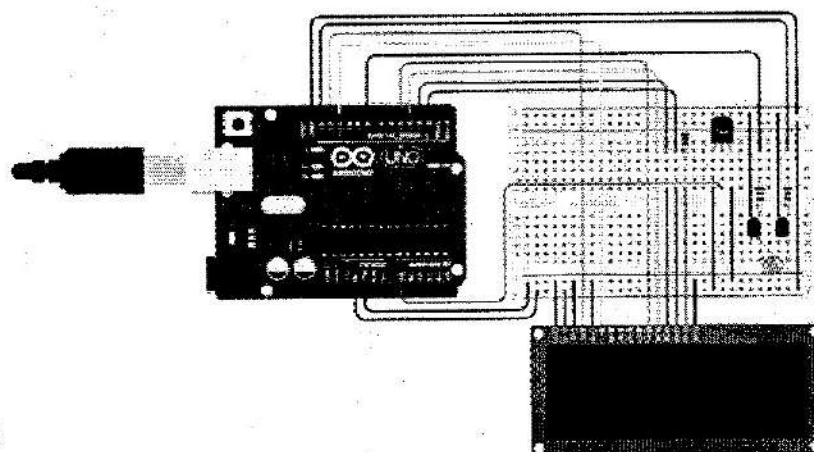
Fonte: <https://www.zabbix.com/documentation/4.0/>

Disponível em: <https://www.zabbix.com/documentation/4.0/manual/introduction/whatsnew400>

4 TINKERCAD

De acordo com Guillermo Melantoni o criador do AutoCad, o Tinkercad é software de simulação online, tanto 3D ou 2D, ele consegue simular circuitos elétricos, analógicos e digitais, é um software gratuito, com uma grande biblioteca de componentes, resolvermos utilizar ele por recomendação de um aluno, a ideia inicial era de fazer a simulação junto do Zabbix, mas não será possível fazer isso pelas limitações do Tinkercad, um integrante do grupo tem conhecimentos técnicos com microcontroladores, e outro tem conhecimento em programação, então isso facilitou para realizar o trabalho. A Figura 7 mostra como é a interface de simulação.

Figura 7 — Simulação Tinkercad



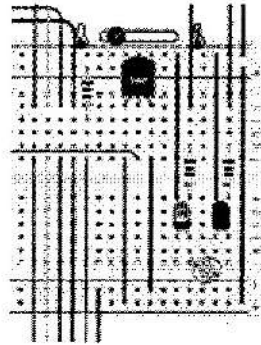
Fonte: Autoria Própria

Nesse pequeno circuito que fizemos se encontra o Arduino Uno, uma protoboard para encaixar os componentes, um visor de LCD 16 x 2, três resistores com a resistência de 220 Ω (ohm), um sensor de temperatura TMP36, dois leds, um da cor azul e outra da cor vermelha, e cabos que fazem a interligação de todos os componentes. A ideia dessa simulação é botar em prática como funciona um sistema de monitoramento, o primeiro protótipo tinha a intenção de colocar um ventilador no lugar do display LCD, mas o Tinkercad não possui, então resolvi colocar o display para mostrar a temperatura, o Arduino uno faz o controle de todos os componentes. A Figura 8 mostra o detalhe.

Conforme a temperatura é alterada no sensor TMP36, o valor é mostrado no display. Caso a temperatura for muito alta, o LED vermelho se acende, simulando o ventilador ligando, e caso a temperatura abaixe, o LED azul se acende, simulando que o ventilador ficará desligado, as figuras 9 e 10 abaixo vão ilustrar isso.

21

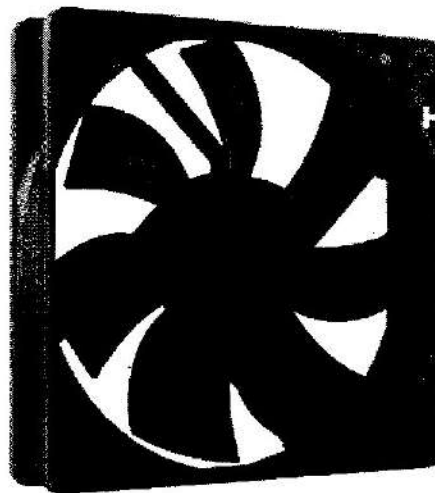
Figura 10 — Temperatura baixa, LED azul se acende



Fonte: Autoria Própria

Ao contrário do que está sendo apresentado, no trabalho físico, seria utilizado um cooler fan para que fossem feitos os testes. O Arduino Shield, que iria fazer a conexão com a internet, e enviaria os dados para a ferramenta Zabbix, e fazer o aviso da mudança de temperatura, assim acionando o cooler fan. Seria utilizado um cooler fan de tamanho padrão de computadores, conforme mostra a imagem abaixo:

Figura 11 – Cooler Fan



Fonte: www.dex168.com.br/produtos/informatica/-/48

5 O QUE É A REDE DE COMPUTADORES ?

Muito foi falado a respeito das ferramentas que serão utilizadas e como deverão ser utilizadas no projeto, e assim utilizar a rede de computadores para integrar tudo. Mas antes disso precisamos saber, o que é a rede de computadores? Para entendermos melhor, será explicado um pouco sobre o que é e como funciona.

Como resultado do rápido progresso tecnológico dos computadores no século XXI, as capacidades de processamento de informações ficaram também mais rápidas. À medida que cresce a capacidade de distribuir informações, torna-se ainda maior a demanda por formas mais sofisticadas de processamento de informação. Durante as primeiras décadas de sua existência, os sistemas computacionais eram altamente centralizados, em geral instaladas em uma grande sala, muitas vezes com paredes de vidro, para que os visitantes pudessem ver aquela "maravilha eletrônica".

A fusão dos computadores e comunicações teve uma profunda influência na forma como os sistemas computacionais são organizados. O conceito de "centro de computação" como uma sala com um grande computador local no qual os usuários levavam o seu trabalho para processamento agora está obsoleto. O velho modelo de um único computador atendendo a todas as necessidades computacionais da organização, foi substituído por outro em que os trabalhos são realizados por um grande número de computadores separados, porém interconectados. Esses sistemas são chamados Redes de Computadores. (Tanenbaum, 2011).

A Rede de Computadores nada mais é do que um conjunto de computadores e outros dispositivos conectados entre si, através de um sistema de comunicação, compartilhando informações e dados. As redes de computadores mais conhecidas hoje são a internet (a mais famosa de todas), a intranet de empresas e a rede local doméstica. Graças ao desenvolvimento das redes de computadores, hoje se torna possível a comunicação via e-mail, videoconferência e mensagens instantâneas, dentre outras novas tecnologias por exemplo.

6 MODELO TCP/IP

Para o funcionamento do trabalho, também será necessária uma breve explicação sobre os protocolos que serão utilizados no projeto. Mas antes, é preciso explicar um pouco sobre o que são estes protocolos. Existe o modelo de referência OSI e o modelo TCP/IP. Com objetivo semelhante ao modelo OSI no que diz respeito a divisão da arquitetura em camadas, o TCP/IP consiste na junção dos protocolos TCP (Transmission Control Protocol – Protocolo de Controle de Transmissão) e o IP (Internet Protocol – Protocolo de Internet), dois dos mais utilizados.

A diferença principal entre as duas estruturas é o número de camadas encontradas em cada modelo. No modelo OSI existem 7 camadas, enquanto o TCP/IP somente 4 camadas, que são: Aplicação, Transporte, Rede. A lógica de posicionamento das camadas sugere que as que estão mais acima encontram-se mais próximas do usuário. Dentro deste conjunto de protocolos, cada uma das camadas é programada para responder por um grupo de tarefas específicas e serviços definidos para garantir a integridade e entrega dos dados trafegados do que vai ser executado na camada acima.

A camada de aplicação é o topo da arquitetura TCP/IP, nesta camada são realizadas a maior parte das requisições para execução das tarefas na rede. Ela faz a comunicação entre os programas e os protocolos de transporte e é responsável por tudo que está relacionado aos serviços de comunicação que visam a interação junto ao usuário. Exemplos de protocolos que estão dentro desta camada: TELNET (Terminal Virtual), FTP (File Transfer Protocol) e DNS (Domain Name System). No modelo TCP/IP, a camada localizada acima da camada Internet agora é chamada Camada de Transporte. A finalidade desta camada é permitir que as entidades pares dos hosts de origem e destino mantenham uma conversão, assim como acontece na camada de transporte OSI. Foram definidos dois protocolos nesta camada, primeiro o TCP (Transmission Control Protocol), que é o protocolo orientado a conexões confiáveis que permite a entrega sem erros de um fluxo de bytes originário de uma determinada máquina em qualquer computador da internet. (Tanenbaum, 2011). O segundo protocolo nesta camada é o protocolo UDP.

Ainda de acordo com Tanenbaum, o User Datagram Protocol (UDP) é um protocolo que trabalha na camada de transporte, sendo responsável pela transferência de dados entre dispositivos, e sua principal característica é a falta de confiabilidade na entrega, ou seja, não existe garantia, e nem confirmação na entrega, mas isso não é necessariamente algo ruim, pois ele tem uma vantagem, que é a velocidade, permitindo uma conexão bastante rápida. Outra vantagem sua é que a comunicação do protocolo não é voltado a conexão, então dessa maneira o UDP consegue enviar informações para vários dispositivos diferentes, apenas com uma saída. O protocolo UDP faz uso do protocolo IP para transportar pacotes com informações UDP entre a origem e destino. Apesar de não recomendado, o Zabbix consegue trabalhar com o protocolo UDP, o monitoramento feito de um serviço que utiliza o protocolo UDP é bem simples, pois o servidor irá apenas verificar se a porta do serviço está ou não aberta, caso não haja conexão de internet no momento de monitoramento do sistema desse projeto, ele iria apenas chegar se o ventilador foi ou não acionado de acordo com a temperatura, porém ele não poderia realizar nenhuma ação pela falta de conexão.

A camada de Internet, também conhecida como inter-redes, é responsável pela permissão de envio de pacotes por hosts a qualquer rede e pela garantia de que esses dados cheguem ao seu destino final. Equivalente ao que é feito na camada de Rede do modelo OSI, na arquitetura TCP/IP a camada de internet tem como embasamento os protocolos IP (Internet Protocol) e ICMP (Internet Control Message Protocol). A camada de Interface correspondente às camadas de enlace de dados e física do OSI, onde ocorre a conexão básica do host com a rede por meio de algum protocolo capaz de enviar pacotes IP. É por meio desta camada que é possível transmitir dados a outros computadores dentro de uma mesma rede física, além de realizar o envio do datagrama recebido pela camada de internet através de meios físicos. Para se manter em funcionamento a camada de Interface de rede utiliza como principais protocolos: Ethernet para Redes Locais (LAN – Local Area Network) e PPP (Point-to-Point Protocol) para Redes de Longa Distância (WAN – Wide Area Network). (Escola Superior de Redes, 2020).

6.1 PROTOCOLO SNMP

De acordo com o livro "SNMP Essential" O Simple Network Management Protocol (SNMP) é um protocolo utilizado na parte de gerenciamento de rede, de maneira ampla qualquer equipamento de rede tem a capacidade de se comunicar via SNMP pois ele é um protocolo padrão no mundo inteiro, então não é possível instalar um SNMP, apenas é possível habilitar ou desabilitar ele. A ferramenta Zabbix e muitos outros aplicativos de monitoramento trabalham com o SNMP, em redes de médio ou grande porte muitos itens são monitorados, e o SNMP consegue lidar muito bem com isso, pois são dados de fácil acesso, como Status de dispositivos de rede, temperatura dos dispositivos, status de dispositivos de rede, processamento de dispositivos, utilização de memória, status de links WAN de roteadores, utilização de links WAN ou interfaces trunk e assim como esses itens, vários outros itens podem ser monitorados, nesse projeto o principal item a ser monitorado é a temperatura do ambiente, e caso ela se eleve o fan se ativa proativamente para diminuir a temperatura. O protocolo SNMP é utilizado para gerenciar redes TCP/IP, e com ele a equipe de suporte podem fazer a gestão e configuração de um servidor na rede, por padrão o SNMP trabalha com o protocolo UDP na porta 161, e uma rede gerenciada pelo SNMP utiliza 3 fatores importantes, o agente SNMP, que podem ser os roteadores ou Switchs da rede, MIB (Management Information Base), que é uma base de dados com um padrão definido que é compreendida pelo gerente SNMP e o gerente SNMP que é o sistema de gestão de rede, como por exemplo o Zabbix.

Em uma rede existe um dispositivo gerenciado que possui um agente SNMP instalado, esse dispositivo pega e guarda informações de gerenciamento e disponibiliza para um NMS por meio do protocolo SNMP, existem vários dispositivos de rede na maior parte das vezes são servidores de acesso, roteadores, dispositivos de armazenamento e entre outros. O sistema NMS é um gerente SNMP responsável pelos aplicativos que são monitorados e controlam os agentes SNMP. Um agente é um critério de software de gestão que fica guardado num dispositivo gerenciado.

O agente tem os dados das informações de gestão e converte estas informações para um formato compatível com o protocolo SNMP e padronizados em bancos de dados chamados MIB (Management Information Base).

No SNMP existem três versões 1, 2c e 3, a versão 1 hoje em dia já não é mais utilizada, mas a 2c e 3 sim, a diferença entre a versão 2c e 3 se dá na parte que a 2c não possui muitos recursos de segurança, e na versão 3 ela possui recursos como autenticação, garantia da integridade das mensagens e criptografia.

6.2 C++

Para fazer o Arduino funcionar, é preciso que exista uma programação, e a linguagem de programação que o Arduino utiliza é a C++. De acordo com o livro "C++ Para Leigos" essa linguagem foi criada em 1980 por Bjarne Stroustrup, e desde então ela vem evoluindo, sendo utilizada na programação orientada a objeto, hoje seu uso é muito vasto e amplo, sendo utilizado em vários programas, como DevC++ e Visual Studio. Por conta de a simulação ter um display LCD, foi preciso utilizar uma biblioteca que já possui todos os recursos para ela funcionar, os pinos 12, 11, 5, 4, 3 e 2 estão ligados no sistema para o LCD funcionar e mostrar a temperatura, o pino A0 será utilizado como a entrada do sensor de temperatura, o pino 8 será responsável para o alerta de temperatura baixa e o 13 para temperatura alta, foi definido como temperatura baixa abaixo de zero graus e temperatura alta acima de 40 graus.

7 METODOLOGIA

O projeto foi ainda na Fatec, pesquisando e observando os componentes que podem se aproveitar, e alguns recursos. A ideia era montar um protótipo até a Fatec Week e poder apresentar para o público, mas um imprevisto aconteceu no meio do mês de Março que acabou afetando os planos; a quarentena por causa do corona vírus. Então muitas coisas teriam que mudar. Foi feito um acordo entre os professores para poder simular o protótipo, visto que nesse distanciamento social, seria muito difícil montar o protótipo físico, mas a entrega do trabalho escrito continuava normal. Conforme a sugestão de um aluno, optamos por utilizar um aplicativo chamado Tinkercad, que poderá fazer a simulação parcial do trabalho. Nesse trabalho foi designado para o ambiente educacional, pois decidimos utilizar esse projeto dentro de salas de aula para fazer o controle da temperatura do ambiente.

Foi feita uma simulação parcial do projeto devido aos recursos limitados do aplicativo. Nessa simulação foi feita toda a parte de automação do trabalho, apenas a ferramenta de monitoramento Zabbix não pôde ser incluída na simulação pois ela não tem esse tipo de integração. No programa Tinkercad foi utilizado um Arduino Shield, uma placa protoboard, um sensor de temperatura TMP36, 3 resistores de 220 Ω , um LED azul, um LED vermelho junto com um fotoresistor e uma tela lcd de 16x2. A ideia da simulação é mostrar como o sensor de temperatura irá funcionar, os LEDs simbolizam o ventilador da sala de aula, e a tela lcd é apenas ilustrativo para mostrar a temperatura que está mudando de acordo com a temperatura. No projeto físico a ideia é praticamente a mesma, porém não haverá uma tela lcd, e no lugar dos LEDs haverá uma ventoinha. No projeto físico também haverá um Arduino Shield que fará a conexão com a internet para se comunicar com o Zabbix para poder enviar informações para a ferramenta sobre o monitoramento, tudo isso será feito e programado no aplicativo Arduino Uno Zabbix Agent, que é um software exclusivo para isso. No sistema real a montagem será feita da seguinte maneira, será necessário um protoboard, um Arduino R3, um Arduino Shield W5100, um LCD 16x2, um sensor de temperatura TMP36, um led vermelho e outro azul e três resistores de 220 Ohms.

Para os testes para fazer a instalação desse sistema, o Arduino Shield W5100 será acoplado ao Arduino Uno nas suas respectivas pinagens para que não haja problema, um cabo será inserido no pino 5V do Arduino Shield W5100 e fará ligação no pino positivo do protoboard, no pino GND será realizado o mesmo porém fazendo ligação no pino negativo do protoboard, esses cabos irão fornecer energia ao sistema inteiro através do protoboard, no pino A0 analógico do protoboard será levado um fio até o pino Vout do sensor de temperatura. No lado direito do Arduino onde se encontra os pinos digitais, no pino 13 será colocado um cabo que levará até o led vermelho que simboliza a temperatura quente junto de um resistor de 220 Ohms, no pino 8 será realizado o mesmo, porém para um led azul que simboliza a temperatura baixa, sobre o sensor de temperatura, o pino GND dele será ligado no negativo do protoboard, e o pino de potência será ligado no positivo do protoboard. Para a ligação do display LCD o pino 12 do Arduino será inserido na entrada RS do LCD, o pino 11 do Arduino será ligado na entrada E do LCD, o pino 5 do Arduino será ligado na entrada db4 do LCD, o pino 4 com o db5, o pino 3 com o db6, e o pino 2 com o db7, todos esses cabos foram ligados para que as informações do Arduino sejam mostradas no LCD. Os pinos GND, VD, RW e LED serão conectados no negativo do protoboard para a sua alimentação e o pino VCC do LCD será conectado no positivo da placa protoboard. Todas essas conexões foram feitas para que o Arduino funcione conforme a programação colocada nele.

Na parte de software será instalado o aplicativo Arduino BasicAct para fazer a programação do sistema a programação dele se encontra logo abaixo nesse trabalho, o Agent Zabbix para fazer a monitoração do circuito e o servidor Zabbix em um computador para poder verificar o monitoramento.

Com tudo instalado e funcionando, para a realização dos testes um cabo com conexão à internet será colocado no Arduino Shield W5100 na sua entrada RJ45, um computador ficará ligado para verificar o monitoramento do Zabbix, no sensor de temperatura será utilizado uma fonte de calor como um soprador térmico, que irá atuar em cima do sensor de temperatura.

Quando a temperatura atingir 40°C, o led vermelho irá se acender, simbolizando um ventilador ligado, e um aviso será enviado ao Zabbix notificando que a temperatura aumentou e o fan foi acionado, quando a temperatura atingi 0°C será desligado o fan e outra mensagem será ligada ao Zabbix notificando que o ventilador foi desligado. Esse sistema está conectado à internet e depende dele para que os avisos no servidor Zabbix sejam notificados.

No aplicativo Tinkercad, foi realizado uma simulação com todos os testes do Arduino e seus componentes, porém, a parte de simulação com rede não foi possível ser feita por limitações do próprio programa, o sistema funcionou de acordo com o previsto em seus testes no sensor de temperatura. Os resultados obtidos foram que o sistema funciona conforme ele foi designado para a sua função, ele monitora a temperatura em tempo real, faz o acionamento automático do fan e realiza o aviso para o servidor Zabbix, através de algumas pesquisas para efeito de comparação foram encontrados alguns trabalhos que seguem a mesma linha de ideia, mas feito com outros sensores como o DHT11, e em outro trabalho foi utilizado o sensor de temperatura DHT22, este trabalho se diferencia dos outros por fazer um monitoramento e controle do sistema através da rede e por utilizar outro sensor de temperatura.

7 CONSIDERAÇÕES FINAIS

Espera-se que os resultados coletados nesse trabalho, possam melhorar a qualidade de vida da sala de aula dos alunos da Fatec, por conta de fatores externos e a pandemia, não foi possível realizar a montagem física do projeto, mas foi realizado uma simulação muito semelhante com os mesmos componentes que seriam utilizados. Com os resultados obtidos com esse trabalho, foi comprovado que com poucos dispositivos e um ambiente de monitoramento via rede é possível automatizar e melhorar o funcionamento de um ambiente escolar, ainda existem melhorias que possam ser feitas, como a instalação de um módulo WiFi e a implementação de outros itens para controle e monitoração, a parte mais complexa do trabalho foi o estudo se era possível fazer a integração do projeto com o Zabbix, mas vendo outros projetos e a experiência anterior dos participantes se chegou na conclusão que era possível fazer a integração e para provar isso, aqui vai 3 citações de trabalhos semelhantes a esse.

Namonografia de Marcos Aurélio, foi realizado um trabalho sobre aferição e monitoramento remoto de temperatura, ele utiliza um sensor diferente, o DTH22, mas que faz a mesma função do TMP 36 que é monitorar temperatura, e ele pode ser integrado também ao Zabbix e Arduino, o projeto de Marcos Aurélio utilizou o Zabbix, um Arduino e um circuito integrado ENC28J60 para fazer a conexão com a internet, o trabalho dele funcionou e conseguiu realizar o monitoramento de temperatura. (AURÉLIO, 2018).

No TCC de Amauri Marcos, ele fez um projeto sobre sistema de monitoramento de temperatura e umidade relativa do ar, Amauri também utiliza um Arduino para integração dos componentes e também utiliza o Arduino IDE para fazer a programação do projeto, nesse trabalho foi usado um sensor DTH11, que além de medir a temperatura, faz a aferição de umidade no ar, apesar do seu trabalho não ser conectado via rede, ele é muito semelhante ao nosso na parte de automação e componentes. (MARCOS, 2017)

No último trabalho citado Lailton Montenegro, realizou um TCC sobre monitoramento de temperatura e umidade de data center utilizando o Arduino e o sistema Zabbix, no trabalho de Lailton, ele usa dois componentes que seriam utilizados nesse trabalho o Arduino Uno e o Ethernet Shield para fazer a conexão com a internet, o sensor de temperatura é diferente, foi utilizado o DTH 11, igual no trabalho de Amauri, foi feita também a utilização do Zabbix para o monitoramento com rede. (MONTENEGRO, 2017)

REFERÊNCIAS

ASHTON, Kevin. **Citação direta:** Essa coisa da Internet das Coisas. **RFIDJournal**. 2021. Disponível em: <https://www.rfidjournal.com/articles/view?4986>. Acesso em: 18 mai. 2020.

Arquitetura TCP/IP: Conceitos Básicos. Escola Superior de Redes, 2020. Disponível em: <https://esr.rnp.br/administracao-e-projeto-de-redes/arquitetura-tcp/ip-conceitos-basicos/>. Acesso em 11/11/2020.

AURÉLIO, Marcos. Aferição e monitoramento remoto de temperatura e umidade em datacenter. Orientador: Dr. José de Ribamar Braga Pinheiro Junior. 2018. 50 f. Monografia (Graduação) - Curso de Engenharia Elétrica, Universidade Federal do Maranhão, Centro de ciências exatas e tecnologia, São Luís, 2018 Disponível em: <https://monografias.ufma.br/jspui/bitstream/123456789/2434/1/MARCOSAURELIOSI LVA.pdf>. Acesso em 1 de setembro de 2021.

IOT: A TECNOLOGIA E O MONITORAMENTO CONTÍNUO. Sensorweb, 2015. Disponível em: <https://sensorweb.com.br/tecnologia-monitoramento-continuo-iot-1/#:~:text=Todos%20os%20objetos%20eletroeletr%C3%B4nicos%20produzidos,feito%20a%20n%C3%ADvel%20de%20sensores>. Acesso em: 20 de maio de 2020

MADEIRA, Daniel. Arduino – Entradas e saídas digitais – Pinos digitais. Disponível em <https://portal.vidadesilicio.com.br/entradas-e-saidas-digitais/>. Acesso em: 14 de maio de 2020

MARCOS, Amauri. Sistema de monitoramento de temperatura e umidade relativa do ar. Orientador: Hélio Roberto Hékis. 2017. 39 f. TCC (Graduação) - Departamento de engenharia biomédica, Universidade Federal Rio Grande do Norte, Centro de tecnologia, Natal, 2017. Disponível em: <https://eb.ct.ufrn.br/wp-content/uploads/2019/03/Amauri-Junior.pdf>. Acesso em 1 de setembro de 2021.

MICROBERTS, Michael. Arduino Básico. Rua Luís Antônio dos Santos 110. Novatec, 2011.

MONTENEGRO, Lailton. Monitoramento de temperatura e umidade de data center utilizando o arduino e o sistema zabbix, Orientador: Prof. Aluizio Ferreiro de Rocha Neto. 2017. 38 f. TCC (Graduação) – Especialização em redes de computadores, Centro Universitário do Rio grande do norte, Liga de ensino do Rio Grande do Norte, Natal, 2017. Disponível em:

<https://pt.slideshare.net/lailtonmontenegro/monitoramento-de-temperatura-e-umidade-de-data-center-utilizando-o-arduino-e-o-sistema-zabbix>. Acesso em 1 de setembro de 2021

REIS LIMA, Janssen. Monitoramento de Redes com Zabbix. Edifício Sir Winston Churchill. BRASPORT, 2014.

SOUZA, Fábio. Arduino UNO. Disponível em <https://www.embarcados.com.br/arduino-uno/>. Acesso em: 23 de junho de 2020

TANENBAUM, Andrew. Redes de Computadores: 5ª Edição. Pearson Universidades, 2011.

VENKATESH, Alladi. Computadores e outras tecnologias interativas para o lar. Disponível em <https://go.galegroup.com/ps/anonymous?id=GALE%7CA19017352&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=00010782&p=AONE&sw=w> .Acesso em 18 de maio de 2020 e 19 de maio 2020.

APÊNDICE

APÊNDICE A – CÓDIGO DE PROGRAMAÇÃO DO ARDUINO

```
1 // Biblioteca LCD
2 #include <LiquidCrystal.h>
3
4 // Inicializa a biblioteca LCD
5 LiquidCrystal LCD(12,11,5,4,3,2);
6
7 // Define o pino analógico A0 como entrada do Sensor de Temperatura
8 int SensorTempPino=0;
9
10 // Define o pino 8 para o alerta de temperatura baixa
11 int AlertaTempBaixa=8;
12 // Define o pino 18 para o alerta de temperatura alta
13 int AlertaTempAlta=18;
14
15 // Define temperatura baixa como abaixo de zero grau Celsius
16 int TempBaixa=0;
17 // Define temperatura alta como acima de 40 graus Celsius
18 int TempAlta=40;
19
20 void setup() {
21     // Define o pino de alerta de temperatura baixa como saída
22     pinMode(AlertaTempBaixa, OUTPUT);
23     // Define o pino de alerta de temperatura alta como saída
24     pinMode(AlertaTempAlta, OUTPUT);
25
26     // Define a quantidade de colunas e linhas do LCD
27     LCD.begin(16,2);
28     // Imprime a mensagem no LCD
29     LCD.print("Temperatura:");
30     // Move o cursor para a primeira coluna e segunda linha do LCD
31     LCD.setCursor(0,1);
32     // Imprime a mensagem no LCD
33     LCD.print("      C      F");
34 }
35
36 void loop() {
37     // Faz a leitura da tensão no Sensor de Temperatura
38     int SensorTempTensao=analogRead(SensorTempPino);
39
40     // Converte a tensão lida
41     float Tensao=SensorTempTensao*5;
42     Tensao/=1024;
43
44     // Converte a tensão lida em Graus Celsius
```

```

45 float TemperaturaC=(Tensao-0.5)*100;
46 // Converte a temperatura em graus Celsius para Fahrenheit
47 float TemperaturaF=(TemperaturaC*9/5)+32;
48 // Muda o cursor para a primeira coluna e segunda linha do LCD
49 LCD.setCursor(0,1);
50 // Imprime a temperatura em graus Celsius
51 LCD.print(TemperaturaC);
52 // Muda o cursor para a decima coluna e segunda linha do LCD
53 LCD.setCursor(9,1);
54 // Imprime a temperatura em graus Fahrenheit
55 LCD.print(TemperaturaF);
56 // Acende ou apaga os alertas luminosos de temperatura baixa e alta
57 if (TemperaturaC>TemperaturaAlta) {
58   digitalWrite(AlerataTempBaixa, LOW);
59   digitalWrite(AlerataTempAlta, HIGH);
60 }
61 else if (TemperaturaC<TemperaturaBaixa) {
62   digitalWrite(AlerataTempBaixa, HIGH);
63   digitalWrite(AlerataTempAlta, LOW);
64 }
65 else {
66   digitalWrite(AlerataTempBaixa, LOW);
67   digitalWrite(AlerataTempAlta, LOW);
68 }
69 delay(1000);
70 // aguarda 1 segundo
71 }

```