

UNIVERSIDADE SANTA CECÍLIA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA MECÂNICA
MESTRADO EM ENGENHARIA MECÂNICA

ALEXANDRE FERNANDO STUCCHI

**SISTEMA DE MONITORAMENTO DE EFICIÊNCIA ENERGÉTICA DO
DATA CENTER DO SIMULADOR PORTUÁRIO APLICANDO O
CONCEITO DE *INTERNET DAS COISAS***

SANTOS/SP
2017

ALEXANDRE FERNANDO STUCCHI

**SISTEMA DE MONITORAMENTO DE EFICIÊNCIA ENERGÉTICA DO
DATA CENTER DO SIMULADOR PORTUÁRIO APLICANDO O
CONCEITO DE *INTERNET DAS COISAS***

Dissertação apresentada à Universidade Santa Cecília como parte dos requisitos para obtenção de título de mestre no Programa de Pós-Graduação em Engenharia Mecânica, sob orientação do Prof. Dr. Mauricio Conceição Mario e coorientação do Me. Lucas Ramos De Pretto.

**SANTOS/SP
2017**

Autorizo a reprodução parcial ou total deste trabalho, por qualquer que seja o processo, exclusivamente para fins acadêmicos e científicos.

621.56 Stucchi, Alexandre Fernando
S921s Sistema de Monitoramento de Eficiência Energética do Data Center do Simulador Portuário aplicando o conceito de *Internet da Coisas* / Alexandre Fernando Stucchi. - 2017.
57 P.

Orientador: Prof. Dr. Mauricio Conceição Mario.
Coorientador: Me. Lucas Ramos De Pretto

Dissertação (Mestrado) -- Universidade Santa Cecília, Programa de Pós-Graduação em Engenharia Mecânica, Santos, SP, 2017.

1. Sensores de temperatura, 2. Prototipagem hardware aberto, 3. Temperatura data center I. Mario, Maurício Conceição, orient. II. De Pretto, Lucas Ramos, coorient. III. Sistema de Monitoramento de Temperatura do Data center do Simulador Portuário aplicando o conceito de *Internet da Coisas*.

DEDICATÓRIA

*Ao meu pai Ivo, que sempre me
conduziu pelos caminhos da
honestidade.*

*A minha mãe Neusa, que sempre
demonstrou sua força e coragem
enquanto esteve conosco.*

*A minha esposa Inêz, luz da minha
existência que sempre me incentivou
nos meus desafios.*

AGRADECIMENTOS

Ao Professor Doutor Mauricio Conceição, por sua paciência, dedicação, entusiasmo e provar que a eletrônica é linda, durante a execução do trabalho de orientação.

Ao Me. Lucas Ramos De Pretto meu parceiro de vários projetos que disponibilizou seu tempo para mais essa empreitada.

A minha esposa Maria Inêz Campos Castanheira, que sempre me apoiou principalmente nas adversidades.

As secretárias da UNISANTA, Sandra e Imaculada, pela atenção e carinho dispensados a mim e a todos alunos.

Ao Serviço Nacional de Aprendizagem Industrial – SENAI, pela oportunidade desta nova etapa na minha carreira docente e poder utilizar toda sua infraestrutura.

A Prof. Marcelo Saraiva Coelho, pelo incentivo e apoio no início desta jornada.

Ao Prof. Daniel Divino, por estar sempre ao meu lado, incentivando-me em todas as fases do meu trabalho.

Ao corpo docente do programa de Pós-Graduação em Engenharia Mecânica, com especial destaque para os professores Dr. João Inácio de Silva Filho e Dr. Marcos Tadeu Tavares Pacheco, pelas orientações no ramo de pesquisa acadêmica.

Aos colegas da turma de mestrando da UNISANTA, em especial ao Fernando Bacic, pela força e o apoio em mais essa jornada.

Ao meu colega de trabalho Prof. Roberto Blanco pelo apoio neste trabalho.

Ao meu colega de trabalho Prof. Carlos Alberto José de Almeida “CAJA” por sempre me encorajar.

A Prof.^a Maria Elisabete Campos Vieira de Castro por realizar a revisão deste trabalho.

RESUMO

Este trabalho aborda a dinâmica de gasto energético com relação ao sistema de refrigeração de *data centers* aliados a plataforma de prototipagem de *hardware* aberto microcontrolado, para um estudo de caso de um *data center* de pequeno porte, localizado em uma escola técnica onde um simulador portuário está instalado. Objetivou-se verificar a dinâmica de gasto energético com relação ao sistema de refrigeração e estudou-se as temperaturas possíveis a partir da norma EIA/TIA 942 que normatiza a construção de *data centers* até sua ativação. Foi verificado através de sensores qual a relação gasta com o sistema de refrigeração e comportamento das temperaturas neste ambiente ao longo de um período, disponibilizando os dados em rede de acordo com o conceito da *internet* das coisas. Verificou-se também qual a melhor temperatura para um melhor aproveitamento energético, segundo critérios da norma aplicada aos *data centers* na classificação básica, seguindo a tendência das grandes empresas de tecnologia da informação que estão aumentando o ponto de equilíbrio da temperatura em seus *data centers* para uma melhor eficiência energética, com esse estudo espera-se uma economia de até 22% do consumo gasto pelo sistema de refrigeração, promovendo assim a sustentabilidade não apenas na unidade escolar em que o estudo realizou-se mas com possibilidades de ser replicado à rede de ensino desta instituição.

Palavras-chave: Sensores de temperatura. Prototipagem *hardware* aberto. Temperatura *data center*. *Internet* das coisas.

ABSTRACT

This work addresses the dynamics of energy expenditure in relation to the cooling system of data centers allied to the microcontrolled open hardware prototyping platform for a case study of a small data center, located in a technical school where a port simulator is installed. The objective was to verify the energy expenditure dynamics in relation to the refrigeration system and studied the possible temperatures from the EIA / TIA 942 norm that normalizes the construction of data centers until its activation. It was verified through sensors what the relation spent with the cooling system and the behavior of the temperatures in this environment over a period, making available the data in network according to the concept of the internet of things. It was also verified the best temperature for a better energy use, according to the criteria applied to the data centers in the basic classification, following the tendency of the large information technology companies that are increasing the temperature balance in their data centers, to this study is expected to save up to 22% of the consumption spent by the refrigeration system, thus promoting sustainability not only in the school unit where the study was carried out but with the possibility of being replicated to the grid institution.

Keywords: Temperature sensors. Open prototyping hardware. *Data center* temperature. Internet of things.

LISTA DE FIGURAS

Figura 1 - Cabine simulador portuário.....	12
Figura 2 - Diagrama geral dos componentes envolvidos no estudo	13
Figura 3 - Servidores em cluster no <i>data center</i>	14
Figura 4 - Sensor de temperatura básico	18
Figura 5 - Encapsulamento - TO-92.....	18
Figura 6 - O sensor LM35 com encapsulamento TO-92	19
Figura 7 - Termoresistência convencional com bainha preenchida.....	19
Figura 8 - Medindo a corrente em um condutor	20
Figura 9 - O sensor de corrente SCT-013-000	21
Figura 10 - Saída de tensão.....	22
Figura 11 - Saída de corrente.....	22
Figura 12 - Arquitetura de sistemas microcontrolados.....	23
Figura 13 - Diagrama de blocos dos dispositivos para medições da temperatura	27
Figura 14 - Esquema eletrônico para a medição de temperatura com LM35 no <i>data center</i>	28
Figura 15 - Sensor LM35 (circulado em vermelho) acoplado ao Arduino Mega 2560	28
Figura 16 - Envio de dados LM35 para porta serial Arduino	29
Figura 17 - O registrador MV1000 e o sensor Pt-100.....	29
Figura 18 - Temperatura externa medida pelo sensor Pt-100	30
Figura 19 – Pt-100 medindo a temperatura externa.....	30
Figura 20 - Medição de corrente com SCT-013-000 no <i>data center</i>	32
Figura 21 - Esquema eletrônico para medição de corrente no <i>data center</i>	32
Figura 22 – Diagrama de blocos das medições da corrente	33
Figura 23 - Código fonte da coleta dos dados da temperatura.....	34
Figura 24 - Inserção das bibliotecas sensor de corrente	36
Figura 25 - Definição dos valores para tensão, corrente e pinagem	36
Figura 26 - Cálculo corrente, exibição dos dados e condição para registro	36
Figura 27 - Ethernet shield	37
Figura 28 - Estrutura de comunicação do shield Ethernet.....	38
Figura 29 – Bibliotecas para comunicação em rede.....	39
Figura 30 - Declaração dos endereços e portas de comunicação	39
Figura 31 - Inicialização do servidor e das conexões - portas	39
Figura 32 - Código fonte cliente Ethernet.....	40
Figura 33 - Continuidade na transmissão de dados.....	41
Figura 34 - Código HTML para acessar a temperatura via rede	41

Figura 35 - Tempo de resfriamento do <i>data center</i>	43
Figura 36 - Configuração temperatura de segurança para desligamento dos servidores.....	43
Figura 37 - Tempo e temperatura do travamento dos servidores do <i>data center</i>	44
Figura 38 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 20 °C.....	45
Figura 39 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 21 °C.....	46
Figura 40 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 22 °C.....	47
Figura 41 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 23 °C.....	48
Figura 42 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 24 °C.....	49
Figura 43 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 25 °C.....	50

LISTA DE TABELAS

Tabela 1 - Classificação dos TIERS norma EIA/TIA - 942	16
Tabela 2 - Características do sensor de temperatura LM35	18
Tabela 3 – Parâmetros do sensor de corrente	21
Tabela 4 - Comparativo entre microcontroladores Atmega	25
Tabela 5 - Síntese dos dados analisados temperatura x funcionamento do compressor.....	50
Tabela 6 - Consumo do sistema de refrigeração de acordo com a temperatura configurada	52
Tabela 7 - Índice de eficiência em função da temperatura.....	52
Tabela 8 - Relação de consumo em Reais (R\$) para cada temperatura em relação as bandeiras da fornecedora de energia.....	53

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 Justificativa.....	14
1.2 Objetivo Geral.....	15
1.3 Objetivos Específicos.....	15
1.4 Fundamentação Teórica.....	16
1.4.1 Sensores.....	17
1.4.2 O sensor de temperatura LM35.....	17
1.4.3 O sensor de temperatura Pt-100.....	19
1.4.4 O sensor de corrente.....	20
1.4.5 Microcontroladores.....	22
1.4.6 Plataforma de prototipagem open hardware - Arduino Mega 2560.....	23
1.4.7 A internet das coisas (internet of things - IoT).....	25
2 MATERIAIS E MÉTODOS.....	27
2.1 Validação dos dados do sensor de temperatura.....	27
2.2 Aferição da temperatura.....	27
2.3 Análise de ambiente.....	31
2.4 Medição da corrente.....	31
2.5 Desenvolvimento de softwares e serviços lógicos.....	33
2.5.1 Análise do código sensor de temperatura.....	34
2.5.2 Análise do código sensor de corrente.....	35
2.5.3 Análise do código do servidor web - Arduino shield.....	37
3 RESULTADOS E DISCUSSÃO.....	42
3.1 Verificação do ambiente.....	42
3.2 Verificação do consumo elétrico.....	45
3.3 Eficiência energética.....	51
4 CONCLUSÃO.....	55
REFERÊNCIAS.....	56

1 INTRODUÇÃO

Data center é o espaço onde se concentram os equipamentos responsáveis pelo processamento e armazenamento de informações consideradas cruciais para o desenvolvimento nos mais variados tipos de negócios, em qualquer tipo de organização, tais como indústrias, órgãos governamentais, instituições de ensino entre outros (MARIN, 2011).

Os *data centers* da atualidade são os antigos CPDs (Centro de Processamento de Dados) no entanto a quantidade de equipamentos ativos instalados nos *data centers* modernos é maior que dos antigos CPDs (MARIN, 2011).

Em um *data center* pode-se encontrar os seguintes sistemas: sala de computador, ar condicionado e controle ambiental, distribuição elétrica, detecção e supressão de incêndio, entre outros (MARIN, 2011).

Em uma escola técnica profissionalizante encontra-se o primeiro laboratório de simulações de operações portuárias da América Latina, sendo investidos R\$ 2,7 milhões. O simulador conforme a figura 1 possui um *data center* onde se concentram os equipamentos destinados ao processamento de dados de maneira ininterrupta e confiável. Este *data center* é equipado com sete servidores que processam as imagens enviadas à cabine do operador de maneira síncrona, além de armazenar os dados de operação destes equipamentos. Para garantir a disponibilidade e ampliar a vida útil de equipamentos em *data centers* é necessário o monitoramento de temperatura.



Figura 1 - Cabine simulador portuário.

O escopo de temperatura ideal ainda é discutido, mas independente de qual é a melhor faixa de temperatura, o monitoramento e manutenção desta são imprescindíveis para a operação do *data center* de nível básico. Considera-se de forma conservadora, que a temperatura global de um *data center* deve estar entre 20°C e 25°C segundo norma TIA/EIA-942 (ANSI/TIA, 2005).

O fator econômico também é muito importante, a *Microsoft* conseguiu redefinindo o ponto de ajuste de resfriamento no *data center* economizar US\$ 250.000,00 anuais, fazendo o processo de monitoramento da temperatura do seu *data center* no Vale do Silício, segundo Don Denning gerente da Lee Technologies que participou do projeto (MILLER, 2008).

A proposta deste estudo é encontrar uma solução de baixo custo para o monitoramento da eficiência energética, utilizando um sistema microcontrolado, visando a preservação dos equipamentos contidos nesse cenário, possibilitando uma capacitação sem interrupção e travamento do simulador, enviando os dados coletados para a rede seguindo o conceito da *internet* das coisas e que possa ser aplicado em diversos setores (empresas e indústrias de pequeno porte, meio acadêmico, entre outros.), para que estes atinjam um nível de operação de excelência. A figura 2 ilustra o diagrama geral dos componentes envolvidos neste estudo.

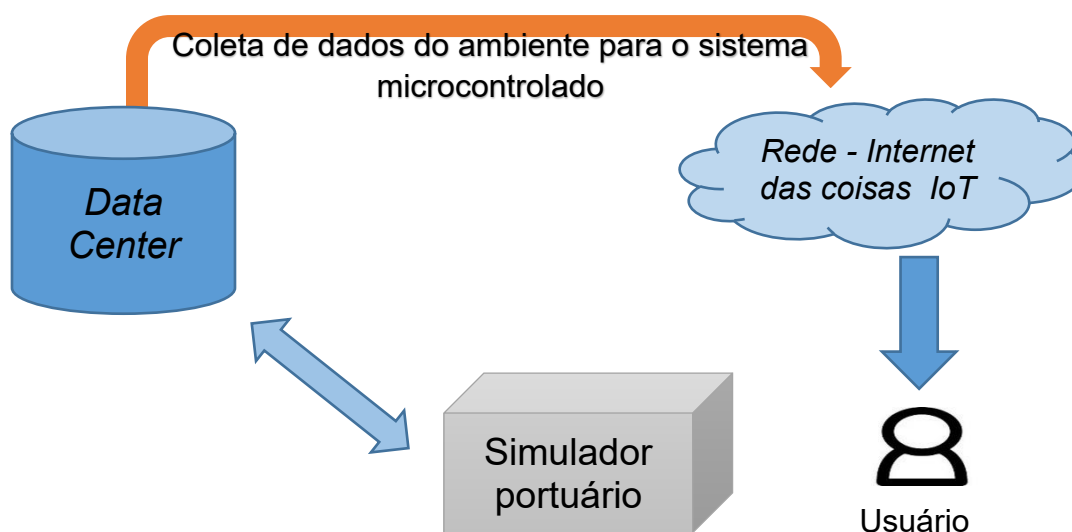


Figura 2 - Diagrama geral dos componentes envolvidos no estudo.

1.1 Justificativa

Monitorar a temperatura do *data center* para que não comprometa a operação do simulador portuário e consequentemente a continuidade operacional da aula, preservando os equipamentos (servidores) de possíveis falhas e travamentos, propondo uma solução de baixo custo. Pode-se encontrar sistemas de automação dedicados ao processo de monitoramento com custo em torno de R\$ 3.500,00, no entanto, nem todos os estabelecimentos possuem recursos disponíveis para pagar esse valor. Além disso, os sistemas comerciais são pouco flexíveis, já que os sistemas embarcados são proprietários.

O *data center* em estudo possui sete servidores ligados em *cluster*¹, conforme figura 3, processando os dados simultaneamente para os *displays* da cabine de operação.

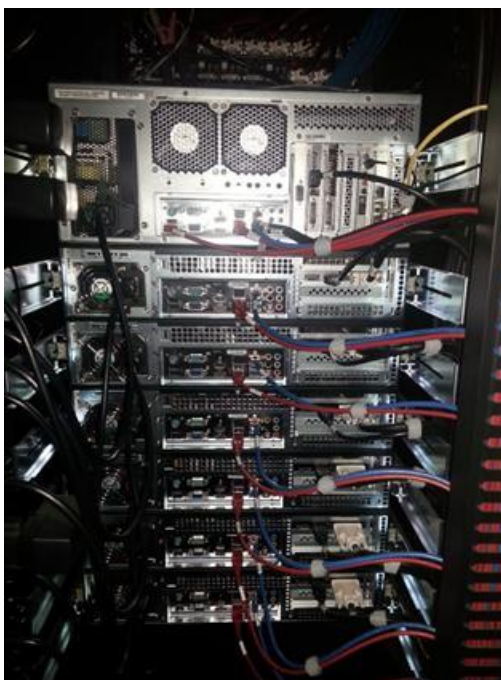


Figura 3 - Servidores em cluster no *data center*.

O sistema proposto a ser estudado pode ter um custo inferior a R\$ 350,00, ou seja, 90% a menos comparado aos equipamentos comercializados, por ser uma solução eletrônica de código aberto baseado em *hardware* e *software easy-to-use* (fácil de

¹ Forma como computadores estão interligados processando e executando uma ação simultaneamente

usar). A plataforma possibilita alta escalabilidade, bastando a aquisição e incorporação de novos *hardwares* de baixo custo.

1.2 Objetivo Geral

Encontrar um ponto de ajuste de resfriamento para reduzir o consumo de energia elétrica do sistema de refrigeração, através da elaboração de um sistema de monitoramento de temperatura.

Protótipos baseados em sistemas eletrônicos microcontrolados irão monitorar a temperatura do *data center* para evitar possíveis falhas ou danos nos equipamentos do *data center* e também no simulador portuário.

1.3 Objetivos Específicos

- Pesquisar sensores de temperatura para aferições de temperatura em ambientes fechados.
 - Pesquisar programação em microcontroladores.
 - Pesquisar programação de linguagem para a web para disponibilizar os dados em rede.
 - Demonstrar a configuração da interface de comunicação *Ethernet* para o envio dos dados pela web (IoT - *internet of things* ou *internet* das coisas).
 - Demonstrar a configuração de servidor web em ambiente microcontrolado.
 - Calcular curvas de calibração do controle de temperatura de um *data center* através de monitoramento com sistemas eletrônicos microcontrolados.
 - Pesquisar o comportamento Temperatura x Gasto Energético para promover maior eficiência em um estudo de caso.

1.4 Fundamentação Teórica

Segundo a EIA/TIA 942 as oscilações de temperatura podem afetar o desempenho e a vida útil dos equipamentos ou causar o *downtime*² situação em que o computador pode desligar sozinho como forma de se proteger de um eventual dano mais grave (ANSI/TIA, 2005).

A norma internacional define as condições mínimas para que um projeto de *data center* funcione, nela estão definidos os critérios para a construção de *data centers*, como a especificação dos requisitos para a construção civil, energia elétrica, ar condicionado, cabeamento estruturado, sistemas de proteção contra fogo, entre outros. Com relação a temperatura o *data center* deve estar em um intervalo de temperatura entre 20°C a 25°C. A norma também classifica de forma progressiva a disponibilidade da infraestrutura em quatro TIERS (camadas), descritos conforme tabela 1(ANSI/TIA, 2005).

Tabela 1 - Classificação dos TIERS norma EIA/TIA – 942 (ANSI/TIA, 2005).

TIER 1: Básico	Susceptível a interrupções planejadas ou não planejadas. Erros operacionais interrupção dos serviços. Permite uma indisponibilidade (<i>downtime</i> , ou seja, tempo que uma aplicação ou sistema permanece indisponível para utilização) anual máximo de até 28,8 horas
TIER 2: Componentes Redundantes	Aplica-se redundância nos equipamentos, esta categoria é menos susceptível a falhas. O ambiente possui piso elevado. A distribuição de circuitos não é redundante. Permite uma indisponibilidade (<i>downtime</i>) anual de até 22 horas.
TIER 3: Manutenção Concorrente	Permite atividades planejadas de manutenção sem a necessidade de interrupção dos serviços. Permite uma indisponibilidade (<i>downtime</i>) anual máximo de até 1,6 horas.
TIER 4: Tolerante a Falhas	Além de possuir infraestrutura e capacidade para operar sob qualquer tipo de ação de manutenção programada, o ambiente é também tolerante a falhas, garantindo o fornecimento de recursos de energia elétrica, ar condicionado, mesmo durante a ocorrência de pior caso intempestivo (enchentes, falta de energia elétrica, falta de água, queda de pequenas aeronaves, etc.). Permite uma indisponibilidade (<i>downtime</i>) anual máximo de até 0,4 horas.

O TIER do *data center* do simulador portuário está classificado como TIER 1.

² Downtime é o tempo sem operação por causa não planejada

1.4.1 Sensores

Sensor é um dispositivo capaz de sentir a variação de uma grandeza física qualquer, entretanto a capacidade de correlacionar essa variação com alguma outra grandeza o caracteriza com um transdutor (SILVA,2015, p.25).

“Termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza que precisa ser medida, como temperatura, pressão, velocidade, corrente, aceleração, posição, etc.” (ALBUQUERQUE; THOMAZINI, 2011, p.16).

O termômetro de mercúrio é exemplo de um sensor transdutor, pois o mercúrio sofre uma expansão volumétrica direta (pressão) em função da temperatura. Mas para que a medição com este aparelho aconteça, é necessário ter câmara de expansão graduada, para que se perceba a variação volumétrica em função de uma escala. Mesmo assim a averiguação da escala é visual.

A larga utilização de sensores aliado aos sistemas de controle proporcionam, cada vez mais, maior produtividade, qualidade, confiabilidade da produção e dos processos, menor custo de produção e a possibilidade de maior eficiência energética.(ALVES, 2010)

1.4.2 O sensor de temperatura LM35

A série LM35 faz parte da linha componentes semicondutores (circuitos integrados) fabricada pela *National Semiconductor*. São termosensores de temperatura para a faixa de graus centígrados (Celsius).

Ele possui um incremento de 10 mV/°C o que significa que a cada aumento de 10mV temos o aumento de 1 °C. Para exemplificar, no caso de uma temperatura ambiente de 30 °C a tensão de saída obtida será de 300 mV.

Algumas aplicações para este sensor são sistemas de gerenciamento para baterias, eletrodomésticos, sistemas de refrigeração entre outros.

Sua característica principal é fornecer uma tensão de saída diretamente proporcional à temperatura com excelente linearidade na faixa de -55 °C a +150 °C.

O dispositivo não gera calor que possa influenciar no resultado das medições pois o seu consumo é menor que 60 μA . A tabela 2 mostra as características do sensor LM35.(INSTRUMENTS, 2016)

Tabela 2 - Características do sensor de temperatura LM35 (LM35 DATASHEET, 2016).

Tensões máximas de alimentação	-0,2 V a +35 V
Tensões de saída	-1,0 V a +6,0 V (máx)
Corrente de saída máxima	10 mA
Ganho típico do sensor	10 mV/°C
Faixas de temperaturas de operação	-55°C a 150°C
Corrente de dreno	< 60 μA

O circuito aplicado nos ensaios é mostrado na figura 4 para uma faixa de temperatura de 2°C a 150°C.

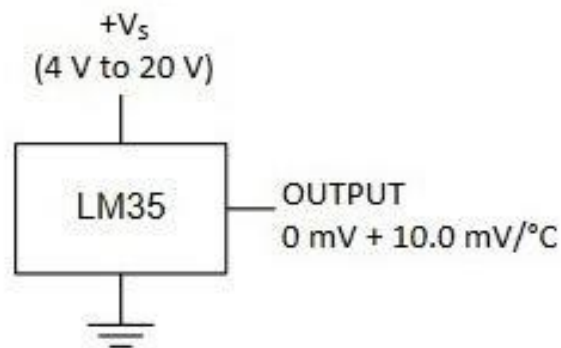


Figura 4 - Sensor de temperatura básico
Fonte: LM35 DATASHEET, (2016).

O encapsulamento do LM35 é o TO-92 muito utilizado em transistores de baixa potência, conforme figura 5.



Figura 5 - Encapsulamento - TO-92
Fonte: LM35 DATASHEET, (2016).

A figura 6 mostra o sensor de temperatura LM35 no encapsulamento TO-92



Figura 6 - O sensor LM35 com encapsulamento TO-92.

1.4.3 O sensor de temperatura Pt-100

O sensor de temperatura Pt-100 é uma termoresistência, muito utilizada na indústria pela estabilidade e precisão, cujo princípio de funcionamento tem como base a variação da resistência do elemento condutor em função da temperatura. Os experimentos com resistências para medição de temperatura iniciaram-se por volta de 1835 com Michael Faraday, mas sua aplicação em processos industriais começa a partir de 1925. O elemento sensor consiste em um fio de platina (Pt) de alta pureza, níquel (Ni) ou cobre, com um encapsulamento de vidro ou cerâmica (ALBUQUERQUE; THOMAZINI, 2011).

A figura 7 mostra a montagem convencional e os principais elementos deste sensor.

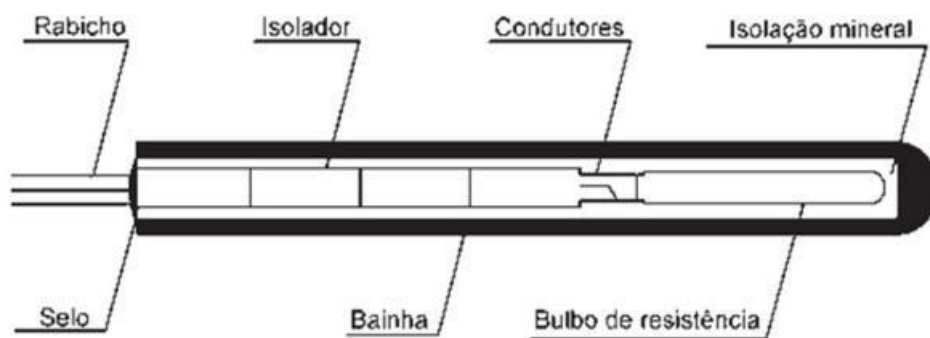


Figura 7 - Termoresistência convencional com bainha preenchida

Fonte: ALBUQUERQUE; THOMAZINI (2011).

As principais características do Pt-100 são:

- a. Possui maior precisão dentro da faixa de utilização do que outros tipos de sensor.
- b. Com ligação adequada, não existe limitação para distância de operação.
- c. Dispensa o uso de fios e cabos de extensão e compensação para ligação, sendo necessários somente fios de cobre comuns.
- d. Se adequadamente protegido, permite a utilização em qualquer ambiente.
- e. Curva de resistência x temperatura mais linear. Menos influenciado por ruídos elétricos (ALBUQUERQUE; THOMAZINI, 2011).

1.4.4 O sensor de corrente

O sensor de corrente SCT-013-000 é um sensor não invasivo que pode medir corrente de até 100A. Usado em projetos com sistemas eletrônicos microcontrolados para automação residencial como medidores de corrente elétrica, proteção de motores AC, iluminação entre outros.

Este sensor funciona como um indutor e responde ao campo magnético ao redor de um condutor portador de corrente. Pela leitura da quantidade de corrente produzida pela bobina é calculado a corrente que está passando pelo condutor, conforme figura 8 (PETRUZELLA, 2014).

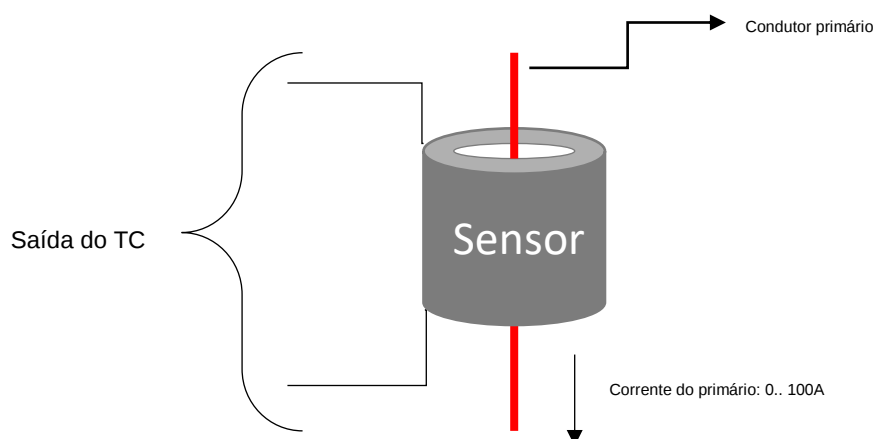


Figura 8 - Medindo a corrente em um condutor.

A figura 9 mostra o condutor dentro do sensor de corrente SCT-013-000 sendo utilizado em uma medição.



Figura 9 - O sensor de corrente SCT-013-000.

O sensor SCT-013-000 possui as seguintes características segundo fabricante:

- a. Corrente de entrada: 0-100A;
- b. Sinal de saída: Corrente/33mA;
- c. Material do núcleo: Ferrite;
- d. Índice de Segurança - Força Dielétrica: 6000V AC/ 1min;
- e. Plug de saída: 3,5mm;
- f. Dimensão abertura: 13 x 13mm;
- g. Temperatura de trabalho: -25 a +70°C (DATASHEET SCT-0133, 2016).

A tabela 3 mostra os tipos de sensores não invasivos com relação ao modo de saída.

Tabela 3 – Parâmetros do sensor de corrente (DATASHEET SCT-0133, 2016).

Modelo	SCT-013-000	SCT-013-030
Input current	0~100 A	0~30 A
Output mode	Current/33 mA	Voltage/ 1V

Modo de saída: Tensão tipo de saída construído no resistor de amostragem, conforme figura 10.

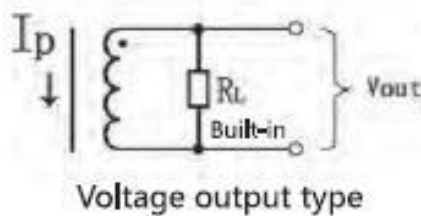


Figura 10 - Saída de tensão

Fonte: DATASHEET SCT-0133 (2016).

Modo de saída: Corrente tipo de saída construído no diodo de proteção, conforme figura 11.

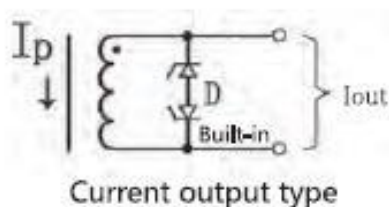


Figura 11 - Saída de corrente

Fonte: DATASHEET SCT-0133 (2016).

1.4.5 Microcontroladores

Os anos 50, 60 e 70 foram importantes para a eletrônica com o surgimento e a evolução dos transistores, circuitos integrados e os microcontroladores.

"O microcontrolador nada mais é que um microcomputador implementado em um único circuito integrado, no qual estão todas as unidades básicas de um computador." (GIMENEZ, 2002, p.14).

"Um microcontrolador um computador de um único *chip* construído para realizar tarefas de controle, como gerenciar um motor ou enviar um dado para uma porta serial." (TOOLEY, 2008, p.315).

Atualmente os microcontroladores são largamente empregados em dispositivos portáteis de baixo custo (XAVIER, 2002). Portanto pode-se dizer que os

microcontroladores em um único *chip* integram elementos computacionais tais como, memórias RAM³ e ROM⁴, CPU⁵, contadores entre outros. A figura 12 mostra a arquitetura de um sistema microcontrolado (GIMENEZ, 2002).

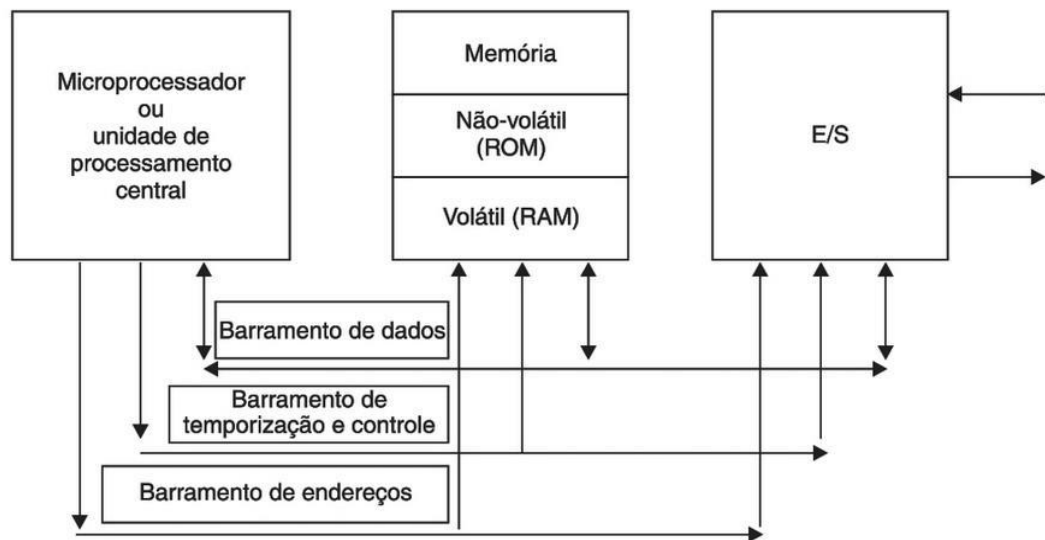


Figura 12 - Arquitetura de sistemas microcontrolados

Fonte: GIMENEZ (2002).

1.4.6 Plataforma de prototipagem open hardware - Arduino Mega2560

Arduino é uma plataforma de prototipagem de código aberto baseado em *hardware* e *software easy-to-use* (fácil de usar), criado pelo italiano Massimo Banzi e pelo espanhol David Cuartielles. Placas Arduino são capazes de ler dados em suas portas de entradas (*input*), conectando nestas uma série de dispositivos, tais como, sensor de luz, temperatura, ou botões tipo *push-bottom* e encaminhar estes dados para uma

³ RAM - **R**andom **A**ccess **M**emory – memória de acesso aleatório é um tipo de memória que permite a leitura e a escrita, utilizada como memória primária em sistemas eletrônicos digitais

⁴ ROM – **R**ead **O**nly **M**emory - memória somente para leitura é um tipo de memória que permite apenas a leitura, ou seja, as suas informações são gravadas pelo fabricante uma única vez e após isso não podem ser alteradas ou apagadas

⁵ CPU – **C**entral **P**rocessing **U**nity – unidade central de processamento é a parte de um sistema computacional, que realiza as instruções de um programa de computador, para executar a aritmética básica, lógica, e a entrada e saída de dado

saída (*output*); como por exemplo a ativação de um motor, ligar um LED – (Diodo Emissor de Luz - *Light Emitting Diode*), publicar algo *online* (MCROBERTS, 2015).

A ação que o Arduino irá executar será realizada pelo conjunto de instruções que serão enviados para o microcontrolador na placa. O Arduino tem como base a programação *Wiring* – plataforma de programação de código aberto para microcontroladores (similar à linguagem C/C++) e para escrever a lógica de programação é necessário o *software* Arduino IDE (*Integrated Development Environment* ou ambiente de desenvolvimento integrado), amigável ao usuário e que possui várias bibliotecas para auxiliar o desenvolvimento da programação (MONK, 2013). São funções do *software* IDE: Permitir o desenvolvimento de um *software* (programa) e enviá-lo à placa para que possa ser executado. (Arduino, 2016)

Ao longo dos anos, o Arduino tem sido a base de diversos projetos, desde projetos do cotidiano por exemplo a automação residencial até protótipos de automóveis que utilizam energia sustentável. A comunidade mundial de *makers*, estudantes, amadores, artistas, programadores e profissionais – reuniu-se em torno desta plataforma *open-source*, suas contribuições acrescentaram grande quantidade de conhecimento acessível, que pode ser de grande ajuda para os iniciantes destas áreas e especialistas similares (ARDUINO, 2016).

Dessa maneira o Arduino é uma ferramenta fácil para prototipagem rápida, destinado a estudantes sem experiência em eletrônica e programação. Assim que chegou a uma comunidade mais ampla, a placa Arduino começou a mudar para se adaptar às novas necessidades e desafios, diferenciando sua oferta de placas de 8 bits simples de produtos para aplicações da *Internet* das coisas, *wearable*⁶, impressão 3D e ambientes corporativos. Todas as placas Arduino são completamente *open-source*, capacitando os usuários para construí-los de forma independente e, eventualmente, adaptá-los às suas necessidades específicas. O *software* também é *open-source*, e está crescendo através das contribuições dos usuários em todo o mundo (STEVAN JR. e SILVA, 2015).

⁶ *Wearable* é a palavra que resume o conceito das chamadas “**tecnologias vestíveis**”, que consistem em **dispositivos tecnológicos que podem ser utilizados pelos usuários como peças do vestuário**.

Arduino Mega 2560 é uma placa da plataforma de prototipagem de código aberto que possui uma gama de recursos para prototipagem e projetos em diversas áreas. Baseada no microcontrolador ATmega 2560 produzido pela Atmel, líder mundial na concepção e fabricação de microcontroladores, soluções de toque capacitivo, lógica avançada, de sinal misto, memória não volátil e componentes de radiofrequência(RF) (ATMEL, 2014).

O Atmega 2560 possui 86 pinos de entradas e saídas digitais, 12 destes podem ser utilizados como saídas PWM⁷. Possui 16 entradas analógicas, 4 portas de comunicação serial. A tabela 4 resume as diferentes configurações do Atmega 2560 com outros microcontroladores também fabricados pela Atmel.

Tabela 4 - Comparativo entre microcontroladores Atmega (ATMEL, 2014).

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

1.4.7 A internet das coisas (*internet of things* - IoT)

"Até o ano 2020 haverá mais de 50 bilhões de dispositivos conectados na Internet e a receita total da *Internet of things* - IoT (Internet das coisas) ultrapassarão os US\$ 1,5 trilhões." (JAVED, 2016).

"Internet das coisas é muito mais do que as coisas ligadas à Internet. É sobre fazer coisas mais inteligentes, dando-lhes a capacidade de sentir, comunicar e responder. Temos cinco sentidos podemos ver, ouvir, paladar, olfato e tato. Da mesma forma, se você adicionar estes sensores para as coisas eles podem fazer o mesmo também". (JAVED, 2016, p.xix).

⁷A **modulação por largura de pulso** PWM sigla em inglês (*Pulse-Width Modulation*) - de um sinal ou em fontes de alimentação envolve a modulação de sua razão cíclica (*duty cycle*) para transportar qualquer informação sobre um canal de comunicação ou controlar o valor da alimentação entregue à carga.

Sendo assim, é possível ligar à Internet um sensor de detecção de fumaça para, remotamente, gerar alertas de uma possível ameaça de incêndio; uma câmera pode ser os nossos olhos, um alto falante pode ser a voz das coisas, um sensor de temperatura para monitorar uma estufa, entre outras aplicações.

Na indústria além de inserir inteligência em objetos comuns é preciso interligá-los com os maquinários que compõem um parque industrial e reconfigurar por completo suas funções, permitindo aos gestores analisarem de forma antecipada falhas nos processos operacionais, o mínimo sinal de perda de produtividade ou estabelecer uma melhor eficiência energética nos equipamentos de uma linha de produção por exemplo (STEVAN JR. e SILVA, 2015).

2 MATERIAIS E MÉTODOS

A seguir serão descritos os materiais empregados durante a execução do presente trabalho, bem como a metodologia geral utilizada para realização de testes e análises.

2.1 Validação dos dados do sensor de temperatura

Para isolar possíveis fontes de erro e garantir a confiabilidade de resultados, antes da realização de qualquer experimento, os dados obtidos pelos sensores LM35 foram validados para aferição de temperatura usando-se, como padrão ouro, o sensor industrial Pt-100. Foram aceitos desvios máximos de $0,3^{\circ}\text{C}$ entre as medidas, para acomodar erros de arredondamento e diferentes tempos de amostragem. A calibração foi realizada via *software* com ajustes corretivos nos valores aferidos.

2.2 Aferição da temperatura

O monitoramento da temperatura interna ao *data center* foi realizado com um de sensor de temperatura LM35 assistidos pelo controlador Atmega2560. Os valores das temperaturas são armazenados através da porta serial entre o Arduino e o microcomputador, para a geração de gráficos estatísticos e posterior análise. O diagrama de blocos da figura 13 representa todos os componentes utilizados neste processo.

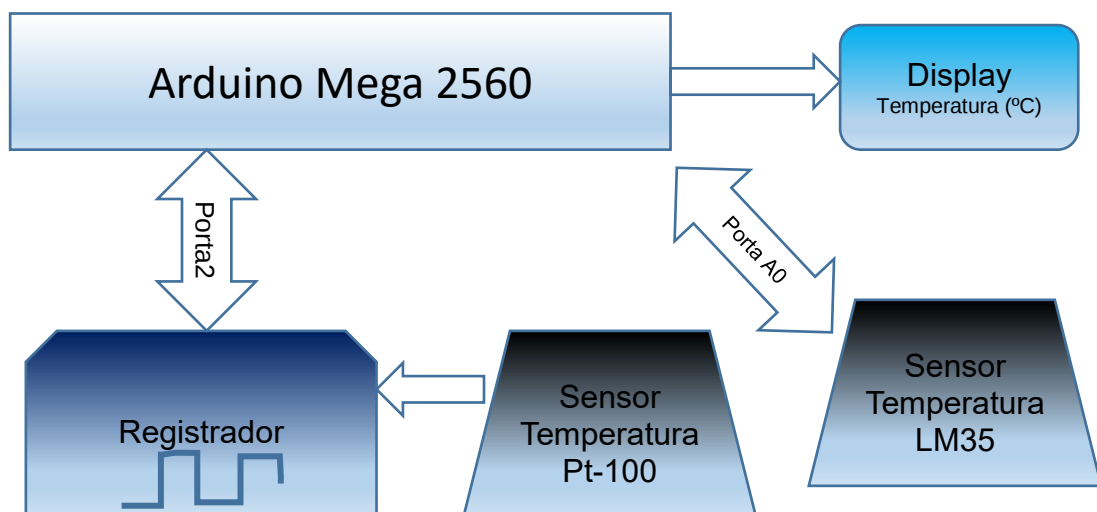


Figura 13 - Diagrama de blocos dispositivos para medições da temperatura.

O sensor de temperatura LM35 e o *display* foram acoplados à placa Arduino Mega conforme o esquema da figura 14.

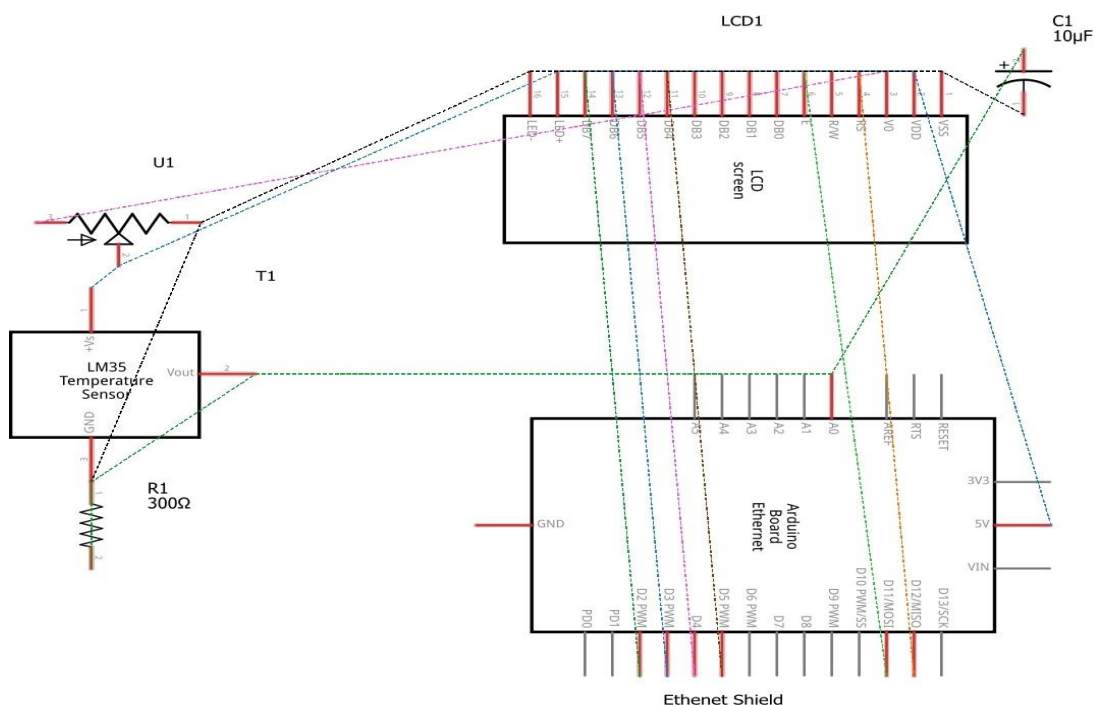


Figura 14 - Esquema eletrônico para a medição de temperatura com LM35 no *data center*.

A figura 15 mostra o sensor LM35 monitorando a temperatura durante os ensaios para calibração do sensor.

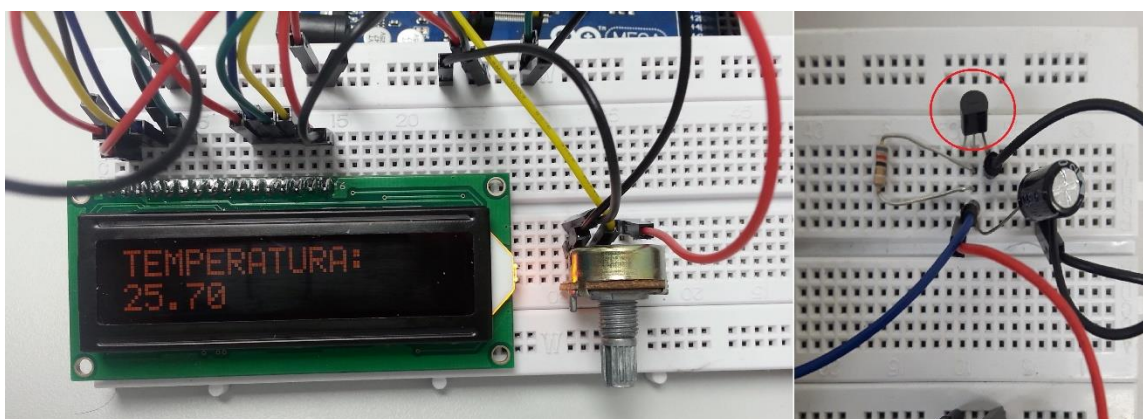


Figura 15 - Sensor LM35 (circulado em vermelho) acoplado ao Arduino Mega 2560.

Os dados coletados pelo sensor LM35 são enviados para a porta serial da placa do Arduino, permitindo a gravação dos dados coletados, conforme figura 16.

```

00:01:21
Temperatura = 21.40 *C
00:01:22
Temperatura = 21.29 *C
00:01:23
Temperatura = 21.40 *C
00:01:24
Temperatura = 21.40 *C
00:01:25
Temperatura = 21.29 *C
00:01:26
Temperatura = 21.40 *C
00:01:27
Temperatura = 21.29 *C
00:01:28

```

Figura 16 - Envio de dados LM35 para porta serial Arduino.

A temperatura externa (porém ainda dentro do prédio da Escola Técnica) é monitorada através do sensor industrial Pt-100. Este sensor foi acoplado ao registrador MV-1000, para possibilitar o registro de dados durante o período de observação, conforme figura 17. Na figura 18 é possível verificar o valor da temperatura externa medida pelo Pt-100 na tela do registrador, sendo este sensor alocado fora do ambiente do *data center*, conforme figura 19.

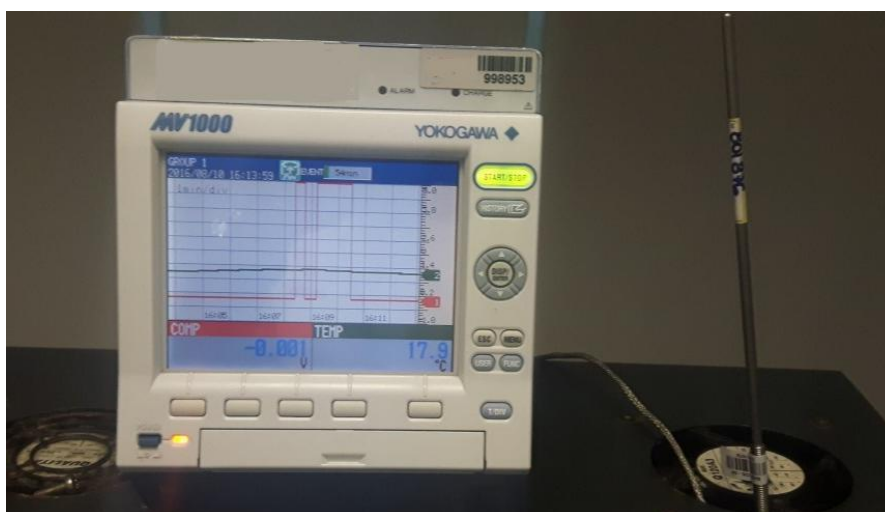


Figura 17 - O registrador MV1000 e o sensor Pt-100.

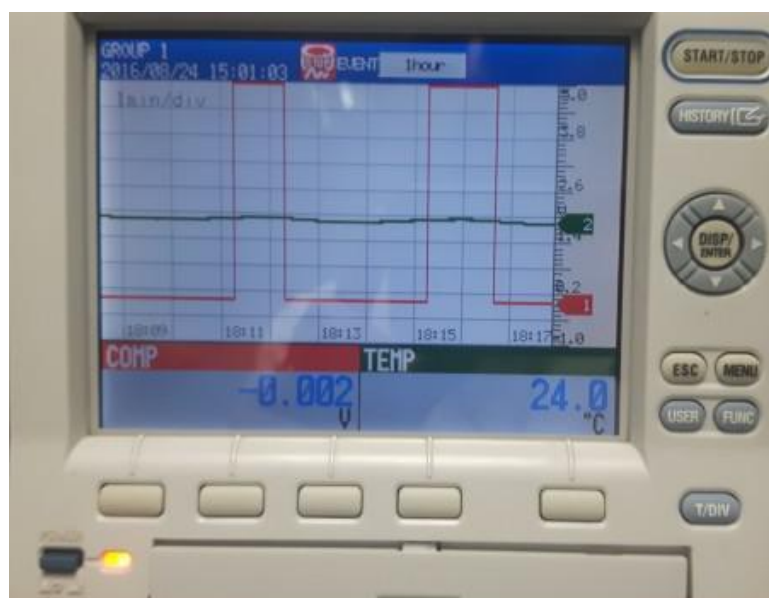


Figura 18 - Temperatura externa medida pelo sensor Pt-100.

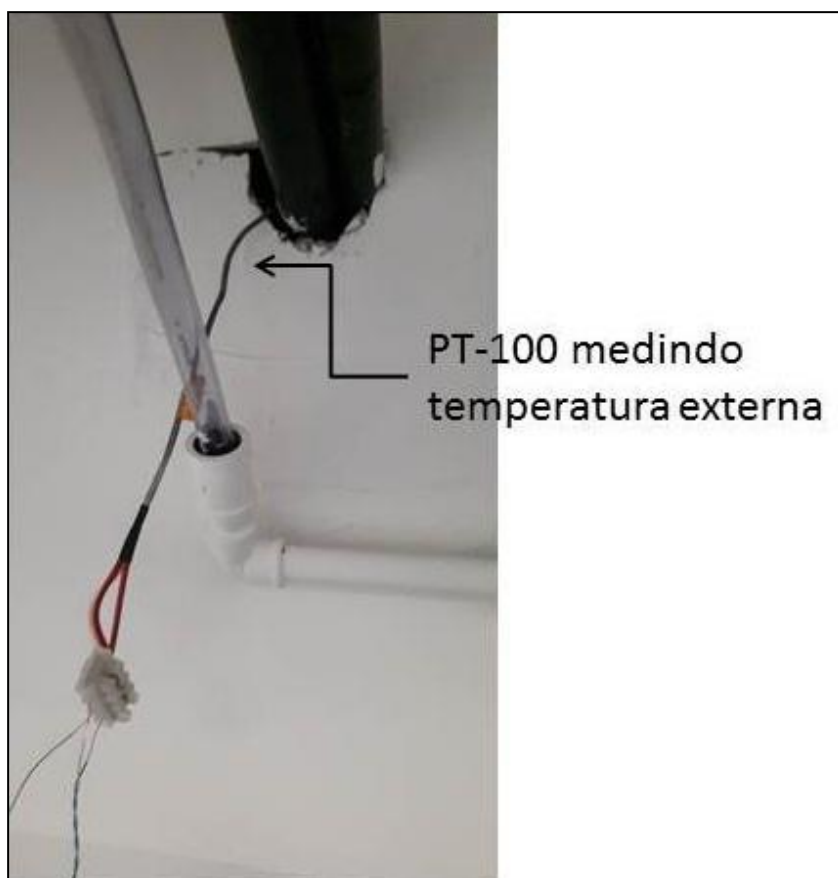


Figura 19 - Pt-100 medindo a temperatura externa.

Além disto, a temperatura da cidade de Santos aferida por centrais meteorológicas, foi acompanhada, para garantir mínimo desvio de temperatura ambiente, i.e. não dependente do experimento.

2.3 Análise de ambiente

Com o objetivo de compreender o comportamento da temperatura interna do ambiente do *data center* analisado, medições de temperaturas foram realizadas em períodos com e sem refrigeração, i.e., com o ar condicionado ligado e desligado. Desta maneira, foi possível observar como a temperatura externa, aliada ao calor gerado por equipamentos funcionando no ambiente, influencia na temperatura interna, foco deste estudo.

Inicialmente, o sistema de refrigeração existente foi configurado para a refrigeração máxima (menor temperatura). Após a temperatura interna se estabilizar em seu mínimo, o sistema de refrigeração foi completamente desligado, e iniciou-se a amostragem da curva de aquecimento. Amostras foram obtidas a cada minuto, de maneira a possibilitar a observação do comportamento a longo prazo. O procedimento foi repetido até que a temperatura interna atingisse o valor crítico de 25°C (máximo permitido por norma). O sistema de refrigeração foi, então, novamente ligado em refrigeração máxima, e amostras com intervalos de um minuto continuaram a ser obtidas. A amostragem foi concluída assim que a temperatura voltou a se estabilizar em seu valor mínimo.

Durante toda a análise a sala permaneceu vazia e com a única porta de acesso fechada, para que fatores externos não tivessem influência nos dados. Além disto, os equipamentos permaneceram em operação normal para simular um dia regular de uso do *data center*.

2.4 Medição da corrente

Para estabelecer uma temperatura ideal garantindo um menor consumo do sistema de refrigeração foi realizado o monitoramento do acionamento do compressor de tal sistema. As medições foram realizadas utilizando o sensor de corrente SCT-103-

000(100A) acoplado a uma placa Arduino Mega 2560, conforme Figura 20 e esquema eletrônico figura 21.

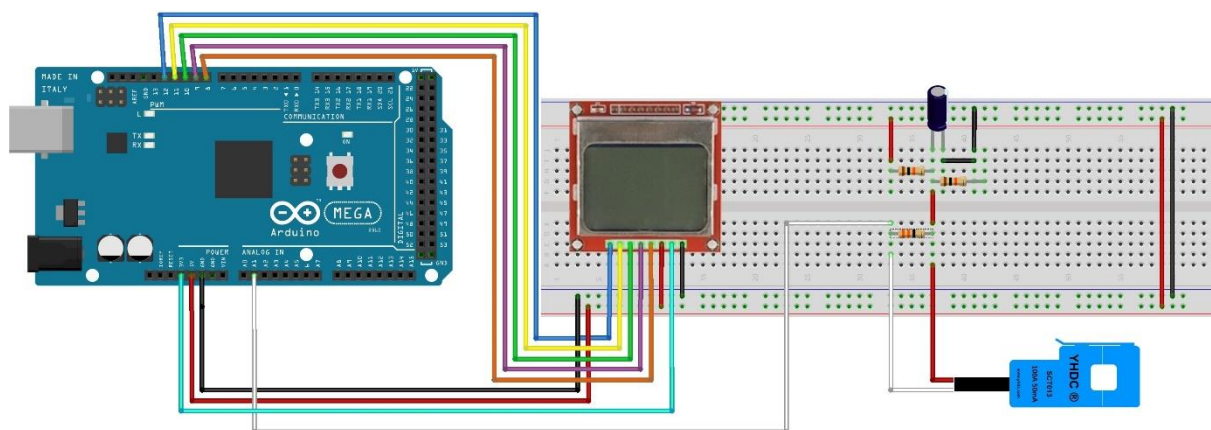


Figura 20 - Medição de corrente com SCT-013-000 no datacenter.

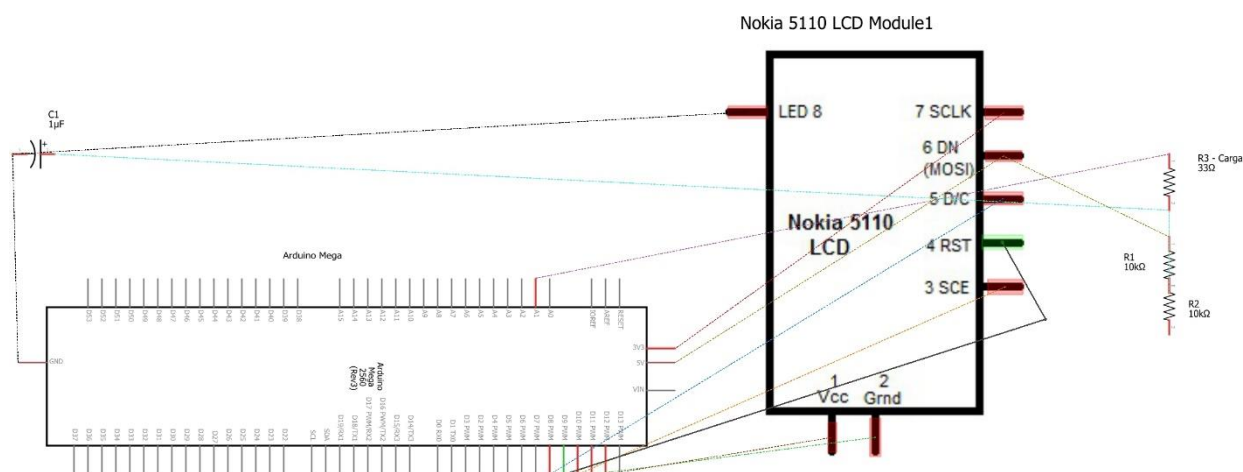


Figura 21 - Esquema eletrônico para medição de corrente no datacenter.

Foi verificado que o acionamento do compressor em estudo consome uma corrente de 10 A. Este fato possibilitou uma abordagem prática para a verificação, via *software* do tempo total de uso do compressor, descrita: o registrador MV-1000 é conectado ao Arduino através de sua porta digital 2. O microcontrolador foi programado para que, enquanto a corrente medida (pelo SCT-103-000 conectado à porta analógica A2 do Arduino) fosse maior ou igual a 10 A, a porta digital 2 receberia um sinal de 5 V, caso contrário a porta permaneceria com 0 V. Esta variação de tensão é amostrada pelo

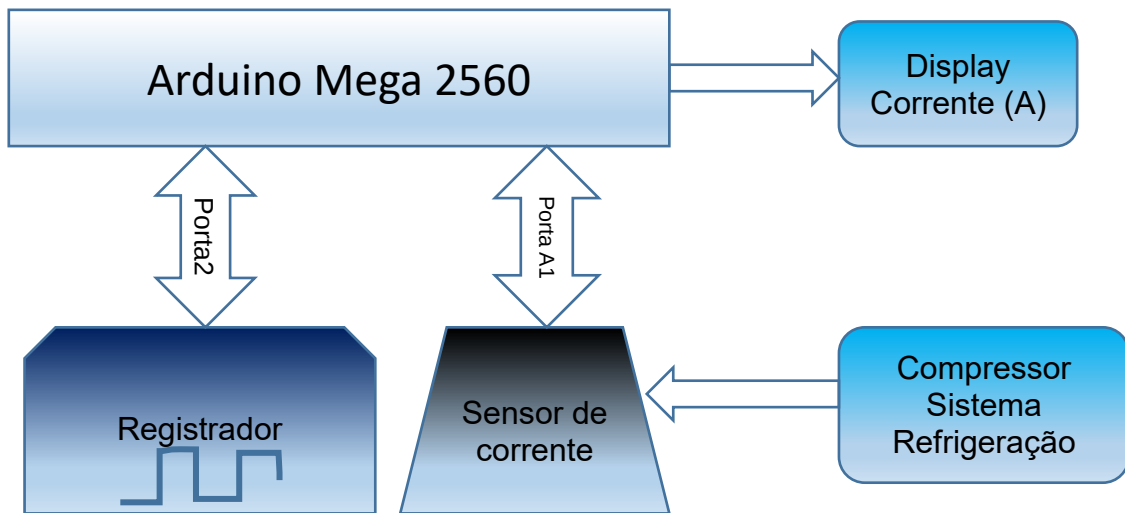


Figura 22 - Diagrama de blocos das medições da corrente.

registrador a cada segundo. O diagrama de blocos da figura 22 ilustra a comunicação entre os dispositivos envolvidos. Apesar do ciclo de atualização do microcontrolador (e, portanto, a possível variação de tensão na porta 2) ser menor do que 1 segundo, o interesse nesta abordagem eram as variações de maior duração - os acionamentos e desligamentos do compressor - e não variações rápidas, portanto a amostragem de 1 Hz foi adequada. Os intervalos de observação foram, tipicamente, de 4 horas, portanto amostragens muito curtas gerariam um volume muito grande de dados de pouco interesse.

2.5 Desenvolvimento de *softwares* e serviços lógicos

A seguir serão descritos o desenvolvimento dos *softwares* e dos serviços lógicos para a conectividade com a *Internet* empregados durante a execução do presente trabalho, detalhes específicos dos códigos fonte serão descritos.

2.5.1 Análise do código sensor de temperatura

O sensor LM35 foi o sensor escolhido para a análise da temperatura no *data center*, este sensor utiliza uma porta analógica do microcontrolador correlacionando variações de tensão em sua saída com variações de temperatura.

A figura 23 mostra o código fonte utilizado para esta análise. O LM35 foi conectado ao Arduino pela porta analógica A0, a variável temperatura armazena o valor do sensor e a variável ADClido realiza a leitura através da função *analogRead*.

Em *void setup()* a porta de comunicação serial foi configurada para os dados serem visualizados no monitor serial do ambiente de desenvolvimento.

Em A0 o valor varia de 0 a 1023, onde 0 corresponde a 0 Volts e 1023 representa 5 Volts. É conhecido que 1°C é igual a 10mV, desta maneira:

$$\text{Tensão em A0} = \text{ADClido} * (5/1023)$$

$$\text{Temperatura} = \text{Tensão em A0} / 10\text{mV}$$

Portanto:

$$\text{Temperatura} = [(\text{ADClido} * (5/1023))] / 10\text{mV}$$

O comando *Serial.print* declara o valor da temperatura no monitor serial do ambiente de desenvolvimento - IDE, com um intervalo de 500 ms para cada nova leitura.

```
const int LM35 = A0; //porta analógica para conexão do LM35
float temperatura; //variável que armazena a valor da temperatura
int ADClido = 0; //variável para leitura do sensor

void setup(){
  Serial.begin(9600); //configuração porta serial
}

void loop(){
  ADClido = analogRead(LM35); //leitura da temperatura na porta A0
  temperatura = (float(ADClido)*5/(1023))/0.01; //calculo do valor da temperatura
  Serial.print("TEMPERATURA:"); //texto apresentado no monitor serial
  Serial.print(temperatura); // valor da variável temperatura exibida no monitor serial
  Serial.println("°C"); //texto apresentado no monitor serial
  delay(500); //intervalo de tempo para cada nova leitura
```

Figura 23 - Código fonte da coleta dos dados da temperatura.

2.5.2 Análise do código sensor de corrente

Conforme já descrito com base nas informações do *datasheet*, os sensores de corrente podem ter na saída uma variação de tensão, ou uma variação de corrente.

No sensor de variação de saída corrente é preciso calcular um componente para o desenvolvimento do protótipo, o “resistor de carga” (*burden resistor*), para gerar a variação de tensão que será armazenada no registrador em conjunto com o Arduino.

Para o cálculo deve-se determinar a corrente máxima a ser medida, como o sensor é de 100A, foi considerado este valor, sabe-se que a expressão para o cálculo da corrente RMS (*root mean square* ou valor médio quadrático) é:

$$I_{rms} = I_p / \sqrt{2}$$

Converter a corrente eficaz RMS para corrente de pico, multiplicando-a por $\sqrt{2}$, logo,

$$\text{Corrente de pico primária (I}_p\text{)} = \text{corrente RMS (I}_{rms}\text{)} \times \sqrt{2} = 100 \text{ A} \times 1.414 = \mathbf{141.4A}$$

A corrente de pico deverá ser dividida pelo número de voltas do transformador de corrente (2000) para determinar a corrente de pico na bobina secundária:

$$\begin{aligned} \text{Corrente de pico secundária} &= \text{Corrente de pico primária} / \text{nº de voltas} = \\ &= 141.4 \text{ A} / 2000 \text{ (nº de voltas)} = \mathbf{0.0707A} \end{aligned}$$

Para melhor exatidão na medição, a tensão através do resistor de carga no pico de corrente deve ser igual a metade da tensão de referência do Arduino ($V_{REF}/2$). Como a tensão de referência no Arduino é de 5V, tem-se:

$$\begin{aligned} \text{Resistor de carga ideal} &= (V_{REF}/2) / \text{corrente de pico secundária} = \\ &= 2.5 \text{ V} / 0.0707A = \mathbf{35.4 \Omega} \end{aligned}$$

O resistor utilizado foi o de 33 Ω , após a preparação do protótipo. Abaixo são descritas as principais linhas de código

São definidas as bibliotecas para o funcionamento do protótipo, “EmonLib.h” é a biblioteca para o funcionamento do sensor de corrente, conforme figura 24.

```
//Carrega as bibliotecas
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
#include "EmonLib.h"
#include <SPI.h>
```

Figura 24 – Inserção das bibliotecas sensor de corrente.

Em seguida são declarados os valores para os pinos do registrador e do sensor de corrente no Arduino, assim como os valores da tensão da rede elétrica e da corrente do sistema de refrigeração do *data center*, conforme figura 25.

```
const int ledPin = 2; //pino registrador Yokogawa
const int ref = 10; //valor da corrente de pico sistema refrigeração
int rede = 220; //Tensao da rede eletrica
int pino_sct = A1; //Pino do sensor SCT
```

Figura 25 – Definição dos valores para tensão, corrente e pinagem.

A seguir é declarado o cálculo da corrente, as linhas para a exibição (*display* e no serial monitor) e a condição para que o registrador registre os momentos em que o compressor do sistema é acionado, quando a condição for uma corrente (Irms) maior ou igual a corrente de referencia (ref = 10A), conforme figura 26.

```
void loop()
{
    //Calcula a corrente
    double Irms = emon1.calcIrms(1480);

    //Mostra o valor da corrente no serial monitor e display
    Serial.print("Corrente : ");
    Serial.print(Irms); // Irms

    if (Irms >= ref){
        digitalWrite(ledPin, HIGH);
        // delay(500);
    }
    else{
        digitalWrite(ledPin, LOW);
    }
}
```

Figura 26 – Cálculo da corrente, exibição dos dados e condição para registro.

2.5.3 Análise do código do servidor web - Arduino shield

O monitoramento dos dados do sensor LM35 devem ser disponibilizados à *Internet*, para isso o *Ethernet Shield* é acoplado ao Arduino Mega2560. O *Ethernet Shield* é representado pela figura 27.



Figura 27 - *Ethernet shield*.

O *Ethernet shield* possui o chip W5100, um controlador *Ethernet* que incorpora os protocolos de comunicação TCP/IP. O controlador *Ethernet* permite a conexão à *Internet* para sistemas embarcados.

No ambiente de desenvolvimento integrado (IDE) do Arduino é necessário, inicialmente, incluir a biblioteca SPI.h - SPI - *Serial Peripheral Interface*, protocolo que realiza a comunicação do microcontrolador com outros componentes permitindo assim a troca de dados entre as camadas e os demais protocolos envolvidos na transmissão conforme figura 28.

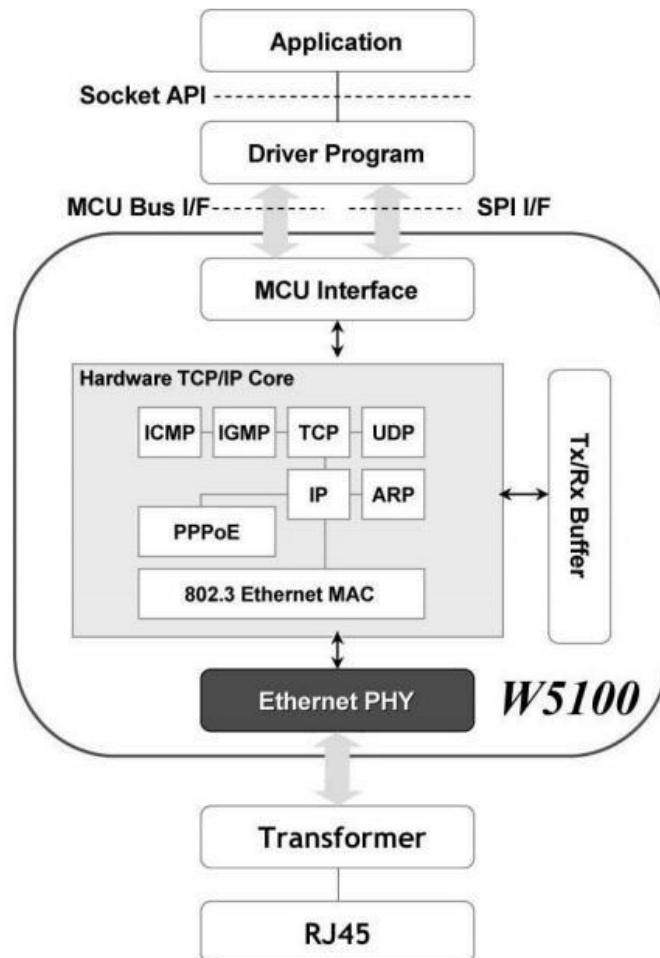


Figura 28 - Estrutura de comunicação do Shield Ethernet
 Fonte: CHIP W5100 SHIELD ETHERNET DATASHEET (2016).

Outra biblioteca importante para a transmissão dos dados da temperatura via Internet é a biblioteca Ethernet.h. do padrão de comunicação de rede local *Ethernet*.

“O padrão *Ethernet* foi originalmente idealizado na tese de doutorado de Robert Metcalfe (fundador da 3Com) enquanto trabalhava nos laboratórios da Xerox. Ele queria encontrar uma solução para conectar computadores em rede local e idealizou que um barramento comum compartilhado entre todas as máquinas era a maneira mais natural de fazê-lo”. (KUROSE; ROSS, 2013, p.348).

O padrão *Ethernet* possui um endereço físico MAC (*Media Access Control*) ou controle de acesso ao meio, que está associado à interface de comunicação, sendo este endereço único em uma rede de computadores, ou seja, não existem duas portas

de comunicação ou dois dispositivos com o mesmo endereço MAC (KUROSE; ROSS, 2013).

A figura 29 exemplifica a inserção das bibliotecas no ambiente de desenvolvimento (IDE).

```
//declaração das bibliotecas para comunicação em rede
#include <SPI.h>
#include <Ethernet.h>
```

Figura 29 - Bibliotecas para comunicação em rede.

Para que a comunicação seja estabelecida entre o usuário utilizando um aplicativo navegador e o servidor que será o Arduino, é necessário inserir os endereços físico e lógico, em seguida é declarado um servidor *Ethernet* definindo qual porta de comunicação será utilizada. A porta padrão é 80, porém a maioria dos roteadores de redes, por questões de segurança, bloqueiam este valor. Portanto foi utilizada a porta 8080, conforme figura 30.

```
//arrays para armazenar o endereço físico (MAC)
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
//declaração do endereço lógico IP
IPAddress ip(10, 201, 1, 230);

// Inicia o servidor na porta 8080
// (port 80 is default for HTTP):
EthernetServer server(8080);
```

Figura 30 - Declaração dos endereços e portas de comunicação.

A próxima etapa é a declaração da rotina de inicialização. É inicializada a comunicação *Ethernet*, passando pelos endereços MAC e IP do dispositivo - *Ethernet.begin(mac, ip)* - Em seguida o servidor deverá começar a monitorar a entrada de conexões - *server.begin()* -, conforme figura 31.

```
// inicia conexão Ethernet e o servidor
Ethernet.begin(mac, ip);
server.begin();
```

Figura 31 - Inicialização do servidor e das conexões – portas.

Em seguida foram declaradas as linhas de código para que o cliente do serviço estabeleça conexão com o servidor. O comando *EthernetClient* verifica se há dados

disponíveis para leitura a partir do servidor. O cliente é o *software* para navegação na web que irá estabelecer a conexão. A condição *if* utiliza uma variável *current_line_is_blank = true*, a requisição HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto) termina com uma linha em branco (*stop bit*). A condição *while* verifica se o cliente ainda está conectado e, em caso positivo, o código fica aguardando em *looping*, conforme figura 32.

```
EthernetClient client = server.available(); //
if (client) {
  // uma requisição http termina com uma linha em branco
  boolean current_line_is_blank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read(); // Recebe caractere do browser e guarda em "c"
      // Se chegamos ao fim da linha (recebeu uma nova linha
      // de caractere) e a linha está em branco, a requisição http termina
      // para que possamos enviar uma resposta!
      if (c == '\n' && current_line_is_blank) {
        // Envia um cabeçalho padrão de resposta HTTP
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println();
        break;
      }
    }
  }
}
```

Figura 32 - Código fonte cliente Ethernet.

Na figura 33 a condição *if* verifica quando a variável "c" for verdadeira, ou seja, início de uma nova linha a ser transmitida; ou "c" for falsa, quando ainda existe um caractere a ser transmitido na linha atual. O *delay* representa o intervalo de tempo para que o navegador do cliente receba os dados e na sequência encerra-se a sessão de transmissão.

```

if (c == '\n') {
    // Se começamos uma nova linha...
    current_line_is_blank = true;
}
else if (c != '\r') {
    // Se ainda temos um caractere na linha atual...
    current_line_is_blank = false;

    // Damos ao navegador um tempo para receber os dados
    delay(10);
    client.stop();// Fecha a conexão

```

Figura 33 - Continuidade na transmissão de dados.

Para que o usuário possa ter acesso à temperatura via rede utilizando um navegador em seu dispositivo a figura 34 mostra o código HTML (*HyperText Markup Language* ou Linguagem de Marcação de Hipertexto) inserida na linguagem C no IDE estabelecendo assim a conexão com a camada aplicação

```

client.println("<html>"); //Início do código HTML
client.println("<h3><body text = 'blue'> Sala dos Servidores </body></h3>");
client.println("<h3><body text = 'blue'> Temperatura</body></h3>");
client.println(temperatura);
client.print("C");
client.println("</html>"); //Fim do código HTML

```

Figura 34 – Código HTML para acessar a temperatura via rede.

3 RESULTADOS E DISCUSSÃO

A seguir serão discutidos os resultados analisados dos experimentos com o sistema de refrigeração e o comportamento da temperatura em condições regidas pela norma internacional EIA/TIA 942 e estabelecer os limites de temperatura adequados a um *data center*. O intervalo de temperatura estipulado pela norma é de (20 – 25 °C), levou-se em consideração a temperatura externa ao *Data center* nos dias em que os dados foram levantados.

3.1 Verificação do ambiente

Para uma análise do comportamento da temperatura do *data center* foi realizado o primeiro experimento onde a temperatura inicial de 22 °C foi definida como ponto de estabilização da temperatura ambiente. A partir do desligamento do sistema de refrigeração a temperatura aumenta gradativamente e de forma não linear, sendo que eventualmente esta iria se estabilizar em um determinado momento, porém a partir de 25 °C, temperatura limite da norma EIA/TIA 942, o sistema de refrigeração foi novamente acionado.

Com o sistema de refrigeração em funcionamento a temperatura cai rapidamente voltando a estar dentro dos limites da norma e gradativamente volta ao ponto de estabilização, com isso verificou-se que tempo para a temperatura atingir o limite da norma e voltar novamente para o ponto de estabilização foi de trinta e três minutos, conforme verificado na figura 35.

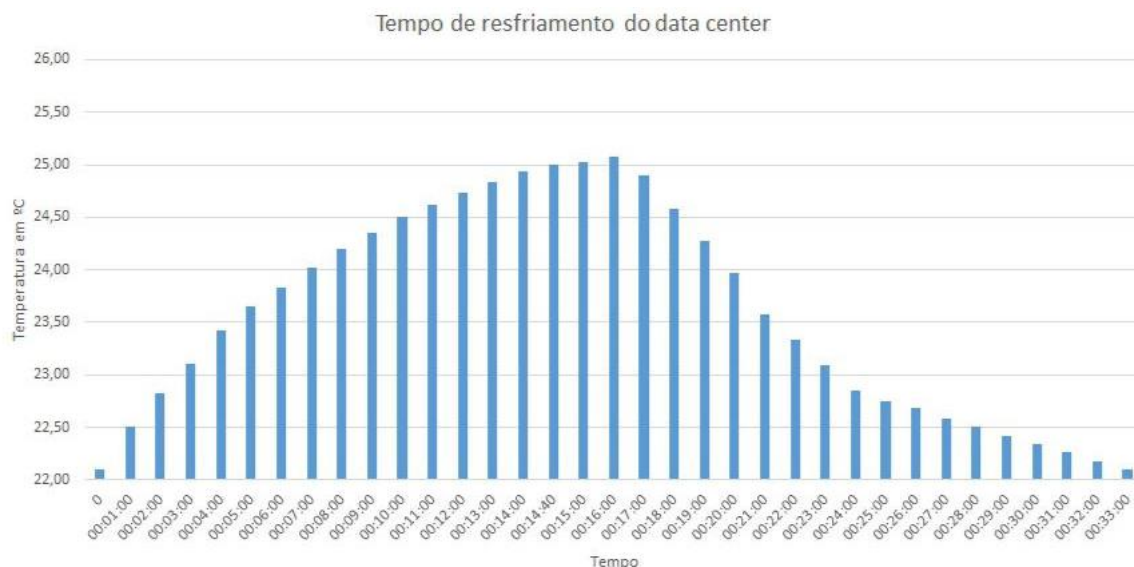


Figura 35 - Tempo de resfriamento do data center.

No segundo experimento com o sistema de refrigeração desligado e com servidores de testes similares aos utilizados no simulador foi constatado um rápido aumento na temperatura do processador destes equipamentos. Para evitar qualquer tipo de dano às máquinas de teste ou as utilizadas no simulador definiu-se no *setup* destes equipamentos uma temperatura de segurança (70 °C) para que as máquinas entrem em processo de *shutdown* automaticamente, conforme figura 36. Os servidores possuem discos redundantes com sistemas espelhados.

Reset Case Open Status	[Disabled]
Case Opened	Yes
Ucore	1.376V
DDR3 1.5V	1.504V
+3.3V	3.392V
+12V	12.302V
Current System Temperature	30°C
Current CPU Temperature	62°C
Current CPU FAN Speed	0 RPM
Current SYSTEM FAN Speed	0 RPM
CPU Warning Temperature	[70°C/150°F]
CPU FAN Fail Warning	[Disabled]
SYSTEM FAN Fail Warning	[Disabled]
CPU Smart FAN Control	[Enabled]
CPU Smart FAN Mode	[Auto]
System Smart FAN Control	[Enabled]

Figura 36 - Configuração temperatura de segurança para desligamento dos servidores.

Com os servidores do simulador ligados e executando atividades de simulação, foi utilizado o mesmo ponto de estabilização e observou-se um aumento gradativo da temperatura, porém o sistema de refrigeração permaneceu desligado passando da temperatura máxima permitida pela norma.

Aos vinte e seis minutos com uma temperatura próxima dos 27,5 ° C, os servidores iniciaram o processo de *shutdown* (downtime), desligando o sistema simulador. Com isso verificou-se que os servidores funcionam mesmo ultrapassando a temperatura máxima da norma, mas por um breve período (dez minutos aproximadamente) conforme figura 37.



Figura 37 - Tempo e temperatura para o travamento dos servidores do *data center*.

O aumento da temperatura para valores acima da norma, com o sistema desligado, ocorre rapidamente, indicando que há uma grande produção de calor no ambiente interno, o que justifica o uso constante do ar condicionado, além disso o tempo para ultrapassar o valor de 25°C foi bem parecido em relação ao primeiro experimento (em torno de 15 minutos nas duas curvas) o que sugere reprodutibilidade dos dados e dá maior confiabilidade aos resultados.

Mesmo o *data center* do simulador portuário sendo classificado como TIER1, as duas análises realizadas evidenciam os limites da norma internacional e justificam o

monitoramento do sistema de refrigeração, pois sem ele a prática operacional do simulador torna-se inviável.

3.2 Verificação do consumo elétrico

Como verificou-se a utilização de um sistema de refrigeração se faz necessário, porém, a sua utilização em regime 24x7 (24 horas por dia, 7 dias por semana) consome grande quantidade de energia elétrica a um alto custo.

Para análise do consumo, foram feitas medições em vários períodos do dia durante uma semana, todas sempre levando em consideração a temperatura externa e a temperatura interna configurada no *data center*.

O gráfico mostra a frequência de acionamento do compressor do sistema de refrigeração em Volts (porta digital do Arduino ligado no registrador) em relação a temperatura em graus Celsius, com o sistema configurado para 20° C a uma temperatura externa média de 24,5 °C. Nota-se que a variação da temperatura está associada ao funcionamento do compressor do sistema, conforme figura 38.

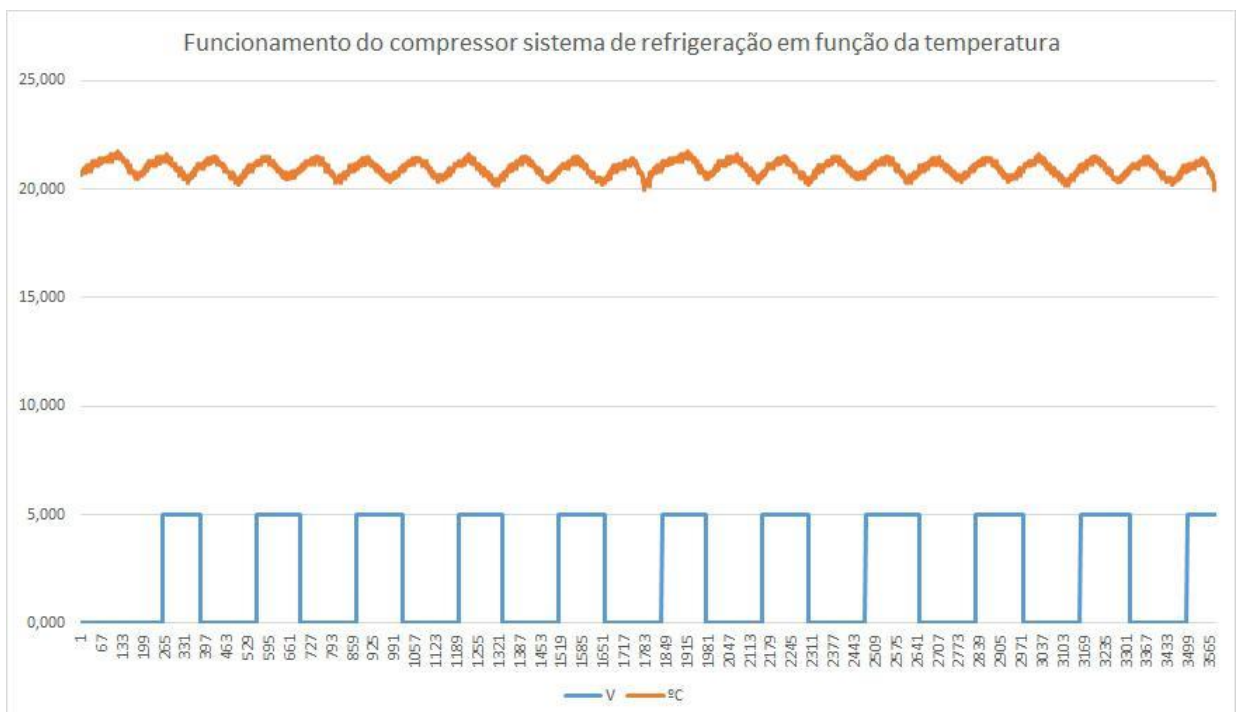


Figura 38 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 20 °C.

A temperatura oscilou entre 20 °C e 21 °C, nota-se que em média o compressor permanece ligado 161 segundos e desligado 176 segundos.

No segundo gráfico o sistema foi configurado para 21 °C a uma temperatura externa média de 25 °C, nesta situação verifica-se que a frequência de acionamento do compressor é em média 250 segundos e tempo de permanência do compressor desligado é de 326 segundos, com a temperatura oscilando entre 21 e 22 °C, conforme figura 39.

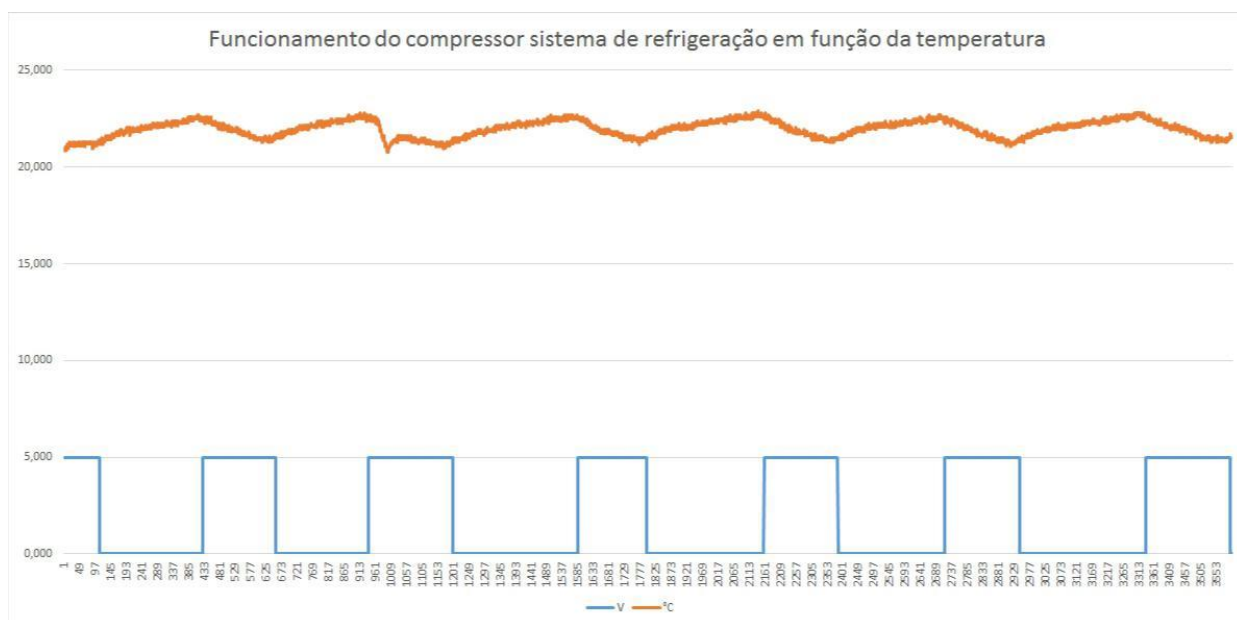


Figura 39 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 21 °C.

A medição seguinte a temperatura foi configurada para 22 °C a uma temperatura externa média de 25 °C, nota-se que frequência de acionamento do compressor é de 133 segundos e desligado 244 segundos, com a temperatura interna do datacenter oscilando entre 22 °C e 23 °C, conforme figura 40.

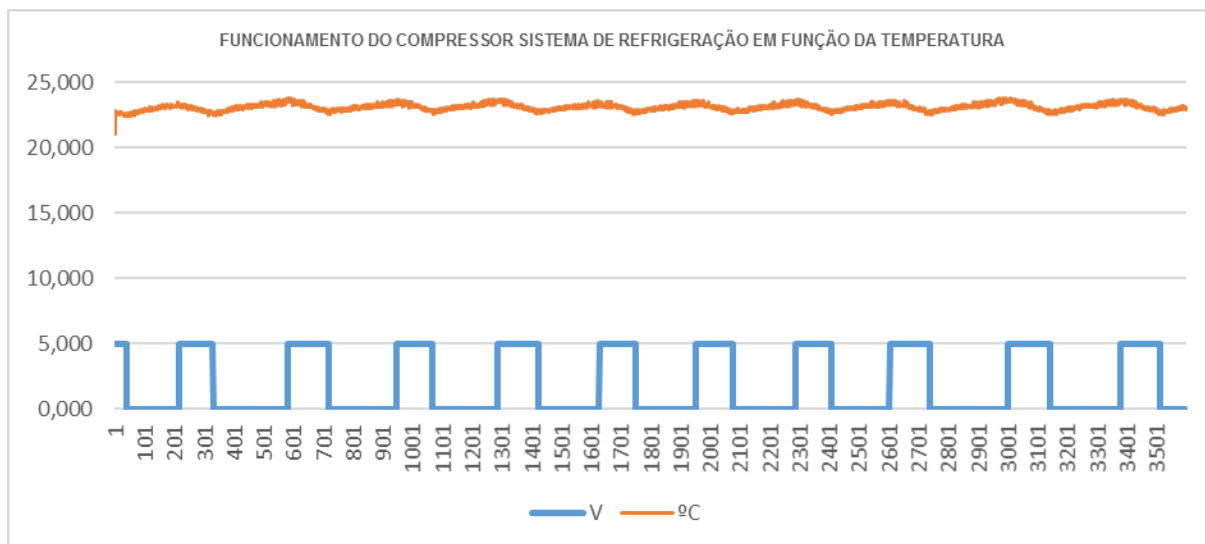


Figura 40 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 22 °C.

No quarto gráfico a temperatura foi configurada para 23 °C a uma temperatura externa média de 24,8 °C, nota-se um maior equilíbrio do sistema para manter a temperatura constante, o tempo de acionamento do compressor é de 138 segundos e 207 segundos desligado, com a temperatura oscilando entre 23 °C e 23,5 °C, conforme figura 41.

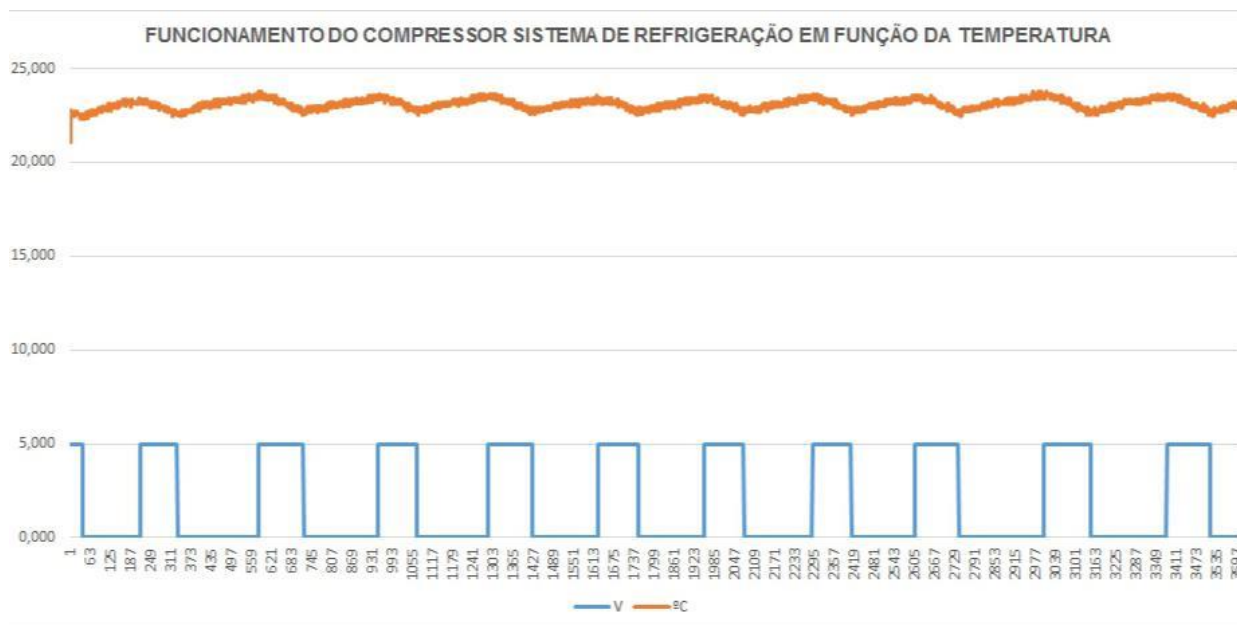


Figura 41 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 23 °C.

A medição seguinte foi configurada para 24 °C a uma temperatura externa média de 25,8 °C nota-se que a temperatura do ambiente é mais constante que as demais medições, porém muito próximo da temperatura máxima segundo norma EIA/TIA 942, oscilando entre 24 °C a 25 °C. O acionamento do compressor é de 123 segundos em média e o compressor desligado manteve-se 189 segundos, conforme figura 42.

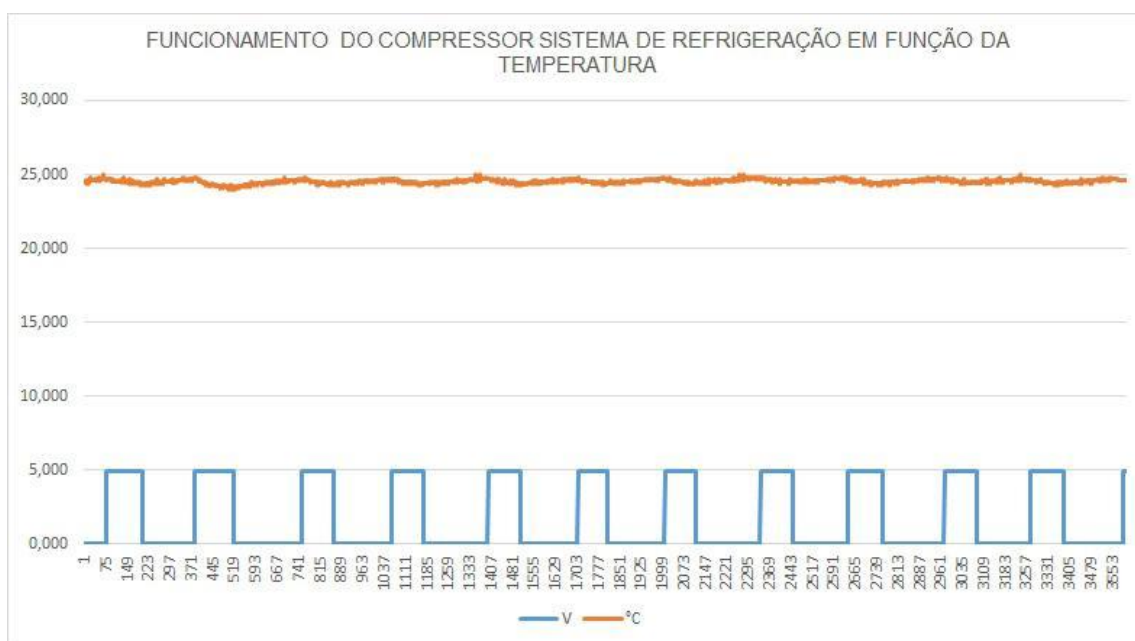


Figura 42 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 24 °C.

A última medição configurada a 25 °C a uma temperatura externa média de 26 °C, o sistema de refrigeração não conseguiu manter a temperatura ambiente dentro do limite da norma, embora tenha o melhor nível de consumo, esta temperatura está muito próxima a temperatura crítica de travamento dos servidores, conforme figura 43.

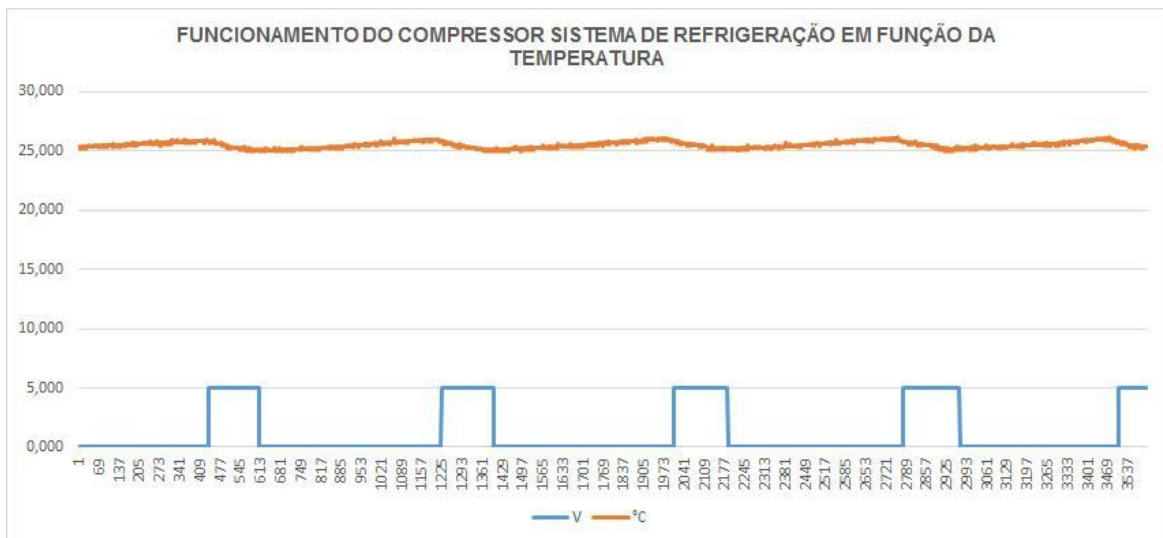


Figura 43 - Funcionamento do compressor sistema de refrigeração em função da temperatura a 25 °C.

Os dados analisados nos gráficos das temperaturas em relação ao acionamento do compressor estão sintetizados na tabela 5. A média de tempo de acionamento do compressor foi calculada no intervalo de 1 hora, quando o pico permanece a 5 volts, tensão enviada para a porta microcontrolador e o mesmo intervalo de tempo quando a tensão enviada é 0 volts, ou seja, o compressor permanece desligado.

Tabela 5- Síntese dos dados analisados temperatura x funcionamento do compressor.

Configuração temperatura interna (°C)	Oscilação da temperatura (°C)	Temperatura externa (°C)	Média de tempo de acionamento do compressor (s)	Média de tempo do compressor desligado (s)
20	20/21	24,5	161	176
21	21/22	25,0	250	326
22	22/23	25,0	133	244
23	23/23,5	24,8	138	207
24	24/25	25,8	123	189
25*	25/26	26,0	-	-

* A temperatura excedeu o limite da norma EIA/TIA 942

3.3 Eficiência energética

A tabela 6 mostra os dados coletados durante o período de testes, de acordo com as temperaturas consideradas pela norma, indicando quantas vezes o compressor do sistema de refrigeração foi acionado, levando em consideração também a potência do equipamento e as horas utilizadas.

O cálculo utilizando para verificar o intervalo de funcionamento do compressor é medido em segundos. Considerando uma hora de execução do sistema, a potência do equipamento é especificada pelo fabricante, as horas utilizadas são calculadas da seguinte forma:

$$\text{horas utilizadas} = \left(\frac{\text{func comp}}{3600} \right) \times 24$$

O consumo mensal é medido em kWh – quilowatt multiplicado por hora. Para este cálculo leva-se em consideração a Potência (P), definida como quantidade de energia (ΔE) por intervalo de tempo (Δt): $P = \Delta E / \Delta t$.

Sendo assim, a energia (ΔE) deve ser dada por potência (P) vezes intervalo de tempo (Δt): $\Delta E = P \times \Delta t$.

Como o período de interesse é de um mês, multiplica-se a expressão por trinta.

$$\Delta E = P \times \Delta t \times 30.$$

Diante dos dados analisados anteriormente, considerando-se que os resultados obtidos possuam repetibilidade para diferentes temperaturas externas, ou seja, os dados observados representam o padrão geral de comportamento da temperatura interna, considerou-se a melhor condição de consumo e eficiência energética a temperatura de 23°C para a configuração do sistema de refrigeração. Na tabela 6 temos um resumo do cálculo de consumo de cada temperatura analisada no período de estudo.

Tabela 6 - Consumo do sistema de refrigeração de acordo com a temperatura configurada.

Temperatura °C	Intervalo de funcionamento do Compressor (s)	Potência equipamento (kW)	Horas utilizadas	KWh mensal
20	1539	2,42	10,26	744,88
21	1530	2,42	10,20	740,50
22	1320	2,42	8,80	638,90
23	1203	2,42	8,02	582,25
24	1273	2,42	8,49	616,13
25	814	2,42	5,43	393,98

Com o estudo conclui-se que para a temperatura configurada a 23 °C o índice de eficiência energética é de 22%, levando em consideração que a pior condição de temperatura ajustada para a climatização é de 20 °C, onde o consumo mensal apresentado na tabela 6 é de 744,88 KW/h.

Para entender os valores do índice de eficiência da Tabela 6, pode-se fazer uma regra de três dividindo os valores das temperaturas do range analisado pela temperatura de maior consumo da tabela 6.

Tabela 7 – Índice de eficiência em função da temperatura.

Temperatura °C	Horas utilizadas	Índice em eficiência %
21	10,20	0,5
22	8,80	14
23	8,02	22
24	8,49	17
25	5,43	47

Deve-se destacar mais uma vez que a temperatura configurada a 25 °C, embora na tabela 7 indique o melhor índice de eficiência, esta quando ajustada no ambiente analisado ultrapassou o limite de 25 °C, desta forma, não atendendo a recomendação da norma.

A tabela 8 demonstra a economia que este estudo representa em Reais (R\$) sem levar em consideração taxas e impostos. Para a base de cálculo foi utilizada a tarifa comercial, sendo esta dividida em:

- TUSD - Tarifa de Uso do Sistema de Distribuição de valor igual a R\$ 0,1804
- TE - Tarifa de Energia de valor igual a R\$ 0,2240

Portanto o valor a ser pago é igual (TUSD + TE + BANDEIRAS)

Tabela 8 – Relação de consumo em Reais (R\$) para cada temperatura em relação as bandeiras da fornecedora de energia (AES Eletropaulo, 2016).

Temperatura °C	KWh mensal	Bandeira Verde Sem acrécimo na tarifa (R\$)	Bandeira Amarela Acrécimo de R\$ 0,015 a cada 1 KW/h consumido	Bandeira Vermelha Acrécimo de R\$ 0,045 a cada 1 KW/h consumido
20	744,88	300,88	312,02	334,34
21	740,50	299,25	310,35	332,55
22	638,90	258,40	267,99	287,16
23	582,25	235,35	244,08	261,54
24	616,13	249,10	258,34	276,82
25	393,98	159,32	165,23	177,05

Considerando a bandeira vermelha a situação de maior consumo o ajuste da temperatura de 20 °C para 23 °C representa em reais uma economia de R\$ 72,80 mês. Em um ano este ajuste representaria uma economia de R\$ 873,60. O estudo ainda pode ser replicado para a rede de ensino desta instituição que atualmente possui 91 unidades, com *data centers* similares, no estado de São Paulo, neste caso os valores economizados seriam na ordem de R\$ 80.000,00 anuais.

Em escala maior grandes empresas como *Microsoft*, *Facebook* e *Google* adotaram o aumento da temperatura para garantir um menor custo e maior eficiência energética. A Intel recomenda também seus clientes a aumentar a temperatura nos *data centers*, segundo a empresa a cada grau aumentado economiza-se até 4% (MILLER, 2008).

Neste estudo constatou-se que para cada grau aumentado a economia chegou a 7% dentro das condições durante os testes.

Desta forma o sistema pode ficar monitorando o *data center* de maneira contínua futuramente, para verificar a confiabilidade do sistema de refrigeração que, como mostrado, possui erros inerentes de manutenção de energia. Além disso, o sistema

de controle pode ter suas informações disponibilizadas na *Internet* e pode ajudar a economizar ainda mais recursos financeiros da escola, visto que qualquer problema na refrigeração será notado e proporcionará uma atitude corretiva rápida, evitando o *downtime* e evitando o desgaste (e quebra) dos equipamentos presentes no *data center* que, como demonstrado também, ocorrem rapidamente assim que a temperatura fica acima dos limites recomendados pela norma.

4 CONCLUSÃO

Neste trabalho foi feito um estudo detalhado em um sistema simulador e seu *data center*, demonstrado a relação entre a temperatura e o consumo de energia, com base na norma EIA/TIA 942, levando-se em consideração que o *data center* estudado é de nível básico (TIER 1).

Com a análise das temperaturas medidas ao longo deste trabalho, pode-se provar a tendência dos grandes *data centers* que estão redefinindo o ponto de ajuste de resfriamento do ambiente, aumentando a temperatura dentro do que é estabelecido pela norma, proporcionando uma melhor eficiência e melhores custos. Com este estudo observou-se que mesmo tendo-se um melhor índice de eficiência energética, a temperatura de 25°C quando configurada no sistema de refrigeração excedeu os limites da norma. A melhor temperatura para o *data center* de nível básico foi de 23°C proporcionando uma redução de custo de até 22%. Nos dias em que a temperatura foi monitorada não foi relatado em nenhum período de *downtime* dos servidores do *data center*.

Para a análise e controle da temperatura utilizou-se sensores de temperatura em conjunto da plataforma de prototipagem eletrônica microcontrolada, que foi adequada para monitorar a temperatura de ambientes fechados, aliado ao conceito de *Internet das coisas*, fornecendo a temperatura em tempo real para os usuários do simulador portuário, como por exemplo, o instrutor do simulador ou o administrador do *data center*, o sistema permite que estes usuários possam monitorar a temperatura.

Essa possibilidade foi demonstrada utilizando o protocolo de comunicação *Ethernet*, muito difundido em redes locais conectadas à *Internet*.

O controle do sistema de refrigeração do *data center* futuramente poderá ser melhorado, alertando os usuários por *e-mail* ou mensagens instantâneas, caso haja um aumento da temperatura em condições adversas ou o acionamento remoto do sistema de refrigeração quando este for desativado por queda de energia.

REFERÊNCIAS

- ANSI/TIA. **TIA STANDARD**. Operational parameters. Arlington, U.S.A: TELECOMMUNICATIONS INDUSTRY ASSOCIATION: 148 p. 2005
- ALBUQUERQUE, P. U. B.; THOMAZINI, D. **Sensores Industriais - Fundamentos e Aplicações**. 8º. ed. São Paulo: Erica / Saraiva, 2011. 225p.
- ALVES, J. L. L. **Instrumentação, controle e automação de processos**. Rio de Janeiro: LTC Editora, 2010.
- ATMEL. **Datasheet microcontrolador Atmega** CORPORATION, A.: 435 p. 2014.
- GIMENEZ, S. P. **Microcontrolador 8051**. 1º. ed. São Paulo: Pearson Editora do Brasil Ltda, 2002. 258p.
- INSTRUMENTS, T. **LM35 Precision Centigrade Temperature Sensors**. Dalas, Texas 2016.
- JAVED, A. **Building Arduino Projects for the Internet of Things - Experiments with Real-World**. 1º. ed. Lake Zurich, Illinois, USA: Apress, 2016. 285p.
- KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet: uma abordagem top-down**. 6º. ed. São Paulo: Pearson Education do Brasil, 2013. 636p.
- MARIN, PAULO S. **Data Centers: desvendando cada passo: conceitos, projetos, infraestrutura física e eficiência energética**. 1º ed. São Paulo: Érica, 2011 321p.
- MCROBERTS, M. **Arduino Básico**. São Paulo: Novatec Editora, 2015.
- MONK, S. **Programação com o Arduino - começando com sketches**. Porto Alegre - RS: Bookman Editora, 2013.
- PETRUZELLA, F. D. **Eletrotécnica II**. Porto Alegre: AMGH Editora, 2014.
- STEVAN JR., S. L.; SILVA, R. A. **Automação e Instrumentação Industrial com Arduino - Teoria e projetos**. São Paulo: Editora Érica, 2015.
- TOOLEY, M. **Circuitos Eletrônicos** . 3º. ed. Rio de Janeiro: Editora Elsevier, 2008. 480p.
- AES Eletropaulo**. Disponível em:< <https://www.aeseletropaulo.com.br/educacao-legislacao-seguranca/simuladores/conteudo/calcule-sua-conta>> Acesso em 26 jan 2017.
- ARDUINO, T. Build an electric go-kart on a budget with arduino**. . Disponível em: < <https://blog.arduino.cc/2016/08/15/build-an-electric-go-kart-on-a-budget-with-arduino/> >Acesso em 15 set. 2016

CHIP W5100 SHIELD ETHERNET DATASHEET. Disponível em: <<http://www.wiznet.co.kr/product-item/w5100/>> Acesso em: 23 out. 2016

Classificação dos TIERS - norma EIA/TIA -942. Disponível em: <http://www.tia-942.org/content/162/289/About_Data_Centers> Acesso em: 8 set. 2016

DATASHEET SCT-0133. Disponível em: <<https://nicegear.co.nz/obj/pdf/SCT-013-datasheet.pdf>> Acesso em: 12 set. 2016

LM35 DATASHEET. Disponível em: <<http://www.ti.com/lit/ds/symlink/lm35.pdf>> Acesso em: 12 set. 2016

MILLER, R. **Google: Raise Your *Data center* Temperature.** Disponível em: <<http://www.datacenterknowledge.com/archives/2008/10/14/google-raise-your-data-center-temperature/>>. Acesso em: 08 ago 2016

XAVIER, F. C. An introduction to the industrial applications of microcontrollers. **Physica Scripta**, v. 2002, n. T97, p. 148, 2002. ISSN 1402-4896. Disponível em: <<http://stacks.iop.org/1402-4896/2002/i=T97/a=029>>. Acesso em: 12 nov 2016