

Controle das variáveis ambientais de um data center utilizando softwares e ferramentas livres

Felipe Vitorino de Sousa* Régia Talina Silva Araújo**
José Jhonnathas Aires da Silva Alencar***

* *Departamento da Indústria, Instituto Federal do Ceará, Campus Juazeiro do Norte (e-mail: lima.felipe44@yahoo.com.br).*

** *Departamento da Indústria, Instituto Federal do Ceará, Campus Juazeiro do Norte, (e-mail: regia@ifce.edu.br)*

*** *Departamento de TI, Instituto Federal do Ceará, Campus Juazeiro do Norte, (e-mail: jhonnatas.aires@ifce.edu.br)*

Abstract: The acclimatization of a data center is an important issue in relation to its performance, as it plays a crucial role in the efficiency of servers, data storage systems, network devices and other equipment belonging to the environment. In order to guarantee climatic conditions of operation and security for equipment of a data center, this work consists in the development of a system of remote monitoring and control of temperature and humidity of a data center of an educational institution located in the hinterland of Ceará. The control and monitoring of humidity and temperature variables is performed through the collection of information in an automated way through sensors. The system also aggregates the implementation of Zabbix software for data storage, monitoring, reminders and graph plotting. The Blynk application is also a tool used to control the variables by remote access via smartphone. The system also makes use of an Arduino Integrated Development Environment (IDE), ESP8266 NodeMCU, infrared receiver and infrared emission LED. As a result there is a system implemented and working, through which there is the extraction and analysis of data in a technologically efficient way that allows the view and conditions of operations among the data center established parameters.

Resumo: A climatização de um data center é um quesito importante em relação ao seu desempenho, pois possui papel crucial para eficiência dos servidores, sistemas de armazenamento de dados, dispositivos de redes, dentre outros equipamentos que podem pertencer a este ambiente. A fim de garantir condições climáticas de operação e segurança para equipamentos de um data center, este trabalho consiste no desenvolvimento de um sistema de monitoramento e controle remoto de temperatura e umidade de um data center de uma instituição de ensino localizada no sertão do Ceará. O controle e monitoramento das variáveis de umidade e temperatura é realizado através da coleta de informações de maneira automatizada por meio de sensores. O sistema também agrega a implementação do software Zabbix para armazenamentos de dados, monitoramento, lembretes e plotagem de gráficos. O aplicativo Blynk também é uma ferramenta utilizada para controlar as variáveis por acesso remoto via smartphone. O sistema também faz uso de um Ambiente de Desenvolvimento Integrado (IDE) Arduino, ESP8266 NodeMCU, receptor infravermelho e LED de emissão de infravermelho. Como resultado tem-se um sistema implementado e funcionando, onde tem-se a extração e análise de dados de forma tecnologicamente eficiente que permitem a visão e condições de operações dentre os parâmetros estabelecidos do data center.

Keywords: NodeMCU; Energy efficiency; Internet of Things; Zabbix; Automation.

Palavras-chaves: NodeMCU; Eficiência energética ; Internet das Coisas; Zabbix; Automação.

1. INTRODUÇÃO

As necessidades de fornecimento de serviços de Tecnologia da Informação (TI) requerem gradativamente mais eficiência computacional, caracteristicamente providas por *data centers*. Os *data centers* institucionais são essenciais para a organização de equipamentos de TI. Portanto, torna-se necessário seu monitoramento com o propósito de assegurar o correto funcionamento dos equipamentos contidos no ambiente. Além disso, utilizar e/ou promover mecanismos que forneçam visualizações de informações a respeito do ambiente de TI, podem melhorar a garantia de disponibilidade dos equipamentos da instituição.

O termo *data center* também entendido como Centro de Processamento de Dados, trata-se de um ambiente destinado a alojar e gerenciar equipamentos de processamento, armazenamento e sistemas ativos de redes, tais como roteadores, *Switches*, *Firewall*, *Servidores* entre outros (Hernández et al., 2017).

Segundo Nadjahi et al. (2018), a maior parcela da energia consumida pelos equipamentos de um *data center* é transformada em calor. Sendo assim, a temperatura precisa ser controlada de modo que os equipamentos de TI inseridos nesse ambiente, operem com segurança. Normalmente, essa temperatura é controlada por meio de unidade de ar-condicionado, fixando um valor constante desejado de temperatura, ao invés de seguir a sugestão variável de temperatura sugeridas por normas e padrões nacionais e internacionais. Acarretando assim, gastos de consumo de energia muitas vezes desnecessários.

Conforme NBR 14565:2013 (Associação Brasileira de Normas Técnicas (2013)) e TIA 942-A (Telecommunications Infrastructure Standard for Data Centers (2012)) sugerem manter a temperatura na entrada de ar dos equipamentos críticos de TI entre 18°C e 27°C, e para umidade relativa do ar indicam manter aproximadamente em torno de 60%. Ashrae (2011) definiu uma escala média de temperatura e umidade para ambientes de TI que agregam ou se caracterizam como *data centers* com intuito de estabelecer parâmetros para garantir desempenho e eficiência energética, nos quais podem ser aplicados nas classes de A1 até A4, conforme a tabela 1.

Tabela 1. Recomendações Médias pela Ashrae (2011).

Requisitos	Mínimo	Máximo
Temperatura Bulbo Seco	18°C	27°C
Faixa de variação	0°C/h	5°C/h
Umidade Relativa	20%	80%

Segundo Ashrae (2011), é classificado como A1, ambientes que possuem servidores corporativos e armazenamento que exige a necessidade de parâmetros climáticos rigorosamente controlados. Estes ambientes englobam, geralmente os servidores principais, ou seja, referem-se a um espaço com diversos equipamentos que devem possuir alta tolerância a falhas. Já os demais A2, A3 e A4 competem aos ambientes que agregam servidores de volume, equipamentos de armazenamento, computadores pessoais e estações de trabalho. Podem pertencer à essas classes escritórios ou laboratórios que necessitam de controles de climatização, entretanto não tão rígidos.

Em relação ainda ao consumo de energia, Camargo et al. (2016) afirmam em sua pesquisa que os *data centers* gastam aproximadamente 2% de consumo energético mundial. Ressaltam também que a maior parte dessa energia está concentrada em pequenas e médias empresas que geralmente não têm condições técnicas e financeiras para monitorar as condições climáticas de seus *data centers*. Podendo assim, resultar em gastos excessivos de custos de energia e redução na vida útil de equipamentos de processamento de dados. Os autores, em sua pesquisa sugerem como solução uma implementação de um sistema aberto para monitoramento ambiental de uma sala de servidores.

Corroborando, Ferreira and Godoy (2016) afirmam que a implementação de dispositivos inteligentes aliados aos dispositivos tecnológico para automação predial, podem permitir desenvolvimento econômico com eficiência energética. Já que esses dispositivos, possuem características que conseguem promover controle e gerenciamento de forma remota, por meio de uma rede de comunicação com ou sem fio, facilitando a integração de todos os dispositivos do ambiente.

Outro fator importante no monitoramento remoto de dispositivos e variáveis é o conceito de Internet das coisas (*Internet of the Things* - IoT). Que trata da possibilidade de transformar objetos em dispositivos capazes de se conectarem por meio de chips ou alguma interface de rádio para comunicação, não limitando a conectividade apenas por meio de celulares e computadores. Os dispositivos que agregam o conceito de IoT podem trocar informações entre si, bem como serem monitorados e controlados remotamente. Assim a concepção de IoT, permite conhecer informações vitais sobre mudanças no ambiente, pois os dispositivos que empregam essa tecnologia são capazes de detectar, processar e transmitir dados que podem ser armazenados e analisados (Parida et al., 2019).

Levando em consideração a importância da IoT, Wei et al. (2017) desenvolveram um sistema de monitoramento em tempo real das variáveis de temperatura e umidade de um *data center* com base nessa tecnologia. O projeto também contempla um ambiente virtualizado criado através da modelagem 3D para simular os componentes. Além disso, o sistema emite alertas para os usuários sobre as mudanças das variáveis, bem como dos equipamentos. Oferece também, controle de parâmetros, tais como, controle de fluxo de ar, umidade e temperatura.

Outro trabalho desenvolvido com conceitos semelhantes é do Utomo et al. (2019), que consiste no desenvolvimento de um sistema também para monitoramento das variáveis de um *data center* com o conceito de IoT. Os pesquisadores utilizaram microcontroladores e códigos livres para desenvolvimento. Esse sistema consegue sinalizar para o usuário eventualidades de alterações nas variáveis do ambiente via *Telegram*.

Nota-se então, pela literatura, que com a junção das técnicas de IoT, com *hardwares* e o *softwares open source* é possível desenvolver sistemas eficientes para monitoramento de ambientes de *data centers*, podendo melhorar assim, a eficiência energética onde o sistema for implementado.

Assim, neste artigo será abordado a construção de um sistema para controle e monitoramento das variáveis ambien-

tais de temperatura e umidade de um pequeno *data center* em tempo real de uma Instituição Federal no interior do Ceará. O projeto utiliza microcontroladores e *softwares* livres e/ou *open source*. O ambiente de monitoramento foi desenvolvido através do Zabbix e o controle das variáveis é efetuado pelo App Blynk, juntamente com sensores, NodeMCU e outros componentes eletrônicos.

2. MATERIAIS E MÉTODOS

Nesta seção será descrita a metodologia e os materiais para o desenvolvimento do projeto de monitoramento e controle de temperatura e umidade do pequeno *data center*.

2.1 Materiais

- Dois NodeMCU ESP8266: O principal diferencial é que se trata de um microcontrolador *open-source* de custo acessível que permite conexão *wifi* capaz de integrar o conceito de IoT;
- Sensor de Temperatura e Umidade DHT22, capaz de mensurar temperaturas que variam entre -40°C até 80°C , possui uma precisão aproximadamente de $0,5^{\circ}\text{C}$. Possibilita medir simultaneamente também a umidade relativa de 0% a 100% com precisão de 2%. Realiza as medições a cada 2 segundos;
- LED TSAL6200 é um LED desenvolvido para emissão de sinais infravermelho, ondas de luz capazes de ser detectadas por receptores infravermelhos. Utilizado no projeto para acionar o ar-condicionado;
- Receptor Universal Infravermelho VS1838B utilizado com o propósito de captar sinal infravermelho do controle remoto do ar-condicionado;
- *Jumpers* para fazer as conexões entre os componentes e os microcontroladores;
- *Protoboards* para montagem do protótipo;
- Resistores para delimitar a corrente para acionamento dos componentes;
- Transistor BC373 utilizado na construção de um circuito eletrônico para acionar o LED emissor de infravermelho devido a corrente insuficiente liberada pelos pinos digitais do microcontrolador.
- Bateria Externa recarregável de 10000mAh para fornecer energia para o microcontrolador.

2.2 Métodos

Inicialmente foi realizado a classificação do *data center*, optando por seguir as normas da Ashrae (2011). O pequeno *data center* em estudo contém um *Rack* com os seguintes componentes: seis *switchs*; nove *patch panels*. Além do *Rack* o ambiente possui um computador *desktop* com uma parte do servidor da instituição; um *nobreak* e um roteador (observar a figura 1). O ambiente trata-se de uma parte ramificada do servidor principal, caracterizando-se como um *data center* do tipo A2, segundo Ashrae (2011).

A arquitetura do sistema proposto foi dividida nos seguintes módulos: implementação do servidor; Unidade de Coleta e Unidade de Controle das variáveis de temperatura e umidade. Conforme ilustra a figura 2.

A Unidade de Coleta de temperatura e umidade tem o papel de mensurar as variáveis e transmitir esses dados

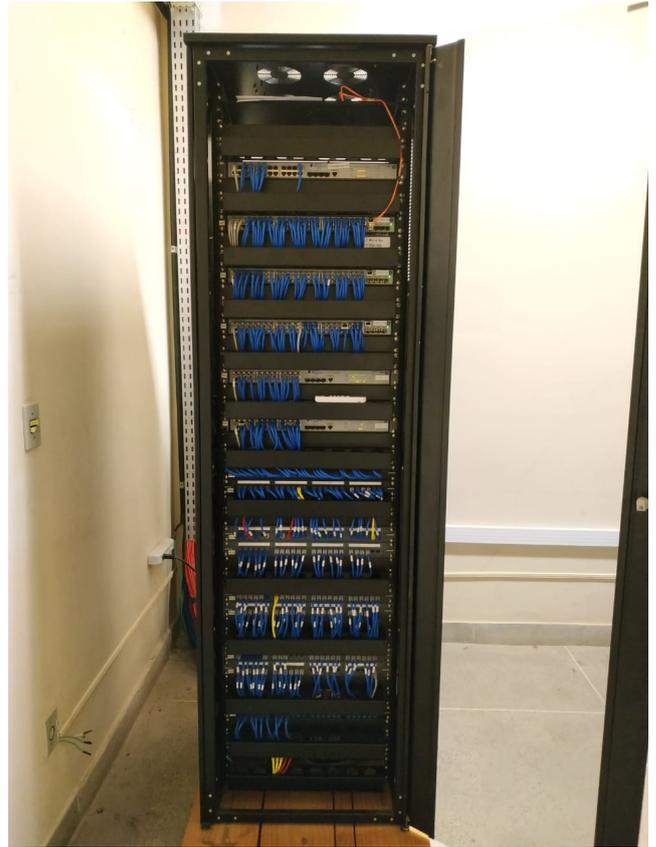


Figura 1. *Rack* para armazenamento dos dispositivos.

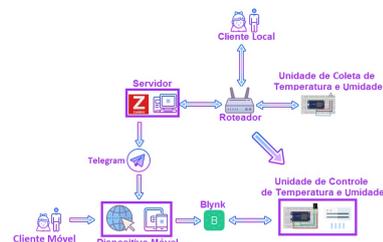


Figura 2. Arquitetura do Projeto.

para o servidor. Já o servidor possui o papel de catalogar os dados recebidos e gerar gráficos, alertas, tela de monitoramento, dentre outras atividades. O cliente móvel recebe notificações do servidor e decidirá sobre o acionamento do sistema da Unidade de Controle ou controlar em uma escala pré-definida a temperatura do sistema.

Na arquitetura do projeto escolhemos definir dois tipos de usuários: Cliente Local e Cliente Móvel. O Cliente Local recebe os gráficos das variáveis coletadas, alertas sonoros e visuais e uma tela de monitoramento. Desta forma ele tem acesso ao monitoramento do sistema, porém tem que estar na rede local do servidor. Já o Cliente Móvel pode estar em qualquer rede, ou seja, interna ou externa, porém com acesso à internet. Esse Cliente recebe apenas notificações do servidor via Telegram sobre o estado do sistema e fará a parte de controle, ou seja, decide sobre o acionamento do sistema ou controla em uma escala pré-definida a temperatura do sistema.

É importante ressaltar que o Cliente Móvel quando recebe notificações de alertas dos eventos no Telegram, recorre ao acesso do sistema da Unidade de Controle de temperatura e umidade. Este sistema é implementado via App Blynk, sendo responsável por controlar os comandos de liga/desliga da central de ar-condicionado e regular a temperatura para 22°C, 24°C ou 26°C. O Blynk informa também se a unidade de controle está *online* para efetuar a ação desejada.

A. Oracle VM VirtualBox

Oracle VM VirtualBox (2019) é um *software* livre de virtualização, com a capacidade de criar máquinas virtuais que possibilita a instalação de diversos sistemas operacionais, no caso deste trabalho o sistema operacional Debian (2019) que também se trata de um *software* livre. A máquina virtual foi utilizada com o propósito de segurança para casos como: necessidades de formatações do computador, transferências ou replicação de toda a instalação e configuração do Zabbix para outro computador.

B. Zabbix

O Zabbix (2019) é uma plataforma *open source* de monitoramento distribuído com gerenciamento centralizado. Oferece uma gama de mecanismos para monitoramento de parâmetros de equipamentos de infraestrutura de TI através do protocolo SNMP ou seus agentes. Através dessa plataforma é possível emitir diferentes tipos de notificações: por *e-mails*, sms, alarmes entre outros (Hernantes et al., 2015).

A instalação do Zabbix foi efetuada por meio de comandos através do terminal do Debian na máquina virtual. Também foram agregados ao Zabbix na máquina virtual o banco de dados MySQL e os agentes do Zabbix, o Zabbix *Sender* e o Zabbix *Agent*. O Banco de dados MySQL tem o intuito de armazenar os dados coletados e transferir os dados para a página *Web* Zabbix, já os agentes efetuam a função de transferir os dados das variáveis temperatura e umidade através do NodeMCU para o Zabbix.

Após a instalação do Zabbix é disponibilizado uma página *web* com um IP definido, como também um usuário e senha padrão (observar figura 3). Porém para que o Zabbix consiga se comunicar com o NodeMCU é preciso integrá-lo com o Zabbix. O *software* oferece interfaces de fácil manuseio, para utiliza-las é necessário atribuir um *token* para o microcontrolador e chaves referentes a cada variável do sensor.

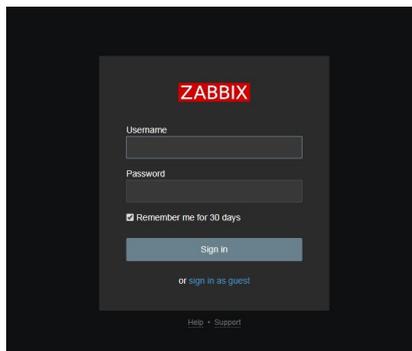


Figura 3. Página web Zabbix.

Com a atribuição do *token* e das chaves é possível ativar recursos correspondentes a cada um na plataforma *Web* Zabbix. Foram cadastradas seis *triggers* no Zabbix, que se referem aos *setpoints* de disparos de informações de temperatura e umidade, bem como para informação de conexão. As *triggers* foram definidas como mostra a tabela 2, conforme a escala média definida por Ashrae (2011).

Tabela 2. *Triggers* criadas.

Nome da <i>Trigger</i>	Severidade	<i>Setpoint</i>
Ping	Informação	Sem resposta por 1 minuto
Temperatura Ideal	Informação	Igual a 24°C
Temperatura Máxima	Alta	Igual a 27°C
Urgente	Desastre	Maior que 30°C
Umidade Mínima	Informação	Igual a 20%
Umidade Máxima	Informação	Igual a 80%

Os valores de 24°C, 27°C, 30°C representam respectivamente, temperatura mínima ideal (24°C); temperatura máxima ideal (27°C) e temperatura urgente que serve em caso de superaquecimento do ambiente (30°C). Para a umidade foram definidos os valores de 20% e 80% para informação da umidade relativa do ambiente mínima e máxima. Por fim, foi criada uma *trigger ping* que sinaliza se a interface gráfica está recebendo os dados do *host* que fornece as informações de temperatura e umidade do ambiente. Todos os valores foram definidos com base na faixa padrão que pode ser implementada para classes A1 até A4 estabelecida pela Ashrae (2011) que consiste em manter o ambiente entre 18°C e 27°C para temperatura e entre 20% e 80% para umidade.

No projeto foram habilitados gráficos para visualizações dos valores das variáveis de temperatura e umidade e desenvolvido um mapa que consiste em cadastrar valores de níveis de temperatura e associá-los a ícones padrões e ícones de sobreposições, como também o estado da conexão entre o microcontrolador e o Zabbix. Quando um valor é atingido o ícone padrão correspondente muda de estado informando ao usuário como se encontra o ambiente através de um ícone de sobreposição como ilustra a figura 4.

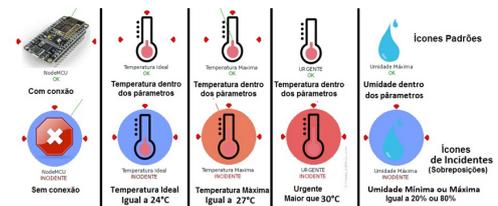


Figura 4. Estados que os ícones podem assumir.

Após todos esses elementos criados, foram anexados em uma única aba denominada como tela de monitoramento (observar a a figura 5). Ainda no Zabbix foram efetuadas outros mecanismos de alertas; implementação de envio de mensagens pelo Telegram, as mensagens informam ao usuário qual valor cadastrado foi atingido; houve também a implementação de um sistema de alerta, quando algum valor é disparado a *interface web* emite um alerta sonoro em bips para o usuário. Uma vez configurado esta plataforma, ela fornece *online* em tempo real o ambiente de monitoramento das variáveis e conexão do microcontrolador.

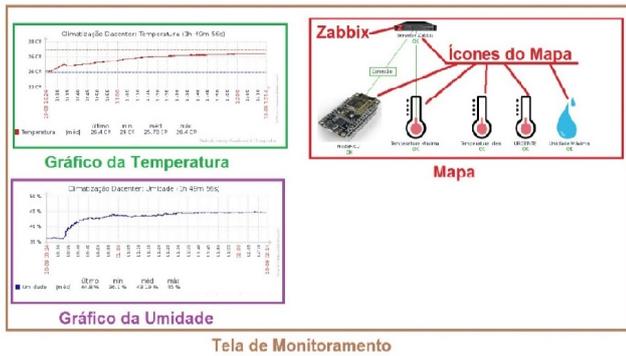


Figura 5. Tela de monitoramento.

C. Unidade de Coleta

O protótipo da Unidade de Coleta de Temperatura e Umidade foi desenvolvido utilizando um circuito que possui um NodeMCU e o sensor DHT22. O sensor coleta as variações de temperatura e umidade, envia esses dados para o NodeMCU ESP 8266 que, por sua vez, transfere para o Zabbix.

A figura 6 ilustra a montagem do circuito. O pino 1 (fio vermelho) é utilizado para receber a alimentação de tensão do NodeMCU, já o pino 2 (fio azul) é o transmissor dos dados coletado pelo sensor que está conectado ao D5 do microcontrolador e o pino Ground (GND - fio preto) está conectado ao pino GND do NodeMCU.

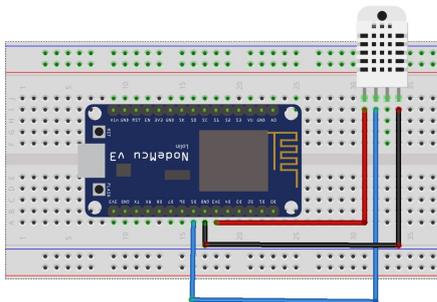


Figura 6. Esquema do circuito do sensor DHT22.

O algoritmo está escrito em C e utiliza a IDE (*Integrated Development Environment*) do Arduino que trata-se de um *software open source* de programação compatível com diversos sistemas operacionais, tais como: Windows, GNU/Linux e MacOS (Arduino, 2019). O algoritmo implementado inclui os arquivos de cabeçalho necessários para integrar o sensor DHT22 com o NodeMCU, bem como um cabeçalho para se conectar à rede *wifi* da instituição, usando o SSID (*Service Set Identificador*) e a senha da rede.

A implementação do algoritmo proposto nesse projeto, inclui também um cabeçalho necessário para interagir com o Zabbix através do Agente Zabbix *Sender*. A IDE permite escrever o código em linguagem C, compilar o algoritmo e fazer o *upload* para a memória *flash* do NodeMCU. Os arquivos da biblioteca do DHT22 necessários para funcionamento dos dispositivos são extraídos para a pasta correspondente aos arquivos das bibliotecas da IDE do Arduino. Por fim, o Zabbix e o microcontrolador se comunicam através do formato JSON.

D. Telegram

O Telegram (2019) é um serviço de troca de mensagens instantâneas baseado em nuvem com sincronização perfeita; portanto podendo ser acessado através de vários dispositivos ao mesmo tempo, tais como, *smartphones*, *tablets* e aplicações *web* e conta com segurança na troca de informações. O Telegram também oferece uma API aberta para desenvolvedores construírem *bots* e integrarem seus serviços (Jalilvand and Neshati, 2020). Assim, a integração do Telegram com o Zabbix foi possível.

Para tanto, a integração envolveu os seguintes passos:

- (1) Criação de um novo *bot* no App Telegram usando o *botfather*.
- (2) Nomear o *bot*.
- (3) Instalar a API do Telegram no diretório do Zabbix na máquina virtual.
- (4) Criar um *script* de alerta com o ID do *Bot*.
- (5) Cadastrar o *script* na *Interface Web* do Zabbix.

Após esse passo, o *bot* passa a atuar como um funcionário virtual recebendo informações do Zabbix e alertando através de mensagens o estado das variáveis no ambiente do *data center*.

E. Unidade de Controle de Temperatura e Umidade

O algoritmo da Unidade de Controle de Temperatura e Umidade também foi desenvolvido via IDE Arduino. Para se efetuar a programação, foi instalada a biblioteca IRremote. Esta biblioteca permite fazer a emissão ou recepção de sinais infravermelhos através de emissores e/ou receptores infravermelhos.

O circuito de clonagem dos sinais infravermelhos é composto por um NodeMCU e um receptor infravermelho. A conexão foi efetuada da seguinte maneira: Pino de Saída (fio azul) do receptor foi conectado ao pino D5 do NodeMCU, o GND (fio verde) ao pino GND do microcontrolador, e o pino de Tensão de alimentação positiva (VCC) (fio vermelho) ao pino 3V3 do NodeMCU. A figura 7 ilustra o esquema da montagem do circuito de clonagem dos sinais infravermelhos.

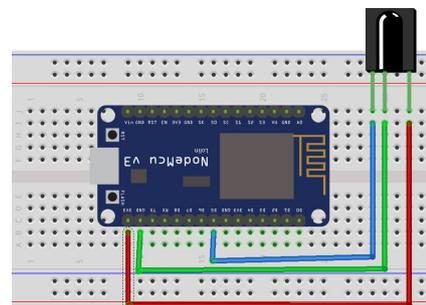


Figura 7. Esquema do circuito de clonagem dos sinais infravermelhos.

Após aplicada a programação no NodeMCU, foi apontado o controle remoto para o receptor infravermelho para clonar os sinais dos botões desejados. Ao pressionar alguma tecla do controle o receptor captura um código em valor hexadecimal e respectivamente seu sinal de frequência e

exibe na serial da IDE do Arduino. Os botões clonados foram: liga/desliga e de alterar o valor da temperatura.

A seguir podemos ver o exemplo de um código gerado na serial da IDE do Arduino por uma das teclas pressionadas e capturado a frequência do sinal infravermelho.

```
●C1600046;
●Raw (100): 8950 -4700 600 -1700 600 -1700 600 -600 600
-600 600 -600 600-600 600 -600 600 -1700 600 -600 600 -
1700 600 -1700 600 -600 600 -600 600-600 600 600 -600 600 -600
600 -600 600 -600 600 -600 600 -550 600 -600 600 -600600
-600 600 -600 600 -600 600 -1700 600 -600 600 -600 600
-600 600 -1700600 -1700 600 -600 600 -600 600 -600 600
-600 600 -600 600 -600 600 -600600 -600 600 -600 600 -600
600 -600 600 -600 600 -600 600 -600 600 -600 600-600 600
-600 600
```

O botão liga/desliga foi clonado para que o ar-condicionado ao ser ligado inicie a refrigeração do ambiente em 22°C. Já o comando de controle de temperatura permite aplicar 24°C ou 26°C.

Após guardar os códigos dos comandos desejados foi elaborado o circuito de emissão dos sinais infravermelhos, que são enviados para o receptor do ar-condicionado. Devido a baixa corrente fornecida pelo microcontrolador, o emissor de infravermelho não conseguia operar, pois o mesmo necessita de uma corrente de 100mA.

Para solucionar o problema foi inserido no circuito de emissão dos sinais infravermelhos, um amplificador de corrente. Utilizando para tanto, um transistor BC337 de atua como uma chave no circuito e um resistor de 10k ohms na base do transistor, como ilustra a figura 8. O transistor BC337 consegue gerar um ganho de corrente de acordo com a sua faixa de hFE. A faixa de hFE do componente utilizado varia entre 160 e 400 hFE.

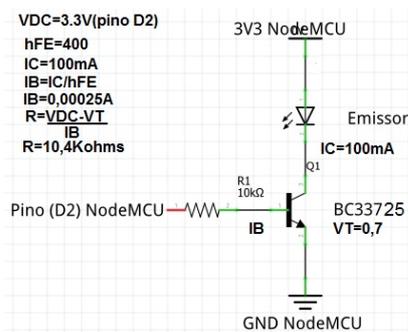


Figura 8. Esquema do circuito de emissão dos sinais infravermelhos.

F. Blynk

O Blynk (2019) é um aplicativo para *smatphones* personalizável que permite controlar remotamente um *hardware* programável, assim como reportar dados do *hardware* ao aplicativo. Com ele, é possível desenvolver interfaces gráficas de controle e utilizar o conceito internet das coisas (Waworundeng et al., 2019). Assim, neste projeto o Blynk foi utilizado para construir a interface do usuário com a Unidade de Controle de Temperatura e Umidade de forma rápida e intuitiva.

Com o App instalado, foi efetuado um cadastro, inserindo um e-mail e senha. Ao criar um projeto no aplicativo é recebido um *auth token* no e-mail. O *auth token* é aplicado na programação do microcontrolador e funciona como uma chave que sincroniza o App instalado no *smartphone* com a Unidade de Controle de Temperatura e Umidade.

Após a liberação do *auth token*, foi desenvolvido o ambiente de acionamento do ar condicionado na interface personalizável do Blynk. No qual, consiste em associar os pinos virtuais do App aos pinos físicos do microcontrolador através de *widgets*, em que resultou na interface representada na figura 9. Os *widgets* v1, v2 e v3 representam respectivamente: Ligar/Desligar o ar-condicionado; acionar a temperatura em 24°C e acionar a temperatura em 26°C.



Figura 9. Interface de controle criada no Blynk.

A programação foi efetuada também via IDE Arduino incluindo os cabeçalhos de conexão *wifi* e funcionamentos do emissor e receptor de sinais infravermelhos; como também o cabeçalho para sincronização com o App Blynk.

O funcionamento da programação dessa parte do projeto, consiste na implementação do algoritmo responsável por enviar os códigos clonados para o banco de dados do Blynk. Em seguida, ocorre um pareamento entre o NodeMCU e o App. Quando pareado o Blynk fica aguardando o usuário efetuar uma ação, ao ser efetuada o microcontrolador envia o sinal correspondente do botão virtual pressionado através do emissor de infravermelho. Desta maneira o receptor do ar-condicionado captura o código recebido da Unidade de Controle e efetua o comando que lhe foi entregue.

3. RESULTADOS E DISCUSSÃO

Propusemos a execução de um projeto utilizando um microcontrolador de baixo custo com ferramentas livres e/ou *open source* com aplicação IoT. Possibilitando o monitoramento em tempo real e o controle de maneira remota do *data center*, por meio da internet, desta forma conseguindo manipular o sistema térmico do ambiente estabelecendo o controle das variáveis de temperatura e umidade dentro de uma escala estabelecida, conforme os valores estabelecidos por Ashrae (2011).

O projeto proposto neste artigo, após implementado, foi colocado para execução nos dias, 12 e 14 de agosto de 2019. Primeiramente a ideia era gerar uma interface para monitoramento, depois implementamos a Unidade de Controle. Entretanto, alguns dias foram necessários para fazer ajustes de posicionamento do emissor infravermelho, bem como a checagem do funcionamento dos alertas recebidos pelo Telegram. O sensor da Unidade de Coleta fica próximo a entrada de ar do *rack*.

Para que a Unidade de Controle de temperatura e umidade funcionasse com precisão, foi diagnosticado que o emissor deveria estar próximo ao receptor do ar-condicionado por volta de 10 cm. Após averiguado a posição correta do emissor (observar a figura 10) a unidade de controle passou a efetuar os comandos de forma que fossem recebidos pelo ar-condicionado.



Figura 10. Posicionamento da Unidade de Controle de Temperatura e Umidade.

Ainda analisando a implementação do projeto, no primeiro dia de teste concluímos que o ar-condicionado permaneceu desligado por cerca de três horas (das 14:58h às 18:06h) como podemos observar nos gráficos gerados na figura 11. Para o segundo dia ao analisarmos os gráficos presentes na figura 12 o sistema conseguiu eficiência econômica por cerca de seis horas dentro dos padrões estabelecidos para este projeto entre 24°C a 27°C com o ar-condicionado desligado. Em nenhum momento o ambiente atingiu os valores definidos para mínimo ou máximo da umidade.

Com base na implementação do projeto, considerando-se 48(h) de uso ininterrupto da central de ar, como operava anteriormente a este trabalho desenvolvido. Analisamos que se resultou aproximadamente em 18,75% de economia energética, que se atribui as 9h do ar-condicionado desligado após a execução deste projeto.

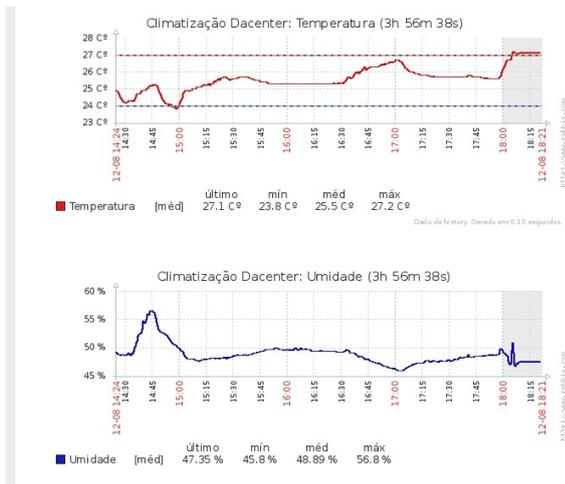


Figura 11. Gráficos de Temperatura e Umidade do dia 12.

Ainda no projeto, o servidor conseguiu notificar ao usuário do cliente móvel todos os estados das temperaturas do ambiente e conexão do sistema desenvolvido. A figura 13 refere-se as notificações do primeiro dia, enquanto a figura 14 ao segundo dia.

A unidade de coleta e controle reagiu como de maneira esperada em relação as suas atribuições definidas, tornando

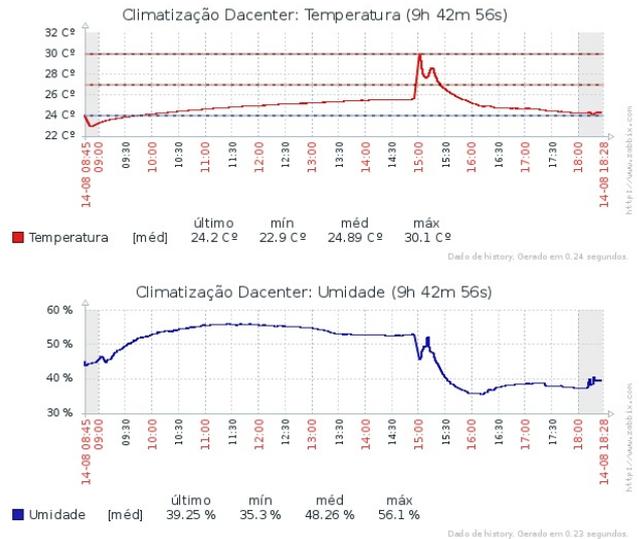


Figura 12. Gráficos de Temperatura e Umidade do dia 14.

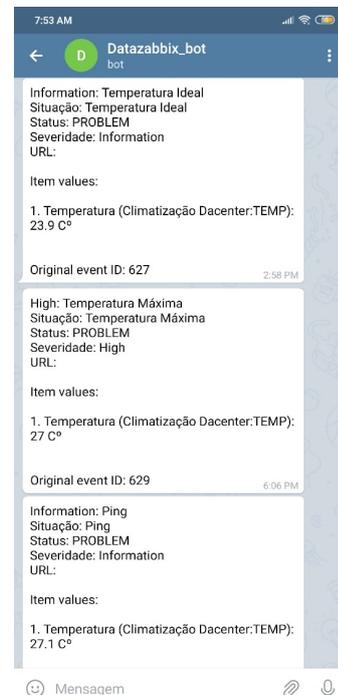


Figura 13. Notificações recebidas no Telegram no primeiro teste.

um ponto significativo para este trabalho. Com base nisto, o produto deste projeto desenvolvido pode ser incorporado às ações de eficiência energética do Campus.

O projeto consegue coletar os dados de temperatura de umidade do ambiente que são exibidos em seus respectivos gráficos. A interface web do Zabbix conseguiu conceber um ambiente de monitoramento ao usuário em tempo real. O sistema proporciona alertas sonoras, bem como a emissão de notificações ao Telegram quando atinge os valores definidos. Diante as notificações e alertas o usuário do cliente local ou do cliente móvel consegue tomar as decisões para ligar/desligar a central de ar-condicionado.



Figura 14. Notificações recebidas no Telegram no segundo teste.

AGRADECIMENTOS

Ao Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Campus Juazeiro do Norte, em especial ao departamento de Tecnologia da Informação, por permitir a pesquisa, análise e implementação do projeto.

REFERÊNCIAS

- Arduino (2019). Arduino website. <https://www.arduino.cc/en/Main/Software>. Accessed: 2019-09-20.
- Ashrae, T. (2011). 9.9 (2011) thermal guidelines for data processing environments—expanded data center classes and usage guidance. *Whitepaper prepared by ASHRAE technical committee (TC)*, 9.
- Associação Brasileira de Normas Técnicas (2013). NBR 14565:2013, Cabeamento estruturado para edifícios comerciais e datacenters.
- BC373 (2007). Datasheet do transistor BC373. <https://www.onsemi.com/pub/Collateral/BC372-D.PDF>. Accessed: 2019-10-15.
- Blynk (2019). Blynk website. <https://blynk.io/>. Accessed: 2019-10-03.
- Camargo, D.S., Miers, C.C., Koslovski, G.P., and Pillon, M.A. (2016). Greenhop: Open source environmental monitoring for small and medium data centers. In *2016 35th International Conference of the Chilean Computer Science Society (SCCC)*, 1–12.
- Debian (2019). Debian website. <https://www.debian.org/index.pt.html>. Accessed: 2019-09-02.
- DHT22 (s.d.). Datasheet do sensor de temperatura e umidade DHT22. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. Accessed: 2019-09-15.
- Ferreira, I.V. and Godoy, E.P. (2016). Integração de internet das coisas e zigbee no contexto de eficiência energética e automação predial. *XXI Congresso Brasileiro de Automática - CBA2016*, 1392–1397.
- Hernantes, J., Gallardo, G., and Serrano, N. (2015). It infrastructure-monitoring tools. *IEEE Software*, 32(4), 88–93.
- Hernández, L., Calderon, Y., Martinez, H., Pranolo, A., and Riyanto, I. (2017). Design of a system for detection of environmental variables applied in data centers. In *2017 3rd International Conference on Science in Information Technology (ICSITech)*, 389–395.
- Jalilvand, A. and Neshati, M. (2020). Channel retrieval: finding relevant broadcasters on telegram. *Social Network Analysis and Mining*, 10, 1–16.
- Nadjahi, C., Louahlia, H., and Lemasson, S. (2018). A review of thermal management and innovative cooling strategies for data center. *Sustainable Computing: Informatics and Systems*, 19, 14 – 28. doi:<https://doi.org/10.1016/j.suscom.2018.05.002>. URL <http://www.sciencedirect.com/science/article/pii/S2210537917304067>.
- NodeMCU (s.d.). Datasheet NodeMCU. https://www.handsontec.com/pdf_learn/esp8266-V10.pdf. Accessed: 2019-09-02.
- Oracle VM VirtualBox (2019). Oracle vm virtualbox website. <https://www.virtualbox.org/>. Accessed: 2019-09-02.
- Parida, D., Behera, A., Naik, J.K., Pattanaik, S., and Nanda, R.S. (2019). Real-time environment monitoring system using esp8266 and thingspeak on internet of things platform. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 225–229.
- Telecommunications Infrastructure Standard for Data Centers (2012). TIA 942-A Telecommunications industry association standards and engineering publications.
- Telegram (2019). Telegram website. <https://telegram.org/>. Accessed: 2019-10-20.
- TSAL6200 (2014). Datasheet do diodo emissor de infravermelho TSAL6200. <https://www.vishay.com/docs/81010/tsal6200.pdf>. Accessed: 2019-10-07.
- Utomo, M.A.P., Aziz, A., Winarno, and Harjito, B. (2019). Server room temperature & humidity monitoring based on internet of thing (IoT). *Journal of Physics: Conference Series*, 1306, 012030. doi:10.1088/1742-6596/1306/1/012030. URL <https://doi.org/10.1088/1742-6596/1306/1/012030>.
- VS1838B (s.d.). Datasheet do receptor de infravermelho VS1838B. <https://www.elecrow.com/download/Infrared%20receiver%20vs1838b.pdf>. Accessed: 2019-10-07.
- Waworundeng, J.M.S., Fernando Tiwow, D., and Tulangi, L.M. (2019). Air pressure detection system on motorized vehicle tires based on iot platform. In *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, volume 1, 251–256.
- Wei, W., Chongchao, P., Yikai, L., and Gang, L. (2017). The research and application of green computer room environmental monitoring system based on internet of things technology. In *IOP Conference Series: Earth and Environmental Science*, volume 94 of *IOP Conference Series: Earth and Environmental Science*, 012171. doi:10.1088/1755-1315/94/1/012171.
- Zabbix (2019). Zabbix website. <https://www.zabbix.com/>. Accessed: 2019-09-02.