

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Monitoramento contínuo de temperatura utilizando Raspberry Pi e Zabbix

Autor: Richard da Silva Francisco de Souza
Orientador: Prof. Dr. Sandro Augusto Pavlik Haddad

Brasília, DF
2022



Richard da Silva Francisco de Souza

Monitoramento contínuo de temperatura utilizando Raspberry Pi e Zabbix

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Sandro Augusto Pavlik Haddad

Brasília, DF

2022

Richard da Silva Francisco de Souza

Monitoramento contínuo de temperatura utilizando Raspberry Pi e Zabbix/
Richard da Silva Francisco de Souza. – Brasília, DF, 2022-
52 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Sandro Augusto Pavlik Haddad

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.

1. Monitoramento. 2. Raspberry Pi. I. Prof. Dr. Sandro Augusto Pavlik Haddad. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Monitoramento contínuo de temperatura utilizando Raspberry Pi e Zabbix

CDU

Richard da Silva Francisco de Souza

Monitoramento contínuo de temperatura utilizando Raspberry Pi e Zabbix

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Trabalho aprovado. Brasília, DF, :

**Prof. Dr. Sandro Augusto Pavlik
Haddad**
Orientador

Prof. Dr. Alex Reis
Convidado 1

**Prof. Dr. Gerardo Antonio Idrobo
Pizo**
Convidado 2

Brasília, DF
2022

Agradecimentos

Agradeço, primeiramente, a Deus por ter me dado suporte emocional em todos os momentos difíceis e forças para continuar batalhando dia após dia.

Quero agradecer aos meus pais e meu irmão, apesar da distância, por todo o incentivo, apoio, compreensão e ajuda durante esses anos.

Agradeço aos meus amigos e companheiros de curso, em especial, à Camilla Alves, por todo o ensinamento, apoio durante os semestres e pela amizade e entusiasmo que se fortaleceu a cada dificuldade passada.

Agradeço ao meu orientador, Professor Sandro, o qual tenho grande admiração e respeito, por ter me guiado por este projeto e por todo o suporte, dedicação e paciência durante o processo de desenvolvimento do presente trabalho.

Por fim, agradeço à Universidade de Brasília por todo o aprendizado nesses anos de curso.

Resumo

A confiabilidade do funcionamento dos servidores dependem, em sua forma mais básica, da garantia de um sistema de alimentação elétrica e do ambiente termicamente apropriado. Levando-se em consideração a confiabilidade da rede elétrica com baterias e geradores, o presente trabalho pretende discutir o uso, na sala de servidores, do monitoramento contínuo da temperatura utilizando a Ferramenta Zabbix e a Raspberry Pi para a aquisição e armazenamento dos dados dos sensores de temperatura. O objetivo é projetar um sistema em tempo real capaz de coletar e armazenar dados, realizar o monitoramento e enviá-los a um cliente na internet. De modo que, ao coletar os dados, haja um histórico confiável do monitoramento da temperatura com o propósito de garantir que os equipamentos, contidos na sala de servidores, estejam com temperatura ideal de funcionamento.

É utilizada uma metodologia Top-Down onde, a partir de uma visão geral do sistema, obtém-se uma fragmentação do projeto para compreensão de todos os subsistemas.

Palavras-chave: Monitoramento. Raspberry Pi. Temperatura. Zabbix.

Abstract

The present work intends to discuss the use, in the data center, of a continuous temperature monitoring using the Zabbix Tool and the Raspberry Pi for the acquisition and storage of data from temperature sensors. The objective of this study is to design a real-time system capable of collecting and storing data, monitoring and sending them to a client online. Therefore, once collecting the data, there will be a reliable temperature monitoring history in order to ensure that the equipment in the data center is working at an ideal operating temperature.

Considering an overview approach of the system, a Top-Down methodology was chosen in which a fragmentation of the project is achieved to elucidate all the subsystems.

Key-words: Monitoring. Raspberry Pi. Temperature. Zabbix.

Lista de ilustrações

Figura 1 – Projeções para o impacto da IoT sobre o PIB mundial até 2020 (em trilhões de dólares) (DA; DE, 2020)	15
Figura 2 – Sensor de Temperatura DHT11 (SARAVATI, 2021)	19
Figura 3 – Sensor de Temperatura DHT22 (DISTRELEC, 2021)	19
Figura 4 – Pinagem do Sensor de Temperatura DHT11 (ROBOTICMENTE.BLOGSPOT, 2021)	19
Figura 5 – Conexão entre Sensor DHT11 e Raspberry (GAY, 2018)	20
Figura 6 – Adaptador Sensor DHT11 ESP01 (USINAINFO, 2022)	20
Figura 7 – Funcionamento da aquisição do sinal da Raspberry pelo sensor (EBRAHIMI-DARKHANEH, 2010)	21
Figura 8 – Envio de sinal para o sensor DHT11 (EBRAHIMI-DARKHANEH, 2010)	22
Figura 9 – Resposta do sensor à Raspberry (EBRAHIMI-DARKHANEH, 2010)	22
Figura 10 – Envio de DADOS do Sensor DHT11 (GAY, 2018)	23
Figura 11 – Formato de envio de dados do Sensor DHT11 (GAY, 2018)	23
Figura 12 – Diagrama de blocos do fluxo de trabalho	27
Figura 13 – Planta baixa da Sala de Servidores (ENGENHARIA, 2020)	30
Figura 14 – Gráfico de Fluxo de Temperaturas (ENGENHARIA, 2020)	31
Figura 15 – Módulo microcontrolador ESP01 (EMBARCADOS, 2022)	32
Figura 16 – Pinagem do Microcontrolador ESP01 (FALCÃO; WONZOSKI, 2019)	33
Figura 17 – Raspberry Pi 3B+ (FOUNDATION, 2021)	34
Figura 18 – Tela de autenticação do Zabbix (ZABBIX, 2021)	37
Figura 19 – Tela <i>Dashboard</i> Zabbix	38
Figura 20 – Teste dos Sensores de Temperatura	38
Figura 21 – Informações do sistema no Zabbix	39
Figura 22 – Tela de criação de Itens	39
Figura 23 – Configuração da <i>Trigger</i> do Sensor 1	40
Figura 24 – E-mail de Incidente	41
Figura 25 – E-mail de Incidente Resolvido	41
Figura 26 – Layout Projeto	42
Figura 27 – Módulo Sensor DHT11 ligado à ESP01	43
Figura 28 – Módulo Sensor DHT11/ESP01 e regulador de tensão	43
Figura 29 – Módulo Sensor DHT11/ESP01 instalado no Servidor	44
Figura 30 – Dashboard Zabbix	45
Figura 31 – Triggers Módulo Sensor DHT11/ESP01	46
Figura 32 – Alertas no Dashboard	47
Figura 33 – E-mail dos Alertas	47

Figura 34 – E-mail do Alerta resolvido	48
Figura 35 – Histórico de Temperatura	48

Lista de tabelas

Tabela 1 – Especificações dos Sensores DHT11 e DHT22 (EBRAHIMI-DARKHANEH, 2010) (CO., 2010)	32
Tabela 2 – Comparação entre Modelos da Raspberry Pi (HUTCHINSON, 2021).	33
Tabela 3 – Tabela Comparativa de Softwares	35
Tabela 4 – Requisitos mínimos de Hardware	37

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas
CFD	Simulação de Dinâmica dos Fluidos
GNL	General Public license
GPIO	General Purpose Input/Output
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
NMS	Network Management System
PHP	Hypertext Preprocessor
PIB	Produto Interno Bruto
RAM	Random Access Memory
RRDTOOL	Sistema de Base de Dados Round-Robin
SD	Secure Digital Card
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SO	Sistema Operacional
SoC	System on a Chip
TCC	Trabalho de Conclusão de Curso
TCP	Protocolo de Controle de Transmissão
UDP	Protocolo de Datagrama do Usuário
WEB	World wide Web
Wifi	Wireless Fidelity

Lista de símbolos

A	Ampére
°C	Graus Celsius
mA	Miliampere
MB	Megabyte
ms	Milissegundo
MHz	Megahertz
Ω	Ohm
s	Segundos
us	Microssegundo
V	Volts

Sumário

1	INTRODUÇÃO	14
1.1	Objetivo Geral	16
1.2	Objetivos Específicos	16
1.3	Estrutura do Trabalho	16
2	REFERENCIAL TEÓRICO	18
2.1	Sala de Servidores	18
2.2	Sensores de Temperatura	18
2.2.1	Sensor DHT11	19
2.2.2	Funcionamento do Sensor DHT11	21
2.2.2.1	Funcionamento do envio de DADOS Sensor-Raspberry	23
2.3	Raspberry Pi	23
2.4	Ferramentas <i>open source</i> de monitoramento	24
2.4.1	Cacti	24
2.4.2	Nagios	25
2.4.3	Zabbix	25
3	METODOLOGIA	27
3.1	Metodologia Top-Down	27
3.2	Bloco I - Referência Bibliográfica	28
3.3	Bloco II - Definição dos Sistemas: Escolha de fontes e formas de coleta de dados	28
3.4	Bloco III - Configuração Raspberry e Zabbix	28
3.5	Bloco IV - Simulações e Testes	28
3.6	Bloco V - Análise e Discussão dos Resultados	29
4	DESENVOLVIMENTO	30
4.1	Sala de Servidores do CINDACTA I	30
4.2	Sensor de Temperatura	31
4.3	Microcontrolador ESP01	32
4.4	Raspberry Pi	33
4.4.1	Sistema Operacional da Raspberry Pi	34
4.5	Software <i>Open Source</i> de Monitoramento de Ativos de Rede	34
4.6	Zabbix	35
4.6.0.1	Pré-requisitos de sistema para instalação do Zabbix	36
4.6.0.2	Interface Web do Zabbix - <i>Front-end</i>	37

5	RESULTADOS E DISCUSSÕES	42
5.1	Cenário do Experimento	42
5.1.1	Montagem dos Sensores	43
5.1.2	Instalação do Módulo Sensor DHT11/ESP01 nos Servidores	44
5.2	Execução e Monitoramento	45
5.2.1	Alertas e Alarmes	46
6	CONCLUSÃO	49
7	TRABALHOS FUTUROS	50
	REFERÊNCIAS	51

1 Introdução

A alta temperatura é um dos fatores mais críticos que afetam diretamente a confiabilidade dos sistemas de computador. Apesar de ser comum a temperatura normal de funcionamento de um sistema ser elevada, há de se considerar que, para sistemas mais sensíveis, um aumento de poucos graus pode ocasionar problemas inexplicáveis. Com o aumento da temperatura, os atrasos de comunicação e os defeitos nos circuitos também aumentam.

Além disso, o ambiente que contém estes sistemas é mantido em temperatura mínima para evitar sobrecarga na temperatura dos processadores, a ocorrência de resultados operacionais indesejados e a redução da vida útil dos equipamentos de TI (ZHANG Y.; FUJITA, 2015).

Segundo (WEISER, 1993), as tecnologias mais profundas são as que desaparecem pois tornam-se parte do dia a dia das pessoas. Ou seja, a mudança no cotidiano da humanidade se adapta às tecnologias do século e o desenvolvimento da Internet das Coisas, *Internet of Things (IoT)*.

Ademais, é impulsionado pelas necessidades de grandes corporações nas quais se beneficiam da previsibilidade, capacidade de codificar e rastrear objetos, agilidade nos processos e redução de custos. Por fim, a IoT é uma ferramenta que, junto com a revolução tecnológica que representa o futuro da computação e das comunicações, faz a integração de diversos sistemas gerando uma grande rede de telecomunicações e de informações.

A Internet das Coisas pode ser definida, de forma geral, como uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se auto-organizar, compartilhar informações, dados e recursos. Isto é, a IoT reage diante de situações e mudanças no ambiente em uma infraestrutura global de objetos físicos em rede, permitindo conectividade a qualquer hora e em qualquer lugar.

Segundo o Banco Nacional de desenvolvimento Econômico e Social (2017), a IoT é posicionada como uma das maiores tendências tecnológicas do setor de Tecnologias da Informação e Comunicação. No Brasil, o setor de Tecnologias da Informação e Comunicação movimentou 15 bilhões de dólares em 2014, o que representou cerca de 9,1% do PIB nacional.

De acordo com estudos econômicos da *Accenture* (2015), em parceria com a *Frontier Economics*, há estimativa de 75% do Produto Interno Bruto (PIB) mundial para o impacto da IoT na economia. A figura 1, apresenta algumas estimativas do impacto da IoT na economia mundial.

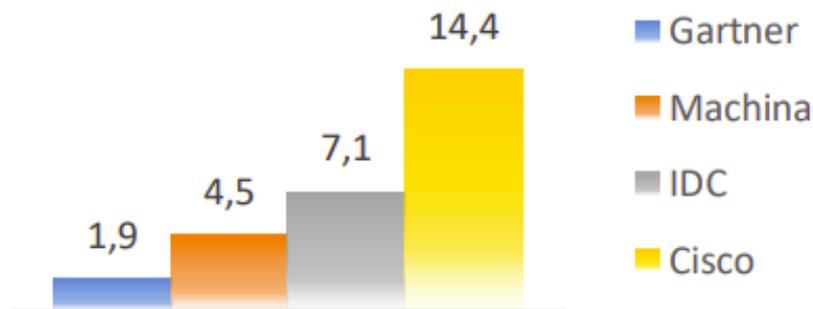


Figura 1 – Projeções para o impacto da IoT sobre o PIB mundial até 2020 (em trilhões de dólares) (DA; DE, 2020)

Vendo a importância da Internet das Coisas no cotidiano e como forma de agregar soluções inteligentes e de baixo custo, é que nasce o referido projeto de conclusão de curso.

Em várias empresas, seja ela de média ou grande porte, há um local reservado para os servidores que são máquinas dedicadas com um ou vários processadores, bancos de memórias, portas de comunicação, softwares e sistema de armazenamento de dados chamados *storages*. Os servidores são responsáveis por executar um conjunto específico de programas e protocolos de comunicação com a finalidade de fornecer serviços para outras máquinas ou clientes.

Posto isto, é de importância ímpar para a empresa que cada dispositivo ali presente esteja funcionando continuamente. Por serem máquinas dedicadas com poder de processamento elevado, torna-se imprescindível que haja um monitoramento contínuo do ambiente das salas de servidores com a finalidade de verificar se os dispositivos ali presentes estão operando na faixa de temperatura indicada pelo fabricante. A operação dos servidores fora da faixa de temperatura pode acarretar degradação e danos imediatos aos equipamentos. Além disso, pode ocorrer o desligamento automático como parte do procedimento das máquinas com a finalidade de não corromper os processos e arquivos necessários para o funcionamento.

Diante de tal contexto, esta monografia tem como principal objetivo mensurar a temperatura da sala de servidores em tempo real e verificar se está adequada às normas de bom funcionamento. Para isso, utilizar-se-á a Raspberry Pi, que é uma tecnologia que se conecta facilmente à Internet e não precisa de módulos para tal, podendo ser programada usando uma variedade de linguagem de programação e sensores de temperatura. Com isso, será possível buscar análises do monitoramento contínuo e direto sem pausas.

Neste projeto, será investigado a execução do Zabbix, que é uma ferramenta de software de monitoramento de código aberto para diversos componentes de TI, incluindo redes, servidores, máquinas virtuais e serviços em nuvem e que será responsável por realizar o monitoramento da temperatura.

Também está incluso no projeto criar um sistema que interprete as informações coletadas e notifique ao usuário caso seja encontrado alguma falha, irregularidade ou baixo desempenho. Desta forma, os problemas detectados poderão ser solucionados de forma muito mais rápida e eficiente.

1.1 Objetivo Geral

O objetivo geral dessa dissertação é projetar um sistema capaz de armazenar dados, realizar o monitoramento e enviá-los a um cliente na internet. De modo que, ao coletar os dados, haja um histórico confiável do monitoramento da temperatura com o propósito de garantir que os equipamentos, contidos na sala de servidores, estejam com temperatura ideal de funcionamento.

1.2 Objetivos Específicos

- Preparação da Raspberry Pi com o Sistema Operacional Raspbian;
- Instalação da versão gerente (*Server*) e agente (*agent*) da ferramenta Zabbix;
- Utilizar o Sensor de Temperatura e umidade (DHT11) para mensurar a temperatura na sala de servidores;
- Configurar o software Zabbix para monitorar os dados de temperatura obtidos pela Raspberry Pi; e
- Integrar os sensores DHT11 com o microcontrolador ESP01 e realizar a aquisição de dados por meio do módulo *Wifi* da Raspberry.

1.3 Estrutura do Trabalho

O estabelecimento dos objetivos propostos é determinado em cinco capítulos.

Capítulo 1: introdução do tema proposto abordando a contextualização do problema, objetivos propostos, objetivos específicos sobre o tema, a motivação, o objetivo geral, os objetivos específicos e a estrutura do trabalho.

Capítulo 2: aborda a fundamentação teórica do trabalho utilizando o referencial teórico com uma revisão bibliográfica. Além disso, é abordado, também, o desenvolvimento da ferramenta Zabbix com a instalação do servidor e hospedeiro na Raspberry Pi. Ademais, é aludido o funcionamento dos sensores DHT11 utilizados no Projeto.

Capítulo 3: desenvolve a metodologia e a topologia utilizadas no trabalho. Além disso, descreve-se o processo e as técnicas utilizadas para chegar aos resultados dos objetivos propostos.

Capítulo 4: é realizado um ambiente de simulação para instalação e configuração dos componentes utilizados nesse projeto.

Capítulo 5: esse capítulo apresenta os resultados obtidos com o desenvolvimento do trabalho. Com isso, são demonstradas as configurações utilizadas na ferramenta Zabbix e os resultados esperados e alcançados.

Capítulo 6: refere-se à conclusão do projeto.

Por fim, o **Capítulo 7** relaciona-se às propostas e atualizações para trabalhos futuros.

2 Referencial Teórico

Para se obter uma boa base dos conceitos teóricos e dos métodos que podem ser utilizados na construção desse Projeto, faz-se necessário um levantamento teórico sólido sobre os conceitos, componentes e métodos utilizados. Parte fundamental para identificar divergências passíveis de mudanças.

2.1 Sala de Servidores

A sala de servidores são instalações que contêm *racks* com computadores dedicados, de alta performance e operação ininterrupta. Além disso, estes equipamentos geram muito calor em um curto intervalo de tempo caso o sistema de refrigeração não funcione corretamente. O aquecimento acima do que é estipulado no manual do fabricante pode ocasionar o desligamento automático do computador, com a finalidade de manter a integridade dos arquivos processados, diminuição da vida útil do equipamento e o desempenho das aplicações. A cada minuto de inatividade destes computadores pode acarretar perda de milhares de dólares para algumas empresas ou, até mesmo, falha na operação de sistemas primordiais para a vida das pessoas como, por exemplo, os de tráfego aéreo.

2.2 Sensores de Temperatura

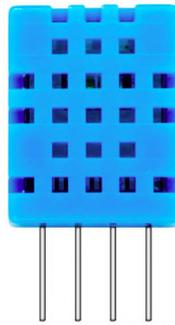
Temperatura é um parâmetro importantíssimo em vários setores diferentes como medicamentos, alimentos, semicondutores e em ambientes com máquinas dedicadas, de alto processamento e que precisam de uma temperatura estável para o correto funcionamento.

De acordo com (BASTOS, 2013), sensores são dispositivos que variam suas propriedades sob a ação de uma grandeza física, fornecendo um sinal que indica essa grandeza. Ou seja, são componentes eletrônicos que realizam a aferição da temperatura e convertem os dados da entrada, na forma analógica, em dados eletrônicos de modo que registre ou monitore as mudanças térmicas ocorridas.

Os sensores de temperatura possuem duas características básicas podendo ser analógicos ou digitais.

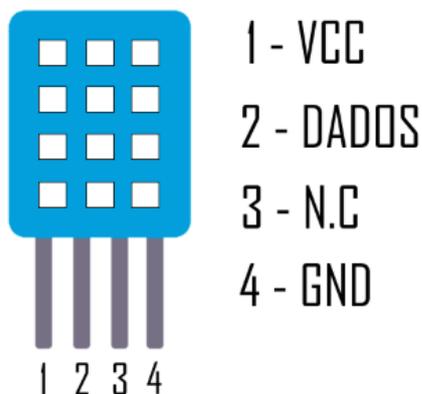
Os sensores analógicos representam sua saída por uma escala de tensão. Os sensores digitais representam dois valores de saída representados como 0 (zero) e 1 (um).

Abaixo, as figuras 2 e 3 representam o modelo dos sensores DHT11 e DHT22.

Figura 2 – Sensor de Temperatura DHT11 ([SARAVATI, 2021](#))Figura 3 – Sensor de Temperatura DHT22 ([DISTRELEC, 2021](#))

2.2.1 Sensor DHT11

O sensor de temperatura DHT11 é fabricado pela empresa D-Robotics UK e é capaz de aferir temperatura operacional de 0 a 50°C e precisão de $\pm 2^\circ\text{C}$. O sensor é digital e tem retorno de resolução de 8 bits. A figura 4 mostra a pinagem do sensor DHT11.

Figura 4 – Pinagem do Sensor de Temperatura DHT11 ([ROBOTIC-MENTE.BLOGSPOT, 2021](#))

O sensor de temperatura requer uma fonte de alimentação de operação entre 3,3V a

5,5V. Com uma fonte de 3,3V, mantêm-se os níveis de sinais dentro da faixa de segurança para o GPIO da Raspberry Pi.

De acordo com ([EBRAHIMI-DARKHANEH, 2010](#)), é recomendado o uso de um resistor de $5k\Omega$, entre o VCC e o DADOS, em uma alimentação de 3.3V para cabos de menos de 20 metros. Contudo, neste projeto utilizaremos o pino de 5V da Raspberry Pi para alimentar o Sensor DHT11 e, com isso, um resistor de $10K\Omega$.

A figura 5 mostra a conexão geral entre o Sensor DHT11 e a Raspberry.

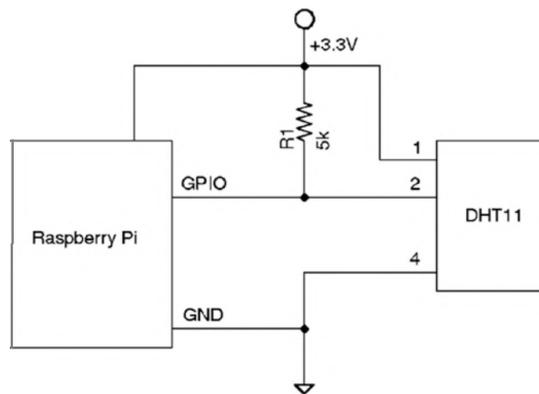


Figura 5 – Conexão entre Sensor DHT11 e Raspberry ([GAY, 2018](#))

Quando a Raspberry Pi está recebendo dados do Sensor a linha do pino GPIO irá flutuar. Por isso, a importância da utilização do resistor para que a linha do pino GPIO volte para o nível de 5V.

Com o intuito de otimizar este projeto, será utilizado o adaptador DHT11 com integração ao ESP01 proporcionando, assim, envio dos dados de temperatura via *Wifi*. Este conjunto eletrônico permite encaixe rápido entre módulo ESP01 e o sensor DHT11. Além disso, há um botão de reset e pinos de alimentação em todo o conjunto, permitindo o monitoramento dos dados longe do microcontrolador.

A figura 6 dá uma visão geral do tamanho do adaptador sensor DHT11 ESP01.

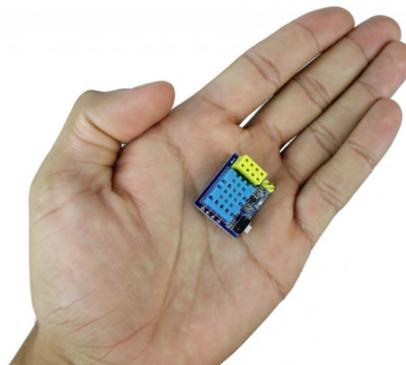


Figura 6 – Adaptador Sensor DHT11 ESP01 ([USINAINFO, 2022](#))

Especificações do Adaptador Sensor DHT11 ESP01:

- Modelo: ESP-01S DHT11 V1.0;
- Compatibilidade: Módulo Wifi ESP8266 ESP-01 ou ESP-01S;
- Tensão: 3.7V - 12VDC (suporta bateria de lítio 3.7V);
- Faixa medição de temperatura: 0 a 50°C;
- Margem de erro da temperatura: $\pm 2^\circ\text{C}$;
- Dimensões (CxLxA): 25 x 21 x 10mm; e
- Peso: 4g.

2.2.2 Funcionamento do Sensor DHT11

O Sensor DHT11 funciona quando estimulado pela Raspberry Pi e, para isso, há uma solicitação no barramento e, por conseguinte, uma resposta.

O processo de comunicação é ditado por uma interface serial onde um barramento único de dados é utilizado para comunicação e sincronização entre a Raspberry e o sensor. O processo é de 4ms e os dados contêm partes decimais e inteiras. A transmissão completa de dados é dada uma resposta de 40 bits de informação do sensor com 8 bits de uma soma de verificação.

A figura 7 mostra o processo de comunicação entre a Raspberry e o Sensor.

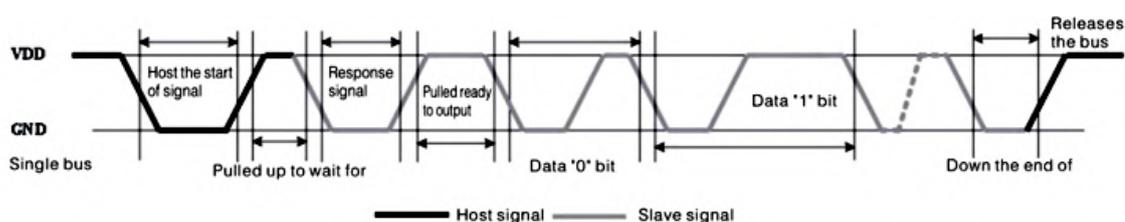


Figura 7 – Funcionamento da aquisição do sinal da Raspberry pelo sensor (EBRAHIMI-DARKHANEH, 2010)

O processo de comunicação é iniciado quando a Raspberry envia um sinal de início ao sensor alterando o modo de funcionamento do DHT11 de baixo consumo de energia para o modo execução.

A figura 8 mostra como é o funcionamento do envio de sinal pela Raspberry e aquisição pelo sensor DHT11. Ou seja, o processo de mudança do nível de tensão do barramento único de dados de alto para baixo dura cerca de 18ms. Após esse tempo, a Raspberry aguarda um período de 20us a 40us para a resposta do DHT11.

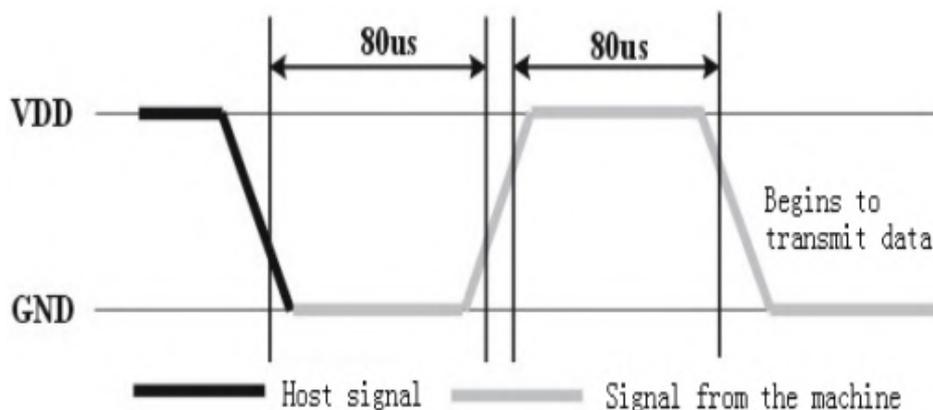


Figura 8 – Envio de sinal para o sensor DHT11 (EBRAHIMI-DARKHANEH, 2010)

Além disso, a figura 9 mostra a resposta do sensor DHT11 à Raspberry que é dada com o envio de um sinal de resposta, de baixo nível de tensão, 0 (zero), durante 80µs e, em seguida, permite que o nível de tensão retorne alto por mais 80µs para sinalizar que há intenção de retornar os dados da Raspberry para o sensor.

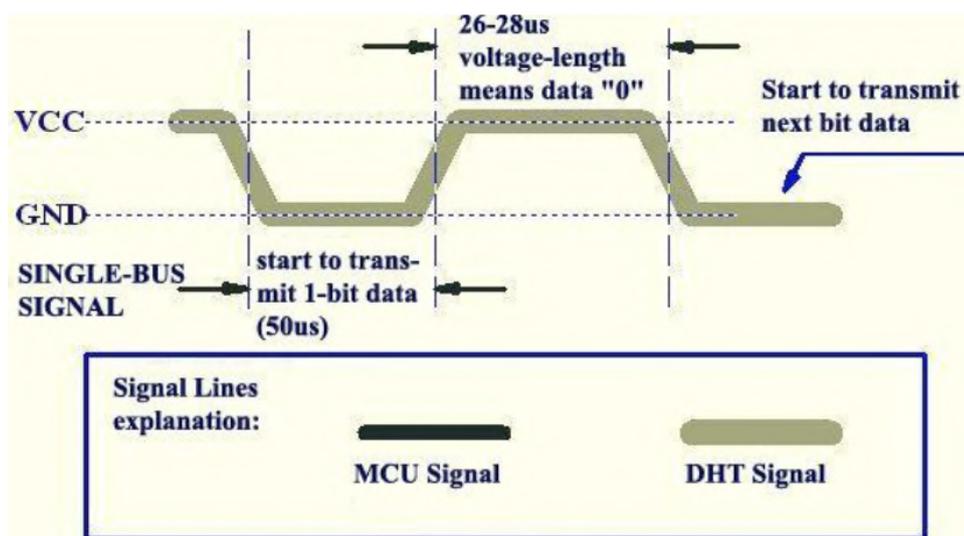


Figura 9 – Resposta do sensor à Raspberry (EBRAHIMI-DARKHANEH, 2010)

Resumo do funcionamento de aquisição de sinal entre a Raspberry e o DHT11:

- 1 O nível de tensão fica inativo devido o Resistor de $10K\Omega$ entre o VCC e o DADOS;
- 2 A Raspberry traz a tensão de nível alto para baixo, por pelo menos 18ms, para fazer uma solicitação de leitura e libera o barramento, permitindo que volte para o estado alto;
- 3 Após uma pausa de 20µs a 40µs, o DHT11 responde trazendo a linha de tensão para baixo por 80µs. Em seguida, permite que a linha retorne para o nível de tensão alto por mais 80µs e, assim, sinalizada a intenção de retornar dados;

- 4 Gravação de 40 bits de informação no barramento de dados da Raspberry pelo sensor DHT11 com cada bit começando com 50us de tensão baixo seguido por 26us a 28us de tensão alta, indicando um bit 0 e 70us de alta para indicar um bit 1;
- 5 A transmissão termina quando o sensor vai para tensão baixa uma vez a cada 50us;
e
- 6 Por fim, o sensor retorna para o estado de alta ociosidade.

2.2.2.1 Funcionamento do envio de DADOS Sensor-Raspberry

O envio de dados do sensor para a Raspberry é feito a cada nível de baixa tensão 50us e o que determina se o bit é 0 (zero) ou 1 (um) é o comprimento do sinal seguinte.

A figura 10 mostra a transição de tensão baixa seguida por uma transição de tensão alta. O final do bit ocorre quando há uma transição da tensão baixa para tensão alta.

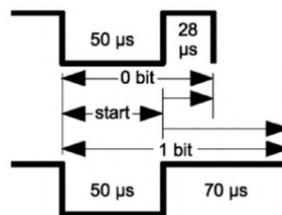


Figura 10 – Envio de DADOS do Sensor DHT11 (GAY, 2018)

Por fim, o sensor envia os dados em 40 bits, transmitindo o bit mais significativo primeiro. A figura 11 representa graficamente o formato de envio de dados do sensor à Raspberry. Ou seja, a folha de dados indica 16 bits de umidade relativa, 16 bits de temperatura em Celsius e uma soma de verificação de 8 bits.

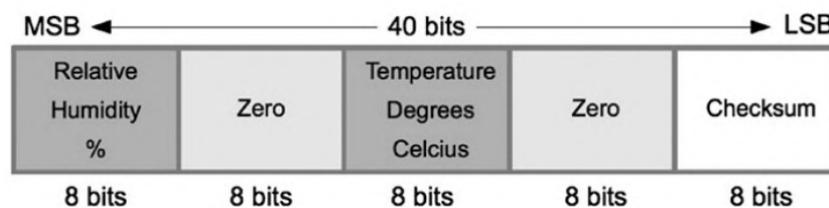


Figura 11 – Formato de envio de dados do Sensor DHT11 (GAY, 2018)

2.3 Raspberry Pi

A Raspberry Pi é um microcomputador de baixo custo criado pela empresa *Raspberry Pi Foundation* que tinha como ideia inicial desenvolver e incentivar jovens e crianças a embarcarem no mundo da programação. Com o passar dos anos, este dispositivo

tornou-se muito popular e há diversos projetos com as mais variadas aplicações onde ele é utilizado.

Não obstante, a placa possui diversos componentes integrados o que torna possível conectá-la a outros dispositivos externos por meio de um grupo de pinos chamados *General Purpose Input/Output (GPIO)*, permitindo a placa ser compatível a trabalhar com aplicações IoT.

2.4 Ferramentas *open source* de monitoramento

As ferramentas de monitoramento são partes essenciais para a gestão de TI e estratégias de segurança em qualquer empresa. Por isso, é importante a escolha de uma ferramenta que atenda às necessidades de cada projeto.

Diante disso, deve-se levar em consideração algumas variáveis tais como o custo associado à ferramenta de monitoramento, disponibilidade das funcionalidades e a quantidade de dispositivos que podem ser monitorados.

Há várias ferramentas *open source* de monitoramento de infraestrutura. Dentre elas, as mais famosas são Cacti, Nagios e Zabbix.

2.4.1 Cacti

Esta ferramenta fornece dados sobre desempenho, avalia diferentes métricas e plota informações por meio de gráficos.

Pontos positivos:

- Armazenamento configurável de históricos;
- Criação de fonte de dados utilizando o MySQL;
- Controle de permissões;
- Envio de alertas via e-mail e *script* personalizado;
- Fácil manipulação;
- Modelos gráficos pré-configurados;
- Número ilimitado de gráfico por host;
- Rápido re-sequenciamento dos itens gráficos;
- Suporte ao protocolo SNMP; e
- Suporte completo ao RRDTOOL (sistema de base de dados round-robin).

Ponto negativo:

- A ferramenta é só uma interface de gerenciamento. Para que seja possível o monitoramento e levantamento de dados em gráficos, necessita-se de softwares complementares como: RRDTool, MRTG ou Nagios.

2.4.2 Nagios

Este software fornece o monitoramento de toda estrutura da rede de serviços e hosts. Os alertas são configuráveis e, caso seja detectado algum problema, é enviado esse alerta por meio de efeitos sonoros, e-mail e SMS.

Pontos positivos:

- Integração com plug-ins: o próprio usuário pode escrever as linhas de comandos para automatizar e facilitar ainda mais o monitoramento da rede, o código pode ser escrito em diferentes linguagens de programação;
- Manutenção preventiva alertando possíveis falhas no sistema; e
- Monitora vários servidores, possibilitando que todos os parâmetros monitorados sejam agregados em uma única interface.

Pontos negativos:

- Apresenta limitação para se adaptar à rápida evolução das redes atuais;
- Falta de recursos gráficos de configuração: exige do usuário certo conhecimento na área de redes; e
- Indisponibilidade de ferramentas de configuração na interface web.

2.4.3 Zabbix

Esta ferramenta monitora vários parâmetros da rede e tem um sistema de alerta bem flexível que permite alertas por e-mail. É estruturado em código aberto, é escrito e distribuído sob a GNL (*General Public license*). Além disso, sua estrutura é dividida em vários componentes conhecidos: *Zabbix sever*, *Zabbix proxy* e *Zabbix agent*.

Pontos positivos:

- Agentes de alto desempenho;
- Interface web;

- Monitoramento distribuído com administração centralizada na web; e
- Suporte para protocolo SNMP.

Ponto Negativo:

- O Unix é o único sistema operacional que pode consistentemente responder ao desempenho necessário para instalação do Zabbix;

3 Metodologia

Esse capítulo apresenta a metodologia definida na construção do trabalho, mostrando em forma de blocos as etapas do trabalho. Para isso, este trabalho considera primordial a busca por uma solução que considere a simplicidade, alto custo-benefício e que atenda a requisitos definidos para etapas futuras de desenvolvimento. Diante disso, foi definida uma metodologia definida como Top-Down, que consiste na divisão por blocos de um sistema geral.

3.1 Metodologia Top-Down

A partir de uma visão geral do sistema, obtém-se blocos menores da estrutura do trabalho para uma melhor compreensão dos subsistemas com a finalidade de se obter melhores observações dos níveis básicos do projeto. Na figura 12 é apresentado o diagrama de blocos do fluxo de trabalho. Cada bloco do fluxo de trabalho define sua função geral e é utilizada como referência pelos sub blocos presentes, o que minimiza o risco de possíveis erros de definição nos blocos de baixo nível, facilitando as alterações e correções no projeto.

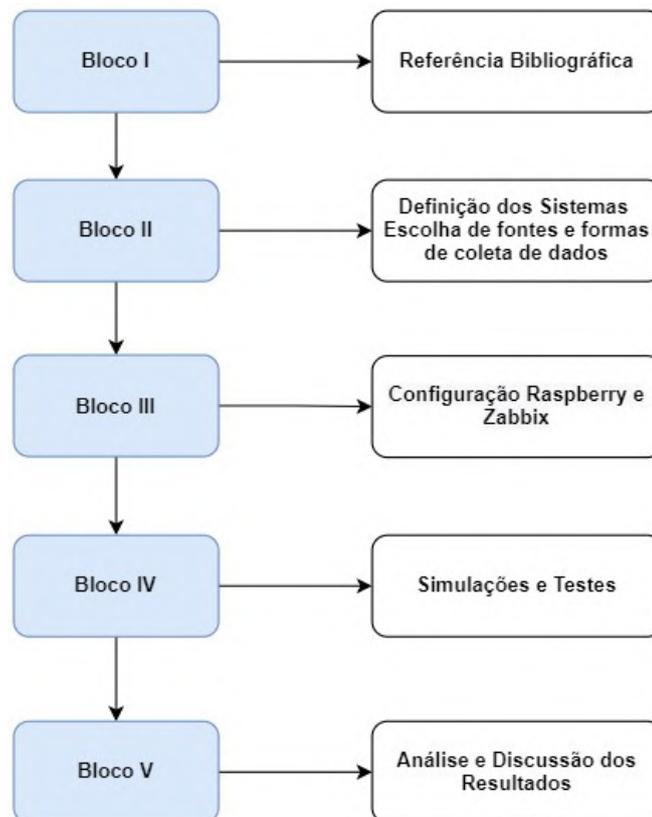


Figura 12 – Diagrama de blocos do fluxo de trabalho

3.2 Bloco I - Referência Bibliográfica

O primeiro bloco apresenta as definições relevantes ao tema após analisado o problema geral. Por meio de uma revisão bibliográfica, são abordados os principais conceitos envolvendo o Monitoramento contínuo de Temperatura, o microcomputador Raspberry Pi 3B+ e o software de monitoramento Zabbix. As definições dos conceitos utilizados serão necessárias para o entendimento do tema e dos subsistemas que o compõem, tornando esta etapa uma das mais importantes de todo o trabalho.

3.3 Bloco II - Definição dos Sistemas: Escolha de fontes e formas de coleta de dados

Os dados de interesse são analisados e definidos a partir deste bloco. Para isso, foi levado em consideração as especificações do projeto e o levantamento bibliográfico realizado no Bloco I. O Monitoramento contínuo da temperatura é realizado pela Raspberry, adquirindo os valores do sensor DHT11, e apresentados no Zabbix. Por meio desta etapa haverá a tradução da grandeza física da temperatura em um nível elétrico proporcional às mesmas possibilitando, assim, a leitura e o tratamento do sinal pelo pelos conversores analógico-digital da Raspberry.

3.4 Bloco III - Configuração Raspberry e Zabbix

Etapa responsável por definir a rede de comunicação para aquisição de dados confiáveis e um caminho fluido, sem perdas de dados, entre o sensor DHT11, a Raspberry e o Zabbix. A topologia do projeto é um mapa de rede que indica segmentos e é única para cada projeto. Com isso, é necessário ajustar a entrada de sinal do sensor DHT11 na Raspberry para que haja uma aquisição do sinal mais confiável, o *frontend* da ferramenta Zabbix para a apresentação dos dados de temperatura coletados, configurar as *triggers* de alarme e o protocolo SNMP para o envio de E-mail.

3.5 Bloco IV - Simulações e Testes

Neste bloco, as topologias do Monitoramento contínuo de temperatura são simuladas. Para a simulação dos sensores foi utilizada a *protoboard* para obter estímulos iniciais, de forma que, esses representem o comportamento dos sinais dos sensores. Posteriormente, serão utilizados os componentes na sala de servidores para obter as características reais.

3.6 Bloco V - Análise e Discussão dos Resultados

Por último, no quinto bloco, é executada a análise e discussão dos resultados obtidos. Desse modo, este bloco finaliza o trabalho proposto, realizando uma análise dos resultados obtidos e dos resultados esperados e, por conseguinte, discussão da coleta e tratamento dos resultados pelo microcomputador. Além disso, são indicados possíveis mudanças para a continuação do projeto.

4 Desenvolvimento

O desenvolvimento do projeto está fundamentado na metodologia apresentada anteriormente. Posto isso, serão apresentados a sala de servidores, os sensores de temperatura DHT11, a Raspberry Pi e o Zabbix. Este tópico tem como objetivo analisar a aquisição dos dados dos sensores de temperatura, pela Raspberry, e o monitoramento do software Zabbix.

4.1 Sala de Servidores do CINDACTA I

Para operação ininterrupta dos equipamentos na sala de servidores, há de se pontuar dois importantes pontos: a energia e a temperatura da sala. A alimentação energética ininterrupta é garantida por por várias fontes de alimentação, banco de baterias e geradores que estão alocadas em linha. Ou seja, caso haja falha de alimentação em uma das linhas a outra assume automaticamente. O resfriamento da sala de servidores é uma questão mais complexa. Por isso, há necessidade de haver pelo menos um sistema confiável de monitoramento da temperatura.

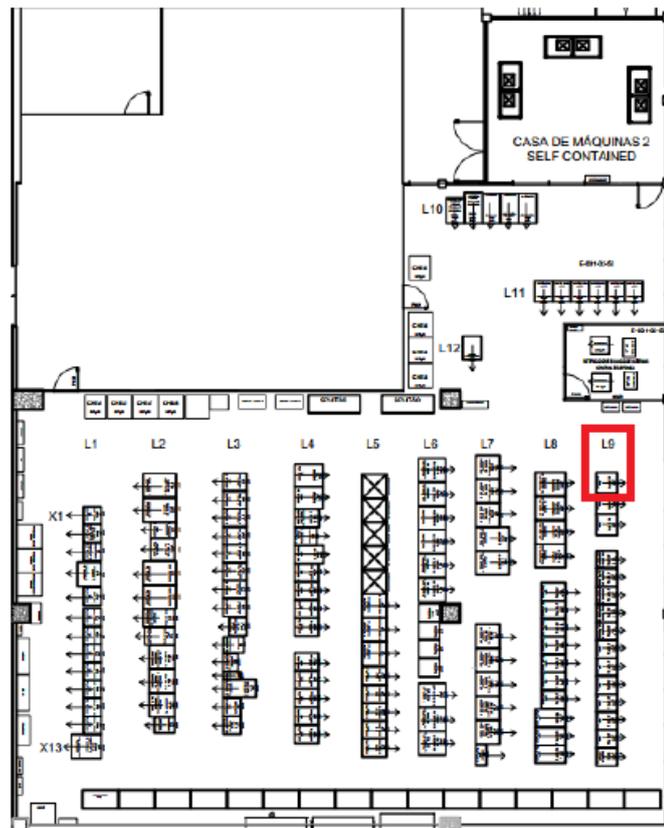


Figura 13 – Planta baixa da Sala de Servidores (ENGENHARIA, 2020)

A figura 13 mostra a planta baixa da sala de servidores e as identificações de cada *Rack* de Servidor.

A identificação é realizada numerando-se as fileiras da esquerda para a direita e os bastidores de cima para baixo. Ou seja, o equipamento **L9x1** está localizado na nona coluna (da esquerda para a direita) e primeira posição (de cima para baixo).

Diante do exposto acima, este projeto irá atender a área crítica, em servidores com maior temperatura dentre toda a sala.

Para a escolha dos locais críticos, será utilizado o estudo de Simulação de Dinâmica dos Fluidos (FFD) da empresa Fox Engenharia e Consultoria.

A figura 14 destaca os servidores críticos, com temperaturas elevadas e preocupantes aos sistemas que integram estes bastidores.

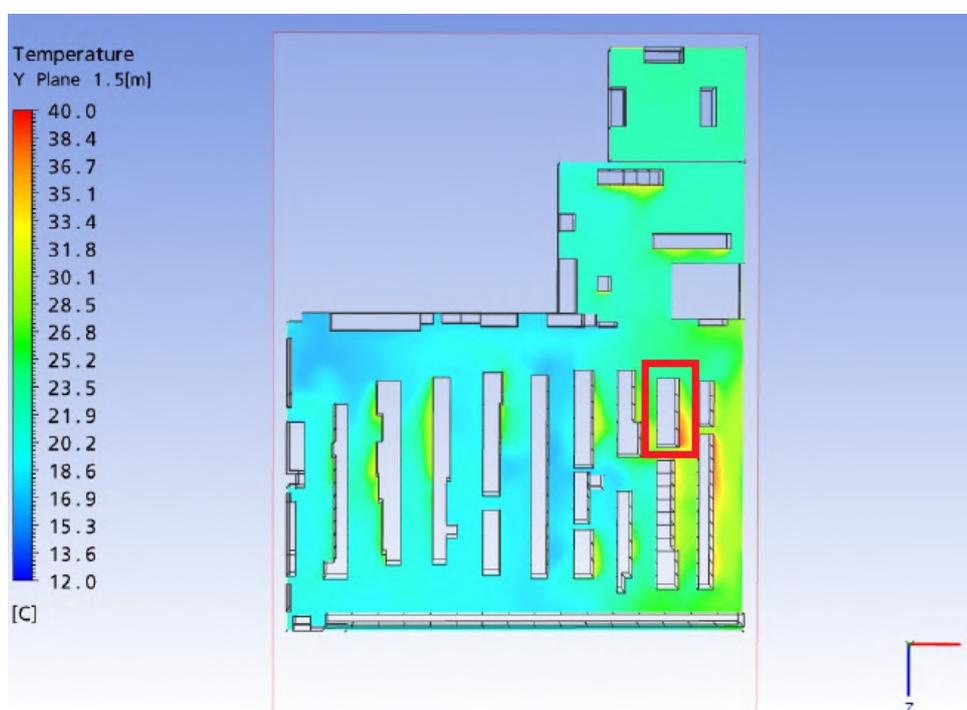


Figura 14 – Gráfico de Fluxo de Temperaturas (ENGENHARIA, 2020)

De acordo com a figura acima, foi proposto a instalação de 4 sensores DHT11 nos respectivos *Racks* L8X1, L8X2, L8X3 e L8X4 por estarem com temperatura mais críticas.

4.2 Sensor de Temperatura

Para o projeto, foram pesquisados dois modelos de sensores de temperatura de baixo custo e de mesmo fabricante. Abaixo, a tabela 1 demonstrada as especificações dos sensores DHT11 e DHT22 de acordo com o *datasheet*.

Especificações	DHT11	DHT22
Tensão de Alimentação	3,3V - 5,5V	3,3V - 6V
Corrente Máxima de Captura	2,5mA	1,5mA
Faixa de Leitura (Temperatura)	0 - 50°C	-40°C - 125°C
Precisão	$\pm 2^{\circ}C$	$\pm 0,5^{\circ}C$
Intervalo entre Medições	1s	2s
Preço Médio	R\$ 15,00	R\$ 50,00

Tabela 1 – Especificações dos Sensores DHT11 e DHT22 ([EBRAHIMI-DARKHANEH, 2010](#)) ([CO., 2010](#))

Optou-se por utilizar o sensor de temperatura DHT11 devido sua faixa de temperatura ser adequada às condições do projeto, ter intervalo curto de resposta entre as medições e ser mais barato.

4.3 Microcontrolador ESP01

A ESP01, figura 15, é um microcontrolador que permite conexão de rede sem fio, via TCP/IP, com outros microcontroladores. Seu principal objetivo é realizar conexão *wifi* para transmissão de dados de remoto.

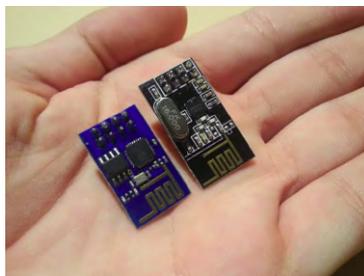


Figura 15 – Módulo microcontrolador ESP01 ([EMBARCADOS, 2022](#))

Especificações do módulo ESP01:

- Chip ESP8266-01;
- Modelo ESP01;
- Tensão de operação: 3,3V;
- Suporte a redes: 802.11 b/g/n;
- Alcance: 90m;
- Comunicação: Serial (TX/RX);
- Suporta comunicação TCP e UDP;

- Dimensões: 25 x 14 x 1mm; e
- Peso: 7g

A figura 16 mostra a pinagem do microcontrolador ESP01.

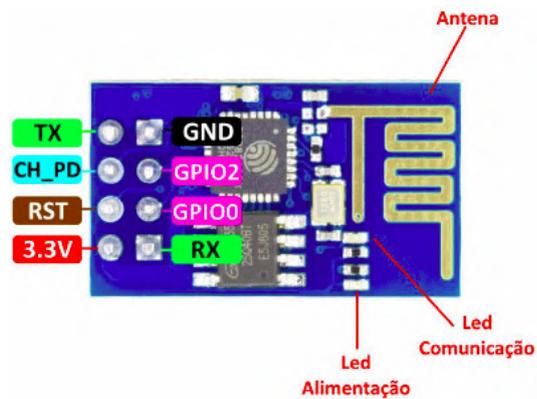


Figura 16 – Pinagem do Microcontrolador ESP01 (FALCÃO; WONZOSKI, 2019)

4.4 Raspberry Pi

O primeiro modelo comercial criado pela *Raspberry Pi Foundation* foi a Raspberry Pi 1 Model B, em 2012 (FOUNDATION, 2021), baseada no Sistema em um Chip (SoC), que incluía um processador ARM1176JZF-S de 700 MHz e 512 de memória RAM.

Com a evolução da tecnologia embarcada e de chips cada vez menores e mais eficientes, versões mais potentes da Raspberry foram criadas. A tabela 2 faz um comparativo dos modelos mais modernos da Raspberry levando-se em consideração a velocidade do processador, do mais fraco ao mais forte, da esquerda para a direita.

	Raspberry Pi Zero	Raspberry Pi 3	Raspberry Pi 3 model B+	Raspberry Pi 4 Model B
Tipo Chip (Broadcom)	BCM2835	BCM2837	BCM2837B0	BCM2711
Tipo Core	ARM1176JZF-S	Cortex-A53 64bit	Cortex-A53 (ARMv8) 64bit	Cortex-A72 (ARM v8) 64bit
Nº Core	1	4	4	4
Clock CPU	1GHz	1.2GHz	1.4Ghz	1.5GHz
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore VI
RAM	512MB	1GB	1GB LPDDR2 (900 MHz)	2GB, 4GB ou 8GB
Wireless	X	802.11n	802.11n 2.4GHz/5.0GHz	802.11ac 2.4GHz/5.0GHz
Bluetooth	X	4.1 (BLE)	4.2 (BLE)	5.0 (BLE)
Consumo	100mA	350mA	500mA	600mA
Valor	R\$ 200,00	R\$ 300,00	R\$ 400,00	R\$ 800,00

Tabela 2 – Comparação entre Modelos da Raspberry Pi (HUTCHINSON, 2021).

Diante do exposto acima, considerou-se utilizar a Raspberry Pi 3 Model B+ por possuir o modelo mais novo e rápido de Wirelees, promovendo rapidez e maior integração

com as tecnologias mais recentes de IoT. Além disso, possui bom processador e seu maior diferencial é o custo-benefício.

A figura 17 a seguir representa a comparação do tamanho da Raspberry Pi 3B+.



Figura 17 – Raspberry Pi 3B+ (FOUNDATION, 2021)

4.4.1 Sistema Operacional da Raspberry Pi

Para o funcionamento da Raspberry, é necessário que um Sistema Operacional (SO) esteja instalado. Além disso, a arquitetura do SO, que é o conjunto de rotinas que oferece serviços ao usuário e às aplicações, será determinístico para o bom funcionamento do Software Zabbix.

Ademais, de acordo com a Documentação do Zabbix, o Servidor Zabbix (*Zabbix Server*) deve estar hospedado em uma máquina baseada na família Unix (Linux ou Mac OS) (ZABBIX, 2021). Diante disso, optou-se por utilizar o Sistema Operacional Raspbian - Raspberry Pi OS.

A Raspbian se refere a uma composição de Raspberry Pi e Debian e é o Sistema Operacional oficial da Raspberry com direito à suporte e atualizações. Além disso, é uma distribuição com kernel Linux podendo assim ser instalado o *Zabbix Server*.

Por fim, não serão abordados os processos de configuração inicial da Raspberry Pi ou do Sistema Operacional Raspbian.

4.5 Software *Open Source* de Monitoramento de Ativos de Rede

A tabela 3 faz um comparativo e demonstra as principais características das ferramentas de monitoramento abordadas neste trabalho.

Tabela Comparativa							
Software	Plataforma	Sem Agentes	SNMP	Gráficos	Alerta	Licença	Controle de Acesso
CACTI	C++, Java	Não	Sim	Sim	Sim	GPL	Sim
NAGIOS	C	Suportado	Plug-in	Sim	Sim	GPL	Sim
ZABBIX	C, PHP	Suportado	Sim	Sim	Sim	GPL	Sim

Tabela 3 – Tabela Comparativa de Softwares

Diante do exposto acima, foi considerada a ferramenta Zabbix como a plataforma mais completa possuindo tanto a interface de monitoramento e alerta em tempo real do Nagios e o histórico de informações e gráficos do Cacti.

4.6 Zabbix

O Zabbix é um software *Open Source* de monitoramento de ativos de rede e, com isso, pode-se obter a performance e disponibilidade de todos os serviços. Ou seja, desde aplicações envolvidas de rede até diferentes equipamentos tais como servidores, roteadores, *hosts*, *switches* e os mais diversos sensores.

A coleta de dados pelo Zabbix pode ser dada por *scripts*, via agente ou por meio do protocolo SNMP - *Simple Network Management Protocol* que é um protocolo padrão de internet para gerenciamento de dispositivos em redes IP. O protocolo SNMP é definido pela IETF - *Internet Engineering Task Force* que é um grupo internacional aberto composto por técnicos, agências, fabricantes, fornecedor e pesquisadores provendo um conjunto de padrões de gerenciamento de rede, incluindo protocolo da camada de aplicação, banco de dados e um conjunto de objetos de dados.

Para armazenar todas as informações requisitadas, o Zabbix utiliza um banco de dados. Diante disso, o Zabbix tem suporte para diversos banco de dados tais como o MariaDB, MySQL, PostgreSQL, SQLite e o Oracle.

Além da sua aplicabilidade, o Zabbix tem interface web amigável e possibilita acesso a informações armazenadas anteriormente que são importantíssimas na hora da consulta e análise de alertas. Estes alertas permitem que os problemas sejam identificados e encaminhados para equipe responsável por tomar decisões de gerência.

A portabilidade do Zabbix é um grande diferencial quando se pensa em utilizar em conjunto com a Raspberry Pi pois sua utilização é dada em diferentes sistemas operacionais como Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, Mac OS X e Windows possibilitando monitoramento entre plataformas diferentes.

Contudo, devido ser uma ferramenta *open source* há uma primitiva em que o seu servidor esteja hospedado em uma máquina Linux ou Mac OS por não haver pacote do

servidor disponível para as versões do Windows.

Conquanto, não há qualquer interferência no monitoramento dos ativos de rede por causa da estrutura que forma o Zabbix.

Por fim, a estrutura do Zabbix está situada em três partes distintas:

- Servidor Zabbix (*Zabbix server*);
- Agente Zabbix (*Zabbix Agent*); e
- Interface do Zabbix (Interface Web).

Servidor Zabbix: Precisa de um banco de dados para a coleta e o armazenamento das configurações e dos dados coletados pelo agente Zabbix. Além disso, deve-se estar hospedado em uma máquina baseada na família Unix (Linux ou Mac OS). Ou seja, o *Zabbix server* realiza o *polling*, a requisição parte do servidor NMS, que é o servidor responsável por realizar o gerenciamento de redes, em direção ao *host* monitorado, e *trapping*, também chamado de *push* ou *passive check*, que é quando o dispositivo gerenciado decide quando enviar os alertas, e são eventos iniciados e efetuados por processos ou aplicações externas, tendo seus resultados submetidos ao NMS para processamento de dados e calcular *triggers* que são expressões lógicas que analisam os dados coletados pelos itens e representam o estado do sistema em relação aos mesmos enviando, assim, notificações aos usuários.

Agente Zabbix: É instalado no sistema que será monitorado pois é o responsável em repassar todas as informações que foram coletadas ao servidor Zabbix. O Agente roda no sistema como um processo *daemon*, processo de segundo plano, ou serviço. Ou seja, o servidor solicita a requisição e o agente processa e retorna os dados.

Interface Zabbix: Permite ao administrador a instalação, configuração e navegação de dados. É acessada via web por meio do próprio *browser* do sistema operacional.

Na construção deste projeto, foi escolhida a versão Zabbix 5.0 por ter suporte ao Sistema Operacional Raspbian, monitoramento mais seguro e confiável que versões anteriores, monitoramento de fácil utilização e gerenciamento e ser uma versão amadurecida no mercado.

4.6.0.1 Pré-requisitos de sistema para instalação do Zabbix

Segundo (ZABBIX, 2021), para um funcionamento mínimo do Zabbix é necessário que o Hardware atenda algumas especificações. Vale ressaltar que estas especificações podem mudar de acordo com a quantidade de *triggers*, *Hosts* e itens a serem monitorados.

A tabela 4 representa as informações retiradas da própria documentação.

Requisitos Mínimos de Hardware	
Memória	128 MB
Armazenamento	256 MB
CPU	Core 2 GB

Tabela 4 – Requisitos mínimos de Hardware

4.6.0.2 Interface Web do Zabbix - *Front-end*

O *front-end* do Zabbix é desenvolvido em linguagem interpretada livre PHP e oferece interface gráfica intuitiva e a possibilidade de geração de gráficos e relatórios.

O acesso à interface gráfica é realizado por meio da página web do Zabbix acessando o endereço `http://ip.do.servidor/zabbix`. Além disso, são necessários os dados de acesso ao banco de dados do Zabbix para autenticação.

A figura 18 mostra a tela de autenticação do Zabbix acessada por meio do IP do servidor em um navegador Web.

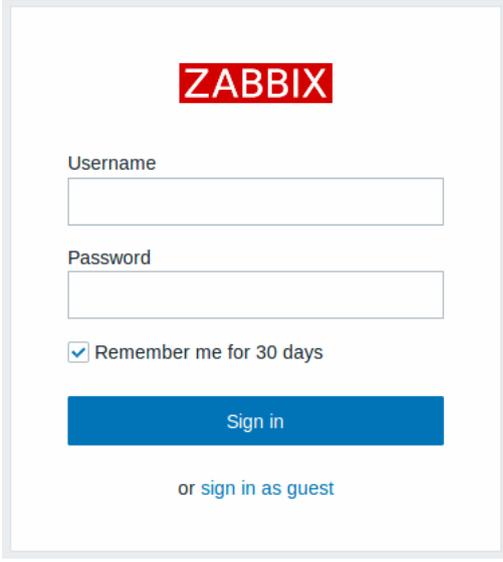
A imagem mostra a interface de autenticação do Zabbix. No topo, há o logotipo "ZABBIX" em um retângulo vermelho. Abaixo dele, há o rótulo "Username" seguido de um campo de entrada de texto. Logo abaixo, há o rótulo "Password" seguido de um campo de entrada de texto. Abaixo do campo de senha, há uma caixa de seleção marcada com um ícone de checkmark, seguida pelo texto "Remember me for 30 days". Abaixo disso, há um botão azul com o texto "Sign in". Na base da tela, há o link "or sign in as guest" em azul.

Figura 18 – Tela de autenticação do Zabbix (ZABBIX, 2021)

Após a autenticação, será apresentada a tela resumo (*Dashboard*), ou tela de apresentação, do Zabbix. Após configurada, essa tela apresentará as informações gerais do sistema, *hosts*, incidentes, gráficos e outras ferramentas configuradas pelo usuário.

A figura 19 representa a tela inicial do Zabbix após configurado pelo usuário.

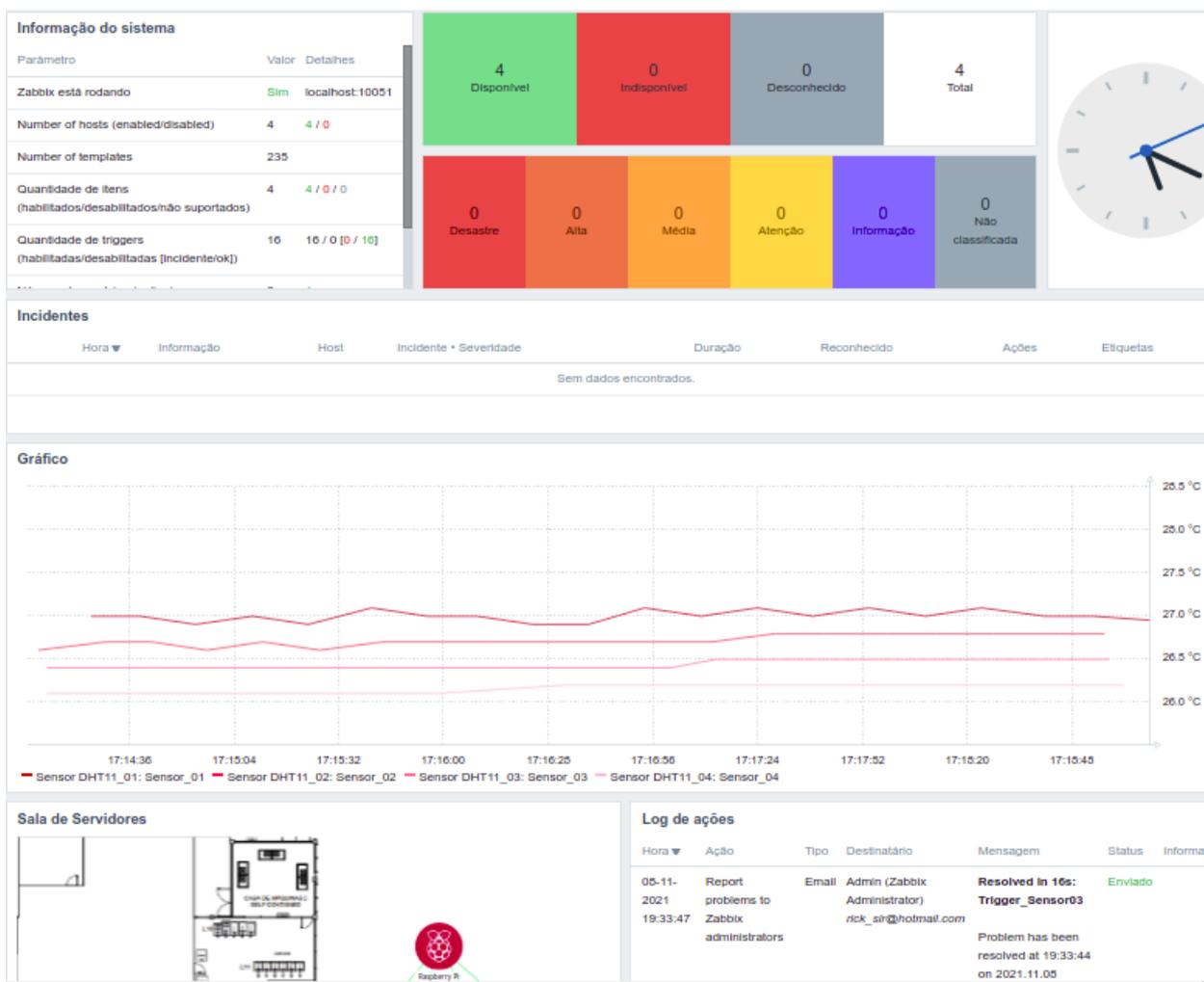


Figura 19 – Tela *Dashboard* Zabbix

A figura 20 representa o acesso do Zabbix ao *Shell Script* do SO da Raspaberry, onde está rodando o código em C++ do sensor DHT11, no qual ele coleta o valor bruto da Temperatura e armazena os dados do sensor. Apesar do sensor representar o valor da umidade e temperatura, é configurado, no Zabbix, fazer a aquisição somente do valor da temperatura que é o escopo do projeto.

```

pi@raspberrypi: ~/Documents
pi@raspberrypi:~/Documents $ ./read_DHT11 7
Humidity : 42.0 %
Temperature : 23.1 C
pi@raspberrypi:~/Documents $ ./read_DHT11 0
Humidity : 44.0 %
Temperature : 22.7 C
pi@raspberrypi:~/Documents $ ./read_DHT11 2
Humidity : 45.0 %
Temperature : 22.8 C
pi@raspberrypi:~/Documents $ ./read_DHT11 3
Humidity : 44.0 %
Temperature : 22.7 C
pi@raspberrypi:~/Documents $

```

Figura 20 – Teste dos Sensores de Temperatura

Para demonstrar estes valores no Zabbix é necessário criar, primeiramente, um *host* para cada sensor. O *host* é uma entidade de rede que você deseja monitorar podendo ser físico ou virtual.

A figura 21 mostra informações do sistema e nela podemos verificar se o Zabbix *server* está rodando, o número de *hosts*, que representam os quatro sensores, a quantidade de itens e *triggers*.

Parâmetro	Valor	Detalhes
Zabbix está rodando	Sim	localhost:10051
Number of hosts (enabled/disabled)	4	4 / 0
Number of templates	235	
Quantidade de itens (habilitados/desabilitados/não suportados)	4	4 / 0 / 0
Quantidade de triggers (habilitadas/desabilitadas [incidente/ok])	16	16 / 0 [0 / 16]
Número de usuários (online)	2	1
Desempenho requerido do servidor, novos valores por segundo	0.27	

Figura 21 – Informações do sistema no Zabbix

Após a criação dos *hosts*, são criados os Itens. A figura 22 mostra a tela de itens que são elementos responsáveis por receber os dados e definirem o que e quando coletá-los. Pode-se observar que o item está ligado diretamente ao *host*.

Itens

Todos os hosts / Sensor DHT11_01 Ativo ZBX SNMP JMX IPMI Aplicações Itens 1 Triggers 4 Gráficos 1 Regras de descoberta Cenários web

Grupos de hosts Selecionar Tipo todos Tipo de informação todos

Hosts Selecionar Intervalo de atualização Histórico

Aplicação Selecionar Estatísticas

Nome

Chave

Aplicar Limpar

Subfiltro afeta somente a área com filtro

<input type="checkbox"/>	Assistente	Nome	Triggers	Chave	Intervalo	Histórico	Estatísticas	Tipo	Aplicações	Status
<input type="checkbox"/>	***	Sensor_01	Triggers 4	dht.pull	15	90d	365d	Agente Zabbix		Ativo

Figura 22 – Tela de criação de Itens

Enquanto os Itens coletam dados, as *Triggers* avaliam estes dados com expressões que definem o limite do aceitável ou não aceitável. Caso algum nível declarado for ultrapassado, será acionado um estado de nome "Incidente" e, quando retornado ao nível aceitável, a *Trigger* volta para o estado "Ok". Nesta imagem, pode-se notar os valores de temperatura definidos para cada nível de severidade e suas funções preditivas.

Severidade	Valor	Nome ▲	Dados operacionais	Expressão	Status
Atenção	OK	Trigger_Sensor01		(TRIGGER.VALUE)=0 and (Sensor DHT11_01:dht.pull.last())>32	Ativo
Desastre	OK	Trigger_Sensor01		(TRIGGER.VALUE)=0 and (Sensor DHT11_01:dht.pull.last())>=35	Ativo
Não classificada	OK	Trigger_Sensor01		(Sensor DHT11_01:dht.pull.last())=0	Ativo
Alta	OK	Trigger_Sensor01		(TRIGGER.VALUE)=0 and (Sensor DHT11_01:dht.pull.last())>=34	Ativo

Figura 23 – Configuração da *Trigger* do Sensor 1

Por fim, é configurado o protocolo SNMP para o envio de e-mail sempre que houver algum incidente e se o problema foi resolvido.

As figuras 24 e 25 demonstram o envio de e-mail, ao usuário cadastrado, no momento que o incidente foi causado com informações do tipo e severidade do incidente. Assim, como o horário que o incidente foi resolvido e o tempo em que permaneceu sem resolver.

Com as informações enviadas por e-mail, é possível realizar ação rápida para mitigar o problema. Abaixo, segue uma breve explanação pontuando as informações que são enviadas pelo Zabbix.

- *Problem start at/has been resolved*: Contém o horário e data em que se iniciou/resolveu;
- *Problem name*: Refere-se à *trigger* específica do sensor;
- *Problem duration*: Para o caso do problema resolvido, mostra o tempo que o incidente durou.
- *Host*: Mostra em qual sensor ocorreu o problema;
- *Severity*: Informa qual o nível de severidade do incidente; e
- *Original problem ID*: Código do incidente, fator importante para análise de Logs do sistema.



Figura 24 – E-mail de Incidente



Figura 25 – E-mail de Incidente Resolvido

Vale ressaltar que, para a realização dos testes, foi montado o circuito dos sensores DHT11 em uma *protoboard* com a finalidade de adquirir os dados de temperatura, realizar a configuração do Zabbix e testar a confiabilidade das informações adquiridas.

5 Resultados e Discussões

Neste capítulo, serão apresentados e discutidos os resultados do projeto experimental e validação do protótipo. Além disso, os dados são tabulados e são gerados gráficos para embasar a análise.

5.1 Cenário do Experimento

Para a realização deste trabalho, foi utilizada a Sala de Servidores como referência de cenário do experimento. foi utilizado o Zabbix como ferramenta do sistema de monitoramento da temperatura. A coleta de dados é realizada a cada 20 segundos e 4 informações por requisição. Cada informação é referente a um sensor, ou seja, serão utilizados 4 Módulos adaptadores DHT11/ESP01.

O cenário do experimento, leva em consideração o dispositivo sensor coletando a Temperatura do ambiente em graus Celsius, no qual, será gerenciado e monitorado por um servidor Zabbix instalado na Raspberry Pi. Por conseguinte, a ferramenta Zabbix realiza a cada 20 segundos a coleta dos dados relativos à falha e desempenho.

Foram utilizados quatro (4) módulos IoT para aferição da temperatura dos servidores mais críticos, um módulo em cada servidor. Além disso, os dados são enviados de forma independente para a ferramenta Zabbix que gerenciará o conjunto de dados e enviará os alertas de criticidade por email utilizando protocolo SMTP. A figura 26 representa o esquemático de ligação e os protocolos de comunicação utilizados no projeto.

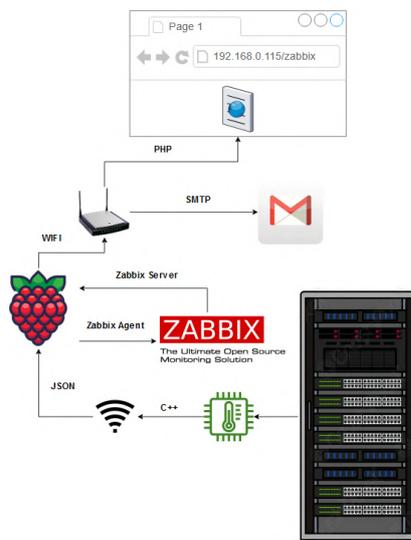


Figura 26 – Layout Projeto

5.1.1 Montagem dos Sensores

A figura 27 apresenta a ligação final do Módulo Sensor DHT11 com a ESP01. Decidiu-se fazer a ligação em separado para que não houvesse influência da temperatura, da ESP01 no módulo Sensor, na aquisição dos dados.



Figura 27 – Módulo Sensor DHT11 ligado à ESP01

Foi utilizada uma fonte ajustável para alimentação do Módulo Sensor DHT11/ESP01, conforme mostra a figura 28. Além disso, foi utilizado *protoboard* para deixar organizado.

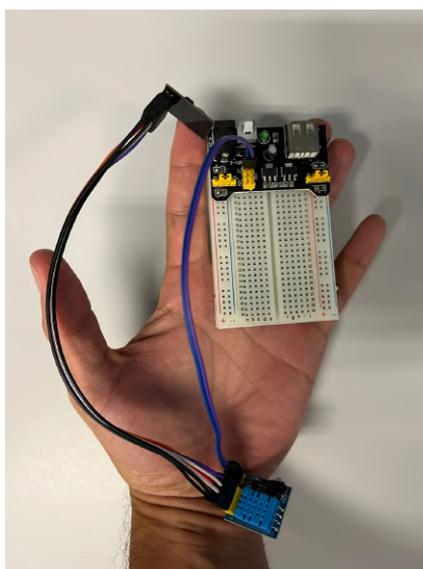


Figura 28 – Módulo Sensor DHT11/ESP01 e regulador de tensão

5.1.2 Instalação do Módulo Sensor DHT11/ESP01 nos Servidores

A figura 29 mostra o resultado final da instalação módulo Sensor. Foi utilizada a régua de energia, que alimenta o servidor, e uma fonte 12V para energizar o projeto.



Figura 29 – Módulo Sensor DHT11/ESP01 instalado no Servidor

Vale ressaltar, que o uso de uma fonte para a alimentação do projeto é primordial para que não haja fluxo de energia excessivo ou insuficiente e, também, como segurança tendo em vista a importância dos equipamentos embarcados no servidor.

5.2 Execução e Monitoramento

Para esta parte do projeto, foram instalados e configurados as ferramentas e sistemas necessários para o monitoramento dos servidores, ou seja, a solução foi colocada em produção. Com isso, teremos como referência a coleta de dados em um prazo de 4 dias ininterruptos e a obtenção dos dados, relativos à falha e desempenho, de temperatura acontecem em um intervalo de 20 segundos.

Nesse período, foram coletados 69.120 amostras de temperatura e requisições do servidor para o agente. Pois, como o servidor dispara requisições a cada vinte segundos, e nesse momento são solicitadas quatro informações de falha e desempenho diferentes, no período de cada prova, tem-se um total de doze rajadas de quatro requisições por minuto. Esse tempo é suficiente para confirmação da temperatura no ambiente.

A figura 30 representa a tela *dashboard*, que é um painel visual que apresenta, de maneira centralizada, um conjunto de informações com indicadores, métricas e alertas dos sistemas, utilizado para o monitoramento contínuo da temperatura.

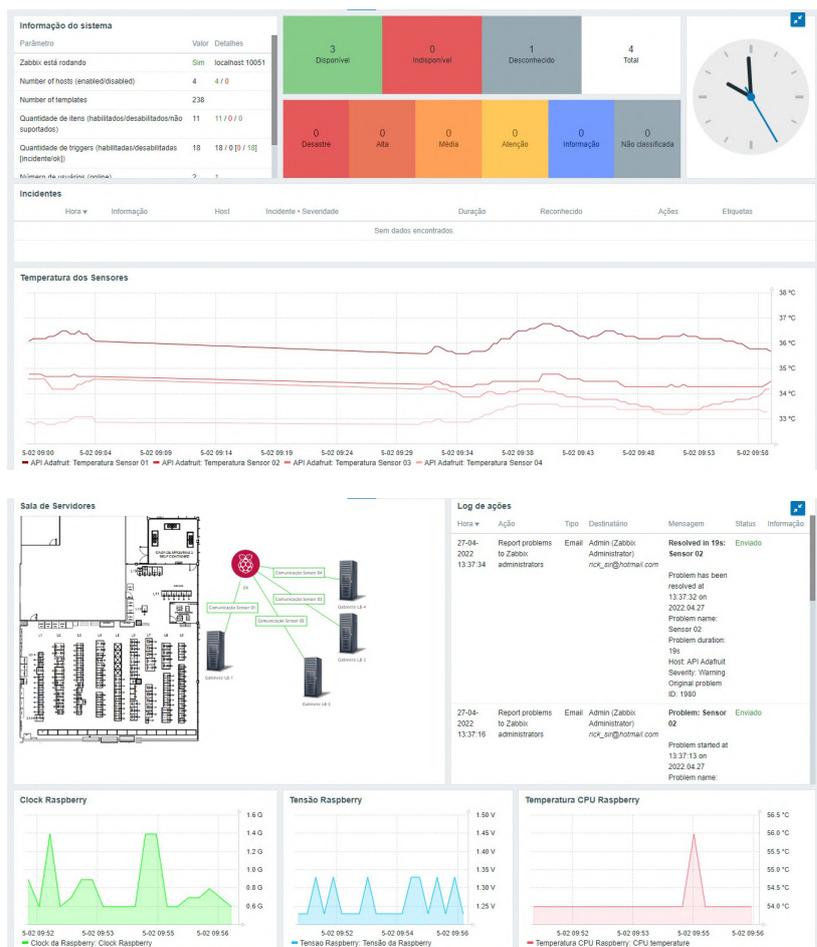


Figura 30 – Dashboard Zabbix

5.2.1 Alertas e Alarmes

Os alertas de temperatura foram configurados por meio das *triggers* do Zabbix. Além disso, o Zabbix envia mensagens eletrônicas (e-mails) em caso de alerta, e quando o alerta é resolvido, por meio do protocolo SMTP.

A figura 31 mostra a configuração das *triggers* no Zabbix.

<input type="checkbox"/>	Severidade ▲	Valor	Nome	Dados operacionais	Expressão	Status
<input type="checkbox"/>	Não classificada	OK	Sensor 02		{API Adafruit.info.sensor03.2.last()}=0	Ativo
<input type="checkbox"/>	Não classificada	OK	Sensor 03		{API Adafruit.info.sensor03.3.last()}=0	Ativo
<input type="checkbox"/>	Não classificada	OK	Sensor 04		{API Adafruit.info.sensor03.4.last()}=0	Ativo
<input type="checkbox"/>	Não classificada	OK	Sensor 01		{API Adafruit.info.sensor03.1.last()}=0	Ativo
<input type="checkbox"/>	Atenção	OK	Sensor 02		{API Adafruit.info.sensor03.2.last()}>40	Ativo
<input type="checkbox"/>	Atenção	OK	Sensor 03		{API Adafruit.info.sensor03.3.last()}>40	Ativo
<input type="checkbox"/>	Atenção	OK	Sensor 04		{API Adafruit.info.sensor03.4.last()}>40	Ativo
<input type="checkbox"/>	Atenção	OK	Sensor 01		{API Adafruit.info.sensor03.1.last()}>40	Ativo
<input type="checkbox"/>	Alta	OK	Sensor 01		{API Adafruit.info.sensor03.1.last()}>45	Ativo
<input type="checkbox"/>	Alta	OK	Sensor 02		{API Adafruit.info.sensor03.2.last()}>45	Ativo
<input type="checkbox"/>	Alta	OK	Sensor 03		{API Adafruit.info.sensor03.3.last()}>45	Ativo
<input type="checkbox"/>	Alta	OK	Sensor 04		{API Adafruit.info.sensor03.4.last()}>45	Ativo
<input type="checkbox"/>	Desastre	OK	Sensor 01		{API Adafruit.info.sensor03.1.last()}>=50	Ativo
<input type="checkbox"/>	Desastre	OK	Sensor 02		{API Adafruit.info.sensor03.2.last()}>=50	Ativo
<input type="checkbox"/>	Desastre	OK	Sensor 03		{API Adafruit.info.sensor03.3.last()}>=50	Ativo
<input type="checkbox"/>	Desastre	OK	Sensor 04		{API Adafruit.info.sensor03.4.last()}>=50	Ativo

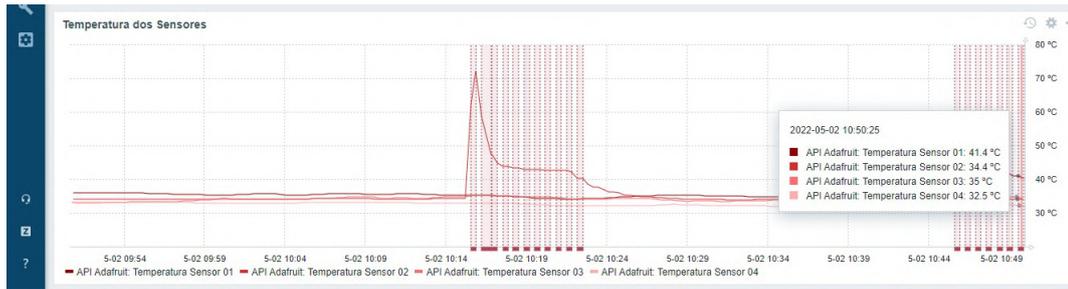
Figura 31 – Triggers Módulo Sensor DHT11/ESP01

Foram configurados quatro níveis de criticidade, padrão Zabbix, e suas respectivas mensagens para os alertas das *triggers*.

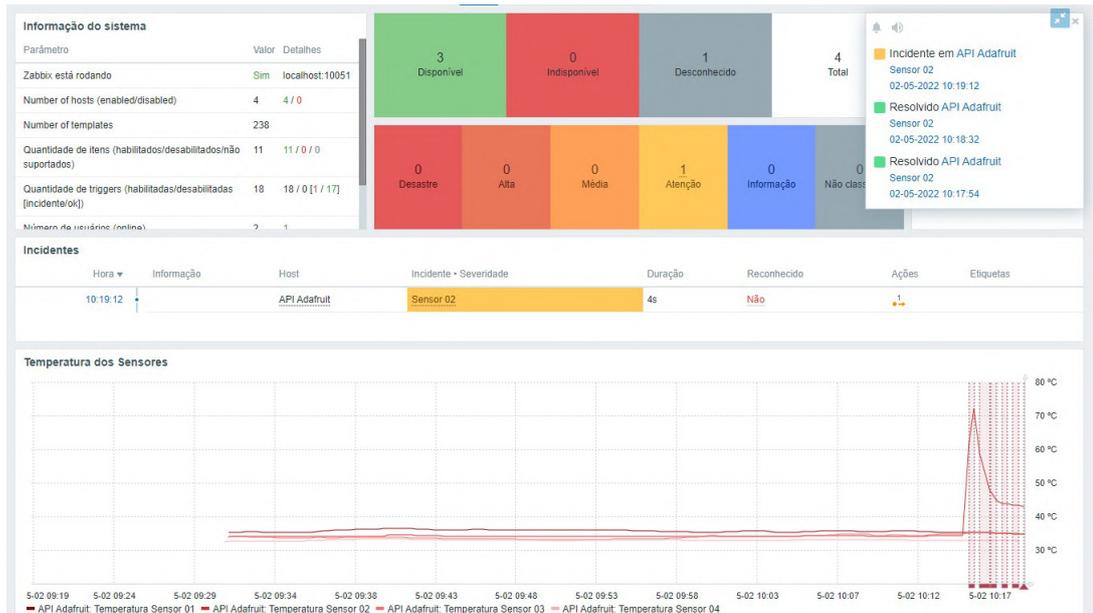
1. Informação: Quando há falha de comunicação do Sensor de temperatura;
2. Atenção: Temperatura maior que 40 graus Celsius;
3. Alta: Temperatura maior que 45 graus Celsius; e
4. Desastre: Temperatura maior que 50 graus Celsius.

A seguir, será demonstrado, em alguns sensores, os testes *in loco* de todo o sistema em funcionamento.

A figura 32 mostra o alerta na tela *dashboard* quando a Temperatura é maior que 40°C.



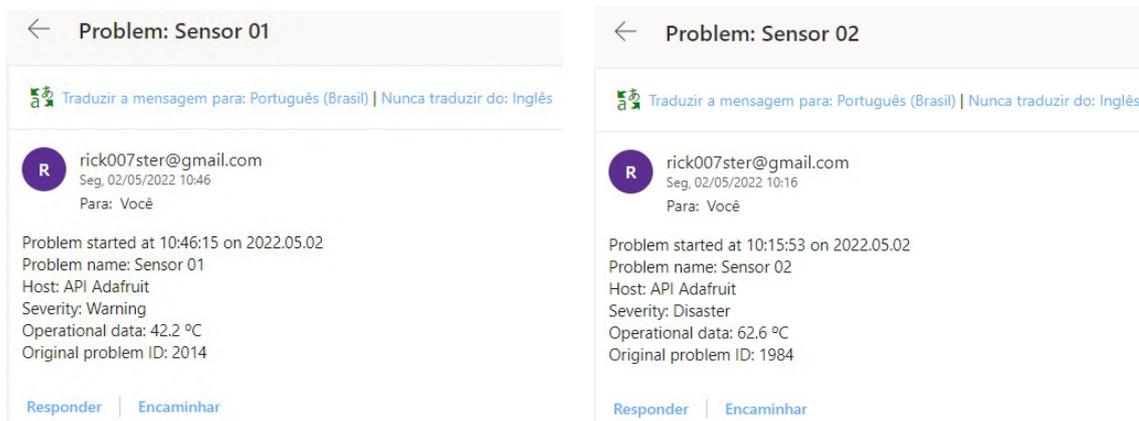
(a) Sensor 01



(b) Sensor 02

Figura 32 – Alertas no Dashboard

Após ocorrer o incidente, é enviado *e-mail* informando o detalhamento do alerta. A figura 33 mostra o e-mail encaminhado pelo Zabbix.

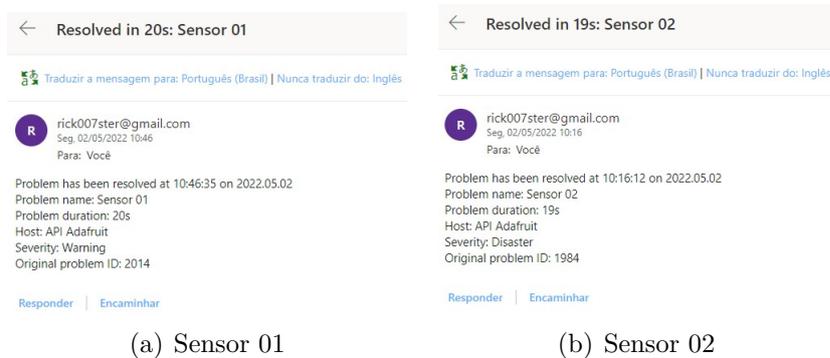


(a) Sensor 01

(b) Sensor 02

Figura 33 – E-mail dos Alertas

Após resolvido o incidente, o Zabbix envia outro e-mail com dados do incidente. A figura 34 mostra o e-mail enviado e a confirmação com o ID de cada alerta gerado.



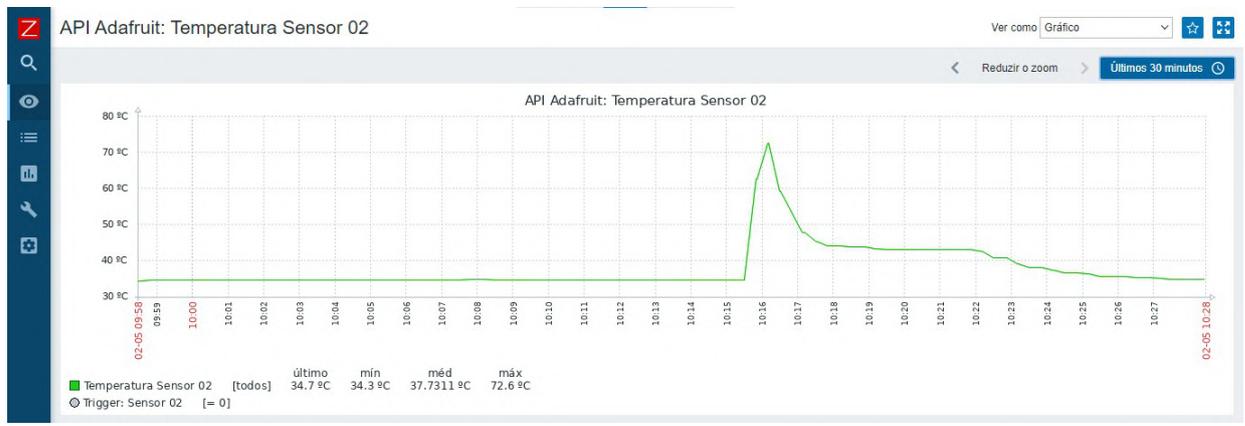
(a) Sensor 01 (b) Sensor 02

Figura 34 – E-mail do Alerta resolvido

Além disso, é possível ver todo o histórico de temperatura coletado pelo agente Zabbix e, com isso, analisar os horários com maiores picos de temperatura e menor temperatura. A figura 35 demonstra o histórico de temperatura.



(a) Sensor 01



(b) Sensor 02

Figura 35 – Histórico de Temperatura

6 Conclusão

O foco deste trabalho foi projetar um sistema capaz de armazenar dados, realizar o monitoramento e enviá-los a um cliente na internet. Entende-se que o monitoramento da temperatura na sala de servidores é de extrema importância e tem a finalidade de garantir que a temperatura esteja conforme a norma técnica dos equipamentos e, assim, manter a integridade de computadores dedicados e sistemas em total funcionamento.

É importante ressaltar que a Internet das Coisas transforma a rotina de várias empresas de setores diversos. Isso porque, é possível otimizar processos com base na geração de dados que são disponibilizados. Nas indústrias, principalmente, a IoT tem ajudado a identificar os problemas a fim de corrigi-los.

A utilização da Raspberry em conjunto com o Zabbix é uma solução com bom custo-benefício, fácil atualização e de integração com diversos sistemas. A utilização de um software de monitoramento em qualquer computador, com acesso a um simples navegador web, proporciona maior praticidade ao técnico que não precisa estar diretamente no local para saber a temperatura da sala de servidores, possibilitando uma resposta rápida a falhas nos sistemas com a ajuda de alarmes de incidentes e um histórico confiável de relatórios para pesquisa de pane.

Ao final deste trabalho, é possível elencar as contribuições à seguir:

- Solução de baixo custo;
- Monitoramento contínuo em vários bastidores da Sala de Servidores;
- Banco de dados com armazenamento da temperatura;
- Sistema de alertas de criticidade;
- Envio de alertas por Email, utilizando-se protocolo SMTP; e
- Menor tempo de respostas à incidentes.

Por fim, conclui-se que a implementação de um sistema de monitoramento na Sala de Servidores, em conjunto com a ferramenta Zabbix, oferece maior segurança, disponibilidade e integridade dos sistemas presentes nos bastidores.

7 Trabalhos Futuros

Aproveitando-se da integração da ferramenta Zabbix, Raspberry e IoT, há possíveis atualizações no projeto, tais como:

- Monitoramento energético dos equipamentos críticos dos bastidores;
- Controle automático dos ar condicionados atrelados à cada tipo de incidente;
- Produção de uma Placa de Circuito Impresso com integração de todos os componentes;
- Criação de um Mapa de Calor com distribuição de Sensores em vários pontos da Sala de Servidores;

Referências

- BASTOS, A. *Instrumentação Eletrônica Analógica e Digital para Telecomunicações*. [S.l.]: Independente, 2013. v. 3. 238 p. Citado na página 18.
- CO., A. E. *Digital-output relative humidity temperature sensor/module DHT22*. [S.l.], 2010. 10 p. Citado 2 vezes nas páginas 9 e 32.
- DA, S. I. L. de O.; DE, J. D. S. O impacto do avanço da internet das coisas no brasil. *Brazilian Journal of Development*, 2020. Citado 2 vezes nas páginas 7 e 15.
- DISTRELEC. Sensor de umidade e temperatura de modelo dht22. In: . [S.l.: s.n.], 2021. Disponível em: <<https://www.distrelec.biz/en/dht22-temperature-and-humidity-sensor-5v-adafruit-385/p/30129240>>. Citado 2 vezes nas páginas 7 e 19.
- EBRAHIMI-DARKHANEH, H. *DHT11 Humidity Temperature Sensor*. [S.l.], 2010. 9 p. Citado 6 vezes nas páginas 7, 9, 20, 21, 22 e 32.
- EMBARCADOS. Módulo esp01. In: . [S.l.: s.n.], 2022. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Citado 2 vezes nas páginas 7 e 32.
- ENGENHARIA, F. Simulação de dinâmica dos fluidos (cfd) data center cindacta i - brasília relatório de análise. Centro Integrado de Defesa Aérea e Controle de Tráfego Aéreo, 2020. Citado 3 vezes nas páginas 7, 30 e 31.
- FALCÃO, P.; WONZOSKI, F. d. O. Desenvolvimento de um sistema de coleta e análise de dados de temperatura e umidade em parreirais de uva em tempo real. *Anuário pesquisa e extensão UNOESC Videira*, 2019. Citado 2 vezes nas páginas 7 e 33.
- FOUNDATION, R. P. Raspberry pi documentation. In: . [S.l.: s.n.], 2021. Disponível em: <<https://www.raspberrypi.com/documentation/>>. Citado 3 vezes nas páginas 7, 33 e 34.
- GAY, W. *Advanced Raspberry Pi*. [S.l.]: Apress Media LLC: Welmoed Spahr, 2018. v. 2. 521 p. Citado 3 vezes nas páginas 7, 20 e 23.
- HUTCHINSON, P. Raspberry pi 4, 3 b+, pi 3, pi 2, b+, a+ comparison chart, version 59. element14. In: . [S.l.: s.n.], 2021. Disponível em: <<https://www.element14.com/community/docs/DOC-68090/1/raspberry-pi-4-3-b-pi-3-pi-2-b-a-comparison-chart>>. Citado 2 vezes nas páginas 9 e 33.
- ROBOTICMENTE.BLOGSPOT. Sensor de umidade e temperatura de modelo dht11. In: . [S.l.: s.n.], 2021. Disponível em: <<https://www.roboticamente.blogspot.com/2019/11/sensor-de-temperatura-e-umidade-dht11.html>>. Citado 2 vezes nas páginas 7 e 19.
- SARAVATI. Sensor de umidade e temperatura de modelo dht11. In: . [S.l.: s.n.], 2021. Disponível em: <<https://www.saravati.com.br/sensor-de-temperatura-dht11-digital>>. Citado 2 vezes nas páginas 7 e 19.

USINAINFO. Adaptador sensor dht11 esp01. In: . [S.l.: s.n.], 2022. Disponível em: <<https://www.usinainfo.com.br/esp8266/adaptador-esp8266-esp-01-com-sensor-de-temperatura-e-umidade-dht11-5139.html>>. Citado 2 vezes nas páginas 7 e 20.

WEISER, M. Ubiquitous computing. *A variation of the article below appeared in IEEE Computer "Hot Topics"*, 1993. Citado na página 14.

ZABBIX. Documentação zabbix. In: . [S.l.: s.n.], 2021. Disponível em: <<https://www.zabbix.com/documentation/5.0/pt/manual/concepts/server>>. Citado 4 vezes nas páginas 7, 34, 36 e 37.

ZHANG Y., P. Z. J. J. L. H.; FUJITA, M. Temperature aware software-based self-testing for delay faults. *School of Software Engineering, Tongji University, China*, 2015. Citado na página 14.