CENTRO UNIVERSITÁRIO FADERGS

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MICROCONTROLADORES

MONITOR DE TEMPERATURA COM SNMP

Gustavo Conforti

Marcelo Jordano C. Menezes

Ronaldo Pinto

Porto Alegre, 2019

# **DEFINIÇÃO DO PROBLEMA**

Fornecer uma solução para monitoramento de temperatura de ambientes, como data centers, CPDs, depósitos de matéria prima, etc. Além do desenvolvimento de captura de temperatura através da plataforma Arduino, o projeto contará com o envio de informações para dispositivos externos.

O projeto surgiu de uma oportunidade comercial na empresa de um dos integrantes. É uma companhia de gerenciamento e implementação de redes, segurança e virtualização para terceiros. A ideia é criar um dispositivo que possibilite a monitoria remota da temperatura do data center do cliente 24/7, como serviço; o termômetro irá enviar dados para as ferramentas de monitoramento da empresa. Cada técnico levará um desses consigo e será bonificado com uma porcentagem de cada implementação.

Atualmente a empresa não disponibiliza controle de temperatura para os clientes, logo seria uma adição ao seu portfólio de produtos.

# **OBJETIVOS**

Proporcionar uma opção barata de geração de dados temperatura, para alimentar ferramentas externas de monitoramento de recursos. Com isto, será possível a configuração de *thresholds* máximos e mínimos e o envio de alertas mediante a ultrapassagem dos mesmos, uma vez os valores lidos pelo dispositivo poderão ser interpretados por softwares terceiros.

O projeto será baseado em três pilares principais. Para hardware, utilizaremos o módulo NodeMCU e o sensor de temperatura DHT11; eles serão responsáveis pela coleta e conversão dos dados climáticos do ambiente. O protocolo SNMP será o encarregado de transmitir essas informações para outros dispositivos em rede.

Como prova de conceito, usaremos uma VM do Zabbix para visualizar os dados gerados pelo termômetro.

1. **ANÁLISE DAS TECNOLOGIAS**
   1. NodeMCU

O NodeMCU é uma plataforma *open source* com o microcontrolador chinês ESP8266. Possui funcionalidade *Wi-Fi* integrada, por isso muitas vezes é chamado comumente de “módulo *Wi-Fi”.*

O ESP8266 pode funcionar no modo *standalone*, ou seja, sendo a MCU do projeto. É possível também utilizá-lo como escravo com interface em outra controladora central do projeto.

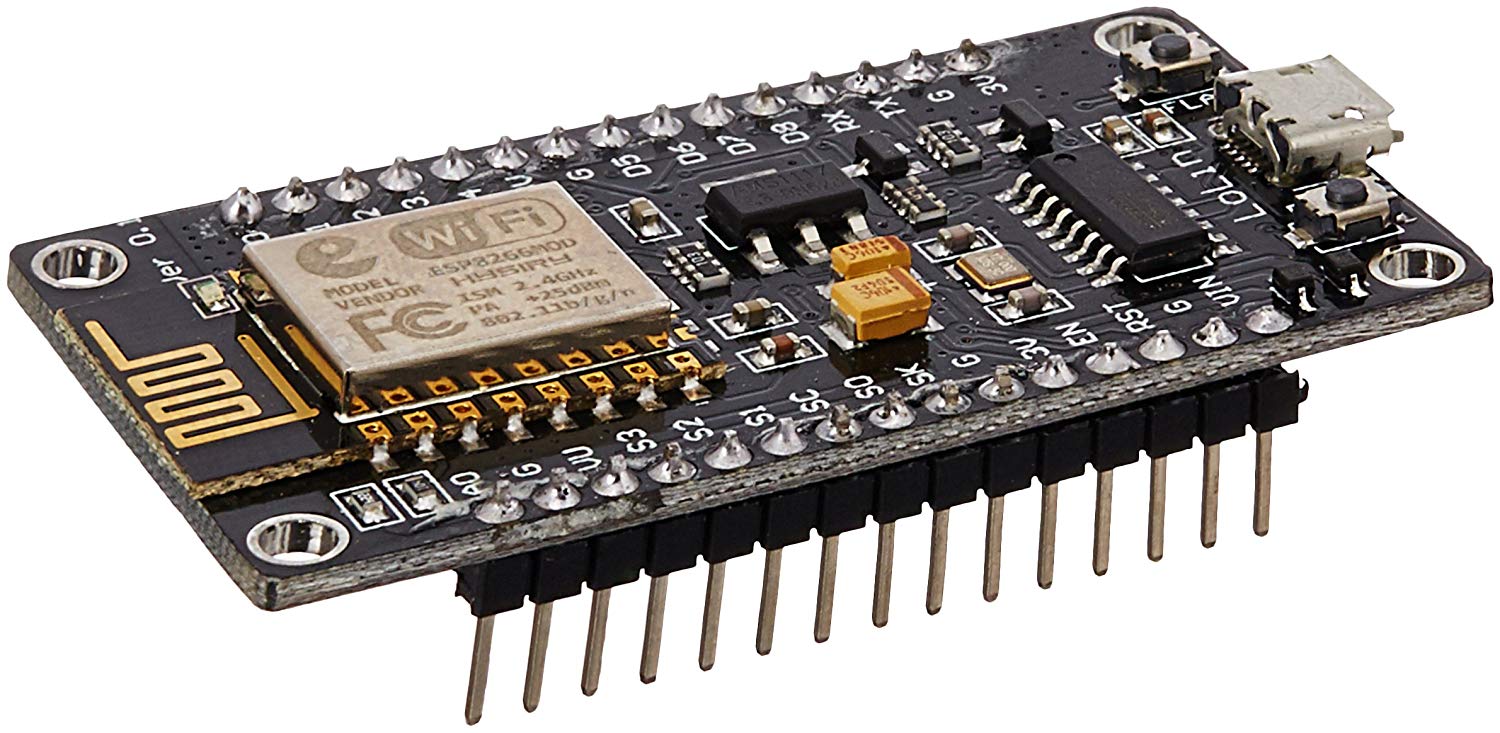


Figura 1 - NodeMCU

Algumas das características do NodeMCU podem ser vistas abaixo:

* Microcontrolador ESP8266-12E;
* Arquitetura RISC de 32 bits;
* Processador pode operar em 80MHz / 160MHz;
* 4Mb de memória flash;
* 64Kb para instruções;
* 96Kb para dados;
* WiFi nativo padrão 802.11b/g/n;
* Opera em modo AP, Station ou AP + Station;
* Pode ser alimentada com 5VDC através do conecto micro USB– Possui 11 pinos digitais;
* Possui 1 pino analógico com resolução de 10 bits
* Pinos digitais, exceto o D0 possuem interrupção, PWM, I2C e one wire;
* Pinos operam em nível lógico de 3.3V
* Pinos não tolerantes a 5V;
* Possui conversor USB Serial integrado;
* Programável via USB ou WiFi (OTA);
* Compatível com a IDE do Arduino ou programável na linguagem de programação desenvolvida por brasileiros, a LUA;
* Compatível com módulos e sensores utilizados no Arduino;
  1. Sensor DHT11

O DHT11 é um sensor de temperatura e umidade complexo, com saída de sinal digital calibrada. Contém medição de umidade de tipo resistivo e um componente de medição de temperatura NTC, conectado a um microcontrolador interno de 8 bits de alto desempenho. Este sensor é altamente utilizado em projetos de automação e provas de conceito de produtos. Possui uma resposta e precisão satisfatória, que aliada com o bom custo benefício o torna uma das principais opções para projetos simples de automação.

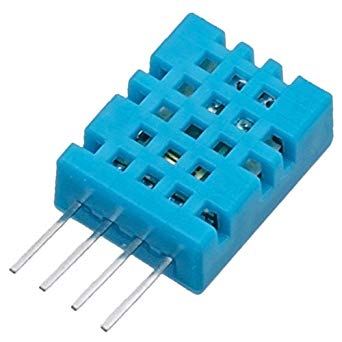


Figura 2 - Sensor de temperatura e umidade DHT11

Especificações técnicas:

* Faixa de medição de umidade: 20 a 90% UR;
* Faixa de medição de temperatura: 0º a 50ºC;
* Alimentação: 3-5VDC (5,5VDC máximo);
* Corrente: 200uA a 500mA, em stand by de 100uA a 150 uA;
* Precisão de umidade de medição: ± 5,0% UR;
* Precisão de medição de temperatura: ± 2.0 ºC;
* Tempo de resposta: 2s;
* Dimensões: 23 x 12 x 5mm (incluindo terminais);
  1. Protocolo SNMP

O Simple Network Management Protocol (SNMP) teve sua origem na RFC 1067 em 1988, evoluindo por algumas versões, estando atualmente na versão 3. É um protocolo da camada 7, que utiliza usualmente a porta 161 do protocolo 17.

É baseado no modelo gerente – agente; a estação de gerenciamento se comporta como cliente e o dispositivo de rede a ser monitorado se comporta como servidor, enquanto que na operação TRAP ocorre o oposto, pois é o dispositivo gerenciado que inicia a comunicação. Como as tarefas mais complexas de processamento e armazenamento de dados ficam com o gerente, o protocolo requer pouco processamento e pouco software, tornando sua adoção extremamente barata.

O protocolo, usado na maioria das vezes em [sistemas de gerenciamento de](https://pt.wikipedia.org/w/index.php?title=Sistemas_de_gerenciamento_de_rede&action=edit&redlink=1) dispositivos ligados em rede, é composto por dois objetos fundamentais: MIB e OID

* + 1. MIB

Estrutura em árvore padronizada que contém os objetos gerenciáveis (OIDs) de um determinado dispositivo de rede. Essa estrutura não tem limites e, de acordo com a necessidade, pode ser atualizada e expandida. Exemplo:

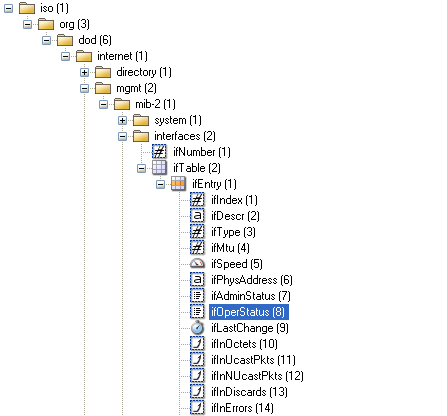


Figura 3 – Exemplo de MIB

* + 1. OID

Uma variável que de fato guarda uma informação do sistema, com uma identificação única denominada (Object IDentification), que é composta por uma seqüência de números que identifica a posição do objeto na árvore da MIB. Exemplo:

iso.3.6.1.2.1.17.4.0 = STRING: Comparitech

iso.3.6.1.2.1.17.4.1 = STRING: APC-3425

iso.3.6.1.2.1.17.4.2 = STRING: 3425EDISON

iso.3.6.1.2.1.17.4.4 = INTEGER: 72

iso.3.6.1.2.1.17.4.5 = STRING: veya

Ou, resolvendo seus nomes:

SNMPv2-MIB::sysContact.0 = STRING: Comparitech

SNMPv2-MIB::sysName.0 = STRING: APC-3425

SNMPv2-MIB::sysLocation.0 = STRING: 3425EDISON

SNMPv2-MIB::sysServices.0 = INTEGER: 72

IF-MIB:: ifDescr.1 =STRING: veya

* + 1. ZABBIX

Uma ferramenta de monitoramento de dispositivos e aplicações em rede. Mesmo sendo totalmente *open-source*, é altamente adotada no ambiente corporativo, devido a sua facilidade de implementação e vasta documentação; o grande número de usuários culmina em um grande número de casos de uso e *templates*, diminuindo o tempo de configuração e troubleshooting de novos monitores ou dashboards. Além disso, o protocolo aberto possibilita uma fácil integração com outras ferramentas, como o Grafana, para o desenvolvimento dashboards complexos, o ELK Stack, para centralização, armazenamento e correlacionamento de eventos, entre outros.

1. **DESCRIÇÃO DA SOLUÇÃO**
2. **METODOLOGIA**
3. **VALIDAÇÃO**
4. **REFERÊNCIAS**

* Datasheet ESP8266. Disponível em:

<https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Acessado em 31/10/2019.

* Datasheet DHT11. Disponível em:

[https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf. Acessado em 31/10/2019](https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.%20Acessado%20em%2031/10/2019).

* Datasheet ESP8266. Disponível em:

<https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Acessado em 31/10/2019.

* Teleco: Tutoriais de Banda Larga - SNMP. Disponível em:

<https://www.teleco.com.br/tutoriais/tutorialsnmp/pagina_2.asp>. Acessado em 31/10/2019.

* Samples de snmpwalk. Disponível em:

<https://networkengineering.stackexchange.com/questions/2990/translating-snmpwalk-output-into-human-readable-format>. Acessado em 31/10/2019.