UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DE CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS

GUILHERME BERGMANN CORREIA

INTEGRAÇÃO DA PLATAFORMA NAGIOS COM DISPOSITIVOS IOT VIA SNMP

CAXIAS DO SUL
2018

GUILHERME BERGMANN CORREIA

INTEGRAÇÃO DA PLATAFORMA NAGIOS COM DISPOSITIVOS IOT VIA SNMP

Trabalho de Conclusão de Curso para obtenção do grau de Bacharel em Ciência da Computação da Universidade de Caxias do Sul.

Orientadora: Professora Dra. Maria de Fátima Webber do Prado Lima.

CAXIAS DO SUL
2018

GUILHERME BERGMANN CORREIA

INTEGRAÇÃO DA PLATAFORMA NAGIOS COM DISPOSITIVOS IOT VIA SNMP

Trabalho de Conclusão de Curso para obtenção do grau de Bacharel em Ciência da Computação da Universidade de Caxias do Sul.

Aprovado(a) em 06 de julho de 2018

BANCA EXAMINADORA

_____
Profa. Dra. Maria de Fátima Webber do Prado Lima
Universidade de Caxias do Sul

_____
Profa. Dra. Carine Geltrudes Webber
Universidade de Caxias do Sul

_____
Profa. Dra. Elisa Boff
Universidade de Caxias do Sul

Dedico este trabalho à minha filha Isadora, à minha esposa Patrícia, aos meus pais Norberto e Juliana, à minha irmã Rafaela e aos meus sobrinhos Theo e Lucca, cujo apoio e incentivos foram fundamentais para que se tornasse possível finalizar esta etapa.

AGRADECIMENTOS

Agradeço à minha esposa Patrícia, aos meus pais Norberto e Juliana, ao meu amigo Rodrigo Minuscoli, por todo o incentivo recebido, e, em especial, aos professores Maria de Fátima Webber do Prado Lima e Dinarte Albuquerque Filho, pelo suporte e dedicação plenos.

"If you dream, it you can do it."

Walt Disney

RESUMO

Esta monografia tem como objetivo conectar dispositivos IoT com o software gerenciador de redes Nagios. O emprego do protocolo SNMP foi fundamental para realização dessa conexão. Para embasar este projeto, foram utilizados diversos conceitos, como noções de gerenciamento de redes TCP/IP e suas ferramentas auxiliares. O software escolhido para gerenciamento de dispositivos foi o Nagios, por se tratar de um software livre e com uma comunidade de desenvolvimento bastante difundida. O dispositivo IoT escolhido foi o HWG STE2 que conta de sensores de temperatura e umidade e é compatível com o protocolo SNMP. Para realizar a conexão entre o software e o dispositivo, se fez o uso da leitura das informações contidas nas MIBs do dispositivo IoT HWG STE2.

Palavras-chave: IoT. SNMP. MIBs. Gerenciamento de redes. TCP/IP. HWG STE2.

ABSTRACT

This project aims to connect IoT devices with the Nagios network management software. The use of the SNMP protocol was fundamental to make this connection. To support this project, several concepts were used, such as the definition of TCP/IP network management and its auxiliary tools. The software chosen for device management was Nagios because it is a free software with and with a widespread development community. The chosen IoT device was the HWG STE2 which is provided with temperature and humidity sensors and is compatible with SNMP protocol. To establish the connection between the software and the device, is used the reading information contained in the MIBs of the device HGW STE2.

Keywords: Computer Network. Network Management. Nagios. IoT. SNMP. MIB. HWG STE2.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notarion One
BER	Basic Encoding Rules
CGI	Common Gateway Interface
CMOT	Common Management Information Protocol over TCP/IP
DHCP	Dynamic Host Configuration Protocol
EGP	Exterior Gateway Protocol
FCAPS	Fault, Configuration, Accounting, Performance and Security
FPGA	Field Programmable Gate Array
GPL	General Public License
GPS	Global Positioning System
ICMP	Internet Control Message Protocol,
IETF	Internet Engineering Task Force
IoT	Internet of Things
IPv4	Internet Protocol Version 6
IPv6	Internet Protocol Version 6
ISO	International Organization for Standardization
ITU-T	Telecommunication Standardization Sector
LAN	Local Area Network
MIB	Management information base
MIT	Massachusetts Institute of Technology
NFC	Near Field Communication
NMS	Network monitoring system
OID	Object Identifier
QoS	Quality of service
REST	REpresentational State Transfer
RFC	Request for Comments
RFID	Radio-Frequency IDentification
SD Card	Secure Digital Card
SMI	Structure of Management Information
SMING	Structure of Management Information Next Generation

SMIv2	Structure of Management Information version 2
sms	Short Message Service
SMS	Short Message Service,
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
TCC	Trabalho de Conclusão de Curso
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
USB	Universal Serial Bus
VPN	Virtual Private Network
XML	Extensible Markup Language
POP3	Post Office Protocol 3
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
NNTP	Network News Transfer Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
CD	Compact Disc
GHz	GIGA-HERTZ.
GB	Giga Bytes
RAM	Random Access Memory
URL	Uniform Resource Locator

LISTA DE SÍMBOLOS

@ Arroba
® Marca Registrada

SUMÁRIO

1 INTRODUÇÃO

O presente trabalho trata da integração de dispositivos IoT com o software de gerenciamento de redes Nagios. O termo IoT (Internet of Things) refere-se a tudo que pode ser ligado à rede: eletrodomésticos, calçados, remédios, veículos, portas, paredes de casa, tênis, enfim, diversos objetos presentes no nosso dia a dia (DAL BERTO; CAGLIARI; COLLING, 2017).

Por definição, a Internet das Coisas (IoT) é uma rede de objetos físicos (PATEL, 2016). O conceito foi criado em 1999 a partir de um membro da comunidade de desenvolvimento de dispositivos RFID (identificação por radiofrequência) e, nos últimos anos, teve seu maior reconhecimento por causa do relevante crescimento de dispositivos móveis presentes em todos lugares, bem como pelo crescimento da computação em nuvem. O termo IoT, como se disse, abrange um cenário onde vários objetos do cotidiano estarão conectados à Internet e comunicando-se mutuamente (EVANS, 2011).

Exemplos simples de como a IoT pode estar presente na vida das pessoas são o dispositivo que permite controlar as luzes da casa de qualquer lugar do planeta pelo celular ou uma porta de garagem que abre sozinha ao detectar que o carro se aproxima; ou, ainda, portas de casa com reconhecimento facial ou biometria, e assim por diante (ENDEAVOR, 2015).

Conforme Patel (2016), a Internet das Coisas refere-se a um tipo de rede para conectar qualquer coisa com a internet. A IoT é baseada em protocolos, com o objetivo de permitir aos equipamentos serem aptos de auto-posicionamento, rastreamento, monitoramento e administração remota. Ele projeta um panorama onde bilhões de objetos poderão "sentir" através de sensores, comunicar-se entre si e compartilhar suas informações, todos interconectados em uma rede pública ou privada baseada em protocolo TCP/IP. Estes objetos terão seus dados coletados para alimentar sistemas de planejamento, gerenciamento e tomadas de decisão.

A Internet representou um grande salto na educação, na comunicação, nos negócios, na ciência, nos governos dos países e em toda a humanidade (EVANS, 2011). Os dispositivos IoT representam um grande passo na capacidade de coletar, analisar e distribuir dados em forma de informações e conhecimento. Ele acredita

que até o ano de 2020 haverá cerca de 50 bilhões de dispositivos conectados à Internet.

> Com um trilhão de sensores integrados no ambiente, todos conectados por sistemas de computação, software e serviços, será possível ouvir a batida do coração da Terra, impactando a interação humana com o globo de forma profunda da mesma forma que a Internet revolucionou a comunicação (HARTWELL in EVANS, 2011, p. 4).

De acordo com Evans, até o presente momento, todos os dispositivos IoT são compostos por redes diferentes de aplicação, como os prédios comerciais, que têm vários sistemas de controle para aquecimento, iluminação e segurança. Estes sistemas muitas vezes não são integrados em uma plataforma de controle única, o que torna o controle heterogêneo devido à falta de padronização dos dispositivos. À medida que a IoT evoluir, essas redes se tornarão mais homogêneas e estarão conectadas com mais recursos de segurança, análise e gerenciamento.

A IoT pode ser vista como uma rede das redes, segundo Evans (2011), que afirma que a IoT irá transformar a Internet em algo sensorial (temperatura, pressão, vibração, iluminação e umidade) e se expandirá rapidamente para locais até agora inatingíveis, como, por exemplo, pacientes que estão ingerindo dispositivos em seus corpos para auxiliar médicos a diagnosticar e determinar causas de determinadas doenças. Sensores extremamente pequenos podem ser colocados em corpos humanos, plantas e animais, bem como em recursos geológicos, conectados à Internet e distribuírem as informações buscadas. Quando são conectados mais objetos do que pessoas à Internet, uma grande perspectiva de oportunidades é aberta para a produção de aplicativos gerenciais nas áreas de automação (EVANS, 2011).

Conforme Evans, dentro do espectro da IoT, as vacas de uma empresa holandesa já estão conectadas, com sensores implantados que possibilitam aos fazendeiros acompanhar os movimentos e monitorar a saúde delas, garantindo maior qualidade de carne para o consumidor final. Em média, cada animal gera em torno de 200 megabytes de dados anualmente.

Evans (2011) cita alguns fatores que contribuem para que contribuem para retardar o desenvolvimento da IoT, sendo que os três principais são:

– A implantação do IPV6, visto que o número de endereços IPV4 estão drasti-

camente acabando, e os possíveis bilhões de novos dispositivos IoT exigirão endereços IP;

– A energia embarcada nos sensores: para que dispositivos IoT tenham seu potencial completo, eles deverão ser autossustentáveis. A troca de baterias de bilhões de dispositivos no planeta inteiro, e até mesmo no espaço, inviabiliza muitos projetos;

– A falta de padrões: embora exista muito progresso na área de normas, ainda não é o suficiente, especialmente no que tange às áreas de segurança, da privacidade, da arquitetura e dos protocolos de comunicação.

Considerando a gama de benefícios que os dispositivos IoT proporcionam, estes problemas devem ser resolvidos em breve. Será apenas uma questão de tempo, inclusive a padronização de gerenciamento. É que o crescente número de dispositivos embarcados, conectados na Internet, promove a necessidade de novas soluções de software para gerenciar dispositivos IoT de maneira eficiente, escalável e inteligente (VILLARI, 2014).

No contexto acima, faz-se necessária uma ferramenta que auxilie o gerenciamento de dispositivos IoT. Até porque, com o número de dispositivos IoT crescendo exponencialmente, houve um aumento proporcional no número de endereços na rede correspondente, e supõe-se que muitos outros dispositivos ainda serão conectados à infraestrutura de rede existente (GUPTA, 2014).

Como resultado, espera-se que o monitoramento se torne mais complexo para os administradores, pois as redes tendem a se tornar cada vez mais heterogêneas. Além disso, o endereçamento para IoTs também se tornará mais complexo, dada a escala em que os dispositivos estão sendo adicionados à rede, tornando a tarefa de gerenciá-los cada vez mais dificultosa.

Conforme a definição abordada, ao considerar dispositivos IoT como equipamentos conectados em rede, uma boa alternativa para gerenciá-los é fazer uso de um software de gerenciamento de redes. Um dos softwares que executa com eficiência este gerenciamento é o Nagios, ferramenta que permite administrar vários dispositivos e serviços disponíveis em uma rede de computadores. Ele contempla várias funcionalidades de gerenciamento, monitoramento e correção de falhas. Além

disso, o software Nagios disponibiliza um grande número de plug-ins, acessórios que podem ser agregados, tornando-o um software robusto e confiável para executar a tarefa de gerenciamento de dispositivos IoT.

O Nagios é uma ferramenta de gerência de redes que permite o monitoramento de infraestruturas de TI, dando ao gerente da rede a capacidade de identificar e solucionar problemas antes que eles se agravem e afetem processos críticos (JUNIOR, 2016). Foi criado em 1999 por Ethan Galstad, e ainda é mantido por ele e sua equipe. É um software licenciado sob os termos da GPL versão 2. Ganhou cinco vezes consecutivas o prêmio Linux Journal Reader's Choice Awards da revista *Linux Journal*, além de outros prêmios, sendo uma ferramenta amplamente difundida no gerenciamento de TI (JUNIOR, 2016).

1.1 QUESTÃO DE PESQUISA

De um lado, confiança e robustez do software de gerenciamento de redes Nagios; do outro lado, a enorme diversidade de equipamento IoT, cada dispositivo com plataforma de gerenciamento individual, diferente e não integradas. Como, dentro deste panorama, realizar a integração do software Nagios com dispositivos IoT?

1.2 OBJETIVO

O objetivo geral desse trabalho é integrar o software de gerenciamento de redes Nagios com dispositivos IoT via SNMP.

1.2.1 Objetivos específicos

1. Investigar a capacidade de gerenciamento fornecido pelo Nagios em dispositivos IoT.
2. Avaliar a utilização do Nagios no gerenciamento de dispositivos IoT.
3. Avaliar a integração de protocolos.

1.3 METODOLOGIA

A metodologia utilizada neste trabalho consiste em um estudo teórico, abrangente e exploratório da literatura disponível sobre o tema a ser estudado. Essa metodologia permite que a análise do tema e o desenvolvimento do conhecimento teórico sobre o assunto permita um melhor entendimento sobre as variáveis do problema proposto, na construção do resultado esperado. Entre as principais atividades estão o estudo de conceitos de gerenciamento de redes de arquitetura TCP/IP; a instalação e compreensão do software Nagios; testes do software Nagios para verificar como deve ser realizada a integração do software com outros dispositivos; a escolha do dispositivo IoT que será utilizado na integração com o Nagios; o estudo da categoria de dados que dispositivo IoT escolhido irá fornecer (MIB) e como realizar a integração deste dado com o Nagios; testes de comunicação entre o Nagios e o dispositivo IoT, visando eficiência de gerenciamento do dado obtido, podendo manipular o dado para que se possa observar estas informações em *real time.*

1.4 ESTRUTURA DO TRABALHO

Para facilitar o entendimento, no capítulo 2 são abordados os conceitos de gerenciamento de rede TCP/IP como também tópicos sobre a arquitetura. Neste capítulo também são abordados são abordados conceitos de agentes e gerentes envolvendo o protocolo SNMP, a sintaxe OiD e as tabelas MIBs. No capítulo seguinte são abordados os conceitos relativos à dispositivos IoT, e uma uma análise do estado da arte relativo as arquiteturas dominantes, como também as formas de gerenciamento delas. O capítulo 4 trata da proposta do trabalho, dos recursos que serão usados, dividindo basicamente em três pontos cruciais: o dispositivo IoT escolhido, o Nagios e a integração deles através do protocolo SNMP. O capítulo 5 demonstra os resultados dos testes relativos a conexão do dispositivo com o Nagios. O capítulo final apresenta um resumo analítico do objetivo proposto.

2 GERENCIAMENTO DE REDES

Conforme Saito (2001), o gerenciamento de rede é um composto de mecanismos operacionais e administrativos necessários para controlar os recursos da rede, manter operacionais os recursos da rede, facilitar o aumento da rede, gerenciar os recursos e controlar o acesso à rede. Kurose (2013) faz um paralelo com situações do mundo real que exemplificam bem o conceito de gerenciamento de redes, tomando como exemplo uma cabine de um avião, onde o piloto tem inúmeros instrumentos para monitorar e controlar o funcionamento da aeronave. Nesta cabine, ele pode checar o combustível, a altitude e a velocidade, entre outras variáveis. Com estas informações ao seu alcance, o piloto “administra” o avião, analisando os dados obtidos remotamente dentro do *cockpit*. Caso alguma variável não esteja de acordo, ele pode tomar decisões corretivas, como alterar a rota ou mesmo usar a rotação mais baixa para economizar combustível. De uma maneira análoga, o gerenciamento de redes baseia-se em controlar os recursos de rede, analisar os seus resultados e tomar ações para correção ou antecipação de falhas. Resumindo, o gerenciamento de redes consiste num conjunto de ações e procedimentos necessários para manter uma rede plenamente funcional.

Zapparoli (2006) define o gerenciamento de redes como a administração de recursos materiais (roteadores, *switches*, modem etc.) e ou lógicos (protocolos), fisicamente distribuídos na rede, assegurando credibilidade/confiabilidade, tempos de resposta (*response times*) aceitáveis e segurança nas informações. As atividades essenciais do gerenciamento de redes consistem

– na detecção e correção de falhas em um tempo mínimo possível;
– no estabelecimento de procedimentos para a previsão de futuros problemas.

O modelo clássico de gerenciamento de redes pode ser resumido basicamente em três etapas (ZAPPAROLI, 2006).

1. Coleta de dados: processo geralmente automático, que consiste na monito-

ração dos recursos geridos;

2. Diagnóstico: compõe o tratamento e análise dos dados coletados. O computador gerente executa uma sucessão de procedimentos (por intermédio de um operador ou não), com a intenção de definir a causa do problema apresentado no recurso gerenciado;

3. Ação ou controle: quando localizada a falha, cabe uma ação sobre ele, com o propósito de solucionar, ou o controle do mesmo, caso a falha não possa ser solucionada de imediato.

Atualmente existem inúmeras ferramentas que auxiliam o administrador de rede na tarefa de gerenciamento, conforme Zapparolli (2006). Estas ferramentas podem ser divididas em quatro categorias:

1. Ferramentas de nível físico: promovem a detecção de falhas em cabos e conexões de hardware;

2. Monitores de rede: efetuam o controle de tráfego;

3. Analisadores de rede: auxiliam no rastreamento e correção das falhas em redes;

4. Sistemas de gerenciamento de rede: permitem a monitoração e controle de uma rede inteira a partir de um ponto central.

2.1 ÁREAS DE GERENCIAMENTO

A ISO 7498-4 (International Organization for Standardization) determina que a gerência de redes pode ser classificada em cinco áreas: gerenciamento de falhas, gerenciamento de configuração, gerenciamento de contabilidade, gerenciamento de desempenho e gerenciamento de segurança. Esta classificação pode ser também referenciada como FCAPS (Fault, Configuration, Accounting, Performance and Security).

Gerenciamento de falhas: para preservar uma rede de computadores de maneira operacional, o administrador necessita cuidar do sistema como um todo e

cada componente dele, para que estejam funcionando dentro da normalidade. Caso ocorra uma falha, o administrador necessita tomar as seguintes providências:

– Determinar onde está o problema; isolar o restante da rede para assegurar que ela continue operando normalmente;

– Ajustar as configurações de rede para minimizar o impacto causado pelo problema.

Uma observação pertinente é que falha não é a mesma coisa que erro. Falha é uma condição anormal que exige a ação de gerenciamento, enquanto que erro é um evento único isolado. Uma falha é provocada pela incapacidade de operar normalmente ou por uma abundante quantidade de erros (STALLINGS, 1999).

Gerenciamento de configuração: tem como finalidade monitorar a configuração da rede e dos dispositivos que fazem parte dela, de maneira que os diversos elementos que compõem o hardware e o software sejam gerenciáveis e rastreáveis; normalmente, este trabalho resulta em um banco de dados que converge a informação topológica de todos sistemas da organização. O gerenciamento de configuração tange tudo que se refere à inicialização da rede e ao desligamento dos recursos dela. O gerenciamento de configuração também se dedica em preservar, acrescer e atualizar as ligações lógicas e físicas entre os diversos dispositivos, bem como o estado dos mesmos durante a operação da rede.

Gerenciamento de contabilidade: o objetivo do gerenciamento de contabilidade é garantir que todos os recursos da rede sejam usados de maneira aceitáveis, por todos os grupos ou pessoas que os acessam. Esta regulamentação otimiza a rede por meio da distribuição dos recursos da rede segundo sua capacidade de atendimento. À medida que usuários fazem uso da rede de maneira ineficiente, o administrador pode executar procedimentos administrativos que melhorem o desempenho, garantindo ou retirando privilégios dos usuários ou outras medidas administrativas.

Gerenciamento de desempenho: tem como objetivo medir e reportar diversos indicadores do desempenho da rede e de sistemas. Os inúmeros componentes de uma rede devem comunicar-se entre si, compartilhando dados e recursos. Quando existem serviços críticos como videoconferências ou VoIP, a rede precisa disponibilizar desempenho dentro de certos limites para que a comunicação flua em sua

normalidade. Conforme Stallings (1999), o gerenciamento de desempenho abrange duas categorias funcionais:

– Monitoramento: é a função de observar as atividades da rede;

– Controle: é a função que permite que se façam ajustes para melhorar o desempenho;

Alguns dos problemas que envolvem o desempenho, com os quais o administrador deve ficar atento:

– Qual a capacidade da rede quando se fala em desempenho?

– Existe algum gargalo?

– A vazão tem diminuído para graus inaceitáveis?

– O tráfego atual é excessivo?

Gerenciamento de segurança: tem como objetivo principal a preservação das informações e o controle de acesso. Senhas e outras informações de permissão devem ser guardadas e compartilhadas. Da mesma maneira deve-se monitorar e controlar o ingresso/acesso aos computadores da rede, assim como coletar e inspecionar arquivos logs e registros de auditoria, bem como habilitar ou desabilitar esses registros. Desempenha também o papel de detectar e prevenir ataques internos ou externos à rede, ataques que podem causar falhas ou causar um estado de ociosidade em algum serviço crucial, ou ainda que crie alguma vulnerabilidade expondo acesso a conteúdo sigiloso armazenado em um sistema da rede. O gerenciamento de segurança é realizado por uma sucessão de ferramentas elaboradas especificamente para desempenhar essa tarefa, tais como:

– sistemas de detecção de intrusão (IDSs);

– sistemas de antivírus;

– *firewall*;

– sistemas de prevenção de intrusão (IPSs);

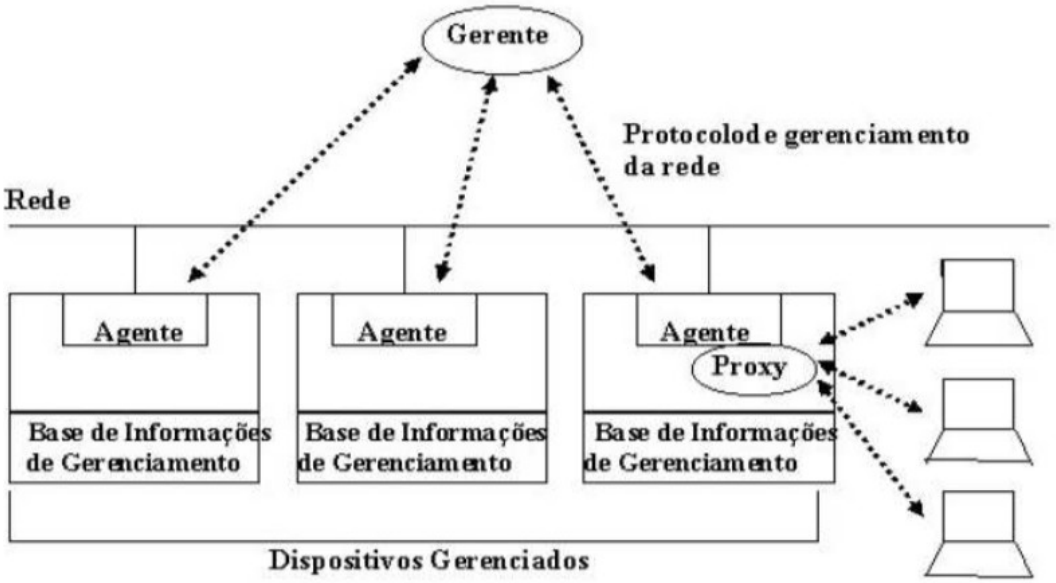
– sistemas de gerenciamento e aplicação de políticas.

O gerenciamento de segurança compreende tanto os sistemas de segurança de rede quanto a segurança física dos servidores e equipamentos-chave de sua infraestrutura, como datacenters, backups etc. Esta segurança de nível físico inclui câmeras e mecanismos de controle de acesso que possam garantir que somente pessoas autorizadas tenha acesso à área e sistemas vulneráveis da rede.

2.2 ARQUITETURA DE GERENCIAMENTO TCP/IP

O modelo geral de gerenciamento de redes TCP/IP é composto pelos seguintes elementos: gerente, agente, base de informação de gerenciamento (MIB) e protocolo de gerenciamento (SNMP).

Figura 1 – Sistema de Gerenciamento de Redes



Fonte: Maria de Fátima Webber do Prado Lima (2018)

O gerenciador atua como uma interface entre o gerente humano (administrador) e a rede. De acordo com Black (2008), o gerenciador é uma plataforma e deve abranger pelo menos quatro itens:

1. Um complexo de aplicações de gerenciamento para análise de dados, monitoramento e acompanhamento de falhas, recuperação de falhas etc.;
2. Uma interface (gráfica ou não) com a qual o administrador de rede possa controlar e monitorar a rede;
3. Capacidade de tradução das solicitações do gerente humano para o sistema gerencial para poder controlar os recursos remotos na rede;
4. Uma base de dados contendo as informações coletadas das MIBs de todos os recursos controlados na rede.

Um processo gerente pode ter a responsabilidade de uma única atividade ou de várias atividades de gerenciamento, transmitindo operações de gerenciamento (*actions*) aos agentes e recebendo notificações (*events*) destes.

Um processo é considerado agente quando uma aplicação executar as instruções enviadas pelo processo gerente. Dessa maneira, esta aplicação passará para o gerente um cenário de todos objetos que estão sendo gerenciados, emitindo notificações sobre os mesmos.

Qualquer dispositivo conectado a uma rede pode ser equipado com um agente. Exemplos: *switches*, estações de trabalho, modems, roteadores e impressoras, podem rodar um agente e, dessa maneira, serem controlados a partir da estação gerencial. Um agente responde às demandas de informações executadas pelo gerente e executa as ações que ele envia, além de poder fornecer informações importantes ao gerente de forma assíncrona, ou seja, ter sido gerada de forma automática pelo agente, sem ser requisitada.

Conforme a RFC 3418 (2002), esta base compõe uma coleção de objetos gerenciados, alocados em um espaço virtual de informações dentro dos agentes, denominados MIBs, elementos da base de Informações, tratados como objetos. O modelo orientado a objetos é um conceito que está relacionado com a ideia de classificar, organizar e abstrair. Um objeto é um dado variável que traduz uma característica do elemento gerenciado. Exemplos do objeto gerenciado: se ele está ativo ou inativo, qual a localização física dele, entre inúmeras variáveis.

Existem ainda categorias de objetos que são padronizadas e podem ser utilizadas para gerenciar equipamentos de diferentes fabricantes. O gerente executa o monitoramento recuperando valores dos objetos coletados da MIB e se comunica com os agentes utilizando um protocolo de gerenciamento de redes TCP/IP. Este protocolo é chamado de Protocolo Simples de Gerenciamento de Rede, ou, em inglês, Simple Network Management Protocol (SNMP).

Além do gerente, agente, SNMP e da MIB, existe também a SMI (Structure of Management Information). De acordo com Esteves (2013), a SMI é um composto de documentos que determinam uma série de regras para a construção de estruturas de gerenciamento. Como exemplo, as sintaxes permitidas para uso e a lista de objetos que integra cada equipamento gerenciado. As seções seguintes detalham todos

componentes envolvidos em uma arquitetura de gerenciamento TCP/IP.

2.3 AGENTE

Conforme Contessa e Polina (2010), o agente é um processo que está sendo executado no host gerenciado (impressora, modem, hub, *switches* etc.) ou próximo a ele, e ele será encarregado pela manutenção de um banco dados local com as informações de controle desse host. Cada host gerenciado por SNMP tem um agente e também uma MIB (base de informações de gerência). Desta maneira, o host gerenciado é visto como um composto de variáveis que caracterizam informações relativas ao seu estado imediato. Essas variáveis ficam acessíveis ao gerente por meios de consultas e podem ser modificadas por ele se assim as variáveis forem definidas. Ao disponibilizar essas variáveis à leitura, o host permite seu monitoramento e, ao receber novos valores do gerente, o host nodo estará sendo controlado. O agente também é encarregado de notificar o gerente no caso da ocorrência de algum comportamento excepcional no host gerenciado. Os hosts gerenciados podem apresentar falhas ou comportamentos inadequados e quando o agente indica que transcorreu algum evento relevante no host, envia pacotes informativos sobre a ocorrência para todas as estações de gerência de sua lista de distribuição de alarmes. Esta operação é efetuada por interrupções (*traps*). Conforme Paiva (2010), *traps* são notificações pré-configuradas que são enviadas pelo agente para o gerente, informando alguma anormalidade. As *traps* informam ao gerente, de maneira assíncrona (sem serem solicitadas), detalhes do que ocorreu de excepcional no host; porém, pode ser necessário que o gerente realize mais consultas para investigação e obtenção de mais detalhes da anormalidade no host.

Contessa e Polina (2010) classificam em dois tipos os agentes SNM. O primeiro tipo é o agente extensível, que oferece suporte apenas à MIB II e utiliza SNMP diretamente (mais detalhes sobre a MIB II podem ser encontradas nas seções seguintes deste trabalho), isto é, possui a implementação de todas funcionalidades do protocolo SNMP, porém não possui suporte a nenhuma MIB, e para que ele responda às requisições de objetos de uma determinada MIB deve haver uma biblioteca adicional que implemente o suporte à MIB. O outro tipo de agente é o

agente estendido, que dispõe somente de funções básicas de comunicação com o dispositivo gerenciado para busca de informações. Este tipo de agente é baseado em um agente principal (extensível), o qual implementa as funções do protocolo SNMP. Dessa forma, o trabalho de resposta às requisições do protocolo SNMP é feito somente pelo agente extensível, ficando para o agente SNMP estendido o trabalho de comunicação com o dispositivo gerenciado e a disponibilização das informações de monitoração ao agente extensível.

Figura 2 – Relação agente X gerente

Fonte: Elaborado pelo autor

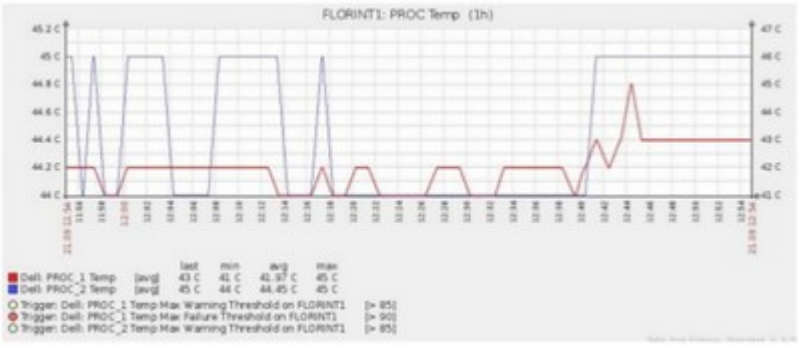
2.4 AGENTE

Na definição de Contessa e Polina (2010), gerente é uma aplicação em execução em uma estação de gerenciamento. Existe a possibilidade de que exista um ou mais gerentes em execução na mesma estação, colaborando entre si para o gerenciamento. Essas aplicações estão aptas a monitorar os agentes através de requisições de informações contidas nas MIBs e a alterar as características dos hosts gerenciados, informando novos valores ao agente. Os gerentes também são os responsáveis pela implementação da política que será adotada na gerência da rede, e eles são acessíveis à pessoa ou entidade responsável pelo gerenciamento do host. O envio de alarmes por correio eletrônico, SMS, chamadas telefônicas, ou outras formas de comunicação com o administrador são comuns nestes tipos de aplicativos gerenciais. Além do envio de alarmes, os gerentes possibilitam a visualização/mensuração das grandezas e estados dos equipamentos, o que é fundamental neste tipo de aplicação. Os gerentes possibilitam (na sua maioria) a apresentação

de gráficos que mostram a evolução de valores ou condições do equipamento ao longo do tempo (dashboards), fornecendo informações sobre tendência de comportamento dos equipamentos.

Na Figura 3, um exemplo de controle de temperatura em CPU.

Figura 3 – Gráfico de temperatura de CPU



Fonte: Oliveira (2010)

Dias e Junior (2002) afirmam que o gerente SNMP é o encarregado pelo monitoramento, pela geração de relatórios e gráficos, e pela tomada de decisões em caso de ocorrência de falhas. O gerente SNMP tem por obrigação a capacidade de interpretar as informações coletadas nos agentes (hosts). Como o protocolo SNMP conta com mais de uma versão, os agentes podem executar versões do protocolo que diferem de dispositivo para dispositivo. Cabe ao gerente de redes da organização padronizar a versão do protocolo utilizada nos dispositivos gerenciados ou então utilizar um software que seja capaz de interpretar todas as versões.

2.5 SMI

Conforme Kurose (2013), a SMI é a linguagem utilizada para definir as informações de gerenciamento presentes em uma host da rede gerenciada. A linguagem é necessária para garantir que a sintaxe e a organização dos dados de gerenciamento sejam padronizados e bem definidos. A SMI tem a função de especificar regras para definir e identificar os tipos e a codificação dos objetos.

De acordo com Gomes (2002), a SMI é um método para descrever objetos

gerenciados, à medida que MIB é a definição (através da sintaxe SMI) dos devidos objetos. Essas variáveis são chamadas de objetos, porém o autor enfatiza para não confundir-se com objetos no sentido de um sistema orientado a objetos. Essas variáveis têm apenas estados e não dispõem de métodos (além da leitura e da escrita de seus valores).

Para Rizo (2011), a SMI é um norma usada para descrever objetos gerenciados e seus relativos comportamentos. Nesse cenário, a MIB pode ser comparada a um banco de dados de objetos gerenciados, no qual são consultados e manipulados por agentes SNMP. O autor entende SMI como um método para definir objetos gerenciados, enquanto que a MIB, é a definição (através da sintaxe SMI) dos objetos gerenciados. A SMI atua como um glossário, que apresenta de que maneira a palavra é pronunciada para, posteriormente, apresentar o conceito desta palavra; já a MIB determina um nome em formato texto de um objeto gerenciado e elucida o seu conceito.

A SMI define três atributos para manipular um objeto:

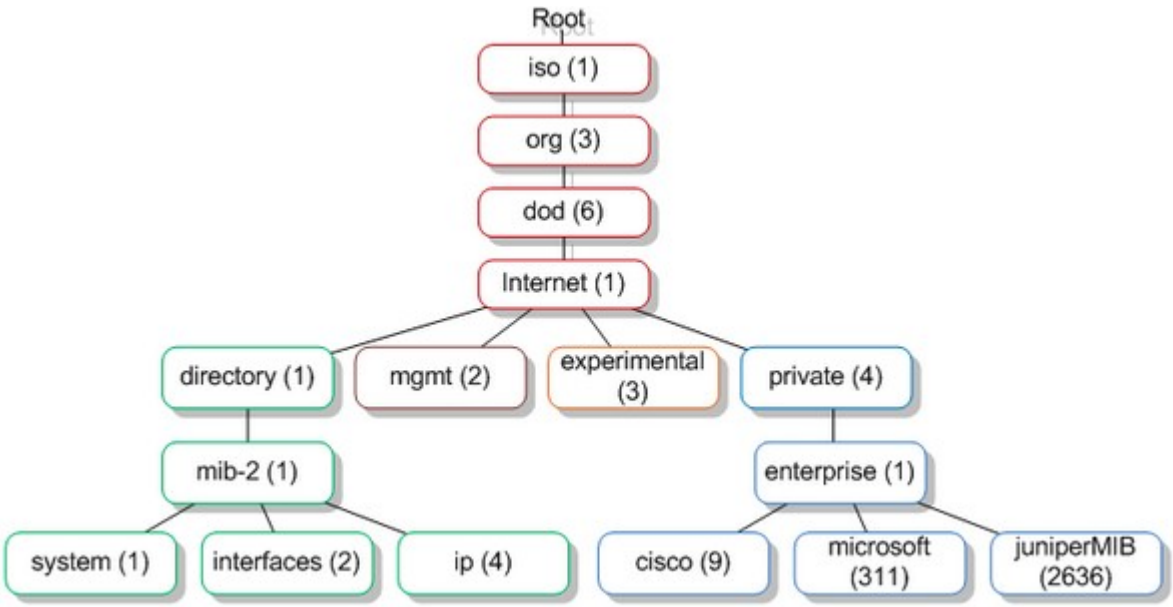
– nome;
– tipos de dados;
– método de codificação.

O nome definido pela SMI deve ser único para cada objeto gerenciado. A nomenclatura é baseada em uma sistema de árvore e utiliza um identificador de objeto hierárquico. O identificador do objeto (OID, Object Identifier) determina um nome exclusivo para um objeto gerenciado. O OID pode apresentar dois formatos: numérico e simbólico.

– Exemplo do numérico: 1.3.6.1.2.1.1.1.0
– Exemplo simbólico: iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0

Os objetos gerenciados são estruturados em um formato de árvore, como na Figura 4.

Figura 4 – Exemplo simbólico de árvore OID



Fonte: https://www.networkmanagementsoftware.com/snmp-tutorial-part-2-rounding-out-the-basics/ (p. 1)

Um ID de objeto é composto por uma coleção de números inteiros baseados nos nós da árvore e limitados pela notação ponto (.). Existe também o formato por símbolos (por nomes, como na Figura 4), que configuram a mesma representação em formato numeral, porém fazendo uso dos nomes das subárvores e folhas.

O nó inicial da árvore (árvore de objetos) é chamado raiz (*root*); os demais nós que tiverem "filhos", são chamados de sub-árvores; e todos os nós que não tiverem "filhos" são chamados de "nós folha".

O tipo de dado de um objeto gerenciado é determinado por um subconjunto da Abstract Syntax Notarion One (ASN.1), que atua especificando a maneira como os dados são representados entre os agentes e gerentes dentro do contexto SNMP. A Tabela 1 demonstra os 11 tipos de dados básicos utilizados pela SMI e definidos na RFC 2578 (http://tools.ietf.org/html/rfc2578).

Tabela 1 – Categoria de dados utilizados pela SMI

Tipo	Tamanho	Descrição
Integer	4 bytes	Valor entre -2^{31}-1
Integer32	4 bytes	Mesmo que Integer
Unsigned32	4 bytes	Valor sem sinal entre 0 e 2^{32}-1
Object String	Variável	String até 65.535 bytes
Object Identifier	Variável	Identificador de objeto
IP Address	4 bytes	Endereço IP, formado por quatro valores inteiros
Counter32	4 bytes	Valor inteiro que pode ser incrementado de 0 a 2^{32}; quando chega ao valor máximo é zerado
Counter64	8 bytes	Contador de 64 bits
Gauge32	4 bytes	Mesmo que Counter32, mas não é zerado automaticamente
Times Ticks		Tempo, medido em centésimos de segundo, transcorrido a partir de algum evento
Bits	4 bytes	String the bits
Opaque	Variável	String não identificada

Fonte: Kurose e Ross (2006)

2.6 ASN.1

Conforme Marty (2003), a ASN.1 é uma linguagem para a descrição dos dados nos padrões de normas de definição. Sua definição é em formato texto não ambíguo, para facilitar a utilização por linguagens de programação. A ASN.1 permite definir modelos de dados para facilitar o uso por vários objetos a um mesmo tipo de definição. A linguagem ASN.1, por se tratar de um formato independente de plataforma e máquina, torna o SNMP viável para lugares onde a demanda gerencial da rede é necessária, não confrontando-se com o sistema utilizado em servidores de rede; como exemplo, outras linguagens de programação. A ASN.1 permite definir modelos de dados para facilitar o uso por vários objetos a um mesmo tipo de definição. Por esse motivo, a implementação dos dados não é considerada ao utilizar-se a

ASN.1.

A ASN.1 foi a linguagem desenvolvida pelo ITU-T e escolhida pela ISO para a definição dos objetos gerenciáveis da MIB. Ela utiliza conceitos de orientação a objeto para definir um recurso, seus atributos e as operações que podem ser executadas por este recurso, quando aplicável.

A notação ASN.1 possui ainda um conjunto de regras denominado BER (Basic Encoding Rules) que define a forma na qual programa escrito nessa ASN.1 é compilado para ser traduzido para a linguagem de máquina do dispositivo de rede. Este programa compilado é então carregado e a MIB passa a ser interpretada corretamente pelo dispositivo.

2.6.1 Sintaxe Básica em ASN.1

– Tipos de dados: Primitivos: Integer, Octel String, Object Idenfier, Null.

– Subtipo de dados: Construtores: Listas e tabelas, Definidos: Nomes alternativos para tipos ASN.1.

A notação utilizada para definir uma ASN.1 utiliza algumas convenções, como apresentadas na Tabela 2.

Tabela 2 – Notação utilizada na definição ASN.1 para seus respectivos tipos.

Item	Convenção	Exemplo
Tipo	inicial maiúscula	DisplayString
valor	inicial minúscula	TRUE
identificador	inicial minúscula	sysDescr
palavras-chave	todas maiúsculas	INTEGER
macros	todas maiúsculas	OBJECT-TYPE
módulos	inicial maiúscula	Oreilly-MIB

Fonte: Mauro e Schimdt (2001)

2.7 MIB

Conforme Kurose e Ross (2006), a MIB pode ser descrita como um banco vir-

tual de informações armazenadas relativas ao objeto gerenciado, informações que em conjunto refletem o estado atual do objeto gerenciado.

Paiva (2010) diz que a MIB é a base de dados acessada pelo protocolo SNMP a fim de obter todas as informações estatísticas possíveis dos equipamentos que estão sendo monitorados, o que possibilita a automatização de grande parte das tarefas de gerência.

Esteves (2013) afirma que a MIB é formada por uma conjunto de objetos que representam a abstração dos recursos de um sistema. Estes objetos podem ser configurados como "permissão de leitura e escrita" ou somente "leitura", dependendo do grau de permissão concedido ao gerente da rede. A permissão de "leitura" autoriza o agente apenas a ler dados da MIB que demonstram o estado atual de um equipamento. A "permissão de leitura e escrita", por sua vez, além da leitura dos dados da MIB, possibilita que o agente realize alterações no objeto gerenciado em tempo real, facilitando uma nova leitura do estado já atualizado. A MIB não contém os dados reais, apenas organiza-os de forma adequada.

Com o crescente aumento do número de equipamentos que podem ser gerenciados, foi criado mais que um tipo de MIB. Esteves (2013) elenca três tipos:

> – MIB II: a MIB II especificada na RFC1213 (http://tools.ietf.org/html/rfc1213), é um aperfeiçoamento da MIB I. A MIB II fornece dados de gerenciamento a respeito e algum equipamento, possibilitando obter estatísticas como, por exemplo, quantidade de pacotes, bytes transmitidos por cada uma das suas interfaces, total de pacotes enviados e recebidos, pacotes com falha, estado das interfaces etc. O objetivo principal da MIB-II é fornecer informações gerais do TCP/IP para a gerência;
> – MIB experimental: como o próprio nome diz, esta MIB está em fase experimental, seus elementos (objetos) estão em período de desenvolvimento e testes. Proporcionam características mais exclusivas sobre a tecnologia nos âmbitos de transmissão e dos dispositivos empregados;
> – MIB privada: seus elementos (objetos) disponibilizam dados mais específicos dos dispositivos gerenciado, como dados do *setup* (configuração) e estatísticas de colisões de pacotes. Com a MIB privada é possivel também efetuar

um *reset* (reinicialização) e/ou desativar alguma porta do roteador. Contessa e Polina (2010) afirmam que a MIB privada pode ser baseada na MIB padrão ou totalmente customizada pelo fabricante do equipamento para o qual foi desenvolvida a MIB. Com esse tipo de MIB é possível até mesmo desabilitar ou reinicializar interfaces do equipamento gerenciado. Com a utilização de MIB privada obtém-se um gerenciamento mais específico, onde a qualidade das informações é superior à qualidade das informações da MIB padrão.

2.7.1 **Estrutura da MIB**

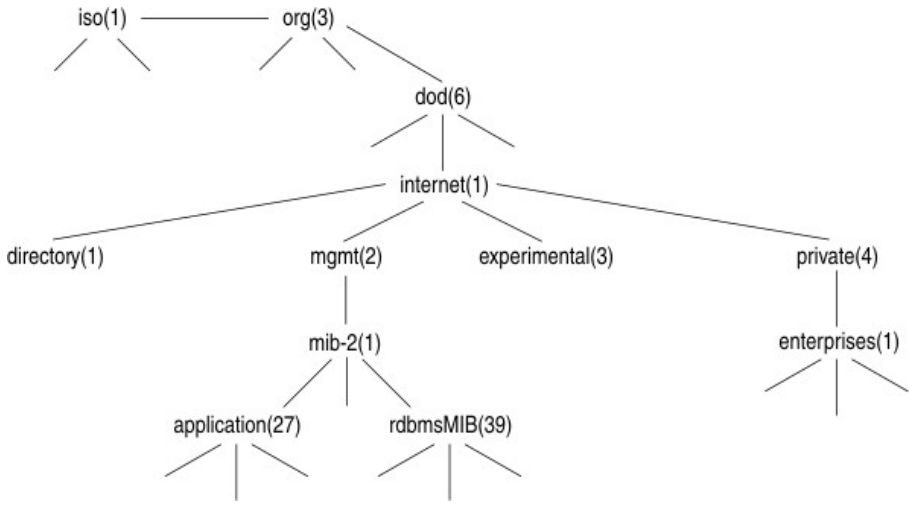
As normas de estruturação da MIB são descritas na SMI (Structure of Management Information). Esta descrição deve considerar a disposição e as condicionantes do equipamento no qual a MIB é provida. Para todo objeto são definidas as subsequentes categorias:

– Object Name: nome do objeto constituido por uma *string* de texto reduzido (*short string*);
– Object Identifier (OID): identificador único do objeto. É composto por números separados por pontos. Exemplo: 1.2.5.6.12989.1.34.4;
– Syntax: sintaxe do objeto. Representa o valor ou formato da informação. Pode ser de um tipo simples como inteiro, OID, *string*; ou pode ser uma sintaxe de aplicação, como, por exemplo, uma medida específica, um contador ou um intervalo de tempo;
– Definição: descrição textual do que é o objeto em questão;
– Acesso: tipo de permissão de acesso concedida. Pode ser somente leitura, leitura e escrita ou sem acesso.

A Figura 5 descreve a organização lógica da MIB. Abaixo do nó .1 fica o nó .1.3, chamado org, que pode ser operado por outras organizações. Abaixo de org situa-se o dod (.1.3.6), pertencente ao United States Department of Defense (Departamento de Defesa dos Estados Unidos), que designou um node para uso na comunidade internet (.1.3.6.1), que é dirgido pela organização IAB (International Activi-

ties Board). Abaixo deste estão os nós directory (.1.3.6.1.1), management (.1.3.6.1.2), experimental (.1.3.6.1.3) e private (.1.3.6.1.4).

Figura 5 – Resumo da hierarquia da MIB



Fonte: https://wiki.sj.ifsc.edu.br/wiki/index.php/Arquivo:Snmp-mib-hierarchy.gif

2.8 MIB II

Abaixo da sub-árvore MIB II estão os objetos utilizados para receber as informações específicas dos dispositivos da rede. A Tabela 3 demonstra os principais grupos de informações presentes na MIB II.

Tabela 3 – Grupo de Informações da MIB II

Grupo	Informação
System (1)	informações básicas do sistema
Interfaces (2)	interfaces de rede
At (3)	tradução de endereços
IP (4)	protocolo IP
ICMP (5)	protocolo ICMP
TCP (6)	protocolo TCP
UDP (7)	protocolo UDP
EGP (8)	protocolo EGP

Transmission (10)	meios de transmissão
Snmp (11)	protocolo SNMP

Fonte: Esteves (2013)

Esteves (2013) define os grupos de objetos da seguinte forma:

1. Sys: informações gerais sobre o sistema, como nome e localização;
2. If: informações sobre as interfaces do dispositivo, incluindo o nome ou número da interface, endereço físico e IP;
3. At: informações sobre a tabela ARP (Address Resolution Protocol);
4. IP: informações relacionadas ao protocolo da Internet, como a tabela de roteamento do dispositivo
5. ICMP: informações relacionadas ao ICMP, como o número de pacotes enviados e recebidos
6. TCP: informações relacionadas ao TCP, como a tabela de conexões;
7. UDP: informações relacionadas ao UDP, como o número de pacotes enviados e recebidos;
8. EGP (Exterior Gateway Protocol): rastreia diversos dados estatísticos sobre o EGP;
9. Transmission: reservado para MIBs específicas de mídia;
10. SNMP: informações relacionadas ao SNMP.

Dentro de cada um dos dez grupos acima listados, existem variáveis que apresentam informações diferentes relacionas ao seu grupo específico. Por exemplo, Kurose e Ross (2006) citam a subdivisão do grupo Sys:

a. sysDescr: completa descrição do sistema (versão, hardware, *O.S*. (sistema operacional));
b. sysObjectID: objeto para identificação do fabricante; sysUpTime: tempo a partir da última reinicialização;
c. sysContact: nome da pessoa de contato;
d. sysName: nome do equipamento gerenciado;
e. sysLocation: localização física do equipamento;

f. sysServices: serviços oferecidos pelo dispositivo.

Na sub-árvore MIB II, cada objeto contém um determinado tipo de dado. Conforme a RFC2578, os tipos de dados considerados pelo SMI são:

a. Integer: normalmente um número inteiro de 32bits. O valor “0” não pode ser empregado para este tipo de dado, senão o objeto não será listado na MIB;
b. *String*: utilizada para especificar textos e deve conter zero ou mais bytes;
c. Counter: é um número de 32 bits, geralmente utilizado para contabilizar quaisquer totais de tráfego de um equipamento, como por exemplo, o total de pacotes recebido e enviados. Esse tipo de dado é crescente, nunca sofrendo decréscimo;
d. Object Identifier (OID): *string* formada por números separados por pontos. Exemplo:1.2.3.6.14988.2.3.6.6.3;
e. Null: representa um objeto que não está sendo utilizado pelo protocolo SNMP;
f. Sequence: utilizado para definir listas de dados. Pode conter zero ou mais tipos;
g. Sequence Of: utilizado para especificar objetos gerenciados formados por dados do tipo *Sequence*;
h. IpAddress: representa endereços de rede do padrão IPv4 (Internet Protocol Version 4). O documento SMI não processa endereços IP segundo o padrão IPv6 (Internet Protocol Version 6), que conta com 128 bits. Na próxima versão do SMI, o SMING (Structure of Management Information Next Generation) estará previsto o tratamento de endereçamento IPv6;
i. Network Address: tem a mesma função do IpAddress, podendo representar tipos de endereços de rede diferentes do IPv4;
j. Gauge: mesmo tipo de dado que o Counter, porém com a diferença de poder aumentar e diminuir seu valor, enquanto o Counter pode apenas aumentar seu valor;
k. TimeTicks: dado numérico de tamanho igual aos dados Counter e Gauge,

geralmente utilizado para medir algum intervalo ou período de tempo, como, por exemplo, o tempo que o equipamento está ligado desde a última reinicialização;

l. Opaque: habilita o armazenamento de codificações do documento ASN.1. em objetos do tipo s*tring;*

m. Unsigned32: exclusivo da versão SMIv2. É utilizado para representar valores numéricos de 0 até 2^{32} -1;

n. Counter64: exclusivo da versão SMIv2. Apresenta o mesmo princípio de funcionamento que o Counter32, porém com 64 bits de tamanho, variando de 0 até 2^{64}-1. Geralmente é utilizado quando números de 32 bits não possibilitam representar valores muito expressivos, que ultrapassem os 32 bits de tamanho limite. Nesses casos é utilizado o Counter64, que possibilita representar valores com até 64 bits;

o. Bits: exclusivo da versão SMIv2. Listagem de bits não negativos de um objeto que está sendo gerenciado.

2.9 SNMP

O protocolo SNMP teve seu início a partir de 1988, como um *feedback* às necessidades de gerenciamento da rede Internet. Conforme Morishita e Moreira (1997), o protocolo foi implementado para ser uma solução provisória, enquanto se esperava o desenvolvimento de um novo protocolo mais completo (CMOT). Esse novo protocolo, o CMOT, foi abandonado por não evoluiur, deixando o SNMP como a única solução de padronização, sendo adotado por todos os fabricantes de equipamento de rede.

É um protocolo pertencente à camada de aplicação da arquitetura TCP/IP e utiliza na camada de transporte os serviços do protocolo UDP para enviar suas informações através da rede IP. Ele é usado em sistemas de gerenciamento de redes a fim de monitorar dispositivos que exijam atenção por parte de seus administradores. De acordo com Soares (2012), o SNMP possui a capacidade de ser aplicado aos mais variados sistemas operacionais, tais como sistemas Unix, Windows, Linux e mac OS, além de impressoras, modems, racks e fontes de alimentação, den-

tre outros, não se restringindo somente a dispositivos físicos, mas com a possibilidade de serem introduzidos em softwares (servidores web e banco de dados, por exemplo).

Morishita e Moreira (1997) dizem que o protocolo SNMP é responsável por transportar as informações entre os agentes e os gerentes, e libera as seguintes classes de operações:

– GET: que permite a obtenção, por parte da estação de gerenciamento, dos valores associados aos objetos da MIB;

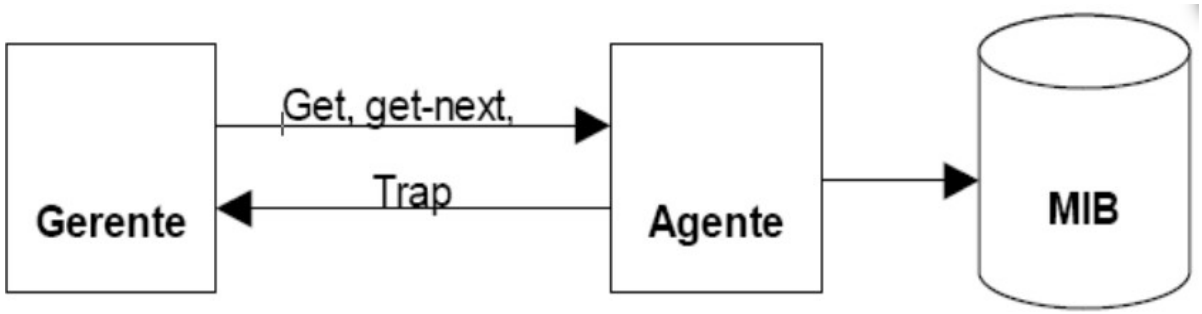
– SET: que possibilita que a estação de gerenciamento efetue alterações dos valores dos objetos em um dado agente;

– TRAP: que autoriza um agente notificar a estação de gerenciamento sobre algum evento importante.

2.9.1 Funcionamento do protocolo

Uma configuração padrão do protocolo SNMP em uma máquina gerencial (Figura 6) apresenta um processo gerente controlando o acesso a uma MIB central, fornecendo uma interface para o gerenciador da rede. O processo gerenciador executa suas funções de gerenciamento de rede sobre UDP parte da classe de protocolos TCP/IP.

Figura 6 – Configuração típica do protocolo SNMP



Fonte: Elaborado pelo autor

Morishita e Moreira (1997) ressaltam uma característica importante do protocolo SNMP, que é a possibilidade de gerenciar elementos de rede que não possuam agentes e elementos que possuam agentes de gerenciamento, mas não trabalhem com o protocolo SNMP. Para realizar este processo ele utiliza os agentes *proxies*, que podem exercer a função de agente para os elementos de rede que não os possuam, ou trabalhar como um *gateway* em nível de aplicação, mapeando as funções de gerenciamento SNMP para as funções de gerenciamento do outro protocolo.

Conforme Esteves (2013), à primeira variante do SNMP foram adicionadas novas funcionalidades, que permitem monitorar cada vez mais recursos dos equipamentos, com maior e mais segurança.

Atualmente existem três versões do protocolo SNMP disponíveis para utilização: SNMPv1, SNMPv2c e SNMPv3.

De acordo com Mauro e Schmidt (2001), a essência do SNMP é uma coleção simples de operações (e das informações adquiridas por essas operações) que permitem ao administrador modificar o estado de alguns dispositivos baseados em SNMP.

2.9.2 SNMP versão 1

Esta é a versão primária do protocolo SNMP e passou a ser utilizada a partir de meados de 1990 É definida em três RFCs e é padronizada pelo IETF.

1. RFC 1155: define a Structure of Management Information (SMI), linguagem utilizada para definir as informações, com o propósito de descrever e nomear os objetos que serão gerenciados;
2. RFC 1212: define a descrição (sintaxe) mais concisa, mas inteiramente consistente com o SMI;
3. RFC 1157: define o Simple Network Management Protocol (SNMP).

Abreu e Pires (2014) consideram fraca a segurança desta versão; as senhas são baseadas em *community strings*, que são simples *passwords*, *strings* em formato texto aberto, que permitem que qualquer ferramenta de gerência que conheça esta *string* obtenha acesso aos dados deste dispositivo. A vulnerabilidade da utiliza-

ção de *community strings* está no fato delas serem enviadas sem criptografia. Dependendo da configuração, através da *community strings* é possível ler e/ou alterar facilmente informações da MIB no agente.

Conforme Santos (2009), para minimizar o risco de captura da informação na *community strings* por usuários não autorizados, é indicada a utilização de *firewalls* para proteger a comunicação SNMP entre os dispositivos e VPNs (Virtual Private Networks) para assegurar a criptografia do tráfego ou então modificar as *communitys* regularmente.

2.9.3 SNMP versão 2

O SNMPv2 acarreta em algumas vantagens, como melhorias na eficiência e no desempenho. Segundo Stallings (1999), após alguns anos de uso da primeira versão do protocolo SNMPv1, algumas deficiências passaram a ser percebidas e as necessidades de melhoria foram identificadas, o que levou ao desenvolvimento da versão do SNMPv2, aprimorada nos quesitos definições de objetos da tabela MIB, procedimentos do protocolo e segurança.

A segunda versão do SNMP é também denominada SNMPv2c (*community string-based SNMPv2*). Ela é definida pelas RFCs 1905, 1906 e 1907 e pelo IETF com o status de experimental. Esta versão além da vantagem no quesito segurança, também teve melhorias nas operações de protocolo com a criação das mensagens InformRequest e GetBulkRequest, que permitem a comunicação entre gerentes, facilitando a gerência descentralizada da rede e também a otimização da recuperação de informações de equipamentos que são monitorados quando necessário, o que tornou possível o gerenciamento distribuído.

2.9.4 SNMP versão 3

O SNMP versão 3 foi criado para suprir uma necessidade de padronização com as variações do SNMPv2, que tentavam criar soluções de segurança para o protocolo. A versão 3 (SNMPv3) é definida nas RFCs 1905, 1906, 1907, 2570, 2571, 2572, 2573, 2574, 2575, 2576 e 2786.

No SNMPv3 ocorreu a inclusão de uma autenticação rigorosa e comunicação privativa entre as entidades gerenciadas. Além das das questões de segurança, o projeto do SNMPv3 também objetivou uma padronização de implementação das entidades (agente/gerente), modularizando suas funcionalidades, o que facilita a evolução de alguns mecanismos do protocolo sem exigir que novas versões sejam lançadas. Outros objetivos eram a manutenção de uma estrutura simples, a facilitação da integração com outras versões e, sempre que possível, o reaproveitamento das especificações existentes. O SNMPv3 incorporou o SMI e o MIB do SNMPv2, assim como também utilizou as mesmas operações do SNMPv2, apenas com uma reescrita da norma para uma compatibilização da nomenclatura.

De acordo com Mauro e Schmidt, a única alteração relevante da terceira versão do protocolo SNMP (SNMPv3) trata os problemas de segurança que se manifestaram nas versões anteriores (SNMPv1 e SNMPv2). A versão não sofreu alterações no protocolo nem sequer recebeu operações novas, e permanece com suporte para todas as operações definidas nas versões anteriores do protocolo.

2.10 CONSIDERAÇÕES DO CAPÍTULO

Com o desenvolvimento e utilização do protocolo SNMP para a gerência de redes, monitorar equipamentos e dispositivos de rede se tornou uma tarefa muito mais simples. Com o acesso aos dados da MIB, é possível coletar informações de diversos recursos presentes no equipamento gerenciado, possibilitando a identificação de falhas nele.

Para executar a coleta e armazenamento dos dados da MIB, é utilizado um software executado em um servidor, denominado gerente SNMP. Além de coletar e armazenar informações, este software possibilita a construção de gráficos baseado nas informações obtidas junto aos agentes SNMP. O software gerente também será responsável por interpretar as informações coletadas dos agentes SNMP e tomar decisões pré-determinadas para cada ocorrência, como enviar alertas para a área encarregada pelo monitoramento toda vez que algum valor obtido junto ao agente ultrapasse algum limite pré-estabelecido na configuração.

3 GERENCIAMENTO DE DISPOSITIVOS IoT (IoT)

Neste capítulo serão abordados conceitos, aplicações, características, tecnologias e arquitetura dos dispositivos IoT existentes no cenário atual. Após esta ênfase teórica será descrito o cenário proposto como objetivo central deste trabalho; posteriormente serão abordadas as características principais e demais informações relevantes ao software Nagios. A escolha do dispositivo IoT para propósitos de testes e integração com o software gerenciador de redes Nagios será a próxima etapa, finalizando com o gerenciamento do dispositivo escolhido para ser gerenciado pelo software Nagios.

3.1 DEFINIÇÃO E APLICAÇÕES

A expressão Internet of Things, IoT (Internet das Coisas), foi utilizada pela primeira vez pelo professor do MIT, Kevin Ashton, em um workshop proferido aos funcionários da empresa Procter & Gamble (P&G) no ano de 1999. O propósito de Ashton, na época, era apresentar à empresa uma solução baseada em transceptores RFID. Na época, por ser tratar de uma nova expressão, IoT foi considerado um conteúdo pouco compreendido (ASHTON, 2009). Hoje, o termo é bastante difundido em artigos científicos e conferências de tecnologia, e ganha importância graças aos avanços tecnológicos em hardware, que permitiram o surgimento de uma grande quantidade de minúsculos computadores que alavancaram a tecnologia (FLEISCH, 2010).

Confrorme Shelby e Bormann (2000), no decorrer dos anos, à medida que as redes de computadores expandiram, mais especificamente por conta da Internet, uma nova revolução emergiu, a Internet das Coisas (Internet of Things), onde milhares de dispositivos embarcados passaram a possuir endereços IP, se tornando parte da Internet. Conforme Wolf (2002), qualquer computador que seja componente de um sistema maior e que se baseia num microprocessador, é considerado um sistema embarcado. O autor define um sistema embarcado como um dispositivo que combina hardware e software usado para executar uma ou várias funções específicas. Um sistema embarcado é constituído por um microcontrolador, mais algum

hardware e software, e age como um dispositivo para um determinado fim ou para mais de um, sendo ele geralmente de baixo custo e de alto desempenho (WU; ZHU; DENG, 2008). Os sistemas embarcados são amplamente utilizados em nosso dia a dia; são TVs, aparelhos de som, centrais de multimídia etc. Em um futuro próximo, os sistemas embarcados se tornarão fundamentais no desenvolvimento das casas inteligentes, indústrias inteligentes e até das cidades inteligentes (WU; ZHU; DENG, 2008). No geral, esses dispositivos possuem recursos limitados em termos de energia, capacidade computacional, memória, largura de banda etc.

O termo IoT eleva a definição de sistemas embarcados para um outro nível. Vasseur e Dunkel (2010) definem dispositivo IoT como um item provido com sensores e atuadores, um microcontrolador, uma porta de comunicação e uma fonte de energia. Os sensores e atuadores permitem ao dispositivo interagir com o mundo físico. O microcontrolador possibilita o processamento e transformação dos dados coletados. A porta de comunicação dá ao objeto inteligente a possibilidade de transmissão dos dados coletados ou de receber dados de outros objetos inteligentes ou gerenciadores de rede. A fonte de energia fornece energia elétrica para que o objeto inteligente possa operar. Conforme Vasseur e Dunkel (2010), a diferença principal entre dispositivos IoT e sistemas embarcados é a porta de comunicação, que não é requerida em um sistema embarcado qualquer.

Figura 7 – Exemplos de sistemas embarcados

Fonte: elaborado pelo autor

Segundo Atzori (2010), IoT consiste na presença de uma diversidade de objetos que interagem entre si com com intuito de de atingirem um objetivo comum. Para que isso aconteça, compartilham informações utilizando métodos de endereçamento unificados e protocolos de comunicação padronizados.

Conforme Fleisch (2010), a IoT baseia-se puramente em interligar objetos de uso cotidiano, do nosso ambiente físico, com a Internet (ambiente virtual), tornando-os, dessa maneira, objetos inteligentes. De acordo com Ashton (2009), a gama de aplicabilidade de dispositivos IoT é enorme, tendo ela se iniciado com aplicações RFID. O RFID é utilizado em grande escala nos dias hoje, como no uso em crachás identificadores, pedágios automáticos em rodovias e gestão de cadeias de suprimentos, entre muitas outras aplicações. Além do RFID, a IoT pode ser encontrada em tecnologias de inteligência artificial e nanotecnologia.

Com esse impacto na vida cotidiana, onde objetos interagem com o ambiente e com outro objetos que os cercam, os efeitos mais visíveis decorrentes dessa tecnologia já são visíveis no ambiente de trabalho e no ambiente doméstico (WU; ZHU; DENG, 2008).

Várias áreas de aplicação IoT são promissoras, como a de transportes, segurança pública, energia, governança, saúde e inúmeras outras. Algumas áreas como a da saúde merecem uma atenção especial, pois além da tecnologia IoT contribuir beneficamente, como no auxílio em situações de emergência, ou mesmo para a longevidade, o impacto da segurança dos dispositivos IoT deve ser levado em consideração, pois a vulnerabilidade da rede pode pôr em risco o usuário ao expor dados não autorizados, já que os sensores destes dispositivos estarão ligados à Internet, considerada um ambiente pouco seguro (ITU, 2005).

Dentro das grande gama de aplicações IoT, citam-se algumas específicas que já fazem parte da realidade, como o despertador inteligente, que faz o ajuste do despertar conforme os horários do transporte público que o usuário utiliza, ou mesmo o display disponível em paradas de trem ou ônibus, que informa o horário de chegada do transporte baseado em informações oriundas de sistemas GPS (MECEWEN; CASSIMALLY, 2013).

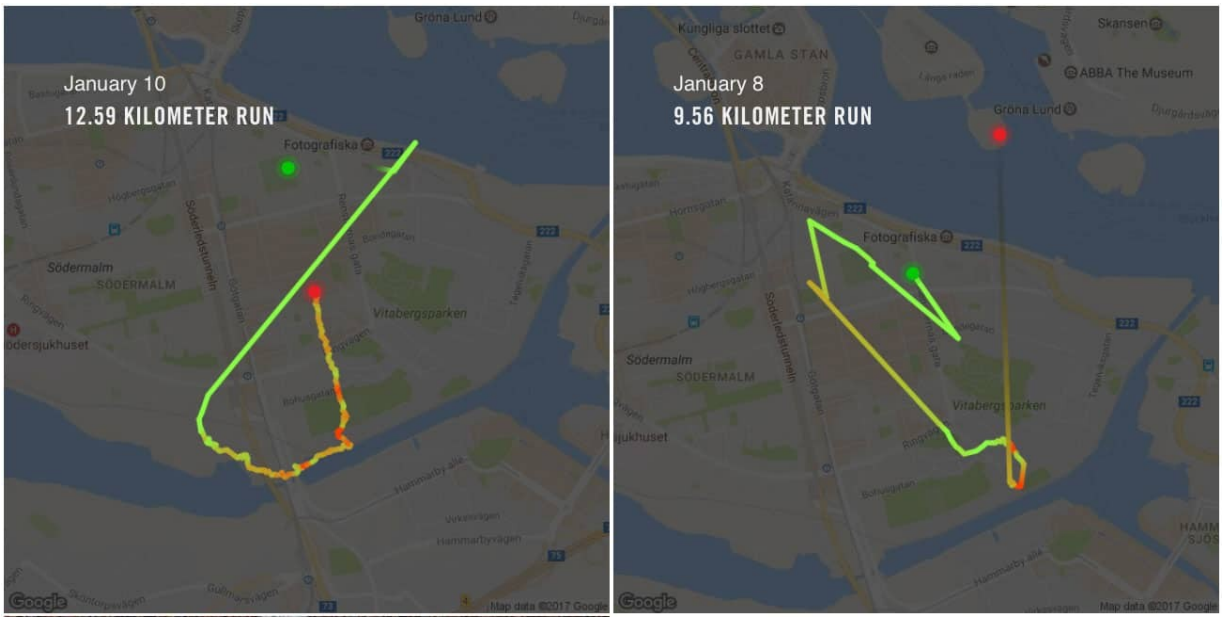
Na área da saúde, um exemplo é o GlowCap®, um dispositivo que avisa ao usuário o horário correto de tomar a medicação e envia um pedido a farmácia local

quando o medicamento está por terminar. Ainda no que diz respeito à área da saúde, existem inúmeros dispositivos vestíveis, conhecidos pelo jargão em inglês *wearable devices,* que, em sua grande maioria, são sensores acoplados ao corpo, seja dentro de casacos ou calçados. Um bom exemplo é o Nike Plus® (Figura 8), um dispositivo que vai dentro do tênis esportivo, tornando possível a coleta de dados como calorias e distância percorrida, entre outros (Figura 9) (MECEWEN; CASSIMALLY, 2013)

Figura 8 – Sensor Nike Plus®

Fonte: Divulgação Nike

Figura 9 – Dados capturados do Nike Plus®

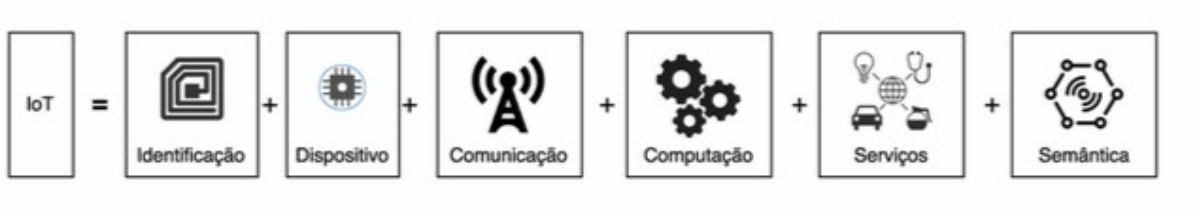


Fonte: Divulgação Nike

3.2 ELEMENTOS IoT

A trajetória e definição da IoT se funde com o progresso das tecnologias de dispositivos, redes, protocolos e virtualização de máquinas (computadores). Desta maneira, os diversos elementos IoT resultam em um sistema que conecta suas partes de processamento objetivando um consumo de dados em *real time.* Desta maneira, Al-Fuqaha (2015) ilustra os principais elementos que compõe uma IoT, conforme a Figura 10.

Figura 10 – Elementos da IoT



Fonte: Al-Fuqaha (2015)

– **Identificação**: é fundamental para reconhecer os objetos de maneira única para conectá-los à Internet. Vários meios são utilizados para identificar, como: RFID, Near Field Communication (NFC) e endereço IP;

– **Comunicação**: em IoT, distintos objetos distribuídos são conectados e entregam serviços específicos. Os dispositivos variados precisam operar com pouca energia e falhas na rede são passíveis de acontecer. Por causa destas características, a escolha da tecnologia aplicada na comunicação é um importante atributo do projeto;

– **Computação**: compõe a parte do processamento dos microcontroladores, (Matriz de Portas Programáveis em Campo), FPGAs e aplicações de softwares que permitam a computação em nuvem *(cloud computing).*

– **Serviços**: os serviços podem ter diversas classificações. Um dos mais relevantes é o serviço de identidade, isto é, toda a aplicação necessita traduzir objetos do mundo real para o mundo virtual, e é necessário identificar essas entidades. O serviço de identidade é usado em todos os tipos de serviços. O serviço de agregação de informação executa a tarefa de coletar e ordenar as informações recebidas pelos dispositivos para que sejam processados e re-

portados na aplicação IoT. Os serviços de colaboração e inteligência agem sobre os serviços de agregação de informação fazendo uso dos dados obtidos para poder tomar decisões e assim reagir de maneira adequada. Os serviços de ubiquidade objetivam prover os serviços de colaboração e inteligência em qualquer momento ou lugar que se façam necessários.

– **Semântica**: aborda a competência de extrair conhecimentos de provedores de serviço. Este conhecimento inclui o descobrimento da informação, o uso de recursos e respectiva modelagem desta informação. Da mesma maneira, inclui o reconhecimento e análise do dado para permitir que as decisões sejam tomadas de maneira correta. É relevante afirmar que a semântica pode ser vista como parte importante para o encaminhamento de demandas a determinados recursos.

3.3 ARQUITETURA DOS DISPOSITIVOS IoT

Nesta seção são abordados conceitos básicos da arquitetura dos dispositivos IoT, que é alicerçada em quatro partes.

– **Processamento e memória**: é composta por algum tipo de armazenamento interno (*storage*), software controlador (*firmware* ou sistema operacional específico), um microcontrolador e um conversor analógico-digital que recebe os dados dos sensores. As CPUs destes dispositivos IoTs geralmente contam com alto poder computacional. Além disso, na maioria das vezes contam com suporte de memória externa do tipo *flash memory* (SD card ou pendrive USB, entre outros tipos). É importante salientar que que os principais requisitos dos componentes em um dispositivos IoT são de baixo consumo de energia e espaço físico reduzidos;

– **Comunicação**: o módulo de comunicação pode ser com fio ou sem; quando ela é *wireless* (sem fio), geralmente se faz uso de rádio de baixa potência, porém a comunicação é de curto alcance e suscetível a perdas constantes;

– Fonte de energia: encarregada pela alimentação do dispositivo, podendo ser de origem elétrica (rede elétrica convencional), baterias, solar ou outras

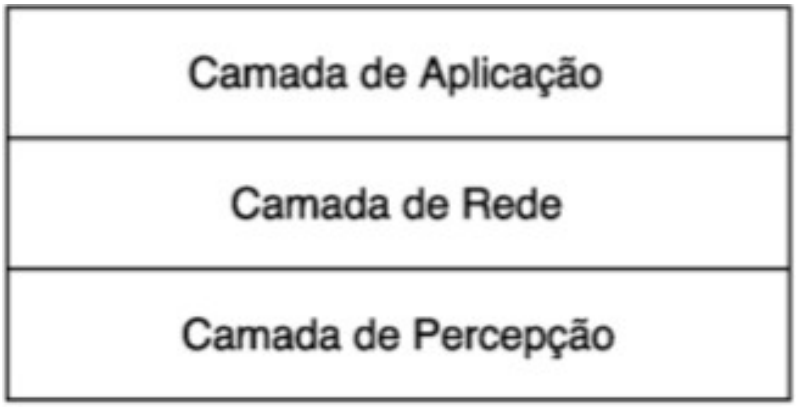
alternativas;

– **Sensores/Atuadores**: os sensores são módulos que efetuam o monitoramento do ambiente, como umidade, temperatura, presença de fumaça e vários outros tipos. Atuadores são dispositivos que atendem a comandos produzindo alguma ação (como trancar uma porta). Podem ser manuais, elétricos ou mecânicos.

Conectar um grande número de dispositivos em rede, garantindo escalabilidade, interoperabilidade, confiabilidade e a qualidade de serviço (QoS), pode representar um grande desafio.

A Figura 11 ilustra o modelo básico de arquitetura base proposta por Wu (2010) com três camadas:

Figura 11 – Arquitetura de três camadas para IoT.

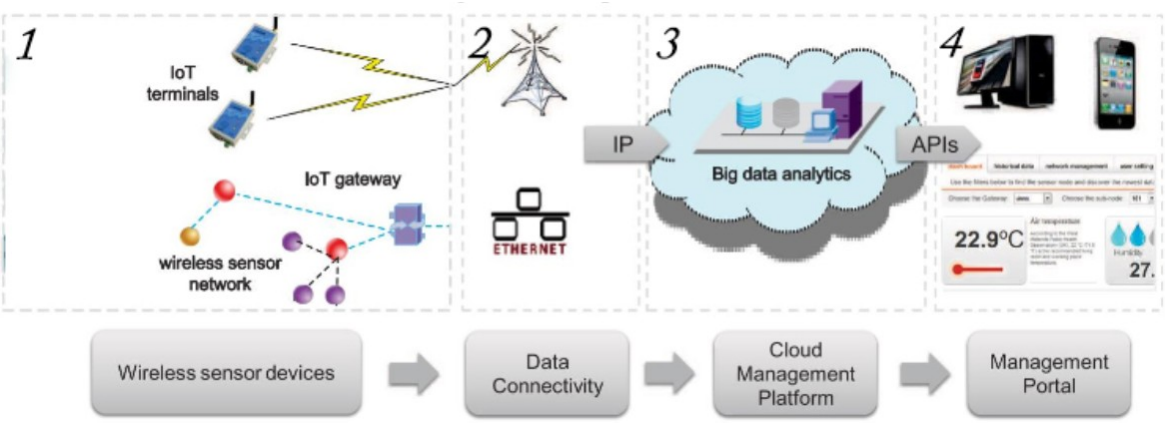


Fonte: WU (2010)

A camada de aplicação é encarregada pelo processamento das informações recebidas e pelo fornecimento dos serviços para os clientes ou outras aplicações. A camada de rede, também chamada de camada de transmissão, efetua a transmissão da informação obtida para a camada de aplicação, atuando como um meio de transporte de dados, e visa abstrair as tecnologias de comunicações, protocolos e serviços de gerenciamento relacionados à ela. A camada de percepção é composta por objetos físicos que, por meio de sensores, coletam e transmitem informações para camada de rede. Esta camada trabalha com a identificação e coleções de objetos de informações específicas através dos sensores.

Sheng (2015) propõe uma arquitetura IoT com destaques para os seguintes elementos (Figura 12):

Figura 12 – Arquitetura IoT segundo Sheng (2015)



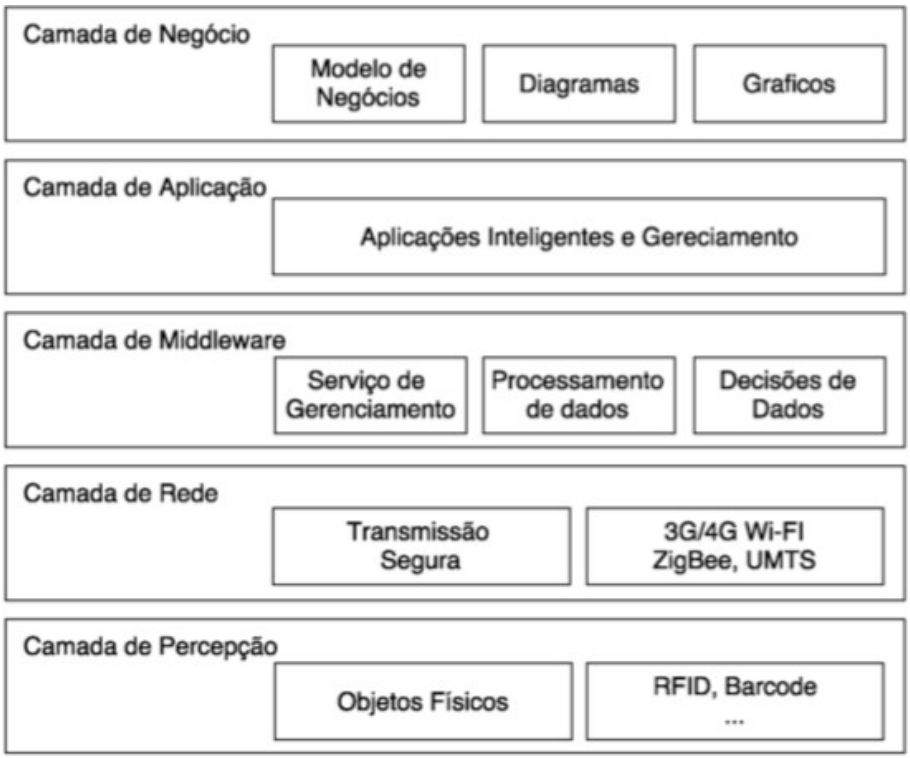
Fonte: Sheng (2015)

1. Dispositivos sensores: formam a base da arquitetura IoT, têm a função de captura de informações de diferentes objetos e enviam a informação capturada para o próximo nível, através de *gateways*, redes *wireless*, cabeadas ou mistas.
2. A conectividade de dados funciona como um *gateway* que traduz os dados capturados anteriormente para um formato padrão, e os envia para a plataforma de gerenciamento a seguir.
3. A plataforma de gerenciamento em nuvem forma o núcleo da IoT, provendo um conjunto comum de operações, como gerenciamento de dispositivos, conversão de protocolo e redirecionamento de rotas.
4. Portal de gerenciamento: ferramenta que permite visualizar os dados capturados, podendo contemplar as funções de gerenciamento e visualização em APIs das variáveis obtidas dos objetos inteligentes.

É possível observar na arquitetura proposta por Sheng (2015) que o gerenciamento em IoT compreende todo o tratamento das informações obtidas dos objetos inteligentes, sejam elas relacionadas à manutenção da rede e também relacionadas aos dados de sensoriamento.

Além das arquiteturas propostas por Sheng e Wu, Khan (2012) propõe uma arquitetura IoT dividida em cinco camadas, como pode ser visto na Figura 13, na página seguinte. A Camada de Negócios é encarregada pelo gerenciamento de todo sistema IoT no que diz respeito a aplicações e serviços. Com isso, é produzido um modelo de negócio, gráficos e digramas baseados nos dados recebidos da Camada de Aplicação.

Figura 13 – Arquitetura de cinco camadas para IoT



Fonte: Khan (2012)

A Camada de Aplicação recupera os dados processados pelo *middleware* para serem empregados em determinada aplicação. A Camada Middleware é responsável por implementar diversos tipos de serviços relacionados a IoT. Cada dispositivo conecta e comunica com apenas outros diapositivos que possuem o mesmo tipo de serviço. Esta camada tem várias funções, como oferecer serviços para aplicações de forma simples, lidar com grande quantidade de eventos, e integrar sistemas e comunicação heterogênea. Khan (2012) ressalta a dificuldade de conectar uma grande quantidade de dispositivos. Assim, entre a Camada de Percepção e a Camada de Rede é mandatória a camada *gateway*, componente que proporciona o ponto de ligação de comunicação. A Camada de Percepção e a

Camada de Rede possuem as mesmas características descritas anteriormente; já as camadas diferentes são a Middleware e a Camada de Negócios. Desta forma, a partir das análises dos resultados, esta camada irá ajudar a determinar as ações futuras e traçar as estratégias.

3.4 TECNOLOGIAS E PROTOCOLOS

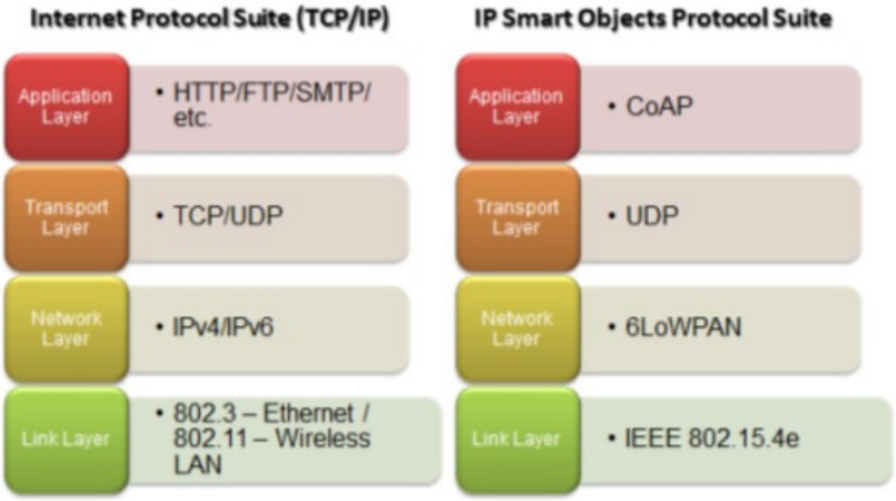
Esta seção detalha os protocolos e padrões CoAP e LoWPAN, utilizados neste trabalho. Estes protocolos de mensagens são utilizados a partir do dispositivo até um *gateway* de IoT.

3.4.1 CoAP

Constrained Application Protocol (CoAP) é um protocolo web apresentado pela Internet Engeneering Task Force (IETF), projetado para atender a requisitos de ambientes de redes com recursos limitados, como as redes de sistemas embarcados. O CoAP suporta comunicações confiáveis e não confiáveis baseado no paradigma requisição/resposta, adicionalmente suportando notificações assíncronas sobre o protocolo UDP (User Datagram Protocol) (PAVENTHAN et al., 2013).

O CoAP se tornou uma das alternativas mais utilizadas como protocolo de Camada de Aplicação para soluções 6LoWPAN. Grande parte das aplicações baseadas no modelo pilha 6LoWPAN se apoia ou é construída sobre o protocolo CoAP. Shelby e Bormann (2009) definem o CoAP como um protocolo leve, de baixo consumo de energia e rápido de aplicação. O protocolo é semelhante ao HTTP (atuando na Camada de Aplicação), exceto que o CoAP usa utiliza a camada de transporte via UDP em vez de TCP. (Figura 14)

Figura 14 – TCP/IP Stack and Smart Objects Protocol Stack



Fonte: o Autor (2018)

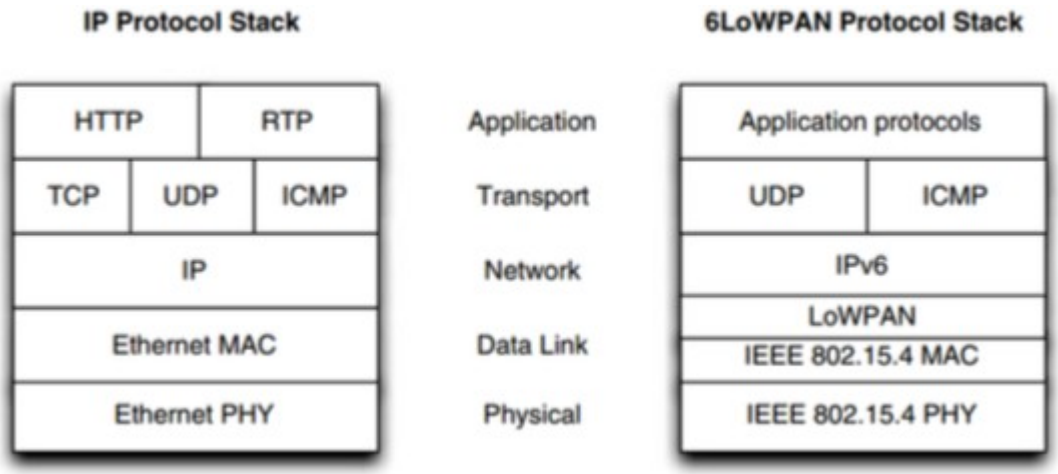
3.4.2 6LoWPAN

É uma coleção de padrões que possibilita o uso eficiente do IPv6 sobre IoTs com poucos recursos de energia e de comunicação (por uma Camada de Aplicação) e, consequentemente, a otimização dos protocolos relacionados. O termo LoWPAN, Low-power Wireless Personal Area Network, é oriundo do padrão IEEE 802.15.4, lançado em 2003 como o primeiro padrão global de rádio de baixa potência. Logo após aquele ano, a ZigBee Alliance desenvolveu uma solução de rede *ad-hoc* sobre o padrão IEEE 802.15.4, que se chamou ZigBee e se tornou um padrão popular no mercado de redes de dispositivos IoTs (SHELBY; BORMANN, 2009).

A arquitetura 6LoWPAN integra ilhas de redes de dispositivos embarcados sem fio onde cada ilha é uma rede LoWPAN. As redes LoWPAN podem ser, simples, estendida e *ad-hoc*. Uma LoWPAN é conectada à Internet por meio de um *edge router* (roteador de borda). Esse equipamento precisa "conhecer" as duas tecnologias e tratar o tráfego que entra e sai das LoWPANs para a Internet, portanto, precisa suportar parte das duas pilhas de protocolo (SHELBY; BORMANN, 2009).

A Figura 15 mostra a comparação entre as *stacks* (estruturas de dados modelo pilhas) no TCP/IP e as 6LoWPAN.

Figura 15 – Comparação entre pilhas IP e 6LoWPAN



Fonte: SHELBY; BORMANN (2009)

3.5 GERENCIAMENTO DE REDES IoT

Para executar a coleta e armazenamento dos dados da MIB, é utilizado um software executado em um servidor, denominado gerente SNMP. Além de coletar e armazenar informações, este software possibilita a construção de gráficos baseado nas informações obtidas junto aos agentes SNMP. O software gerente também será responsável por interpretar as informações coletadas dos agentes SNMP e tomar decisões pré-determinadas para cada ocorrência, como enviar alertas para a área responsável pelo monitoramento toda vez que algum valor obtido junto ao agente ultrapasse algum limite pré-estabelecido na configuração. Conforme Lima, Fresse e Rousseau (2014), as redes computacionais sempre foram marcadas pela complexidade, diversidade e crescimento. Esses fatores dificultam a manutenção desses ambientes computacionais. Além disso, as redes se tornaram fatores cruciais nos modernos de negócio. Portanto, monitorar e garantir sua confiabilidade e desempenho é absolutamente fundamental.

De acordo com Lima, Fresse e Rousseau (2014) quanto maior o tamanho de uma rede de computadores, maior é a complexidade da rede e por isso existe a dificuldade do gerenciamento de forma manual. A adoção de um software de gerenciamento é um passo essencial, visto que a garantia de estabilidade e desempenho são fatores determinantes para um ambiente de redes otimizado e funcional.

Uma maneira de manter uma rede de computadores em bom funcionamento é o uso de soluções que permitam o gerenciamento dos elementos dessas redes. O gerenciamento de redes agrega todas as ferramentas, procedimentos, métodos e atividades para executar esta manutenção e provisionamento de elementos em atividade na rede. As redes IoTs também trazem dificuldades semelhantes às redes comuns. Além disso, dispositivos inteligentes costumam ter diversas limitações de recursos e acrescentam ainda mais complexidade à tarefa de gerenciamento dessas redes. Para a manutenção deste dispositivos sensores, por exemplo, monitorando seu desempenho ou enviando comandos ao nó sensor, é essencial que se use um protocolo de comunicação eficiente e que não consuma recursos consideráveis (SHENG et al., 2015).

Integrar dispositivos IoT por meio de TCP/IP pode trazer inúmeros benefícios, como enumeram Shelby e Bormann (2009):

– Tecnologias baseadas no protocolo IP existem há anos e já provaram ser plenamente escaláveis;
– A tecnologia IP é uma especificação aberta;
– Ferramentas para gerenciamento e diagnóstico de redes IP já existem e poderiam ser usadas;
– Dispositivos IP poderiam ser conectados facilmente com outras redes IP;
– Novas redes IP poderiam aproveitar infraestruturas já existentes.

3.6 CONSIDERAÇÕES DO CAPÍTULO

Tecnologias associadas a IoTs estão em pleno desenvolvimento, por isso nenhuma delas pode ser considerada como uma tendência. Como citado no capítulo, as questões cruciais para o desenvolvimento de IoT esbarram no suprimento de energia, no protocolo de comunicação e na arquitetura da rede. A questão de processamento já não é um fator determinante, pois a velocidade que os processadores se desenvolvem nos fatores tamanho físico e capacidade de processamento de dados é muito maior que os fatores ligados à energia e o desenvolvimento de um protocolo padrão. O protocolo TCP/IP que se expande para o endereçamento IPV6 indica ser o mais promissor em termos de protocolo de comunicação.

4 PROPOSTA DE SOLUÇÃO

O número de dispositivos microprocessados conectados ao nosso cotidiano e que estão interligados à Internet teve um crescimento exponencial. Neste cenário, a heterogeneidade nos protocolos de gerenciamento destes dispositivos faz parte do contexto. Como não existe uma arquitetura padrão para a construção destes dispositivos IoTs com propósitos diversos, criou-se uma lacuna de como gerencia-los, já que cada um deles utiliza uma maneira de gerenciamento singular e poucos se encaminham para uso de algum protocolo já pré-estabelecido no mercado. Apostando nessa lacuna de qual protocolo usar para gerenciar dispositivos IoTs, foi usado para fins de testes o protocolo SNMP, um protocolo que existe desde 1988, e que já na sua terceira versão pode ser considerado um protocolo consolidado nos quesitos de simplicidade de uso e segurança.

A proposta deste trabalho é alicerçada em três aspectos principais: um protocolo de gerenciamento consolidado (SNMP), um dispositivo a ser gerenciado (agente IoT) e um software gerenciador apto a gerenciar dispositivos com suporte SNMP (gerente SNMP). Como resultado da união destes três itens, dispõe-se de um sistema capacitado para gerenciar informações captadas do mundo real, com o objetivo de informar o usuário em tempo real sobre qualquer eventual mudança nas variáveis lidas captadas pelo dispositivo IoT. Dentro deste panorama, o dispositivo IoT escolhido para propósitos de teste foi um termômetro ambiental com suporte a SNMP e o software destinado ao gerenciamento foi o Nagios.

4.1 DISPOSITIVO A SER GERENCIADO

Dentro da enorme gama de dispositivos IoT presentes, cada um com uma finalidade específica, foi escolhido um dispositivo que tivesse como uma de suas características o uso do protocolo SNMP como uma de suas formas de gerenciamento. O dispositivo escolhido foi o STE2, dispositivo fabricado na República Tcheca pela empresa HW Group.

O STE2 é um dispositivo com suporte a SNMP que disponibiliza quatro conexões para sensores, dois do tipo "contato seco", voltado para sensores digitais, que

tem a finalidade de informar ao hardware o valor 1 para aberto e 0 para fechado. Este tipo de conexão pode ser usada com detectores de fumaça e com sistemas de fechamento de portas, entre outros, em que a resposta é digital (1 ou 0). A outras duas conexões são capazes de medir a temperatura e a umidade, uma vez que a informação passada pelo sensor ao hardware, não é apenas um bit, e sim um valor numérico (no caso é um integer). Se forem utilizados sensores de umidade e temperatura (sensores que acompanham o produto) pode-se estabelecer um intervalo de valores pré-definidos que, quando excedidos, o STE2 envia alarmes através do próprio sistema (software próprio baseado em interface web) ou através de mensagens via protocolo SNMP.

A comunicação do dispositivo STE2 pode ser feita via Ethernet ou WiFi ou por ambas, característica que permite ao usuário uma conexão redundante, onde um endereço assume a comunicação caso o outro apresente falha. Neste trabalho será feita a conexão ao dispositivo via Ethernet.

Figura 16 – Dispositivo STE2

Fonte: HW Group (fabricante)

Principais características do STE2:

– Conexão Ethernet e Wi-Fi padrão 802.11 b / g / n (2.4GHz);

– Suporta operação simultânea de Ethernet e Wi-Fi;

– Fonte de alimentação de 5V;

– Instalação simples com suporte a DHCP;

– Servidor WEB embutido – sem necessidade de software além de um *browser;*

– Conectividade NMS (SNMP e MIB) e XML;

– Sporte a autenticação TLS (Gmail);

– Sistema protegido com senha;

– Envio automático de alertas (email e sms) em condições de alta e baixa temperaturas;

– Suporte a exportação de dados para Excel®;

– Suporte SNMP para versões 1 e 2;

– Conexão de sensores através de duas porta RJ11 (padrão telefônico);

– Duas esperas para sensores digitais, como detector de fumaça, sensor de fechamento/abertura de portas, entre outros.

Principais utilizações do STE2:

– Alarme para falha de ar condicionado: mudanças na temperatura apontam para uma falha de uma unidade de ar condicionado.

– Alarme para falhas em sistemas de refrigeração: mudanças na temperatura apontam para uma falha de uma unidade de refrigeração.

– Alarme para falhas em sistemas de laboratório: mudanças na umidade apontam para o excesso ou carência de umidade.

(Nas páginas 94 e 100, respectivamente anexos A e B, constam a tabela MIB e as OIDs fornecidas pelo equipamento STE2.)

De acordo com a proposta de gerenciamento de um dispositivo IoT, será feito o uso de um equipamento para capturar informações de temperatura e umidade dos seus sensores, manipulando a informação de posse dos conhecimentos adquiridos com o estudo do protocolo SNMP e dispondo estas informações pelo software Nagi-

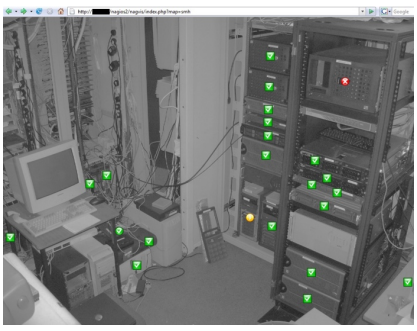
os.

O dispositivo será conectado junto a uma rede corporativa, dentro de um ambiente propício para alteração de valores de temperatura e umidade – uma sala que dispõe de sistema de ar-condicionado, que pode ser ligado e desligado para ocasionar alteração nos valores, a serem medidos pelo equipamento STE2 – e desta forma simular o comportamento das variáveis e como esta informação chega ao usuário final.

4.2 NAGIOS

O software Nagios, é um sistema de monitoramento de redes de código aberto, distribuída sob a licença GPL, que permite monitorar *hosts* e serviços, alertando o usuário (gestor da rede) quando existe a ocorrência de problemas nos *hosts* ou serviços e também quando estes problemas são resolvidos. O Nagios nasceu com o nome de NetSaint e foi mantido com este nome até meados de 2002. Originalmente o Nagios possui poucos recursos nativos para gerenciamento, entretanto ele é uma ferramenta dinâmica e possui suporte aos mais diversos tipos de *plugins.* Podem-se citar alguns como o Centreon[1], que é um *plugin* comercial que oferece uma interface onde é se possível instalar e implantar facilmente e com eficácia um monitoramento de desempenho. Outro exemplo relevante de é o NagVis, que é um *pluigin* distribuido como *freeware* que possibilita a elaboração de mapas gráficos da rede (Figura 17).

Figura 17 – Plugin NagVis, que permite a integração dos dados Nagios, e uma figura de um CPD



Fonte http://www.nagvis.org/screenshots

[1] https://www.centreon.com/

De acordo com Becker e Moura (2012), os *plugins* são programas intermediários que agem entre o Nagios e os dispositivos a serem monitorados, sempre visando agregar funções de monitoramento ao software gerente.

Os sinais de alerta gerados pelo Nagios em resposta a uma desconformidade nos serviços ou hosts monitorados, podem ser enviados por email, sms, ligação telefônica até mesmo via whatsapp para a pessoa ou área responsável, informando o nome do dispositivo que apresenta alguma variação nos parâmetros pré-definidos. Por exemplo, um roteador configurado para trabalhar dentro de uma faixa de temperatura, caso saia desta faixa pré-estabelecida, o responsável será alertado pelo Nagios.

Conforme o site do fabricante[2], o software Nagios fornece monitoramento de todos os componentes de infra-estrutura de rede considerados críticos, incluindo aplicativos, serviços, sistemas operacionais, protocolos de rede, métricas de sistemas, bancos de dados e demais itens que fazem parte da infra-estrutura de rede. Conforme o site do fabricante, centenas de plugins de empresas parceiras fornecem o monitoramento de praticamente todos os aplicativos, serviços e sistemas internos. O software Nagios gera gráficos que permitem às organizações planejarem atualizações de infraestrutura antes que os sistemas desatualizados os apanhem de surpresa. Também de acordo com site do fabricante, uma interface de configuração web permite que os administradores distribuam o controle do gerenciamento do monitoramento e das configurações do sistema, e muito mais, para os usuários finais e membros da equipe, com facilidade. Os assistentes de configuração guiam os usuários nos processos de monitoramento de novos dispositivos, serviços e aplicativos, em uma plataforma bastante intuitiva. De acordo com o fabricante do software estas são as principais características da plataforma:

– Detecção de falha na infra-estrutura da rede gerenciada;

– Os alertas gerados podem ser enviados para um grupo ou apenas um usuário responsável pelo monitoramento;

[2] https://www.hw-group.com/device/ste2

– Oferece escalabilidade, podendo monitorar milhares de *hosts;*

– Monitora diversos serviços de rede, entre eles: SMTP, POP3, HTTP, HTTPS, NMTP, ICMP, SNMP;

– Monitora carga de processador, uso de disco, logs de sistemas na maioria dos sistemas operacionais do mercado;

– Suporte a SSH e a SSL; Interação com banco de dados;

– Uso de computação paralela, ou seja, quando houver muitos itens sendo monitorados, não existe o risco de algum não ser checado;

– Capacidade de definir a rede hierarquicamente através de equipamentos "pai", permitindo a diferenciação de equipamentos que estão indisponíveis daqueles que estão inalcançáveis.

De acordo com a proposta de gerenciamento de um dispositivo IoT, utilizando a plataforma Nagios, será feito o uso de uma versão de testes do programa, no caso a versão XI, válida para até 60 dias após a sua instalação. Existem versões gratuitas do Nagios, porém versões com menos recursos. Nesse trabalho se fará uso da versão mais completa da atualidade, a XI, que contempla todas as funcionalidades acima citadas.

O Nagios será instalado em uma máquina virtual VMware®, dentro do sistema operacional nativo do Macbook Pro®, onde poder-se determinar todas as configurações, como memória, uso de núcleos, uso de disco etc. Mesmo com uma diversa gama de funções, os requisitos computacionais para executar o Nagios são poucos. Após instalado, o Nagios poderá acessado de qualquer máquina/dispositivo móvel dentro da rede intranet, com as devidas credenciais de acesso.

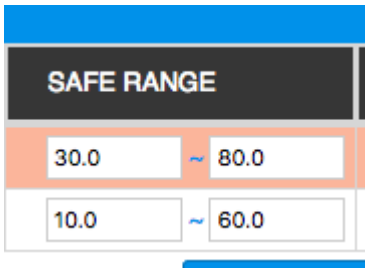
4.3 GERENCIAMENTO DO DISPOSITIVO

O primeiro passo para gerenciar o dispositivo, é o Nagios “entender” o que ele irá gerenciar. Para isto, o fabricante do dispositivo STE2 disponibiliza uma tabela MIB do equipamento para o Nagios compilar e usar como referências em suas consultas. Após esta compilação, através de assistentes, será feita a conexão entre o gerenciador Nagios e o dispositivo STE2 através de TCP/IP.

O gerenciamento do STE2 através do Nagios poderá ser feito de maneira ativa ou passiva. A forma ativa efetua a conexão do Nagios com o STE2 (através de conexão UDP e porta 161) onde o software gerente, em um período pré-determinado, fará consultas às variáveis de temperatura e umidade. Nesta consulta já estão pré-determinados os valores considerados seguros para a umidade e a temperatura. Fora desta faixa de segurança, o Nagios emitirá notificações de alerta para o administrador de rede ou de algum grupo pré-definido.

A forma passiva é o monitoramento (porta 162) através das TRAPs do STE2, o que significa dizer que o envio de mensagens do dispositivo é automático, no caso de algum valor do sensor estiver fora da faixa considerada segura. Estes valores para as TRAPs são pré-definidos no dispositivo. No caso, foi pré-definida a faixa segura de 30% a 80% de umidade relativa do ar e de 10º a 60º Celsius para a temperatura. Fora destas faixas, as TRAPS são automaticamente acionadas e enviadas para o Nagios assumir o papel de notificar o administrador de rede ou de algum grupo pré-definido.

Figura 18 – Faixa segura de medição dos sensores STE2 através de interface web



Fonte: o Autor (2018)

4.4 CONSIDERAÇÕES DO CAPÍTULO

Nesta etapa de integração, todos os conceitos relativos ao protocolo SNMP serão colocados em práticas envolvendo as OIDs e as MIBs fornecidas pelo dispositivo STE2. O Nagios assumirá o seu papel de gerente, estando em alerta sobre qualquer variação pré-definida nos sensores de temperatura e umidade no dispositivo em questão. O Nagios assume o papel de organizador do comando SNMP; ao efetuar a conexão, ele facilitará a integração do agente e do gerente por meio de

uma interface web. Mesmo efetuando essa conexão com o auxílio de um *wizard,* será possível visualizar os comandos que ele realiza em background.

5 IMPLEMENTAÇÃO E TESTES DE GERENCIAMENTO DO DISPOSITIVO

Nesta seção foi especificado todo o processo de integração entre o dispositivo escolhido STE2 e o software Nagios, bem como foram detalhados todos os aspectos de instalação de cada item (STE2 e Nagios). Qualquer dispositivo com suporte SNMP trará características diferentes na hora de instalação, então, não é possível abordar uma ligação genérica de um dispositivo, pois cada um terá a sua peculiaridade. O STE2 conta com uma interface web no próprio sistema, que facilita para o usuário na obtenção do arquivo MIB, como também para a visualização em tempo real das OIDS (Figura 19). Por outro lado, a comunidade de desenvolvimento da plataforma Nagios é extensa, o que facilita a documentação dos procedimentos envolvidos.

Figura 19 – Visualização em tempo real das OIDS no STE2

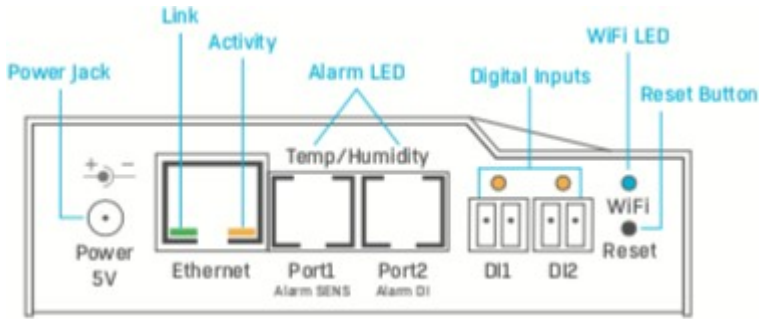
Oid Key	Value	Description	Data Type
1.3.6.1.2.1.1.1.0	STE2	System Description	string
1.3.6.1.2.1.1.2.0	1.3.6.1.4.1.21796.4.9.	System ObjectID	objid
1.3.6.1.2.1.1.3.0	6188400	System UpTime	timeticks
1.3.6.1.2.1.1.4.0	STE2: For more information try http://www.HW-group.com	System Contact	string
1.3.6.1.2.1.1.5.0	STE2	System Name	string
1.3.6.1.2.1.1.6.0		System Location	string
1.3.6.1.2.1.1.7.0	72	System Services	integer
1.3.6.1.4.1.21796.4.9.1.1.1.1	1	1. Input Index	integer
1.3.6.1.4.1.21796.4.9.1.1.1.2	2	2. Input Index	integer
1.3.6.1.4.1.21796.4.9.1.1.2.1	0	1. Input Value, 0=Open, 1=Close	integer
1.3.6.1.4.1.21796.4.9.1.1.2.2	0	2. Input Value, 0=Open, 1=Close	integer
1.3.6.1.4.1.21796.4.9.1.1.3.1	Input 1	1. Input Name	string
1.3.6.1.4.1.21796.4.9.1.1.3.2	Input 2	2. Input Name	string
1.3.6.1.4.1.21796.4.9.1.1.4.1	0	1. Input State, 0=Normal, 1=Alarm	integer
1.3.6.1.4.1.21796.4.9.1.1.4.2	0	2. Input State, 0=Normal, 1=Alarm	integer
1.3.6.1.4.1.21796.4.9.3.1.1.1	1	1. Sensor Index	integer
1.3.6.1.4.1.21796.4.9.3.1.1.2	2	2. Sensor Index	integer
1.3.6.1.4.1.21796.4.9.3.1.2.1	Sensor 2594	1. Sensor Name	string
1.3.6.1.4.1.21796.4.9.3.1.2.2	Sensor 3594 te	2. Sensor Name	string
1.3.6.1.4.1.21796.4.9.3.1.3.1	4	1. Sensor State	integer
1.3.6.1.4.1.21796.4.9.3.1.3.2	1	2. Sensor State	integer
1.3.6.1.4.1.21796.4.9.3.1.4.1	21.6	1. Sensor String Value	string
1.3.6.1.4.1.21796.4.9.3.1.4.2	33.4	2. Sensor String Value	string
1.3.6.1.4.1.21796.4.9.3.1.5.1	216	1. Sensor Value	integer
1.3.6.1.4.1.21796.4.9.3.1.5.2	334	2. Sensor Value	integer
1.3.6.1.4.1.21796.4.9.3.1.6.1	26220A570520086C	1. Sensor SN	string
1.3.6.1.4.1.21796.4.9.3.1.6.2	280A0E570520081B	2. Sensor SN	string
1.3.6.1.4.1.21796.4.9.3.1.7.1	4	1. Sensor Unit, 1=C, 2=F, 3=K, 4=%	integer
1.3.6.1.4.1.21796.4.9.3.1.7.2	1	2. Sensor Unit, 1=C, 2=F, 3=K, 4=%	integer
1.3.6.1.4.1.21796.4.9.3.1.8.1	2594	1. Sensor ID	integer
1.3.6.1.4.1.21796.4.9.3.1.8.2	3594	2. Sensor ID	integer

Fonte: o Autor (2018)

5.1 INSTALAÇÃO DO STE2

O dispositivo STE2 é de fácil instalação, a embalagem acompanha um adaptador de energia, um sensor de temperatura e um de umidade (sensores com conexão padrão RJ11) e um CD para instalação.

Figura 20 – Esquema de ligação do STE2



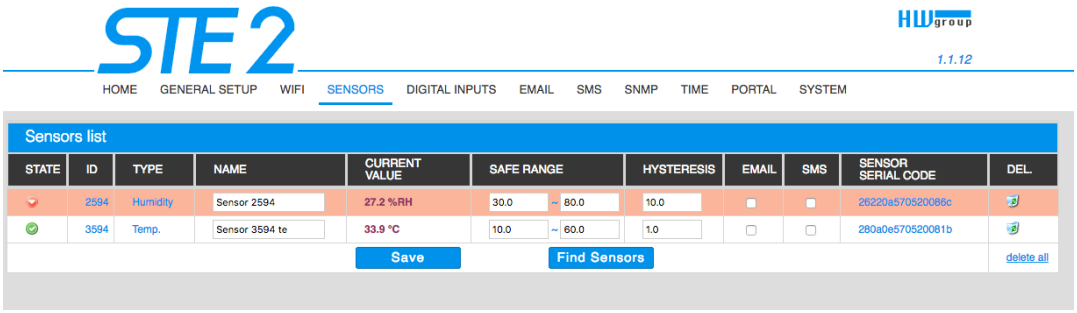
Fonte: Manual STE2

O dispositivo IoT STE2 dispõe de conectividade TCP/IP wireless ou por cabo (padrão Ethernet). Automaticamente quando conectado à rede local (via cabo), é atribuido um endereço IP (por DHCP). Caso seja necessária a introdução de um endereço IP manual, o equipamento também permite ao usuário. Como acompanha um CD para instalação, com o auxílio de um aplicativo, o usuário pode fazer a configuração do aparelho via wireless. Após o dispositivo ser reconhecido na rede (obtendo um endereço IP válido), é só digitar em um navegador o endereço IP do dispositivo STE2. Neste primeiro acesso foram ajustadas as configurações como hora, data e usuário, para proteger o dispositivo com um login e senha, assim como se faz com *routers*. Dentro desta interface também estão disponíveis as configurações de rede (com e sem fio), configurações sobre atualizações de *firmware* e um sistema próprio de gerenciamento dos sensores, onde se pode estipular as faixas seguras de operação dos sensores. O próprio sistema disponibiliza ao usuário o gerenciamento dos sensores sem o uso de um gerenciador externo. É um sistema bastante simplificado, no qual se estipula uma faixa de operação dos sensores que, quando saem desta faixa, o usuário, já pré-cadastrado, recebe um email ou sms do sistema.

O equipamento é acompanha por dois sensores, um para medição de tem-

peratura e o outro para a medição de umidade (higrômetro). Ambos os sensores foram conectados na porta RJ11 ao equipamento STE2. Dentro da interface web de gerenciamento do STE2, foi efetuada a busca automática dos sensores; localizados, foram gravados no equipamento, como a figura abaixo demonstra.

Figura 21 – Lista de sensores

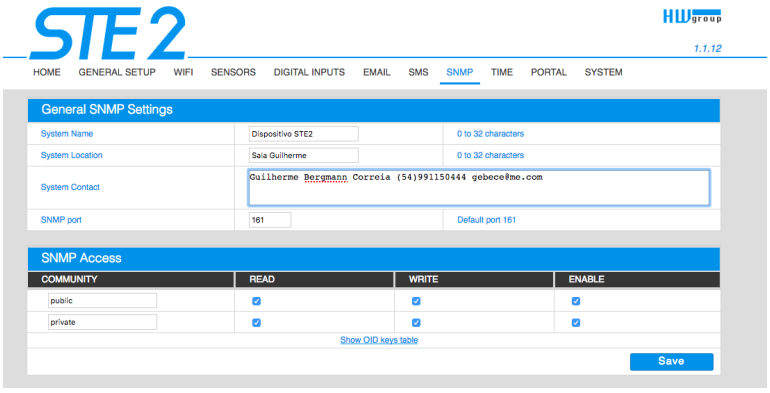


Fonte: software gerenciador STE2

Na nossa instalação via cabo de rede (Ethernet), o endereço IP atribuído ao equipamento (automaticamente via DHCP) foi 192.168.25.29.

Dentro da interface de controle do STE2, existe a aba SNMP. Nela foi possível ajustar *strings,* como nome do aparelho, localização do aparelho e as informações do gerente de rede, como também a porta de operação do SNMP (161 por padrão). Nesta mesma aba SNMP, pode-se efetuar os ajustes de acesso ao SNMP das chamadas *communities* que podem ser publicas ou privadas.

Figura 22 – Aba SNMP da interface web do STE2



Fonte: o Autor (2018)

Na ultima aba do da interface web (aba SYSTEM) o STE2 dispõe das tabelas MIBs e a lista de OIDs para o dispositivo. No Anexo A deste trabalho encontram-se a as MIBs baixadas diretamente da interface web do STE2

No Anexo B deste trabalho, encontram-se as OIDs a serem monitoradas, no caso serão monitoradas temperatura e umidade.

OID temperatura-> 1.3.6.1.4.1.21796.4.9.3.1.4.2

OID umidade-> 1.3.6.1.4.1.21796.4.9.3.1.4.1

Após ligado na energia e conectado no cabo de rede junto a LAN, o dispositivo STE2 foi recebeu automaticamente o endereço 192.168.25.29 (por DHCP), com a portas 161 e 162 abertas

5.2 INSTALAÇÃO DO NAGIOS

De acordo com Black (2008), para rodar o Nagios é necessário apenas um computador executando Linux (ou variações do Unix) e um compilador C. Originalmente, ele foi desenvolvido para rodar em Linux, mas já existem pacotes personalizados para distribuições Linux comuns, como Fedora, Ubuntu, SUSE e Debian. É comum também a instalação do Nagios em ambientes virtuais como VMWARE®. De acordo com Becker e Moura (2012), a configuração do Nagios é baseada em arquivos.

– nagios.cfg: é o arquivo principal da configuração (geralmente localizado em /usr/local/nagios/etc/nagios.cfg). Contém várias diretivas que afetam a operação do Nagios, e onde são feitas as referências para os outros arquivos de configuração;

– cgi.cfg: é o arquivo onde são configuradas as CGIs (*Common Gateway Interface*), usadas para dar suporte à funcionalidades extras do Nagios (geralmente localizado em /usr/local/nagios/etc/cgi.cfg);

– commands.cfg: arquivo onde são configuradas as instruções permitidas para serem executadas pelo Nagios (geralmente localizado em /usr/local/nagios/etc/commands.cfg);

– contacts.cfg: arquivo onde são formatados os contatos que serão notifica-

dos quando houver eventos que emitirem alertas. Nesse arquivo também é possível a elaboração de grupos de contatos (arquivo geralmente localizado em /usr/local/nagios/etc/contacts.cfg);

– timeperiods.cfg: arquivo onde se configura os tempos/períodos de monitoramento, como, por exemplo, um serviço não essencial que precisa estar ativo somente durante a semana, pode ter o monitoramento desativado nos finais de semana, para não transmitir alertas desnecessários já que o seu funcionamento não é vital nesse período (arquivo geralmente localizado em /usr/local/nagios/etc/timeperiods.cfg).

A plataforma Nagios trabalha sobre o Linux Kernel, e, para sua instalação, não é necessário grandes recursos computacionais. De acordo com o fabricante, o mínimo para o Nagios operar é um processador de 1GHz, 1 GB de memória RAM e espaço em disco e 8GB. Porém, a configuração recomendada pelo fabricante é a seguinte:

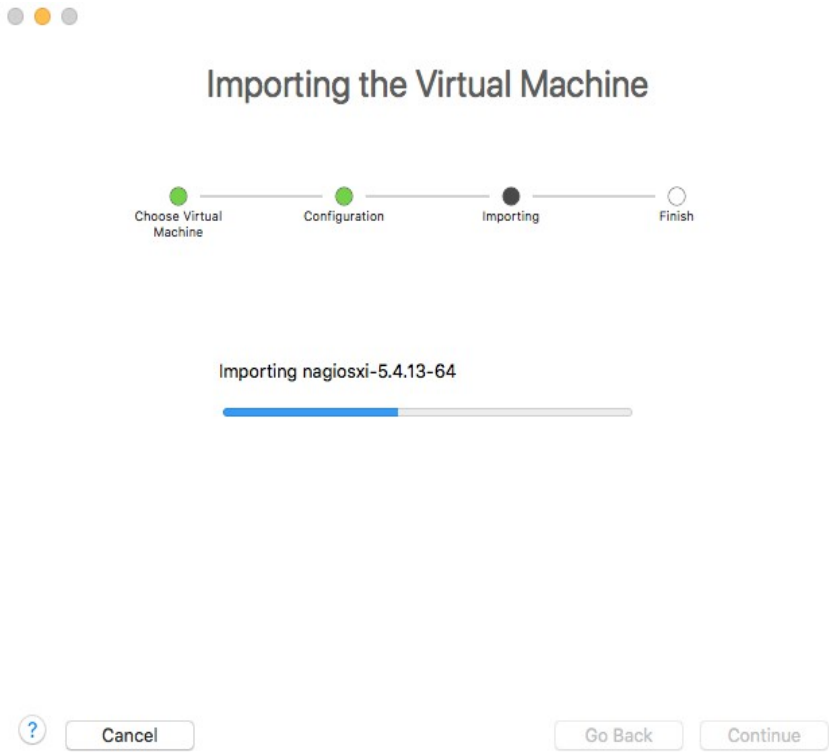
– Processador com 2GHz ou mais; 2GB de memória RAM;
– 40GB em espaço de disco;
– Configuração de discos RAID 5 (por motivos de redundância).

Neste trabalho foi utilizada a versão XI do Nagios para propósitos de testes e conexão com dispositivo IoT. Para realizar a instalação do Nagios foi utilizada uma versão disponível para testes (60 dias) do arquivo no formato OVA, que é um formato disponível para VMWARE® (arquivo nagiosxi-5.4.13-64.ova de 1.54GB). VMware® é um software de máquina virtual que permite a instalação e utilização de um sistema operacional dentro de outro. Foi feito o download do arquivo formato OVA direto no site do fabricante do software Nagios. O arquivo OVA é uma máquina virtual, já pré-configurada, que contém todo o núcleo do sistema Linux (para poder rodar o Nagios sobre ele). O Linux usado nesta máquina virtual foi o Linux CentOS, que trata de uma distribuição Linux de classe corporativa derivada de códigos fonte gratuitamente distribuídos pela e mantida pela GLP. Para a instalação foi consultada também a documentação disponibilizada no site oficial da empresa Nagios. Para fins

de testes, o Nagios foi instalado em um computador Apple Macbook Pro®, com 16GB de memória RAM e 240GB de capacidade de disco.

A figura abaixo demonstra a importação do arquivo OVA pelo baixado no site oficial da empresa Nagios.

Figura 23 – Importação da máquina virtual Nagios pelo WMWARE



Fonte: o Autor (2018)

Após a instalação da máquina virtual contendo o Nagios XI, a interface (Linux) foi exibida da seguinte maneira (já com o servidor Nagios rodando em background).

Figura 24 – Interface Linux com servidor Nagios em background

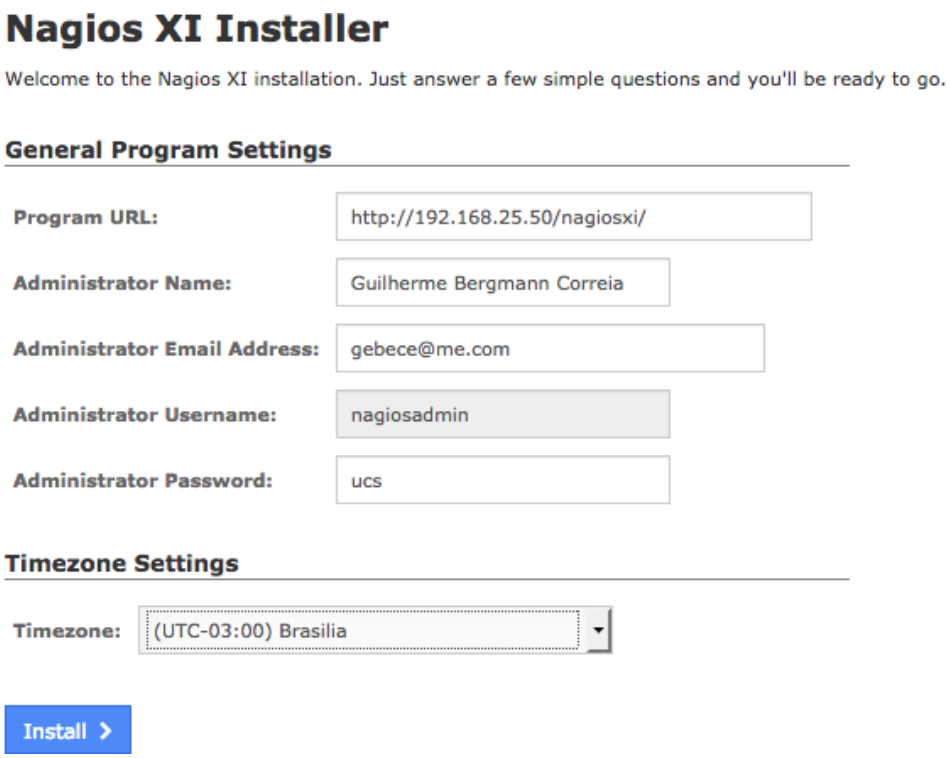
Fonte: o Autor (2018)

Caso necessária alguma configuração direta no ambiente Linux, é necessário entrar com as credencias (login e senha) já pré-estabelecidas na documentação (root e nagiosxi). Caso contrário, o acesso ao software Nagios é efetuado diretamente pelo *browser* local da máquina, ou em qualquer *browser* que encontra- se na mesma LAN. Com o uso da interface web, a utilização do Nagios torna-se mais atrativa e agradável, substituindo a edição dos arquivos de configuração em interface de linha de comando pela edição em interface web.

O acesso do software Nagios foi feito via *browser,* via endereço IP pré-determinado pelo software. No caso, o endereço que a interface do Nagios foi disponibilizada foi 192.168.25.50.

A primeira tela do Nagios após entrar-se com o endereço 192.168.25.50 no navegador, trata de alguma configurações rápidas, como a url de acesso ao sistema, nome do usuário, endereço de email, login e senha para acesso a interface web do Nagios e, por último, o *timezone*. (Figura 25)

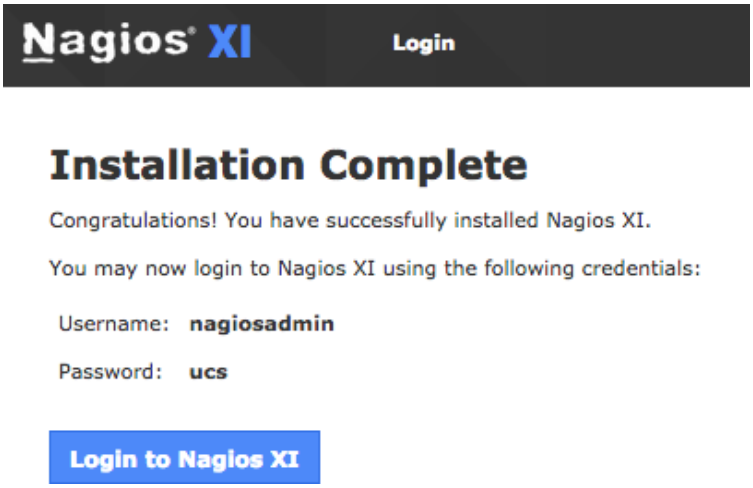
Figura 25 – Interface Web/Menu de instalação Nagios



Fonte: o Autor (2018)

Após este rápido *setup* a interface web do Nagios alerta que o Nagios foi instalado com sucesso. (Figura 26).

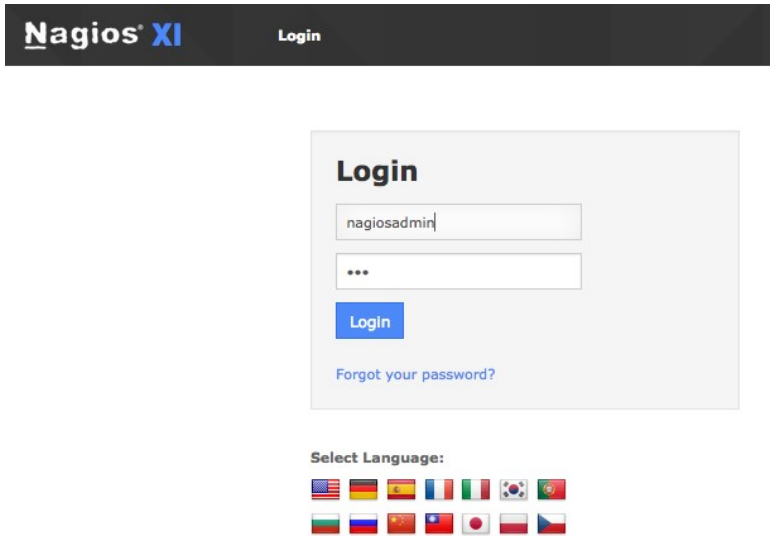
Figura 26 – Tela final de instalação do Nagios/Interface Web



Fonte: o Autor (2018)

A partir deste ponto, o acesso ao sistema web do Nagios é feito pelo endereço 192.168.25.50, onde é solicitado as credenciais do usuário.

Figura 27 – Acesso ao sistema web do Nagios



Fonte: o Autor (2018)

Antes do login, pode-se escolher pela língua a ser usada. Foi escolhida a língua inglesa pela disponibilidade/acessibilidade de documentação e tutoriais existentes na rede Internet.

5.3 COMUNICAÇÃO ENTRE O DISPOSITIVO E O NAGIOS

Após instalados o dispositivo STE2 e o Nagios, nos endereços 192.168.25.29 e 192.168.25.50, respectivamente, foi efetuada a primeira etapa da conexão entre eles. A primeira etapa de integração foi a instalação da função de TRAPS no Nagios, função que não acompanha a versão que instalada. Para isso, foram necessários alguns comandos direto no terminal do Nagios. Para acesso ao terminal, foram srequerido login e senha. Após login e senha no terminal do Nagios, foi necessário executar os seguintes comandos (figura 29).

Figura 28 – Comandos para a instalação de TRAPS no Nagios

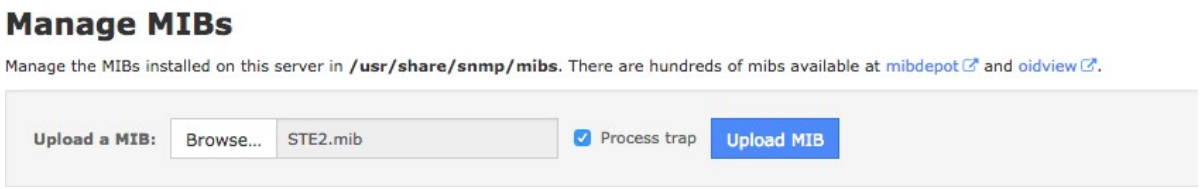
```
cd /tmp
wget https://assets.nagios.com/downloads/nagiosxi/scripts/NagiosXI-SNMPTrap-setup.sh
sh ./NagiosXI-SNMPTrap-setup.sh
```

Fonte: https://assets.nagios.com/downloads/nagiosxi/docs/Integrating_SNMP_Traps_With_Nagios_XI.pdf

Monitorar um sensor de temperatura via SNMP requer um arquivo de definição da MIB. O passo seguinte foi instalar a MIB do dispositivo STE2 no Nagios, MIB que foi disponibilizada pelo dispositivo no endereço 192.168.25.29/STE2.mib.

Uma vez baixado o arquivo STE2.mib na máquina local, foi necessário instalar a tabela MIB através da interface web do Nagios, seguindo o caminho: Admin > System Extensions > Manage MIBs.

Figura 29 – Instalação da MIB STE2 na interface web do Nagios



Fonte: o Autor (2018)

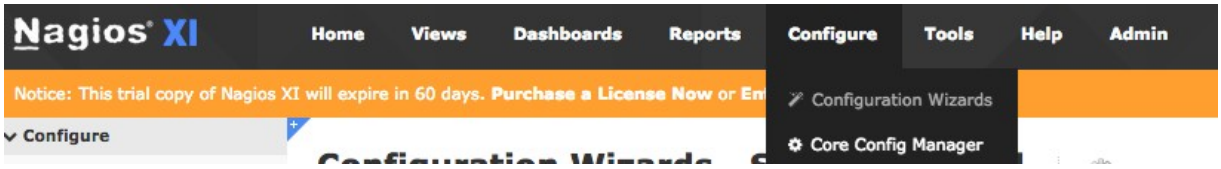
Foi selecionada a opção *process trap*, a fim de permitir que o Nagios execute TRAPS a partir da MIB que foi efetuada o upload.

Com o upload do arquivo MIB, obteve-se uma resposta do sistema: “MIB file successfully processed.”

O passo seguinte foi usar o assistente SNMP (*wizard*) do Nagios para configurar as conexões com o dispositivo STE2. Este passo foi efetuado navegando na interface web do Nagios (aba superior) pelo seguinte caminho:

Configure > Configuration Wizards (Figura 31).

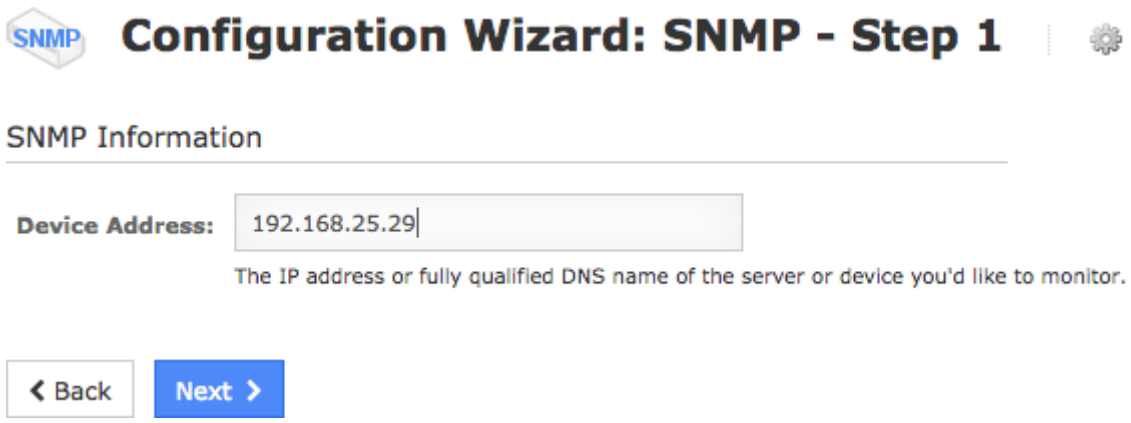
Figura 30 – Opção “configuration wizard” da interface web do Nagios



Fonte: o Autor (2018)

Dentro dos inúmeros assistentes de conexão do Nagios, foi selecionado o assistente SNMP. (Figura 31)

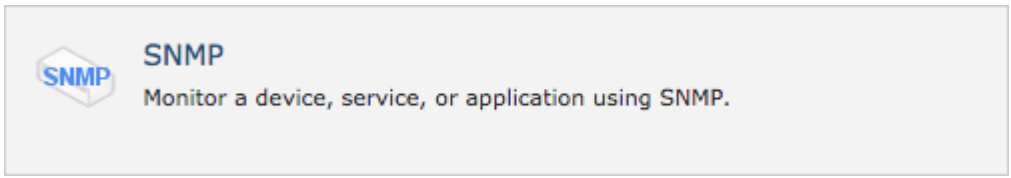
Figura 31 – Assistente SNMP via interface do Nagios



Fonte: o Autor (2018)

O próximo passo, já dentro do assistente, foi inserir o endereço 192.168.25.29 do dispositivo STE2 (Figura 32)

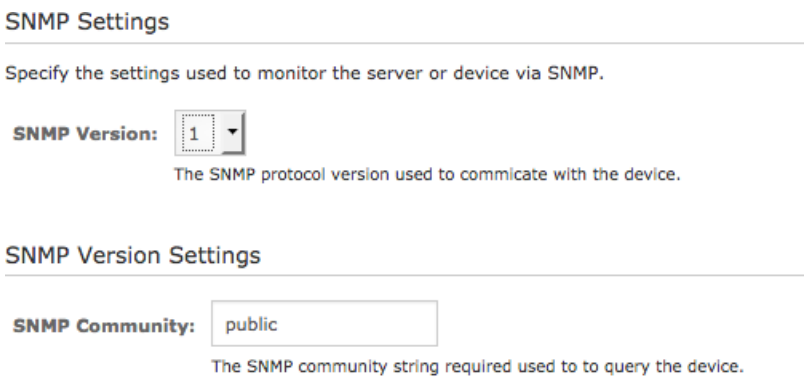
Figura 32 – Passo 1 da conexão via assistente SNMP do Nagios



Fonte: o Autor (2018)

Após esta etapa de endereçamento, na página seguinte foi inserida a SNMP community do dispositivo STE2, que é do tipo *public,* a versão do SNMP, que é a versão 1 e as OIDs que serão monitoradas via SNMP.

Figura 33 – Community e versão do SNMP na configuração da conexão SNMP na interface web do Nagios

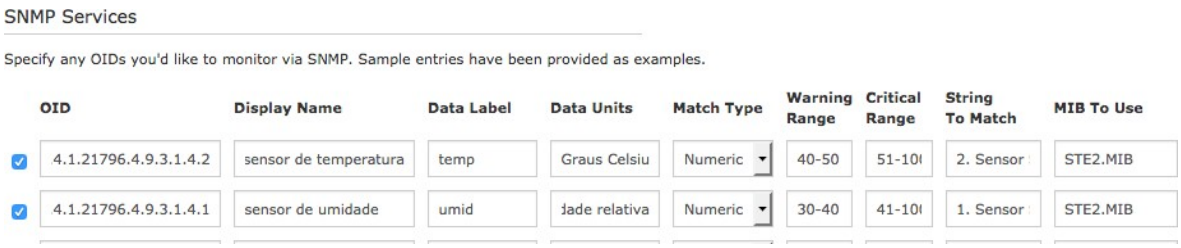


Fonte: o Autor (2018)

No Anexo A deste trabalho, estão as OIDs que podem ser monitoradas. Foram escolhidas as seguintes OIDs para esta etapa:

– OID temperatura: 1.3.6.1.4.1.21796.4.9.3.1.4.2
– OID umidade: 1.3.6.1.4.1.21796.4.9.3.1.4.1

Figura 34 – OIDS monitoradas do dispositivo STE2 no gerenciador web do Nagios



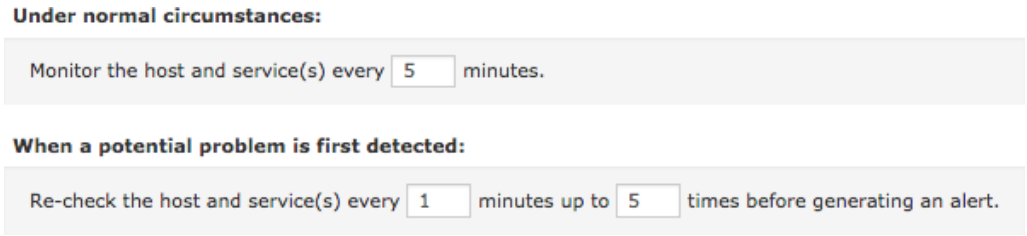
Fonte: o Autor (2018)

Nesta etapa, o assistente de conexão SNMP solicita nove campos para preenchimento, que correspondem às definições da OIDs: o nome da OID (disponível

no Anexo B), o nome para exibição, a unidade a ser usada (graus Celsius e percentual de umidade relativa), o tipo da unidade (*numeric),* a faixa de atenção da unidade, a faixa crítica da unidade, a *string* correspondente à OID (disponível na aba *system* do gerenciador web do STE2, onde é a possível a consulta das OIDs online) e a MIB para consulta (STE2.mib). Os valores foram preenchidos de acordo com a documentação do Nagios e do STE2. A faixa de atenção (Warning Range) para temperatura inicia-se entre 40 graus e 50 graus Celsius. A faixa crítica (Critical Range) a partir de 51 graus e 100 graus Celsius. Para a leitura de umidade, a faixa de atenção inicia-se entre 40% e 50% de umidade relativa do ar; a faixa crítica entre 51% e 100% de umidade relativa do ar.

Na etapa posterior é possível efetuar os ajustes de tempo para checagem dos parâmetros inseridos. Foi deixado o valor padrão do Nagios: cinco minutos para checagem sob circunstâncias normais. O outro campo assume que, quando existe algum valor na faixa de atenção, é feita uma re-checagem a cada minuto até cinco vezes antes de emitir um alerta ao usuário cadastrado.

Figura 35 – Parâmetros de checagem de tempo dos OIDS inseridos para controle do STE2

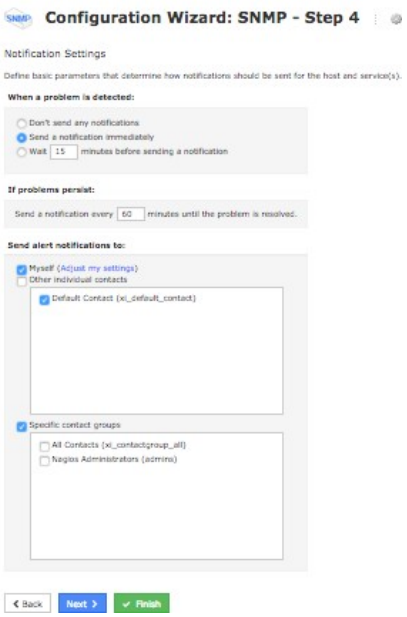


Fonte: o Autor (2018)

A próxima etapa do assistente SNMP é a respeito das notificações, permitindo as seguintes opções ao usuário (Figura 36):

– Quando um problema é detectado: Não enviar notificações, enviar imediatamente, ou aguardar X minutos antes de enviar um notificação.
– Quanto um problema persiste: Enviar notificacões a cada X minutos até o problema for resolvido.
– Enviar alertas para: É possível enviar individualmente o alerta ou para algum grupo pré-determinado no sistema.

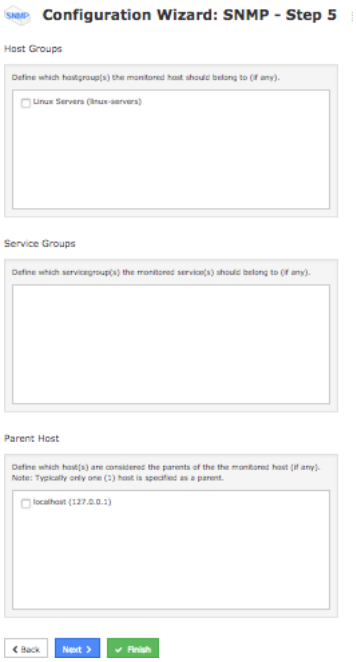
Figura 36 – Sistema de notificações no Nagios



Fonte: o Autor (2018)

Na próxima etapa, pode-se definir a existência de um *host* pai, ou algum grupo de *hosts* a que o STE2 pertença, não possuindo aplicação para a nossa integração, uma vez que o *host* STE2 está ligado diretamente na LAN e o *host* pai (roteador) não possui suporte SNMP. (Figura 37)

Figura 37 – Grupo de *hosts*



Fonte: o Autor (2018)

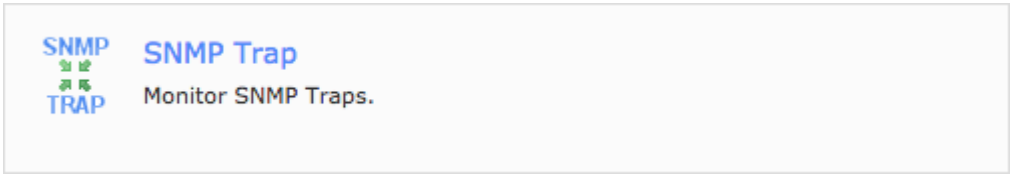
Na ultima etapa do assistente, após o preenchimento de todos os campos listados anteriormente, após a submissão, o Nagios emite uma mensagem que as configurações foram salvas com sucesso e que os serviços de monitoramento foram reiniciados.

Figura 38 – Configuração final do assistente SNMP Nagios

Fonte: o Autor (2018)

Pode-se efetuar uma conexão passiva do dispositivo STE2 com o Nagios, através de TRAPS. Este passo foi efetuado, navegando na interface *web* do Nagios (aba superior) pelo seguinte caminho*:* Configure > Configuration Wizards (Figura 39). Dentro da gama de assistentes, foi escolhido o SNMP TRAP.

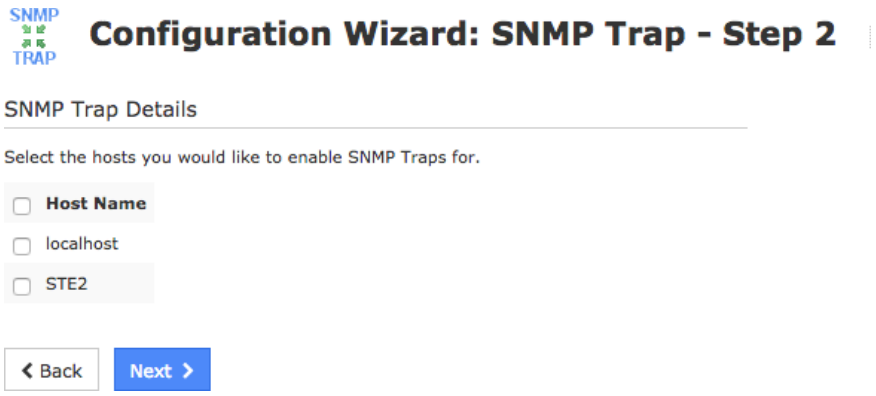
Figura 39 – Assistente SNMP TRAP



Fonte: o Autor (2018)

Na próxima etapa do assistente, é solicitado qual *host* será monitorado:

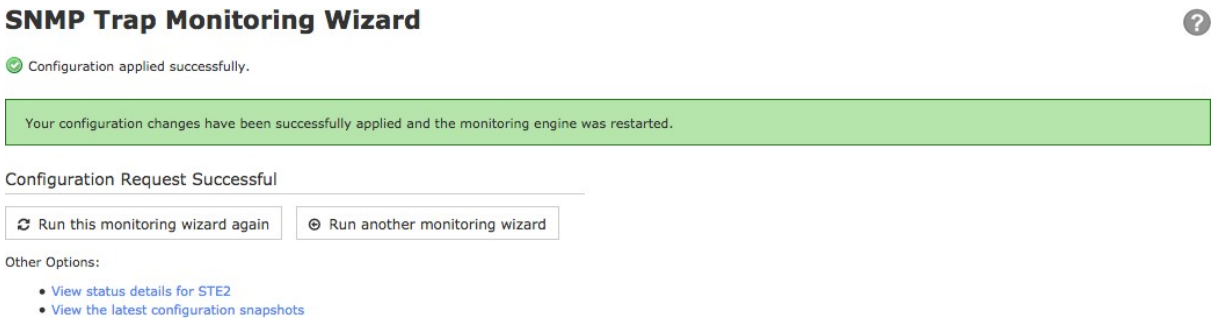
Figura 40 – *Hosts* para monitoramento TRAP



Fonte: o Autor (2018)

O Host STE2 foi selecionado e, automaticamente, pela porta 162 se deu o início do monitoramento TRAPS do STE2. As configurações subsequentes são apenas sobre notificações e grupos de *hosts*; no final do assistente, após submissão, o Nagios emite uma mensagem que as configurações de TRAPS foram salvas com sucesso e que os serviços de monitoramento foram reiniciados como segue na figura abaixo:

Figura 41 – Configuração final do assistente SNMP TRAPS do Nagios



Fonte: o Autor (2018)

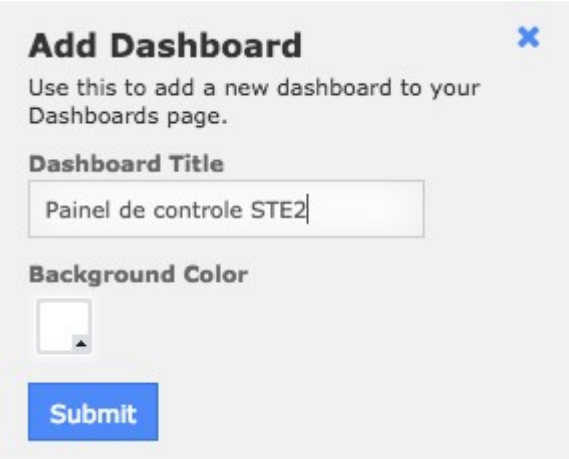
5.4 TESTES

Após a conexão do Nagios com STE2, através do assistente, foi configurado no sistema Nagios o tempo que ele efetua as consultas, para efetuar-se uma consul-

ta de temperatura e umidade em tempo real. Para isso foram criados *dashlets* que são elementos customizáveis para a criação de *dashboards.*

Foram criados dois *dashlets*: um para a temperatura e um para a umidade relativa do ar. A criação é bastante simples, dentro da interface web do Nagios. Na aba *dashboards,* primeiramente se cria um novo *dashboard* (botão Add New Dashboard).

Figura 42 – Criação de um novo *dashboard* na interface web do Nagios

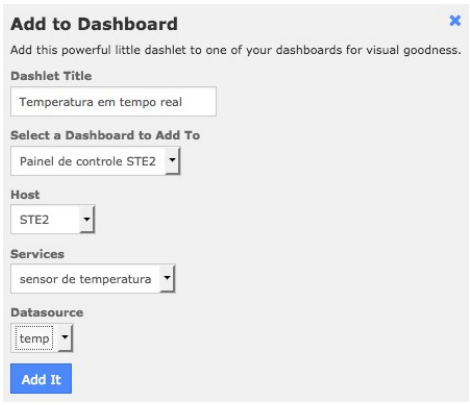


Fonte: o Autor (2018)

Foi criado um *dashboard* com o nome "Painel de Controle STE2", com a cor de fundo branca. A partir da criação do *dashboard,* serão adicionados *dashlets*, elementos que exibirão as variáveis que serão mostrados no *dashboard* criado.

O primeiro *dashlet* criado foi o de temperatura em tempo real do STE2. O Nagios oferece uma variedade de tipos de *dashlets,* a grande maioria fornecida por desenvolvedores terceiros. Foi usado, para o *dashboard* criado, o *dashlet* do tipo *gauge*, desenvolvido pela própria Nagios. Para criar um *dashlet,* foram solicitados os seguintes campos: Título do *dashlet,* o *dashboard* que ele faz parte, o *host* a ser conectado, o serviço a ser usado e a fonte de dados. (Figura 43)

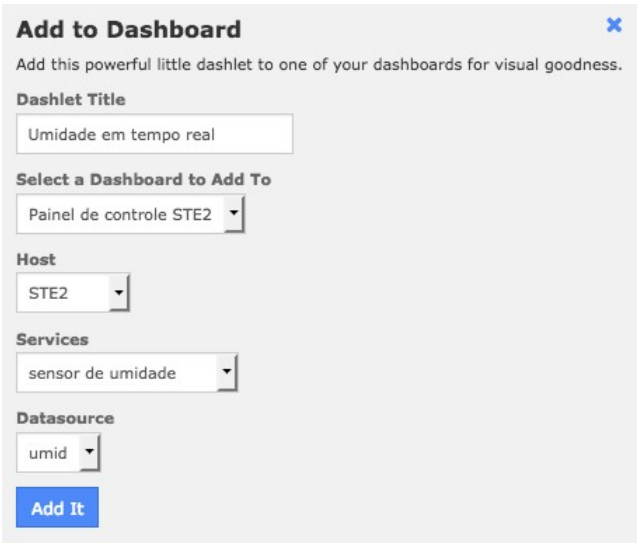
Figura 43 – Criação do *dashlet* de temperatura



Fonte: o Autor (2018)

Da mesma forma, foi criado o *dashlet* para umidade (Figura 44):

Figura 44 – Criação do *dashlet* de umidade



Fonte: o Autor (2018)

Como resultado, obtem-se o *dashboard* denominado Painel de Controle STE2, com dois *dashlets*, um de temperatura (temp) e um de umidade em tempo real.

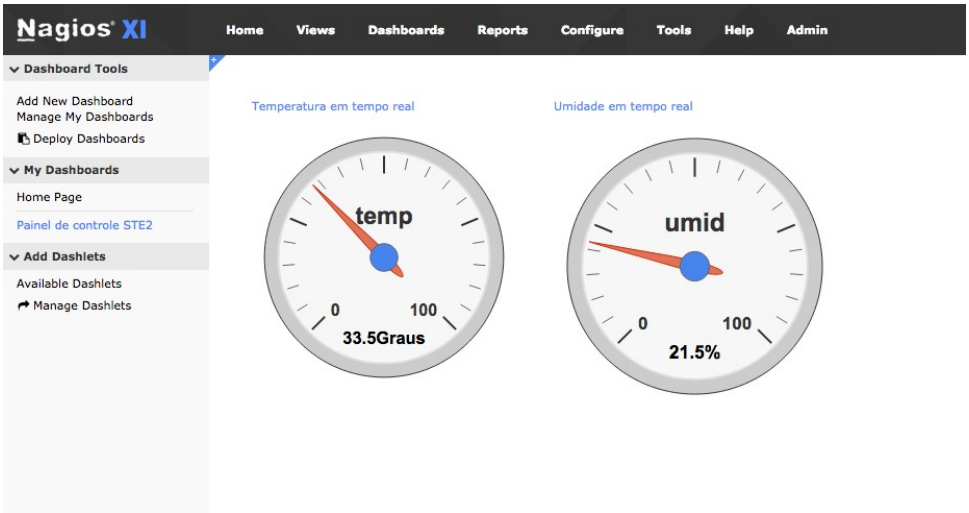


Figura 45 – *Dashboard* com *dashlets* de temperatura e umidade

Fonte: o Autor (2018)

Dentro da forma ativa de gerenciamento, o comando em nível de terminal para checagem de temperatura assume este formato (Figura 46):

Figura 46 – Sintaxe do comando check_snmp em terminal Linux

```
[nagios@new-host-9 ~]$ /usr/local/nagios/libexec/check_snmp -H
192.168.25.29 -o 1.3.6.1.4.1.21796.4.9.3.1.4.2 -C public -P 1 -l
"temperatura" -u "Graus Celsius" -m STE2.MIB -w 20-30 -c 31-100
SNMP CRITICAL - temperatura *33.5* Graus Celsius |
temperatura=33.5Graus Celsius;20;31;
```

Fonte: o Autor (2018)

As faixas de atenção e as faixas críticas de temperatura foram alteradas para os seguintes valores: faixa de atenção (*Warning Range*) de 40-50 para 20-30 e a faixa crítica de 31 para 100, para poder-se gerar notificações no sistema.

Figura 47 – Sintaxe com as faixas de atenção e críticas alteradas

```
[nagios@new-host-9 ~]$ /usar/local/nagios/libexec/check_snmp -H
192.168.25.29 -o 1.3.6.1.4.1.21796.4.9.3.1.4.2 -C public -P 1 -l
"temperatura" -u "Graus Celsius" -m STE2.MIB -w 40-50 -c 51-100
SNMP OK - temperatura 33.5 Graus Celsius | temperatura=33.5Graus
Celsius;40;51;
```

Fonte: o Autor (2018)

No segundo comando, a resposta foi "SNMP CRITICAL", uma vez que a temperatura lida foi de 33.5º Celsius e isso ficou dentro da faixa crítica criada.

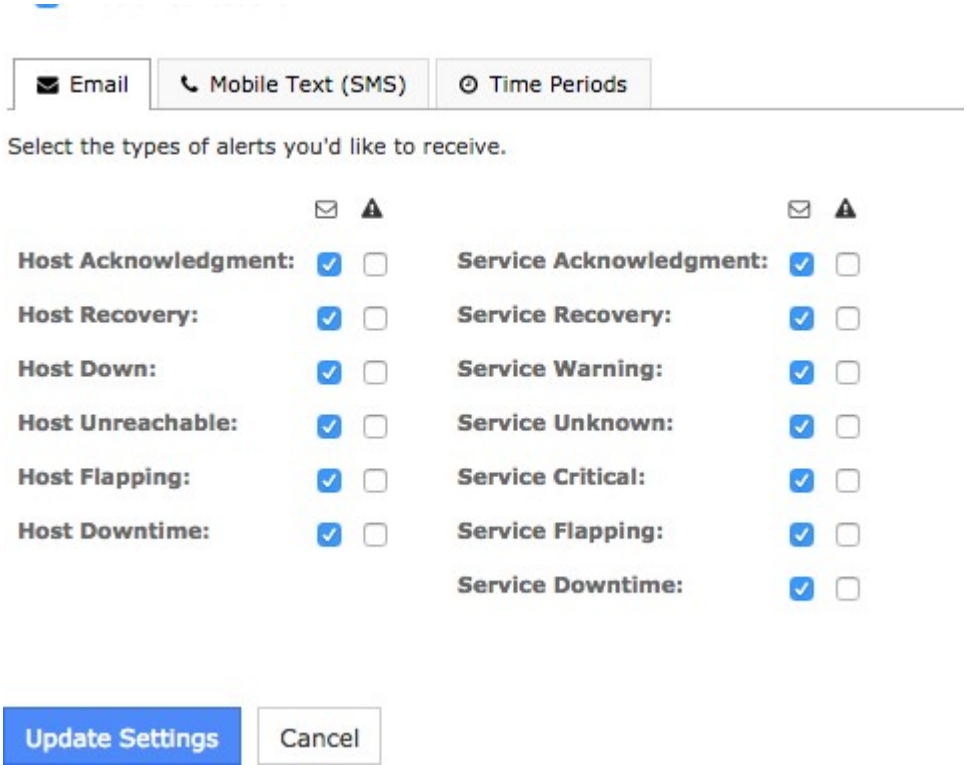
Figura 48 – Nagios responde a nova faixa crítica de temperatura

Host	Service	Status	Duration	Attempt	Last Check	Status Information
STE2	sensor de temperatura	Critical	20s	1/5	2018-06-25 23:07:19	SNMP CRITICAL - temperatura *33.5* Graus Celsius

Fonte: o Autor (2018)

Dentro das notificações que o usuário recebe, pode-se escolher quais tipos:

Figura 49 – Notificações de usuário

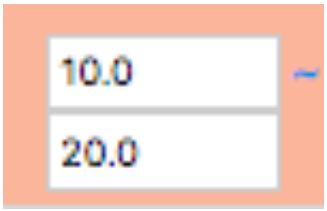


Fonte: o Autor (2018)

Como obteve-se na simulação um nível crítico de temperatura atingido, o Nagios se encarrega de notificar o usuário pré determinado na conexão SNMP. É possível customizar as mensagens de email e sms enviadas pelo Nagios.

Dentro das respostas passivas de checagem, foi feito o mesmo procedimento, porém, a faixa crítica de temperatura foi ajustada dentro do equipamento STE2. A faixa foi mudada de 10º Celsius a 60º Celsius para 10º a 20º Celsius. (Figura 50)

Figura 50 – Mudança de faixa de segurança
de temperatura no STE2



Fonte: o Autor (2018)

Desta maneira, o equipamento STE2 começou a gerar alertas de checagem passiva, as TRAPS, criando um alerta no software NAGIOS, que, ao contrário da maneira anterior, não fez uma checagem de valores: o software já recebeu o alerta devido à mudança de faixa de segurança do dispositivo.

5.5 CONSIDERAÇÕES DO CAPÍTULO

O uso do protocolo SNMP para gerenciamento de dispositivos IoTs é feita de de maneira simples com o Nagios. Mesmo com uma sintaxe rígida do protocolo, o Nagios permite, com facilidade, manipular as informações oriundas de qualquer dispositivo. Em um paralelo com uma linguagem de programação, o Nagios funciona como uma linguagem de alto nível para a execução do protocolo; a interface web da versão XI é bastante amigável e intuitiva. O Nagios, aliado ao protocolo SNMP, é uma ferramenta de extrema utilidade, podendo ser aplicada em vários campos

quando é preciso uma checagem constante de variáveis críticas.

O equipamento STE2 também é um equipamento que conta com uma interface amigável. Através dela tem-se acesso às MIBs e as OIDS usadas na conexão, o que facilitou muito esta etapa. A criação de *dashboard* customizada do sistema Nagios é um produto bastante simples de ser criado, uma vez que a conexão SNMP siga todas as suas etapas.

6 CONCLUSÃO

Este trabalho procurou explanar de maneira sintética a conexão de um software com um hardware. A escolha do protocolo de comunicação entre eles partiu de uma análise de protocolos existentes que o tempo ajudou a consolidar. No estudo de caso foi utilizado o protocolo SNMP com o software NAGIOS e, por fim, um dispositivo IoT, que poderia ser qualquer dispositivo que, através dos seus sensores, trouxesse a leitiura de algum dado do mundo físico para o virtual.

Em busca do objetivo deste trabalho, um estudo de caso foi proposto e executado em tempo hábil, podendo dessa maneira gerar e comprovar os resultados esperados na eficiente união destas três tecnologias. No estudo de caso foi utilizado um equipamento dotado de sensores de temperatura e umidade, mas certamente o que foi analisado aqui, independente das variáveis que foram gerenciadas, pode servir como base para estudos sobre qualquer outro dispositivo gerenciável pelo Nagios, ou mesmo qualquer outro software gerenciador de redes. Para sua continuidade como tecnologia, o uso do protocolo SNMP, ao mesmo tempo que se demonstrou sólido e eficaz nos seus resultados, depende muito dos fabricantes de dispositivos IoT em adotá-lo como padrão. Como dispositivos IoTs ainda são bastante difusos em relação as suas tecnologias e arquitetura, o cenário em relação ao futuro de protocolos gerenciais ainda não é suficientemente claro para poder-se afirmar que existe um protocolo que possa ser visto como tendência.

Dentro da etapa de conexão e testes, o software Nagios se mostrou eficiente no quesito capacidade de gerenciamento do dispositivo STE2, comprovando a solidez que o produto oferece em gerenciamento de redes, porém para chegar-se ao resultado desejado, as etapas que o assistente de conexão SNMP do Nagios oferece não são claras o bastante para efetuar a conexão. Foi necessária a busca de in-

formações fora de sua documentação original para chegar-se ao resultado final. O gerenciamento de dispositivos IoTs através do Nagios não é uma tarefa fácil, pois existe uma heterogeneidade de padrões de comunicação entre os dispositivos IoTs e o Nagios, que por sua vez não se mostrou eficiente no uso de seu assistente de conexão SNMP. Talvez por meio de outros protocolos, os assistentes de conexão existentes no Nagios possam ser considerados mais eficazes. A integração de protocolos se mostrou eficiente, uma vez que o gerenciador Nagios suporta até a versão 3 do protocolo SNMP e contempla toda gama de comandos para gerenciamento, porém não existe uma rigidez de sintaxe no preenchimento das instruções de conexão, dando margem a erros de conexão. Caso os campos forem preenchidos de maneira errada (sintaxe), a conexão não é estabelecida e ao mesmo tempo não fica claro para o usuário o motivo dessa disfunção. No caso é necessário investigar os *logs* de serviço para poder identificar o erro. A interface web do Nagios não conta com um console de depuração em tempo real, o que pode tornar a conectividade de dispositivos uma tarefa difícil caso o usuário não tenha conhecimento suficiente dos protocolos envolvidos. Mesmo com estas dificuldades, o objetivo geral do trabalho foi atingido, conseguiu-se estabelecer a conexão do dispositivo IoT com o software Nagios através do uso do protocolo SNMP.

Os dispositivos IoTs fazem parte de uma revolução tecnológica que deixa claro um horizonte cada vez mais repleto de *wearables,* eletrodomésticos, *gadgets* de uso pessoal e automóveis, todos eles partilhando de dados coletados no mundo físico. A maneira que consegue-se extrair essa grande quantidade de dados que estão sendo gerados nestes dispositivos é uma grande questão, pois cada vez mais os dados do plano físico estão sendo transformados em bits*,* e a forma de transformar esta informação de bits para formatos gerenciáveis define o quão importante o gerenciamento de dispositivos IoT se tornou essencial.

O estudo de caso contemplou a coleta de variáveis genéricas, bits que se transformaram em unidades de medida, e este foi o principal objetivo atingido, pois, usando as mesmas métricas e as devidas ferramentas, podem-se transformar quaisquer bits em dados. Ao invés de coletar-se dados de temperatura e umidade podería ter-se usado um sensor barométrico para medir e gerenciar a informação da pressão atmosférica, ou mesmo um detector de fumaça, que através de uma análise

visual pode disparar um contato seco e gerar uma notificação. São inúmeras as possibilidades envolvidas no espectro de IoT.

Além do Nagios, concorre em termos de tecnologia para gerenciamento de redes o software Zabbix, que saiu na frente ao lançar por primeiro uma interface gráfica. Tanto o Nagios como o Zabbix cumprem o que prometem, um com algumas vantagens e outro com algumas desvantagens, mas o peso maior para escolha do Nagios foi a comunidade envolvida e o número de *plugins* que o contempla, *plugins* que conseguem conectar dispositivos nunca antes sequer conectados.

Com a conclusão deste trabalho, dentro de uma atmosfera tecnológica de crescimento exponencial, onde novas tecnologias nascem com velocidade meteórica, é concretizado, atestado e concluído com um excelente *feedback* o uso de uma tecnologia que ninguém tem certeza do seu futuro. Mas sabe-se que, onde ela for empregada, o resultado será satisfatório, porque, além de simples, o SNMP é independente de linguagem, e pode ser inserido em qualquer dispositivo IoT com facilidade, apenas seguindo as métricas já criadas e consolidadas pelo IETF. O horizonte é incerto, tudo pode mudar em muito pouco tempo, porém a tecnologia que cumpre seu papel terá espaço neste futuro próximo.

REFERÊNCIAS

A., MCEWEN; H., CASSIMALLY. **Designing the Internet of Things**. Disponível em: <http://books.google.com.br/books?id=iYkKAgAAQBAJ>>. Acesso: 23 mai. 2018.

ABREU, Fabiano Rocha; PIRES , Herbert Domingues. **Gerência de Redes** . Disponível em:<http://www.midiacom.uff.br/~debora/redes1/pdf/trab042/SNMP.pdf>. Acesso: 26 jun. 2018.

AL-FUQAHA, A. et al.. **Internet of things: A survey on enabling technologies, protocols, and applications**. s, 2015. Disponível em: <https://ieeexplore.ieee.org/document/7123563/>. Acesso: 26 jun. 2018.

ASHTON, Kevin. **hat Internet of Things Thing: In the real world, things matter more t han ideas.** . 2009.. Disponível em: <http://www.rfidjournal.com/articles/view?4986>. Acesso: 26 jun. 2018.

ATZORI, L. **The internet of things: A survey**. Computer Networks, 2010. pp. 2787-2805.

BECKER, Pedro Cristiano; MOURA, Marcos Daniel De. **Utilização da ferramenta Nagios para monitoramento de sinal de antenas de rede wireless**. 2012. Disponível em: <http://sites.setrem.com.br/stin/2012/anais/Pedro.pdf>. Acesso: 26 jun. 2018.

BLACK, Tomas Lovis. **Comparação de ferramentas de gerenciamento de redes**. 2008. Disponível em: <http://www.lume.ufrgs.br/bitstream/handle/10183/15986/000695315.pdf? sequence=1>. Acesso: 11 jun. 2018.

CHEOL-MIN, K.; HYUNG-WOO, K.; SEOK-JOO, K.. **Implementation of CoAP/6LoWPAN over BLE Network for IoT Services** . Pukyong National University,: Pukyong, 2016.

CONTESSA, Diego Fraga; POLINA, Everton Rafael. **Gerenciamento de equipamentos usando o Protocolo SNMP**. Disponível em: <http://paginas.unisul.br/carlos.luz/admredes/unidade1/ArtigoSNMP.pdf>. Acesso: 21 abr. 2018.

DAL BERTO, Jean Carlo; CAGLIARI, Aléssio Inácio; COLLING, Juliane. Automação residencial: protótipo de janela automatizada com Arduíno. **http://ecoinovar.com.br**. Santa Maria, 2017. Disponível em: <http://ecoinovar.com.br/cd2017/arquivos/artigos/ECO1729.pdf> Acesso: 26 jun. 18.
DIAS, Beethovem Zanella; JUNIOR, Nilton Alves. **Protocolo de Gerenciamento SNMP**. 2002. Disponível em: <http://www.rederio.br/downloads/pdf/nt00601.pdf>. Acesso: 21 abr. 2018.

ENDEAVOR. A internet das coisas traz uma grande mudança na forma como fazemos tudo. Leia mais em Endeavor @ https://endeavor.org.br/internet-das- coisas/. **endeavor.org.br/**. 2014. Disponível em: <https://endeavor.org.br/internet- das-coisas/>. Acesso: 26 jun. 2018.

ESTEVES, Antonio Matheus Benaion . **Sistema de monitoramento de redes baseado nos protocolos SNMP e Spanning Tree**. Rio de Janeiro, 2013 Dissertação () - Centro Brasileiro de Pesquisas Físicas, 2013.

EVANS, Dave. A Internet das Coisas Como a próxima evolução da Internet está mudando tudo. **www.cisco.com**. 2011. 4 p. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_i ot_ibsg_0411final.pdf>. Acesso: 26 jun. 2018.

FLEISCH, E. **What is the Internet of Things?** An Economic Perspective. In: Auto-Id labs white paper. 2010.

GOMES, GUSTAVO CARVALHO . **Habilitação de informações de gerenciamento SNMP para o sistema wireless da UFLA**. TCC Ciência da Computação – Universidade Federal de Lavras: Lavras, 2002.

GUPTA, Udit. **Monitoring in IoT enabled devices**: Gupta. Tese. Pittsburgh, Pennsylvania: Carnegie Mellon University, 2014.

ITU, International Telecommunication Union. **ITU internet reports 2005: The Internet of Things**. Geneva, 2005. Disponível em: <http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf>. Acesso: 26 jun. 2018.

JUNIOR, Vanderlei Freitas; COSTA, Gabriel Cesar . **Tecnologia e redes de computadores**. 2016. Disponível em: <http://redes.sombrio.ifc.edu.br/wp- content/uploads/sites/7/2015/12/Livro-Tecnologia-e-Redes-de-Computadores-2016.pdf>. Acesso: 26 jun. 2018.

KHAN, R. et al. “Future internet: The internet of things architecture, possible applications and key challenges.” In: **10th International conference on frontiers of information technology**, 2012. 257-260 p.

KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a Internet: uma abordagem top-down**. 3. ed. São Paulo: Pearson Education do Brasil, 2006.

KUROSE, Jim. **Redes de computadores e a Internet: uma abordagem top-down**. 5. ed. São Paulo: Pearson Education do Brasil, 2013.
LIMA, Otávio Alcantra de; FRESSE, Virginie; ROUSSEA, Frédéricu. **Evaluation of SNMP-like protocol to manage a NoC emulation platform**. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7082776/>. Acesso: 21 abr. 2018.

LUIZ FERNANDO G. , SOARES. **Redes de Computadores:das LANs, MANs e WANs às Redes ATM**. Rio de Janeiro: Campus, 2012.

MARTY, Edson; GONÇALVES, Marcelo Mikosz (Org). **Gerenciador de ambientes computacionais**. TCC (Engenharia da Computação) – UNICENP: Curitiba, 2003.

MAURO, Douglas R.; SCHMIDT, Kevin J. **Essential**: SNMP. United States of America: O'REILLY, 2001.

MORISHITA, Fábio Teruo ; MOREIRA, Edson dos Santos (Org). **Uma avaliação evolutiva dos protocolos de gerenciamento da Internet e suas implementações: SNMPvl, SNMPv2 e SNMPv3**. Tese (Ciências de Computação). Universidade Federal de São Carlos. São Paulo, 1997.

OLIVEIRA, Rafael Rodrigues. **Fundamentos de gerenciamento de redes corporativas e proposta de implementação utilizando SNMP**. 2010. Disponível em: <https://pt.slideshare.net/marleigrolli/snmp-rafael-rodriques>. Acesso: 26 jun. 2018.

PAIVA, Pedro Gustavo de Farias. **Ambiente integrado para gerenciamento da rede interna da Secretaria da Receita da Paraíba**. 2010. Disponível em: <http://www.cin.ufpe.br/~pasg/gpublications/pgfp10-monografia-esp.pdf>. Acesso: 21 abr. 2018.

PATEL, Keyur K; PATEL, Sunil M. Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. **http://ijesc.org/**. Vadodara, Gujarat, India. Disponível em: <http://ijesc.org/upload/8e9af2eca2e1119b895544fd60c3b857.Internet ofThings-IOT Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges.pdf>. Acesso: 26 jun. 2018.

PAULO ANDRÉ, Zapparoli. **Proposta de gestão de disponibilidade através do gerenciamento de rede Internet – um estudo de caso**. Dissertação (Engenharia de Computação). IPT: São Paulo, 2006.

PAVENTHAN, A. et al.. **WSN monitoring for agriculture: Comparing SNMP and emerging CoAP approaches**. 2016. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6757167>. Acesso: 21 abr. 2018.

RFC 3418. Disponível em: <https://tools.ietf.org/html/rfc3418>. Acesso: 26 jun. 2018.

RIZO, Eduardo Henrique. **SMI – Structure of Management Information**. 2011. Disponível em: <http://www.eduardorizo.com.br/2011/10/25/smi-structure-of-management-information/>. Acesso: 21 abr. 2018.

SAITO, Junior Toshiharu. **Desenvolvimento de um modelo de gerenciamento de redes de telecomunicações utilizando a plataforma CORBA**. Tese. UNICAMP: Campinas, 2001.

SANTOS, Cinthia Cardoso dos. **Gerenciamento de redes com a utilização de software livre**. 2009. Disponível em: <http://www3.iesam-pa.edu.br/ojs/index.php/sistemas/article/viewFile/442/374> Acesso: 21 abr. 2018.

SHELBY, Zach; BORMANN, Carsten. **6LoWPAN: the wireless embedded internet .**: Volume 43 de Wiley Series on Communications Networking & Distributed Systems. 1. ed. John Wiley & Sons, 2009.

SHENG, Z.. **Recent Advances in Industrial Wireless Sensor Networks Towards Efficient Management in IoT**. 2015. Disponível em:<https://core.ac.uk/download/pdf/131360637.pdf>. Acesso em: 10 mai. 2018.

SOARES, Alexandre Seixas et al. **O Simple Network Management Protocol (SNMP)**. Rio de Janeiro. Disponível em:<https://www.gta.ufrj.br/grad/10_1/snmp/componentes.htm>. Acesso em: 01 jun. 2018.

SNMP Tutorial Part 2: Rounding Out the Basics. Disponível em: <https://www.networkmanagementsoftware.com/snmp-tutorial-part-2-rounding-out-the-basics/>. Acesso: 26 jun. 2018.

STALLINGS, W.. **SNMP, SNMPv2, SNMPv3, and RMON 1 and 2.**.3. ed. Massachusetts: Addison-Wesley, 1999.

SUTARIA, Ronak ; GOVINDACHARI, Raghunath. **Making sense of interoperability: Protocols and Standardization initiatives in IOT**. 2013.

THE INTERNET of Things. Executive Summary. . 2005. Disponível em: <http://www.itu.int/osg/spu/publicat ions/internetofthings>. Acesso: 26 jun. 2018.

VASSEUR, J. P.; DUNKELS, A. **Interconnecting smart objects with IP. Interconnecting smart objects with IP.** Burlington, 2010. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780123751652000223>. Acesso: 26 jun. 2018.

VILLARI, Massimo et al. AllJoyn Lambda: An architecture for the management of smart environments in IoT. **ieee.org**. Hong Kong, China, 2014. Disponível em: <http://ieeexplore.ieee.org/document/7046676/>. Acesso: 26 jun. 2018.

WU, M. et al. “Research on the architecture of internet of things.” In: 3RD International Conference on Advanced Computer Theory and Engineering (ICACTE). 2010.

WU, X; ZHU, Y; DENG, X. “Design and implementation of embedded snmp network management manager in web – based mode”. In: Proc. IEEEE Asia – Pacific Ser-

vices Computing Conference, APSCC. 2008.

ANEXO A – TABELA MIB DO EQUIPAMENTO STE2

```
-- STE2 MIB 1.01
-- History:
--
-- 1.00 8.9.2015 Marek Hummel - Created
-- 1.01 20.3.2017 Marek Hummel - FIX. STE-MIB DEFINITIONS => STE2-MIB DEFINITIONS


STE2-MIB DEFINITIONS ::=
BEGIN IMPORTS
 OBJECT-TYPE FROM RFC-1212
 enterprises FROM RFC1155-SMI
 DisplayString FROM RFC1213-MIB;

--
-- Type Definitions
--
PositiveInteger ::= INTEGER (1..2147483647) -- 0x7FFF FFFF

UnitType ::= INTE-
  GER { none (0),
  celsius (1),
  fahrenheit (2),
  kelvin
  (3),
  per-
  cent(4)
 }

OnOff ::= INTEGER {
  off (0),
  on (1)
 }

InputAlarmState ::= INTE-
  GER { normal (0),
  alarm (1)
```

```
 }

IOName ::= DisplayString (SIZE (0..16))

SensorState ::= INTEGER {
  invalid (0),
  normal (1),
  outofrangelo (2),
  outofrangehi (3),
  alarmlo (4),
  alarmhi (5)
 }

SensorSN ::= DisplayString (SIZE (0..16))
SensorName ::= DisplayString (SIZE
(0..16)) SensorValue ::= INTEGER
SensorID ::= INTEGER
SensorString ::= DisplayString (SIZE (0..10))

--
-- Node Definitions
--

hwgroup OBJECT IDENTIFIER ::= { enterprises
21796 } x390 OBJECT IDENTIFIER ::= { hwgroup 4 }
ste2 OBJECT IDENTIFIER ::= { x390 9 }

-- Application Info ------------------------------------------------------------

info OBJECT IDENTIFIER ::= { ste2 70 }

infoAddressMAC OBJECT-TYPE
 SYNTAX DisplayString (SIZE
 (0..17)) ACCESS read-only
 STATUS mandatory DE-
 SCRIPTION
  "MAC address in text form.
  It is here to distinguish devices in trap messages."
 ::= { info 1 }
```

```
-- Input Dry Contacts ----------------------------------------------------

inpTable OBJECT-TYPE
 SYNTAX SEQUENCE OF InpEntry
 ACCESS not-accessible
 STATUS       mandatory
 DESCRIPTION
  "A list of binary input entries."
 ::= { ste2 1 }

inpEntry  OBJECT-TYPE
 SYNTAX   InpEntry   AC-
 CESS      not-accessible
 STATUS       mandatory
 DESCRIPTION
  "An entry containing information applicable
  to a particular binary input."
 INDEX { inpIndex }
 ::= { inpTable 1 }

InpEntry ::= SEQUENCE {
 inpIndex   PositiveInteger,
 inpValue OnOff,
 inpName   IOName,   inpAlarm-
 State InputAlarmState
}

inpIndex  OBJECT-TYPE
 SYNTAX PositiveInteger
 ACCESS  not-accessible
 STATUS       mandatory
 DESCRIPTION
  "The binary input index."
 ::= { inpEntry 1 }

inpValue  OBJECT-TYPE  SYN-
 TAX OnOff
 ACCESS        read-only
 STATUS mandatory
```

```
 DESCRIPTION
  "The binary input value."
 ::= { inpEntry 2 }

inpName          OBJECT-TYPE
 SYNTAX IOName
 ACCESS         read-only
 STATUS       mandatory
 DESCRIPTION
  "The binary input name."
 ::= { inpEntry 3 }

inpAlarmState   OBJECT-TYPE
 SYNTAX         InputAlarmState
 ACCESS read-only
 STATUS     mandatory     DE-
 SCRIPTION
  "The binary input alarm state."
 ::= { inpEntry 4 }

-- Sensors ---------------------------------------------------------------

sensTable OBJECT-TYPE
 SYNTAX SEQUENCE OF SensEntry
 ACCESS not-accessible
 STATUS       mandatory
 DESCRIPTION
  "A list of sensor table entries. The number
  of entries corresponds with number of detected sensors."
 ::= { ste2 3 }

sensEntry    OBJECT-TYPE
 SYNTAX   SensEntry   AC-
 CESS  not-accessible  STA-
 TUS  mandatory  DESCRIP-
 TION
  "An entry containing information applicable to a
  particular sensor."
```

```
INDEX { sensIndex }
 ::= { sensTable 1 }

SensEntry ::= SEQUENCE {
 sensIndex  PositiveInteger,
 sensName   SensorName,
 sensState   SensorState,
 sensString  SensorString,
 sensValue   SensorValue,
 sensSN SensorSN, sensU-
 nit UnitType,
 sensID SensorID
}

sensIndex      OBJECT-TYPE
 SYNTAX PositiveInteger AC-
 CESS not-accessible STATUS
 mandatory DESCRIPTION
  "The sensor index."
 ::= { sensEntry 1 }

sensName  OBJECT-TYPE
 SYNTAX      SensorName
 ACCESS read-only STA-
 TUS mandatory DESCRIP-
 TION
  "The sensor name."
 ::= { sensEntry 2 }

sensState  OBJECT-TYPE
 SYNTAX SensorState AC-
 CESS read-only STATUS
 mandatory DESCRIPTION
  "The sensor state."
 ::=  {  sensEntry  3  }
sensString OBJECT-TYPE
```

```
 SYNTAX    SensorString
 ACCESS          read-only
 STATUS        mandatory
 DESCRIPTION
  "The string representation of sensor value."
 ::= { sensEntry 4 }

sensValue    OBJECT-TYPE
 SYNTAX          SensorValue
 ACCESS   read-only   STA-
 TUS  mandatory  DESCRIP-
 TION
  "The integer (decimal * 10) representation
  of sensor value."
 ::= { sensEntry 5 }

sensSN OBJECT-TYPE
 SYNTAX SensorSN
 ACCESS  read-only
 STATUS mandatory
 DESCRIPTION
  "The sensor Serial number."
 ::= { sensEntry 6 }

sensUnit    OBJECT-TYPE
 SYNTAX   UnitType   AC-
 CESS read-only STATUS
 mandatory       DESCRIP-
 TION
  "The sensor unit."
 ::= { sensEntry 7 }

sensID OBJECT-TYPE
 SYNTAX  UnitType  AC-
 CESS   read-only   STA-
 TUS    mandatory    DE-
 SCRIPTION
  "The sensor ID."
 ::= { sensEntry 8 }

END
```

ANEXO B — DESCRIÇÃO OIDS DO EQUIPAMENTO STE2

STE2 SNMP OID description

System Values:

.1.3.6.1.2.1.1.1.0 System Description (string)
.1.3.6.1.2.1.1.2.0 System ObjectID (objid)
.1.3.6.1.2.1.1.3.0 System UpTime (timeticks)
.1.3.6.1.2.1.1.4.0 System Contact (string)
.1.3.6.1.2.1.1.5.0 System Name (string)
.1.3.6.1.2.1.1.6.0 System Location (string)
.1.3.6.1.2.1.1.7.0 System Services (integer)
.1.3.6.1.4.1.21796.4.9.70.1.0 System MAC address (string)

Input Dry Contacts

.1.3.6.1.4.1.21796.4.9.1.1.1.n Input Index (integer, NUM (1..x))
.1.3.6.1.4.1.21796.4.9.1.1.2.n Input Value (integer, 0=Open, 1=Close)
.1.3.6.1.4.1.21796.4.9.1.1.3.n Input Name (string, SIZE (0..16))
.1.3.6.1.4.1.21796.4.9.1.1.4.n Input Alarm State (integer, 0=Normal, 1=Alarm)

Sensors Values, (n = 1..x)

.1.3.6.1.4.1.21796.4.9.3.1.1.n Sensor Index (integer, NUM (1..x))
.1.3.6.1.4.1.21796.4.9.3.1.2.n Sensor Name (string, SIZE (0..16))
.1.3.6.1.4.1.21796.4.9.3.1.3.n Sensor State (integer, 0=Invalid, 1=Normal, 2=OutOfRangeLo, 3=OutOfRangeHi, 4=AlarmLo, 5=AlarmHi)
.1.3.6.1.4.1.21796.4.9.3.1.4.n Sensor String Value (string, SIZE (0..10))
.1.3.6.1.4.1.21796.4.9.3.1.5.n Sensor Value (integer, current value *10)
.1.3.6.1.4.1.21796.4.9.3.1.6.n Sensor SN (string, SIZE (0..16))
.1.3.6.1.4.1.21796.4.9.3.1.7.n Sensor Unit (integer, 0=unknown, 1=°C, 2=°F, 3=°K, 4=%)
.1.3.6.1.4.1.21796.4.9.3.1.8.n Sensor ID (integer, NUM (0..x))