

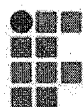


MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE GOIÁS - CÂMPUS URUAÇU

ANTÔNIO ALVES COSTA NETO

**IMPLEMENTAÇÃO DE UM ESQUEMA DE GERENCIAMENTO DE
ENERGIA UTILIZANDO O SNMP**

URUAÇU
2023



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| ☐ Tese | ☐ Artigo Científico |
| ☐ Dissertação | ☐ Capítulo de Livro |
| ☐ Monografia – Especialização | ☐ Livro |
| ☒ TCC - Graduação | ☐ Trabalho Apresentado em Evento |
| ☐ Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: Antônio Alves Costa Neto
Matrícula: 20181050100175
Título do Trabalho: Implementação de um esquema de gerenciamento de energia utilizando o SNMP

Autorização - Marque uma das opções

- ☒ Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
- ☐ Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ____/____/____ (Embargo);
- ☐ Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

- ☐ O documento está sujeito a registro de patente.
☐ O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
☐ Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

- o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Unuaces, 30/06/2023
Local Data

Antônio Alves Costa Neto

Assinatura do Autor e/ou Detentor dos Direitos Autorais



ANTÔNIO ALVES COSTA NETO

IMPLEMENTAÇÃO DE UM ESQUEMA DE GERENCIAMENTO DE ENERGIA UTILIZANDO O SNMP

Projeto do Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Uruaçu ligado ao Ministério da Educação, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Área de concentração: Análise e Desenvolvimento de Sistemas

Linha de pesquisa: Computação Verde

Orientador: Maurílio Humberto Rodrigues Miranda
Instituto Federal de Ciência e Educação de
Tecnologia de Goiás

URUAÇU
2023

Dados Internacionais de Catalogação na Publicação (CIP)

Costa Neto, Antônio Alves.

C837i Implementação de um esquema de gerenciamento de energia utilizando o SNMP
[manuscrito] / Antônio Alves Costa Neto. - 2023.
LIV, 56 f., il.

Orientador: Me. Maurílio Humberto Rodrigues Miranda.

Trabalho de Conclusão de Curso - Instituto Federal de Educação, Ciência e
Tecnologia de Goiás: Campus Uruaçu, Tecnologia de Análise e Desenvolvimento
de Sistemas , 2023.

Bibliografia.

Inclui figuras, lista de figuras e lista de siglas.

1. Tecnologia verde. 2. Energia - gerenciamento. 3. Economia ambiental.
4. Zabbix (Software). 5. Tecnologia da Informação – aspectos ambientais. I.
Miranda, Maurílio Humberto Rodrigues (orientador). II. Instituto Federal de
Educação, Ciência e Tecnologia de Goiás. III. Título.

CDD: 005.8

Ficha catalográfica elaborada pela Bibliotecária Sabrina Gisele da Silva Felix CRB1/2561



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS URUAÇU

Antônio Alves Costa Neto

Implementação de um Esquema de Gerenciamento de Energia Utilizando SNMP

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas ao Departamento de Áreas Acadêmicas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Campus Uruaçu.

Aprovado em 26 de junho de 2023.

(Assinado eletronicamente)

Prof. Me. Maurílio Humberto Rodrigues
Instituto Federal de Goiás - Campus Uruaçu - Orientador

(Assinado eletronicamente)

Prof. Me. Alessandro Siqueira da Silva
Instituto Federal de Goiás - Campus Uruaçu

(Assinado eletronicamente)

Esp. Mirelle Ferreira Alves
Instituto Federal de Goiás - Campus Uruaçu

Documento assinado eletronicamente por:

- Alessandro Siqueira da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 05/07/2023 18:16:22.
- Mirelle Ferreira Alves, TECNICO DE LABORATORIO AREA, em 30/06/2023 17:41:16.
- Maurilio Humberto Rodrigues Miranda, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 30/06/2023 17:22:39.

Este documento foi emitido pelo SUAP em 30/06/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifg.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 426075
Código de Autenticação: 1ba631fb46



Instituto Federal de Educação, Ciência e Tecnologia de Goiás
Rua Formosa, Qds. 28 e 29, Loteamento Santana, URUAÇU / GO, CEP 76400-000
(62) 3357-8179 (ramal: 8179)

RESUMO

ALVES COSTA NETO, Antônio. Implementação de um Esquema de Gerenciamento de Energia Utilizando o SNMP. 2023. 45 f. Projeto do Trabalho de Conclusão de Curso – Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Uruaçu. Uruaçu, 2023.

O presente trabalho discorrerá sobre a importância da tecnologia verde em novas áreas de desenvolvimento e a necessidade de controlar o uso de energia nos sistemas de redes de computadores. A Tecnologia verde vem para trazer uma abordagem de incentivar iniciativas empresariais no rumo da tecnologia, que pode ser aplicada tanto no ramo de *software* e *hardware* e com isso surge a proposta de implementar e gerenciar de uma forma mais efetiva os sistemas de redes com o *software* de código aberto Zabbix, com a ajuda do protocolo *SNMP* (*Simple Network Management Protocol* - Protocolo de Gerenciamento de Rede Simples) da camada de aplicação e o Arduino UNO, que juntos permitem monitorar a energia do sistema que está em execução e comunicar ao administrador caso haja alguma falha, detalhando a hora e quanto tempo durou o problema ocorrido.

Palavras-chave: *SNMP*, TI verde, Zabbix, Energia, Arduino UNO.

ABSTRACT

ALVES COSTA NETO, Antônio. Implementation of an Energy Management Scheme Using SNMP. 2023. 45 f. Projeto do Trabalho de Conclusão de Curso – Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Uruaçu. Uruaçu, 2023.

The present work will discuss the importance of green technology in new areas development and the need to control energy use in grid systems of computers. Green Technology comes to bring an approach to encourage business initiatives in the direction of technology, which can be applied both in the field of software and hardware and with that comes the proposal to implement and manage in a way most effectively network systems with Zabbix open source software, with the help of SNMP protocol (Simple Network Management Protocol) of Simple Network) of the application layer and the Arduino UNO, which together allow monitor the power of the system that is running and communicate to the administrator if there is a failure, detailing the time and how long the problem occurred.

Keywords: *SNMP, Green IT, Zabbix, Energy, Arduino UNO.*

LISTA DE FIGURAS

Figura 1 – Estrutura de comunicação	7
Figura 2 – Estrutura de um <i>PDU</i> do <i>SNMPv2</i>	8
Figura 3 – Criação de um <i>host</i>	9
Figura 4 – Criação de um item	10
Figura 5 – Criação de um <i>trigger</i>	12
Figura 6 – Criação de um <i>template</i>	13
Figura 7 – Formúla de potência	17
Figura 8 – Estrutura de um Arduino UNO	19
Figura 9 – Tela inicial da <i>IDE</i>	20
Figura 10 – Sensor de Corrente Não Invasivo 100A SCT-013	21
Figura 11 – Placa <i>Xbee Shield</i>	22
Figura 12 – Módulo Xbee S2C	23
Figura 13 – <i>Xbee Explorer USB</i>	23
Figura 14 – Módulo <i>XBee</i> identificado pelo <i>XCTU</i>	27
Figura 15 – Acoplando a <i>Xbee Shield</i> ao Arduino UNO	28
Figura 16 – Conectado os sensores ao Arduino UNO físico	29
Figura 17 – Conectado os sensores ao Arduino	30
Figura 18 – Código para leitura da corrente	31
Figura 19 – <i>SCT-013</i> Conectado a uma extensão	32
Figura 20 – Módulo <i>XBee</i> S2C Coordenador sendo reconhecido	33
Figura 21 – Mensuração do consumo energético	33
Figura 22 – Vídeo em 1080p	34
Figura 23 – Vídeo em 1440p	35
Figura 24 – Vídeo em 2160p	35
Figura 25 – Teste em jogos	36
Figura 26 – Instalação do Zabbix	36
Figura 27 – Alterando a senha do servidor Zabbix	37
Figura 28 – Servidor Zabbix	37
Figura 29 – Configuração do host	38
Figura 30 – <i>SNMPD</i> ativo	38
Figura 31 – Atribuindo <i>IP</i> a porta 161	39
Figura 32 – Configurando o <i>SNMPD</i>	39
Figura 33 – <i>IP's</i> com acesso pelo agente <i>SNMP</i>	40
Figura 34 – Resultado da coleta do agente <i>SNMP</i> nas <i>MIB's</i>	40
Figura 35 – Criação do Item para monitoramento da mensuração	41

LISTA DE ABREVIATURAS E SIGLAS

<i>API</i>	<i>Application Programming Interface</i>
IFG	Instituto Federal de Educação, Ciência e Tecnologia de Goiás
<i>MAID</i>	<i>Massive Array of Idle Disks</i>
<i>MIB</i>	<i>Management Information Base</i>
<i>PDU</i>	<i>Protocol Data Unit</i>
<i>VNI</i>	<i>Visual NetWorking Index</i>
<i>SGMP</i>	<i>Simple Gateway Management Protocol</i>
<i>SNMP</i>	<i>Simple Network Management Protocol</i>
<i>SNMPv1</i>	<i>Simple Network Management Protocol</i> versão 1
<i>SNMPv2</i>	<i>Simple Network Management Protocol</i> versão 2
<i>SNMPv3</i>	<i>Simple Network Management Protocol</i> versão 3
<i>TCO</i>	<i>Total Cost of Ownership</i>
<i>TI</i>	<i>Information Technology</i>
<i>C</i>	<i>Coulumb</i>
<i>HZ</i>	<i>Hertz</i>
<i>V</i>	<i>Volt</i>
<i>USB</i>	<i>Universal Serial Bus</i> Universal Serial Bus
<i>LED</i>	<i>Light Emitting Diode</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>UDP</i>	<i>User Datagram Protocol</i> User Datagram Protocol
<i>TCP</i>	<i>Transmission Control Protocol</i>
<i>SD</i>	<i>Secure Digital</i>
<i>DNS</i>	<i>Domain Name System</i>

<i>CO₂</i>	Dióxido de carbono
<i>SGBD</i>	<i>Data Base Management System</i>
<i>EUA</i>	<i>United States</i>
<i>SSD</i>	<i>Solid State Drive</i>
<i>TLS</i>	<i>Transport Layer Security</i>
<i>AC</i>	Corrente Alternada
<i>DC</i>	Corrente Contínua
<i>SMS</i>	<i>Short Message Service</i>
<i>A</i>	<i>Ampère</i>
<i>W</i>	<i>Watt</i>
<i>E</i>	Energia
<i>CPU</i>	<i>Central Processing Unit</i>
<i>GPU</i>	<i>Graphic Processing Unit</i>
<i>SI</i>	Sistema Internacional de Unidades
<i>MAC</i>	Media Access Control

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 Objetivos	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
1.2 Justificativa	2
1.3 Organização	3
2 – FUNDAMENTAÇÃO TEÓRICA	4
2.1 Computação Verde	4
2.1.1 <i>Data Centers</i> verdes	4
2.1.2 Computação em Nuvem verde	4
2.1.3 Armazenamento de dados verdes	5
2.1.4 <i>Softwares</i> verdes	5
2.1.5 Comunicações e redes verdes	5
2.2 <i>SNMP</i>	5
2.2.1 <i>SNMPv1</i>	7
2.2.2 <i>SNMPv2</i>	7
2.2.3 <i>SNMPv3</i>	8
2.3 Zabbix	8
2.3.1 Host	9
2.3.2 Item	10
2.3.3 <i>Trigger</i>	10
2.3.4 Template	12
2.4 Energia	13
2.4.1 Tipos de Energia	13
2.4.2 Características de Energia Elétrica	15
2.4.3 Tensão Elétrica (Voltagem)	16
2.4.4 Eficiência Energética	17
2.5 Arduino	18
2.5.1 Arduino UNO	18
2.5.2 Ambiente de desenvolvimento integrado (IDE)	19
2.5.3 Sensores	20
2.5.3.1 Sensor de Corrente Não Invasivo 100A SCT-013	21
2.5.4 <i>Xbee Shield</i>	21
2.5.4.1 Protocolo Zigbee	22
2.5.4.1.1 Módulo Xbee S2C	22

2.5.4.1.2	Xbee Explorer USB	23
3	– METODOLOGIA	24
3.1	Abordagem Metodológica	24
3.2	Delineamento do Estudo	24
3.3	Coleta de Dados	24
3.4	Análise dos Dados	25
4	– IMPLEMENTAÇÃO	26
4.1	Configurando os Módulos XBee S2C	26
4.1.1	Módulo <i>XBee</i> S2C Coordenador	27
4.1.2	Módulo <i>XBee</i> S2C Roteador	27
4.2	Configurando o Arduino UNO	28
4.3	Coleta dos Resultados	34
4.4	Configurando o Zabbix	36
5	– CONSIDERAÇÕES FINAIS	42
	Referências	43

1 INTRODUÇÃO

Com o surgimento das novas aplicações da tecnologia, a movimentação de dados já ultrapassou a escala de exabytes. De acordo com o *Cisco Annual Internet Report* (Relatório Anual da Cisco sobre Internet), neste ano de 2023, estima-se que 29.3 bilhões de dispositivos estejam conectados a *Internet*, há mais tráfego a atravessar as redes globais do que em todos os “anos de *internet*” anteriores até ao ano de 2016 (CISCO, 2020). E isso virá de todos nós, dos nossos equipamentos e da forma de como utilizamos a *internet*. A eletricidade consumida anualmente pela movimentação global de dados em 2016 estava estimada em 100 terawatts-hora, gerando um custo de mais de 20 bilhões de dólares por ano (CISCO, 2014).

Cada vez mais os custos de energia estão se tornando um fator importante no *TCO* (*Total Cost of Ownership*- Custo Total de Propriedade) tanto dos *Data Centers* (Centro de armazenamento de dados) quanto da computação das empresas. A soma dos custos de *hardware*, *software*, pessoal e energia gera o *TCO*. O uso de energia de equipamentos de *TI* (*Information Technology* - Tecnologia da Informação), também gera um impacto ambiental que deve ser analisado, pois consome muita matéria-prima. Grandes organizações evidenciam o interesse em reduzir a marca de CO_2 deixada pelos equipamentos de *TI* (FRANCISCO; CHRISTENSEN, 2008).

A *TI Verde*, também conhecida como computação verde é o estudo e a prática de projetar, fabricar e usar computadores, servidores, monitores, impressoras, dispositivos de armazenamento e sistemas de rede e comunicação de forma eficiente e eficaz. Outro conceito que envolve a *TI Verde*, refere-se a tecnologias e sistemas de informação ambientalmente saudáveis. Existem três abordagens para melhorar a sustentabilidade ambiental na *TI*: o *design*, fabricação, uso e descarte eficiente e eficaz de *hardware* dos computadores, *software* e sistemas de comunicação com nenhum ou mínimo impacto no meio ambiente. Empresas entusiastas em tecnologia devem aproveitar o poder da *TI* e sistemas de informação para capacitar, ou seja, apoiar, auxiliar e alavancar outras iniciativas ambientais, ajudar a criar consciência entre as partes interessadas e promover mais iniciativas verdes de *TI* para inovar ambientes com processos de eficiência energética e práticas de negócios ambientalmente sustentáveis (MURUGESAN; GANGADHARAN, 2012).

Com as novas tecnologias, é possível reduzir significativamente o consumo de energia, basta alterar a forma de como os *hardwares* e *softwares* são utilizados. A maioria dos *hardwares* funcionam mesmo quando não estão sendo usados, porque nós, usuários, os deixamos ligados desnecessariamente, consumindo eletricidade. Com algumas medidas, como habilitar o recurso de gerenciamento de energia, desligar o sistema quando não estiver em uso, usar protetores de tela, faz com que o desperdício de energia seja reduzido sem prejudicar o desempenho da máquina (MURUGESAN; GANGADHARAN, 2012).

O *software* Zabbix, apesar de já está a muito tempo no mercado, é consolidado no ramo empresarial de redes, apresenta atualizações constantes e eficientes. Além de possuir código aberto, ele também apresenta um sistema bem completo de monitoramento, pois possui suporte a alguns *SGBD's* (Sistema de Gerenciamento de Banco de Dados), oferece suporte a vários sistemas operacionais, possui integração com Arduino e outros softwares.

Com essas funcionalidades citadas é possível através da integração entre o Zabbix e o Arduino, medir o consumo energético de um sistema. Para que a comunicação aconteça é necessário baixar uma biblioteca chamada *Agentuino*, pois a biblioteca permite que o protocolo *SNMP* (*Simple Network Management Protocol* - Protocolo de gerenciamento de redes simples), que já vem junto com o pacote do Zabbix, colete informações e através do *MIB* (*Management Information Base*- Base de Informação de Gerenciamento), informe sobre anomalias ao que ocorrem no dispositivo gerente.

1.1 Objetivos

Nesta seção, serão apresentados os objetivos gerais e específicos do presente estudo. Esses objetivos fornecem uma direção clara para a pesquisa e definem o propósito geral do trabalho.

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é implementar um esquema de monitoramento de consumo energético através do Zabbix com o Arduino UNO e seus sensores, com o intuito de auxiliar o gerenciamento do consumo energético dos sistemas a ser monitorados.

1.1.2 Objetivos Específicos

- Analisar o consumo energético de um ambiente de TI;
- Aumentar a eficiência energética de um ambiente de TI;
- Contribuir com o desenvolvimento de tecnologias verdes.

1.2 Justificativa

A necessidade de uma gestão eficaz e consciente do consumo energético tem se tornado cada vez mais relevante devido às crescentes preocupações com a sustentabilidade e a redução dos impactos ambientais. Ao monitorar e analisar o consumo energético dos sistemas, é possível identificar padrões de consumo, gargalos e comportamentos anômalos, permitindo a implementação de medidas corretivas e a otimização do uso de energia.

O uso do Zabbix, uma plataforma de monitoramento amplamente reconhecida, em conjunto com o Arduino UNO e seus sensores, proporciona uma solução eficiente e de baixo custo para o monitoramento em tempo real do consumo energético. Com essa

implementação, será possível obter dados precisos sobre o consumo de energia dos sistemas monitorados, permitindo a identificação de oportunidades de economia e ações direcionadas para a redução do consumo.

Portanto, este estudo justifica-se pela necessidade de aplicar um esquema de monitoramento de consumo energético utilizando o Zabbix e o Arduino UNO, com o objetivo de auxiliar o gerenciamento do consumo energético dos sistemas a serem monitorados. Ao promover a conscientização sobre o consumo de energia e fornecer informações relevantes para a tomada de decisões, espera-se contribuir para a promoção da eficiência energética e a adoção de práticas sustentáveis.

1.3 Organização

O presente trabalho está organizado da seguinte forma. O capítulo 2 é a fundamentação teórica. A seção 2.1 discorrerá sobre alguns dos principais tópicos estudados na Computação Verde, tais como *Data Centers*, computação em nuvem, armazenamento de dados, *softwares*, redes de comunicação, no qual todos possuem enfoque nas tecnologias verdes. A seção 2.2 é sobre o *SNMP* no qual é discorrido sobre o protocolo e posteriormente sobre suas versões e melhorias que foram incrementadas com o decorrer das atualizações. Na seção 2.3 é abordará sobre o Zabbix e suas funcionalidades. A seção 2.4 explicará sobre energia e seus tipos, suas características, corrente alternada e contínua, tensão e potência. Na seção 2.5 o assunto será o Arduino e seus sensores. O capítulo 3 é explicada a metodologia para desenvolver a implementação do esquema. O capítulo 4 é parte da implementação, detalhada de forma que possa se seguir os passos para replicação dos resultados. Já no capítulo 5, consta as considerações finais, os pontos importantes a serem analisados e o que foi absolvido com toda essa implementação.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo vamos fundamentar sobre os tópicos que englobam computação verde, *SNMP*, *Zabbix*, *Arduino* e Energia, para que se possa entender mais sobre eles e posteriormente gerar a implementação do protocolo de forma eficiente.

2.1 Computação Verde

TI verde é um termo que engloba as tecnologias de informação ambientalmente corretas sobre sistemas, aplicativos e práticas a serem tomadas. Com isso algumas dúvidas de como deveríamos tornar a *TI* mais ecológica e quais subsistemas de *TI* podem ser tecnologicamente mais sustentáveis. É fato, que todo subsistema e periférico de *TI* pode ser mais eficientes. Isso acaba englobando os computadores, *notebooks* e servidores, *Data Centers* e computação em nuvem, *software*, sistemas de armazenamento e sistemas de protocolos de rede e comunicação (MURUGESAN; GANGADHARAN, 2012).

2.1.1 *Data Centers* verdes

Os *Data Centers* verdes possuem sistemas de *TI*, sistemas de ar condicionado, sistemas elétricos e mecânicos e os prédios que o abrigam para operar na máxima eficiência energética, com redução na emissão de gás carbônico e mínimos impactos ambientais. Os *Data Centers* de empresas multinacionais, possuem um grande número de computadores servidores e na última década, esse número aumentou seis vezes, para entorno de 30 milhões e a energia total usada pelos servidores representou 0,6% e com a energia necessária para o resfriamento e outros serviços auxiliares, o uso de eletricidade sobe para 1,2,% do consumo total de eletricidade nos EUA em 2005 (MURUGESAN; GANGADHARAN, 2012).

2.1.2 Computação em Nuvem verde

A computação em nuvem é uma transição da computação como produto para a computação como serviço, compartilhada e escalável sob demanda. Para atender às crescentes demandas dos serviços de computação em nuvem, os fornecedores usam *Data Centers* em larga escala que consolidam milhares de servidores com outra infraestrutura, como redes de refrigeração, armazenamento e comunicação. No entanto o crescimento de armazenamento em nuvem gera um aumento do consumo de energia ocasionados pelos *Data Centers*. A computação em nuvem é uma solução verde, pois a infraestrutura em nuvem engloba dois elementos críticos de uma *TI* verde: eficiência de recursos e eficiência de energia (MURUGESAN; GANGADHARAN, 2012).

2.1.3 Armazenamento de dados verdes

Em um *Data Center* os sistemas de armazenamento consomem entre 24% e 40% do uso total de energia de *TI* e são os maiores consumidores de energia dos *Data Centers*. Existem várias abordagens que tem como foco melhorar a eficiência do armazenamento de dados, sendo elas: *MAID* (*Massive Array of Idle Disks* - Matriz Massiva de Discos Ociosos), redução de disco, armazenamento em camadas e unidades de *SSD's* (*Solid State Drive* - Disco de Estado Sólido) são usados para melhorar a eficiência energética e reduzir os custos gerais de armazenamento de dados persistentes. Um fator relevante é a deduplicação de dados que gera a eliminação de dados redundantes de granulação grossa. Normalmente ocorre na linha em que os dados estão duplicados e alocados, à medida que os dados estão fluindo ou após o processo ter sido gravado, gerando somente o armazenamento de dados únicos (MURUGESAN; GANGADHARAN, 2012).

2.1.4 Softwares verdes

O *software* desempenha um papel importante na determinação do consumo geral de energia e eficiência computacional. Um *software* ineficiente em termos de computação pode frustrar todos os benefícios de gerenciamento de energia incorporados ao *hardware*. Os *softwares* podem ser classificados em quatro categorias, sendo elas: *Software* ecológico reduz o consumo energético ao ser executado; *Software* incorporado que ajuda outras ferramentas a evitarem o consumo exagerado de energia; *Software* de relatórios de sustentabilidade que busca elencar atributos relacionados ao desenvolvimento, como atributos de modificabilidade, reutilização, portabilidade e desempenho, como tempo e eficiência computacionais, usabilidade e confiabilidade do *software* (MURUGESAN; GANGADHARAN, 2012).

2.1.5 Comunicações e redes verdes

As redes de comunicações nos permitem comunicar e compartilhar informações, comprar, aprender e socializar de forma *on-line*, que por consequência torna o nosso ambiente de trabalho mais inteligente e nos permite fazer várias atividades. Os sistemas tradicionais de rede e os protocolos de comunicação não são especialmente projetados para obter eficiência energética, com isso se torna necessário a utilização de rede verde que trás maneiras de minimizar o impacto das redes no ambiente, usando tecnologias, produtos e protocolos (MURUGESAN; GANGADHARAN, 2012).

2.2 SNMP

O *SNMP* foi desenvolvido para ajudar no monitoramento e gerenciamento de dispositivos conectados a rede. Ele se trata da evolução do protocolo *SGMP* (*Simple Gateway Management Protocol* - Protocolo de monitoramento de porta simples), que servia apenas para monitorar roteadores de Internet (SOARES, 2011).

O *SNMP* é composto por três elementos: *MIB*, agente e gerente. O *MIB*, é uma base de dados estatísticos e de controle, usado para descrever o status de um agente. O agente está presente em cada nó de uma rede, quando executado ele envia informações coletadas via *SNMP* para o gerente. O gerente recebe as informações coletadas pelos agentes e faz requisições em caso de necessidade através de comandos como:

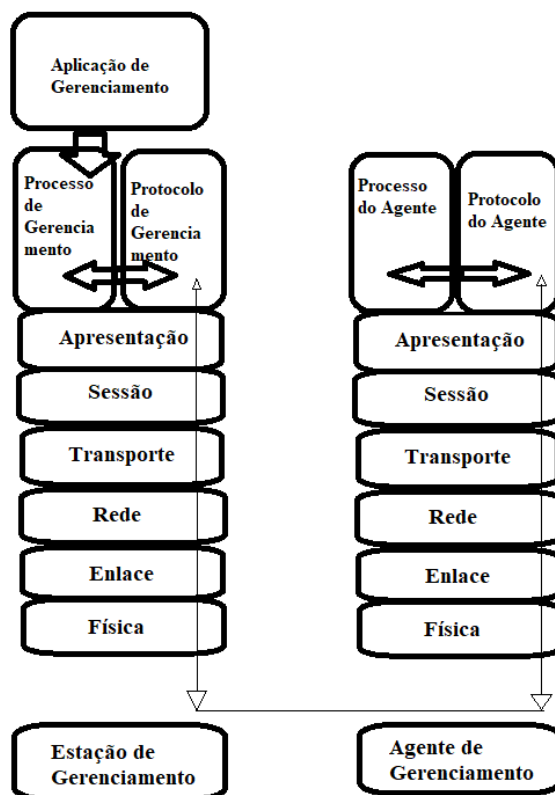
- *Get Request*: Permite obter um registro;
- *Get Next Request*: Usado para obter o registro seguinte de uma lista ou um bloco de registro;
- *Get Response*: Usado pelo agente para responder que dados deve obter e definir os pedidos do gestor;
- *Set Response*: Usado para iniciar e fazer uma mudança para um valor do elemento de rede;
- *GetBulk Request*: Usado como alternativa para fazer uma única retorno de vários *Get Next Request*;
- *Trap*: Que é uma mensagem gerada e transmitida pelo agente para alertar o gerente de alguma possível alteração no sistema.

O *SNMP* consiste em alguns padrões para o gerenciamento da rede, possuindo um protocolo na camada de aplicação, um *software* de banco de dados e um conjunto de objetos. Ele basicamente trata os dados de gerenciamento como variáveis pelo seu sistema, podendo assim descrever a configuração do ambiente a ser observado (SOARES, 2011).

O fato das variáveis poderem ser lidas e até escritas pelas aplicações de gerenciamento, permitem que as configurações do dispositivo da rede sejam modificadas. Quando se trata de uma aplicação tradicional, onde se tem uma série de sistemas para serem gerenciados, um ou mais sistemas podem ser tornar responsáveis por administrar os demais. Para que se tenha as informações de rede dos sistemas o *SNMP* deixa um componente normalmente denominado como agente para coletar e enviar os dados para o sistema gerenciador, quando exposto os dados, tendem a ficar organizados mostrando as rotas, o nome do sistema, memória disponível e o número de processos ativos (SOARES, 2011).

A comunicação entre o agente e o gerente se dá pelo uso de módulos de *softwares* que são adicionados em cada um desses. O primeiro deles é o responsável por habilitar o transporte da informação de gerenciamento dos dispositivos e por implementar o protocolo *SNMP*. Já o outro, é chamado de processo de gerenciamento no administrador e processo de agente no agente. Os serviços relacionados ao gerenciamento de processos têm a responsabilidade de realizar a gestão dos programas na camada de aplicação, além de fornecer a interface para o protocolo de gerenciamento da rede. O subagente desempenha o papel de integrador dos processos que acessam as informações solicitadas pela aplicação de gerenciamento da rede, atuando como uma interface para o protocolo de gerenciamento em questão. Essa comunicação segue o exemplo da figura 1 (SOARES, 2011).

Figura 1 – Estrutura de comunicação



Fonte: Soares (2011)

2.2.1 *SNMPv1*

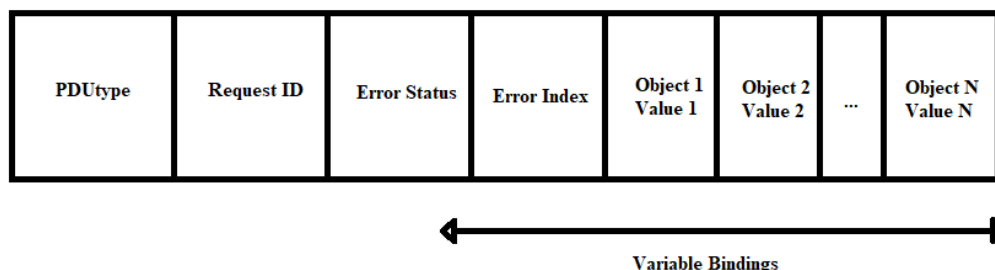
O *SNMPv1* representa a primeira versão do protocolo *SNMP*, caracterizada por possuir um esquema de autenticação considerado fraco, no qual o único mecanismo de segurança disponível são os "nomes de comunidade". Um nome de comunidade é um grupo de gerenciamento que possui permissões específicas, permitindo a utilização das instruções *SET* e *GET* em relação a um determinado parâmetro para os membros dessa comunidade. Os nomes de comunidade são armazenados localmente, o que implica que cada agente que implementa o *SNMP* deve registrar as permissões concedidas a cada comunidade de gerenciamento para o uso de seus parâmetros. Uma falha significativa identificada nessa versão é que qualquer pessoa que tenha conhecimento do nome de comunidade e dos privilégios necessários pode enviar comandos *SNMP* pela rede (SOARES, 2011).

2.2.2 *SNMPv2*

O modelo *SNMPv2* trouxe algumas melhorias sendo elas a inclusão de *bit strings*, endereços de rede e contadores. As *bit strings* são definidas apenas nessa versão e engloba zeros ou mais *bits* nomeados que especificam um valor. A comunicação entre agentes e gerentes em uma rede *SNMP* é baseada nas especificações das chamadas *PDU* (*Protocol*

Data Unit - Unidades de Dados de Protocolo)(IGOR; BRUNO, 2011). Todas essas PDUs seguem a estrutura da seguinte figura:

Figura 2 – Estrutura de um *PDU* do *SNMPv2*



Fonte: Igor Drummond e Bruno Lange (2011)

2.2.3 *SNMPv3*

O novo modelo foi desenvolvido com o intuito de solucionar as lacunas de segurança presentes nas versões anteriores. Sua finalidade é proporcionar uma série de vantagens, garantindo maior proteção aos usuários. Ele conta com dois componentes principais de segurança: o Modelo de Segurança Baseado em Usuário e o Modelo de Controle de Acesso Baseado em Visões. O modelo de segurança baseado em usuário foi elaborado para salvaguardar os pacotes do *SNMPv3* contra ameaças, como a modificação de mensagens, alteração do fluxo de mensagens e divulgação indevida de informações. Por sua vez, o modelo de controle de acesso baseado em visões é empregado para determinar os direitos de acesso de um usuário remoto em relação à visualização ou alteração de um valor na *MIB* local (IGOR; BRUNO, 2011).

2.3 Zabbix

A ferramenta Zabbix é um software de código aberto desenvolvido pelo Alexei Vladishev e atualmente é atualizado e distribuído pela Zabbix SIA. Zabbix é amplamente usado por empresas para o monitoramento de sistema, pois permite visualizar diversos parâmetros de rede de computadores e saúde e integridade dos servidores. Esse programa permite o administrador, através de pré-configurações, supervisionar toda a infraestrutura do sistema em que a ferramenta está atuando, pois a mesma permite mapear todos os dados, podendo detalhar através *widgets* personalizáveis e *dashboards* os gráficos que estão sendo monitorados, mas para isso é necessário criar um *Host*, agrupar itens e atribuir um *template* para ele (ZABBIX, 2022a). Essa ferramenta também conta com o protocolo *TLS* (*Transport Layer Security* - Segurança da Camada de Transporte), que tem como objetivo facilitar a segurança e privacidade de dados na *internet*, pois ele criptografa a

comunicação entre os computadores e o servidor de hospedagem no momento em que um domínio é acessado (MICROSOFT, 2022).

No Zabbix, existem algumas ferramentas denominadas *Host*, *Item*, *Trigger* e *Template*, por conta delas, o administrador consegue definir o que vai ser monitorado, que é definida no *Host*, o que devemos buscar de informação e quando essa informação deve ser coletada, que é configurada no *Item*, estabelecer parâmetros de até onde é considerado aceitável aqueles dados chegarem, que é estabelecido na *Trigger*. Já o *template*, tem como funcionalidade automatização de forma pratica, permitindo o administrador acoplar todas as configurações feitas em um *Host* e associar a outro.

2.3.1 Host

O *Host* no Zabbix é definido como uma entidade na rede, podendo ser física ou virtual que o administrador deseja monitorar. A definição do que é um *Host* na aplicação é muito flexível, pode ser um servidor real, um *switch*, uma máquina virtual ou até mesmo uma aplicação ou serviço. A figura a seguir, demonstra os campos que devem ser preenchidos para se criar um *host*, devendo associar obrigatoriamente um nome e um grupo de *hosts* (ZABBIX, 2022c).

Figura 3 – Criação de um *host*

A interface de criação de um host no Zabbix apresenta uma barra de navegação superior com as seguintes opções: Host (selecionada), IPMI, Etiquetas, Macros, Inventário, Criptografia e Mapeamento de valor. O formulário principal contém os seguintes campos:

- * Nome do host**: Campo de texto obrigatório.
- Nome visível**: Campo de texto opcional.
- Templates**: Campo de texto com o placeholder "informe aqui o argumento para pesquisa".
- * Grupos de hosts**: Campo de texto obrigatório com o placeholder "informe aqui o argumento para pesquisa".
- Interfaces**: Exibe o texto "No interfaces are defined." e um link "Adicionar".
- Descrição**: Área de texto grande para a descrição do host.
- Monitorado por proxy**: Menu suspenso com a opção "(sem proxy)".
- Ativo**: Caixa de seleção marcada com um ícone de checkmark.

No rodapé do formulário, há dois botões: "Adicionar" (em azul) e "Cancelar" (em branco).

Fonte: BANZI e COL. (2022a)

2.3.2 Item

No Zabbix, um item é uma entidade que representa uma métrica que você deseja monitorar, como uso de *CPU* (*Central Processing Unit* - Unidade Central de Processamento), uso de memória, tráfego de rede, disponibilidade de um serviço, entre outros. Ele é responsável por coletar e armazenar os dados relevantes para posterior análise e exibição de gráficos. A criação de um item no Zabbix envolve os seguintes passos: Selecionar o *host* para o qual se deseja adicionar o item, adicionar um nome; a chave que é usada para recuperar as informações coletadas pelo agente do Zabbix ou por outros métodos de monitoramento suportados; o tipo de item que retorna o valor em numérico e caractere, e posteriormente o intervalo de atualização, que é a frequência em que os dados serão atualizados. Segue figura 4 para exemplificar os campos em que os dados devem ser inseridos. (ZABBIX, 2022d).

Figura 4 – Criação de um item

The screenshot shows the 'Create Item' form in Zabbix. The form is divided into two tabs: 'Item' (selected) and 'Pré-processamento'. The 'Item' tab contains the following fields:

- * Nome**: A text input field.
- Tipo**: A dropdown menu with 'Agente Zabbix' selected.
- * Chave**: A text input field.
- Tipo de informação**: A dropdown menu with 'Numérico (inteiro sem sinal)' selected.
- * Interface do host**: A dropdown menu with '127.0.0.1:10050' selected.
- Unidades**: A text input field.
- * Intervalo de atualização**: A text input field with '1m' entered.
- Intervalo customizado**: A section with two sub-sections: 'Tipo' (with 'Flexível' and 'Agendamento' buttons, 'Flexível' is selected) and 'Intervalo' (with a text input field containing '50s'). Below this is a blue 'Adicionar' button.
- * Período de retenção do histórico**: A section with two sub-sections: 'Do not keep history' and 'Storage period' (with a text input field containing '90d').
- * Período de retenção das estatísticas**: A section with two sub-sections: 'Do not keep trends' and 'Storage period' (with a text input field containing '365d').

Fonte: BANZI e COL. (2022a)

2.3.3 Trigger

A *trigger* tem como objetivo avaliar automaticamente os dados, ela possui uma expressão que define o valor limite para o dado recebido. Se esse valor for ultrapassado pelo dado recebido, a *trigger* irá ser acionada e irá para o estado de incidente, informando ao administrador que algo ocorreu e que necessita de atenção. Após a correção do incidente

a *trigger* volta para o estado normal. A *trigger* permite na sua configuração, definir qual é o nível do problema que foi encontrado, podendo ser não classificado, informativo, alerta, perigo e alto risco (ZABBIX, 2022b).

Para criar uma *trigger* no Zabbix é necessário ir na opção do Zabbix chamada **configuração** no menu principal e selecionar **gatilhos** na subseção *Host*, clicar na opção **criar gatilho** para adicionar uma nova *trigger*, e preencher os seguintes campos:

- Nome: Um nome descritivo para a *trigger*.
- Expressão: A expressão define a condição que aciona a *trigger*. Ela é construída usando funções, operadores lógicos e os valores dos itens monitorados.
- Severidade: Escolhe o nível de severidade associado ao alerta gerado pela *trigger*, como informação, aviso, médio, alto ou desastre.
- Valor de recuperação: Campo para definir uma expressão para determinar quando o problema associado à *trigger* foi resolvido e o alerta pode ser considerado como recuperado.

Após a criação da *trigger*, o Zabbix monitorará continuamente os itens associados e avaliará as condições definidas na expressão da *trigger*. Quando uma condição é atendida, a *trigger* é acionada e um alerta é gerado. Com essas notificações geradas pelas *triggers*, a ferramenta Zabbix permite que o administrador possa configurar para os alertas chegarem no seu celular via *SMS* (*Short Message Service* - Serviço de Mensagens Curtas), *e-mail* ou outros aplicativos de mensagens (ZABBIX, 2022e). Segue figura 5, para exemplificar a criação de um gatilho.

Figura 5 – Criação de um *trigger*

The screenshot displays the Zabbix web interface for creating a new trigger. The 'Trigger' tab is selected. The configuration fields are as follows:

- Name:** High CPU utilization (over {CPU.UTIL.CRIT}% for 5m)
- Event name:** High CPU utilization (over {CPU.UTIL.CRIT}% for 5m)
- Operational data:** Current utilization: {ITEM.LASTVALUE1}
- Severity:** Not classified, Information, **Warning**, Average, High, Dis
- Expression:** {New host:system.cpu.util.min(5m)}>{CPU.UTIL.CRIT}
- Expression constructor:** (link)
- OK event generation:** Expression, Recovery expression, None
- PROBLEM event generation mode:** Single, Multiple
- OK event closes:** All problems, **All problems if tag values match**
- Tag for matching:** (empty field)
- Allow manual close:** ☐
- URL:** (empty field)
- Description:** CPU utilization is too high. The system might be slow to respond.
- Enabled:** ☒

Fonte: BANZI e COL. (2022a)

2.3.4 Template

No Zabbix, os *templates* desempenham um papel crucial na automação, pois permitem a organização de itens, *triggers* e outras entidades de maneira reutilizável e facilmente vinculável a hosts. Quando são ativados, os *templates* facilitam a agrupação de itens, *triggers* e outras entidades, permitindo uma associação simplificada aos hosts. Ao serem vinculados a um *host*, todas as entidades presentes no template são importadas para o *host* em questão. Essencialmente, temos um modelo de monitoramento pronto para ser aplicado rapidamente a um novo *host* (ZABBIX, 2022f).

Para criar um template é necessário acessar a interface *web* do Zabbix, clicar em **configuração**, **templates** e na opção **criar template**. Será aberto uma nova aba com alguns campos a serem preenchidos, sendo eles:

- Nome: Para reconhecer o *template*.

- Grupos de *hosts*: Selecione um ou mais grupos de *hosts* aos quais o *template* será associado. Isso determina quais *hosts* poderão usar o *template*.
 - Visibilidade: Define a visibilidade do *template*, que pode ser público ou particular.
- A figura abaixo é um exemplo de criação de *template*.

Figura 6 – Criação de um *template*

A interface de criação de um template apresenta o seguinte layout:

- Templates** (título da seção)
- Abas: **Templates** (selecionada), Etiquetas, Macros, Mapeamento de valor
- * Nome do template: [campo de texto]
- Nome visível: [campo de texto]
- Templates: [informe aqui o argumento para pesquisa] [ícone de lupa]
- * Template groups: [informe aqui o argumento para pesquisa] [ícone de lupa]
- Descrição: [área de texto grande]
- Botões: **Adicionar** (em azul) e **Cancelar** (em branco com contorno azul)

Fonte: BANZI e COL. (2022a)

2.4 Energia

A energia é utilizada constantemente por nós de diferentes maneiras, tanto nos motores de máquinas, como no nosso corpo. Por ser um termo muito amplo, as pesquisas relacionadas a questão energética é vasta, envolvendo desde o uso de recursos naturais até os aspectos da eficiência das novas tecnologias (CARVALHO; COL., 2012).

De acordo com a física a energia é uma grandeza física que se conserva. Na concepção atual, energia corresponde ao conceito desenvolvido juntamente com a Termodinâmica a partir do início do Século XIX, em outros termos, entende-se como a capacidade de realizar uma ação (TORRES, 2012).

2.4.1 Tipos de Energia

A energia se apresenta de diversas formas, que podem ser transformadas. É importante ressaltar que apenas nos processos de conversão pode se notar a existência de energia, pois surge na divisão do sistema como calor ou como trabalho. Calor é definido como o fluxo energético decorrente de diferença de temperatura. Trabalho é todo processo análogo à elevação de um peso (CARVALHO; COL., 2012).

Entende-se como recursos energéticos as reservas de energias renováveis e não renováveis, sendo classificadas da seguinte maneira:

- Fontes não renováveis: São recursos que se esgotam mais rapidamente do que se repõem. Exemplos são os derivados de petróleo, os elementos radioativos, energia geotérmica e o gás natural. Para que ocorra a transformação dessas fontes primária em eletricidade, utilizamos combustão, fissão ou processos geotérmicos. Com esse meio, obtemos a geração elétrica conhecida como geração termelétrica (REIS, 2015).
- Fontes renováveis: São os recursos que se reabastecem de forma mais rápida do que são consumidas. Exemplos desse meio são as hídrica, solar, eólica, geotérmica e biomassa. Algumas dessas fontes precisam passar por tratamentos e operações em usinas que não sejam maiores que a sua vida útil (REIS, 2015).

Abaixo, está descrito alguns exemplos de energia, sendo elas:

- Energia potencial elástica: É uma forma de energia armazenada em objetos que possuem a capacidade de voltar à sua forma original após terem sido deformados. Ela está relacionada à elasticidade de um material e depende da quantidade de deformação sofrida. Quanto maior for a deformação, maior será a energia potencial elástica armazenada. Essa energia é liberada quando o objeto retorna à sua forma original. Um exemplo comum é a energia potencial elástica em uma mola comprimida ou esticada. A energia potencial elástica pode ser convertida em outras formas de energia quando o objeto é liberado, como energia cinética ou trabalho mecânico (HALLIDAY; COL., 2009).
- Energia elétrica: É uma forma de energia resultante do movimento ordenado de cargas elétricas através de um condutor. É gerada a partir de fontes como usinas hidrelétricas, termoeletrônicas, solares, eólicas, entre outras. A energia elétrica é amplamente utilizada em residências, indústrias e em diversos setores da sociedade. Ela pode ser convertida em outras formas de energia, como luz, calor, movimento mecânico e som, e é transportada por meio de redes de distribuição até os locais de consumo. A unidade de medida da energia elétrica é o kilowatt-hora (kWh)(HALLIDAY; COL., 2009).
- Energia térmica: É a forma de energia associada à temperatura de um sistema. Ela está relacionada ao movimento das partículas que compõem um objeto ou substância. Quanto mais rápido esses movimentos, maior é a energia térmica. A energia térmica pode ser transferida entre sistemas por meio de condução, convecção ou radiação. É uma forma de energia que pode ser convertida em outras formas, como trabalho mecânico ou energia elétrica. A energia térmica é amplamente utilizada em processos industriais, sistemas de aquecimento e resfriamento, e é fundamental para entender fenômenos relacionados à temperatura e calor (HALLIDAY; COL., 2009).
- Energia nuclear: É uma forma de energia liberada durante reações nucleares, que ocorrem no núcleo dos átomos. Ela é gerada através de processos como a fissão nuclear, em que núcleos atômicos pesados são divididos em fragmentos menores, ou

a fusão nuclear, em que núcleos leves se combinam para formar um núcleo mais pesado (HALLIDAY; COL., 2009).

- Energia cinética: É uma forma de energia associada ao movimento de um objeto. É determinada pela massa do objeto e pela sua velocidade. Quanto maior a massa e maior a velocidade de um objeto, maior será a sua energia cinética. A energia cinética pode ser convertida em outras formas de energia, como calor ou trabalho, e é uma medida da quantidade de trabalho que um objeto pode realizar devido ao seu movimento (HALLIDAY; COL., 2009).
- Energia mecânica: É a soma das energias cinética e potencial de um sistema físico. A energia cinética está associada ao movimento de um objeto, enquanto a energia potencial está relacionada à posição ou configuração do objeto em um campo de força, como a gravidade. A energia mecânica total de um sistema é conservada quando não há dissipação de energia por forças externas não conservativas, como atrito ou resistência do ar. A energia mecânica pode ser convertida entre suas formas cinética e potencial dentro do sistema, mas sua quantidade total permanece constante (HALLIDAY; COL., 2009).
- Energia potencial gravitacional: É a forma de energia associada à posição de um objeto em relação a um campo gravitacional. É a energia que um objeto possui devido à sua altura em relação à superfície da Terra ou a outro referencial. Quanto maior a altura, maior a energia potencial gravitacional. Essa forma de energia pode ser convertida em outras formas de energia quando o objeto cai ou é liberado de sua posição elevada (HALLIDAY; COL., 2009).

Nem sempre a fonte de energia está pronta para o uso, na verdade na maioria das vezes ela precisa ser convertida e armazenada. Os sistemas de energia são construídos em uma sequência de processos, que progressivamente adquirem-se, converte e, eventualmente, se armazena e distribui para atender aos diversos usos na sociedade.

Após percorrer sobre as formas de energia, aquela que abordaremos no decorrer deste trabalho é a elétrica, pois é ela que é fornecida através de tensão alternada até chegar em nossas casas e serem convertida em contínua através de uma fonte.

2.4.2 Características de Energia Elétrica

A energia elétrica é gerada por meio de diversas fontes, como usina hidroelétrica, usina termoeletricas movidas a gás natural ou carvão mineral, biomassa, usina nuclear, usina eólica, usina fotovoltaica, entre outras. Após a geração, a energia é transportada por meio de linhas e torres de transmissão de alta tensão. Ao chegar nas cidades, é necessário que a eletricidade passe por transformadores de tensão localizados nas subestações, a fim de reduzir sua voltagem. Antes de chegar às residências, a energia ainda precisa passar pelos transformadores de distribuição, instalados em postes, para que sua voltagem seja adequada, geralmente para 110 ou 220 volts (HAWARY, 2008).

A corrente elétrica é o fluxo ou movimento ordenado de cargas elétricas em um condutor, como um fio metálico. É uma grandeza física que representa a quantidade de carga elétrica que atravessa uma seção transversal do condutor em um determinado intervalo de tempo. A corrente elétrica é medida em ampères (A) e é causada pela diferença de potencial elétrico aplicada aos extremos do condutor, que impulsiona as cargas a se deslocarem. É importante ressaltar que a corrente elétrica pode ser de natureza contínua (corrente contínua - *DC*) ou alternada (corrente alternada - *AC*), dependendo do tipo de fonte de energia ou sistema elétrico utilizado (TORRES, 2012).

A corrente contínua tem sua tensão e corrente unidirecionais, transportando cargas por meio do fluxo de elétrons, portanto, seu circuito possui polaridade positiva e negativa, resumindo, o tipo de onda e intensidade não interferem em sua polaridade (TORRES, 2012).

Para que uma corrente alternada vire contínua é necessário o uso de transistores, fontes e diodos.

- Transistores: Tem como função controlar o fluxo de corrente elétrica.
- Fontes: Convertem a energia alternada em contínua, alterando a sua voltagem.
- Diodos: Tem como função permitir a energia ir somente em um sentido, respeitando a sua polaridade.

A corrente alternada é um tipo de corrente elétrica cuja direção e intensidade variam periodicamente ao longo do tempo. Na corrente alternada, as cargas elétricas oscilam de forma contínua, alternando sua direção em intervalos regulares. Isso ocorre devido à variação do potencial elétrico aplicado, que ocorre em forma de onda senoidal. A corrente alternada é amplamente utilizada em sistemas de distribuição de energia elétrica, sendo gerada por meio de geradores e transformada em diferentes níveis de tensão para alimentar residências, indústrias e outros locais. A frequência da corrente alternada é medida em *hertz* (Hz), sendo comum a utilização de corrente alternada com frequência de 50 Hz ou 60 Hz, dependendo do país ou região (TORRES, 2012).

2.4.3 Tensão Elétrica (Voltagem)

Na eletricidade uma das grandezas elétricas conhecidas é a voltagem. Por mais que essa palavra não exista, ela se popularizou devido o termo *Volt* (V), que é a unidade que representa a tensão elétrica, que é definida pela diferença entre o potencial elétrico de dois pontos, onde um ponto possui excesso de elétrons e vai para o outro com falta de elétrons. Sendo a força necessária para movimentar os elétrons e criar uma corrente elétrica (TORRES, 2012).

O *Volt*(V) é a unidade de medida utilizada para representar a diferença de potencial elétrico, também conhecida como tensão elétrica. Ele é utilizado para quantificar a força ou pressão que impulsiona as cargas elétricas em um circuito elétrico. O *volt* é definido como a diferença de potencial elétrico entre dois pontos de um condutor quando uma corrente de

1 ampère passa através dele, dissipando uma potência de 1 *Watt* (W). Em outras palavras, o *volt* representa a quantidade de energia elétrica transferida por unidade de carga elétrica em um circuito. O *volt* é uma das unidades básicas do Sistema Internacional de Unidades (SI) e é amplamente utilizado em todo o mundo para medir e descrever a tensão elétrica em sistemas elétricos e eletrônicos (MATTEDE, 2016).

Potência pode ser definida como a transformação da energia elétrica em outro tipo de energia, sendo calor, movimento ou som. Sua forma de medida é a *watt*, tendo como a seguinte definição expressada na figura abaixo:

Figura 7 – Formúla de potência

$$P = \frac{E}{t}$$

Fonte: TORRES (2012)

Sendo "E" a energia transformada e "t" o tempo gasto nessa transformação. Com isso, 1 W é igual a 1 *Joules* por segundo. Para se medir a potência de um componente basta calcular através da seguinte fórmula: Potência (P) = Diferença de Potencial (V) . Corrente (I). Os resistores por exemplo, são classificados de acordo com sua potência máxima de dissipação. Ao tentar dissipar mais potência do que a suportada, ele queimará. Um detalhe que é bom ressaltar é a diferença entre potência nominal e real, pois quando um dispositivo eletrônico possui em sua descrição: 220V e 100W, não quer dizer que a potência dissipada por ele será sempre de 100W, mas que será de 100W quanto estiver ligado a uma diferença de potencial de 220V (TORRES, 2012).

2.4.4 Eficiência Energética

O termo eficiência tem como significado conseguir o melhor resultado com o mínimo de erros e custos. Ao economizar no consumo elétrico, evitamos a necessidade imediata de construção de novas usinas geradoras e sistemas elétricos associados, disponibilizando recursos para outras áreas e ajudando na preservação do meio ambiente (CARVALHO; COL., 2012).

Em 2001 devido a crise em que nosso país passou, foi sancionada a Lei n.10.295/2001, que tinha como objetivo a Política Nacional de Conservação e Uso Racional de Energia (HENRIQUE; COL., 2001). Vale ressaltar que essa Lei não foi a primeira medida adotada pelo governo, existem diversos Órgãos Públicos e Mistos que buscam legislar e conscientizar sobre o consumo de forma exagerada.

Por ser um tema que é relacionado a políticas públicas, tanto o governo como a iniciativa privada tem interesse em desenvolver projetos que incentivem a produção, geração

e consumo de forma eficiente. No começo de 2021, o Governo Federal, publicou o Plano Nacional de Eficiência Energética que propõe diversas formas de promover a eficiência energética, inclusive propostas de financiamento de iniciativas de eficiência energética (EDISON; COL., 2014).

2.5 Arduino

Arduino é uma plataforma de prototipagem eletrônica baseada em hardware e software de código aberto. Consiste em uma placa de circuito integrado com um microcontrolador programável e uma variedade de pinos de entrada/saída que permitem a conexão de sensores, atuadores e outros componentes eletrônicos (BANZI; COL., 2022a).

Existem diversas placas de Arduino, porém a mais utilizada é a Arduino Uno, que permite ler entradas de luz em um sensor, medir voltagem, acionamento através de um botão e transformá-la em uma saída como por exemplo ligando um *led*, ativando um motor, exibindo uma informação, porém para isso ser feito é necessário um conjunto de instruções previamente digitadas e enviadas para o ambiente de desenvolvimento integrado (IDE). Para isso ser feito, devemos utilizar a linguagem de programação Arduino que é baseada em *Wiring* (CURVELLO; COL., 2014). Essa linguagem trata-se de uma abstração de *hardware* desenvolvida em linguagem C/C++ para programação de microcontroladores. (MOTA, 2015)

2.5.1 Arduino UNO

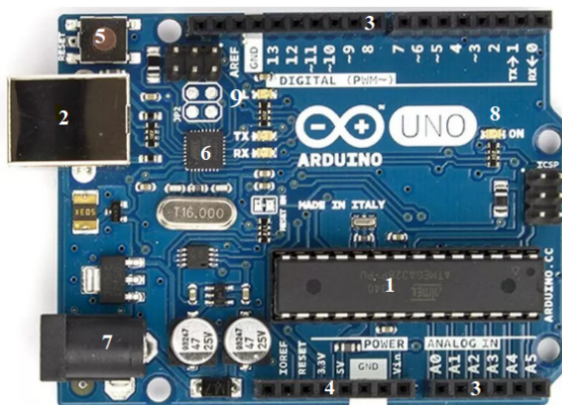
O Uno possui o *hardware* simples, porém eficiente, possuindo os seguintes blocos em sua composição:

1. Microcontrolador: É um dispositivo eletrônico integrado em um único *chip* que contém um processador, memória e periféricos de entrada/saída. Ele é projetado para executar tarefas específicas de controle em sistemas eletrônicos, oferecendo uma solução compacta e programável para funções de controle em sistemas embarcados.
2. Conector *USB*: Conecta o Uno a outro dispositivo. É por onde o computador e o Arduino se comunicam com o auxílio de um cabo *USB*, podendo ser uma opção de alimentação da placa.
3. Pinos de entrada e saída: O Arduino UNO possui 26 pinos, sendo 14 portas digitais de entrada e saída, 6 pinos de entrada analógica e 6 saídas analógicas, sendo responsáveis por fazer com que o Arduino interaja com o meio externo.
4. Pinos de Alimentação: Responsável pelo fornecimento de diferentes valores de tensão para os componentes do projeto.
5. Botão de Reiniciar: Responsável por reiniciar a placa Arduino.
6. Conversor Serial-*USB* e *LED's* TX/RX: Possibilita a comunicação entre o computador e o microcontrolador. Quando está tendo a comunicação os *LED's* TX e RX acendem.

7. Conector de Alimentação: Responsável por receber a energia de alimentação externa, que pode ter uma tensão de no mínimo 7 *Volt's* e no máximo 20 *Volt's* e uma corrente mínima de 300Ma.
8. *LED* de Alimentação: Possibilita saber se a placa está recebendo energia.
9. *LED* Interno: *LED* conectado ao pino digital 13.

Abaixo está o modelo da placa do Arduino UNO, seguindo a ordem dos itens elencados acima, conforme figura a seguir:

Figura 8 – Estrutura de um Arduino UNO



Fonte: BANZI e COL. (2022a)

2.5.2 Ambiente de desenvolvimento integrado (IDE)

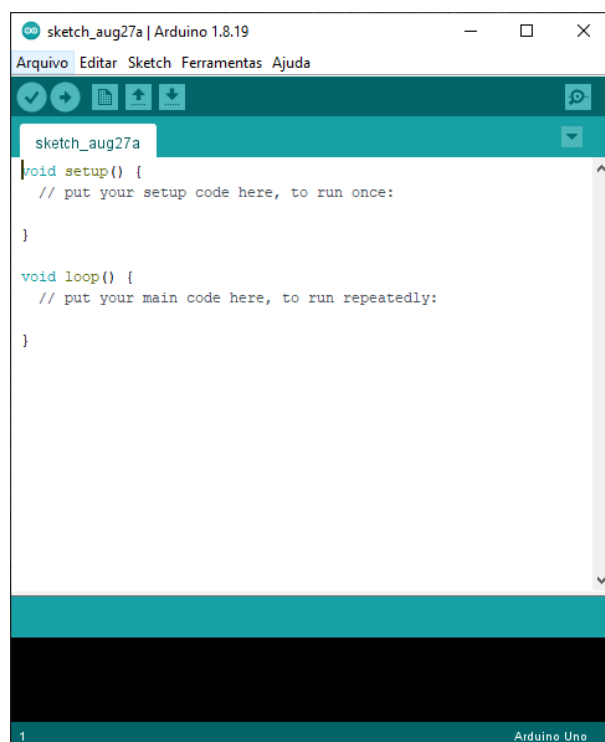
A IDE (*Integrated Development Environment* - Ambiente de desenvolvimento integrado) pode ser obtida pelo site oficial do Arduino. No próprio página *Web* existe uma opção chamada *LEARN*, que ao clicar te redireciona para outro endereço que entrega os guias explicativos de como funciona o hardware e o software. Na opção de *SOFTWARE*, pode ser feito o *download* da versão mais recente da Arduino *IDE*, ela é disponibilizada para o sistema Operacional *Windows*, *Linux* e *Mac OS X*. O software possui um ambiente de desenvolvimento simples, ele apresenta duas funções denominadas de *setup()* e *loop()*. A *IDE* do Arduino possui um editor de código que suporta a linguagem de programação baseada em C/C++, na qual os *sketches* do Arduino são escritos. O editor fornece recursos básicos de edição, como destaque de sintaxe, indentação automática e verificação de erros básicos (BANZI; COL., 2022b).

O Arduino *IDE* possui um sistema integrado para gerenciamento de bibliotecas, permitindo que você instale, atualize e remova bibliotecas facilmente. As bibliotecas contêm códigos pré-compilados que fornecem funcionalidades adicionais, como controle de dispositivos específicos, protocolos de comunicação, manipulação de sensores, entre outros. Também é incluso na *IDE* um compilador que transforma o código fonte escrito em C/C++ em código de máquina compatível com a plataforma Arduino. Ele também

possui um carregador de *bootloader*, que permite fazer o *upload* do código compilado para a placa Arduino, geralmente usando uma conexão *usb* (BANZI; COL., 2022b).

Existe também, o recurso chamado *Monitor Serial*, que permite a comunicação bidirecional entre a placa Arduino e o computador. Ele é útil para depurar o código e exibir informações em tempo real, como mensagens de depuração, leituras de sensores ou respostas da placa. A *IDE* suporta uma ampla variedade de placas Arduino, podendo selecionar a placa específica que se está usando e a IDE configura automaticamente as opções de compilação e carregamento adequadas (BANZI; COL., 2022b). Na figura abaixo, é demonstrado um exemplo de tela da *sketch* do Arduino UNO.

Figura 9 – Tela inicial da *IDE*



Fonte: BANZI e COL. (2022c)

2.5.3 Sensores

Os sensores vieram para revolucionar o mercado, pois são dispositivos de entrada de valores que compõem outros eletrônicos, por exemplo os teclados e mouses. Graças aos sensores foi possível automatizar muitos sistemas e assim, evitar desperdícios nas empresas. Existem diversos modelos de sensores, alguns ajudam a medir a temperatura, luminosidade, humidade, força, distância e corrente elétrica. Por conta dessa diversidade, os sensores são constantemente usados na área da robótica, pois viabilizam os usuários a terem várias funcionalidades em seus projetos, podendo inclusive a otimizar os processos e consequentemente aumentar a eficiência energética (KIMMO; COL., 2014).

A acoplação de sensores com o Arduino UNO é fundamental para que o projeto se expanda e com a ajuda de alguns sensores em específicos, é possível fazer a comunicação com *softwares*. Um bom exemplo é o Sensor de Corrente Não Invasivo 100A SCT-013, que permite o usuário medir a corrente elétrica de um sistema e através de um *Ethernet Shield*, comunicar com algum *software* de monitoramento, como o ZABBIX.

2.5.3.1 Sensor de Corrente Não Invasivo 100A SCT-013

Existem diversos modelos de sensores que realizam a medição da corrente elétrica, ambos possuem transformador de corrente (TC), o que os diferem é se o seu núcleo é sólido ou dividido. O sensor de corrente não invasivo 100A SCT-013 é um modelo de sensor transformador de corrente, que possui como significado da sigla SCT (*Split-core Current Transformer* - Transformador de Corrente de Núcleo Dividido), e ele também possui característica de ser não invasivo por não precisar estar em contato direto com a parte elétrica do circuito, e para que a medição ocorra ele utiliza bobinas (SINEM, 2017).

A figura 10 mostra um Sensor de Corrente Não Invasivo 100A SCT-013:

Figura 10 – Sensor de Corrente Não Invasivo 100A SCT-013



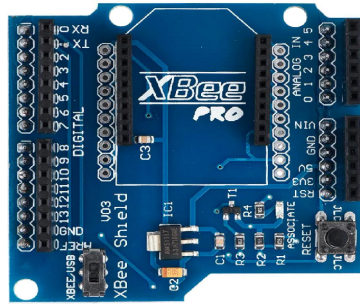
Fonte: Sinem (2017)

2.5.4 Xbee Shield

O *Xbee Shield* é o *hardware* que pode ser acoplado ao Arduino UNO e que da mesma forma, permite que o módulo *XBee S2C* seja interligado nela. Com essas conexões a placa faz que os dados que estão sendo coletados pelo Arduino UNO se comuniquem com outros dispositivos via *Wireless* através do protocolo *Zigbee*. Vale ressaltar que para que haja essa comunicação é necessário ter dois módulos *XBee S2C*, um conectado no projeto e outro no computador através do adaptador chamado *XBee Explorer USB*, através desse adaptador é possível configurar os módulos no *software* gratuito chamado *XCTU*.

Exemplo de placa *Xbee Shield*:

Figura 11 – Placa Xbee Shield



Fonte: Autor (2023)

2.5.4.1 Protocolo Zigbee

Devido a alta popularização da IOT (*Internet of Things* - Internet das Coisas) que chegou junto com a proposta de automatizar os aparelhos eletrônicos, o *Zigbee* se tornou um protocolo de comunicação e transferência *Wireless* bastante utilizado em sensores devido o seu baixo consumo energético, alcance de transmissão e o custo de aquisição (EZEQUIEL, 2004). É um protocolo que segue o padrão estabelecido pelo Comitê IEEE 802.15.4 (*Institute of Electrical and Electronics Engineers* - Instituto de Engenheiros Eletricistas e Eletrônicos), oferecendo fundamentos para o padrão WPAN (*Wireless Personal Area Networks* - Rede de área pessoal).

2.5.4.1.1 Módulo Xbee S2C

O *XBee S2C* é um módulo que através da rádio frequência realiza a comunicação serial para estabelecer conexão sem fio entre dois dispositivos, em área aberta o seu sinal percorre até 1200m² e em áreas fechadas até 60m², sendo um dispositivo que trabalha com diferentes topologia de redes (CHAGLIA, 2019).

Figura 12 – Módulo Xbee S2C



Fonte: Autor (2023)

2.5.4.1.2 Xbee Explorer USB

É um adaptador que possui suporte ao *software XCTU* e é responsável por conectar um dos módulos *XBee S2C* ao computador, permitindo ao administrador configurar os parâmetros dos módulos de forma prática para facilitar a transmissão via *wireless* dos dados (CHAGLIA, 2019). A figura 13, demonstra o *Xbee Explorer USB*.

Figura 13 – *Xbee Explorer USB*

Fonte: Autor (2023)

3 METODOLOGIA

Conforme preconiza Thomas Samuel Kuhn:

A metodologia científica é um conjunto de práticas e normas que orientam a atividade científica. Ela envolve a formulação de hipóteses, a coleta e análise de dados, a interpretação dos resultados e a comunicação dos achados científicos. No entanto, a metodologia científica não é um conjunto fixo e universal de regras, mas sim um conjunto de convenções compartilhadas dentro de uma comunidade científica particular. Essas convenções podem variar ao longo do tempo e entre diferentes campos científicos. Além disso, a metodologia científica também é influenciada por fatores sociais, políticos e históricos, que moldam as prioridades e os paradigmas dominantes em uma determinada época (KUHN, 1970).

Diante disto, nesta seção, serão descritos os procedimentos adotados para realizar a pesquisa, incluindo a abordagem metodológica, o delineamento do estudo, a coleta e análise dos dados. Essa metodologia foi escolhida com base nos objetivos da pesquisa e nas questões de pesquisa estabelecidas anteriormente.

3.1 Abordagem Metodológica

Para atingir os objetivos deste trabalho, foi escolhido uma abordagem quantitativa. Essa escolha foi feita considerando a natureza do problema de pesquisa e a disponibilidade de recursos. A abordagem quantitativa permitirá medir os fenômenos de interesse de forma adequada.

3.2 Delineamento do Estudo

O estudo adotou um delineamento experimental, de acordo com as necessidades da pesquisa. Esse delineamento proporcionou uma estrutura adequada para coletar os dados necessários e responder às questões de pesquisa de maneira eficiente.

3.3 Coleta de Dados

A coleta de dados foi realizada por meio de observações, conforme apropriado para a abordagem metodológica escolhida. Para a coleta de dados foi utilizado o Arduino UNO, sensor SCT013 e o módulo *XBee S2C*, que foram adaptados através de códigos de programação para transmitirem os dados coletados para a plataforma Zabbix, afim de possibilitar monitoração do consumo energético do sistema. O procedimento de coleta de dados foi baseado em instruções contidas no livro (REIS, 2020), tendo como local de coleta os computadores da sala 106 do Instituto Federal de Goiás - Câmpus Uruaçu.

3.4 Análise dos Dados

A análise dos dados foi realizada utilizando análise de conteúdo e estatística descritiva, conforme apropriado para a abordagem metodológica adotada. Os dados foram organizados e categorizados. Como forma de análise de dados, eles são expostos no site do Zabbix, em formato de gráfico, podendo ser selecionado a forma de exibição, sendo em colunas, barras, setores e linhas.

4 IMPLEMENTAÇÃO

Para começar a implementação do projeto, foram necessários os seguintes itens:

- 01 – Placa Arduino UNO R3;
- 01 – Sensor de corrente não invasivo 100A SCT-013;
- 01 – *Protoboard*;
- 02 - Resistores de 10 K Ω ;
- 01 – Resistor de 300 Ω ;;
- 01 – Capacitor eletrolítico de 100uF;
- 05 - *Jumpers*;
- 02 - Módulos *XBee S2C*;
- 01 - Placa *XBee Shield Pro*;
- 01 - *XBee Explorer USB*;

Também foi necessário um computador com acesso a *internet*, o computador utilizado foi fornecido pelo Instituto Federal de Goiás - Câmpus Uruaçu, que nele continha as seguintes especificações:

- Processador: *Quad Core (AMD A10 PRO- 7800B R7)*
- Memória: *DDR4 8GB*;
- Placa de vídeo: Integrada
- Memória de Armazenamento: *HD* de 500GB;
- Sistema Operacional: *Ubuntu 22.04.2 LTS(Jammy Jellyfish)*;

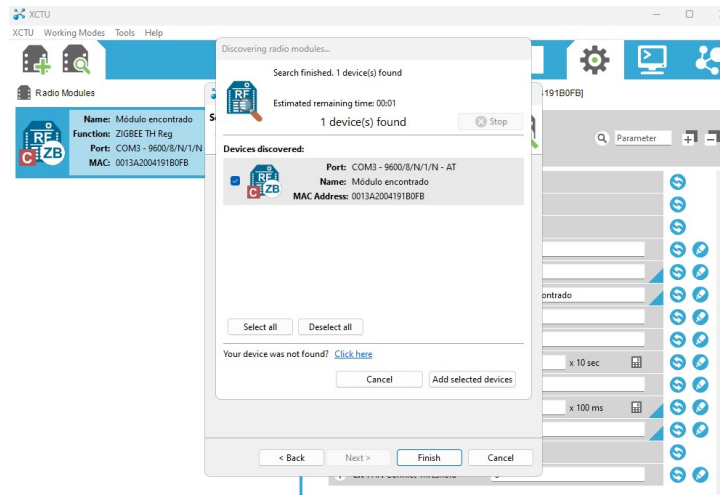
Foram utilizados alguns *softwares* para possibilitar a implementação, sendo:

- *XCTU*: Responsável pela configuração dos módulos *XBee S2C*, versão 6.5.12 Linux x64;
- Arduino IDE 2.0.0: Para codificar e configurar o Arduino UNO;
- Zabbix: Para que haja a comunicação do Arduino UNO com o servidor Zabbix.

Com o itens necessários, foi dado início pela configuração dos módulos *XBee S2C*.

4.1 Configurando os Módulos XBee S2C

Conexão física do *XBee Explorer USB* ao computador e inicialização do *XCTU*. Este *software* por padrão vem na língua inglesa, na opção *Discover radio Devices* (Descobrir dispositivos de rádio) é selecionada a porta em que o *XBee Explorer USB* foi conectado e em *Next*(Próximo), vai aparecer uma outra tela pedindo para configurar os parâmetros da porta, nessa tela não precisa alterar nenhuma informação. Na próxima tela serão mostrados os dispositivos descobertos, o módulo é selecionado e adicionado na opção *Add selected devices*. Feito isso aparecerá o módulo no canto esquerdo, conforme figura a seguir:

Figura 14 – Módulo *XBee* identificado pelo *XCTU*

Fonte: Autor (2023)

Nessa parte é necessário configurar um *XBee* S2C para ser o coordenador e outro para se o roteador, é necessário que ambos estejam na mesma rede com identificadores de PAN iguais, a diferença é que o módulo roteador usará um *driver* diferente que será explicado abaixo.

4.1.1 Módulo *XBee* S2C Coordenador

Clicando no módulo será dado início à leitura de configuração, que será o coordenador. Na nova aba que aparece à direita a opção *Update Firmware* configura a função desse módulo. Somente a versão do *firmware* deve ser trocada para 405F *Newest* e clique em *finish* (Finalizar).

Na parte escrita *Networking*-(Rede) são alterados os seguintes campos:

- *ID PAN ID*: 1234
- *CE Coordinator Enable*-(Ativar Coordenador): *Enable*[1];

Em *Addressing*-, foi alterado os seguintes campos:

- *DL Destination Address Low*-(Endereço de destino de baixa frequência): FFFF;
- *NI Node Identifier*-(Identificador de nó): Coordenador;

Depois de realizar essas configurações clique em *Write Radio Settings* para salvar as alterações desse módulo.

4.1.2 Módulo *XBee* S2C Roteador

A leitura de configuração do módulo de roteamento será iniciada com apenas um clique e será configurado na opção *Networking* (Rede), com alterações dos seguintes campos:

- *ID PAN ID*: 1234

- *JV Channel Verification: Enable*[1];
- *CE Coordinator Enable*-(Ativar Coordenador): *Disable*;

Em *Addressing*-, foi alterado os seguintes campos:

- *DL Destination Address Low*-(Enderesso de destino de baixa frequência): FFFF;
- *NI Node Identifier*-(Identificador de nó): Roteador;

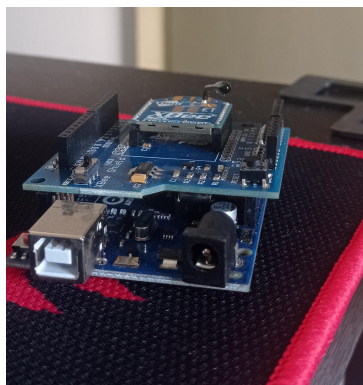
Depois de realizar essas configurações as alterações são salvas em *Write Radio Settings*.

Agora ambos os módulos estão devidamente configurados. Para testar se está havendo a comunicação, abra um novo *XCTU*, resumindo, em cada um dos *XCTU* terá que ter um módulo conectado, após isso entre no terminal do *XCTU* e clique no ícone do *SERIAL* e digite no *console*, se as letras saírem no terminal do outro é sinal que foi configurado corretamente.

4.2 Configurando o Arduino UNO

Primeiramente, devemos acoplar os seguintes itens para facilitar na hora de ligar os *jumpers* aos componentes. Comecei por acoplar o módulo *XBee S2C* à *XBee Shield*, posteriormente conectei ao Arduino UNO, conforme a figura 15.

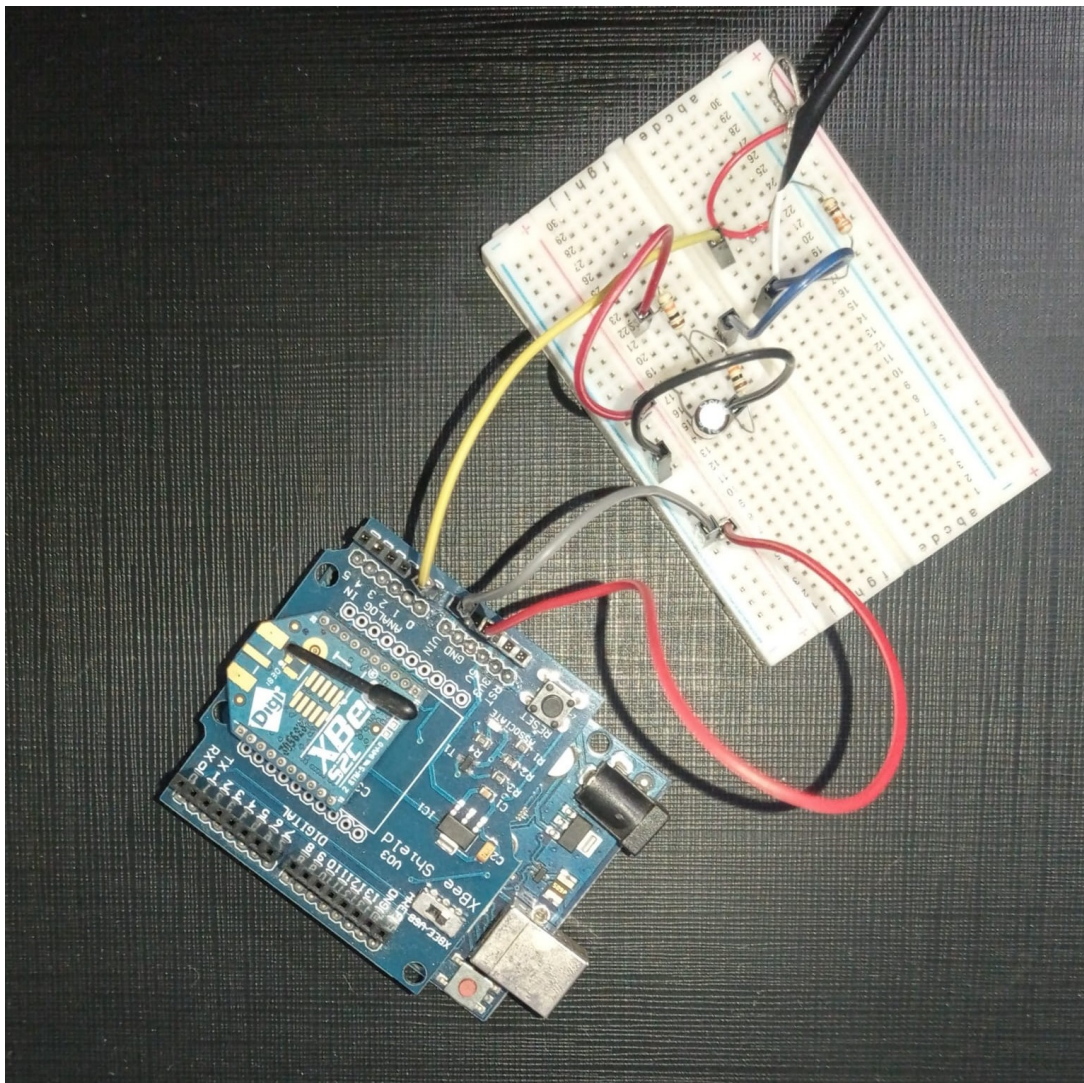
Figura 15 – Acoplando a *Xbee Shield* ao Arduino UNO



Fonte: Autor (2023)

Vale ressaltar, que não foi utilizada uma entrada P2 para conectar o sensor de corrente não invasivo 100A SCT-013 ao Arduino UNO, portanto, foi necessário adaptar o cabo e separar os fios para conectar na *protoboard*, após essa observação, os *jumpers* e os demais componentes ficaram conectados da seguinte forma:

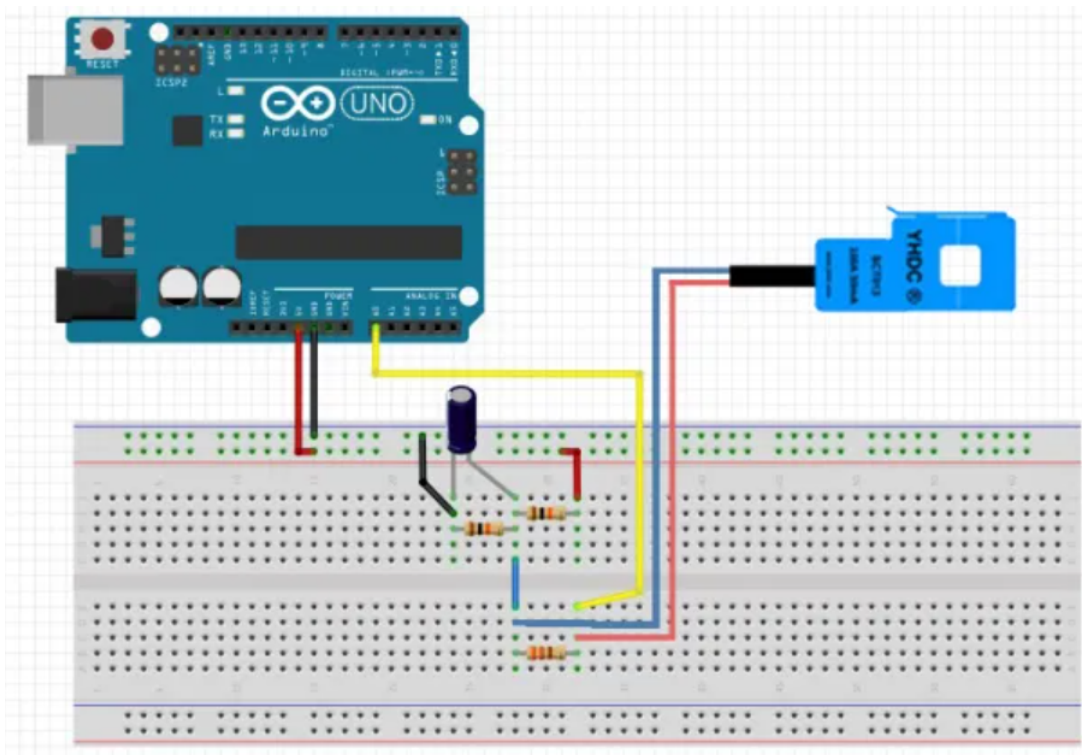
Figura 16 – Conectado os sensores ao Arduino UNO físico



Fonte: Autor (2023)

Para demonstrar as conexões de forma mais organizada, a próxima figura foi criada através da ferramenta *Tinkercad*. Ressalto que na ferramenta não havia o sensor SCT-013, *XBee Shield* e os módulos *XBeeS2C* para uma melhor exemplificação.

Figura 17 – Conectado os sensores ao Arduino



Fonte: Autor (2023)

Lembrando que o sensor de corrente foi conectado ao aparelho que teve sua corrente elétrica medida, passando por somente um dos seus fios, pois, conforme a corrente produz um campo elétrico no sentido oposto ao do outro fio os campos magnéticos se anulam, por esse fator é necessário que o sensor esteja ligado a um deles separadamente para não haver anulação de campo magnético.

Para configurar o código da *IDE* do Arduino UNO é necessário baixar sua *IDE* fornecida gratuitamente no site do Arduino e para que o código funcionasse tive que utilizar a biblioteca *Emonlib* que foi desenvolvida pela *OpenEnergyMonitor* com a intenção de facilitar a integração de cálculos mais precisos em relação a corrente a ser medida ao Arduino. Após ter feito o download, é necessário descompactar e transferir a biblioteca para a pasta *Libraries* da *IDE* do Arduino UNO e ativá-la.

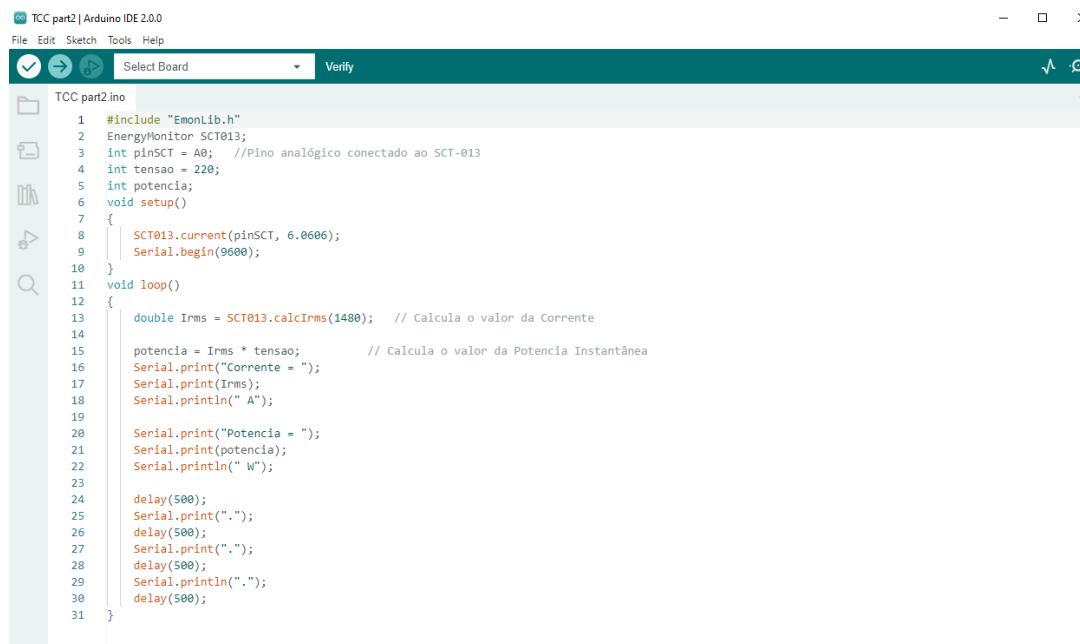
O código que foi utilizado no presente trabalho foi baseado em (SINEM, 2017). Inicialmente foi incluída a biblioteca no código. Depois, foi declarado o objeto com o nome *SCT013*, com uma variável do tipo *Int* chamada *pinSCT*, com valor igual A0, isso está dizendo em qual porta analógica o *SCT* foi conectado. Posteriormente, são necessárias mais duas variáveis do tipo *Int* chamadas tensão e potência, a tensão teve seu valor igual á 220V, pois esse valor é referente a tensão da rede do estado de Goiás. A potência será utilizada para receber o valor da multiplicação provinda da tensão e da corrente.

No *void setup()* é necessário chamar a função *SCT013.current(pinSCT,6.0606)*, para realização do cálculo da corrente de forma eficaz. Também será chamado o *Se-*

`rial.begin(9600)` para habilitar a comunicação serial entre o Arduino e o computador.

Para que o valor da corrente seja obtido foi utilizado a função `void loop()` seguida da função `SCT013.calcIrms()` para direcionar o valor da corrente a variável `Irms`. O valor utilizado no código entre parênteses foi de 1480, esse valor se ferente a quantidade de amostras que a função irá ler. E com isso, através dos valores informados, a multiplicação da corrente *rms* pela tensão informada nos gera um valor de potência gerada pelo dispositivo.

Figura 18 – Código para leitura da corrente



Fonte: Autor (2023)

Quando a codificação estiver feita, o *SKETCH* deve ser salvo. Agora o Arduino UNO é conectado sem estar com a *XBee Shield* acoplada ao computador, e com a IDE do Arduino UNO aberta, é necessário o upload do código para o Arduino. Posteriormente, é feita a conexão do módulo *XBee S2C* denominado roteador no *XBee Shield* e acoplado ao Arduino UNO, após fazer isso o equipamento já estará apto para transmitir os dados coletados pelo Arduino UNO.

Para verificar o funcionamento dos passos elencados até aqui, foi conectado o Arduino UNO no computador, passado o sensor *SCT-013* sobre um dos fios de uma extensão que estava conectada na tomada, conforme figura abaixo

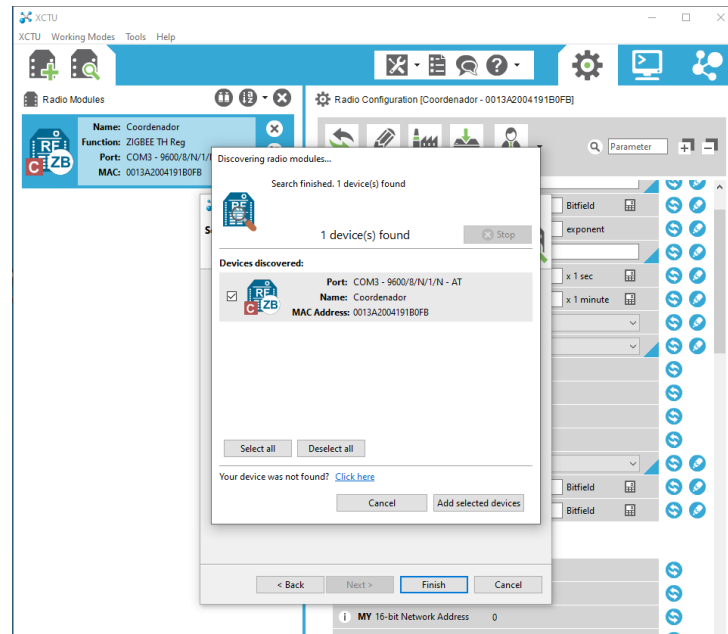
Figura 19 – SCT-013 Conectado a uma extensão



Fonte: Autor (2023)

Após, foi conectado um carregador de celular, depois foi inserido o módulo *XBee* S2C denominado coordenador no computador através do *XBee Explorer* e aberto o *software XCTU*. No *software*, cliquei em *Discover radio devices* e selecionei a porta *USB Serial Port* (Porta Serial *USB*), foi reconhecido nesta porta o Módulo *XBee* S2C denominado coordenador:

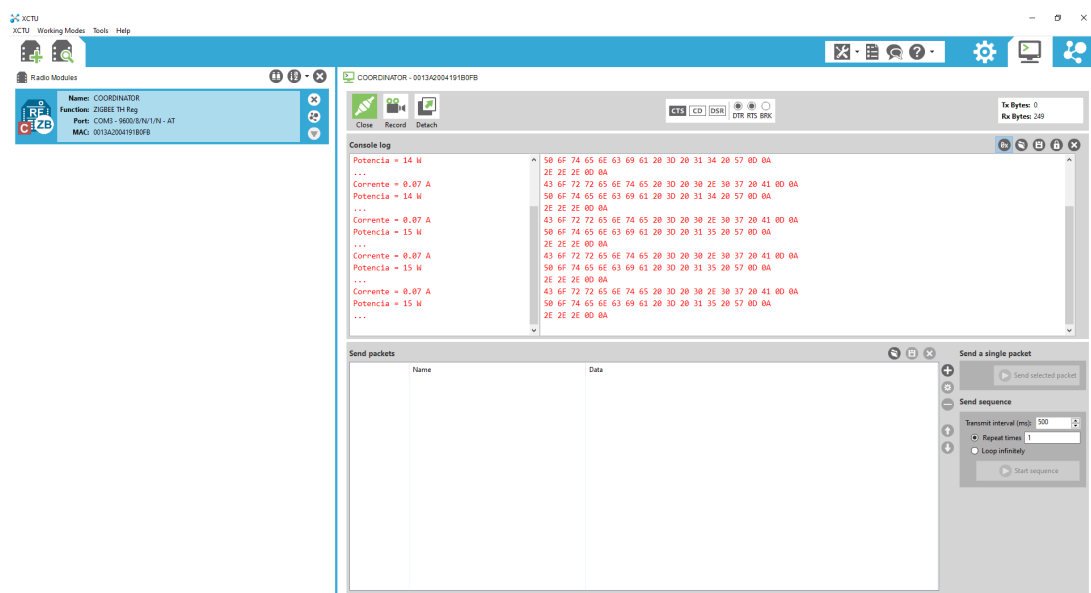
Figura 20 – Módulo XBee S2C Coordenador sendo reconhecido



Fonte: Autor (2023)

Depois que o módulo foi adicionado, cliquei na opção *Switch to consoles working mode* (Alternar para o modo de trabalho dos módulos), nesta nova aba pode ser vista a opção *Open the serial connection with the radio module* (Abrir conexão serial via rádio com o módulo serial), ao habilitar essa opção deverá aparecer informações no *Console log* (Registro do módulo), a respeito do consumo energético carregador, conforme a figura abaixo:

Figura 21 – Mensuração do consumo energético



Fonte: Autor

Como forma de mensurar o consumo energético, foi realizado testes no computador dessa Instituição Federal de Goiás - Câmpus Uruaçu, colocando a máquina em estresse com vídeos em diferentes resoluções. Vale ressaltar que a resolução de uma imagem se refere à quantidade de detalhes visuais que podem ser discernidos na imagem, que é determinada pelo número de *Pixels*(quadros) em uma determinada área da imagem. Quanto mais *pixels*, maior será a resolução e mais detalhes serão visíveis. O *hardware* responsável por esse processamento dos *pixels* é a *CPU* ou a *GPU*(*Graphic Processing Unit*).

4.3 Coleta dos Resultados

Após essa breve explicação, segue os resultados obtidos após utilizar a máquina em diferentes situações:

Na figura 21 a resolução de vídeo escolhida foi 1080p, nesta resolução é possível perceber que não é exigido tanta perfomace da máquina, portanto o consumo é baixo.

Figura 22 – Vídeo em 1080p



Fonte: Autor (2023)

Partindo para a figura 22, o aumento de resolução para 1440p, também não é suficiente para gerar uma alteração na corrente e potência, e a diferença entre a figura anterior não é perceptível energeticamente.

Figura 23 – Vídeo em 1440p



Fonte: Autor (2023)

No entanto, na figura 23, por estar em uma resolução de 2160p, já demonstra alteração na corrente e na potência, exigindo mais recursos da máquina e gerando maior consumo energético.

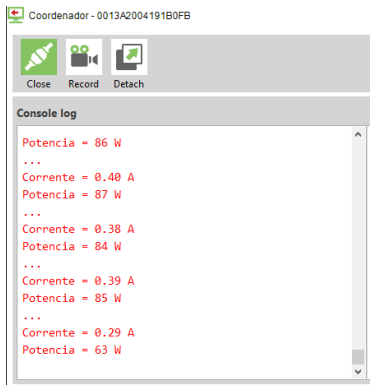
Figura 24 – Vídeo em 2160p



Fonte: Autor (2023)

Realizando mensuração em um segundo computador, utilizando jogos para provocar stress, foi obtido a seguinte informação:

Figura 25 – Teste em jogos



Fonte: Autor (2023)

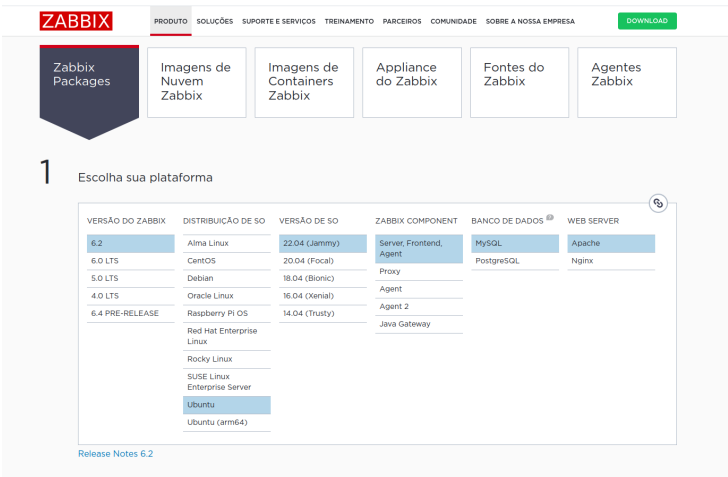
4.4 Configurando o Zabbix

O passo a passo para a instalação do Zabbix é disponibilizada pelo próprio site (ZABBIX, 2022a). Para que a instalação ocorra normalmente, confira se o seu sistema atende aos requisitos necessários solicitados pelo fabricante em (ZABBIX, 2022g), vale ressaltar que para evitar qualquer erro durante a instalação dos pacotes, abra o terminal e instale o repositório *UNIVERSE*, conforme o comando:

- *add-apt-repository universe*
Posteriormente utilize o comando para atualizar o sistema da máquina
- *sudo apt update*

O site fornece os códigos para serem executados no terminal do sistema operacional conforme as informações preenchidas previamente pelo administrador. A versão do Zabbix, distribuição do sistema operacional, versão do sistema operacional, componente Zabbix, banco de dados e servidor *web* utilizados, foram escolhidos conforme a figura 25.

Figura 26 – Instalação do Zabbix



Fonte: Autor (2023)

Em seguida, é necessário a criação de um banco de dados, nessa parte o administrador deverá atentar-se para alterar a senha solicitada e não esquecê-la. A parte em específico pode ser vista conforme a figura 26.

Figura 27 – Alterando a senha do servidor Zabbix

```

c. Criar banco de dados inicial
Make sure you have database server up and running.

Execute os seguintes passos em seu host de banco de dados.

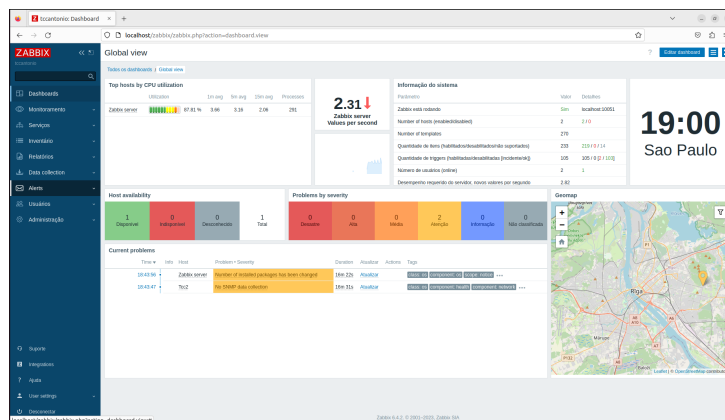
# mysql -uroot -p
password
mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user zabbix@localhost identified by 'password';
mysql> grant all privileges on zabbix.* to zabbix@localhost;
mysql> set global log_bin_trust_function_creators = 1;
mysql> quit;

```

Fonte: Autor (2023)

Assim que todos os passos para instalação fornecidos pelo site do Zabbix tiver sido executado, no navegador deve ser digitado *localhost/zabbix* com inserção de usuário e senha fornecido na documentação denominada guia rápido que está disponível no próprio site do Zabbix, e a seguinte tela deverá ser exibida:

Figura 28 – Servidor Zabbix



Fonte: Autor (2023)

Para continuar com a execução da implementação é necessário criar um *Host*, clique em configuração, posteriormente em *Host*, criar um novo *Host* que se encontra na parte superior direita, e preencher as informações solicitadas, fica ressaltado que nesta parte é necessário colocar o *IP* do *Host*, que no caso em questão deverá ser um *IP*, de preferência fixo, que deve ser atribuído ao XBee S2C coordenador na hora de configurá-lo, mas como não foi possível atribuir um *IP* ao módulo XBee S2C nesta implementação, essa parte não foi concluída da forma adequada. No entanto, foi dada continuidade para detalhar os demais passos. Sendo assim, a configuração do *Host* deverá ficar da seguinte forma:

Figura 29 – Configuração do host

Fonte: Autor (2023)

O próximo passo é abrir o terminal e fazer algumas verificações e alterações. Para começar certifique-se que o *SNMPD* e *SNMP* estão instalado na máquina, caso não esteja é necessário utilizar os seguintes comandos:

- `sudo apt-get install snmpd`

e

- `sudo apt-get install snmp`

Depois execute o seguinte comando:

- `service snmpd status`

A seguinte tela deverá ser exposta:

Figura 30 – *SNMPD* ativo

```
root@uru88806lab1061:/home/stt# service snmpd status
● snmpd.service - Simple Network Management Protocol (SNMP) Daemon.
   Loaded: loaded (/lib/systemd/system/snmpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-15 19:44:11 -03; 58s ago
     Main PID: 11266 (snmpd)
       Tasks: 1 (limit: 13036)
        Memory: 3.7M
           CPU: 58ms
      CGroup: /system.slice/snmpd.service
              └─11266 /usr/sbin/snmpd -lOw -u Debian-snmp -g Debian-snmp -i -snux mteTrigger mteTriggerConf -f

mai 15 19:44:11 uru88806lab1061 systemd[1]: Starting Simple Network Management Protocol (SNMP) Daemon....
mai 15 19:44:11 uru88806lab1061 snmpd[11266]: /etc/snmp/snmpd.conf: line 1: Error: Blank line following ]##### token.
mai 15 19:44:11 uru88806lab1061 snmpd[11266]: net-snmp: 1 error(s) in config file(s)
mai 15 19:44:11 uru88806lab1061 snmpd[11266]: /etc/snmp/snmpd.conf: line 1: Error: Blank line following ]##### token.
mai 15 19:44:11 uru88806lab1061 snmpd[11266]: net-snmp: 1 error(s) in config file(s)
mai 15 19:44:11 uru88806lab1061 systemd[1]: Started Simple Network Management Protocol (SNMP) Daemon..
root@uru88806lab1061:/home/stt#
```

Fonte: Autor (2023)

Com esse comando é possível ver o estado em que se encontra o agente *SNMP* na máquina. Após isso utilize o comando:

- `netstat -an | grep :161`

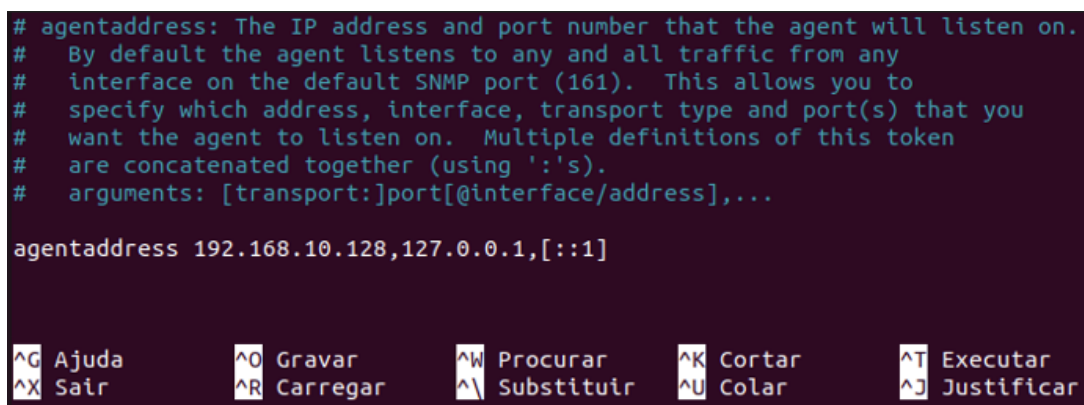
O comando acima, retornará o *IP* que foi atribuído a porta definida no *Host* criado no Zabbix. Para que o *IP* do módulo XBee S2C *IP* seja reconhecido pelo agente *SNMP* e visto na rede, é necessário utilizar outro comando. O próximo comando é para

abrir o arquivo de configuração do *SNMPD* através de um editor de texto, no código abaixo foi utilizado o nano:

- `sudo nano /etc/snmp/snmpd.conf`

Uma nova interface será aberta, que deverá ter o campo *agentaddress* alterado. Nesta linha o *IP* que o agente *SNMP* vai buscar as informações deverá ser adicionado, no caso o *IP* que deve ser adicionado é o que o administrador atribuiu ao módulo XBee S2C coordenador. No exemplo a seguir, foi atribuído o próprio *IP* da máquina utilizada, seguido por vírgula do *IP* do servidor Zabbix, ficando conforme a figura a seguir:

Figura 31 – Atribuindo *IP* a porta 161



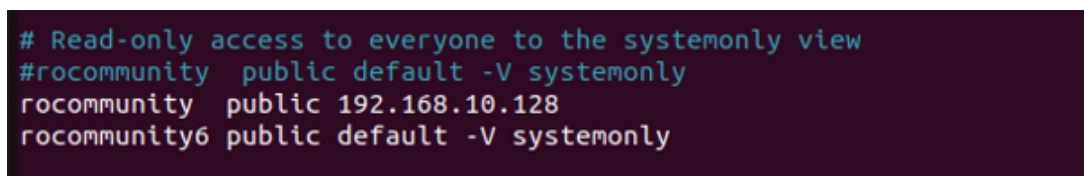
```
# agentaddress: The IP address and port number that the agent will listen on.
# By default the agent listens to any and all traffic from any
# interface on the default SNMP port (161). This allows you to
# specify which address, interface, transport type and port(s) that you
# want the agent to listen on. Multiple definitions of this token
# are concatenated together (using ':'s).
# arguments: [transport:]port[@interface/address],...

agentaddress 192.168.10.128,127.0.0.1,[:1]
```

Fonte: Autor (2023)

Ainda no editor de texto, deve-se alterar o campo denominado *rocommunity public default -V systemonly*, pois essa parte é responsável pela comunidade que disponibilizará os dados lidos pelo *SNMP* no servidor Zabbix, devendo ser colocado *rocommunity public 'IP do módulo XBee S2C'*, e salvar, ficando conforme modelo abaixo:

Figura 32 – Configurando o *SNMPD*



```
# Read-only access to everyone to the systemonly view
#rocommunity public default -V systemonly
rocommunity public 192.168.10.128
rocommunity6 public default -V systemonly
```

Fonte: Autor (2023)

Para certificar que tudo está ocorrendo normalmente, saia do editor de texto, e utilize o comando:

- `sudo service snmpd restart`

Com isso os serviços referentes ao *SNMPD* serão reiniciados, atualizando as configurações impostas. Ao utilizar novamente o comando:

- `netstat -an | grep :161`

O resultado esperado deverá ser conforme a figura abaixo, alterando somente o *IP* que foi atribuído a máquina:

Figura 33 – *IP's* com acesso pelo agente *SNMP*

```
sti@uru88806lab1061:~$ sudo netstat -an | grep :161
udp        0      0 127.0.0.1:161      0.0.0.0:*
udp        0      0 192.168.10.128:161 0.0.0.0:*
udp6       0      0 :::1:161           :::*
```

Fonte: Autor (2023)

Após a realização de todos esses passos, utilize o seguinte comando:

- *snmpwalk -v 2c -c public 'ip do coordenador'*

Esse comando é responsável por listar todas as *MIB's* que estão sendo lidas pelo agente *SNMP* que está associado ao *IP* informado. O (-v) é a versão do protocolo *SNMP* que está sendo utilizado, o -c é referente a comunidade que foi definida como pública no editor de texto. A figura abaixo, mostra o retorno após inserir o comando acima:

Figura 34 – Resultado da coleta do agente *SNMP* nas *MIB's*

```
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.97.105.108.3 = INTEGER: 1
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.97.105.108.4 = INTEGER: 1
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.97.105.108.5 = INTEGER: 1
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.105.114.101.1 = INTEGER: 1
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.105.114.101.2 = INTEGER: 1
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.105.114.101.3 = INTEGER: 1
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.105.114.101.4 = INTEGER: 1
iso.3.6.1.2.1.88.1.3.1.1.5.6.95.115.110.109.112.100.12.95.116.114.105.103.103.101.114.70.105.114.101.5 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.2.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = ""
iso.3.6.1.2.1.88.1.4.2.1.2.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = ""
iso.3.6.1.2.1.88.1.4.2.1.2.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.105.108.117.114.101 = ""
iso.3.6.1.2.1.88.1.4.2.1.2.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.105.114.101.100 = ""
iso.3.6.1.2.1.88.1.4.2.1.2.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.82.105.115.105.110.103 = ""
iso.3.6.1.2.1.88.1.4.2.1.3.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = Hex-STRING: 80
iso.3.6.1.2.1.88.1.4.2.1.3.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.105.108.117.114.101 = Hex-STRING: 80
iso.3.6.1.2.1.88.1.4.2.1.3.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.108.108.105.110.103 = Hex-STRING: 80
iso.3.6.1.2.1.88.1.4.2.1.3.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.82.105.115.105.110.103 = Hex-STRING: 80
iso.3.6.1.2.1.88.1.4.2.1.4.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.4.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.105.108.117.114.101 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.4.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.105.114.101.100 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.4.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.82.105.115.105.110.103 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.5.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.5.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.105.108.117.114.101 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.5.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.105.114.101.100 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.2.1.5.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.82.105.115.105.110.103 = INTEGER: 1
iso.3.6.1.2.1.88.1.4.3.1.1.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = OID: iso.3.6.1.2.1.88.2.0.3
iso.3.6.1.2.1.88.1.4.3.1.1.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.105.108.117.114.101 = OID: iso.3.6.1.2.1.88.2.0.4
iso.3.6.1.2.1.88.1.4.3.1.1.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.105.114.101.100 = OID: iso.3.6.1.2.1.88.2.0.1
iso.3.6.1.2.1.88.1.4.3.1.1.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.82.105.115.105.110.103 = OID: iso.3.6.1.2.1.88.2.0.2
iso.3.6.1.2.1.88.1.4.3.1.2.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = STRING: "snmpd"
iso.3.6.1.2.1.88.1.4.3.1.2.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.105.108.117.114.101 = STRING: "snmpd"
iso.3.6.1.2.1.88.1.4.3.1.2.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.105.114.101.100 = STRING: "snmpd"
iso.3.6.1.2.1.88.1.4.3.1.2.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.82.105.115.105.110.103 = STRING: "snmpd"
iso.3.6.1.2.1.88.1.4.3.1.3.6.95.115.110.109.112.100.95.108.105.110.107.85.112 = STRING: "linkUpDown"
iso.3.6.1.2.1.88.1.4.3.1.3.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.105.108.117.114.101 = STRING: "triggerFall"
iso.3.6.1.2.1.88.1.4.3.1.3.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.70.97.108.108.105.110.103 = STRING: "triggerFire"
iso.3.6.1.2.1.88.1.4.3.1.3.6.95.115.110.109.112.100.95.109.116.101.84.114.105.103.103.101.114.82.105.115.105.110.103 = STRING: "triggerFire"
iso.3.6.1.2.1.92.1.1.1.0 = Gauge32: 1440
iso.3.6.1.2.1.92.1.2.1.0 = Counter32: 0
iso.3.6.1.2.1.92.1.2.2.0 = Counter32: 0
root@uru88806lab1061:/home/sti#
```

Fonte: Autor (2023)

Espera-se que quando configurado corretamente o módulo *XBee S2C* coordenador, o resultado apresentado será a *MIB* referente aos dados do consumo energético, posteriormente deverá ser convertido essa *MIB* em parâmetros, com o comando:

- *snmptranslate 'colocar o parâmetro a ser monitorado'*

Após esse comando, será mostrado a chave que deverá ser atribuída ao campo *Chave* no servidor Zabbix. Essa chave faz referencia ao valor que está sendo coletado,

neste caso será dados de entrada, pois é referente a porta habilitada no editor de texto anteriormente. O texto que será exibido segue esse exemplo *IF-MIB::ifInOctets.1*, alterando apenas a numeração de acordo com a porta.

O administrador, posteriormente, deverá entrar novamente no servidor Zabbix e clicar na opção *Coleta de dados*, depois em *Host*, posteriormente em *Itens*, *criar item* no canto superior direito. Na próxima interface, o administrador deverá preencher o campo *Nome*, selecionar no campo *Tipo* e utilizar o *SNMP agent*, atribuir a chave obtida no passo anterior ao campo *Chave*, definir o valor em que esses dados serão atualizados no gráfico, logo após, clicar em adicionar.

Figura 35 – Criação do Item para monitoramento da mensuração

Todos os hosts / Tcc2 Ativo **SNMP** Itens 72 Triggers 28 Gráficos 15 Regras de descoberta 5 Cenários web

Item Etiquetas Pre-processamento

* Nome

Tipo

* Chave Selecionar

Tipo de informação

* Interface do host

* SNMP OID

Unidades

* Intervalo de atualização

Intervalo customizado

Tipo	Intervalo	Período	Ação
☒ Flexível	☐ Agendamento	50s	1-7:00:00-24:00

[Adicionar](#) [Remover](#)

* Período de retenção do histórico Storage period

* Período de retenção das estatísticas Storage period

Mapeamento de valor Selecionar

Preencha o campo do inventário do host

Descrição

Ativo ☒

[Adicionar](#) [Testar](#) [Cancelar](#)

Fonte: Autor (2023)

5 CONSIDERAÇÕES FINAIS

O objetivo geral proposto foi parcialmente atingido, tendo em vista o surgimento de dificuldades em associar um *IP* fixo ao endereço *MAC* (*Media Access Control* - Controle de acesso ao meio) dos módulos *XBee* S2C. Ao tentar associar o *IP* apresentava erro de compatibilidade e, somente com essa associação seria possível realizar a transmissão dos dados coletados pelo sensor *SCT-013* para ferramenta *Zabbix* utilizando o agente *SNMP*, que realizaria a coleta dessas informações nos módulos, e com isso atingir o objetivo geral do presente trabalho. No entanto, foi realizada a comunicação dos módulos e a visualização dos dados obtidos pelo *SCT-013*, no *software XCTU*.

Apesar dessa dificuldade ter sido encontrada, este trabalho proporciona conhecimento sobre o protocolo *SNMP*, *Zabbix*, *Arduino Uno* e sensores. Também oferece uma visão do problema em questão e fornece percepções importantes para a prática profissional e com base no material bibliográfico explorado, foi possível entender e apresentar conceitos que ajudam na conscientização e na elaboração de sistemas de monitoramento que auxiliam as pessoas no processo de desenvolvimento de tecnologia. Vale citar que as informações obtidas neste estudo podem ser aplicadas para melhorar a qualidade do serviço e aumentar a eficiência do processo em questão. A tendência é que este estudo possa ser útil para outras pesquisas futuras e para a tomada de decisão em áreas afins.

Referências

- BANZI, M.; COL. **Arduino**. [S.l.], 2022. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction/>>. Acesso em: 17 de julho de 2022. Citado 6 vezes nas páginas 9, 10, 12, 13, 18 e 19.
- BANZI, M.; COL. **Arduino**. [S.l.], 2022. Disponível em: <<https://www.arduino.cc/en/software>>. Acesso em: 17 de julho de 2022. Citado 2 vezes nas páginas 19 e 20.
- BANZI, M.; COL. **Arduino**. [S.l.], 2022. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 17 de julho de 2022. Citado na página 20.
- CARVALHO, V. A. N.; COL. **Eficiência Energética: Fundamentos e Aplicações**. 1. ed. Fubai Eletrobrás, 2012. 315 p. Disponível em: <<https://www.amazon.com.br/Eficiencia-Energertica-Augusto-Nelson-Carvalho-ebook/dp/B09WL5219G>>. Citado 2 vezes nas páginas 13 e 17.
- CHAGLIA, D. **Diseño de sistema de monitoreo de acceso para prevenir accidentes en centros de educación inicial utilizando módulos xbee-s2c**. [S.l.], 2019. Disponível em: <<http://repositorio.ug.edu.ec/bitstream/redug/46684/1/CHAGLIA%20MEJIA%20LISSETTE%20DANIELA.pdf>>. Acesso em: 28 de setembro de 2022. Citado 2 vezes nas páginas 22 e 23.
- CISCO. **O tráfego mundial de dados móveis aumentará 7 vez entre 2016 e 2021**. [S.l.], 2014. Disponível em: <<http://abntex2.googlecode.com/>>. Acesso em: 13 de setembro de 2014. Citado na página 1.
- CISCO. **Cisco Annual Internet Report (2018–2023) White Paper**. [S.l.], 2020. Disponível em: <<https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>>. Acesso em: 18 de janeiro de 2023. Citado na página 1.
- CURVELLO, A.; COL. **Editorial: Linguagens de Programação para Sistemas Embarcados**. 1. ed. Serra - ES:, 2014. 40 p. Disponível em: <<https://embarcados.com.br/editorial-linguagens-para-sistemas-embarcados/#Wiring/>>. Acesso em: 17 de julho de 2022. Citado na página 18.
- EDISON, L.; COL. **Plano Nacional de Eficiência Energética**. [S.l.], 2014. Disponível em: <<https://www.gov.br/mme/pt-br/assuntos/secretarias/spe/publicacoes/plano-nacional-de-eficiencia-energetica/documentos/plano-nacional-eficiencia-energetica-pdf.pdf/view>>. Acesso em: 15 de setembro de 2022. Citado na página 18.
- EZEQUIEL, E. **ZigBee/IEEE 802.15.4 Summary**. [S.l.], 2004. Disponível em: <<https://pages.cs.wisc.edu/~suman/courses/707/papers/zigbee.pdf>>. Acesso em: 28 de setembro de 2022. Citado na página 22.
- FRANCISCO, B.; CHRISTENSEN, K. **Managing Energy Use in a Network with a New SNMP Power State MIB**: Documentos técnicos e científicos brasileiros compatíveis com as normas abnt. [S.l.], 2008. 46 p. Disponível

em: <https://www.researchgate.net/publication/224345526_Managing_energy_use_in_a_network_with_a_new_SNMP_Power_State_MIB>. Acesso em: 30 de setembro de 2020. Citado na página 1.

HALLIDAY, D.; COL. **Fundamentos de Física - Mecânica**. 1. ed. LTC - Livros Técnicos e Científicos, 2009. 361 p. Disponível em: <<https://portal.vidadesilicio.com.br/wp-content/uploads/2020/04/Arduino-Basico-Vol.1.pdf>>. Acesso em: 17 de julho de 2022. Citado 2 vezes nas páginas 14 e 15.

HAWARY, M. E.-A. E. **Introduction to Eletrical Power Systems**. 2. ed. John Wiley Sons, Inc., 2008. 397 p. Disponível em: <<https://www.wiley.com/en-gb/Introduction+to+Electrical+Power+Systems-p-9780470408636>>. Citado na página 15.

HENRIQUE, C. F.; COL. **Recebendo Notificação de Problemas**. [S.l.], 2001. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/leis_2001/110295.htm>. Acesso em: 15 de setembro de 2022. Citado na página 17.

IGOR, D. A.; BRUNO, L. R. **Simple Network Management Protocol(SNMP): Snmp**. [S.l.], 2011. 7 p. Disponível em: <https://www.gta.ufrj.br/grad/11_1/snmp/index.html>. Acesso em: 24 de setembro de 2020. Citado na página 8.

KIMMO, K.; COL. **Primeiros Passos com Sensores**. 1. ed. Novatec, 2014. 160 p. Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=&id=ew-sBAAQBAJ&oi=fnd&pg=PA112&dq=Primeiros+passos+com+sensores&ots=eW4pD9NFWZ&sig=YVxtBHeCWtVj8IGsp7MsBZu4wVk#v=onepage&q=Primeiros20passos20com20sensores&f=false>>. Citado na página 20.

KUHN, T. S. **The Structure of Scientific Revolutions**. 2. ed. The University of Chicago Press, 1970. 226 p. Disponível em: <E-book:<https://www.amazon.com.br/Structure-Scientific-Revolutions-Thomas-Kuhn/dp/0226458083>>. Citado na página 24.

MATTEDE, H. **Tensão elétrica x voltagem**. [S.l.], 2016. 10 p. Acesso em: 27 de abril de 2022. Citado na página 17.

MICROSOFT. **Protocolo TLS**. [S.l.], 2022. Disponível em: <<https://docs.microsoft.com/pt-br/windows-server/security/tls/transport-layer-security-protocol>>. Acesso em: 09 de Agosto de 2022. Citado na página 9.

MOTA, A. D. **Apostila Arduino Básico**. 1. ed. Serra - ES:: Vida de Silício, 2015. 40 p. Disponível em: <<https://portal.vidadesilicio.com.br/wp-content/uploads/2020/04/Arduino-Basico-Vol.1.pdf>>. Acesso em: 17 de julho de 2022. Citado na página 18.

MURUGESAN, S.; GANGADHARAN, G. R. **Harnessing Green It**. 1. ed. India: John Wiley Sons, Ltd., 2012. Citado 3 vezes nas páginas 1, 4 e 5.

REIS, L. B. D. **Geração de Energia Elétrica**. 2. ed. Manole Ltda., 2015. 480 p. Disponível em: <<https://www.estantevirtual.com.br/livros/lineu-belico-dos-reis/geracao-de-energia-eletrica/280925633>>. Acesso em: 28 de agosto de 2022. Citado na página 14.

REIS, L. J. dos. **Monitoramento com Zabbix**. 2. ed. BRASPORT Livros e Multimídia Ltda, 2020. 250 p. Disponível em: <<https://www.amazon.com.br/Monitoramento-com-Zabbix-2a-edi%C3%A7%C3%A3o-ebook/dp/B08M2KH775>>. Citado na página 24.

SINEM, D. **Sensor de Corrente Alternada com Arduino**. [S.l.], 2017. Disponível em: <<https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/>>. Acesso em: 20 de setembro de 2022. Citado 2 vezes nas páginas 21 e 30.

SOARES, A. E. c. S. **Simple Network Management Protocol: Snmp**. [S.l.], 2011. 9 p. Disponível em: <https://www.gta.ufrj.br/grad/10_1/snmp/index.htm>. Acesso em: 24 de setembro de 2020. Citado 3 vezes nas páginas 5, 6 e 7.

TORRES, G. **Eletrônica para autodidatas, estudantes e técnicos**. 1. ed. Novaterra Editora e Distribuidora LTDA, 2012. 433 p. Disponível em: <<https://www.estantevirtual.com.br/livros/gabriel-torres/eletronica-para-autodidatas-estudantes-e-tecnicos/>>. Acesso em: 10 de junho de 2022. Citado 3 vezes nas páginas 13, 16 e 17.

ZABBIX. **Introdução ao Zabbix**. [S.l.], 2022. Disponível em: <https://www.zabbix.com/documentation/1.8/en/manual/about/overview_of_zabbix>. Acesso em: 09 de Agosto de 2022. Citado 2 vezes nas páginas 8 e 36.

ZABBIX. **Nova trigger**. [S.l.], 2022. Disponível em: <<https://www.zabbix.com/documentation/5.2/pt/manual/quickstart/trigger>>. Acesso em: 09 de Agosto de 2022. Citado na página 11.

ZABBIX. **Novo Host**. [S.l.], 2022. Disponível em: <<https://www.zabbix.com/documentation/5.2/pt/manual/quickstart/host>>. Acesso em: 09 de Agosto de 2022. Citado na página 9.

ZABBIX. **Novo Item**. [S.l.], 2022. Disponível em: <<https://www.zabbix.com/documentation/5.2/pt/manual/quickstart/item>>. Acesso em: 09 de Agosto de 2022. Citado na página 10.

ZABBIX. **Recebendo Notificação de Problemas**. [S.l.], 2022. Disponível em: <<https://www.zabbix.com/documentation/5.2/pt/manual/quickstart/notification>>. Acesso em: 09 de Agosto de 2022. Citado na página 11.

ZABBIX. **Recebendo Notificação de Problemas**. [S.l.], 2022. Disponível em: <<https://www.zabbix.com/documentation/5.2/pt/manual/quickstart/template>>. Acesso em: 09 de Agosto de 2022. Citado na página 12.

ZABBIX. **Requisitos de sistema**. [S.l.], 2022. Disponível em: <<https://www.zabbix.com/documentation/5.2/pt/manual/installation/requirements>>. Acesso em: 23 de Agosto de 2022. Citado na página 36.