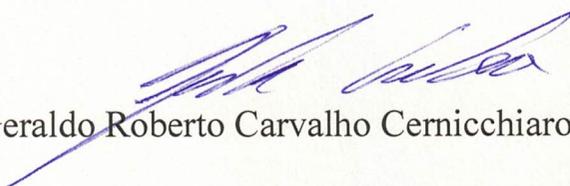


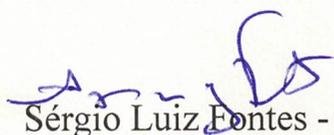
"SISTEMAS SUPERVISÓRIO DE MEDIDAS EM ESTRUTURAS DE BANCO DE DADOS APLICADO A EXPERIMENTOS GEOFÍSICOS"

ÍTALO CLEY BORGES DE SANTANA MAURÍCIO

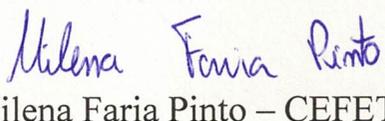
Dissertação de Mestrado Profissional em Física
com ênfase em Instrumentação Científica,
apresentada no Centro Brasileiro de Pesquisas
Físicas do Ministério da Ciência Tecnologia e
Inovação. Fazendo parte da banca examinadora
os seguintes professores:



Geraldo Roberto Carvalho Cernicchiaro – Orientador/CBPF



Sérgio Luiz Fontes - Coorientador/ ON



Milena Faria Pinto – CEFET-RJ



Emanuele Francesco La Terra - ON

Rio de Janeiro, 29 de julho de 2024.



CBPF

Centro Brasileiro
de Pesquisas Físicas

UNIDADE DE PESQUISA DO MCTI

Sistema supervisorio de medidas em estrutura de banco de dados aplicado a experimentos geofísicos

Ítalo Maurício

Rio de Janeiro

Julho - 2024

Centro Brasileiro de Pesquisas Físicas

Ítalo Maurício

Sistema supervisorio de medidas em estrutura de banco de dados aplicado a experimentos geofísicos

Dissertação submetida à Coordenação de Formação Científica do Centro Brasileiro de Pesquisas Físicas como requisito para a obtenção do título de mestre profissional em Física com ênfase em Instrumentação Científica

Orientador: Dr. Geraldo Cernicchiaro

Coorientador: Dr. Sérgio Fontes

Rio de Janeiro

Julho - 2024

Sumário

Sumário	3
Lista de Abreviações e Siglas	4
Resumo	7
1. Introdução	8
2. Sistema de Monitoramento	9
2.1 - Protocolos de Comunicação	11
2.1.1 - Protocolos Seriais	13
2.1.2 - Protocolo TCP/IP	19
3. Banco de Dados	22
3.1 - Redis	23
3.2 - Aplicação do Redis	28
3.3 - Entendendo os Modelos de Dados do Redis	29
3.4 - Arquitetura e Topologia	32
3.5 - Clustering e Disponibilidade	32
3.6 - Console	33
3.7 - Alternativas	34
4. Ambiente Operacional	36
4.1 Máquina Virtual	36
4.1.1 - Tipos de Hypervisors	37
4.2 - Interface com o Usuário	39
4.2.1 - Acesso Remoto	39
4.2.2 - Painel	41
5. Métodos Geofísicos	42
5.1 Eletrorresistividade	42
5.2 Magnetotelúrico - MT	45
6. Metodologia	48
6.1 - Arquitetura do Sistema de Monitoramento	48
6.2 - Processos de Configuração do Sistema	50
6.2.1 - Configuração do Servidor em Nuvem	50
6.2.2 - Inicialização da Máquina Local	52
6.2.3 - Loop Principal- Autenticação e Inicialização do Sistema de Aquisição	54
6.2.4 - Loop Secundário - Configuração do Equipamento	55
6.4.5 - Loop de Medidas	56

7. Sistemas de Medidas Geofísicas	57
7.1 - Medidas de Resistividade	57
7.1.1 - Interface Serial	59
7.1.1.2 - Interface de Controle	60
7.2 - Medidas Magnetotélúricas de Longo Período	60
7.2.1 - Características e Especificações da Unidade Digitalizadora	63
7.2.2 - Características e Especificações do Fluxgate	63
7.2.3 - Interface de Controle	64
8. Resultados e Discussão	66
8.1 - Conexão com Nuvem	66
8.2 - Medidas Realizadas	67
9. Transmissão de Dados - Satélite	71
9.1 Starlink	71
9.2 Aplicação do Sistema de Monitoramento	72
10. Conclusão	75
Referências	76

Lista de Abreviações e Siglas

ACID: Refere-se a um conjunto de propriedades que garantem a confiabilidade das transações em um banco de dados relacional. As propriedades ACID são: Atomicidade, Consistência, Isolamento e Durabilidade.

API: Interface de Programação de Aplicações.

BASE: Contrapõe-se ao ACID. É um acrônimo para *Basically Available, Soft state, Eventually consistent*. Este modelo prioriza a disponibilidade e aceita consistência eventualmente, o que é comum em bancos de dados NoSQL.

Cassandra: Um exemplo de banco de dados NoSQL do tipo *wide column store* (família de colunas).

CSS: Folhas de Estilo em Cascata.

GUI: Interface Gráfica do Usuário.

HTML: Linguagem de Marcação de Hipertexto.

IDE: Ambiente de Desenvolvimento Integrado.

JSON: Notação de Objetos JavaScript.

MT: Magnetotelúrico

NoSQL: *Not Only SQL*. Refere-se a um tipo de banco de dados que não utiliza o modelo relacional. Inclui diversos tipos de bancos de dados, como chave-valor, documento, coluna e grafos.

PyQt: Um conjunto de ferramentas para criar aplicações desktop em Python.

RAID: *Redundant Array of Independent Disks*. É uma tecnologia que utiliza múltiplos discos para melhorar a performance e/ou a confiabilidade do armazenamento de dados.

Redis: Remote Dictionary Server

Remi: Uma biblioteca para criar interfaces de usuário na web em Python.

SCADA: *Supervisory Control and Data Acquisition*

SQL: *Structured Query Language*. É uma linguagem padrão para manipulação de bancos de dados relacionais. Utilizada para consultar, inserir, atualizar e excluir dados em bancos de dados que seguem o modelo relacional.

SSDs: *Solid State Drives*. São dispositivos de armazenamento de dados que utilizam memória flash para armazenar dados de forma persistente. São mais rápidos que os discos rígidos tradicionais.

TCP/IP: *Transmission Control Protocol/Internet Protocol*

Tkinter: Uma biblioteca gráfica para Python.

WxPython: Outra biblioteca gráfica para Python.

DB: *Data Bus*

DCE: Equipamento de Terminação de Circuito de Dados (*Data Circuit-terminating Equipment*).

DTE: Equipamento Terminal de Dados (*Data Terminal Equipment*).

EAI: Aliança das Indústrias Eletrônicas (*Electronic Industries Alliance*).

RS-232: Padrão de comunicação serial para equipamentos de dados.

RTS/CTS: Pronto para enviar / Pronto para receber, usado para controlar o fluxo de dados.

USB: *Universal Serial Bus*.

V.24: Norma de comunicação serial, frequentemente usada em conjunto com RS-232.

Resumo

Resumo - O sistema supervisorio apresentado nesta pesquisa, aplicado ao monitoramento e aquisiçao de dados geofisicos, desenvolvido em Python e baseado em microprocessadores, possibilita o monitoramento e reconfiguracao dos sistemas de medidas, armazenamento de dados em nuvem e transmissao de dados em tempo real para multiusuarios, simultaneamente. A estrutura de dados chave/valor da arquitetura do modulo de comunicacao e implementada no banco de dados NoSQL, Redis - *Remote Dictionary Server*, adequado para monitoramento de sistemas em tempo real. Os sistemas de medidas geofisicas estabelecem conectividade com o microprocessador de comunicacao serial e o servidor hospedeiro do banco de dados. Os dados adquiridos pelos equipamentos geofisicos sao enviados ao microprocessador, armazenados em um diretorio local e em seguida alocados em chaves do banco de dados. Desta maneira, os multiusuarios com acesso a estrutura de dados podem enviar comando de medidas aos equipamentos geofisicos, consultar seus respectivos status, alem de terem acesso aos dados adquiridos pelos sistemas de medidas instalados em campo. O prototipo foi testado com sucesso em sistemas de medidas geofisicas utilizados em projetos de pesquisa do Observatorio Nacional.

Palavras-chaves: *Sistema supervisorio, sistemas de medidas, banco de dados, Redis, microprocessador, equipamentos geofisicos.*

Abstract - The supervisory system presented in this work, applied to the monitoring and acquisition of geophysical data, developed in Python and based on microprocessors, enables the monitoring and reconfiguration of measurement systems, data storage in the cloud, and real-time data transmission to multiple users simultaneously. The key/value data structure of the communication module architecture is implemented in the NoSQL database, Redis - Remote Dictionary Server, suitable for real-time system monitoring. Geophysical measurement systems establish connectivity with the serial communication microprocessor and the hosting server of the database. The data acquired by the geophysical equipment is sent to the microprocessor, stored in a local directory and subsequently allocated to database keys. In this way, multiple users with access to the data structure can send measurement commands to the geophysical equipment, check their respective statuses, and access the data acquired by the measurement systems installed in the field. The prototype was successfully tested on geophysical measurement systems used in research projects at the National Observatory.

Keywords: *Supervisory system, measurement systems, database, Redis, microprocessor, geophysical equipment.*

1. Introdução

Alguns equipamentos geofísicos, como resistivímetros e sistemas magnetotelúricos, desempenham um papel crucial na aquisição de dados geofísicos para estimar a condutividade elétrica da subsuperfície [2][7]. Esses valores de condutividade são essenciais para caracterizar a subsuperfície e avaliar possíveis riscos, como deslizamentos de encostas, rupturas de barragens, dinâmicas de aquíferos e condições de aterros sanitários. Além disso, esses métodos geofísicos têm aplicações importantes na identificação de depósitos minerais e hidrocarbonetos, especialmente no caso de sistemas magnetotelúricos.

Normalmente, a aquisição de dados geofísicos é realizada por operadores em campo, que interagem diretamente com os equipamentos. Contudo, em ambientes perigosos para os operadores, a coleta diária dessas medidas pode ser inviabilizada devido aos desafios encontrados. Nesse contexto, os sistemas de monitoramento remoto possibilitam a supervisão contínua dos sistemas de medida 24 horas por dia, permitindo a atuação de diferentes operadores ou algoritmos automatizados de detecção de falhas. Estas possibilidades ampliam significativamente a funcionalidade e a capacidade operacional dos equipamentos geofísicos disponíveis no mercado.

A crescente quantidade de estações geofísicas magnetotelúricas e de eletrorresistividade instaladas em atividades de campo, associadas a projetos de pesquisa, tem impulsionado a necessidade de supervisão desses sistemas. O monitoramento em tempo real desses equipamentos é crucial para garantir medidas mais precisas, a continuidade dos dados e a caracterização das estruturas geológicas monitoradas, além de identificar possíveis falhas dos sistemas de medidas. A possibilidade de reconfiguração remota dos equipamentos para diferentes *setups*, sem intervenção presencial dos operadores, não só reduz os custos das campanhas, mas também otimiza os recursos financeiros.

Diante desse cenário, o módulo de comunicação e transmissão de dados proposto nesta pesquisa oferece um sistema de acesso seguro e robusto aos dados geofísicos. Esse módulo é capaz de realizar medidas e transmitir os dados adquiridos em campo para estações de controle e supervisão localizadas fora da área de risco. A transmissão e acesso aos dados é efetuada de maneira segura e econômica, baseada em uma arquitetura de banco de dados, o que viabiliza a instalação do módulo de comunicação em diversas estações de aquisição de medidas geofísicas.

Um dos principais objetivos do módulo de comunicação desenvolvido nesta pesquisa é aumentar a capacidade operacional dos sistemas de medidas, especialmente porque a maioria desses dispositivos não possui um sistema de transmissão e monitoramento de dados. Além disso, a possibilidade de coletar dados, reconfigurar os equipamentos e monitorar a dinâmica dos sistemas de aquisição de forma remota, proporcionada por este módulo, agrega robustez e qualidade às medições realizadas em campo.

O protótipo utilizado nas campanhas de aquisição de dados magnetotelúricos do projeto de pesquisa PETRONAS MULTIFÍSICA do Observatório Nacional garantiu a supervisão da estação base em tempo real e acesso às medidas remotamente. Tais contribuições proporcionaram avaliação contínua das medidas, acesso diário aos dados e seu respectivo processamento, algo que até então não tinha sido realizado em projetos de pesquisa da instituição.

2. Sistema de Monitoramento

É comum ouvir falar sobre o constante monitoramento ao qual estamos submetidos, especialmente agora que a Internet das Coisas (*Internet of Things - IoT*) se tornou cada vez mais acessível para um público amplo. A afirmação de que "nunca antes na história fomos tão monitorados" é amplamente aceita. O termo "monitoramento" é aplicado em diversos contextos, desde redes sociais, relógios inteligentes e câmeras de vigilância até a inteligência artificial de algoritmos e outras ferramentas digitais. No entanto, a essência da palavra – o ato de "estar sendo acompanhado" – permanece inalterada. No contexto deste projeto, o sistema de monitoramento proposto mantém essa essência. Estaremos, de fato, acompanhando os equipamentos de medição instalados em campo em tempo real. Além disso, teremos a capacidade de interferir nos sistemas de medição e reconfigurá-los conforme necessário, enviando comandos remotamente. [54].

O conceito de monitoramento adotado nesta pesquisa apresenta notáveis semelhanças com os princípios e técnicas utilizados no monitoramento de sistemas e no controle industrial. Em instalações industriais, as variáveis são constantemente medidas, e conclusões são extraídas após o processamento e a interpretação dos dados. Esses sistemas de monitoramento seguem rigorosamente uma série de normas e padrões estabelecidos pela indústria.

Os Sistemas de Supervisão e Aquisição de Dados - SCADA (*Supervisory Control and Data Acquisition*) também referidos como sistemas supervisórios SCADA, empregam *softwares* para monitorar e supervisionar variáveis e dispositivos em processos industriais, não industriais e equipamentos interligados por meio de servidores e *drivers* de comunicação específicos [46][47]. Na Figura 1, há um fluxograma simplificado que ilustra a estrutura e o funcionamento do modelo SCADA proposto nesta pesquisa [11][26].

Os sistemas supervisórios baseados em SCADA são amplamente utilizados em operações de plantas industriais, sendo especialmente frequentes na automação e controle industrial. A automação e controle industrial envolvem o uso de tecnologias para operar e gerenciar processos industriais com mínima intervenção humana. Esses sistemas são amplamente aplicados em diversas indústrias, como a manufatureira, petroquímica, farmacêutica e alimentícia, com o objetivo de aumentar a eficiência, segurança e qualidade dos produtos.

A automação e controle industrial proporcionam um aumento significativo na eficiência ao reduzir o tempo de produção e melhorar o uso de recursos, ao mesmo tempo em que garantem a consistência e precisão nos processos, resultando em produtos de melhor qualidade. Além disso, esses sistemas minimizam o risco de acidentes através do monitoramento e controle em tempo real de variáveis críticas, aumentando a segurança. A automação também contribui para a redução de custos, diminuindo a necessidade de intervenção manual, o desperdício de material e otimizando o consumo de energia.

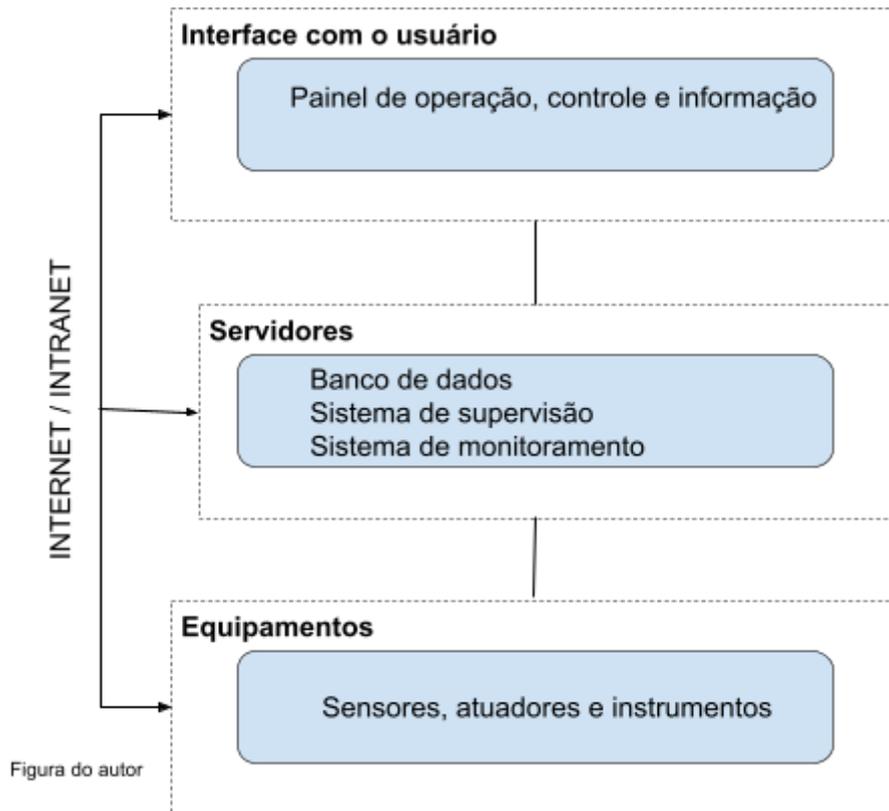


Figura 1 - Arquitetura simplificada em três camadas do sistema SCADA

No contexto científico, as aplicações desenvolvidas nesta pesquisa mostraram que, por meio do SCADA, é possível realizar medições dos instrumentos instalados em campo, criar gráficos de tendências, manter registros históricos das variáveis do processo, plotar séries temporais e realizar análises espectrais e estatísticas dos dados coletados. A capacidade de acessar os dados em tempo real remotamente aprimora o diagnóstico da dinâmica dos processos e o controle dos sistemas de medição monitorados [55]. Essas análises proporcionam uma resposta ágil para a tomada de decisões assertivas, permitindo correções ou a manutenção dos processos de forma eficiente e estável [26].

Os sistemas SCADA requerem um meio físico para realizar a aquisição de dados dos sistemas de medidas *in situ* e a transmissão das informações ao controlador. Geralmente, esse meio físico adere a protocolos de comunicação como RS-232, USB ou *Ethernet*. Inerentemente, os sistemas SCADA oferecem robustez, praticidade e modularidade, sendo compatíveis com diferentes sistemas operacionais, *hardwares* e sistemas de medidas, mantendo-se constantemente atualizados. Além disso, é crucial manter as informações e dados adquiridos em um nível elevado de segurança para garantir a privacidade e veracidade das informações. Os SCADA normalmente são desenvolvidos por empresas comerciais com base em códigos e arquiteturas fechadas. No entanto, neste projeto, o SCADA apresenta uma arquitetura sofisticada, baseada em código aberto, que pode ser facilmente replicada, mantendo os níveis necessários de segurança para preservar as informações dos sistemas de medidas e os dados adquiridos. A autenticação de login e senha do usuário no banco de dados é suficiente para garantir a segurança do sistema supervisorio e evitar violações na estrutura de dados do Redis [37][38].

Os recursos empregados na implementação do SCADA deste projeto são exclusivamente constituídos por softwares e *hardwares* de código aberto. Destaca-se o computador de placa única Raspberry Pi 3 Model B [32], utilizado para estabelecer a comunicação entre os sistemas de aquisição, equipamentos geofísicos, softwares, servidores e os processos de controle desenvolvidos na linguagem Python 3.[1][22][28]

2.1 - Protocolos de Comunicação

O conceito de protocolo de comunicação é amplamente conhecido e frequentemente comparado à linguagem verbal usada por grupos étnicos. Assim como na comunicação humana, onde regras, procedimentos e a compreensão dos significados de palavras e símbolos são fundamentais para a interação, no campo das máquinas e da automação, a existência de protocolos de comunicação é igualmente essencial.

Esses protocolos asseguram que as "conversas" entre sistemas sejam precisas, eficientes e isentas de perda de informações. Os dispositivos utilizados na automação industrial e residencial estão se tornando cada vez mais inteligentes, incorporando recursos da informática para controlar processos e fornecer à sala central de controle informações detalhadas sobre a dinâmica da produção. Essa integração de tecnologias promove uma comunicação eficaz e contribui para a evolução da automação em diversos setores [19].

Quando máquinas e sistemas de automação interagem, é essencial que haja uma padronização na forma como trocam informações. Isso não apenas assegura a clareza na comunicação, mas também promove a interoperabilidade entre diferentes dispositivos e fabricantes. Alguns aspectos adicionais relacionados aos protocolos de comunicação em automação são:

Padronização: A existência de padrões de comunicação, como Modbus, Profibus, OPC (*OLE for Process Control*), e outros, permite que dispositivos de diferentes fabricantes interajam de maneira consistente.

Eficiência: Protocolos eficientes garantem uma comunicação rápida e confiável entre os dispositivos, minimizando atrasos e perdas de dados.

Segurança: Muitos protocolos incluem medidas de segurança para proteger a integridade e confidencialidade das informações transmitidas.

Interoperabilidade: A capacidade de dispositivos de diferentes fabricantes trabalharem juntos é fundamental, especialmente em ambientes industriais onde uma variedade de equipamentos pode ser utilizada.

Comando e Controle: Os protocolos não apenas facilitam a troca de informações, mas também especificam como os dispositivos devem ser comandados e controlados.

Monitoramento Remoto: Permitem o monitoramento e controle remoto de sistemas, o que é crucial em automação industrial e residencial moderna.

Integração de Sistemas: Protocolos de comunicação facilitam a integração de sistemas diversos, como sistemas de controle, sensores, atuadores e sistemas de monitoramento.

IoT e Indústria 4.0: Em ambientes de Internet das Coisas (IoT) e Indústria 4.0, protocolos como MQTT são comuns para a comunicação entre dispositivos conectados.[6]

Ao adotar protocolos de comunicação sólidos, os sistemas de automação podem funcionar de maneira mais eficiente, ser mais flexíveis e adaptáveis às mudanças nas demandas do ambiente, e proporcionar uma gestão mais eficaz dos processos industriais e residenciais.

Os protocolos de comunicação e transmissão de dados são conjuntos de regras que especificam como os dispositivos devem comunicar e trocar informações entre si. Esses protocolos são essenciais para garantir uma comunicação eficiente e confiável em redes de computadores e sistemas distribuídos. Os protocolos de comunicação mais comuns são brevemente descritos nos parágrafos posteriores [20].

O TCP/IP (*Transmission Control Protocol/Internet Protocol*) serve como a espinha dorsal da Internet e redes locais, desempenhando um papel fundamental na transmissão de dados. Ele fragmenta a comunicação em pacotes, garantindo uma entrega confiável, sendo essencial para a estrutura e operação da Internet.

O HTTP (*Hypertext Transfer Protocol*) é o protocolo utilizado para a transferência de dados na World Wide Web. Além de definir como os navegadores e servidores interagem, o HTTPS (*Hypertext Transfer Protocol Secure*) é uma versão segura do HTTP, empregando criptografia SSL/TLS para proteger a comunicação, assegurando a privacidade e integridade dos dados transmitidos.

O FTP (*File Transfer Protocol*) é dedicado à transferência de arquivos entre computadores, viabilizando o upload e download de arquivos. Enquanto isso, o SMTP (*Simple Mail Transfer Protocol*) é responsável pelo envio de e-mails, utilizado pelos servidores de e-mail para encaminhar mensagens.

No universo do recebimento de e-mails, temos o POP3 (*Post Office Protocol 3*) e o IMAP (*Internet Message Access Protocol*). Enquanto o POP3 baixa e remove mensagens do servidor, o IMAP as mantém no servidor, proporcionando diferentes abordagens para gerenciar o correio eletrônico.

O DNS (*Domain Name System*) desempenha um papel vital, traduzindo nomes de domínio em endereços IP, facilitando a navegação na internet usando nomes em vez de números IP.

O DHCP (*Dynamic Host Configuration Protocol*) é responsável por atribuir automaticamente endereços IP e configurações de rede a dispositivos em uma rede, simplificando a administração da infraestrutura de rede.

Para o gerenciamento de dispositivos em redes IP, temos o SNMP (*Simple Network Management Protocol*), que permite monitorar e controlar dispositivos de rede.

No âmbito das comunicações sem fio, o *Bluetooth* é um padrão que facilita a transferência de dados entre dispositivos próximos, sendo comumente utilizado em smartphones, fones de ouvido e impressoras.

O Modbus, por sua vez, é um protocolo de comunicação empregado em sistemas de automação industrial, possibilitando a interação entre dispositivos de controle, como PLCs (Controladores Lógicos Programáveis).

Em sistemas embarcados e veículos, o CAN (*Controller Area Network*) é um protocolo de comunicação projetado para garantir comunicação confiável em ambientes ruidosos.

O *WebSocket* é um protocolo que viabiliza a comunicação bidirecional em tempo real entre o

navegador e o servidor, sendo amplamente utilizado em aplicações web interativas.

O MQTT (*Message Queuing Telemetry Transport*) é um protocolo leve de mensagens, especialmente útil para dispositivos com recursos limitados, sendo uma escolha comum em IoT (*Internet of Things*) [6].

O *Near Field Communication* - NFC permite a comunicação sem fio de curto alcance entre dispositivos, encontrando aplicações em pagamentos móveis e transferência de dados. Essa diversidade de protocolos desempenha papéis específicos, contribuindo para a riqueza e eficiência da comunicação em diversos contextos tecnológicos.

Esses protocolos assumem funções específicas em diversos cenários, assegurando uma comunicação eficaz e segura em uma variedade de ambientes e aplicações.

A seguir, serão discutidos de forma concisa os tipos de protocolos integrados no sistema de monitoramento destacado neste projeto. O propósito é oferecer ao leitor uma compreensão sucinta e prática do funcionamento desses protocolos.

2.1.1 - Protocolos Seriais

O protocolo RS-232 é um padrão amplamente adotado para comunicação serial, proporcionando a transmissão eficiente de dados binários entre dispositivos por um meio físico, mais precisamente por pares de fios. Esse protocolo estabelece as diretrizes para a transmissão, recepção e interpretação dos dados.[3][5]

No protocolo de comunicação RS-232, existem duas linhas dedicadas, uma para transmissão e outra para recepção de dados. Essas linhas são utilizadas para o envio e recebimento de dados, respectivamente. A transmissão de dados ocorre de maneira sequencial, em formato de bits. Cada transmissão inicia com um bit de partida, seguido pelos bits de dados e finaliza com um ou mais bits de parada. A velocidade de transferência de dados é determinada pela taxa de transmissão, medida em *baud rate*. Ajustar corretamente essa taxa é de suma importância para garantir uma comunicação sincronizada entre os sistemas, evitando perdas ou alterações nos dados transmitidos [49].

Adicionalmente, o protocolo RS-232 define outros parâmetros importantes para a comunicação entre dispositivos. Entre eles, destacam-se o número de bits de dados, a opção de paridade e o controle de fluxo, que é opcional. Esses parâmetros são fundamentais para assegurar uma comunicação precisa e confiável [12][13]. É importante ressaltar que o protocolo RS-232 é amplamente utilizado em aplicações industriais e comerciais, devido à sua robustez e confiabilidade. No entanto, devido às suas limitações de distância e velocidade, novos padrões, como o RS-485 e o *Ethernet*, estão sendo cada vez mais adotados em aplicações modernas.

Resumidamente, o protocolo RS-232 define as normas para a transmissão de dados em comunicação serial, abrangendo a estrutura dos dados, a velocidade de transmissão e outros parâmetros essenciais. Sua aplicação abrange uma variedade de dispositivos, incluindo modems, impressoras, terminais de computador e diversos outros.

Embora tenha sido estabelecido no início da década de 60, por volta de 1962, o padrão RS-232, também conhecido como EAI RS-232C ou V.24, ainda é amplamente utilizado na comunicação entre máquinas, computadores e controladores. No módulo de comunicação proposto nesta pesquisa, o RS-232 foi empregado para estabelecer a comunicação entre os

equipamentos geofísicos e o microprocessador. Para realizar esta conexão foi necessário utilizar o cabo conversor USB/RS-232 haja vista que o microprocessador utilizado tem no seu *hardware* a porta de comunicação USB.

O protocolo de comunicação RS-232 é caracterizado por vários aspectos essenciais que delineiam seu funcionamento. Utilizando sinais elétricos, o RS-232 representa dados binários, onde um sinal positivo (tensão positiva) denota o bit 0, enquanto um sinal negativo (tensão negativa) denota o bit 1. Os conectores RS-232, geralmente DB-9 (9 pinos) ou DB-25 (25 pinos), possuem pinos designados para transmitir e receber dados, controle de fluxo, terra e outras funções.

Na transmissão assíncrona do RS-232, não há um clock compartilhado entre os dispositivos. Cada byte de dados é precedido por um bit de início e seguido por bits de parada. Configurável para diferentes tamanhos de palavra, geralmente 7 ou 8 bits, o RS-232 pode ou não incluir bits de paridade para verificação de erro, e normalmente utiliza um ou dois bits de parada para indicar o final de um byte.

A taxa de transmissão, medida em "bauds" ou "bps" (bits por segundo), deve ser configurada de forma idêntica em ambos os dispositivos para garantir uma comunicação adequada. Mecanismos como RTS/CTS (*Ready To Send/Clear To Send*) podem ser empregados para controlar o fluxo de dados e evitar sobrecargas no buffer.

A distância efetiva de transmissão do RS-232 é limitada, geralmente a alguns metros, devido à sensibilidade ao ruído e perda de sinal. Em termos de aplicações, o RS-232 é utilizado em diversas situações, desde conexões de modem até interfaces de comunicação entre dispositivos de controle e sistemas de medidas industriais.

Apesar de ter sido amplamente utilizado, o RS-232 tem sido em grande parte substituído por tecnologias mais modernas, como USB e *Ethernet*, em muitos cenários. No entanto, ainda é encontrado em alguns dispositivos e em ambientes industriais. Uma das abordagens mais simples para a troca de dados entre máquinas e computadores, e vice-versa, é através de uma linha de comunicação serial, sendo o padrão RS-232 proeminente nesse contexto. Esse padrão é comumente utilizado em aplicações de curta distância, especialmente em situações que envolvem a comunicação entre máquinas, sensores e os computadores responsáveis por sua supervisão.

O protocolo RS-232 pressupõe a presença de aterramento entre os dois dispositivos envolvidos na comunicação de dados, conhecidos como DTE (*Data Terminal Equipment*) e DCE (*Data Circuit-terminating Equipment*). O DTE, frequentemente representado por um PC, assume a função de terminal de dados, enquanto o DCE, que pode ser um dispositivo de comunicação externo, é responsável por receber e/ou enviar sinais de controle ao PC.

Os sinais transmitidos pelo padrão RS-232 são representados por níveis de tensão e seguem uma forma serial, o que implica na transmissão sequencial de bits, um por vez, através de uma única linha de comunicação bilateral, operando em modo "full-duplex". As tensões na faixa de +3 a +15 volts são reconhecidas como níveis baixos, enquanto tensões de -3 a -15 volts são reconhecidas como níveis altos [3]. Vale observar que a lógica negativa, utilizada nesse caso, é implementada visando proporcionar maior segurança na transmissão dos dados. Algumas documentações técnicas podem mencionar valores de tensão estendidos até -25 V e +25 V.

Na figura 2.1 temos um gráfico em que as faixas de sinais são representadas.

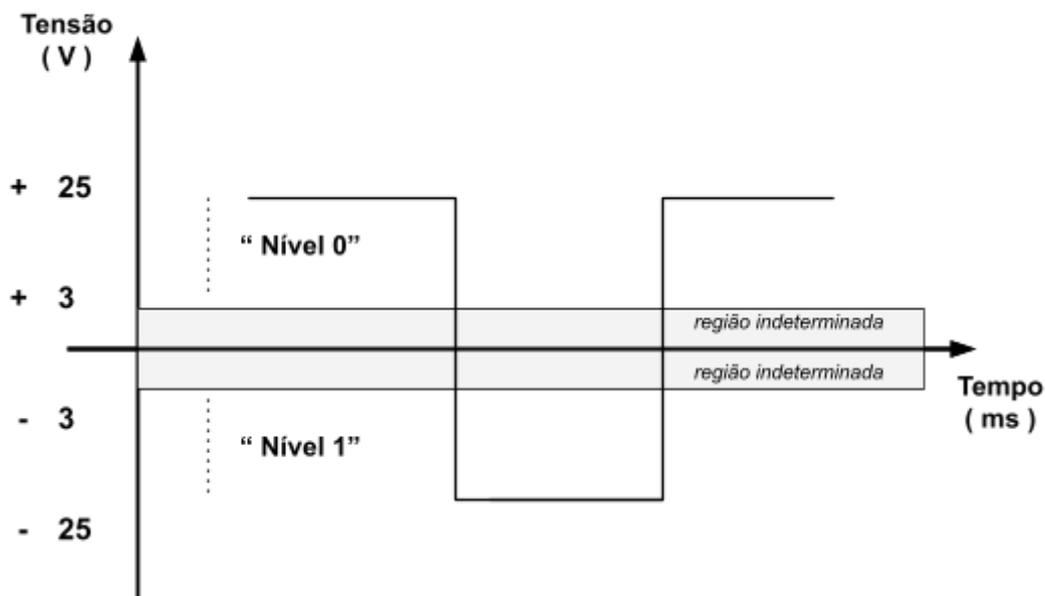


Figura do autor

Figura 2.1 - Representação da transmissão de sinais no protocolo RS-232

É importante notar que a faixa entre -3 V e +3 V representa uma região de transição, "indeterminada", onde não há um reconhecimento claro de nível lógico, similar à faixa presente em circuitos lógicos digitais convencionais. É interessante observar que algumas variações ocorrem nessa faixa. Por exemplo, no padrão V.10, a faixa morta está definida entre +0,3 V e -0,3 V, e existem dispositivos que podem operar com níveis de tensão ainda menores, abaixo de 1 V. A utilização de tensões positivas e negativas atribui ao sistema a designação de "diferencial". A adoção de sinais diferenciais tem o propósito de minimizar os efeitos de ruídos, bem como a influência de aterramentos. A ampla faixa de valores de tensão possibilita que os sinais sejam transmitidos por longas distâncias sem uma atenuação significativa que comprometa sua amplitude, preservando, assim, a integridade do sinal.

As conexões entre os dispositivos DTE e DCE em um sistema RS-232 fazem uso de diversos tipos de conectores, cujas pinagens são padronizadas. Neste projeto, foi empregado o conector DB9, conforme ilustrado na Figura 2.1.1. A pinagem deste conector DB-9 segue a norma EIA-574, sendo comumente encontrado nas portas de comunicação de PCs e utilizado em comunicação de dados (RS-232/V.24).

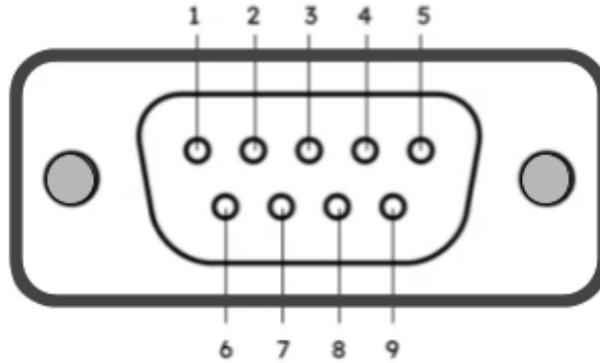


Figura do autor

Figura 2.1.1 - Conector DB9 utilizado no protocolo RS232

- | | | |
|--------------------------------------|--------------------------------------|----------------------------------|
| 1 - Data Carrier Detect - DCD | 4 - Data Terminal Ready - DTR | 7 - Request to Send - RTS |
| 2 - Received Data - RX | 5 - Signal ground - GND | 8 - Clear to Send - CTS |
| 3 - Transmitted - TX | 6 - Data set Ready - DSR | 9 - Ring Indicator - RI |

O dispositivo opera no modo de terminal simples, onde cada linha apresenta um único driver e um receptor. A extensão máxima suportada do cabo é de 15 metros, com uma velocidade máxima de transmissão de dados atingindo 20 kb/s. A tensão máxima de saída é de +/- 25 V, enquanto o sinal de saída do driver, quando carregado, varia entre +/- 5 V e +/- 15 V.

Além disso, a impedância de carga do driver é especificada entre 3 a 7 k Ω , e a taxa de crescimento máxima é de 30 V/ps. A resistência de entrada do receptor varia de 3 k a 7 k Ω . Essas especificações fornecem uma compreensão abrangente do desempenho e das características elétricas do dispositivo em questão.

O padrão RS-232, intrinsecamente vinculado à transmissão de bits associados ao padrão ASCII (American Standard Code for Information Interchange), tem como objetivo padronizar a representação de letras, números, acentos, sinais diversos e alguns códigos de controle, baseados no alfabeto romano. O ASCII é uma codificação de caracteres que atribui um valor numérico único a cada letra, número e símbolo comumente utilizado em computadores e dispositivos de comunicação. Essa codificação possibilita que computadores e dispositivos de comunicação processem texto de maneira consistente, independentemente do sistema operacional ou do dispositivo, incluindo caracteres alfanuméricos, pontuação, símbolos matemáticos e de controle, sendo amplamente empregado em comunicações de dados, programação de computadores, processamento de texto e em diversas áreas da computação.

A Figura 2.1.2 ilustra como um caractere ASCII é transmitido. A forma mais comum de transmissão de sinais é assíncrona, onde o transmissor não precisa estar sincronizado com o receptor, pois este é informado sobre o início e o fim de cada "pacote de dados", utilizando bits de start e stop.

Dessa forma, o sinal é composto por bits individuais transmitidos sequencialmente em um "pacote" de tamanho definido no formato ASCII. A quantidade de bits em cada pacote pode variar de 5 a 8, sendo enviados após um sinal de start, reconhecido quando a linha, normalmente no nível 1 (negativa), transita para o nível 0 (positiva).

No flanco descendente do sinal (passagem de 1 para 0), ocorre o reconhecimento do início da transmissão da mensagem. No final do pacote de bits de dados, é enviado um bit de paridade para verificar a correção da informação, seguido por um bit de stop.

Se esse foi o último pacote, o nível da linha permanece alto (1); no entanto, se um novo pacote precisa ser enviado, ocorre uma nova transição do nível 1 para o zero, reconhecida como início, e o processo se repete. O sincronismo desse modo de transmissão é baseado no bit de start de cada pacote. Contudo, para altas velocidades, o formato utilizado inclui um ou dois bytes de sincronismo, conforme apresentado na Figura 2.1.2

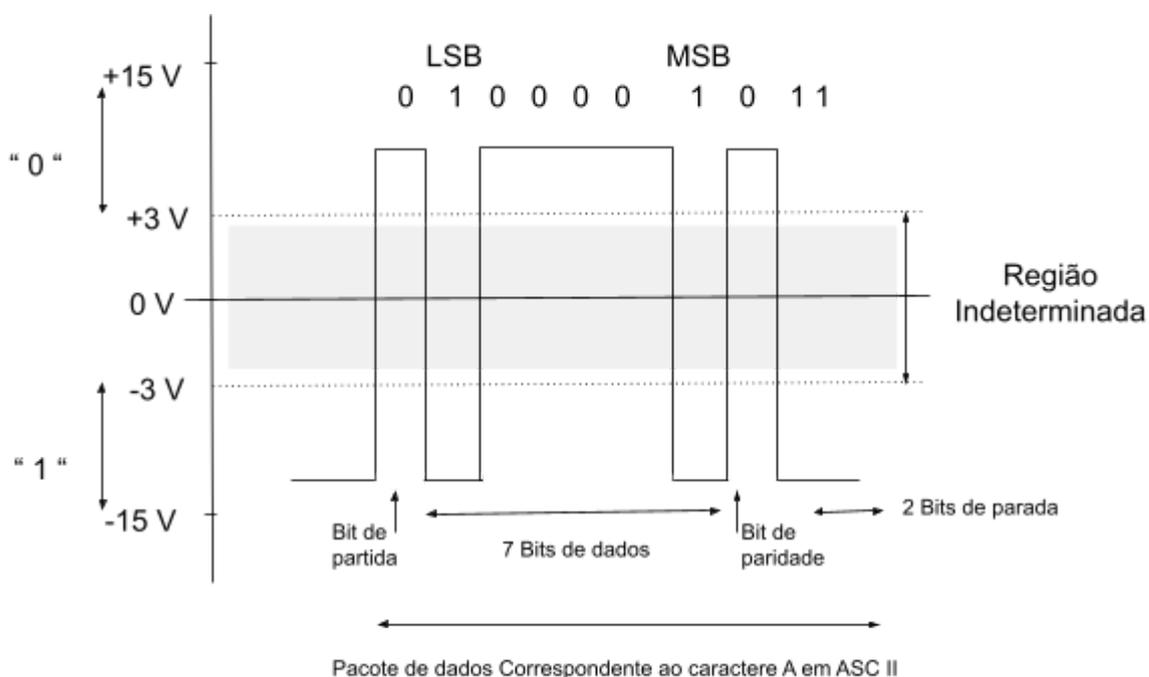


Figura do autor

Figura 2.1.2 - Representação da transmissão de caracteres através do protocolo RS-232

No contexto do código ASCII, esse byte é representado por 0 0010110, seguido então por dois bytes de dados. Para manter uma transmissão contínua, bytes de sincronismo devem ser intercalados conforme necessário [50][51].

Interfaces USB - Universal Serial Bus

As interfaces USB (Universal Serial Bus) são uma tecnologia de conectividade amplamente adotada, proporcionando transferência de dados e alimentação para diversos dispositivos eletrônicos. Desenvolvida para simplificar e unificar interfaces, a USB oferece uma solução versátil, conectando periféricos e dispositivos a computadores e outros sistemas eletrônicos.

A topologia da USB apresenta uma arquitetura estrela, onde um hub central se conecta a dispositivos periféricos. Os conectores USB, como o Type-A e o Type-B, juntamente com o mais recente Type-C, estão presentes em diversos dispositivos, proporcionando flexibilidade e praticidade. Diferentes versões da USB, como USB 1.1, USB 2.0, USB 3.0, USB 3.1 e USB 3.2, oferecem melhorias na velocidade de transferência de dados. Cada geração subsequente aumenta a largura de banda, proporcionando taxas de transferência mais rápidas. A capacidade de fornecer energia aos dispositivos é uma característica distintiva da USB. A especificação Power Delivery (PD) permite a transferência de energia bidirecional, possibilitando carregamento e fornecimento de energia entre dispositivos.

A técnica de multiplexação permite que um único conector USB suporte diversas funções simultaneamente, como transferência de dados, alimentação, áudio e vídeo. Essa abordagem simplifica a conectividade em dispositivos modernos. A capacidade de hot swapping é uma característica fundamental da USB, permitindo a conexão e desconexão de dispositivos durante a operação sem a necessidade de reinicialização. A USB utiliza classes de dispositivos padronizadas para facilitar a interoperabilidade. Exemplos incluem dispositivos de armazenamento em massa, áudio, impressoras, câmeras, teclados, entre outros. O protocolo de comunicação USB adota uma abordagem mestre-escravo, onde o host controla a comunicação com os dispositivos periféricos. O host gerencia a alocação de largura de banda e coordena as transmissões de dados.

A evolução contínua da tecnologia USB, exemplificada pelo USB4, proporciona velocidades de transferência ainda mais rápidas e suporte a uma ampla gama de funcionalidades. Em resumo, as interfaces USB desempenham um papel crucial na conectividade moderna, impulsionando a praticidade e eficiência em uma variedade de contextos.

Adaptador USB/RS-232 - Uma Ponte Entre o Passado e o Presente da Comunicação Serial

A evolução rápida da tecnologia muitas vezes deixa para trás interfaces e padrões mais antigos, mas a necessidade de conectar dispositivos legados persiste. Um exemplo clássico desse cenário é a transição da comunicação serial RS-232 para as interfaces USB (Universal Serial Bus). O adaptador USB/RS-232 surge como uma solução engenhosa para superar as diferenças tecnológicas e garantir a compatibilidade entre dispositivos mais antigos e computadores modernos. Neste projeto, foi essencial empregar o adaptador USB/RS-232 juntamente com seus drivers correspondentes para estabelecer a conexão e facilitar a comunicação entre os equipamentos geofísicos e a Raspberry Pi 3 Model B.

A comunicação serial RS-232, uma vez dominante em muitos dispositivos eletrônicos e industriais, cedeu lugar à proliferação das portas USB em computadores e laptops. Essa mudança, embora representasse um avanço em termos de velocidade e facilidade de uso, deixou um legado de dispositivos equipados com portas RS-232 que ainda desempenham funções críticas em diversas aplicações. O adaptador USB/RS-232 atua como uma ponte entre esses dois mundos distintos. Este pequeno dispositivo converte os sinais USB modernos em sinais RS-232, permitindo que dispositivos mais antigos continuem a se comunicar com sistemas mais recentes.

A arquitetura do adaptador USB/RS-232 geralmente inclui um chip conversor USB-Serial e uma porta RS-232, muitas vezes apresentando conectores DB-9 ou DB-25. O chip conversor desempenha um papel crucial na tradução de dados, convertendo os pacotes USB em sinais seriais RS-232 compreensíveis para dispositivos conectados.

Uma das principais vantagens do adaptador é a sua facilidade de uso plug-and-play. Não requer instalação complicada de drivers em sistemas operacionais modernos, tornando-se uma solução acessível e prática. No entanto, desafios como limitações de velocidade de transmissão e questões de energia podem surgir, dada a natureza diferente das interfaces.

O adaptador USB/RS-232 encontra aplicações em uma variedade de setores, desde a indústria até entusiastas de eletrônica. Ele é frequentemente utilizado para programação de dispositivos embarcados, configuração de equipamentos industriais, e até mesmo para conectar instrumentos de medição mais antigos a computadores modernos.

À medida que a tecnologia continua evoluindo, é interessante observar como as necessidades de conectividade entre interfaces antigas e modernas serão realizadas. O adaptador USB/RS-232, por enquanto, continua sendo uma ferramenta valiosa, preservando a funcionalidade de dispositivos em um ambiente cada vez mais digital.

2.1.2 - Protocolo TCP/IP

O TCP/IP, estabelecido em 1969, é um conjunto de protocolos de comunicação utilizado para facilitar a troca de dados entre dispositivos conectados à mesma rede. Originado a partir de dois protocolos fundamentais, o TCP (Transmission Control Protocol - Protocolo de Controle de Transmissão) e o IP (Internet Protocol - Protocolo de Internet, ou também conhecido como protocolo de interconexão), esse conjunto de protocolos desempenha um papel crucial no módulo de comunicação para estabelecer conexão e transmitir dados entre o banco de dados e o Redis e os sistemas implantados em campo [20].

Os protocolos que compõem o TCP/IP podem ser associados facilmente ao modelo de rede de computadores *OSI - Open System Interconnection*, conforme apresentado na Tabela 2.1. Cada camada é encarregada de um conjunto específico de tarefas, transmitindo informações para as camadas superiores. Quanto maior o número da camada, mais próxima ela está do usuário, sendo a camada de aplicação a mais próxima.

Modelo OSI e TCP/IP

Camada	Exemplo
4 - Aplicação (5 ^a , 6 ^a e 7 ^a camada OSI)	HTTP, HTTPS, FTP, DNS Essa parte contém todos os protocolos para um serviço específico de comunicação de dados em um nível de processo-a-processo (por exemplo: como um web browser deve se comunicar com um servidor da web).
3 - Transporte (4 ^a camada OSI)	TCP, UDP, SCTP Essa parte controla a comunicação host-a-host. protocolos como OSPF, que é executado sobre IP, podem também ser considerados parte da

	camada de rede.
2 - Internet (3ª camada OSI)	Para TCP/IP o protocolo é IP, MPLS Essa parte é responsável pelas conexões entre as redes locais, estabelecendo assim a interconexão. Protocolos requeridos como ICMP e IGMP é executado sobre IP, mas podem ainda ser considerados parte da camada de rede; ARP não roda sobre IP.
1 - Enlace (Interface com Rede) (1ª e 2ª camada OSI)	Essa é a parte responsável por enviar o datagrama recebido pela camada de "Internet" em forma de um quadro através da rede. Tecnologias usadas para as conexões: <i>Ethernet</i> rede com fio e Wi-Fi rede sem fio. No modelo OSI, essa camada também é física, porém, é dividido em duas partes: física e enlace de dados. A física é a parte do <i>hardware</i> (os cabos das redes com fio) e o enlace de dados é a parte lógica do <i>hardware</i> : endereço MAC de origem e destino; controle de enlace lógico; controle de acesso ao meio.

Tabela 2.1 - Associação entre o Modelo OSI e o protocolo TCP/IP

O TCP/IP, um protocolo de comunicação, oferece uma ampla gama de benefícios cruciais em diversas áreas, proporcionando uma base sólida para a conectividade moderna. No âmbito da padronização, o TCP/IP estabelece um padrão roteável, sendo reconhecido como o protocolo mais completo e amplamente adotado na contemporaneidade. Todos os sistemas operacionais modernos oferecem suporte ao TCP/IP, tornando-o a base predominante para o tráfego na maioria das grandes redes.

Quanto à interconectividade, o TCP/IP desempenha um papel fundamental ao facilitar a conexão entre sistemas heterogêneos. Utilitários padrões como FTP (*File Transfer Protocol*) e Telnet são disponibilizados, simplificando o acesso e a transferência de dados entre sistemas diversos.

No que se refere ao roteamento, o TCP/IP possibilita a conexão de tecnologias antigas e novas à Internet. Além disso, opera com protocolos de linha, como P2P (Point to Point Protocol), para facilitar conexões remotas por meio de linhas discadas ou dedicadas. A integração a mecanismos IPCs e interfaces amplamente utilizados por sistemas operacionais, como *sockets* do Windows e NetBIOS, contribui para a versatilidade do protocolo.

A robustez do TCP/IP é evidenciada por sua escalabilidade e adaptabilidade multiplataforma. Adequado para sistemas operacionais cliente/servidor, o TCP/IP viabiliza o uso de aplicações de grande porte em diferentes pontos, demonstrando sua eficácia em ambientes diversificados.

No contexto da Internet, o TCP/IP é essencial para a conectividade. Redes locais distribuem servidores de acesso à Internet (*proxy servers*), e os hosts locais se conectam a esses servidores para obter acesso à Internet. A configuração adequada do TCP/IP é fundamental para alcançar essa conectividade, consolidando ainda mais sua importância nas operações cotidianas.

2.2.1 - Funcionamento do protocolo TCP/IP

O protocolo TCP/IP opera em quatro camadas distintas, conforme é apresentado na Figura 2.2.1. Inicia-se com a comunicação do programa na camada de aplicação, que abriga protocolos específicos, como SMTP (para e-mails), FTP (para transferência de arquivos) e HTTP (para navegação na internet). Cada programa executa um protocolo correspondente à camada de aplicação, dependendo de sua finalidade [20].

Após o processamento das requisições, o protocolo da camada de aplicação se comunica com o TCP, localizado na camada de transporte. Esta camada é responsável por dividir os dados enviados pela camada de aplicação e transmiti-los para a camada de internet. Durante o recebimento dos dados, a camada de transporte se encarrega de organizar os pacotes recebidos em ordem, pois estes podem ser recebidos fora de sequência, além de verificar a integridade do conteúdo dos pacotes.

Na camada de internet, onde o protocolo IP (*Internet Protocol*) está implementado, os pacotes recebem informações adicionais de endereços virtuais. São acrescentados os endereços dos computadores que estão enviando e recebendo dados, conhecidos como endereços IP. Em seguida, o pacote é encaminhado para a camada mais baixa, a Interface de Rede, e quando os dados alcançam essa camada, são chamados de datagramas.

A Interface de Rede recebe os pacotes enviados pela camada de internet e os transmite pela rede. Atualmente, a arquitetura mais utilizada para comunicação entre computadores em redes locais é a *Ethernet*, que opera em diferentes faixas de velocidade e pode ser implementada por meio de cabos (como cabo de par trançado CAT5 ou CAT6) ou de forma wireless (WI-FI). Os pacotes transmitidos pela rede são chamados de quadros.

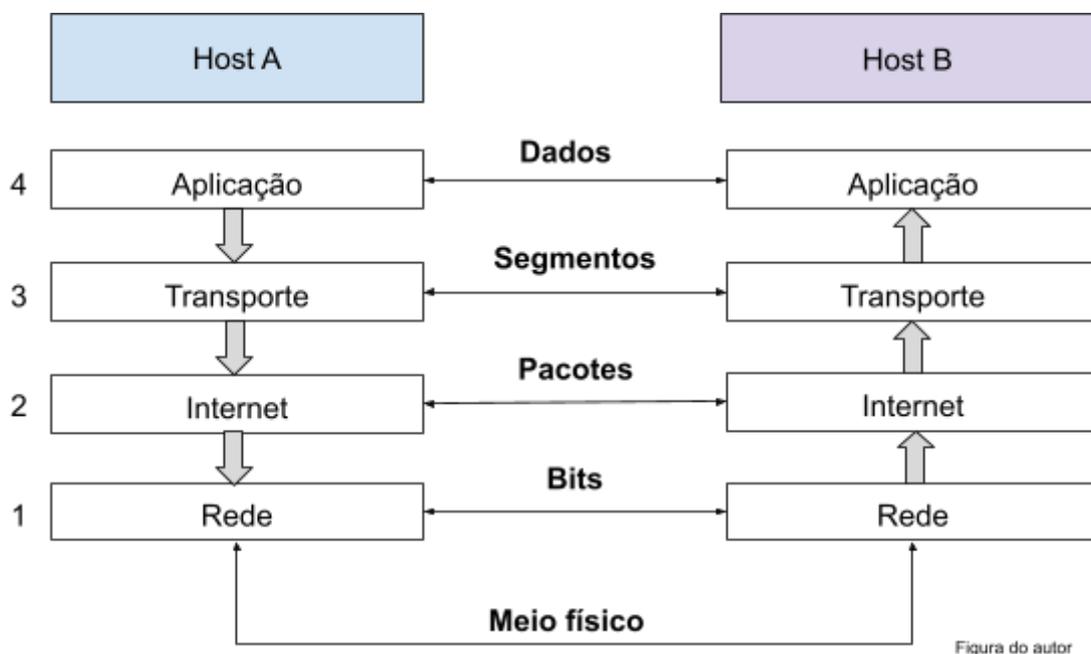


Figura 2.2.1 - Representação do modelo TCP/IP através de camadas

3. Banco de Dados

Um banco de dados consiste em um conjunto de informações organizadas e estruturadas de acordo com padrões definidos pelo desenvolvedor. Para utilizar o banco de dados de maneira eficiente e segura, os usuários precisam familiarizar-se com os padrões e características da sua estrutura, permitindo consultas, atualizações e armazenamento seguro de novas informações. Na era digital, o termo 'banco de dados' está intrinsecamente relacionado às informações armazenadas eletronicamente em servidores dedicados ou storages [43]. nesta pesquisa, o banco de dados será utilizado para armazenar os comandos de configuração, dados adquiridos pelos sistemas de medidas e status dos sistemas.

Existem diferentes tipos de bancos de dados, sendo os dois principais:

1. Bancos de dados relacionais (RDBMS - *Relational Database Management System*): Esses bancos de dados organizam as informações em tabelas, onde os dados são estruturados em linhas e colunas. O SQL (*Structured Query Language*) é uma linguagem comumente usada para consultar e manipular bancos de dados relacionais.
2. Bancos de dados não relacionais (*NoSQL ou Not Only SQL*): Esse tipo de banco de dados abrange várias tecnologias que não seguem o modelo relacional. Eles são mais flexíveis e adequados para casos de uso específicos, como armazenamento e processamento de grandes volumes de dados não estruturados.

Os bancos de dados relacionais e não relacionais representam dois tipos principais de sistemas de armazenamento de dados. Enquanto os bancos de dados relacionais organizam dados em tabelas com linhas e colunas, facilitando a consulta e manipulação através de SQL, os bancos de dados não relacionais adotam modelos diversos, como chave-valor, documento, família de colunas e gráfico. Essa abordagem oferece maior flexibilidade no armazenamento e gerenciamento de dados, especialmente para conjuntos extensos e não estruturados. Exemplos conhecidos incluem Oracle e MySQL para os relacionais, e MongoDB e Cassandra para os não relacionais.

Os bancos de dados relacionais aderem às propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), garantindo a conclusão bem-sucedida de um conjunto de operações e assegurando a precisão dos dados. Por outro lado, os bancos de dados não relacionais seguem as propriedades BASE (*Basically Available, Soft state, Eventually consistent*), priorizando a disponibilidade sobre a consistência imediata, embora alguns possam oferecer conformidade ACID com compensações de desempenho.

Quanto ao desempenho, os bancos de dados relacionais dependem do subsistema de disco, podendo ser otimizados com a utilização de SSDs (*Solid State Drives*), RAID (*Redundant Array of Independent Disks*) e otimização de índices e consultas. Já os bancos de dados NoSQL têm seu desempenho influenciado pela latência de rede, o tamanho do cluster de *hardware* e a aplicação que os utilizam. Ambos podem ser aprimorados conforme suas características específicas.

Os desafios de escala nos bancos de dados relacionais surgem devido a seus esquemas rígidos. Escala vertical ou horizontalmente requer estratégias especiais, como particionamento e shard. Em contrapartida, os bancos de dados NoSQL são altamente escaláveis, distribuindo a carga de trabalho para diversos nós e fornecendo dados em conjuntos menores.

Bancos de dados relacionais garantem integridade e consistência, porém, com flexibilidade limitada. Os não relacionais são mais flexíveis, mas oferecem consistência eventual. O desempenho dos bancos de dados relacionais está ligado ao subsistema de disco, enquanto nos NoSQL, depende da latência de rede e do tamanho do cluster de *hardware*. A escalabilidade é um desafio para os relacionais, enquanto os não relacionais se destacam nesse aspecto.

Os bancos de dados desempenham um papel crucial em aplicativos e sistemas modernos, abrangendo desde websites e aplicativos móveis até sistemas empresariais complexos, assegurando eficiência, segurança e escalabilidade na manipulação de informações. No âmbito deste projeto, o banco de dados é essencial para garantir o funcionamento do sistema supervisorio. Todas as configurações, comandos de medição e armazenamento dos dados coletados são realizados por meio das estruturas de dados definidas no banco de dados Redis.

3.1 - Redis

O Redis - Remote Dictionary Server, é um banco de dados NoSQL de código aberto que opera com o modelo chave-valor [14]. Ele transcende as limitações dos bancos de dados NoSQL convencionais, oferecendo uma variedade de recursos avançados essenciais para atender às demandas de aplicativos modernos [34][35].

O termo "NoSQL" refere-se a um conjunto diversificado de tecnologias desenvolvidas para armazenar dados. NoSQL engloba uma série robusta de tecnologias que possibilitam a persistência de dados com o alto desempenho exigido por aplicativos contemporâneos em escala na Internet.

Vale ressaltar que "SQL" é uma sigla para "*Standard Query Language*", representando uma linguagem padrão para manipulação de dados em bancos de dados relacionais. Em contrapartida, o Redis faz parte da categoria NoSQL, destacando-se por sua abordagem baseada em chave-valor.

Características do bancos de dados Redis

Nesta seção, abordaremos algumas características do banco de dados Redis e suas respectivas aplicações. Algumas delas serão mencionadas de forma breve; para uma compreensão mais aprofundada, recomendamos aos leitores que consultem a documentação oficial do Redis [34].

Chave/valor

Há quatro tipos principais de bancos de dados NoSQL, cada um com um propósito específico para o qual é mais adequado: chave/valor, coluna, documento e gráfico. Um exemplo proeminente é o Redis, que se classifica como um banco de dados NoSQL estruturado na forma de chave/valor.

No modelo chave/valor, os dados são acessados e identificados por chaves, funcionando de maneira similar às chaves primárias em bancos de dados relacionais. Cada peça de informação é associada a um valor correspondente à sua chave.

Um exemplo prático de um par chave/valor seria:

"key": 280304

Aqui, "key" representa a chave, enquanto 280304 é o valor associado a essa chave. A característica da chave, isto é, string, inteiro, float, hash e hyperlog é definida automaticamente ao incluir o tipo de valor desejado.

O Redis é *case sensitive* em relação às chaves. Isso significa que as chaves são diferenciadas entre maiúsculas e minúsculas. Por exemplo, a chave "Key" não é a mesma que "key" ou "KEY". Se houver a necessidade de realizar consultas de chaves sem considerar a diferença entre maiúsculas e minúsculas, a solução é converter todas as chaves para um formato consistente (por exemplo, todas em letras minúsculas ou todas em letras maiúsculas). O Redis não oferece uma configuração nativa para tornar as chaves insensíveis a maiúsculas e minúsculas. O Redis é *case sensitive* para chaves, mas é possível implementar uma lógica própria para torná-las insensíveis a maiúsculas e minúsculas, se necessário.

Uma abordagem possível é criar um padrão de expressão regular que corresponda a variações de maiúsculas e minúsculas para uma determinada chave. Por exemplo, é possível usar a seguinte função em Python para gerar um padrão de expressão regular que corresponde a diferentes combinações de maiúsculas e minúsculas:

```
# -----  
  
def toIgnoreCasePattern(s):  
    pattern = "  
    for c in s:  
        if c.isalpha():  
            pattern += f"[{c.upper()} {c.lower()}]"  
        else:  
            pattern += c  
    return pattern  
  
# Exemplo de uso:  
chave = "Key"  
padrao = toIgnoreCasePattern(chave)  
print(padrao) # Saída: "[Mm][Uu][Ss]:[li][Cc]"  
# -----
```

Essa abordagem é específica para a linguagem de programação Python 3 e deve ser aplicada ao criar ou consultar chaves no Redis.

O Redis se destaca não apenas como um banco de dados NoSQL, mas também como um banco de dados multi modelo, oferecendo suporte a uma ampla gama de funcionalidades. Isso inclui recursos como pesquisas, mensagens, streaming, gráficos, entre outros, indo além do armazenamento de dados convencional.

Um banco de dados multimodelo é um sistema de gerenciamento que combina vários tipos de bancos de dados em um único backend. A maioria dos sistemas de gerenciamento de bancos

de dados suporta apenas um modelo de banco de dados. Por outro lado, os bancos de dados multimodelo armazenam, consultam e indexam dados de diferentes modelos.

Os bancos de dados multimodelo oferecem as vantagens da persistência poliglota sem precisar combinar diferentes modelos. Essa abordagem flexível permite armazenar dados de diferentes maneiras, resultando em programação ágil e flexível e redução da redundância de dados.

Um recurso vital do banco de dados multimodelo é a capacidade de transformar dados de um formato para outro. Por exemplo, dados no formato JSON podem ser rapidamente transformados em XML. A conversão de formatos de dados fornece agilidade adicional e facilita o cumprimento de requisitos específicos do projeto. Os bancos de dados multimodelo são úteis em vários casos de uso. Por exemplo, facilitam o armazenamento e gerenciamento de várias fontes de dados, estendem os recursos do modelo e fornecem ambientes de dados híbridos.

Em suma, um banco de dados multimodelo é uma plataforma de dados integrada única que pode armazenar, acessar e processar diferentes tipos de dados para realizar várias tarefas. Com um banco de dados multimodelo, você pode unificar vários tipos de dados e modelos em uma única solução, sem ter tecnologias individuais para cada propósito específico.

Os bancos de dados multimodelo proporcionam uma abordagem que permite interagir com dados independentemente do modelo de dados subjacente. O Redis, através dos Módulos Redis, oferece funcionalidade multimodelo completa. A escolha do Redis como banco de dados multimodelo proporciona maior flexibilidade aos desenvolvedores de aplicativos dentro de uma organização [39].

Armazenamento de dados

O Redis mantém os dados na memória para acesso rápido e assegura o armazenamento persistente dos dados, bem como a replicação de conteúdo na memória para cenários de alta disponibilidade em produção. Quando se discute o armazenamento de dados, o conceito de durabilidade torna-se crucial. Durabilidade refere-se à capacidade de garantir que os dados estejam disponíveis em caso de falha de algum componente do banco de dados.

O Redis oferece suporte a vários modos para assegurar a durabilidade, adaptando-se à maioria das estruturas de dados e aos requisitos específicos do ambiente, Tabela 3.1.

hello world	string
000100000111111000111000	Bitmap
{9846}{7771777798}{030739767}	Bitfield
{a: "hello", b: "world"}	Hash
[A>B>C<C]	List

{A<B<C}	Set
{A:1, B:2, C:3}	Sorted Set
{A:(50.1,0,5)}	Geospatial
01101101 01101111 01101101	Hyperlog
{id1=time1.seq((a: "foo", a: "bar"))}	Stream

Tabela 3.1 - Tabela com as estruturas suportadas pelo Redis

O Redis oferece suporte a uma variedade de estruturas de dados. De fato, pode ser útil conceber o Redis como um sistema de armazenamento especializado em estruturas de dados, indo além de ser apenas um armazenamento NoSQL simples de chave/valor.

As estruturas de dados suportadas incluem:

- **Strings (Cadeias de Caracteres):**

Descrição: Simples sequências de caracteres.

Uso Típico no Redis: Armazenamento de valores simples, como um número inteiro, texto ou dados binários.

- **Bitmap:**

Descrição: Uma sequência de bits.

Uso Típico no Redis: Útil para representar estados de algo, como dias da semana em que um usuário visitou um site (presença/ausência).

- **Bitfield:**

Descrição: Permite manipular sequências de bits em uma cadeia de caracteres.

Uso Típico no Redis: Realizar operações específicas em bits de um número armazenado.

- **Hash (Tabela de Hash):**

Descrição: Um mapeamento de campos associados a valores.

Uso Típico no Redis: Armazenamento de dados estruturados, como informações de um usuário (nome, idade, e-mail).

- **List (Lista):**

Descrição: Uma coleção ordenada de elementos.

Uso Típico no Redis: Implementação de filas ou pilhas, mantendo a ordem de inserção.

- **Set (Conjunto):**

Descrição: Uma coleção de elementos únicos, sem ordem específica.

Uso Típico no Redis: Armazenamento de itens únicos, como tags associadas a um post.

- **Sorted Set (Conjunto Ordenado):**

Descrição: Similar a um conjunto, mas cada elemento tem uma pontuação associada, permitindo ordenação.

Uso Típico no Redis: Armazenamento de classificações, como pontuações de jogadores em um jogo.

- **Geospatial (Geoespacial):**

Descrição: Armazena dados geoespaciais, como coordenadas de latitude e longitude.

Uso Típico no Redis: Aplicações de mapeamento e localização.

- **Hyperlog (HyperLogLog):**

Descrição: Estrutura de dados probabilística para estimar a cardinalidade (número de elementos distintos).

Uso Típico no Redis: Contagem aproximada de elementos únicos, como contagem de visitantes únicos.

- **Stream (Fluxo):**

Descrição: Uma estrutura de log distribuído.

Uso Típico no Redis: Captura e armazena eventos em tempo real, como logs ou mensagens em um sistema distribuído.

Essas estruturas de dados fornecem uma ampla variedade de funcionalidades, permitindo que o Redis seja utilizado em diversos cenários e aplicativos.

O Redis, com suas estruturas de dados projetadas para atender a casos específicos, destaca-se como uma solução versátil para aplicativos modernos com demanda intensiva de dados. Além das estruturas de dados, o Redis suporta o padrão Publish/Subscribe (PubSub) e outros padrões, ampliando sua aplicabilidade em diversos cenários.

Operando predominantemente como um software back-end em ambientes Unix, como Linux e macOS, o Redis também pode ser executado como um contêiner Docker no Microsoft Windows, proporcionando flexibilidade de implementação. O servidor Redis aguarda constantemente conexões de clientes, seja programaticamente ou por meio da interface de linha de comando (CLI), facilitando a interação direta com os dados armazenados.

No âmbito dos clientes e desenvolvedores, o Redis oferece suporte a diversas linguagens de programação por meio de uma variedade de bibliotecas de clientes. Essas bibliotecas, como o Redis-py para Python, possibilitam que os desenvolvedores interajam programaticamente com os dados do servidor Redis. Vale destacar que as bibliotecas são geralmente compartilhadas sob licenças de código aberto, promovendo a colaboração e a flexibilidade na escolha da integração, embora as licenças possam variar entre os projetos específicos.

Memória e persistência

Não existe um processo formal estabelecido para a criação de um banco de dados no Redis, assim como não é obrigatório seguir um procedimento formal para a criação de tabelas. No Redis, a inclusão direta de dados no banco, sem a necessidade de definir previamente qual é o tipo da estrutura, é realizada por meio do comando **SET**.

Redis.SET ('CHAVE', 'VALOR')

Usuários acostumados com a formalidade na criação e definição de bancos de dados podem inicialmente experimentar um certo desconforto com o processo aparentemente menos formal de criação e manipulação de dados no Redis. No entanto, é precisamente por meio dessa flexibilidade que a verdadeira eficácia e poder do Redis se destacam. Aumentando a velocidade das manipulações das estruturas de dados, além de permitir que novas estruturas sejam inseridas instantaneamente em um processo já existente.

Os dados são armazenados na memória de acesso aleatório (RAM) do servidor Redis. Isso significa que, à medida que mais dados são adicionados, há um aumento correspondente no uso da RAM. O Redis em Flash surge como uma solução para estender a capacidade de armazenamento, integrando memória baseada em flash. Em intervalos configuráveis, que variam de acordo com a quantidade de dados modificados, o Redis realiza a persistência, gravando o conteúdo do banco de dados no disco. Esse processo garante uma combinação eficaz de alta velocidade de acesso proporcionada pela RAM e a durabilidade adicional conferida pela persistência em memória flash.

A persistência de dados no disco é crucial para garantir a durabilidade em situações de falhas de software ou *hardware* que possam tornar o servidor indisponível. Essa prática assegura que as informações essenciais não sejam perdidas, permitindo a recuperação após eventos adversos.

Além da persistência em disco, em ambientes de produção, são frequentemente empregados métodos adicionais para garantir durabilidade e disponibilidade contínua. O *clustering*, por exemplo, é uma abordagem comum que envolve a configuração de múltiplos nós do Redis em um cluster. Esse arranjo não apenas distribui a carga, melhorando o desempenho, mas também proporciona redundância, assegurando que, mesmo se um nó falhar, outros possam manter a operação contínua. Essas estratégias combinadas fortalecem a robustez e confiabilidade do sistema Redis em ambientes críticos.

3.2 - Aplicação do Redis

A seguir, serão apresentadas algumas características do Redis para diferentes contextos e aplicações Web. Vale ressaltar que as aplicações não estão limitadas aos tópicos apresentados.

- **Caching e Conjunto de dados**

O Redis destaca-se na gestão eficiente de armazenamento em cache devido aos seus tipos de dados nativos e ao uso otimizado da memória. O desempenho robusto do Redis permite a realização de recomendações e análises de clientes em tempo real. A adoção do Redis no

Flash torna a análise de conjuntos de dados volumosos uma solução econômica. Nesse cenário, o Redis Enterprise Flash é empregado para ampliar a capacidade de RAM.

- **Mensagens/fila**

O Redis desempenha um papel significativo na gestão eficiente de grandes volumes de dados em tempo real, especialmente quando lidamos com a geração massiva de dados provenientes de milhares ou até milhões de sensores. Sua eficiente implementação do mecanismo de publish/subscribe permite a fácil coleta, transmissão e ingestão desses dados, tornando-o uma escolha valiosa para cenários que demandam análise e processamento dinâmicos.

A importância dos modelos de dados reside na maneira como os dados são organizados em um banco de dados, e a escolha de um modelo pode influenciar a forma como seu aplicativo interage com esses dados. Enquanto modelos relacionais historicamente não se alinhavam perfeitamente com certas aplicações ou domínios, tecnologias NoSQL, como o Redis, proporcionam uma flexibilidade significativa. No contexto do Redis, que é um banco de dados multi modelo, os dados podem ser representados de maneiras diversas, sendo capazes de atender a uma variedade de casos de uso simultaneamente. Essa característica permite que os dados sejam utilizados de maneira mais adequada para cada aplicação específica, adaptando-se dinamicamente às necessidades variáveis do sistema.

3.3 - Entendendo os Modelos de Dados do Redis

Utilizar qualquer tipo de armazenamento de dados requer tomar decisões sobre a representação dos dados dentro desse ambiente. Esse modelo determina como os dados são inseridos no banco de dados e como são recuperados.

No Redis, os dados são armazenados utilizando chaves. As chaves podem ser praticamente qualquer coisa, pois são seguras binariamente. Por exemplo, você pode utilizar uma imagem como chave, embora a maioria das chaves seja composta por strings simples.

O Redis oferece uma variedade de comandos para manipular dados de diferentes tipos. Alguns comandos notáveis podem ser encontrados nesta seção, incluindo SET e GET. O comando SET cria ou modifica um valor associado a uma chave específica, enquanto o comando GET recupera o valor associado à chave fornecida.

É importante observar que os valores são sobrescritos pelo comando SET. Isso significa que, se você chamar SET duas vezes para a mesma chave, o último valor será aquele que fica armazenado e é recuperado.

Os valores associados a uma chave podem ser formatados de diversas maneiras, permitindo a criação de modelos de dados específicos para atender às necessidades específicas da organização.

Strings e bitmaps

O tipo de valor mais simples no Redis é uma string. Um valor pode ser adicionado ao banco de dados utilizando o comando SET. Ao utilizar o comando SET, apenas uma chave e um valor são requisitos mínimos para criar a entrada. Por exemplo, para criar uma chave chamada "user" com um valor "usuario", basta executar este comando na CLI do Redis:

```
> SET user usuario
```

Embora as aspas duplas tenham sido utilizadas para este valor de *string*, elas não são estritamente necessárias quando o valor consiste em apenas uma palavra. Assim, um valor simples de *string*, como "usuario", foi armazenado no banco de dados e pode ser recuperado utilizando o comando GET.

> GET user

O comando acima retorna o seguinte valor:

“ usuario ”

Existem inúmeros outros comandos para manipular e trabalhar com dados do tipo *string* e similares no Redis. No entanto, é importante observar que comandos destinados a dados numéricos não podem ser utilizados em dados do tipo *string*.

Diretamente associados às *strings* estão os bitmaps, que representam uma forma particular de armazenamento de *strings*. Ao usar um bitmap, você pode representar muitos elementos de dados que estão simplesmente ligados (1) ou desligados (0).

Essa abordagem é útil em operações em que apenas dois valores possíveis são necessários, como determinar se um usuário está ativo ou inativo. Devido à simplicidade desses dois valores, você pode representar eficientemente esses dados.

O Redis oferece comandos específicos para trabalhar com bitmaps, incluindo SETBIT e GETBIT, utilizados para criar ou modificar um valor e recuperar um valor, respectivamente. Outros comandos incluem BITOP e BITFIELD.

Padrões e Estruturas de Dados

- **Pub/sub**

Neste exemplo, o canal "clima" é criado ou, se já existir, a mensagem "temp:25c" é enviada para todos os clientes que estão inscritos nesse canal.

PUBLISH clima "temp:25c"

Para se inscrever em um canal e receber mensagens, os clientes usam o comando **SUBSCRIBE**, como no exemplo abaixo:

SUBSCRIBE clima

Dessa forma, o cliente se torna um assinante do canal "clima" e receberá todas as mensagens publicadas nesse canal. O Redis, sendo extremamente eficiente em manipulação de strings e comunicação, torna esse modelo pub/sub uma opção valiosa para sistemas distribuídos e eventos em tempo real.

Essa capacidade do Redis de atuar como um intermediário eficiente em sistemas de troca de mensagens contribui para sua versatilidade e amplia ainda mais sua utilidade em arquiteturas de software distribuídas e orientadas a eventos.

A mensagem é publicada no canal chamado clima, independentemente de haver clientes inscritos. Se houver um cliente inscrito, o cliente receberá uma mensagem como a seguinte:

1) "message"

2) "clima"

3) "temp:25c"

Os clientes se inscrevem em um canal com o comando **SUBSCRIBE**.

Supõe-se que o cliente conheça o formato das mensagens e seja capaz de analisar as mensagens recebidas corretamente. Como outros tipos de dados no Redis, os canais do editor pub/sub podem ser divididos para criar uma estrutura hierárquica por convenção. Por exemplo, a criação de um canal meteorológico por código postal pode ter esta aparência:

PUBLISH clima:280393temp:25c

Os clientes podem então assinar o código postal específico para atualizações meteorológicas. Os clientes também podem se inscrever em um padrão curinga para todas as subchaves de clima, com o comando **PSUBSCRIBE**

PSUBSCRIBE clima:*

Módulos

Os módulos específicos do Redis estendem as funcionalidades básicas do banco de dados, adicionando recursos especializados. O Redis Graph, por exemplo, é um módulo que possibilita o armazenamento e consulta de dados em formato de grafo. Essa funcionalidade é particularmente útil para aplicativos que envolvem relações complexas entre diferentes entidades, como redes sociais, sistemas de recomendação, ou em qualquer domínio onde as relações entre os dados são cruciais.

Outro módulo notável é o Redisearch, que oferece recursos de indexação e pesquisa em texto completo no Redis. Ele se destaca em cenários em que é necessário realizar consultas complexas em grandes conjuntos de dados textuais, como em mecanismos de busca.

O Redis TimeSeries é mais um desses módulos especializados, permitindo o armazenamento e consulta de séries temporais de dados. Essa funcionalidade é ideal para lidar com dados que variam ao longo do tempo, como informações de sensores, métricas de desempenho ou qualquer conjunto de dados temporal.

Esses módulos não apenas ampliam as capacidades do Redis, mas também o tornam mais flexível e adaptável a uma variedade mais ampla de casos de uso. Embora sejam instalados separadamente, uma vez integrados ao Redis, podem ser usados de maneira transparente com os comandos padrão do banco de dados. Essa modularidade reforça a versatilidade do Redis, permitindo que os usuários escolham e implementem funcionalidades específicas que atendem melhor às necessidades de seus projetos.

3.4 - Arquitetura e Topologia

A arquitetura e topologia do Redis são cuidadosamente delineadas para garantir não apenas desempenho veloz, mas também uma disponibilidade excepcional. Essa arquitetura tem como pilares elementos essenciais, começando pelo modelo chave/valor, onde dados são armazenados em pares, associando uma chave única a um valor que pode ser de diversos tipos, como *strings*, *hashes*, listas, conjuntos e *sorted sets*.

O Redis opera como um banco de dados em memória, mantendo os dados na RAM para acesso instantâneo, proporcionando, assim, velocidade elevada tanto para leitura quanto para gravação. Embora seja um banco de dados em memória, o Redis oferece opções de persistência, permitindo que dados sejam periodicamente salvos em disco para recuperação após reinicializações.

A flexibilidade do Redis se estende à sua configuração, podendo ser tanto um servidor único quanto um cluster para garantir alta disponibilidade. No modo servidor único, todos os dados residem em uma instância, enquanto no modo cluster, os dados são particionados e distribuídos entre diversas instâncias.

A capacidade de replicação é outra característica importante do Redis, permitindo que servidores secundários mantenham cópias dos dados do servidor principal, proporcionando escalabilidade e alta disponibilidade. Em ambientes de cluster, o Redis adota o particionamento (*sharding*) para distribuir dados entre várias instâncias, cada uma sendo responsável por uma parte específica do conjunto de dados.

Além disso, o Redis oferece um mecanismo eficaz de Publicação/Assinatura (Pub/Sub) para possibilitar comunicação assíncrona entre diferentes partes de uma aplicação. A execução atômica de muitos comandos assegura consistência nas operações realizadas.

Sua versatilidade é evidenciada pelos diferentes modos de operação, incluindo standalone, sentinel (alta disponibilidade) e cluster. Oferece conectividade através de diversos clientes e suporta várias linguagens de programação. Ferramentas de monitoramento e administração estão disponíveis para facilitar o gerenciamento de instâncias Redis.

A escalabilidade é outra faceta destacada do Redis, permitindo escalamento tanto vertical (acréscimo de recursos em um servidor único) quanto horizontal (divisão de dados entre vários servidores, *sharding*). Quando configurado como cluster, distribui dados entre nós, sendo cada nó capaz de conter réplicas para garantir resiliência.

Essa arquitetura flexível do Redis pode ser configurada de acordo com requisitos específicos, oferecendo tanto o modo standalone quanto o modo cluster para atender às diversas demandas de desempenho, escalabilidade e alta disponibilidade de uma aplicação.

3.5 - Clustering e Disponibilidade

Clustering refere-se à prática de combinar múltiplos recursos computacionais, como servidores, para trabalhar como uma única entidade coesa. Essa abordagem tem vários objetivos, incluindo distribuição eficiente de carga, aumento da capacidade de processamento e redundância para garantir a continuidade operacional.

Em um contexto mais específico, o clustering de servidores visa melhorar o desempenho e a confiabilidade dos sistemas. Servidores agrupados podem compartilhar tarefas, equilibrar a carga de trabalho e, em alguns casos, oferecer tolerância a falhas. Existem dois tipos principais de clustering:

Clustering de Balanceamento de Carga distribui as solicitações entre os servidores disponíveis para otimizar o uso dos recursos e melhorar a eficiência, evitando a sobrecarga de um único servidor. Clustering de tolerância a falhas garante a continuidade operacional, mesmo se um dos servidores falhar, transferindo automaticamente os serviços para um servidor funcional.

Alta disponibilidade (HA) refere-se à capacidade de um sistema ou serviço estar operacional e acessível durante a maior parte do tempo possível, buscando minimizar o tempo de inatividade. Elementos-chave incluem redundância, detecção de falhas, recuperação rápida e escalabilidade para lidar com aumentos de carga.

Em muitos casos, clustering e alta disponibilidade estão interligados para criar ambientes robustos e confiáveis. Um cluster pode proporcionar balanceamento de carga e tolerância a falhas, enquanto as práticas de alta disponibilidade garantem a continuidade operacional e a rápida recuperação de qualquer interrupção.

Exemplos práticos incluem bancos de dados em cluster, onde vários servidores trabalham juntos para distribuir consultas, e servidores web em cluster, onde o balanceamento de carga distribui o tráfego entre vários servidores para garantir a disponibilidade contínua do serviço. Sistemas empresariais críticos, como ERPs, frequentemente utilizam clustering e alta disponibilidade para garantir operações contínuas.

Em resumo, a implementação eficaz de clustering e alta disponibilidade é essencial para garantir a confiabilidade e o desempenho de sistemas computacionais em ambientes críticos. Essas práticas não apenas melhoram a eficiência operacional, mas também proporcionam resistência a falhas e a capacidade de escalar conforme necessário.

Um ambiente de produção geralmente demanda um nível específico de desempenho e redundância. O desempenho do banco de dados é garantido por meio de diversas estratégias, incluindo clustering e sharding.

Um fragmento de banco de dados representa uma parte de um banco de dados maior. As porções de um conjunto de dados são distribuídas entre vários servidores, sendo cada servidor responsável por um subconjunto específico de dados. Essa abordagem distribui a carga de trabalho entre os servidores.

3.6 - Console

Neste projeto, o servidor hospedeiro do banco de dados Redis está instalado na nuvem da máquina virtual do Redis Labs. O Redis Labs é uma empresa que oferece serviços e soluções relacionadas ao banco de dados Redis, um banco de dados em memória open-source que é amplamente utilizado para armazenamento em cache, processamento de dados em tempo real e outras aplicações onde o acesso rápido aos dados é essencial.

A Redis Labs fornece uma plataforma gerenciada chamada Redis Enterprise, que é uma versão comercial do Redis. Essa plataforma é projetada para atender a requisitos específicos de desempenho, escalabilidade e segurança em ambientes empresariais.

O Redis Labs oferece diversos recursos e serviços, incluindo o Redis Enterprise Cloud, que é uma solução de banco de dados em nuvem gerenciada para o Redis. Eles fornecem suporte técnico para organizações que utilizam o Redis Enterprise, além de serviços de treinamento e consultoria para ajudar na implementação e otimização do Redis. Para garantir a segurança e conformidade dos dados armazenados, oferecem ferramentas especializadas. O Redis Enterprise Software pode ser implantado tanto em ambientes locais quanto em nuvens privadas, e o Redis Insight é uma ferramenta dedicada ao monitoramento e visualização de bancos de dados Redis.

Para estabelecer comunicação com o Redis e armazenar as informações na máquina virtual, RedisLabs, é necessário uma autenticação com o login do endpoint configurado no servidor da máquina virtual. No exemplo a seguir temos o endereço do Redis - `host_Redis`, a porta - `port_Redis` e a senha de acesso `password_Redis`.

Redis = Redis.Redis (host_Redis = IP, port_Redis = porta de acesso, password_Redis = senha)

Bloco de comunicação com o Redis

host_Redis="Redis-15032.c8.us-east-1-2.ec2.cloud.Redislabs.com" # STRING

port_Redis = 15032 # INTEIRO

password_Redis = "senha" # STRING

3.7 - Alternativas

Recentemente, o Redis mudou sua licença de distribuição, seguindo uma tendência observada em outras empresas de *software open source*, como MongoDB e Elastic. Desde então, o Redis e seus módulos principais são distribuídos sob uma licença dupla: a *Redis Source Available License v2 (RSALv2)* e a *Server Side Public License v1 (SSPLv1)*. Esta mudança visa proteger o investimento da empresa em código aberto, especialmente contra o uso não remunerado por provedores de serviços em nuvem Redis.

O RSALv2 permite o uso, cópia, distribuição e modificação do software, mas com duas principais restrições: não pode ser comercializado como um serviço gerenciado, e as notificações de licenciamento não podem ser removidas. O SSPLv1, por outro lado, exige que quaisquer modificações feitas para oferecer o software como serviço também sejam liberadas sob a mesma licença Redis.

Com a mudança de licença do Redis, muitos desenvolvedores e empresas estão procurando alternativas para evitar possíveis restrições comerciais. A seguir, algumas das principais alternativas ao Redis:

Valkey

Valkey é um projeto extremamente promissor, essencialmente uma continuação do Redis, mas mantendo as licenças no modelo de código aberto. O nome deriva da estrutura de banco de dados chave-valor (key-value = Valkey). Empresas e indivíduos que anteriormente apoiavam

o Redis, como AWS, Google e Oracle, também contribuem para este projeto. *Valkey* foi lançado pela *Linux Foundation*, destacando seu forte compromisso com a comunidade de código aberto.

Garnet

Garnet é um projeto de pesquisa da Microsoft Research, focado em armazenamento de cache remoto. Projetado para oferecer alto desempenho, extensibilidade e baixa latência, Garnet é escalonável por thread em um único nó e suporta clusters fragmentados com replicação, checkpoints, failover e transações. Ele pode operar tanto na memória principal quanto em armazenamento em camadas (como SSD e Azure Storage) e é compatível com o protocolo RESP do Redis.

KeyDB

KeyDB é outra alternativa open source ao Redis, projetada para lidar com cargas de trabalho intensas, com benchmarks de nó único superando 1 milhão de operações por segundo. KeyDB é multithread e promete desempenho superior ao Redis por nó, além de trazer melhorias como replicação ativa e expiração de sub-chaves.

DragonFly

DragonFly é uma alternativa ao Redis que foca em oferecer alta performance e compatibilidade com as APIs do Redis. Projetado para escalabilidade, DragonFly pode lidar com mais de 1 TB de dados e 4 milhões de QPS em um único nó, simplificando a gestão de ambientes de grandes volumes de dados.

Existem várias alternativas open source ao Redis, cada uma com suas próprias vantagens e melhorias. Projetos como Valkey, Garnet, KeyDB e DragonFly não apenas oferecem funcionalidades semelhantes, mas também introduzem novos recursos e melhorias de desempenho. Testar essas opções pode ajudar a determinar a melhor ferramenta para atender às suas necessidades específicas.

Cada uma dessas alternativas tem suas próprias vantagens e desvantagens, dependendo do caso de uso específico e das necessidades de escalabilidade, consistência e performance. Avaliar esses fatores pode ajudar a escolher a melhor solução para substituir o Redis em seu projeto.

4. Ambiente Operacional

Nesta seção, serão apresentados em detalhes os componentes que compõem o ambiente operacional do sistema de monitoramento desenvolvido nesta pesquisa. Os ambientes e ferramentas selecionados para essa implementação foram escolhidos com base na experiência prévia e no nível de familiaridade do autor, garantindo assim uma integração eficiente e adequada às necessidades do projeto. No entanto, é importante enfatizar que a estrutura operacional aqui descrita não é rígida ou definitiva. O leitor é encorajado a explorar e adaptar esses componentes conforme necessário, podendo substituir ou acrescentar elementos que melhor atendam às suas necessidades específicas ou ao contexto particular em que o sistema será aplicado. Essa flexibilidade permite que a estrutura seja ajustada a diferentes cenários e requisitos, assegurando a relevância e a aplicabilidade do sistema em uma variedade de situações.

4.1 Máquina Virtual

Uma máquina virtual - VMs é um ambiente computacional que opera como uma máquina independente dentro de um sistema hospedeiro. Em vez de depender de *hardware* físico dedicado, as VMs utilizam recursos de um servidor físico compartilhado para criar ambientes isolados. Uma Máquina Virtual (VM) é composta por vários componentes principais, dentre estes podemos salientar:

Sistema Operacional: Similar a um computador físico, um sistema operacional convidado é instalado em cada máquina virtual. Cada máquina virtual possui seu próprio sistema operacional, podendo ser diferente do sistema operacional do host.

VMware Tools: Esse conjunto de utilitários aprimora o desempenho do sistema operacional convidado da máquina virtual e facilita sua gestão.

Configuração de Compatibilidade: Através do vSphere Client, cada máquina virtual é associada a uma versão compatível de host, cluster ou datacenter ESXi por meio da aplicação de uma configuração de compatibilidade.

Dispositivos de Hardware: Cada dispositivo de *hardware* virtual desempenha a mesma função para a máquina virtual que o *hardware* em um computador físico. Cada máquina virtual possui recursos de CPU, memória e disco.

Hipervisor: Esse software separa os recursos utilizados pela máquina virtual do *hardware* e os provisiona adequadamente. O hipervisor trata os recursos de computação, como CPU, memória e armazenamento, como um pool que pode ser facilmente realocado entre os guests existentes ou para novas máquinas virtuais. Ele é responsável por criar e gerenciar máquinas virtuais, alocando recursos do *hardware* físico para cada VM e garantindo que operem de forma isolada.

Esses componentes trabalham em conjunto para criar um ambiente virtual que se assemelha a um sistema de computação com sua própria CPU, memória, interface de rede e armazenamento. Esse sistema virtual é construído a partir de um sistema de *hardware* físico.

A virtualização por meio de Máquinas Virtuais (VMs) oferece uma série de benefícios essenciais, incluindo Isolamento, Consolidação de Servidores, Recuperação de Desastres, Testes e Desenvolvimento, Flexibilidade e Portabilidade.

As VMs funcionam de maneira independente umas das outras e em relação ao host, garantindo que as operações em uma VM não afetem as demais. Este isolamento é fundamental para melhorar a segurança e a estabilidade do ambiente, pois a segregação evita que problemas em uma VM afetem as outras.

A prática de consolidação de servidores é eficientemente viabilizada pela virtualização. Vários sistemas operacionais e aplicativos podem ser executados simultaneamente em um único servidor físico, otimizando eficazmente os recursos. A virtualização permite a consolidação de várias VMs em um único servidor, maximizando a utilização de recursos e diminuindo os custos operacionais.

No que diz respeito à recuperação de desastres, a virtualização proporciona eficiência ao permitir a cópia e transferência rápidas das VMs entre servidores. Isso agiliza o processo de recuperação, reduzindo significativamente o tempo de inatividade e minimizando os impactos nos negócios.

Os ambientes virtuais são ideais para testes de software, desenvolvimento e experimentação, sem comprometer o ambiente de produção. Esta separação permite que mudanças e ajustes sejam testados sem riscos desnecessários.

Em termos de flexibilidade, as VMs oferecem um gerenciamento eficiente de recursos, podendo ser facilmente escalonadas, movidas e copiadas conforme necessário. A agilidade nos processos de criação, configuração e gerenciamento de VMs permite uma rápida adaptação às demandas, podendo ser dimensionadas tanto vertical quanto horizontalmente.

A portabilidade é outra vantagem notável, permitindo a fácil movimentação de VMs entre servidores físicos. Esta flexibilidade facilita as operações de manutenção e gestão do ambiente, possibilitando a migração de cargas de trabalho sem interrupções.

4.1.1 - Tipos de Hypervisors

Existem dois tipos principais de hypervisors:

- **Hypervisors Tipo 1 (Bare Metal):** Instalados diretamente no hardware, esses hypervisors não requerem um sistema operacional hospedeiro adicional. Exemplos incluem VMware ESXi, Microsoft Hyper-V Server, e KVM.
- **Hypervisors Tipo 2 (Hospedados):** São instalados sobre um sistema operacional hospedeiro existente. Exemplos incluem VMware Workstation, Oracle VirtualBox e Microsoft Hyper-V (quando instalado em um sistema operacional hospedeiro).

VMware ESXi: É um hipervisor gratuito oferecido pela VMware. Ele permite a implantação de vários servidores em uma grande máquina através da virtualização. O VMware ESXi é um hipervisor desenvolvido pela VMware que trabalha na virtualização de computadores de nível empresarial.

VMware ESX: É um software comercial da VMware. Ele é semelhante ao ESXi, mas é uma versão comercial.

Xen: É um hipervisor open source que permite a criação, execução e gerenciamento simultâneos de várias máquinas virtuais em um computador físico.

Citrix XenServer: É um hipervisor gratuito oferecido pela Citrix. Ele é uma plataforma de virtualização de servidor que oferece desempenho de virtualização para sistemas operacionais do servidor e do cliente virtualizados.

Microsoft Hyper-V Server: É um hipervisor gratuito oferecido pela Microsoft. Ele fornece virtualização de nível empresarial para seu datacenter e nuvem híbrida.

Nutanix AHV: É um hipervisor gratuito oferecido pela Nutanix. É uma plataforma de virtualização segura e moderna que oferece uma solução leve e eficiente para a execução de máquinas virtuais em clusters Nutanix no local ou na nuvem pública.

KVM (*Kernel-based Virtual Machine*): É um hipervisor open source integrado ao kernel Linux. Com a KVM, você pode transformar o Linux em um hipervisor, permitindo que uma máquina host execute vários ambientes virtuais isolados.

VMware Workstation: É um software de virtualização que permite aos usuários configurar várias máquinas virtuais (VMs) e usá-las simultaneamente com a máquina real. Cada máquina virtual pode executar seu próprio sistema operacional, como o Windows, Linux ou variantes do BSD.

Oracle VirtualBox: É um software de virtualização de plataforma cruzada de código aberto desenvolvido pela Oracle. Ele permite que os desenvolvedores forneçam código mais rápido executando vários sistemas operacionais em um único dispositivo.

Microsoft Hyper-V: É uma plataforma de virtualização desenvolvida pela Microsoft que permite a criação e gerenciamento de máquinas virtuais em um ambiente de servidor.

Esses hipervisores são responsáveis por criar e gerenciar máquinas virtuais, alocando recursos do *hardware* físico para cada VM e garantindo que elas operem de forma isolada.

As máquinas virtuais desempenham um papel essencial em ambientes de computação em nuvem, centros de dados e desenvolvimento de software, proporcionando eficiência e flexibilidade no uso de recursos computacionais.

4.2 - Interface com o Usuário

Em informática, a interface com o usuário (UI - *User Interface*) refere-se ao ponto de interação entre o usuário e um sistema computacional. É o espaço onde a comunicação entre humanos e máquinas ocorre, permitindo que os usuários interajam e controlem o sistema. A interface com o usuário abrange todos os elementos e comandos que um usuário pode utilizar para interagir com um programa de software, dispositivo de hardware ou sistema.

A UI pode assumir diferentes formas, desde interfaces gráficas de usuário (GUI), que envolvem elementos visuais como botões, menus e janelas, até interfaces de linha de comando (CLI), onde os usuários interagem por meio de comandos de texto. Além disso, há interfaces táteis, de voz e gestuais, cada uma projetada para oferecer uma experiência de interação específica.

Uma boa interface com o usuário é projetada para ser intuitiva, eficiente e amigável, permitindo que os usuários realizem tarefas de forma fácil e eficaz. O design da UI é uma parte crucial do desenvolvimento de software e sistemas, pois influencia diretamente na usabilidade e na experiência do usuário.

4.2.1 - Acesso Remoto

Com o avanço da comunicação, em especial a internet, aumentou gradualmente a necessidade de controlar e supervisionar sistemas fora do alcance físico dos usuários. Alguns sistemas de medidas, controle e processo foram pioneiros no modelo de controle remoto [15]. A instrumentação industrial foi sem dúvidas a precursora desse movimento e se mantém à frente no que diz respeito ao avanço e desenvolvimento de tecnologias que proporcionam o monitoramento remoto.

Uma das ferramentas bastante utilizadas no âmbito industrial e recentemente residencial, com o avanço da domótica, é o *Virtual Networking Computing* - VNC [33]. O VNC como é popularmente conhecido, é um sistema que oferece acesso remoto aos computadores conectados à rede mundial por meio de interfaces gráficas. Os usuários, conectados a outro computador através do VNC, podem utilizar as funcionalidades visuais como se estivessem sentados diante do computador fora do campo de visão. Apesar de utilizar recursos visuais, a máquina em que o servidor VNC está rodando, não precisa de display físicos.

O VNC Viewer é um software que permite visualizar e interagir com a área de trabalho de um computador remotamente. VNC é uma sigla para "*Virtual Network Computing*". Esse tipo de software é usado para acesso remoto a computadores, o que é útil em situações em que você precisa controlar um computador a partir de outro local.

A forma como o VNC funciona é através de um servidor e um visualizador (viewer). O computador que você deseja acessar remotamente precisa ter um servidor VNC instalado e em execução. O VNC Viewer é então usado no computador de onde você deseja controlar o outro.

O processo geralmente envolve o seguinte:

- **Instalação do Servidor VNC:** O computador que você deseja acessar remotamente precisa ter um servidor VNC instalado. Existem várias implementações de servidores VNC disponíveis, como TightVNC, RealVNC, entre outros.
- **Configuração do Servidor:** Depois de instalar o servidor, você geralmente precisa configurá-lo, definindo uma senha para a conexão, por exemplo.
- **Instalação do VNC Viewer:** No computador do qual você deseja acessar o outro, você instala o VNC Viewer.
- **Conexão Remota:** Você inicia o VNC Viewer e fornece o endereço IP ou o nome do computador que possui o servidor VNC. Em alguns casos, você também precisará fornecer a senha definida durante a configuração do servidor.
- **Controle Remoto:** Uma vez conectado, você verá a área de trabalho do computador remoto na janela do VNC Viewer. Você pode interagir com ele como se estivesse sentado diretamente na frente desse computador.

O VNC é frequentemente usado para suporte técnico remoto, administração de servidores à distância, ou simplesmente para acessar um computador de casa quando você está fora.

O VNC é baseado no protocolo Remote Framebuffer - RFB, um protocolo simples aberto para acesso remoto a interfaces gráficas de usuário. Como funciona no nível do framebuffer, é aplicável a todos os sistemas e aplicativos de janelas, incluindo *Microsoft Windows*, *macOS* e o *X Window System*. RFB é o protocolo usado em *Virtual Network Computing* (VNC) e seus derivados.

Por ser baseado no RFB o VNC é compatível com diferentes plataformas e sistemas operacionais. Isso significa que o VNC é multiplataforma e pode ser utilizado sem problemas para acessar e monitorar computadores com diferentes sistemas operacionais. Devido a sua grande abrangência, o VNC foi incorporado a sistemas menos robustos como microprocessadores e adaptados a sistemas operacionais dedicados a estes *hardwares*.

O módulo de comunicação proposto nesta pesquisa utiliza o VNC instalado no Raspberry Pi 3 Model B. Com essa implementação, é possível acessar e controlar remotamente o Raspberry Pi, além de ter acesso direto aos periféricos conectados às suas portas USB, como equipamentos, sensores e sistemas de controle. Esse acesso via VNC permite que o usuário responsável pela manutenção, com privilégios de *root*, possa redefinir ou modificar as configurações do sistema conforme necessário, a partir de qualquer dispositivo com VNC instalado, aumentando assim a abrangência e flexibilidade do sistema supervisorio.

O VNC Viewer, software que habilita o acesso remoto, é facilmente instalado na Raspberry. É necessário acessar o site <https://www.realvnc.com/pt/connect/download/viewer/raspberrypi/> fazer o download do executável e instalá-lo. Outra possibilidade, a mais comum, é instalar a aplicação através do terminal utilizando os seguintes comandos:

- `sudo apt-get update`
- `sudo apt-get install realvnc-vnc-server`

Esse tipo de acesso, possibilita o controle direto do mini-computador e a transferência de arquivos tanto para a máquina remota, na sala de controle, quanto para o microprocessador instalado em campo, por meio do *SSH - Secure Socket Shell* nativo desta aplicação. Estes arquivos podem ser os arquivos das medidas realizadas, arquivos de configuração ou outros arquivos de interesse do usuário. Diante de tais possibilidades, fica claro a viabilidade de utilizar o recurso do VNC Viewer na Raspberry.

4.2.2 - Painel

PySimpleGUI é uma biblioteca de interface gráfica de usuário (GUI) para Python, projetada para simplificar a criação de interfaces de maneira notavelmente acessível. Sua característica distintiva é a abordagem direta e intuitiva, contrastando com algumas bibliotecas mais complexas disponíveis. O foco principal do PySimpleGUI é proporcionar simplicidade no desenvolvimento de GUIs. Especialmente orientado a iniciantes, o objetivo é garantir que seja fácil de aprender e usar.

Uma vantagem importante é o suporte multi-plataforma. Construído sobre bibliotecas gráficas estabelecidas como Tkinter, WxPython, PyQt e Remi, o PySimpleGUI possibilita que os programas criados sejam executados em diversas plataformas.

Além disso, oferece suporte a vários *backends* gráficos, permitindo que os desenvolvedores escolham a biblioteca subjacente de acordo com suas necessidades ou preferências. A flexibilidade abrange desde interfaces gráficas clássicas de desktop até interfaces baseadas na web. A documentação do PySimpleGUI é abrangente, repleta de exemplos e tutoriais para facilitar o início e a melhoria contínua das habilidades dos desenvolvedores. Contando com uma comunidade ativa, o PySimpleGUI dispõe de fóruns de suporte e recebe contribuições regulares para seu desenvolvimento contínuo.

A aplicação eficiente do PySimpleGUI é notável em projetos de pequeno e médio porte, onde a simplicidade é valorizada mais do que recursos avançados. Utilizando um estilo declarativo, os desenvolvedores descrevem como desejam que a interface seja, enquanto a biblioteca cuida da criação e atualização da GUI conforme necessário. O PySimpleGUI proporciona uma abordagem prática e fácil para criar interfaces gráficas em Python, destacando-se como uma escolha interessante para projetos que buscam eficiência e agilidade.

5. Métodos Geofísicos

De maneira geral, os métodos geofísicos podem ser definidos como a aplicação de princípios da física e matemática para descrever as propriedades da subsuperfície terrestre. Algumas características físicas utilizadas para essa descrição incluem condutividade elétrica, impedância acústica, densidade, susceptibilidade magnética e radiométrica [18]. Nesta pesquisa, os sistemas de medidas geofísicos monitorados são utilizados para caracterizar a subsuperfície por meio do contraste da condutividade elétrica.

Essas técnicas encontram ampla aplicação em diversas áreas, como exploração mineral, monitoramento ambiental, investigações geotécnicas, estudos hidrogeológicos e pesquisa de petróleo e gás, entre outras. A escolha do método específico a ser empregado depende dos objetivos particulares da investigação e das características do ambiente geológico em foco.

5.1 Eletrorresistividade

A aplicação de técnicas de medição de condutividade elétrica, utilizando conceitos e princípios geofísicos, tem se tornado cada vez mais comum nas atividades de exploração de recursos naturais e no monitoramento ambiental. Essas técnicas permitem inferir informações sobre a subsuperfície sem a necessidade de realizar sondagens e observações diretas das camadas subterrâneas. No passado, a geofísica era aplicada quase que exclusivamente na prospecção de hidrocarbonetos, minerais e reservatórios de água. Com o tempo, no entanto, as aplicações geofísicas expandiram-se, tornando-se comuns em contextos urbanos, desde a avaliação de terrenos para construção civil até o monitoramento de encostas e aterros sanitários. Alguns equipamentos geofísicos, originalmente desenvolvidos para outras finalidades, não foram concebidos para essas novas aplicações, o que cria a necessidade de adaptá-los para atender às demandas emergentes.

Entre os métodos geofísicos disponíveis, o método de eletrorresistividade é o que mais se destaca na solução de problemas urbanos mencionados anteriormente. Esse método permite a caracterização geológica local por meio da inferência da resistividade aparente da subsuperfície.

O método da eletrorresistividade baseia-se na Lei de Ohm e mede o contraste da condutividade elétrica da subsuperfície. Ele utiliza correntes elétricas induzidas artificialmente na superfície para investigar as propriedades elétricas do subsolo. A partir das medições do potencial, é possível determinar a existência de corpos condutores e identificar estruturas geológicas.

No método geofísico de eletrorresistividade, existem diversas técnicas de aquisição de dados. Essas técnicas podem ser divididas em dois grandes grupos: Sondagem Elétrica Vertical (SEV) e Caminhamentos Elétricos.

A Sondagem Elétrica Vertical (SEV), também conhecida como sondagem elétrica ou sonda de expansão, é uma técnica utilizada principalmente para estudar interfaces horizontais ou quase horizontais no subsolo. Durante o processo, os eletrodos de corrente e potencial são mantidos

em um espaçamento relativo constante, e o arranjo é gradualmente expandido em torno de um ponto central fixo. Dessa forma, as medições são realizadas à medida que as correntes penetram em profundidades progressivamente maiores. Essa técnica é amplamente empregada em levantamentos geotécnicos para determinar a espessura da sobrecarga e em hidrogeologia para identificar zonas horizontais de estratos porosos [18].

O caminhamento de separação constante (CST), também chamado de "perfilagem elétrica," é uma técnica utilizada para identificar variações verticais na resistividade do subsolo. Nesse método, os eletrodos de corrente e potencial são mantidos a uma distância fixa e são movidos progressivamente ao longo de um perfil. O CST é amplamente utilizado na prospecção mineral para localizar falhas, zonas de cisalhamento e detectar corpos com condutividade anômala. Além disso, é empregado em levantamentos geotécnicos para identificar variações na profundidade do embasamento e a presença de descontinuidades abruptas. Uma série de caminhamentos CST com espaçamento fixo entre os eletrodos pode fornecer resultados detalhados sobre essas variações [18].

O setup para realizar medidas de resistividade consiste basicamente um sistema de quatro eletrodos, dois deles de corrente **A** e **B** são usados para injetar uma corrente elétrica conhecida (**I**) e dois eletrodos de potencial **M** e **N** são usados para medir a diferença de potencial (ΔV) do espaçamento entre eles. A partir da relação entre corrente e tensão, lei de Ohm, estima-se a resistividade aparente (ρ_a) Eq. 1 da subsuperfície.

$$\rho_a = k \cdot \Delta V / I \quad (1)$$

ρ_a - Resistividade aparente

k - Fator geométrico

ΔV - Diferença de potencial

I - Corrente injetada no solo

Existem diferentes arranjos ou configurações de eletrodos que podem ser usados no método de eletrorresistividade, dependendo dos objetivos do estudo e das características do subsolo. Os arranjos de aquisição podem ser classificados em pelo menos cinco grupos:

- **Wenner:**

Este arranjo envolve quatro eletrodos, igualmente espaçados, dispostos em linha reta. Um par de eletrodos é usado para injetar a corrente elétrica, enquanto o outro par é usado para medir a diferença de potencial.

- **Schlumberger:**

O arranjo Schlumberger é semelhante ao Wenner, mas com a diferença de que a distância entre os eletrodos de corrente é aumentada. Isso permite investigar camadas mais profundas.

- **Dipolo-Dipolo:**

Neste arranjo, um par de eletrodos é usado para injetar corrente, e dois eletrodos adjacentes são usados para medir a diferença de potencial. Isso é repetido ao longo de uma linha de medição.

- **Pole-Pole:**

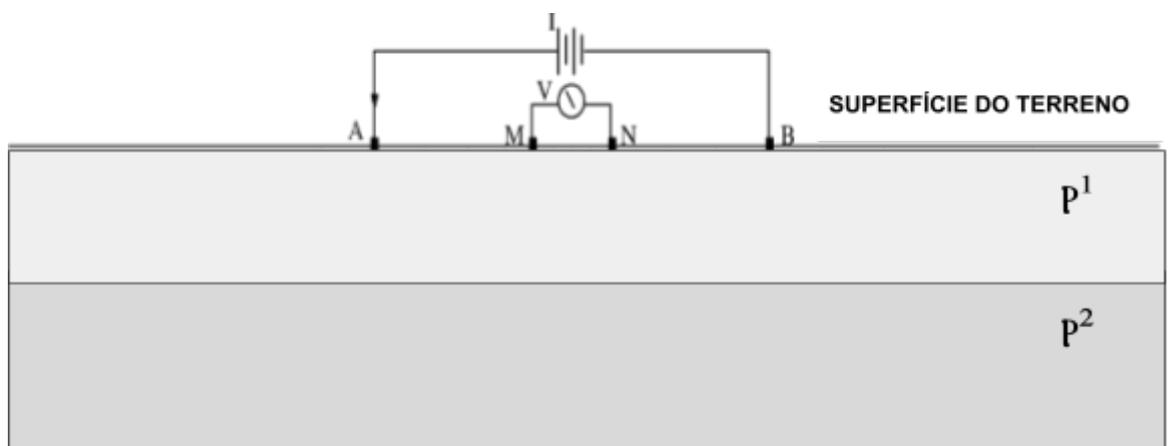
Um eletrodo é usado para injetar corrente e outro para medir a diferença de potencial. Isso é repetido para diferentes combinações de eletrodos, formando várias configurações. Os eletrodos são colocados a grandes distâncias considerando-se no infinito. A distância deve ser no mínimo 3 vezes a profundidade que se deseja investigar.

- **Polo-Dipolo:**

O arranjo pólo-dipolo, também conhecido como tripolo, consiste em três eletrodos móveis ao longo do perfil de caminhamento: um eletrodo de corrente (A) e dois eletrodos de potencial (MN), com espaçamento constante entre eles ($a = MN$). O eletrodo de corrente que completa o circuito, B, é mantido fixo a uma distância muito grande, chamada de "infinito", que geralmente corresponde de dez a vinte vezes a maior abertura entre os eletrodos extremos do arranjo na linha de levantamento. Essa grande distância é escolhida para garantir que o eletrodo B tenha influência mínima sobre o restante do arranjo.

A escolha do arranjo depende da aplicação específica e das condições geológicas do local. Alguns arranjos são mais adequados para investigações em grande escala, enquanto outros são mais eficazes para estudos detalhados em áreas específicas. O objetivo é obter dados que permitam uma interpretação precisa da distribuição da resistividade elétrica nas camadas em subsuperfície.

Os arranjos são caracterizados e distinguidos pelo espaçamento entre os eletrodos de corrente e tensão. A seguir a figura 5.1 sintetiza o arranjo do princípio do método de eletrorresistividade.



AB - ELETRODOS DE CORRENTE

P - RESISTIVIDADE DAS CAMADAS

MN - ELETRODOS DE POTENCIAL

Figura do autor

Figura 5.1 - Arranjo geral do sistema de aquisição do método da eletrorresistividade

5.2 Magnetotelúrico - MT

O método magnetotelúrico (MT) é uma técnica geofísica que mede a resistividade das rochas no subsolo usando medições no domínio da frequência. Esse método parte do princípio de que o campo eletromagnético da Terra é influenciado pelas propriedades elétricas da subsuperfície. Em sua essência, o MT determina a condutividade elétrica das formações rochosas ao calcular a razão entre os campos elétrico (E) e magnético (B) [7]. A estimativa da impedância (Z) da subsuperfície terrestre é expressa conforme a Equação 2.

$$Z = E * (i, j) / B * (i, j) \quad (2)$$

Comumente utilizado na exploração de petróleo, especialmente em áreas onde a reflexão sísmica é inviável ou muito cara [27], o MT é também destacado na exploração de bacias sedimentares, no estudo de regiões cratônicas para compreender as estruturas crustais profundas, e na prospecção de depósitos minerais em grandes profundidades [10].

A base do método reside na indução eletromagnética, conforme a Lei de Faraday. Quando um campo magnético varia em intensidade ao longo do tempo numa determinada região, isso provoca a geração de correntes elétricas no subsolo. Essas correntes resultam da interação entre o campo magnético variável e os materiais condutores presentes nas camadas subterrâneas.

Ao utilizar as equações de Maxwell, o método magnetotelúrico descreve a propagação de campos elétricos e magnéticos na subsuperfície, estabelecendo relações entre as variações espaciais e temporais desses campos com as propriedades elétricas dos materiais presentes. No entanto, devido à complexidade dessas equações, que envolvem derivadas parciais, a resolução analítica torna-se desafiadora. Assim, a abordagem predominante é representar as equações em termos de impedâncias, expressando as relações entre os campos elétricos e magnéticos [16]. Essas equações podem ser resolvidas numericamente, fornecendo informações cruciais sobre a resistividade dos materiais subsuperficiais [10].

As equações fundamentais do eletromagnetismo que descrevem a propagação da onda eletromagnética no meio são a base do método MT. Dentre as equações podemos destacar duas: A equação da continuidade do campo magnético (B), equação 3 e a equação de Maxwell para o campo elétrico (E), equação 4.

$$\nabla \times B = \mu \sigma E + \mu E dE/dt \quad (3)$$

$$\nabla \times E = - dB/dt \quad (4)$$

O arranjo das instalações dos sistemas de medidas magnetotélúricas obedecem a perpendicularidade entre os campos elétricos e magnéticos, de modo que as componentes registradas são ortogonais entre si. São realizadas medições das séries temporais dos campos elétricos e magnéticos naturais na superfície terrestre, utilizando dois conjuntos de eletrodos para medir o campo elétrico e um sensor magnético, *fluxgate*, para medir o campo magnético. A seguir, na figura 5.2 é apresentado um modelo de arranjo simétrico bastante utilizado nas aquisições de dados MT. Os canais elétricos são representados por dipolos de 100m E_x - direção Norte-Sul e E_y - direção Leste - Oeste. Por sua vez, os campos magnéticos são representados por B_x na direção Norte, B_y na direção Leste e B_z na direção vertical.

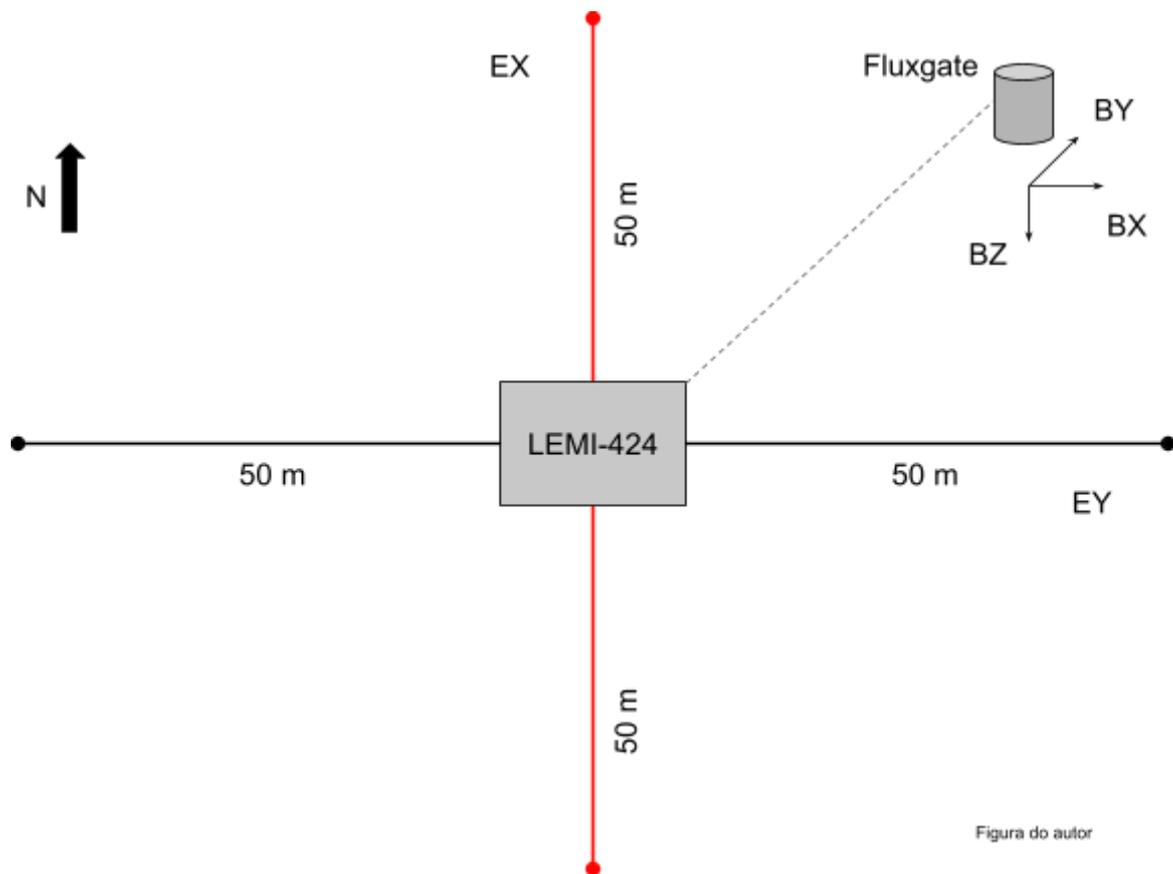


Figura do autor

Figura 5.2 - Arranjo utilizado na aquisição de dados MT

A aquisição de dados no método magnetotélúrico (MT) envolve a medição das componentes horizontais dos campos elétricos e magnéticos naturais na superfície terrestre, com o processamento geralmente realizado no domínio da frequência por meio da Transformada de Fourier (FT). O MT é uma técnica versátil, que se adapta a investigações em profundidades que variam de dezenas de metros a dezenas de quilômetros. Sua principal vantagem é a capacidade de explorar diferentes profundidades sem depender de fontes artificiais, o que reduz o impacto ambiental. No entanto, uma desvantagem significativa do MT é a complexidade na obtenção de dados em áreas com elevado ruído eletromagnético [44].

O método magnetotelúrico envolve a análise dessas medições em diversas frequências, normalmente abrangendo intervalos que vão de frações de milihertz (mHz) a vários quilohertz (KHz). À medida que correntes elétricas são induzidas no subsolo, elas geram ondas eletromagnéticas que se propagam por diferentes profundidades e escalas de tempo. A penetração dessas ondas no subsolo é influenciada pela resistividade elétrica dos materiais presentes. A resistividade elétrica dos materiais subterrâneos afeta a amplitude e a fase das ondas eletromagnéticas. Materiais mais condutores apresentam menor resistividade, impactando a forma como as ondas se propagam e são registradas nas medições de campo.

Após a coleta, os dados são processados no domínio da frequência, permitindo a análise de como as propriedades elétricas do subsolo variam com a frequência das ondas eletromagnéticas. Com base nesses dados, os geofísicos realizam uma modelagem inversa para determinar a distribuição da resistividade elétrica em diferentes profundidades, fornecendo assim informações detalhadas sobre a estrutura geológica e a composição do subsolo.

6. Metodologia

Este capítulo apresentará os materiais, métodos e processos utilizados na implementação do sistema supervisorio para os sistemas de medidas de eletrorresistividade, Supersting R8, e magnetotelúrico, LEMI-424. A abordagem começa pela apresentação da arquitetura do sistema de monitoramento, destacando como os componentes de hardware e software foram integrados para garantir uma operação eficiente e precisa. Em seguida, serão discutidas as etapas de configuração dos processos, que envolvem desde a instalação dos equipamentos até a calibragem e a parametrização dos sistemas. Essa descrição abrangente proporcionará uma compreensão completa das ferramentas e técnicas empregadas, permitindo ao leitor avaliar a eficácia do sistema supervisorio desenvolvido e sua aplicabilidade em diferentes cenários de monitoramento geofísico.

6.1 - Arquitetura do Sistema de Monitoramento

O módulo de comunicação foi desenvolvido integrando diferentes bibliotecas em Python e hardware open source. Essa integração confere ao projeto um caráter multidisciplinar [52].

Utilizando a linguagem de programação Python 3 [30] e algumas bibliotecas (Redis, Kivy, PySimpleGUI), juntamente com o microprocessador Raspberry Pi 3 Model B, projetamos o módulo de comunicação proposto nesta pesquisa.

A arquitetura do sistema supervisorio, representada na Figura 6.1, envolve a comunicação serial entre um sistema de medida (o equipamento geofísico) e o microprocessador [12]. O banco de dados Redis, programado nas rotinas em Python executadas no microprocessador, permite o envio de comandos, consulta de resultados e status do equipamento. Interfaces gráficas foram desenvolvidas para facilitar essas tarefas, possibilitando o envio de comandos e consulta das chaves do Redis, onde são armazenados valores das medidas e status dos processos e equipamentos.

Com base nessas aplicações, é possível estabelecer comunicação e supervisão dos equipamentos a partir de uma sala de controle. A comunicação remota entre a sala de controle e o equipamento, fora do campo de visão do usuário, é realizada através da estrutura de dados configurada no banco de dados.

A arquitetura proposta possibilita a interação com os sistemas de medidas por meio de interfaces e aplicativos de controle [52]. As interfaces de controle podem ser escritas em Python, de acordo com as especificações dos sistemas de medidas, ou utilizar o software fornecido pelo fabricante. Em ambos os casos, a interface de controle envia comandos ao equipamento e recebe os valores das medidas realizadas. Os comandos podem incluir calibração, aquisição, reconfiguração ou reset, cada um com sua rotina específica [52]. Os valores das medidas são repassados ao aplicativo de controle, alocados no Redis e armazenados na memória local do microprocessador em um arquivo de texto.

O aplicativo de controle gerencia o fluxo de comandos e armazena os valores das medidas nas chaves do banco de dados Redis. O acesso às chaves e valores do banco de dados é realizado

em tempo real. A estrutura do aplicativo de controle, escrito em Python, estabelece conexão com o Redis ao ser inicializado.

O banco de dados Redis possibilita que todos os comandos e armazenamentos de medidas sejam realizados em nuvem. Um servidor dedicado, uma máquina virtual ou uma máquina dedicada armazena a estrutura de chave/valor do banco de dados. Após a instalação, a conexão com o Redis é estabelecida por meio do endereço IP, porta de acesso e senha. Isso significa que o acesso às informações pode ser realizado de qualquer máquina ou aplicativo que tenha as informações de segurança [29].

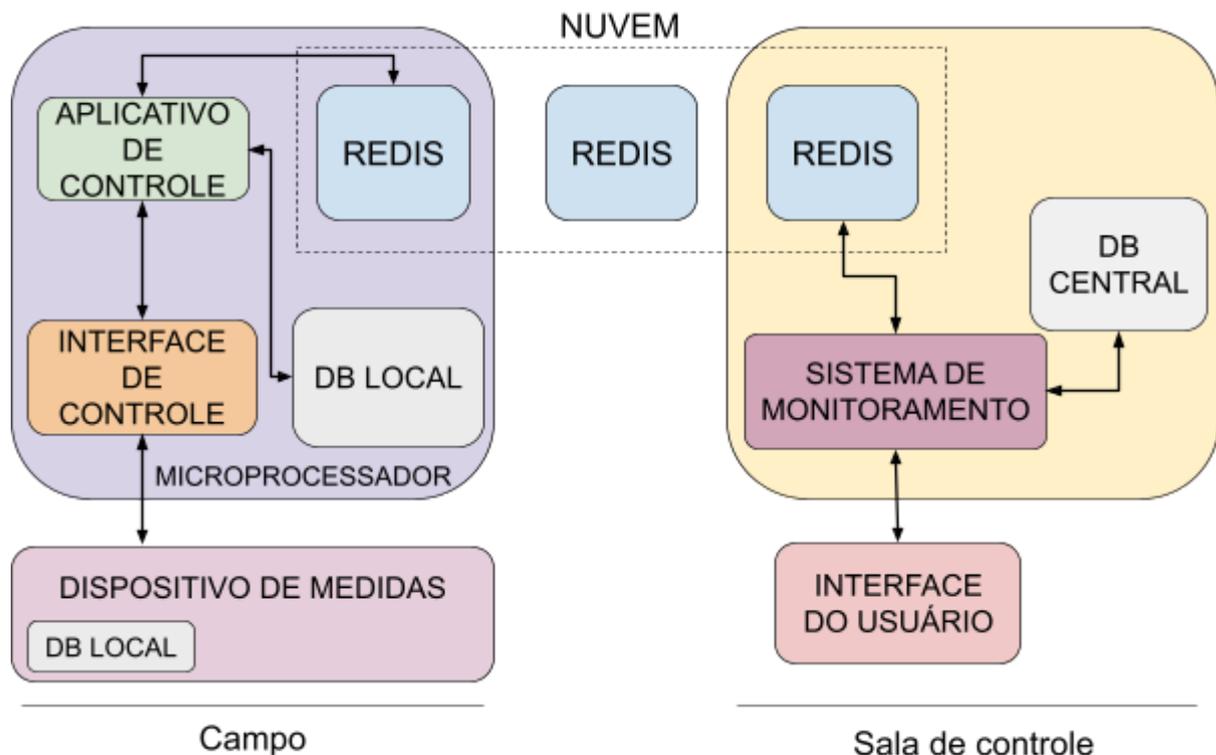


Figura do autor

Figura 6.1 - Representação do funcionamento da estrutura de supervisão remota

O usuário remoto tem à disposição interfaces gráficas (GUIs) desenvolvidas em PySimpleGUI. Através dessas interfaces, é possível verificar a disponibilidade dos equipamentos e enviar requisições de medidas.

Nessas GUIs, os usuários inserem requisições de consulta e comandos de aquisição. Durante o processo de solicitação de medidas, o front-end do módulo de comunicação associa a opção selecionada às chaves do Redis. Para cada valor armazenado nas respectivas chaves, há uma rotina em Python que executa o tipo de medida escolhido. O microprocessador de campo consulta a chave através do aplicativo de controle, envia o comando de configuração ao equipamento que executa a medida e, em seguida, armazena os resultados em chaves dedicadas para receber os valores das medidas. Na solicitação de consulta, status e resultados, o front-end verifica os valores das chaves dedicadas a armazenar essas informações e os exibe

ao requisitante. O sistema de monitoramento tem a capacidade de interagir com o equipamento por meio dos comandos enviados ao banco de dados ou diretamente através da conexão com o Raspberry via VNC Viewer.

Dentre os inúmeros microprocessadores disponíveis, a Raspberry Pi 3 Model B V1.2 foi escolhida para implementar o projeto devido à familiaridade com o dispositivo e à disponibilidade, obtida em colaboração com o PEGBr (www.pegbr.on.br). Além disso, a configuração do *hardware* foi um fator considerado na escolha do microprocessador. A capacidade de utilizar as interfaces nativas de rede, portas USB e saída HDMI se mostraram excelentes para implementar o modelo da arquitetura idealizada.

O sistema operacional embarcado no microprocessador é o Raspbian, baseado no Debian. Esse sistema estável recebe atualizações continuamente, garantindo a perenidade ao *hardware*.

Conforme apresentado no diagrama da Fig. 6.1, existe redundância no armazenamento das medidas. As medidas são armazenadas em quatro instâncias: na memória local do equipamento, no *storage* do microprocessador, no Redis e no *storage* da máquina do usuário. Caso seja necessário liberar espaço no banco de dados, os valores armazenados nas chaves podem ser excluídos. Isso significa que os dados armazenados no Redis podem ser temporários, e o usuário tem a liberdade de definir por quanto tempo os valores deverão ser mantidos.

6.2 - Processos de Configuração do Sistema

Nesta seção serão consideradas algumas etapas de configuração e inicialização do sistema. É importante frisar que os fluxogramas simplificados ilustram as etapas realizadas para configurar o banco de dados, instalado no servidor e na máquina local, raspberry, dedicada a realizar a interface com os sistemas de medidas instalados em campo.

6.2.1 - Configuração do Servidor em Nuvem

A Figura 6.2.1 ilustra as etapas necessárias para configurar o banco de dados. Inicialmente, é preciso definir o servidor e suas especificações. O banco de dados Redis pode ser instalado em um servidor local, como uma máquina dedicada, ou em um servidor virtual na nuvem. nesta pesquisa, o Redis foi hospedado em um servidor virtual disponibilizado pelo RedisLab. Em ambos os casos, o Redis é considerado um banco de dados na nuvem, o que significa que os valores armazenados nas chaves do banco podem ser acessados de qualquer máquina que tenha acesso ao Redis.

Ao utilizar o RedisLab, a inicialização da nuvem e suas configurações, como ponto de acesso e capacidade de armazenamento, são determinadas pelos mantenedores do site. O ponto de acesso do Redis é definido pelos seguintes requisitos:

- ***host_Redis*** (string) - Semelhante ao endereço IP
- ***port_Redis*** (inteiro) - porta de acesso
- ***password_Redis*** (string) - senha definida pelo usuário

Após a configuração do servidor, conforme indicado no fluxograma, torna-se possível utilizar as definições do banco de dados nos códigos em Python. Nas próximas seções, será exemplificado como estabelecer a conexão com o Redis e ter acesso às suas respectivas chaves/valores.

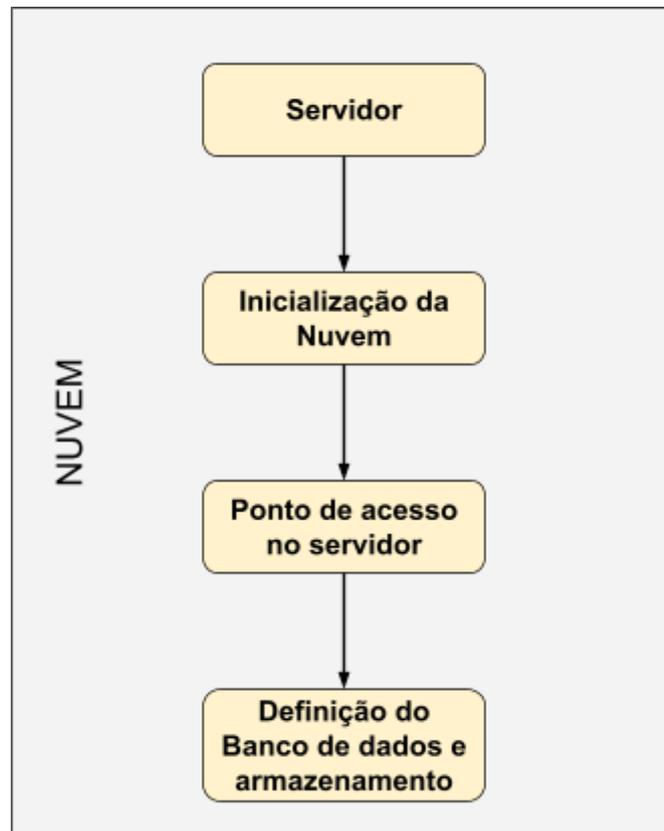


Figura 6.2.1 - Fluxograma de inicialização do banco de dados

6.2.2 - Inicialização da Máquina Local

A máquina local, representada pelo microprocessador Raspberry, assume a responsabilidade de estabelecer a conexão entre o sistema de medida e o banco de dados. Após a conexão bem-sucedida com o Redis, torna-se possível identificar os valores associados às chaves da estrutura de dados e enviar comandos ao equipamento. O acesso às medidas pode ser realizado diretamente por meio da leitura da porta serial ou através dos arquivos armazenados no banco de dados local. Para realizar essas operações, é necessário seguir as etapas representadas pelo fluxograma na Figura 6.2.2.

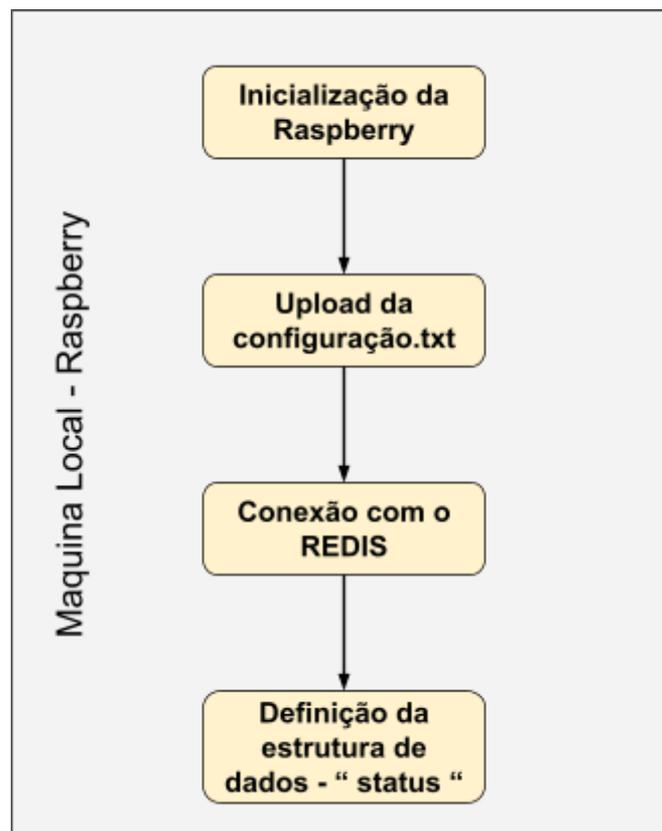


Figura 6.2.2 - Fluxograma de inicialização do sistema

A inicialização da Raspberry ocorre após o acionamento do sistema de alimentação de 5V/2A. Nesta etapa, pressupõe-se que o sistema operacional Raspbian já está instalado e configurado no cartão SD.

Após a inicialização, uma rotina em Python é automaticamente executada. Nessa rotina, ocorre a configuração da porta serial à qual o sistema de medida está conectado e do ponto de acesso ao Redis. Os parâmetros de configuração são definidos em um arquivo chamado "file_config.txt" presente no mesmo diretório do código Python.

Após a conexão com o Redis, é criada uma estrutura de dados no formato chave/valor. Essa estrutura define as chaves necessárias para realizar a configuração, medidas e obter o status dos sistemas de medida. A Tabela 6.2 apresenta a estrutura de dados simplificada do sistema supervisor.

STATUS	inteiro	Condição do equipamento
MEDIDA	float	Armazena as medidas
COMANDO	string	Qual medida deverá ser realizada
EQUIPAMENTO	string	Modelo do Equipamento
ESTAÇÃO	string	Local de instalação

Tabela 6.2 - Estrutura de dados simplificada do sistema de monitoramento

6.2.3 - Loop Principal- Autenticação e Inicialização do Sistema de Aquisição

Após a inicialização e configuração da máquina local, o programa principal é executado para estabelecer a comunicação entre os dispositivos de medida e o banco de dados Redis. O fluxograma apresentado na Figura 6.2.3 sintetiza o processo de conexão entre o banco de dados e a máquina local. Antes de iniciar o processo de medidas, é necessário validar as requisições enviadas pelos usuários por meio do banco de dados.

A estrutura de dados definida pelas chaves do Redis é verificada antes de iniciar o processo de medida. A verificação das chaves indica a possibilidade de executar o tipo de medida desejada, a disponibilidade do equipamento e seu respectivo status. Após a verificação, os comandos básicos, como play, pause, stop e status, são enviados para o segundo estágio do fluxo de processamento, que é a configuração do sistema de medida.

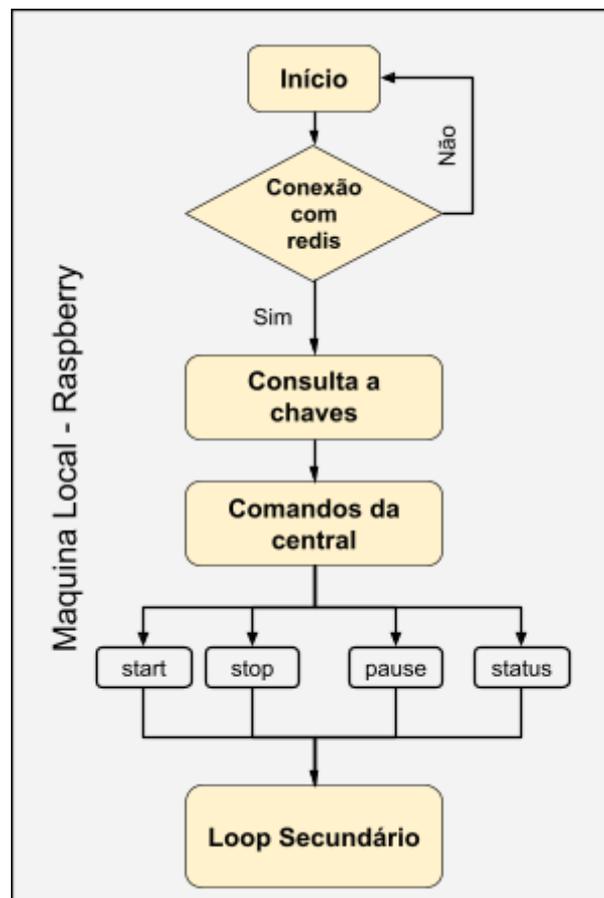


Figura 6.2.3: Fluxograma de Autenticação e Inicialização do sistema de aquisição

6.2.4 - Loop Secundário - Configuração do Equipamento

Após a validação das requisições, ou seja, após verificar a disponibilidade do equipamento, é necessário configurar o sistema de medida para o arranjo desejado. A validação da requisição é realizada pela sala de controle, avaliando a disponibilidade dos equipamentos e o tipo de medida desejada. Após a validação, a estrutura de dados é verificada. Os valores armazenados na estrutura de dados estão associados às rotinas de configuração do equipamento geofísico. Após a configuração, o sistema estará habilitado para executar a medida. O fluxograma apresentado na Figura 6.2.4 resume o processo de configuração do equipamento. Os comandos utilizados nas sub-rotinas de configuração são comandos ASCII enviados pelo programa em Python através da porta serial.

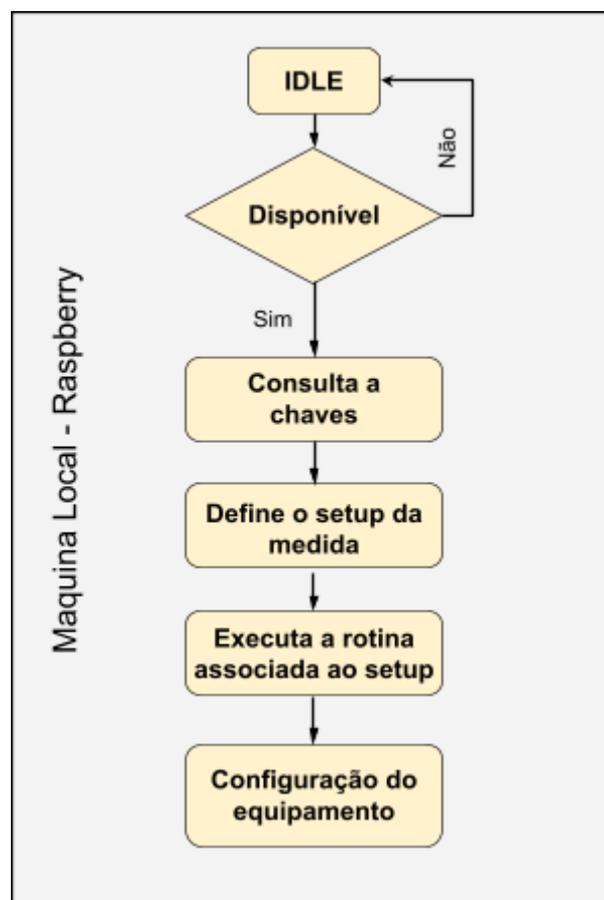


Figura 6.2.4 : Loop de configuração do sistema de medida

6.4.5 - Loop de Medidas

O terceiro bloco do sistema de aquisição consiste em executar a rotina do tipo de medida selecionada pelo usuário e armazenar os respectivos valores no Redis e na memória local da Raspberry, conforme representado na Figura 6.4.5. Para cada tipo de medida, há uma sub-rotina específica, cujos comandos em ASCII são enviados ao equipamento via porta serial. Após executar as medidas, o equipamento envia os valores através da porta serial. A sub-rotina recebe os valores e os armazena no banco de dados local no formato .txt. Em seguida, verifica a conexão com o Redis e salva as medidas na estrutura de dados definida previamente. Caso a conexão com o Redis não seja estabelecida, após cinco tentativas, um flag indicando que não foi possível salvar a medida é adicionado a um arquivo .txt. Após restabelecer a conexão, uma rotina é executada para realizar o upload dos dados nas chaves do Redis.

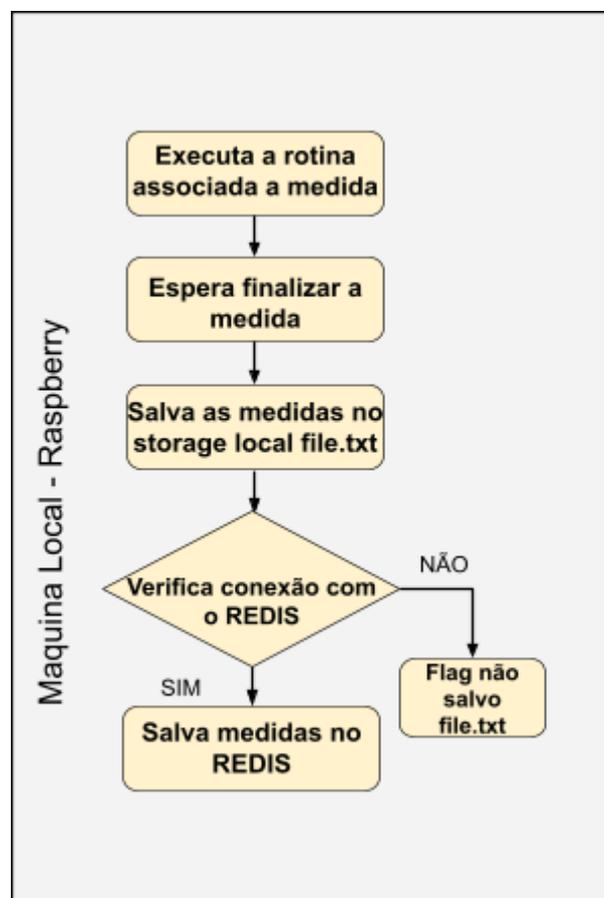


Figura 6.4.5: Loop de medidas

7. Sistemas de Medidas Geofísicas

Os sistemas de medidas obedecem a uma estrutura organizada de componentes inter-relacionados projetados para realizar a medição de grandezas físicas. Esses sistemas são fundamentais em diversas áreas, desde ciências físicas e engenharia até aplicações industriais e experimentos científicos [21]. Eles consistem em vários elementos essenciais:

Sensor ou Transdutor - É o componente responsável por converter a grandeza física a ser medida em um sinal elétrico ou outro tipo de sinal mensurável. Exemplos incluem termômetros, fotodetectores, acelerômetros, entre outros.

Instrumentação Eletrônica - Esta parte do sistema processa o sinal gerado pelo sensor. Pode envolver amplificação, filtragem, conversão analógico-digital (ADC), entre outros processos.

Exibição ou Indicação - O resultado da medição é geralmente apresentado ao usuário ou ao sistema de controle em uma forma compreensível. Pode ser um mostrador analógico, um display digital ou até mesmo um sinal elétrico enviado para outro sistema.

Unidades de Processamento e Controle - Em muitos sistemas de medida, há componentes dedicados ao controle e à gestão do processo de medição. Isso pode incluir microprocessadores ou microcontroladores que executam algoritmos específicos para otimizar a precisão da medição.

Fonte de Energia - Muitos sistemas de medida requerem uma fonte de energia para alimentar os componentes eletrônicos, sensores e outros dispositivos do sistema.

Interface de Comunicação - Em alguns casos, é necessário que o sistema de medida comunique os resultados ou receba comandos externos. Isso pode envolver interfaces como USB, *Ethernet*, ou outras formas de conectividade.

A qualidade de um sistema de medida depende da precisão, exatidão e confiabilidade de seus componentes, bem como da calibração adequada para garantir resultados precisos e consistentes. Esses sistemas desempenham um papel crucial em setores como automação industrial, pesquisa científica, monitoramento ambiental, entre outros.

7.1 - Medidas de Resistividade

É um resistímetro e medidor IP - Polarização Induzida, multi-canal, fabricado pela Advanced Geosciences Inc. (AGI), bastante utilizado em pesquisas geofísicas [2]. É considerado um dos equipamentos mais precisos e menos ruidosos do mercado. A capacidade de registrar os dados em 8 canais simultâneos, reduz drasticamente o tempo de aquisição.

O resistímetro multicanal mencionado é uma ferramenta crucial em pesquisas geofísicas. Destaca-se por sua precisão e baixo nível de ruído, sendo considerado um dos equipamentos mais confiáveis do mercado. Sua capacidade de registrar dados em oito canais simultâneos é especialmente vantajosa, proporcionando uma significativa redução no tempo de aquisição.

Este equipamento é empregado para estimar a condutividade da subsuperfície, desempenhando um papel fundamental em diversas aplicações geofísicas. Suas funcionalidades encontram aplicação em áreas como detecção de cavidades, exploração de

água subterrânea e identificação de áreas contaminadas. Esses estudos geofísicos se beneficiam significativamente da eficácia e confiabilidade do resistímetro multicanal, contribuindo para uma análise mais precisa e abrangente da subsuperfície.

A figura 7.1.0 ilustra o resistímetro Super Sting R8 IP e suas respectivas conexões. A tabela 7.1.0 correlaciona os números apresentados na ilustração com as respectivas funções operacionais.



Fig. 7.1.0 - Resistímetro SuperSting R8 [2]

1	Conector de alimentação
2	Fusível
3	Chave seletora ON/OFF
4	Luz do Display
5	Conexão para transmissor externo
6	Indicador externo para medidas e modo booster
7	Porta serial para upload do firmware
8	Display em cristal líquido

9	Teclado
10	Conector para cabos dos eletrodos ou switch box
11	Terminal positivo e negativo para correntes
12	Terminal positivo e negativo para tensão
13	Terminal de teste
14	Conexão serial com o PC
15	Conector para desenvolvimento futuro

Tabela 7.1.0 - Relação dos conectores do resistivímetro Super Sting R8 IP

7.1.1 - Interface Serial

A porta serial do equipamento identificada na imagem anterior, foi adaptada para estabelecer comunicação com o sistema de monitoramento. Esta porta serial é dedicada à comunicação com o software do fabricante instalado no computador utilizado para trabalhos de campo.

Inicialmente foi necessário identificar quais eram os comandos que o equipamento utilizava para se comunicar com o software. Para isso foi necessário monitorar a porta USB do computador e identificar os comandos associadas às funções do sistema operacional do equipamento. Foi instalado o software Serial Port Monitor no computador de campo para inspecionar e rastrear os comandos em ASCII. Após o envio de comandos via software do fabricante foi definido o padrão utilizado pelo equipamento.

Após esta etapa um código em python foi escrito para estabelecer comunicação com o equipamento e controlá-lo através dos comandos enviados pelo teclado do computador. Que posteriormente foram enviados automaticamente pelas rotinas escritas em Python.

7.1.1.2 - Interface de Controle

Em situações em que não há interfaces desenvolvidas pelo fabricante compatíveis com a Raspberry Pi, a criação de uma interface de controle em Python se mostra uma solução eficaz. O monitoramento do tráfego da porta serial para identificar os comandos ASCII associados às funções do equipamento é uma abordagem inteligente. Isso permite uma integração mais direta e personalizada com o sistema de medidas, adaptando-o às necessidades específicas do projeto.

A capacidade de observar e compreender o tráfego da porta serial durante o uso do software do fabricante é fundamental para a implementação bem-sucedida da interface de controle personalizada. A abordagem de monitoramento e identificação de comandos é comumente empregada em projetos de automação e controle, proporcionando uma maior flexibilidade e controle sobre o hardware utilizado [53].

7.2 - Medidas Magnetotélúricas de Longo Período

O LEMI-424 é um avançado sistema de aquisição de dados eletromagnéticos projetado para o método magnetotélúrico. Este sistema de medidas é composto por uma unidade registradora Figura 7.2.1 e um magnetômetro do tipo fluxgate analógico Figura 7.2.2. A unidade registradora foi desenvolvida para digitalizar e armazenar os dados magnéticos proveniente do sensor fluxgate e os sinais elétricos captados pelos eletrodos não polarizáveis, Figura 7.2.3. Na Figura 7.2.4 temos o e-lines responsável pela conexão dos eletrodos ao sistema de digitalização.

Equipado com cinco canais independentes, incluindo um sensor magnético do tipo fluxgate, o sistema registra as componentes do campo eletromagnético terrestre nas direções Norte-Sul (H_x), Leste-Oeste (H_y) e Vertical (H_z). Além disso, possui dois pares de eletrodos não polarizáveis dedicados ao registro da variação do campo elétrico em mV/m , tanto nas direções Norte-Sul quanto Leste-Oeste.

Operando em baixa frequência, com uma taxa de amostragem em torno de 1 Hz, o LEMI-424 possibilita estimar a condutividade de estruturas em profundidades consideráveis, alcançando centenas de quilômetros de profundidade. O tempo necessário para a aquisição desses dados pode variar entre 7 a 12 dias consecutivos. Dada essa extensão de tempo, o monitoramento remoto dos dados adquiridos torna-se crucial. Qualquer alteração no posicionamento vertical ou horizontal do fluxgate pode comprometer as medidas, exigindo uma nova instalação. O mesmo se aplica a eletrodos que possam ser desconectados ou alterados durante o processo de aquisição.



Figura 7.2.1 - Unidade registradora do LEMI - 424



Figura 7.2.2 - Sensor magnético do tipo fluxgate



Figura 7.2.3 - Eletrodos não-polarizáveis utilizados na aquisição de dados MT

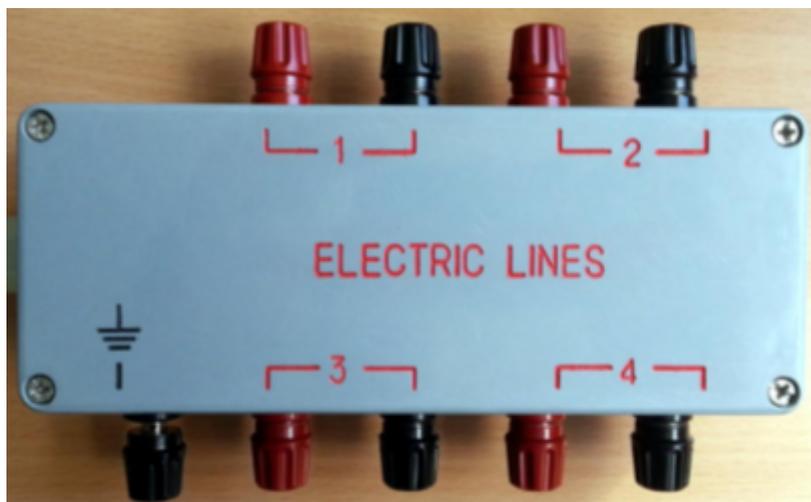


Figura 7.2.4 - Electric lines

7.2.1 - Características e Especificações da Unidade Digitalizadora

O equipamento LEMI 424 é notável por suas características distintas que o posicionam como uma escolha avançada para aplicações geofísicas. Com alta resolução e precisão, este dispositivo garante medições confiáveis em ambientes desafiadores. O ruído extremamente baixo contribui para a exatidão dos dados coletados, enquanto a presença de 4 canais elétricos e 4 canais magnéticos oferece uma capacidade abrangente de aquisição de informações.

A deriva temporal e térmica excepcionalmente baixa aprimora a estabilidade do equipamento ao longo do tempo, assegurando resultados consistentes. Além disso, seu baixo consumo de energia otimiza a eficiência operacional, sendo uma escolha econômica e sustentável.

A capacidade de sincronização via satélite adiciona uma dimensão de precisão, proporcionando uma referência temporal confiável para as medições. A interface do usuário é intuitiva, graças ao display gráfico com tela sensível ao toque, simplificando a configuração e monitoramento.

Projetado para resistir a condições adversas, o LEMI 424 apresenta um estojo plástico à prova d'água, garantindo a durabilidade do equipamento mesmo em ambientes úmidos. A oferta de dois modelos de unidades de proteção contra raios permite adaptabilidade às necessidades específicas de proteção.

Em termos de especificações técnicas, o equipamento opera na banda de frequência DC-0.5 Hz, com uma faixa de medida de ± 2450 mV. A notável resolução dos canais elétricos (ADC 32 bits) atinge 2 nV, proporcionando uma capacidade de detecção extremamente sensível. Com uma taxa de amostragem de 1 por segundo, o dispositivo atende às demandas de coleta de dados em tempo real.

Para armazenamento eficiente, o LEMI 424 está equipado com um cartão SD de 8 GB. A conectividade é garantida por meio de saída digital e controle USB, oferecendo versatilidade na transferência e análise de dados. A temporização GPS, juntamente com a determinação de coordenadas e altitude, é facilitada pelo comprimento do cabo da antena de 3 metros.

A ampla faixa de temperatura operacional, de -20 a $+60^{\circ}\text{C}$, torna o LEMI 424 adequado para uma variedade de condições ambientais. A fonte de alimentação (5-20) V proporciona flexibilidade na escolha de fontes de energia, enquanto o baixo consumo de energia, inferior a 0,35 W, contribui para a eficiência energética do dispositivo.

Em termos de portabilidade, a unidade eletrônica pesa 2,0 kg, enquanto a unidade de proteção contra raios pesa 1,1 kg, facilitando o transporte e manuseio durante operações de campo. O LEMI 424 emerge como uma solução abrangente, combinando precisão, durabilidade e eficiência em um único dispositivo.

7.2.2 - Características e Especificações do Fluxgate

O sensor magnético, magnetômetro, do tipo fluxgate é uma peça fundamental para medições magnéticas precisas, e o modelo em questão apresenta características notáveis, destacando-se pela sua alta resolução e precisão. Com baixo ruído e desvio de temperatura, esse sensor proporciona resultados confiáveis mesmo em ambientes desafiadores.

A inclusão de um canal de temperatura é uma vantagem adicional, permitindo uma compreensão mais abrangente das condições ambientais durante as medições. A facilidade de instalação e manutenção simplificam a integração desse sensor em diversas aplicações geofísicas.

A eficiência energética é uma característica proeminente, evidenciada pelo baixo consumo de energia do sensor fluxgate. Na saída analógica, a faixa de medida atinge ± 65000 nT, oferecendo uma capacidade de detecção ampla para variações magnéticas. A banda de frequência para o magnetômetro abrange de DC a 10 Hz, permitindo a análise de diferentes componentes do campo magnético.

O fator de transformação da saída analógica é de $25 \mu\text{V/nT}$, contribuindo para a precisão na leitura dos dados. O nível de ruído em 1 Hz é extremamente baixo, com valor igual ou inferior a $10 \text{ pT}/\sqrt{\text{Hz}}$, assegurando a sensibilidade do sensor mesmo em condições de baixo sinal.

A deriva de temperatura é mantida abaixo de $0.3 \text{ nT}/^\circ\text{C}$, garantindo a estabilidade das medições ao longo do tempo. O erro de ortogonalidade dos componentes é mínimo, ficando abaixo de 30 minutos de arco, o que é crucial para assegurar a precisão nas medições multidirecionais.

A faixa de temperatura de operação do sensor fluxgate varia de -20 a $+60^\circ\text{C}$, tornando-o adequado para uma ampla gama de condições ambientais. A fonte de alimentação requer apenas 5 V, proporcionando uma opção flexível e eficiente.

Em termos de peso, o sensor com suporte e cabo de 10 metros pesa 3 kg, tornando-o relativamente leve para fácil transporte e instalação. O comprimento do cabo de conexão do sensor pode variar de 6 a 75 metros, oferecendo versatilidade na disposição e posicionamento do sensor em diferentes cenários.

Em suma, o sensor fluxgate apresenta-se como uma solução abrangente e confiável para aplicações que demandam medições magnéticas precisas, combinando alta tecnologia, eficiência energética e robustez em um único dispositivo.

7.2.3 - Interface de Controle

Para esta linha de equipamentos, o fabricante disponibilizou uma interface gráfica que é compatível com a Raspberry Pi. Essa interface possibilita a configuração e a execução de medidas diretamente no equipamento. Para automatizar esses comandos, desenvolvemos uma rotina em Python capaz de inicializar o programa executável fornecido pelo fabricante conforme ilustrado na Figura 7.2.3.

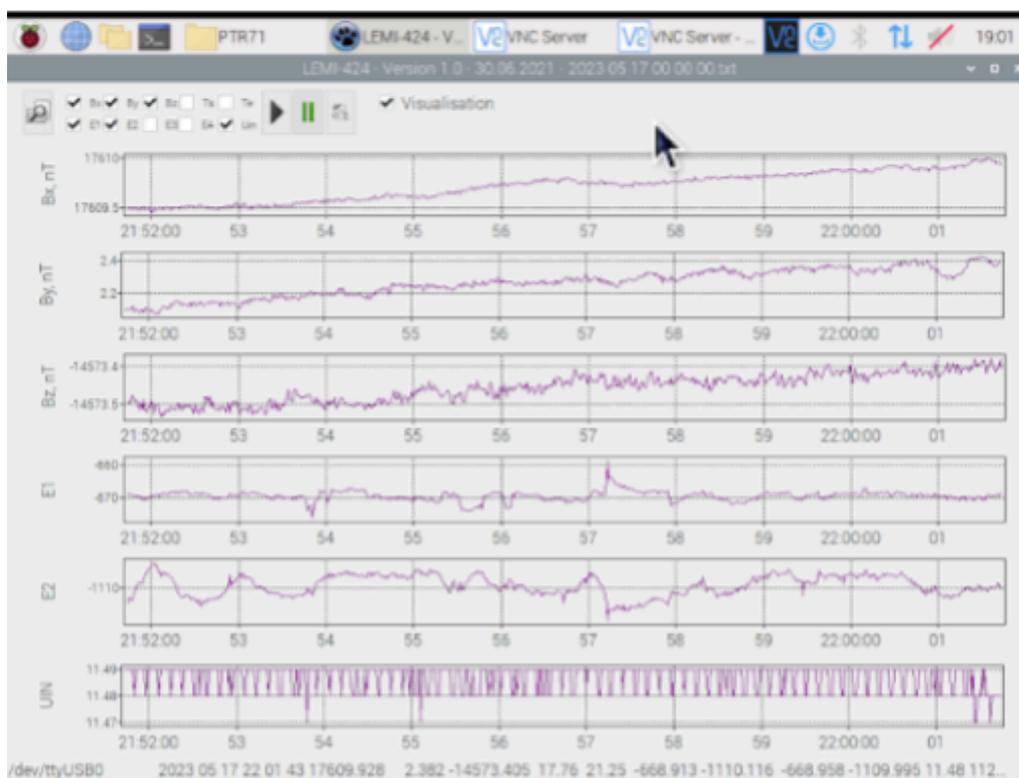
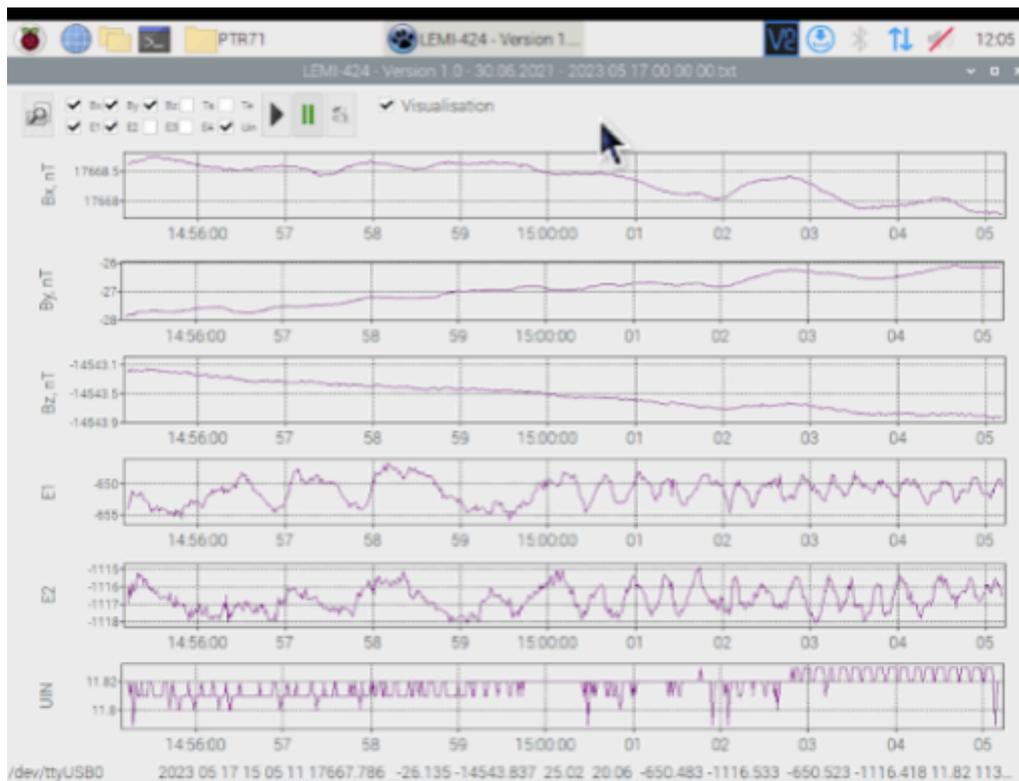


Figura 7.2.3 - Interface de controle do LEMI-424 fornecida pelo fabricante

Uma vez que a interface é iniciada, o equipamento começa a realizar medidas de forma contínua. Os dados resultantes são salvos no diretório da máquina local e, posteriormente, são armazenados no banco de dados Redis, conforme explicado na seção anterior.

8. Resultados e Discussão

Os resultados apresentados nesta seção destacam a eficácia da arquitetura do sistema supervisorio proposto no módulo de comunicação detalhado neste artigo. Esses resultados são divididos em duas categorias principais: os dados coletados pelos sistemas de medição, apresentados graficamente, e as conexões automáticas estabelecidas entre os componentes de *hardware* e *software*, notavelmente o banco de dados em nuvem Redis.

Os resultados obtidos com os ensaios técnicos podem ser subdivididos em duas categorias: Ensaios realizados em laboratório e em atividades de campo. Ao longo do texto o leitor terá a oportunidade de conhecer as etapas implementadas e os resultados obtidos de cada experimento. De antemão, menciono que as atividades de campo realizadas fazem parte da minha rotina de trabalho e estão intimamente ligadas a projetos de pesquisas do Observatório Nacional. Nenhuma destas campanhas foram realizadas exclusivamente para realizar testes do sistema supervisorio, o sistema foi implantado nas estações de medidas previamente definidas.

Os experimentos foram implementados no laboratório multiusuário Pool de Equipamentos Geofísicos do Brasil - PEGBr¹ sediado no campus do Observatório Nacional. O uso dos sistemas de medidas e os demais dispositivos eletrônicos foram bondosamente compartilhados pela instituição. Os equipamentos se encontram disponíveis à comunidade científica para uso exclusivo em projetos de pesquisa e ensino. Acredito que em breve o sistema supervisorio também fará parte deste acervo científico.

8.1 - Conexão com Nuvem

O servidor hospedeiro do banco de dados Redis foi criado por meio de uma conta gratuita fornecida no site www.Redislab.com. A máquina virtual do RedisLab oferece um espaço de 30MB para contas gratuitas, o que é suficiente para testar o modelo da arquitetura proposta nesta pesquisa. Para criar uma conta gratuita, é necessário fornecer um e-mail válido e definir uma senha. Após o registro, é possível criar um endpoint do banco de dados. As seguintes informações são então configuradas para estabelecer a conexão com o Redis: endereço IP do servidor, porta de acesso e a senha definida pelo usuário. Esses parâmetros são essenciais para garantir uma conexão adequada ao banco de dados Redis hospedado no RedisLab.

¹ <http://www.pegbr.on.br/>

```
Redis = Redis.Redis(host_Redis = IP, port_Redis = porta de acesso, password_Redis =  
senha)
```

```
# Bloco de comunicação com o Redis
```

```
host_Redis="Redis-15032.c8.us-east-1-2.ec2.cloud.Redislabs.com" # STRING
```

```
port_Redis = 15032 # INTEIRO
```

```
password_Redis = "senha" # STRING
```

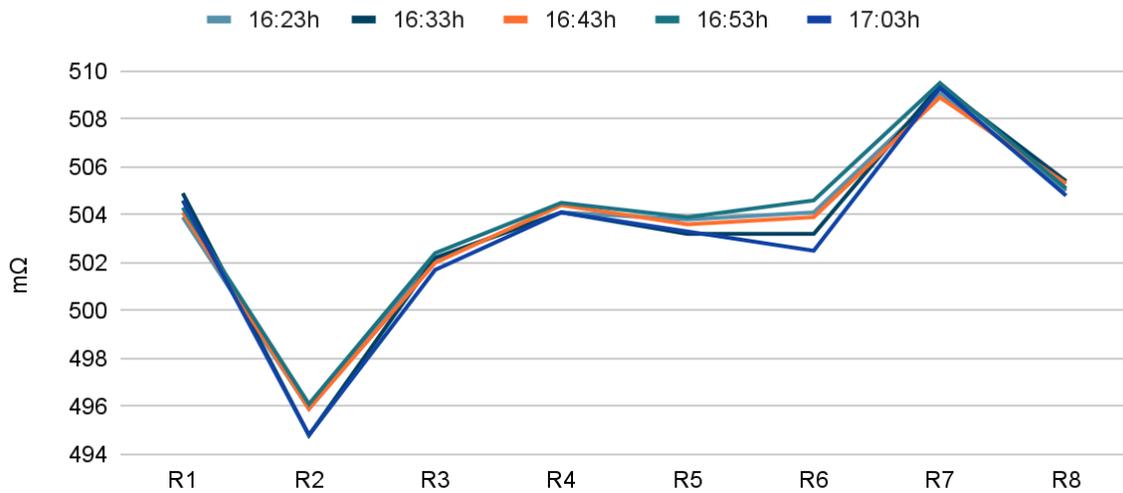
Após a conexão bem-sucedida, os usuários têm a capacidade de criar e acessar estruturas de dados específicas de seu interesse usando o modelo chave/valor no banco de dados Redis hospedado no RedisLab. No sistema supervisorio proposto, as rotinas escritas em Python automatizam a comunicação com o Redis sempre que são inicializadas. Isso implica que, ao iniciar o sistema supervisorio, as rotinas Python automaticamente estabelecem uma conexão com o banco de dados Redis, permitindo um acesso contínuo e dinâmico às informações armazenadas. Essa abordagem automatizada simplifica e agiliza o processo de comunicação e manipulação de dados no contexto do sistema supervisorio.

8.2 - Medidas Realizadas

Os resultados obtidos destacam o funcionamento adequado da arquitetura de comunicação proposta no módulo de comunicação. Os dados adquiridos pelo equipamento geofísico SuperSting R8 foram registrados em tempo real no banco de dados Redis e no diretório da Raspberry, conforme a rotina programada. Para este sistema de medida, a interface de controle foi desenvolvida em Python, e a comunicação foi estabelecida diretamente com o equipamento por meio de comandos ASCII enviados para a porta serial [25]. Para este modelo de aquisição, as rotinas em python envia comandos ASCII pré-definidos na rotina de acordo com o tipo de arranjo solicitado. Neste experimento foi utilizado o arranjo de teste de canais, para verificar o funcionamento do modelo da arquitetura e avaliar a repetibilidade do sistema de medidas.

Na Figura 8.2, é possível observar as cinco primeiras medidas, assim como a última, realizadas em intervalos de 10 minutos. Na Figura 8.2.1, são apresentadas as medidas efetuadas no intervalo de 1 hora. É possível observar que a dinâmica do sistema de medidas permaneceu estável, fornecendo os valores de medidas esperados. Os valores de resistência encontram-se na faixa recomendada pela fabricante, situando-se entre 490 e 515 mΩ. Esses dados referem-se às medidas realizadas na 'Test Box' para a verificação dos canais dos eletrodos. Essas medidas prévias são sumamente importantes para avaliar a confiabilidade do equipamento antes das medidas de campo serem realizadas.

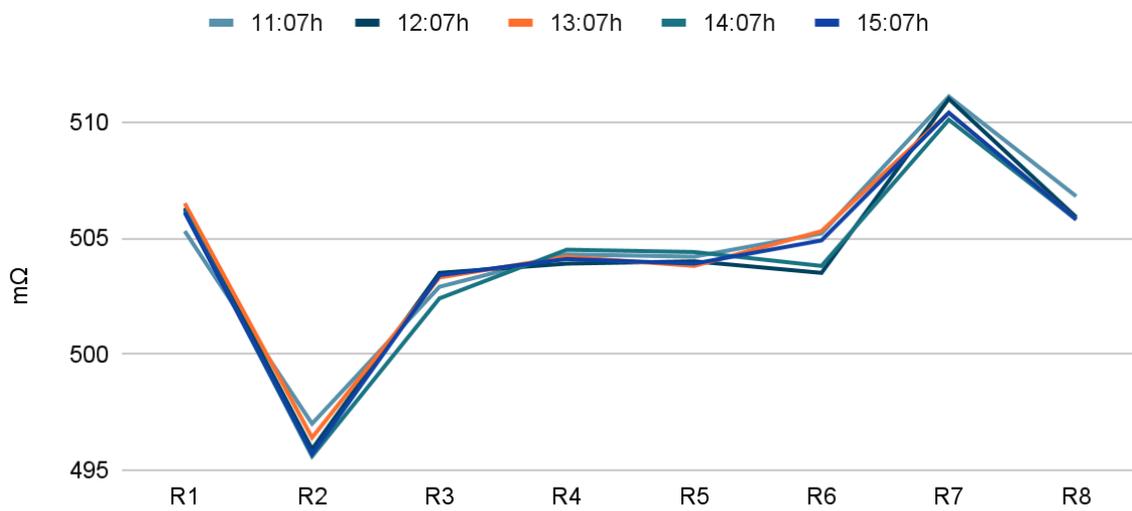
Medidas realizadas em intervalos de 10min



Canais do Supersting

Figura 8.2 - Medidas realizadas em intervalos de 10 min

Medidas realizadas em intervalos de 1h

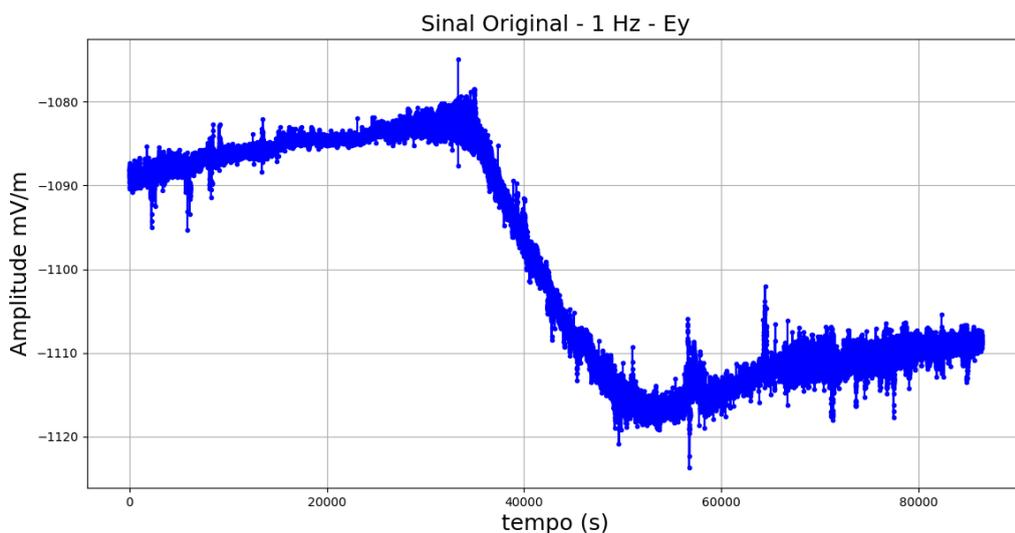
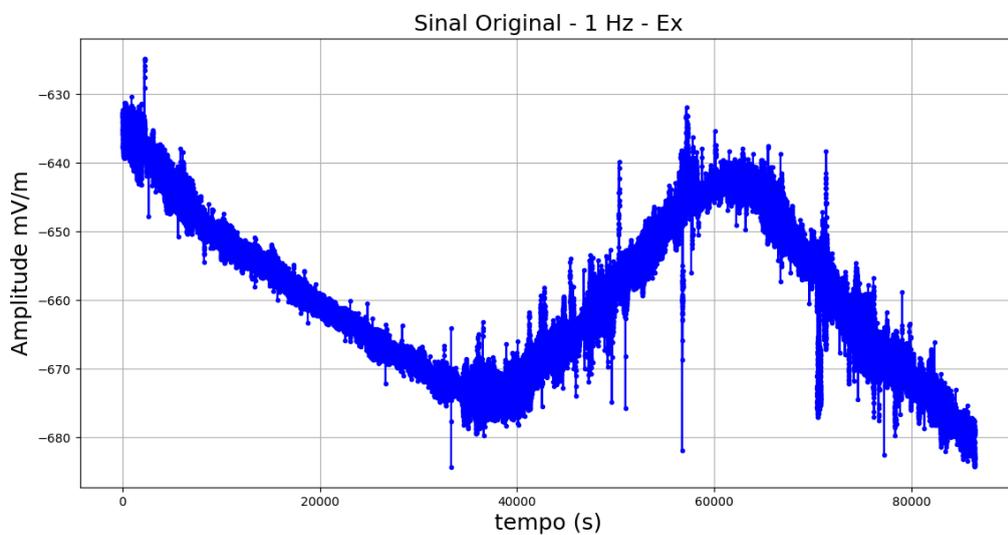


Canais do Supersting

Figura 8.2.1 - Medidas realizadas em intervalos de 1h

As medidas realizadas com o LEMI-424 e armazenadas no banco de dados podem ser visualizadas na Figura 8.2.3. Os valores dos 5 canais registrados são exatamente idênticos aos valores armazenados na memória do equipamento e no armazenamento local da Raspberry. Essas medidas foram realizadas na cidade de Ibertioga-MG durante uma campanha de campo promovida pelo Observatório Nacional, em maio de 2023, como parte do projeto INVESTIGAÇÕES MULTIFÍSICA, financiado pela PETRONAS.

As medidas representam o registro do campo eletromagnético adquirido com a taxa de amostragem de 1Hz. As componentes do campo elétrico são registradas nas direções norte/sul (Ex) e leste/oeste (Ey). Da mesma forma, os campos magnéticos são registrados nas direções norte (Bx), leste (By) e vertical (Bz). Esses dados representam as informações fundamentais do campo eletromagnético que estão sendo monitoradas e registradas para análise no contexto do sistema supervisorio.



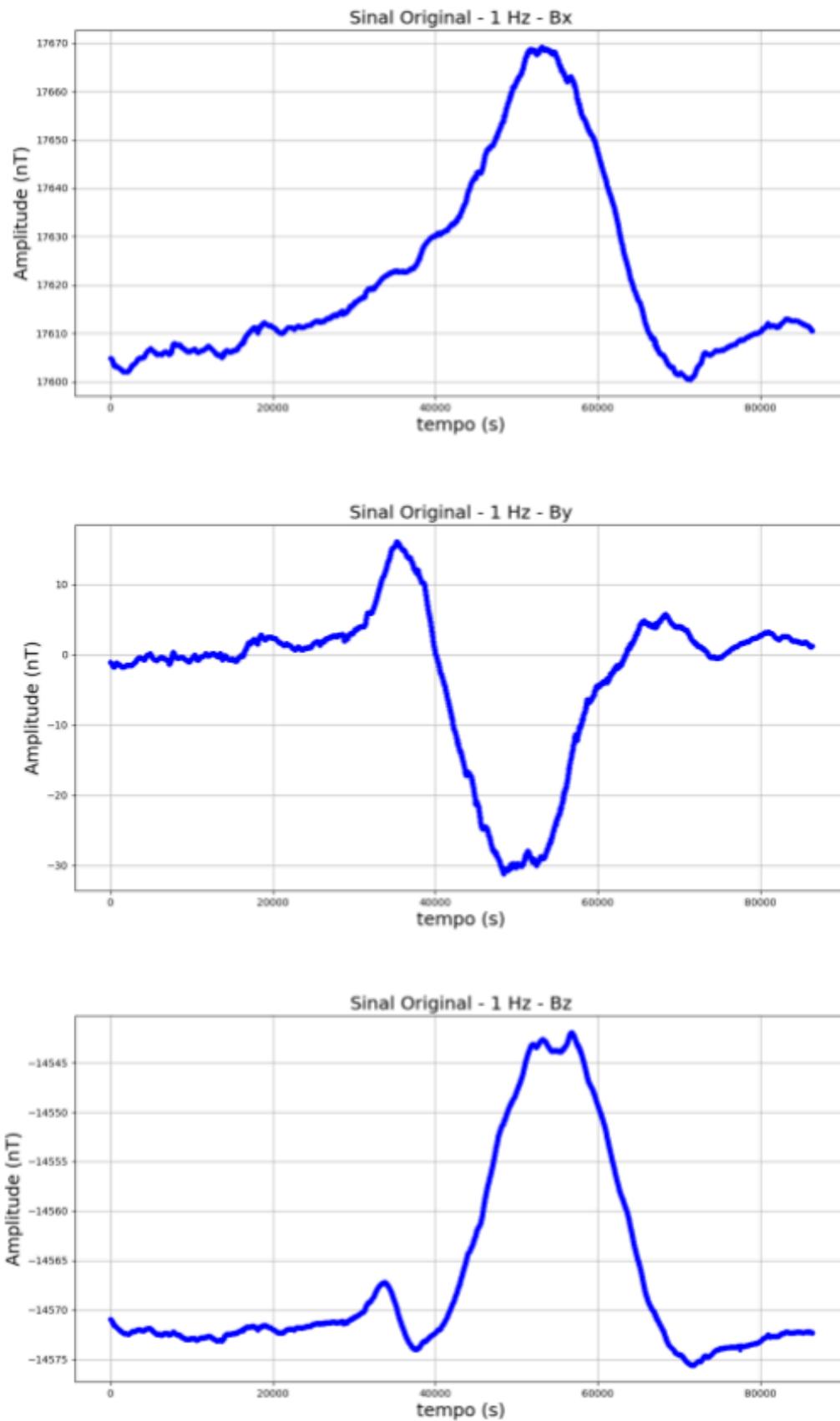


Figura 8.2.3 - Medidas dos campos elétricos e magnéticos realizadas pelo LEMI-424 em 17 de Maio de 2023 na cidade de Ibertioga - MG

9. Transmissão de Dados - Satélite

9.1 Starlink

O Starlink, um empreendimento iniciado em 2015 e associado à SpaceX de Elon Musk, busca proporcionar acesso à internet de alta velocidade e baixa latência em áreas remotas e rurais, bem como melhorar a conectividade em regiões com limitações ou ausência de opções de banda larga convencional. Diferentemente das redes móveis como 3G, 4G e 5G, ou das infraestruturas baseadas em cabo coaxial e fibra ótica, o Starlink não requer uma antena para funcionar. Basta ao usuário instalar a antena em um local aberto, conectá-la ao modem e ter acesso à internet [48].

Essa solução é especialmente relevante para comunidades em regiões rurais que frequentemente enfrentam dificuldades para obter acesso à internet cabeada ou sinal de celular confiável. A principal inovação do Starlink reside na sua constelação de satélites em órbita baixa da Terra, que opera em altitudes entre 340 km e 1.200 km. Esta configuração, com "milhares de satélites" mais próximos da Terra, contribui para reduzir significativamente a latência, uma das principais limitações das tecnologias de comunicação via satélite.

A rede do Starlink continua a se expandir, com a SpaceX lançando regularmente mais satélites. Em julho de 2023, por exemplo, foram adicionadas 48 unidades, totalizando mais de 4.700 satélites em órbita. Esta tecnologia oferece assim acesso à internet de alta velocidade e cobertura global, tornando-se uma solução promissora para áreas remotas em todo o mundo.

O sistema Starlink funciona da seguinte maneira:

Satélites em Órbita Baixa da Terra (LEO): A constelação Starlink é composta por milhares de satélites em órbita baixa da Terra, formando uma rede que cobre grandes áreas do globo terrestre.

Terminal do Usuário: Os usuários do Starlink precisam de uma antena receptora, conhecida como terminal do usuário, que se comunica com os satélites em órbita. Este terminal é relativamente pequeno e fácil de instalar, podendo ser montado em telhados ou em qualquer área aberta com uma visão clara do céu.

Comunicação com Satélites: O terminal do usuário se comunica com os satélites Starlink que estão sobrevoando a área. O sistema utiliza uma tecnologia de feixes direcionais para estabelecer uma conexão de alta velocidade e baixa latência.

Backbone de Internet: Os satélites Starlink estão conectados a uma rede terrestre de servidores, que por sua vez estão interligados à infraestrutura de internet global. Isso permite que os usuários do Starlink acessem conteúdo na internet da mesma forma que fariam com qualquer outra conexão de banda larga [17].

O Starlink tem o potencial de ampliar o acesso à internet em várias partes do mundo, especialmente em áreas remotas onde a infraestrutura de comunicação é limitada ou inexistente, proporcionando conectividade de alta velocidade e permitindo o acesso a informações e serviços online. Em situações de desastres naturais ou emergências, o Starlink pode ser uma alternativa confiável e rápida quando as redes terrestres falham, ajudando equipes de resgate e comunidades afetadas. Em áreas rurais com poucas opções de internet, ele pode promover o desenvolvimento econômico ao facilitar o acesso à educação, trabalho

remoto e outros serviços online [42]. No entanto, o projeto enfrenta desafios como preocupações ambientais com o aumento de lixo espacial e questões regulatórias sobre o uso do espectro para comunicações via satélite.

9.2 Aplicação do Sistema de Monitoramento

Neste estudo, realizamos uma avaliação do sistema de monitoramento empregando a conectividade do Starlink, pertencente ao Pool de Equipamentos Geofísicos do Brasil (PEGBr). O propósito central foi a integração de diferentes sistemas de medidas geofísicas por meio do módulo de comunicação proposto. Em um ambiente controlado de laboratório, testamos a conectividade simultânea de quatro sistemas de medidas geofísicas: Sismógrafos Quanterra Q330S+, Resistivímetro SuperSting R8, ADU07e, ADU08e e LEMI-424. Os resultados dos testes de comunicação ratificaram a viabilidade e estabilidade do sistema ao estabelecer comunicação com estes sistemas conforme a figura 9.2.1.

A Figura 9.2.2 ilustra o arranjo empregado para criar a rede de monitoramento. Nele, os sistemas SuperSting R8 e LEMI-424 conectam-se à Raspberry Pi 3 Model B através da comunicação USB, o que viabiliza a interação individual com os equipamentos ao selecionar a porta serial correspondente. Já os sistemas de medição restantes, ADU07e, ADU08e e Quanterra Q330S+, são identificados pelo sistema mediante os endereços IP definidos na intranet da rede local. Para concretizar esse intento, a utilização de um roteador foi crucial para interligar os sistemas, permitindo o acesso através da Raspberry Pi.

A Raspberry Pi se conecta à internet/starlink por meio da VLAN - Wifi. Essa configuração foi implementada para assegurar que mudanças no IP da Starlink não afetem a conectividade e o acesso aos sistemas de medição geofísica, a mudança de IP da Starlink não garante que os IP fixos dos equipamentos geofísicos estarão na mesma faixa de rede, tal mudança implicaria na perda de comunicação entre o sistema e os equipamentos geofísicos.

Após os testes bem-sucedidos em laboratório, o sistema foi implantado na estação remota temporária, localizada na região de Unamar-RJ, março de 2024. Nesta ocasião foram utilizados apenas os sistemas magnetotelúricos LEMI-424 e ADU08e, Figura 9.2.3. O período de teste foi desafiador devido às fortes chuvas e à falta de incidência solar, fatores cruciais para a manutenção do sistema de alimentação. Durante esse período, observou-se uma boa estabilidade do sistema, conectividade consistente com o banco de dados, baixa latência, alta velocidade e eficiência na transferência de arquivos através do VNC Viewer conforme a Figura 9.2.4.

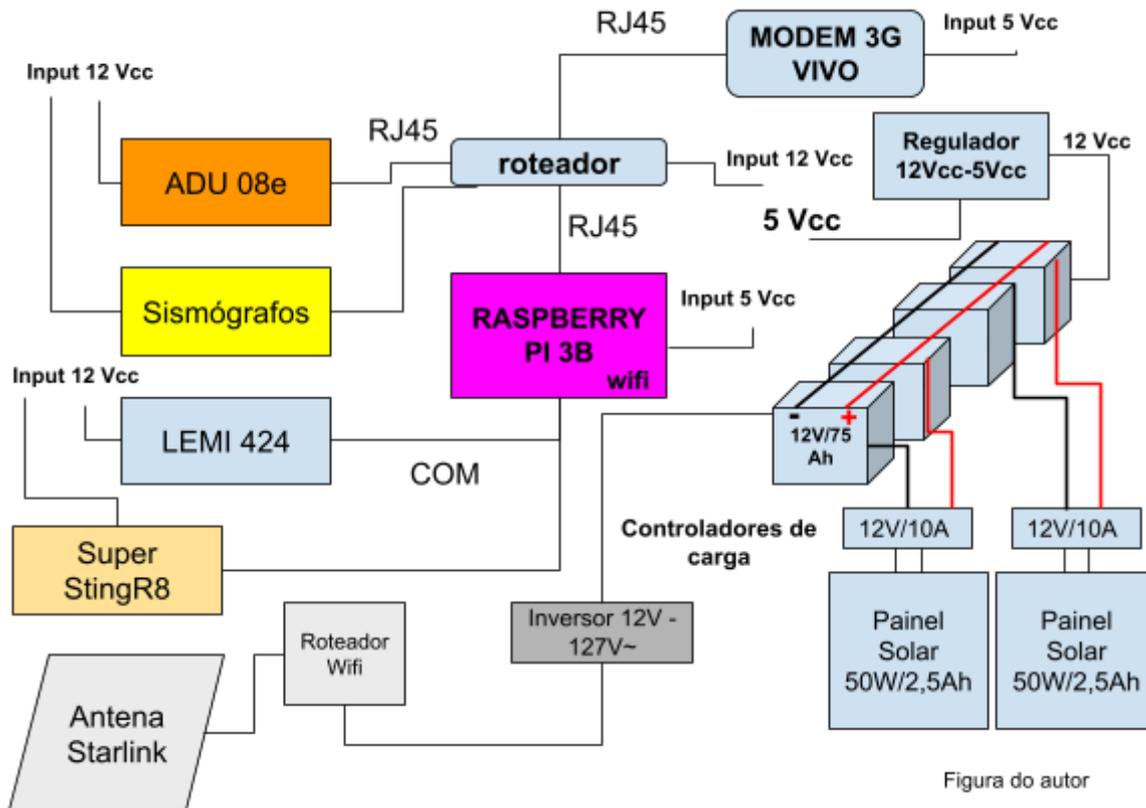


Figura 9.2.2 - Fluxograma do sistema de monitoramento utilizado em laboratório



Figura 9.2.3 - Instalação do sistema supervisorio na estação remota março 2024

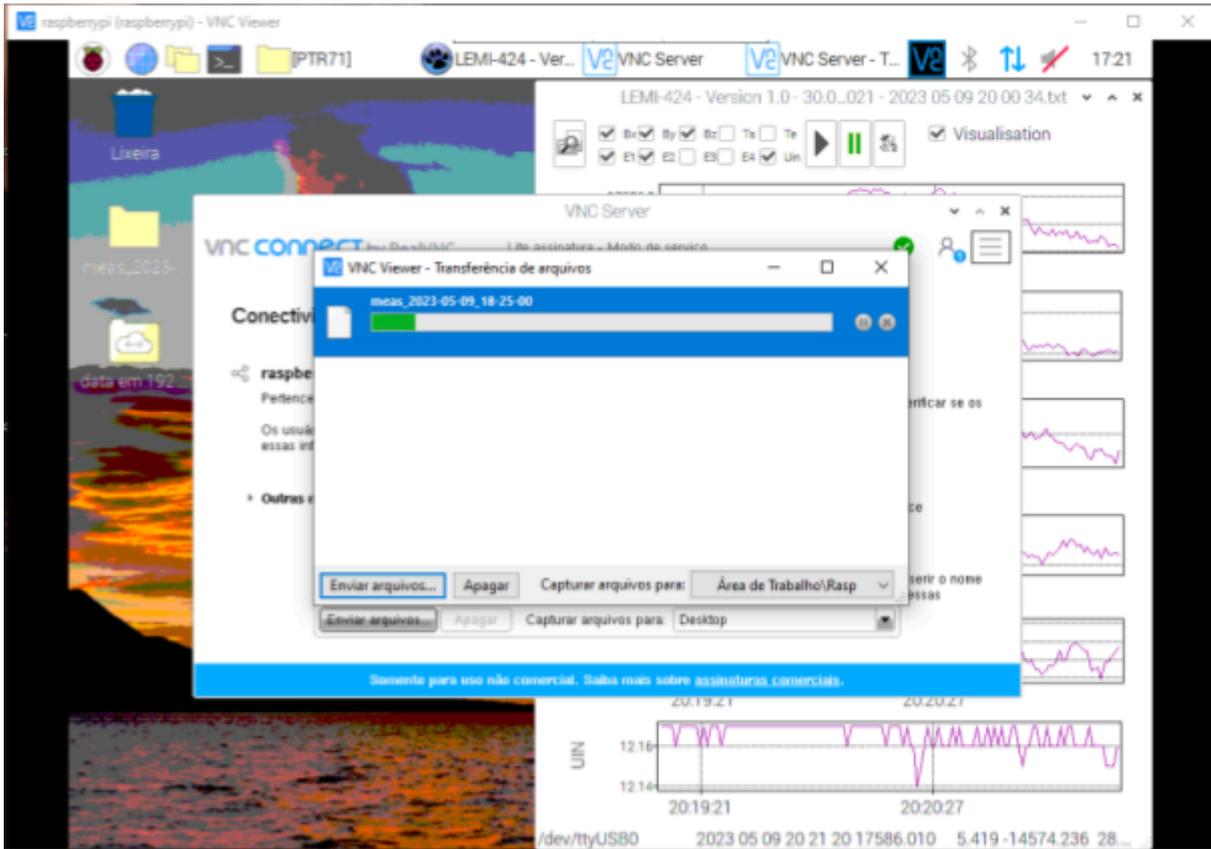


Figura 9.2.4 - Transferência de arquivos através do VNC utilizando a Starlink

10. Conclusão

O sistema supervisorio desenvolvido neste projeto representa uma inovação marcante na gestão e operação de equipamentos empregados em pesquisas magnetotéluricas e eletroresistividade, focado no monitoramento e aquisição de dados geofísicos. Construído em software aberto e fundamentado em microprocessadores, o sistema vai além do simples monitoramento contínuo dos sistemas de medidas, possibilitando reconfiguração remota, armazenamento eficiente em nuvem e transmissão de dados em tempo real para vários usuários simultaneamente.

A arquitetura do módulo de comunicação é notável, destacando-se pela sua estrutura de dados chave/valor implementada no banco de dados NoSQL Redis. Essa configuração prova ser uma solução eficaz para o monitoramento em tempo contínuo. A estrutura de banco de dados em memória de código aberto, Redis, adequado para aplicações de tempo real, baixa latência, facilita a conectividade eficiente entre os equipamentos geofísicos, o microprocessador e o servidor hospedeiro do banco de dados.

A capacidade de estabelecer conectividade com os equipamentos geofísicos através da porta serial ou por interface de rede, armazenar dados localmente e, em seguida, aloca-los em chaves no Redis, cria um ambiente dinâmico onde múltiplos usuários podem interagir, enviando comandos, consultando status e acessando simultaneamente os dados adquiridos pelos sistemas de medidas instalados em campo.

O protótipo foi submetido a testes bem-sucedidos em três etapas do projeto de pesquisa conduzido pelo Observatório Nacional. Esses testes ocorreram entre maio e julho de 2023 em Ibiritoga-MG, entre setembro e novembro do mesmo ano em Lagoa Dourada-MG e em março de 2024 em Unamar-RJ como parte do projeto INVESTIGAÇÕES MULTIFÍSICAS, financiado pela PETRONAS. Durante essas fases, o sistema demonstrou eficácia na supervisão remota das estações geofísicas. Esta conquista não só contribui para a redução de custos operacionais, otimizando os recursos financeiros em campanhas de aquisição, mas também apresenta uma abordagem inovadora para aprimorar a qualidade e continuidade dos dados adquiridos. No âmbito técnico-científico, o sistema oferece uma gestão mais eficiente dos equipamentos, marcando um passo significativo para o futuro da aquisição de dados geofísicos.

Diante do exposto, considero que há viabilidade técnica e operacional para instalar o sistema supervisorio em outras estações interessadas no monitoramento contínuo de dados científicos. Especificamente, a aplicação do sistema supervisorio em estações de monitoramento de dados multifísicos, observatórios magnéticos, estações sismológicas instaladas em ilhas e em locais onde a conexão com a internet é possível apenas por meio de sistemas de satélites.

Referências

1. FERNÁNDEZ-PACHECO, A.; MARTIN, S.; CASTRO, M. Implementation of an Arduino remote laboratory with Raspberry Pi. In: Proc. IEEE Global Eng. Educ. Conf. (EDUCON), 2019, pp. 1415–1418.
2. ADVANCED GEOSCIENCES, INC. The SuperSting with Swift automatic resistivity and IP system Instruction Manual. Austin: Advanced Geosciences, Inc., 2005.
3. BOYLESTAD, R. L.; NASHELSKY, L. Electronic Devices and Circuit Theory. 11th ed. Upper Saddle River: Prentice Hall, 2012.
4. BROWN, S.; VRANESIC, Z. Fundamentals of Digital Logic with Verilog Design. 3rd ed. New York: McGraw-Hill, 2014.
5. BUCHLA, D. M.; FLOYD, T. Fundamentals of Analog Circuits. 2nd ed. Upper Saddle River: Pearson, 2012.
6. BUYYA, R.; DASTJERDI, A. V.; GHOSH, S. K. Internet of Things: Principles and Paradigms. Burlington: Morgan Kaufmann, 2016.
7. CAGNIARD, L. Basic theory of the magnetotelluric method. Geophysics, v. 8, pp. 605-635, 1953.
8. CHAMBERS, J. E. et al. Electrical resistivity tomography and its application to geotechnical investigations. Geological Society, London, Engineering Geology Special Publications, 2006.
9. DORF, R. C.; SVOBODA, J. A. Introduction to Electric Circuits. 9th ed. Hoboken: Wiley, 2018.
10. FERGUSON, I. J.; LOUDEN, K. E. A review of the principles and applications of the magnetotelluric method. Geophysics, 1973.
11. FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. Feedback Control of Dynamic Systems. 7th ed. Upper Saddle River: Pearson, 2015.

12. HARRIS, D. M.; HARRIS, S. L. Digital Design and Computer Architecture. 2nd ed. Cambridge: Morgan Kaufmann, 2012.
13. HART, D. Electronic Principles: Physics, Models, and Circuits. 8th ed. Stamford: Cengage Learning, 2019.
14. CARLSON, J. Redis in Action. Greenwich: Manning Publications, 2013.
15. SARIK, J.; KYMISSIS, I. Lab kits using the Arduino prototyping platform. In: Proc. IEEE Frontiers Educ. Conf. (FIE), 2010, pp. T3C-1–T3C-5.
16. JONES, A. G.; FERGUSON, I. J. The Magnetotelluric Method: Theory and Practice. Cambridge: Cambridge University Press, 2018.
17. KASSEM, M. M. et al. A browser-side view of Starlink connectivity. In: Proceedings of the 22nd ACM Internet Measurement Conference, 2022, pp. 151–158.
18. KEAREY, P. et al. An Introduction to Geophysical Exploration. 3rd ed. Wiley-Blackwell, 2013.
19. KRANZ, M. Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry. Indianapolis: Wiley, 2016.
20. KUROSE, J. F.; ROSS, K. W. Redes de Computadores e a Internet: Uma Abordagem Top-Down. 8th ed. Bookman, 2021.
21. LATHI, B. P. Sinais e Sistemas Lineares. 2nd ed. Bookman, 2006.
22. LUTZ, M. Learning Python: Powerful Object-Oriented Programming. O'Reilly Media, 2013.
23. MAKSIMOVIĆ, M. et al. Raspberry Pi as Internet of Things hardware: Performances and constraints. In: Proc. 1st Int. Conf. Elect., Electron. Comput. Eng., 2014, vol. 3, no. 8, pp. 1–6.

24. MALIK, S. U. R. et al. Internet of Things: A Review of Surveys Based on Context Aware Intelligent Services. In: Journal of Network and Computer Applications, v. 97, p. 199-221, 2017.
25. MCKINNEY, W. Python Para Análise de Dados: Tratamento de Dados com Pandas, NumPy e IPython. Novatec Editora, 2018.
26. OGATA, K. Engenharia de Controle Moderno. 4th ed. Pearson, 2010.
27. OKWUEZE, E. C. et al. A review of magnetotelluric method and its applications to petroleum exploration in Nigeria. Journal of Petroleum & Gas Exploration Research, 2015.
28. JAMIESON, P.; HERDTNER, J. More missing the Boat—Arduino, Raspberry Pi, and small prototyping boards and engineering education needs them. In: Proc. IEEE Frontiers Educ. Conf. (FIE), 2015, pp. 1–6.
29. ORDUÑA, P. et al. WebLabLib: New approach for creating remote laboratories. In: AUER, M. E.; RAM, B. K. (Eds.). Cyber-Physical Systems and Digital Twins. Cham: Springer, 2020, pp. 477–488.
30. Python.org. Disponível em: <<https://www.python.org/>>. Acesso em: 29 jan. 2024.
31. RABAEY, J. M.; CHANDRAKASAN, A.; NIKOLIC, B. Digital Integrated Circuits: A Design Perspective. 2nd ed. Upper Saddle River: Pearson, 2003.
32. Raspberry Pi Documentation. Disponível em: <<https://www.raspberrypi.com/documentation/>>. Acesso em: 29 set. 2023.
33. RealVNC. Disponível em: <<https://www.realvnc.com/en/>>. Acesso em: 04 fev. 2024.
34. Redis Explained. Disponível em: <<https://architecturenotes.co/Redis/>>. Acesso em: 21 set. 2022.
35. Redis Labs. Disponível em: <<https://Redis.com/>>. Acesso em: 29 jan. 2024.

36. ROWLAND, C. et al. *Designing Connected Products: UX for the Consumer Internet of Things*. Sebastopol: O'Reilly Media, 2015.
37. SCHWAB, K. *The Fourth Industrial Revolution*. New York: Crown Business, 2016.
38. SEBORG, D. E. et al. *Process Dynamics and Control*. 4th ed. Hoboken: Wiley, 2010.
39. CHEN, S. et al. Towards scalable and reliable in-memory storage system: A case study with Redis. In: *Proc. IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 1660–1667.
40. WERNER, S. et al. Cloud-based remote virtual prototyping platform for embedded control applications: Cloud-based infrastructure for large-scale embedded hardware-related programming laboratories. In: *Proc. 13th Int. Conf. Remote Eng. Virtual Instrum. (REV)*, 2016, pp. 168–175.
41. SEDRA, A. S.; SMITH, K. C. *Microelectronic Circuits*. 7th ed. Oxford: Oxford University Press, 2015.
42. SHAENGCHART, Y.; KRAIWANIT, T. Starlink satellite project impact on the Internet provider service in emerging economies. *Research in Globalization*, v. 6, p. 100132, 2023.
43. SILBERSCHATZ, A. et al. *Fundamentos de Sistemas de Bancos de Dados*. 6th ed. LTC, 2011.
44. SIMPSON, F.; BAHR, K. *Practical Magnetotellurics*. Cambridge: Cambridge University Press, 2005.
45. SINCLAIR, B. *IoT Inc: How Your Company Can Use the Internet of Things to Win in the Outcome Economy*. Cupertino: Independent Publishing Platform, 2016.
46. SKIENA, S. J.; DASGUPTA, A. *Introduction to Feedback Control*. New York: Cambridge University Press, 2018.
47. STENERSON, J. *Industrial Automation and Process Control*. Upper Saddle River: Pearson, 2002.

48. TANVEER, H. B. et al. Making Sense of Constellations: Methodologies for Understanding Starlink's Scheduling Algorithms. In: Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies, 2023, pp. 37–43.
49. TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. Digital Systems: Principles and Applications. 11th ed. Upper Saddle River: Pearson, 2020.
50. TOKHEIM, R. L. Digital Electronics: Principles and Applications. 8th ed. New York: McGraw-Hill, 2017.
51. TOKHEIM, R. L. Electronics: Analog and Digital. 4th ed. New York: McGraw-Hill, 2014.
52. VILLAR-MARTÍNEZ, A. et al. Improving the Scalability and Replicability of Embedded Systems Remote Laboratories. IEEE ACCESS, v. 7, pp. 164164-164185, 2019.
53. WAKERLY, J. F. Digital Design: Principles and Practices. 4th ed. Upper Saddle River: Pearson, 2006.
54. WU, H. et al. Internet of Things for Smart Cities. IEEE Internet of Things Journal, v. 1, n. 1, pp. 22-32, 2014.
55. WONHAM, W. M.; CAI, K. Supervisory Control of Discrete-Event Systems. Springer Cham, 2019.