

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

Sistema de telemetria remota para emissoras de televisão com base no protocolo SNMP

Autor: Leonardo de Souza Takehana
Orientador: Dr. Diogo Caetano Garcia

Brasília, DF
2023



Leonardo de Souza Takehana

Sistema de telemetria remota para emissoras de televisão com base no protocolo SNMP

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

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Dr. Diogo Caetano Garcia

Brasília, DF

2023

Leonardo de Souza Takehana

Sistema de telemetria remota para emissoras de televisão com base no protocolo SNMP/ Leonardo de Souza Takehana. – Brasília, DF, 2023-
56 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Diogo Caetano Garcia

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2023.

1. Telemetria. 2. SNMP. I. Dr. Diogo Caetano Garcia. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Sistema de telemetria remota para emissoras de televisão com base no protocolo SNMP

CDU 02:141:005.6

Leonardo de Souza Takehana

Sistema de telemetria remota para emissoras de televisão com base no protocolo SNMP

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

Trabalho aprovado. Brasília, DF, 25 de Julho de 2023:

Dr. Diogo Caetano Garcia
Orientador

Dr. Daniel Costa Araújo
Convidado 1

Dr. Leonardo Aguayo
Convidado 2

Brasília, DF
2023

Agradecimentos

Eu gostaria de agradecer a toda minha família e amigos, em especial aos meus pais, Nelson e Joelma e minha namorada Luara por todo o apoio incondicional durante minha graduação.

A equipe da TV Oeste, que me ajudou na realização dos testes, em especial aos técnicos do setor de engenharia e ao coordenador Paulino por todo o apoio na realização desse trabalho.

E aos professores do curso de engenharia eletrônica da Universidade de Brasília, em especial ao meu orientador Prof. Dr. Diogo Caetano Garcia, pois sem eles nada disso seria possível.

Resumo

Este trabalho relata o desenvolvimento de um sistema de telemetria para estações de retransmissão de TV, com base no protocolo SNMP, para aferir dados diretamente dos transmissores, com o intuito de monitorar remotamente informações sobre a transmissão e evitar deslocamentos desnecessários, além de identificar possíveis falhas. O desenvolvimento foi feito utilizando um microcomputador Raspberry Pi conectado a outros equipamentos, como um Arduino, um *modem* 3G e um módulo de relés, a fim de realizar a medição de dados analógicos e digitais e enviá-los para um servidor central, através de códigos utilizando as linguagens de programação C, Python, Javascript e SQL. O sistema desenvolvido também é capaz de obter imagens do local e transmiti-las ao usuário, além de receber comandos para alterar parâmetros do ambiente. O sistema foi testado em uma estação de retransmissão real, monitorando por cerca de uma semana obtendo dados reais do local e obteve resultados satisfatórios onde os dados foram coletados na estação e exibidos para serem acompanhados na emissora.

Palavras-chave: Telemetria. SNMP. Emissoras de Televisão. RTV. Retransmissoras. Sistemas Embarcados.

Abstract

This work reports the development of a telemetry system for TV retransmission stations, based on the SNMP protocol, to measure data directly from the transmitters, in order to remotely monitor information about the transmission and avoid unnecessary displacements, in addition to identifying possible failures. The development was done using a Raspberry Pi microcomputer connected to other equipment, such as an Arduino, a 3G modem and a relay module, in order to measure analog and digital data and send them to a central server, through codes using the programming languages C, Python, Javascript and SQL. The developed system is also capable of obtaining images of the location and transmitting them to the user, in addition to receiving commands to change environment parameters. The system was tested in a real relay station, monitoring for about a week obtaining real data from the site and obtained satisfactory results where the data were collected at the station and displayed to be followed at the station.

Key-words: Telemetry. SNMP. Television Broadcasters. RTV. Relays. Embedded systems.

Lista de ilustrações

| | |
|---|----|
| Figura 1 – Ilustração representativa de funcionamento da transmissão do sinal de TV. | 25 |
| Figura 2 – Diagrama representativo de funcionamento de uma retransmissora (SILVA, 2022). | 25 |
| Figura 3 – Diagrama de comunicação de um sistema de telemetria. Fonte: (SILVA, 2022) | 26 |
| Figura 4 – Árvore de identificadores de objetos ASN.1. Fonte (KUROSE; ROSS, 2014) | 28 |
| Figura 5 – Raspberry Pi 3 Model B+. Fonte (PI, 2018) | 29 |
| Figura 6 – Arduino Nano. Fonte: (ARDUINO, 2008) | 30 |
| Figura 7 – Diagrama de blocos do sistema proposto. | 33 |
| Figura 8 – Esquemático placa multiplexadora. | 37 |
| Figura 9 – Vista superior do gabinete desenvolvido. | 38 |
| Figura 10 – Vista frontal do gabinete desenvolvido. | 38 |
| Figura 11 – Vista traseira do gabinete desenvolvido. | 38 |
| Figura 12 – Vista isométrica do gabinete desenvolvido. | 38 |
| Figura 13 – Protótipo finalizado. | 39 |
| Figura 14 – Exterior do abrigo. | 41 |
| Figura 15 – Interior do abrigo. | 42 |
| Figura 16 – Estação de telemetria instalada no abrigo. | 43 |
| Figura 17 – Sensor de Presença, sensor de temperatura e webcam instalados. | 44 |
| Figura 18 – <i>Switch</i> de rede e módulo de <i>reset</i> instalados. | 44 |
| Figura 19 – Tela de login. | 45 |
| Figura 20 – Tela de entradas digitais. | 46 |
| Figura 21 – Tela de entradas analógicas. | 46 |
| Figura 22 – Tela de dados SNMP. | 47 |
| Figura 23 – Tela de envio de comandos. | 47 |
| Figura 24 – Tela de espelho da <i>webcam</i> | 48 |
| Figura 25 – Gráfico da variação da potencia transmitida em função do tempo entre a noite do dia 8 de julho de 2023 e a manhã do dia 9 de julho de 2023. | 50 |
| Figura 26 – Gráfico do sensor de temperatura em função do tempo no dia 12/07/2023. | 50 |
| Figura 27 – Gráfico do sensor de umidade em função do tempo no dia 12/07/2023. | 51 |
| Figura 28 – Gráfico do sensor de presença em função do tempo no dia 10/07/2023. | 52 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 – Lista de materiais | 34 |
| Tabela 2 – Conexões Raspberry | 35 |
| Tabela 3 – Conexões Arduino | 35 |
| Tabela 4 – Tabela de OID SNMP | 49 |

Lista de abreviaturas e siglas

| | |
|--------|--|
| 3G | Terceira geração de redes móveis. |
| API | <i>Application Programming Interface.</i> |
| Anatel | Agência Nacional de Telecomunicações. |
| ARMv8 | <i>Advanced RISC Machine</i> versão 8. |
| ASI | <i>Asynchronous Serial Interface.</i> |
| DVB-S2 | <i>Digital Video Broadcasting - Satellite 2.</i> |
| FPGA | <i>Field-Programmable Gate Array.</i> |
| HTTP | <i>Hyper-Text Transfer Protocol.</i> |
| I2C | <i>Inter-Integrated Circuit.</i> |
| IANA | <i>Internet Assigned Numbers Authority.</i> |
| IBGE | Instituto Brasileiro de Geografia e Estatística. |
| IP | <i>Internet Protocol.</i> |
| ISDB-T | <i>Integrated Services Digital Broadcasting Terrestrial.</i> |
| LPDDR2 | <i>Low-Power Double Data Rate 2 .</i> |
| MIB | <i>Management Information Base.</i> |
| MPEG-2 | <i>Moving Picture Experts Group 2</i> - Formato de codificação de vídeo. |
| MPEG-4 | <i>Moving Picture Experts Group 4</i> - Formato de codificação de vídeo. |
| OID | <i>Object Identifier.</i> |
| PIR | <i>Passive Infra Red</i> - Sensor utilizado para medição de presença. |
| PWM | <i>Pulse Width Modulation.</i> |
| REST | <i>Representational State Transfer.</i> |
| SD | <i>Secure Digital.</i> |
| SDI | <i>Serial Digital Interface.</i> |

| | |
|-------|--|
| SDRAM | <i>Synchronous dynamic random-access memory.</i> |
| SNMP | <i>Simple Network Management Protocol.</i> |
| SPI | <i>Serial Peripheral Interface.</i> |
| SQL | <i>Structured Query Language.</i> |
| SSH | <i>Secure Socket Shell.</i> |
| UART | <i>Universal Synchronous Receiver/Transmitter.</i> |
| VPN | <i>Virtual Private Network.</i> |

Sumário

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 19 |
| 1.1 | Contexto | 19 |
| 1.2 | Justificativa | 20 |
| 1.3 | Objetivos | 20 |
| 1.3.1 | Objetivo Geral | 20 |
| 1.3.2 | Objetivos Específicos | 21 |
| 1.4 | Estrutura do Documento | 21 |
| 2 | REVISÃO BIBLIOGRÁFICA | 23 |
| 2.1 | Sistemas Embarcados | 23 |
| 2.2 | Transmissão do Sinal de Televisão | 24 |
| 2.3 | Telemetria | 25 |
| 2.4 | SNMP | 26 |
| 3 | METODOLOGIA PROPOSTA | 29 |
| 3.1 | Materiais e Conceito-chave | 29 |
| 3.1.1 | Raspberry Pi | 29 |
| 3.1.2 | Arduino | 30 |
| 3.1.3 | Comunicação 3G | 30 |
| 3.1.4 | Bancos de Dados | 31 |
| 3.1.5 | SSH | 31 |
| 3.1.6 | VPN | 31 |
| 3.1.7 | <i>Back-end</i> | 31 |
| 3.1.8 | <i>Front-end</i> | 32 |
| 3.2 | Sistema Proposto | 32 |
| 3.2.1 | Hardware | 32 |
| 3.2.2 | <i>Software</i> | 39 |
| 3.3 | Ambiente de Teste | 40 |
| 4 | RESULTADOS EXPERIMENTAIS | 45 |
| 4.1 | Apresentação dos Dados | 45 |
| 4.2 | Monitoramento Via Protocolo SNMP | 47 |
| 4.3 | Monitoramento de Temperatura e Umidade | 49 |
| 4.4 | Monitoramento de Entradas Analógicas | 51 |
| 4.5 | Monitoramento de Entradas Digitais | 51 |
| 4.6 | Envio de Comandos | 51 |

| | | |
|-----|--|-----------|
| 4.7 | Conexão e Envio dos Dados | 52 |
| 5 | CONCLUSÕES E PASSOS FUTUROS | 53 |
| 5.1 | Considerações Finais | 53 |
| 5.2 | Passos Futuros | 53 |
| | REFERÊNCIAS | 55 |

1 Introdução

Este Capítulo busca contextualizar o sistema proposto, trazendo as justificativas e objetivos que embasaram o desenvolvimento do texto, além de demonstrar a estrutura utilizada no mesmo.

1.1 Contexto

O sistema comercial brasileiro de televisão opera com grande sucesso a mais de 50 anos. Mesmo com o surgimento do *streaming*, os canais de televisão ainda continuam sendo um dos principais meios de comunicação no Brasil. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), a porcentagem de casas brasileiras que possuíam esse equipamento em 2021 era de 95,5% ([Instituto Brasileiro de Geografia e Estatística \(IBGE\), 2022](#)), levando informações para a maior parte dos municípios, com notícias, esportes e entretenimento, que muitas vezes chegam a locais mais remotos onde não há internet. Além disso, 91,6% das casas com televisão, possuíam conversores de sinal de televisão digital aberta.

Para que o sinal alcance grandes distâncias, é necessário uma estrutura robusta, capaz de receber o sinal de TV via satélite e retransmiti-lo através de uma torre local. Todo o aparato para realizar esse processo geralmente se encontra em pequenas cabanas, localizados em pontos próximos a torres de transmissão. Dentro destes locais se encontram receptores que captam o sinal de satélite e depois repassam o sinal de vídeo para transmissores de televisão digital, contando também com outros periféricos necessários para o funcionamento, como ares-condicionados para manter a temperatura de funcionamento, *no-breaks* para falhas de energia, geradores de energia elétrica, dentre outros. Estes locais são comumente referidos como abrigos de retransmissão ou simplesmente retransmissoras.

Após a digitalização do sinal de TV, que teve início em 2007 ([Agência Nacional de Telecomunicações \(Anatel\), 2015](#)), os equipamentos das estações de retransmissão se tornaram bem mais complexos, pois o processo de modulação de sinal foi completamente renovado. Para isso, os transmissores mais novos começaram a codificar o sinal de áudio e vídeo em sinais discretos, utilizando técnicas de compressão digital de vídeo. Esses transmissores contam também com sistema embarcados que permitem a gerência e a configuração desses sistemas. Em equipamentos mais completos, é possível acessar esses dados via protocolo de rede, para configuração do equipamento, além de uma série de dados que podem ser obtidos pelo protocolo SNMP (Simple Network Management Protocol), como potência direta, potência refletida, temperatura do equipamento, dentre outros. Com isso, é possível manuseá-los de uma maneira mais rápida e intuitiva, conectando-os a um com-

putador, diferente dos transmissores analógicos, que muitas vezes eram configurados com a ajuda de módulos específicos, ajustes em potenciômetros, ou *jumpers* entre conectores.

1.2 Justificativa

Os transmissores de TV digital, apesar de modernos, ainda são bastante sensíveis a variáveis do ambiente, como temperatura do abrigo, qualidade da rede elétrica e umidade relativa do ar, podendo causar danos aos equipamentos. Outro fator importante é que, por conta de sua operação constante por longos períodos, muitas vezes esses dispositivos podem apresentar erros ou travarem, sendo necessário reiniciá-los para que voltem. Além disso, os equipamentos também estão sujeitos a furtos, seja do transmissor, receptor, ou até de um ar-condicionado.

Todos esses fatores apresentam riscos à qualidade do sinal, o que em caso de falha podem gerar prejuízo monetário, visto que anúncios e demais publicidades podem deixar de ser exibidas. Muitas vezes, quando isso ocorre é necessário enviar uma equipe de técnicos ao local, que podem demorar dias para chegar, pois dependendo da distância do local para a central da emissora, o diagnóstico do problema pode ser dificultado. Além disso, essas viagens para reparo geram grandes custos, o que por muitas vezes consome uma grande parcela do orçamento de manutenção da emissora, sendo que às vezes é necessário apenas reiniciar algum equipamento (PISKA; PEREIRA; NASCIMENTO, 2018).

Com o intuito de auxiliar na redução dos custos com manutenções e viagens desnecessárias, e para diagnosticar de maneira ágil os problemas de transmissão de sinal, esse trabalho se baseia no desenvolvimento de uma estação de telemetria utilizando o mini computador Raspberry Pi 3 modelo B+, para monitorar dados de uma retransmissora de TV utilizando a transmissão via protocolo SNMP. Com base no exposto, foram definidos os objetivos gerais e específicos do projeto.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolvimento de um sistema embarcado para a telemetria de uma estação de retransmissão, para minimizar custos com deslocamento, medindo os dados do sistema e do ambiente, podendo receber comandos e acionar relés para reiniciar equipamentos e exibir alarmes para uma central em caso de algum problema no abrigo, e enviando e recebendo dados através da internet.

1.3.2 Objetivos Específicos

- Medir dados de transmissores e receptores através do protocolo SNMP;
- Medir dados de sensores analógicos e digitais como temperatura, umidade, presença no ambiente, tensão da rede elétrica, dentre outros que o usuário desejar;
- Armazenar dados medidos num sistema de banco de dados;
- Reiniciar equipamentos através de relés;
- Enviar os dados coletados para uma central através de internet 3G;
- Receber comandos a partir da central através de internet 3G;
- Retirar fotos do local e envia-las a central para monitoramento do local;
- Exibir informações medidas em uma interface gráfica acessada via navegador;
- Permitir que o usuário adicione sensores ou modifique informações a serem obtidas;
- Exibir alarmes em caso de falha em algum dos sistemas a serem medidos.

1.4 Estrutura do Documento

Este trabalho foi dividido em 5 capítulos, onde cada um possui tópicos e objetivos diferentes:

- O segundo capítulo trata sobre o referencial teórico contendo informações sobre sistemas embarcados, transmissão de televisão e telemetria;
- O terceiro capítulo conta com a metodologia proposta, onde em primeiro momento são abordados os materiais utilizados no projeto, como o Raspberry Pi, o Arduino, o protocolo SNMP, Comunicação 3G, Bancos de dados relacionais e o *back-end* e *front-end* da aplicação. A segunda parte da metodologia trata sobre o sistema proposto, fazendo uma análise sobre como o sistema foi implementado e quais os passos para a construção;
- O quarto capítulo demonstra os resultados obtidos com relação ao sistema, e o que foi implementado.
- O quinto capítulo conclui o trabalho, apresentando as considerações finais e os passos futuros na implementação do projeto.

2 Revisão Bibliográfica

2.1 Sistemas Embarcados

De acordo com Carro e Wagner (CARRO; WAGNER, 2003), um sistema embarcado é formado por pelo menos um microprocessador ou microcontrolador capaz de processar dados e informações. Geralmente projetado para um uso específico, possuem potencial de baixo consumo energético, memórias responsáveis por armazenar o *software* e os dados, e barramentos de comunicação, onde os sinais e dados são transmitidos e periféricos, que são os componentes externos integrados ao sistema.

Exemplos claros de suas aplicações estão em eletrodomésticos, carros, televisões e várias outras aplicações do cotidiano, onde geralmente há um sistema que recebe informações, por meio de algum transdutor e realiza o processamento, para que o sistema execute determinada ação através de atuadores.

Além disso, um sistema embarcado pode ser implementado em diferentes arquiteturas de processamento (BARROS; CAVALCANTE, 2010):

Processadores de propósito geral: possuem características de alto processamento e flexibilidade, mas devido ao alto custo geralmente não são muito utilizados para a maioria das situações. Uma subcategoria desses elementos são os processadores embarcados, que incorporam vários dispositivos para baratear a solução.

Microcontroladores: geralmente são processadores de *software* com custo relativamente baixo, que englobam várias funções, como conversores analógico-digitais, conversores digital-analógicos, temporizadores, contadores, interfaces seriais, memórias de instruções e dados, controladores de interrupções, dentre outros. Apesar disso geralmente possuem baixo poder de processamento.

ASIPs (*Application-Specific Instruction-Set Processors*): processadores que são desenvolvidos para determinada função, como controle ou processamento de sinais, fazendo seu conjunto de instruções e periféricos serem escolhidos para a melhor razão de custo-benefício.

Hardware específico: pode ser feito através do uso de portas lógicas, transistores e outros componentes eletrônicos, fazendo com que o projetista ganhe total controle sobre a implementação do sistema, porém perdendo na flexibilidade do sistema para manutenções futuras.

FPGA (*Field-Programmable Gate Array*): similar ao modelo anterior, essa opção permite que o projetista também tenha total controle sobre o que será implementado no sistema, porém, utilizando programação para gerar o resultado, onde através de código, os blocos da FPGA podem se conectar internamente, para gerar um sistema dedicado de acordo com as especificações do projetista.

***System-on-Chip*:** Sistemas versáteis, pois englobam vários itens do projeto num único chip, muitas vezes fazendo combinações de processadores com microcontroladores ou FPGAs, juntando as vantagens de mais de uma arquitetura a baixo custo.

2.2 Transmissão do Sinal de Televisão

O sistema de televisão no Brasil conta com uma grande estrutura para transmissão e retransmissão de sinal. Em 2022, o país possuía 642 geradores de programação e aproximadamente 24 mil retransmissoras de sinal ([COMUNICAÇÕES, 2022](#)). Estas últimas geralmente se encontram distantes das estações geradoras, o que dificulta o trabalho de monitoração.

As estações geradoras criam e transmitem o sinal de TV, contendo uma grande gama de equipamentos necessários, que podem variar de câmeras e microfones, até codificadores e multiplexadores de sinais. Nessas estações, o sinal é gravado, codificado e depois modulado para ser transmitido. Esse processo conta com diversas etapas intermediárias, que podem variar de acordo com o tipo de programação ([SILVA, 2022](#)). Em grande parte das emissoras, o sinal é transmitido com cabos coaxiais através do protocolo SDI (*Serial Digital Interface*), que apresenta baixas taxas de perda de dados a pequenas distâncias e sem atrasos. Esse sinal depois é codificado por um *encoder* que comprime o sinal para padrões de vídeo como o MPEG-2 ou MPEG-4, e depois os envia para multiplexadores, que multiplexam os diversos fluxos de sinais e os enviam em um único sinal no formato ASI (*Asynchronous Serial Interface*), que após isso é enviado para os moduladores.

Na estação geradora, o sinal pode ser modulado de duas formas. O envio pode ser feito para o satélite através de DVB-S2 (*Digital Video Broadcasting - Satellite 2*), ou via terrestre através do padrão ISDB-T (*Integrated Services Digital Broadcasting Terrestrial*), um padrão de transmissão japonês com algumas alterações feitas no Brasil para operar a TV digital, que atinge um raio médio de 40 km, geralmente sendo utilizado para o sinal em apenas uma cidade.

As estações retransmissoras, por sua vez, contam com dois equipamentos principais, os receptores de satélite, que recebem o sinal DVB-S2 gerado pela estação geradora e o decodificam, e o transmissor de sinal digital, que realiza o processo de transmissão terrestre, também no padrão ISDB-T, para uma cidade. Geralmente esse processo é feito em pequenos contêineres, ou salas de alvenaria, que possuem a única finalidade de transmitir

esse sinal. Além disso, contam com *no-breaks* e geradores de energia em caso de falha na energia do local, e ares-condicionados para refrigerar os equipamentos. O processo de transmissão pode ser visto na Figura 1 e na Figura 2.

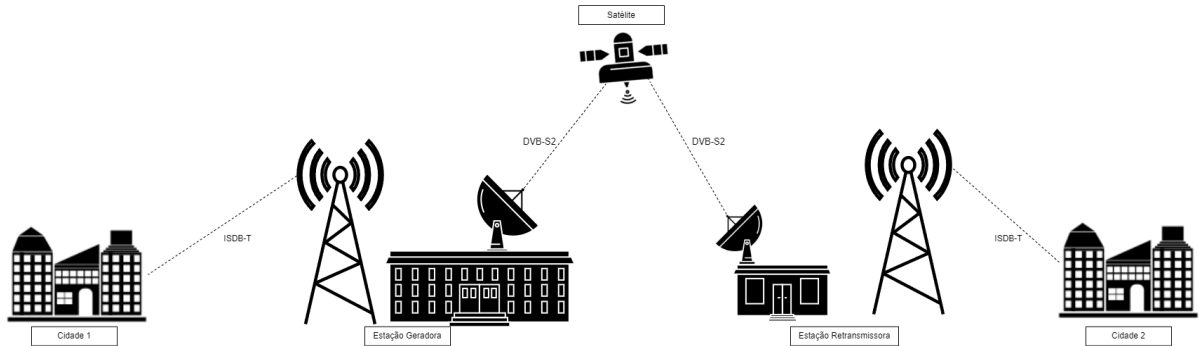


Figura 1 – Ilustração representativa de funcionamento da transmissão do sinal de TV.

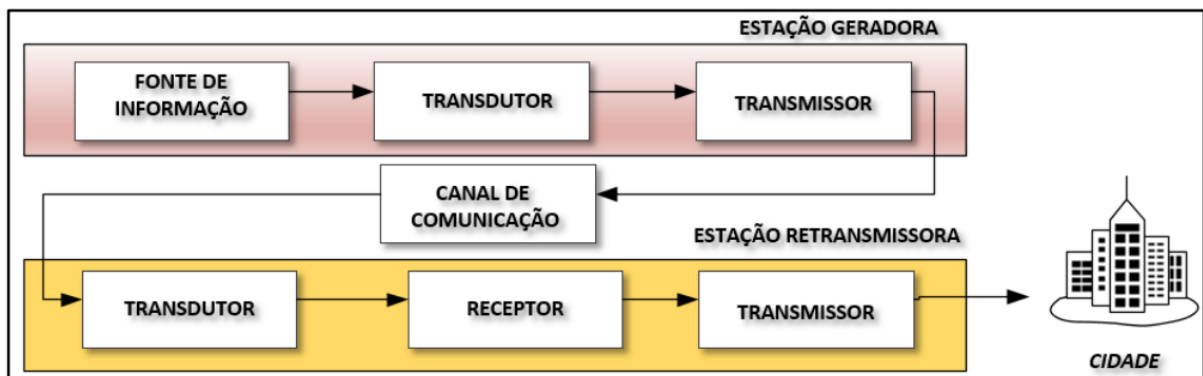


Figura 2 – Diagrama representativo de funcionamento de uma retransmissora (SILVA, 2022).

2.3 Telemetria

Telemetria consiste na medição de grandezas à distância (MATTOS, 2004). Sistemas de telemetria podem ser compostos por um hospedeiro e um servidor (Figura 3), onde o primeiro coleta as informações através de um transdutor, processa-as e as transmite a um servidor, através de um canal de comunicação com ou sem fio, que processa as informações e as exibe aos administradores do sistema (SILVA, 2022). Em grande parte dos casos, quando se trata de comunicação remota, a rede de telefonia móvel é a melhor escolha, vista que tem grande abrangência de sinal e possui uma taxa de comunicação satisfatória.

O uso de sistemas de telemetria é fundamental quando se trata de sistemas que necessitam de alta confiabilidade e se encontram em locais inacessíveis, se tornando uma grande base para tomadas de decisão em tempo real. Com a ajuda desses sistemas, é

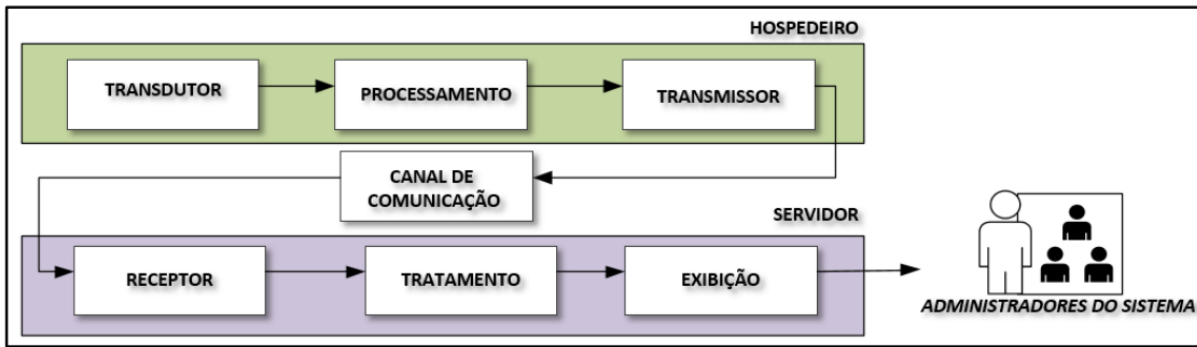


Figura 3 – Diagrama de comunicação de um sistema de telemetria. Fonte: (SILVA, 2022)

possível identificar e premeditar falhas, e com isso atuar em erros de maneira rápida e eficaz. Aplicações incluem o setor aeroespacial, o setor de navegação e as telecomunicações. No último caso, uma única emissora pode ser responsável pela transmissão do sinal em dezenas de cidades, cada uma com uma estação de retransmissão, contendo um receptor via satélite e um transmissor de TV digital. No caso de falha em um desses equipamentos, a cidade toda ficará sem sinal, e na falta de um sistema de telemetria a equipe responsável pela manutenção só irá saber da situação caso algum telespectador informe.

2.4 SNMP

O SNMP (*Simple Network Management Protocol*, ou protocolo simples de gerenciamento de redes) surgiu em 1988, como uma maneira de gerenciar diferentes dispositivos através do protocolo internet (IP - *Internet Protocol*) com um conjunto simples de instruções (MAURO; SCHMIDT, 2005). Como explicado por (TANENBAUM, 2011), o SNMP é baseado em um modelo cliente-servidor, onde o dispositivo gerenciado atua como servidor e o gerenciador de rede atua como cliente.

O gerenciador de rede, chamado de gerente, envia solicitações ao dispositivo gerenciado para obter informações sobre seu estado e configurá-lo, se necessário. O dispositivo gerenciado, chamado de agente, responde a essas solicitações com informações sobre seu estado, como uso da CPU, uso de memória e tráfego de rede. O SNMP também permite que o gerenciador de rede configure o dispositivo gerenciado, como definir a tabela de roteamento em um roteador.

O protocolo é dividido em versões, cada uma com diferentes recursos e melhorias. De acordo com (KUROSE; ROSS, 2014), a versão mais antiga é a SNMPv1, seguida pela SNMPv2c e pela SNMPv3, onde cada uma teve recursos incrementados como métodos e principalmente segurança. Além disso, o SNMP, consiste de diversos métodos para o gerenciamento dos dados, de exemplo estão os métodos *Get*, *Set*, *GetNext*, *GetBulk*, *Walk* e *Trap*, uma explicação sobre cada pode ser vista abaixo.

- O método *Get* é usado para obter informações sobre um objeto específico em um dispositivo gerenciado.
- O método *Set* é usado para configurar um objeto específico em um dispositivo gerenciado.
- O método *GetNext* é usado para obter informações sobre o próximo objeto na MIB (Management Information Base).
- O método *GetBulk* é usado para obter informações sobre vários objetos em um único pedido.
- O método *Walk* é usado para coletar informações de todos os objetos em uma determinada subárvore, permitindo que o gerenciador de rede explore a estrutura hierárquica da MIB e colete informações sobre todos os objetos que estão sob um determinado nó.
- O método *Trap* é usado pelo dispositivo gerenciado para informar ao gerenciador de rede sobre um evento ou condição importante, como uma falha de hardware ou um pico de tráfego. O dispositivo gerenciado envia uma mensagem Trap ao gerenciador de rede, que pode tomar medidas para lidar com a situação.

Além dos métodos, a estrutura dos arquivos MIB (*Management Information Base*) é fundamental para a organização dos dispositivos gerenciados. De acordo com (KUROSE; ROSS, 2014) estrutura dos arquivos MIB funciona como uma árvore hierárquica, onde a raiz é representada pelo objeto "iso", que é subdividido em subárvores. A subárvore "internet" é a mais relevante para o SNMP e é dividida em outras subárvores para cada organização que usa o protocolo para gerenciar seus dispositivos de rede. Cada subárvore é identificada por um número de identificação de objeto ou OID (*Object Identifier*) único atribuído pela IANA (*Internet Assigned Numbers Authority*).

Dentro de cada subárvore, existem objetos específicos para cada dispositivo gerenciado, identificados por OIDs únicos. Os OIDs são estruturados como uma sequência de números separados por pontos, indicando a localização do objeto na árvore de objetos do MIB. Cada objeto tem uma sintaxe definida e um significado, que são especificados no arquivo MIB correspondente, um exemplo de visualização de uma árvore MIB, pode ser vista na figura 4.

Na área de transmissão de TV o protocolo SNMP é bastante utilizado. Marcas renomadas como a Ericsson e Hitachi oferecem diversos modelos de transmissores de TV digital que utilizam o protocolo. Entre os exemplos estão o transmissor de TV digital Ericsson EN8190, que permite a transmissão de sinais de televisão digital terrestre (DTT)

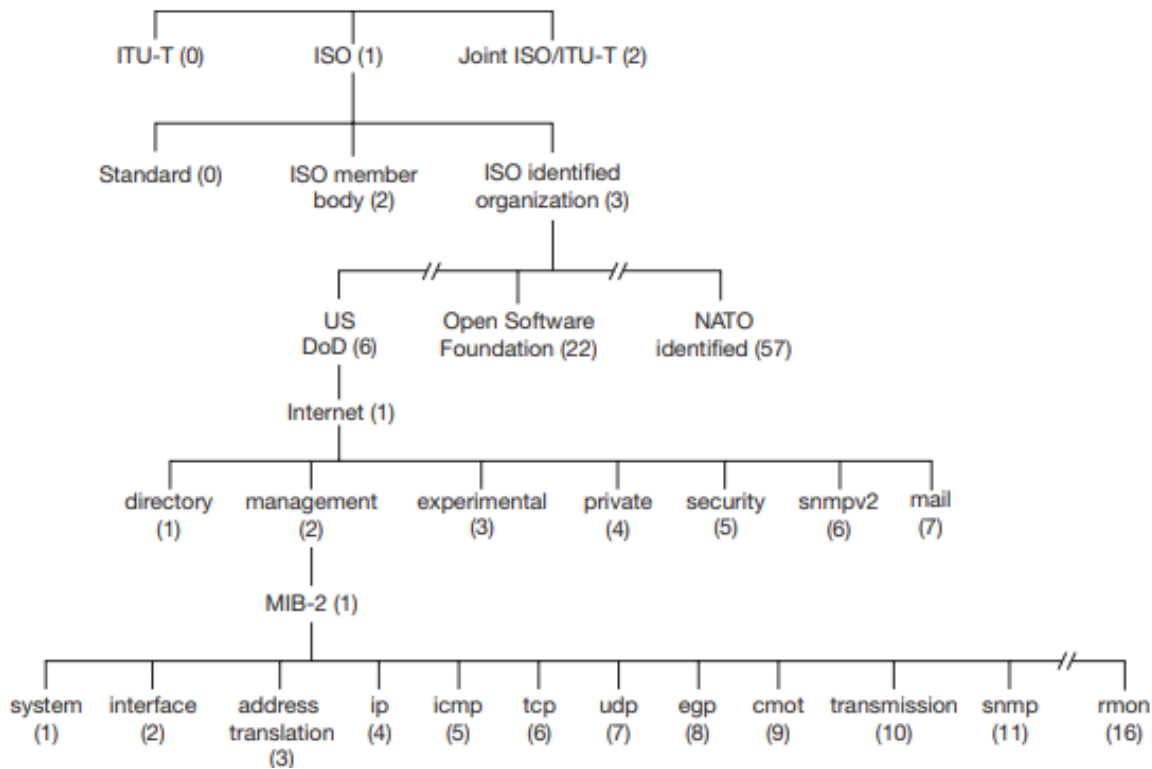


Figura 4 – Árvore de identificadores de objetos ASN.1. Fonte (KUROSE; ROSS, 2014)

em diversas faixas de frequência, o transmissor Hitachi Kokusai Electric America HZ-510E, que possui modulação de alta eficiência, compatibilidade com diversos padrões de TV digital e suporte a SNMP para gerenciamento remoto e o transmissor GatesAir Maxiva ULXTE, que oferece alta potência de transmissão, suporte a múltiplas faixas de frequência e opções de gerenciamento remoto através do protocolo SNMP.

3 Metodologia Proposta

Neste Capítulo são apresentados os materiais e procedimentos utilizados para a construção do sistema, bem como sua base teórica.

3.1 Materiais e Conceito-chave

3.1.1 Raspberry Pi

O Raspberry Pi é um computador construído numa placa única, lançado em fevereiro de 2012 pela Raspberry Pi Foundation. Por seu poder de processamento relativamente alto, baixo custo e tamanho compacto, é frequentemente utilizado em projetos de sistemas embarcados, possuindo uma série de aplicações, que podem ser implementadas de maneira simples com a ajuda de um sistema operacional.

Visando construir um sistema que atendesse aos requisitos exigidos, um sistema de telemetria foi construído com o auxílio de um Raspberry Pi 3 Model B+, que possui um processador Broadcom BCM2837B0, ARMv8 de 1.4 GHz em uma arquitetura de 64-bits quadcore, 1 GB de memória LPDDR2 SDRAM, e que permite a instalação de um sistema operacional com a ajuda de um cartão de memória SD (PI, 2018).



Figura 5 – Raspberry Pi 3 Model B+. Fonte (PI, 2018)

O sistema operacional escolhido foi o Ubuntu-mate, uma distribuição Linux que possui suporte para a arquitetura da placa. Além disso, com a ajuda desse sistema é possível configurar diversos protocolos de comunicação, como SPI, UART e I2C, que possibilitam o envio de dados através de outros periféricos, podendo também instalar programas de terceiros para rodar um servidor web, um banco de dados e aplicações em linguagens como C, Python e Javascript.

3.1.2 Arduino

Arduino é uma placa de prototipagem que se tornou bastante popular devido à sua facilidade de uso e versatilidade, contendo um microcontrolador que pode ser programado, com entradas e saídas digitais e analógicas. O Arduino foi escolhido porque a placa Raspberry Pi não contém portas analógicas.

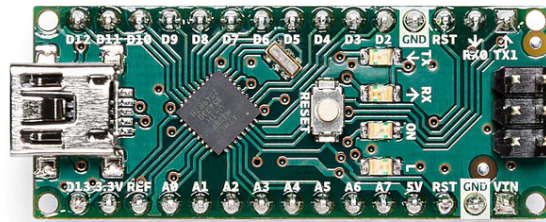


Figura 6 – Arduino Nano. Fonte: (ARDUINO, 2008)

De acordo com (NANO, 2018) placa possui 13 entradas/saídas digitais configuráveis, sendo que duas são para comunicação UART, 2 para interrupções externas, 6 para PWM, 4 para comunicação SPI, além de 8 portas analógicas, sendo 2 que podem ser configuradas para comunicação I2C.

Para o projeto, uma placa modelo Arduino Nano foi utilizada com o intuito de atuar como um leitor de entradas analógicas e digitais, para evitar danos no Raspberry Pi por corrente de fuga oriunda dos sensores, onde a placa fará a leitura dos níveis de tensão dos sensores e realizará o envio através de UART para o Raspberry como uma string, contendo o endereço da porta, seguido do valor medido.

3.1.3 Comunicação 3G

A tecnologia 3G representa a terceira geração de padrões de telefonia móvel, tendo como base as normas da União Internacional de Telecomunicações (CARDOSO, 2008), e possui alta abrangência no Brasil. Para envio dos dados da estação de retransmissão para a estação geradora, o sistema proposto utiliza o protocolo HTTP (*Hyper-Text Transfer Protocol*), que é amplamente utilizado para a transferência de dados em sites na internet. Para obter acesso à internet, o sistema se conecta a um *modem* 3G através do próprio sistema operacional, dessa forma garantindo o envio dos dados.

3.1.4 Bancos de Dados

Bancos de dados são sistemas de manutenção de registros, coletando dados e até arquivos de forma organizada (DATE, 2004). Para o sistema proposto, o sistema de banco de dados utilizado foi o PostgreSQL, um projeto *open source* de sistema gerenciador de banco de dados relacional, que faz o gerenciamento dos registros através da linguagem SQL (*Structured Query Language*), o que garante a integridade dos dados evitando redundância.

3.1.5 SSH

O SSH (Secure Shell) é um protocolo de rede que fornece um método seguro de comunicação entre dispositivos. Ele foi desenvolvido como uma alternativa ao Telnet, que transmitia informações de forma não criptografada, tornando-as suscetíveis a interceptação. O SSH utiliza técnicas de criptografia para garantir a confidencialidade, integridade e autenticação dos dados transmitidos, tornando-o amplamente utilizado em ambientes que requerem comunicações seguras, como administração remota de servidores e transferência de arquivos. De acordo com (TANENBAUM, 2011), o SSH emprega algoritmos criptográficos, como o RSA e o Diffie-Hellman, para estabelecer chaves de sessão e proteger os dados durante a transmissão. Ele também oferece recursos avançados, como autenticação de dois fatores e tunelamento, proporcionando uma camada adicional de segurança. O SSH é uma ferramenta essencial para a segurança de redes e sistemas, ajudando a proteger contra ameaças e ataques maliciosos.

3.1.6 VPN

Uma VPN (Virtual Private Network) é uma tecnologia de rede que estabelece uma conexão segura e criptografada sobre uma rede pública, como a Internet. Essa conexão permite que usuários remotos acessem recursos privados de uma rede de forma segura, como se estivessem fisicamente presentes nessa rede local. Segundo (TANENBAUM, 2011), uma VPN utiliza protocolos de tunelamento e criptografia para garantir a confidencialidade, integridade e autenticidade dos dados transmitidos.

Ao estabelecer uma conexão VPN, os dados são encapsulados em pacotes protegidos criptograficamente, o que impede que terceiros interceptem ou modifiquem as informações transmitidas. Além disso, uma VPN pode ser configurada para mascarar o endereço IP do usuário remoto, proporcionando anonimato e proteção contra rastreamento.

3.1.7 *Back-end*

O *Back-end* representa o processamento executado do lado do servidor, ou *server-side*, desempenhando as regras de negócio do sistema, o acesso a dados e a segurança de

ambos (ROCHA et al., 2019). Para a confecção do sistema no *back-end*, foi utilizado o FastAPI, um *framework* de alta performance construído em Python para ser completo e rápido de programar. Nele, o programador declara as rotas e os métodos, e o sistema gera o servidor de maneira automática, o que facilita a exportação de arquivos no formato REST.

3.1.8 Front-end

Em oposição ao *back-end*, o *front-end* representa o processamento do lado do cliente, ou *client-side*, onde o usuário interage diretamente com a interface (ROCHA et al., 2019). Para o desenvolvimento do *front-end*, foi utilizado o ReactJS, um *framework* JavaScript, desenvolvido pelo Facebook e baseado em componentes, que facilitam o reaproveitamento de código, pois o mesmo trecho pode ser utilizado em diferentes contextos, apenas importando o componente utilizado.

3.2 Sistema Proposto

Para solucionar o problema de medições do local e rede de forma remota, são utilizadas algumas formas de medição de dados, tanto por meio de protocolos como por *hardware*.

As medições do local foram feitas por meio de sensores e integrados por meio de uma API para serem mostrados em tempo real em um servidor. Para o monitoramento dos transmissores, foi utilizado o protocolo SNMP, que confere o OID (*object identifier*) de cada objeto, de forma a extrair as informações presentes na rede. Foi configurado no sistema proposto um agente SNMP, para que ele seja capaz de obter dados do transmissor digital através desses protocolos, e através dos comandos *snmpget* e *snmpwalk* consiga navegar pelos dados disponibilizados pelos dispositivos.

Dessa forma, o sistema permite medir dados de rede e do ambiente em que o aparelho se encontra de forma remota. Caso algo de anormal ocorra, será possível controlar o sistema para reiniciar o equipamento.

3.2.1 Hardware

A Figura 7 apresenta o diagrama de blocos do sistema desenvolvido. A alimentação do sistema é dada por um *plug AC* que se conecta à fonte de tensão *DC* para o resto dos sistemas, podendo ser ligada ou desligada por uma pequena chave localizada no painel. O *Raspberry Pi 3 model B+* é o responsável por realizar o processamento e envio dos dados, se conectando aos periféricos e também controlando algumas das entradas e saídas digitais. O *Arduino Nano* se comunica com o *Raspberry* e é capaz de realizar a medida dos

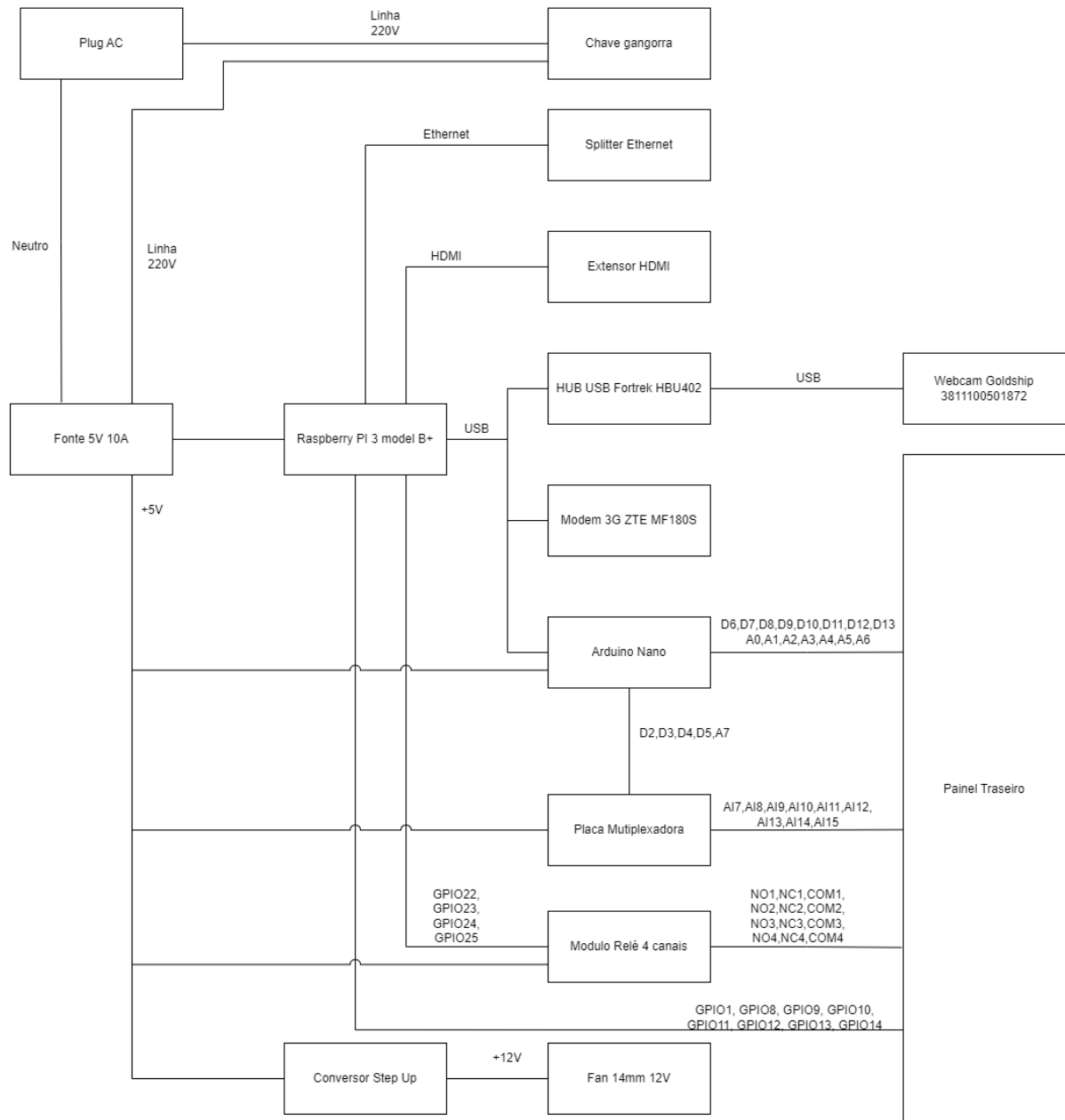


Figura 7 – Diagrama de blocos do sistema proposto.

sensores analógicos além de executar e ler alguns comandos digitais. A Tabela 1 apresenta a lista de materiais do sistema proposto e suas informações sobre os custos, pesquisados no dia 26 de julho de 2023, onde o valor total do sistema foi estimado em 1398,39 reais.

Tabela 1 – Lista de materiais

| Material | Quantidade | Valor da unidade |
|---|------------|------------------|
| Raspberry Pi 3 B+ | 1 | R\$ 620,00 |
| Arduino Nano | 1 | R\$ 74,90 |
| Módulo de Relés de 4 canais | 1 | R\$ 24,90 |
| Fonte 5 V 10 A 50 W Lumi | 1 | R\$ 35,59 |
| Modem 3G ZTE MF180S | 1 | R\$ 79,00 |
| HUB USB Fortrek HBU402 | 1 | R\$ 27,31 |
| Placa multiplexadora com 74HC4067 | 1 | R\$ 25,00 |
| Cabo HDMI Macho Fêmea Extensor 30 cm | 1 | R\$ 29,00 |
| RJ45 Ethernet Splitter Cable | 1 | R\$ 67,20 |
| Webcam Goldship 3811100501872 | 1 | R\$ 22,99 |
| Módulo Conversor Step Up XL6009 | 1 | R\$ 14,90 |
| Fan 40mm 12V | 2 | R\$ 13,80 |
| Gabinete para <i>Rack</i> 1U com conectores | 1 | R\$ 350,00 |

Os elementos que compõem o sistema estão descritos a seguir:

Raspberry Pi 3 model B+: componente central do projeto, o dispositivo foi configurado utilizando o sistema operacional *Ubuntu Mate*. Ele armazena os dados internamente em um banco de dados, contendo uma série de programas que rodam em paralelo para realizar a medição e o envio dos dados. Dentre eles, temos (a) um servidor *web nginx* que disponibiliza uma imagem da câmera USB com o software *Motion*; (b) um código para medição dos dados do *Arduino Nano* via porta serial, para monitoramento das entradas e saídas digitais; (c) uma porta exclusiva para um sensor de temperatura e umidade; (d) um código para monitorar os dados via SNMP e outro para realizar o envio desses dados. Além disso, o dispositivo foi configurado com uma VPN (*Virtual Private Network*) para se comunicar com o servidor fazendo com que a transferência de dados seja criptografada e protegida.

Todos os executáveis foram configurados como serviços no *Linux*, de forma a serem reiniciados automaticamente em caso de falha.

O *Raspberry Pi* possui suas GPIO conectadas as entradas e saídas digitais e suas conexões podem ser vistas na Tabela 2.

Arduino Nano: responsável por fazer o controle e a medição dos sinais analógicos, além de alguns sinais digitais. Os dados são medidos pelo dispositivo e são enviados via porta serial USB para o *Raspberry Pi*, além de realizar o controle da placa multiplexadora para expandir as entradas analógicas.

Tabela 2 – Conexões Raspberry

| Origem | Destino |
|---------------|-----------------------------|
| GPIO 8 | Porta para Sensor DHT11 |
| GPIO 9 | Digital Input Ch0 |
| GPIO 10 | Digital Input Ch1 |
| GPIO 11 | Digital Input Ch2 |
| GPIO 12 | Digital Input Ch3 |
| GPIO 22 | Digital Output Ch0 (Relé 1) |
| GPIO 23 | Digital Output Ch1 (Relé 2) |
| GPIO 24 | Digital Output Ch2 (Relé 3) |
| GPIO 25 | Digital Output Ch3 (Relé 4) |
| GPIO 13 | Digital Output Ch4 |
| GPIO 14 | Digital Output Ch5 |
| GPIO 1 | Digital Output Ch6 |

O *Arduino* possui suas portas conectadas ao painel traseiro e também a placa multiplexadora como pode ser visto na Tabela 3.

Tabela 3 – Conexões Arduino

| Origem | Destino |
|---------------|--------------------------|
| D2 | Placa multiplexadora S0 |
| D3 | Placa multiplexadora S1 |
| D4 | Placa multiplexadora S2 |
| D5 | Placa multiplexadora S3 |
| D6 | Digital Input Ch4 |
| D7 | Digital Input Ch5 |
| D8 | Digital Input Ch6 |
| D9 | Digital Input Ch7 |
| D10 | Digital Input Ch8 |
| D11 | Digital Output Ch7 |
| D12 | Digital Output Ch8 |
| D13 | Digital Output Ch9 |
| A0 | Analog Input Ch0 |
| A1 | Analog Input Ch1 |
| A2 | Analog Input Ch2 |
| A3 | Analog Input Ch3 |
| A4 | Analog Input Ch4 |
| A5 | Analog Input Ch5 |
| A6 | Analog Input Ch6 |
| A7 | Placa multiplexadora Sig |

Módulo Relé de 4 canais: serve para emitir sinais digitais para equipamentos caso necessário, especialmente para cortar a alimentação dos equipamentos ou reiniciá-los em caso de falhas. Este módulo representa os canais de 0 a 3 das saídas digitais, onde cada um possui um estado normalmente aberto e normalmente fechado.

Fonte 5 V 10 A 50 W Lumi: é a fonte de 5V responsável por alimentar todo o sistema, com conexões ao *Raspberry*, ao *Arduino*, à placa multiplexadora, ao conversor *Step Up* e ao painel traseiro.

Modem 3G ZTE MF180S: conecta o *Raspberry Pi* com a internet através de uma conexão USB.

HUB USB Fortrek HBU402: expande 4 portas USB para o painel, de maneira a não precisar abrir o equipamento quando for necessário conectar um *mouse* e/ou um teclado.

Placa multiplexadora com 74HC4067: é responsável por expandir as portas analógicas do sistema. Além disso, foram adicionados alguns divisores resistivos de tensão, de maneira a aumentar o valor máximo de tensão aceito pelo sistema. O esquemático do sistema pode ser visto na Figura 8.

Com a ajuda dos divisores de tensão, os canais analógicos de 8 a 15 possuem um *range* de tensão variado, podendo medir de 0 a 50V nos canais 15 e 14, 36V nos canais 13 e 12, 14V nos canais 11, 10 e 9, 12V no canal 8, e 5V nos canais 0 a 7.

Cabo HDMI Macho-Fêmea Extensor 30 cm: estende a conexão HDMI do *Raspberry* para o painel, de maneira a não precisar abrir o dispositivo toda vez que for conectar um monitor.

RJ45 Ethernet Splitter Cable: expande 3 portas *Ethernet* para o painel, de maneira a não precisar abrir o equipamento para conectar um cabo de rede, além de permitir a conexão de 3 equipamentos simultaneamente.

Webcam Goldship 3811100501872: é responsável por capturar imagens do ambiente e envia-las ao servidor quando solicitado. Por questão de economia de dados, o *streaming* é feito apenas quando solicitado e não durante todo o momento.

Módulo Conversor Step Up XL6009: realiza a conversão da tensão de 5V da fonte para 12V, para alimentar as duas ventoinhas.

Fan 40mm 12V: faz o arrefecimento do sistema, através de um espaço de ventilação localizado no painel traseiro.

Gabinete para Rack 1U com conectores: para acoplar todo o sistema, foi feito um gabinete de aço com as dimensões para *rack* de uma unidade, permitindo acoplarlo na maioria dos locais em dificuldades. O desenho tridimensional do dispositivo modelado a partir do software CATIA v5 pode ser visto na Figura 12. O painel frontal e o traseiro foram cortados em uma máquina CNC, para que no frontal fosse aberto o espaço para os conectores USB, RJ45 e HDMI, dos periféricos. Além

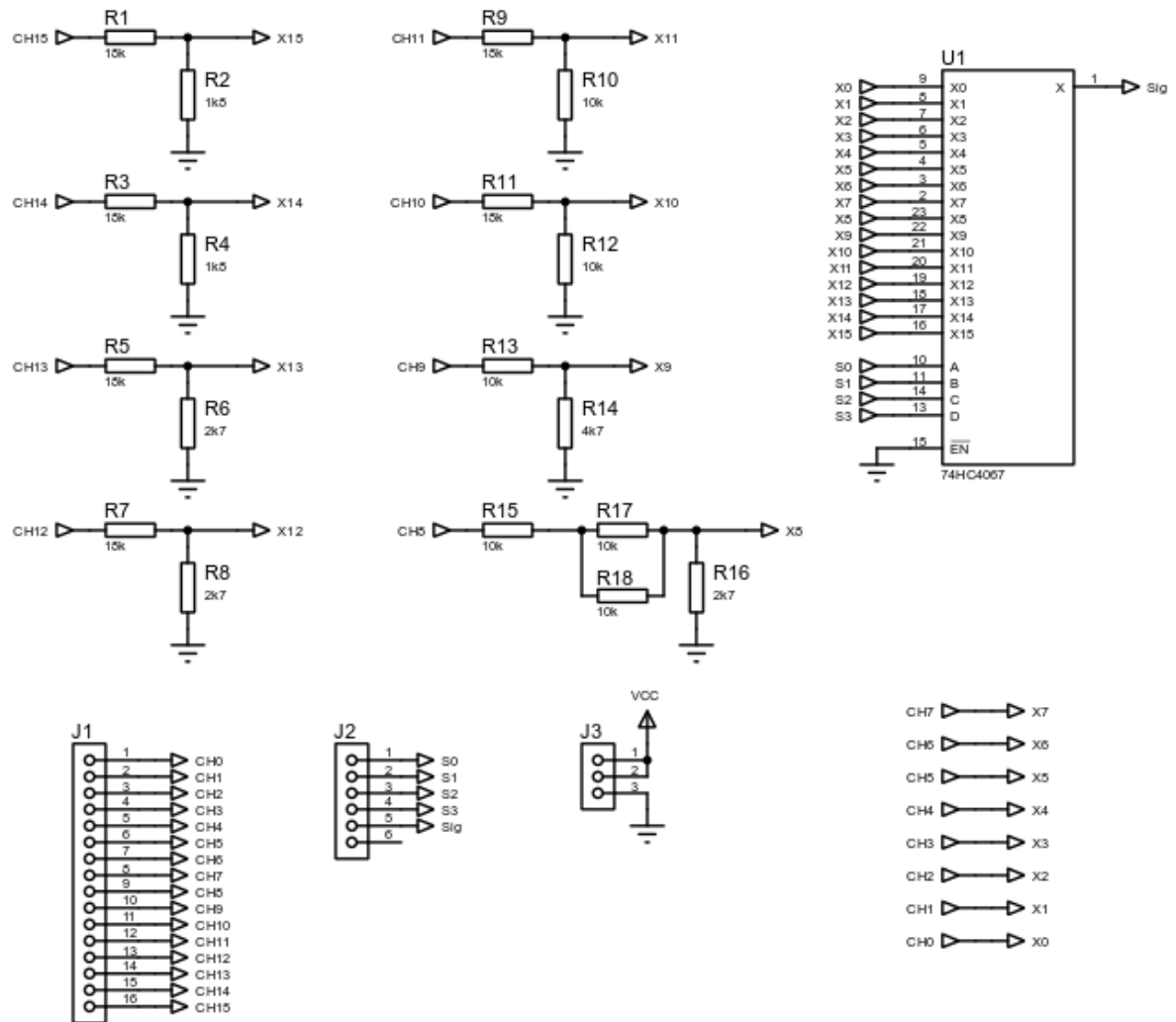


Figura 8 – Esquemático placa multiplexadora.

disso, foi posicionada uma chave gangorra em série com a entrada da fonte para ligar e desligar o equipamento. No painel traseiro, foi adicionado um *plug AC* para alimentar o sistema, alguns furos para ventilação e os conectores do tipo KRA, sendo 4 de 3 vias para as saídas de relé e 6 de 8 vias para as saídas digitais e entradas analógicas e digitais e também alimentação de sensores externos. As vistas superior, frontal, traseira e isométrica podem ser vistas respectivamente nas Figuras 9, 10, 11 e 12.

O protótipo desenvolvido correspondeu com fidelidade ao projeto, onde as disposições dos componentes se mantiveram, conforme pode ser visto na Figura 13

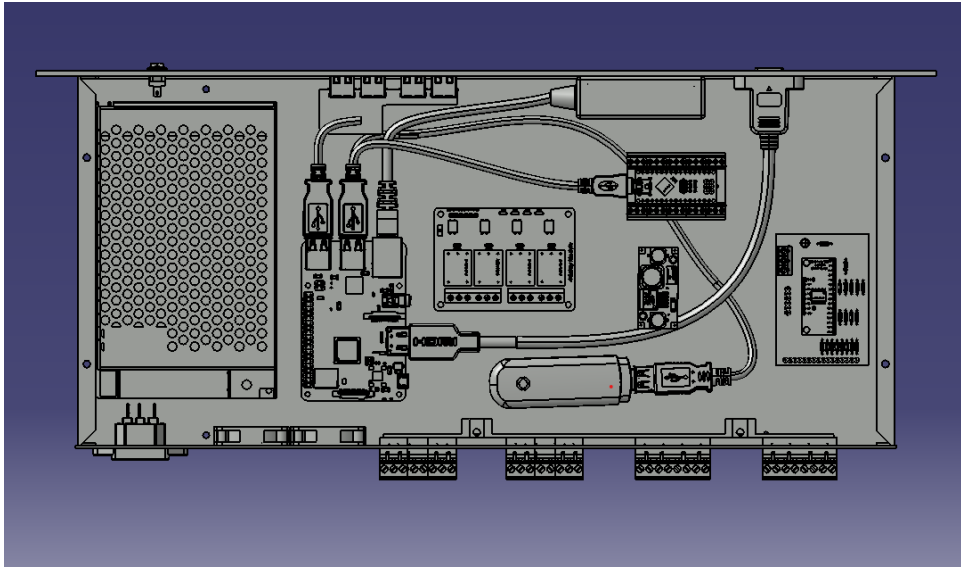


Figura 9 – Vista superior do gabinete desenvolvido.



Figura 10 – Vista frontal do gabinete desenvolvido.

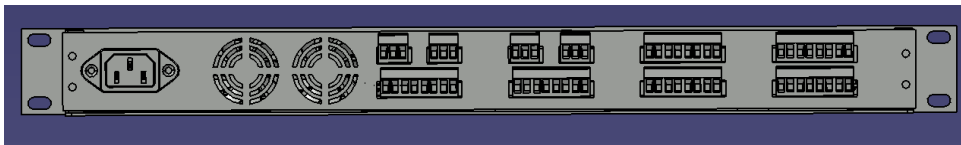


Figura 11 – Vista traseira do gabinete desenvolvido.

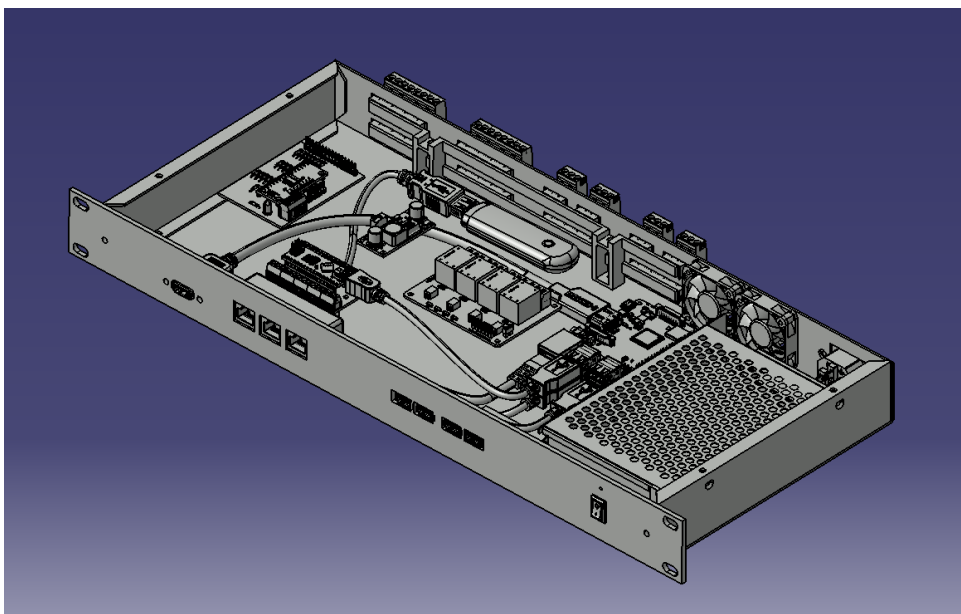


Figura 12 – Vista isométrica do gabinete desenvolvido.

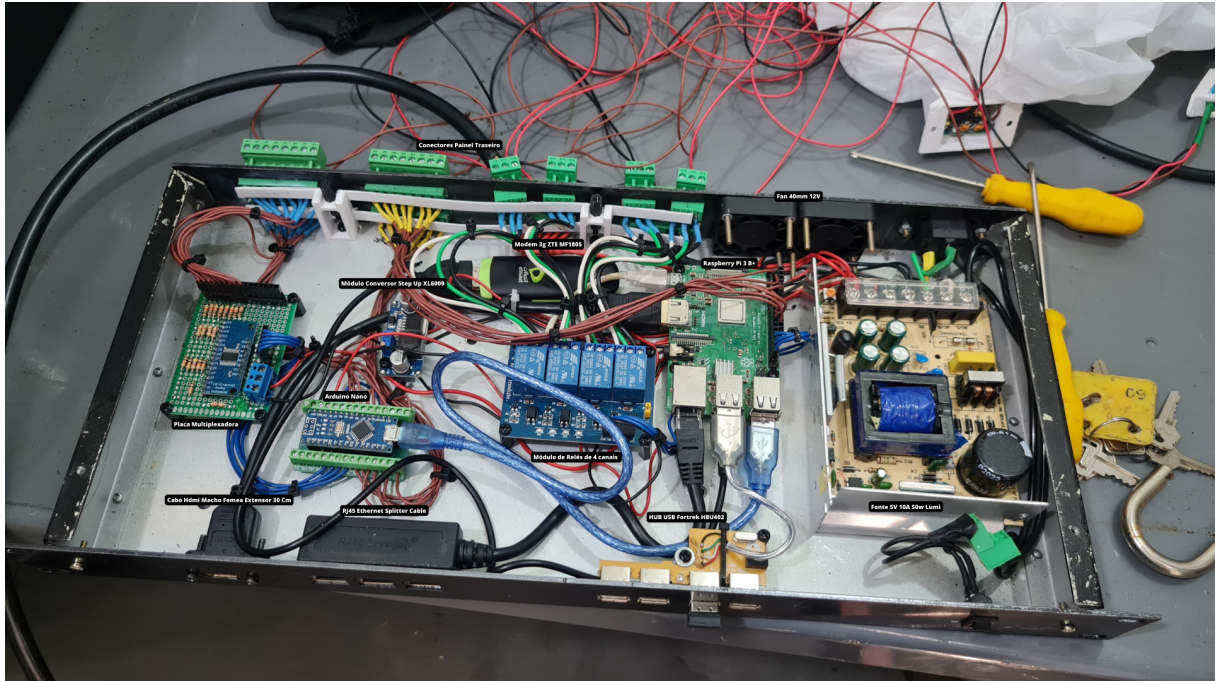


Figura 13 – Protótipo finalizado.

3.2.2 Software

O *software* desenvolvido foi dividido em várias partes, para facilitar o entendimento. A maioria das partes foi feita em linguagem C, a não ser se indicado na descrição a seguir:

Leitor SNMP: lê todos os dados SNMP presentes nos OID's disponíveis no banco de dados e os salva no banco de dados.

Leitor de temperatura e umidade: lê os dados de temperatura e umidade através do sensor DHT11, e armazena-os no banco de dados.

Leitor serial: lê todos os dados de entradas analógicas e digitais medidas pelo Arduino através da porta serial, e os armazena em arquivos de texto.

Escrita digital: escreve os dados das saídas digitais de acordo com um comando que pode ser executado remotamente através do SSH, e caso a saída digital esteja no Arduino ela será salva em um arquivo de texto e enviado pelo leitor serial.

Leitor digital: lê os valores presentes nas portas digitais ou no arquivo de texto salvo pelo leitor serial, e salva o resultado no banco de dados.

Leitor analógico: lê os valores analógicos no arquivo de texto salvo pelo leitor serial e salva o resultado no banco de dados.

Servidor de vídeo: usando o software Motion, disponibiliza a câmera remota em um servidor web.

Envio de dados: obtém os dados armazenados no banco de dados e envia através de protocolo HTTP para o servidor remoto.

Servidor Web: é um servidor Linux na nuvem, hospedado no *Google Cloud Platform*, com uma API Python com o *framework FastAPI*, que recebe as informações do equipamento no formato JSON e armazena-os no banco de dados. Além disso, contém um módulo web escrito em JavaScript com o *framework ReactJS* para exibir os dados. Este servidor está conectado em uma rede privada virtual com a estação de telemetria, o que permite que ela envie comandos através de SSH para controlar totalmente a estação e reiniciar equipamentos.

3.3 Ambiente de Teste

Para a execução dos testes, o ambiente utilizado foi uma estação de retransmissão da TV Oeste, afiliada da Rede Globo localizada na cidade de Barreiras na Bahia. A estação escolhida fica na cidade de Ibotirama, a aproximadamente 200km da sede central da emissora.

A estação da cidade de Ibotirama possui uma estrutura simples, montada em um contêiner metálico, contendo um receptor, um *no-break* e um transmissor de sinal digital, conforme pode ser visto nas Figuras 14 e 15.

Para executar os testes, foram acoplados alguns sensores ao equipamento, dentre eles um sensor de presença e um sensor de temperatura e umidade, que foram conectados ao canal de entrada digital 4 e ao canal exclusivo para DHT11 presentes no painel traseiro. Também foi adicionado um módulo de *reset* em série com o relé do canal 1 do equipamento para realizar os comandos digitais, e a porta de saída de rede foi conectada a um *switch* de rede de 5 portas Multilaser RE305, no qual o mesmo já estava conectado ao transmissor e ao receptor de sinal. O dispositivo instalado e seus periféricos podem ser vistos nas Figuras 16, 17 e 18.



Figura 14 – Exterior do abrigo.



Figura 15 – Interior do abrigo.



Figura 16 – Estação de telemetria instalada no abrigo.



Figura 17 – Sensor de Presença, sensor de temperatura e webcam instalados.



Figura 18 – *Switch* de rede e módulo de *reset* instalados.

4 Resultados Experimentais

O produto desenvolvido pode ser dividido em alguns subprodutos, sendo eventualmente agrupados em um só para atingir o objetivo final de ser um sistema capaz de monitorar remotamente uma estação de retransmissão de emissora de televisão.

4.1 Apresentação dos Dados

Os dados enviados pela estação de telemetria são armazenados no servidor, e então disponibilizados em uma API que trabalha exclusivamente com JSON. Uma forma de fácil visualização foi construída a partir de uma aplicação em *Javascript* em combinação com o *framework ReactJS*, que foi dividido em diversas telas para monitorar os dados como visto a seguir:

Tela de login: é a tela inicial da aplicação (Figura 19). Após acesso do usuário com as credenciais, ela realiza uma requisição para o *back-end*, que retorna um *token* de acesso para autenticar o usuário.

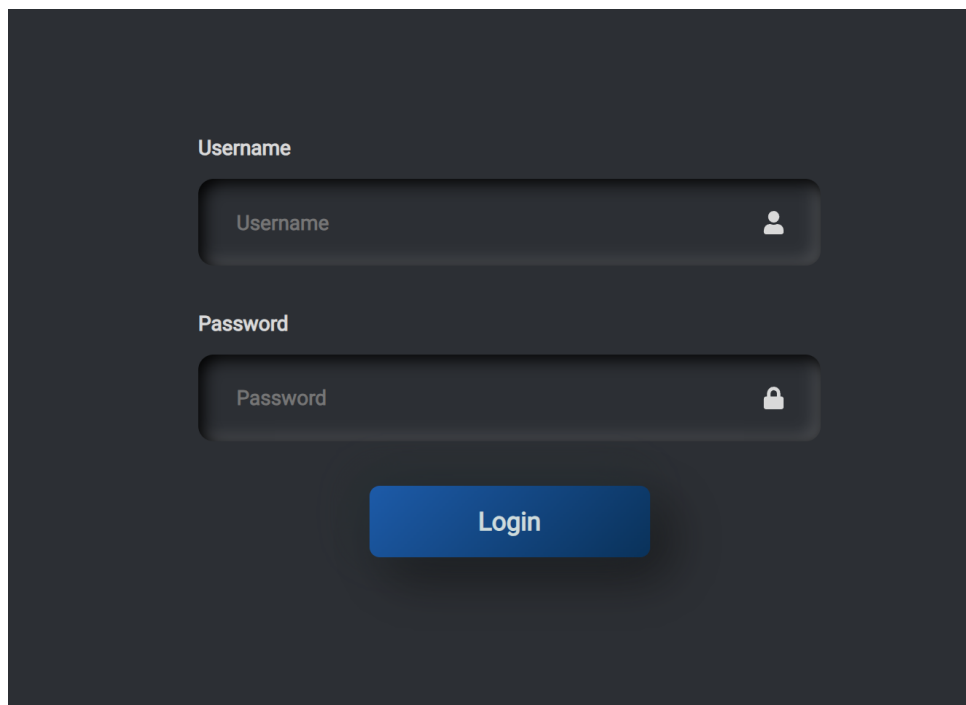
A imagem mostra a interface de login de uma aplicação web. O fundo é escuro cinza. No topo, o rótulo "Username" está em branco. Abaixo dele, há um campo de entrada de texto com o placeholder "Username" e um ícone de usuário à direita. Logo abaixo, o rótulo "Password" está em branco. Abaixo dele, há um campo de entrada de texto com o placeholder "Password" e um ícone de cadeado à direita. No centro da tela, há um botão azul com o texto "Login" em branco.

Figura 19 – Tela de login.

Dados digitais: é a tela responsável por indicar o estado das entradas digitais (Figura 20), onde é indicado o nome da entrada digital configurada, o canal de entrada do

dispositivo, a data da última atualização, e o estado da entrada. Caso se encontre em cor verde, ela está normal, e caso esteja em cor vermelha há alguma anomalia.

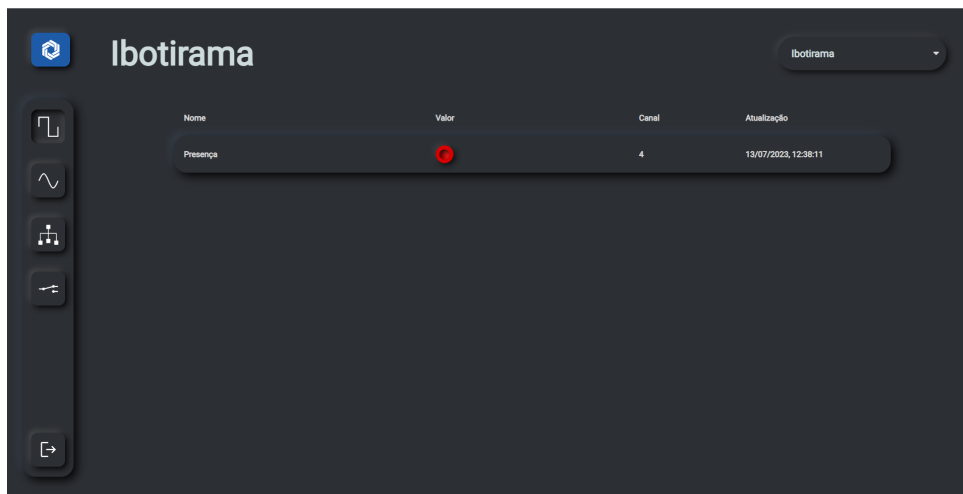


Figura 20 – Tela de entradas digitais.

Dados analógicos: é a tela responsável por indicar o valor das entradas analógicas e também dos dados de temperatura e umidade (Figura 21), onde é indicado qual o nome da entrada analógica configurada, o canal de entrada do dispositivo, a data da última atualização, além do valor. Pode ser configurada uma função, onde o usuário pode definir um ajuste para a leitura, como por exemplo dividir o valor lido por 10, ou multiplicá-lo por alguma constante para realizar uma correção na medida.

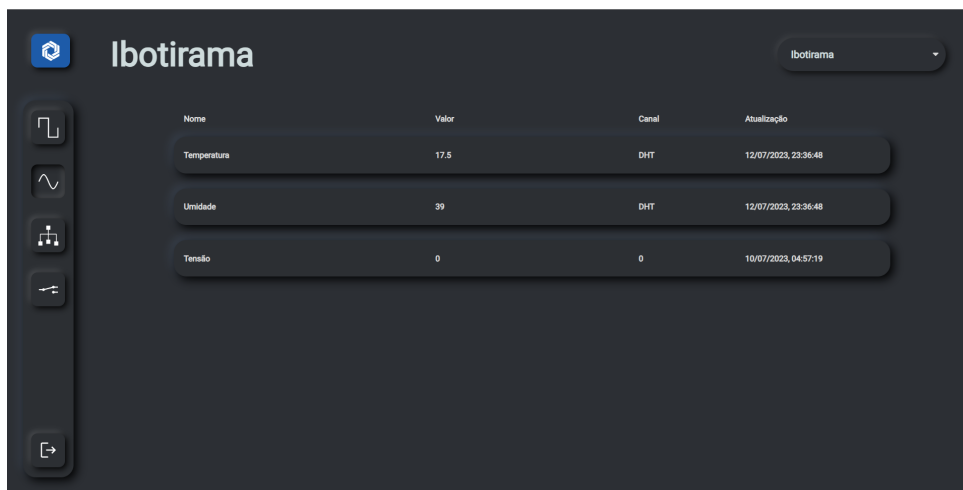
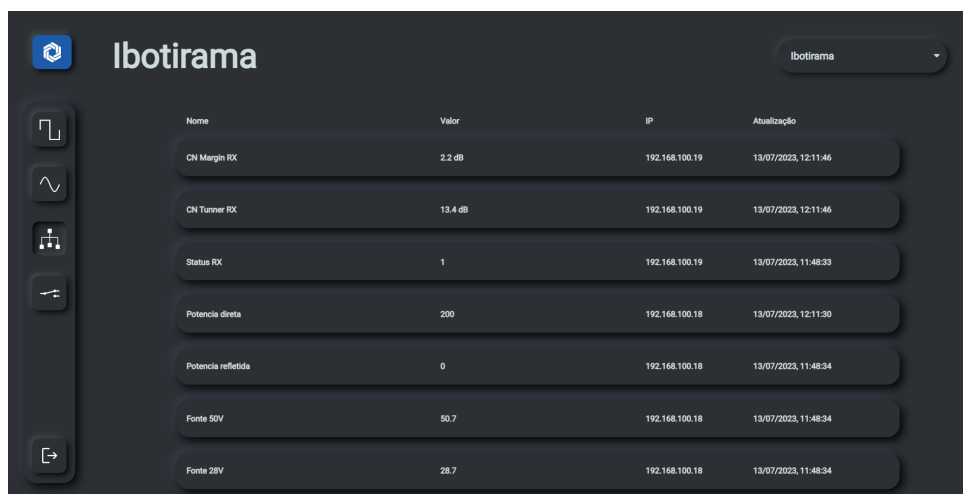


Figura 21 – Tela de entradas analógicas.

Dados SNMP: é a tela responsável por indicar o valor dos dados obtidos via protocolo SNMP (Figura 22), onde é indicado qual o nome da entrada OID configurada, o IP do equipamento no qual deseja-se obter o dado, a data da última atualização, além do valor. Assim como nas medidas analógicas, pode ser configurada um ajuste através de uma função para tornar a visualização mais fácil e real.

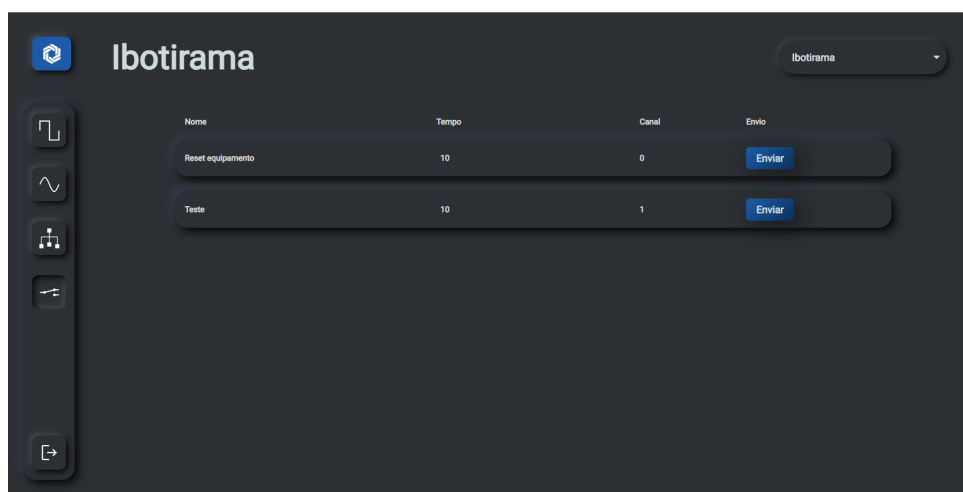


The screenshot shows the 'Ibotirama' interface with a table of SNMP data. The table has four columns: 'Nome', 'Valor', 'IP', and 'Atualização'. The data rows are as follows:

| Nome | Valor | IP | Atualização |
|--------------------|---------|----------------|----------------------|
| CN Margin RX | 2.2 dB | 192.168.100.19 | 13/07/2023, 12:11:46 |
| CN Turner RX | 13.4 dB | 192.168.100.19 | 13/07/2023, 12:11:46 |
| Status RX | 1 | 192.168.100.19 | 13/07/2023, 11:48:33 |
| Potencia direta | 200 | 192.168.100.18 | 13/07/2023, 12:11:30 |
| Potencia refletida | 0 | 192.168.100.18 | 13/07/2023, 11:48:34 |
| Fonte 50V | 50.7 | 192.168.100.18 | 13/07/2023, 11:48:34 |
| Fonte 28V | 28.7 | 192.168.100.18 | 13/07/2023, 11:48:34 |

Figura 22 – Tela de dados SNMP.

Envio de comandos: é a tela responsável por enviar comandos à estação de telemetria (Figura 23), onde é indicado qual o nome da saída digital configurada, o canal de saída do dispositivo e o tempo que o comando deve durar, sendo este último alterável pelo usuário. A tela conta também com um botão para enviar o comando.



The screenshot shows the 'Ibotirama' interface with a table for sending commands. The table has four columns: 'Nome', 'Tempo', 'Canal', and 'Envio'. The data rows are as follows:

| Nome | Tempo | Canal | Envio |
|-------------------|-------|-------|--------|
| Reset equipamento | 10 | 0 | Enviar |
| Teste | 10 | 1 | Enviar |

Figura 23 – Tela de envio de comandos.

Câmera: é a tela responsável por espelhar a *webcam* da estação de telemetria através da VPN (Figura 24). Ela é obtida através da rota/câmera, sendo possível acompanhar o ambiente em tempo real, porém é suscetível a quedas na conexão da estação remota.

4.2 Monitoramento Via Protocolo SNMP

Para as medições de SNMP, o código funcionou corretamente. Foram registradas 17 OID's conectadas em dois dispositivos: o receptor digital e o transmissor digital (Tabela 4). Os itens presentes na medição se referem a:



Figura 24 – Tela de espelho da *webcam*.

CN Margin Rx: (Carrier-to-Noise Margin) diferença entre a potência do sinal do canal (portadora) e o nível de ruído presente no sistema.

CN Tuner Rx: (Carrier-to-Noise Tuner) relação C/N do *tuner* do receptor, que é responsável por sintonizar e receber o sinal da portadora no sistema de comunicação.

Status RX: indica a situação do receptor. Caso ele esteja funcionando, deve ser 1.

Potência direta: potencia em Watts que está sendo transmitida para a antena.

Potência refletida: potencia em Watts que está sendo refletida pela antena.

Fonte 50V: tensão da fonte de 50V do transmissor.

Fonte 28V: tensão da fonte de 28V do transmissor.

Corrente M1: corrente consumida pelo módulo 1 do transmissor.

Corrente M2: corrente consumida pelo módulo 2 do transmissor.

Corrente M3: corrente consumida pelo módulo 3 do transmissor.

Corrente M4: corrente consumida pelo módulo 4 do transmissor.

Temperatura M1: temperatura interna do módulo 1 do transmissor.

Temperatura M2: temperatura interna do módulo 2 do transmissor.

Temperatura M3: temperatura interna do módulo 3 do transmissor.

Temperatura M4: temperatura interna do módulo 4 do transmissor.

Temperatura *Driver*: temperatura interna do *driver* do transmissor.

Tabela 4 – Tabela de OID SNMP

| Título | IP | OID | Equipamento |
|--------------------|----------------|-----------------------------------|-------------|
| CN Margin RX | 192.168.100.19 | .1.3.6.1.4.1.1773.1.3.208.2.2.6.0 | Receptor |
| CN Tuner RX | 192.168.100.19 | .1.3.6.1.4.1.1773.1.3.208.2.2.5.0 | Receptor |
| Status RX | 192.168.100.19 | .1.3.6.1.4.1.1773.1.3.208.2.1.7.0 | Receptor |
| Potencia direta | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.6.0 | Transmissor |
| Potencia refletida | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.7.0 | Transmissor |
| Fonte 50V | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.1.0 | Transmissor |
| Fonte 28V | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.3.0 | Transmissor |
| Corrente M1 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.4.0 | Transmissor |
| Corrente M2 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.5.0 | Transmissor |
| Corrente M3 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.6.0 | Transmissor |
| Corrente M4 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.7.0 | Transmissor |
| Corrente Driver | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.12.0 | Transmissor |
| Temperatura M1 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.13.0 | Transmissor |
| Temperatura M2 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.14.0 | Transmissor |
| Temperatura M3 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.15.0 | Transmissor |
| Temperatura M4 | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.16.0 | Transmissor |
| Temperatura Driver | 192.168.100.18 | .1.3.6.1.4.1.25026.11.1.101.21.0 | Transmissor |

Os dados foram medidos durante uma semana, e foi possível notar pouca variação na maioria dos itens. Um exemplo disso é a potencia direta do transmissor, que se manteve praticamente inalterada, conforme apresentado na Figura 25, que retrata o período entre a noite do dia 8 de julho de 2023 e a manhã do dia 9 de julho de 2023.

É possível que alguns pacotes sejam perdidos durante as requisições, havendo nesse caso uma falha na identificação, o que pode fazer com que o programa de leitura SNMP retorne um erro. Por conta disso, o programa foi iniciado como um serviço do sistema operacional, garantindo que caso haja qualquer erro ele seja reiniciado.

4.3 Monitoramento de Temperatura e Umidade

O sensor de temperatura e umidade utilizado foi o DHT11, acoplado diretamente ao *Raspberry*, de forma a executar continuamente a medição da temperatura e umidade e enviar os dados para o servidor. Durante os testes, o ar-condicionado, que era a principal fonte de arrefecimento do abrigo, apresentou alguns problemas técnicos. Por conta disso, as amplitudes térmicas da estação durante o dia variavam em mais de 30 graus, o que interferia diretamente na medida de temperatura e umidade dentro do abrigo. As Figuras

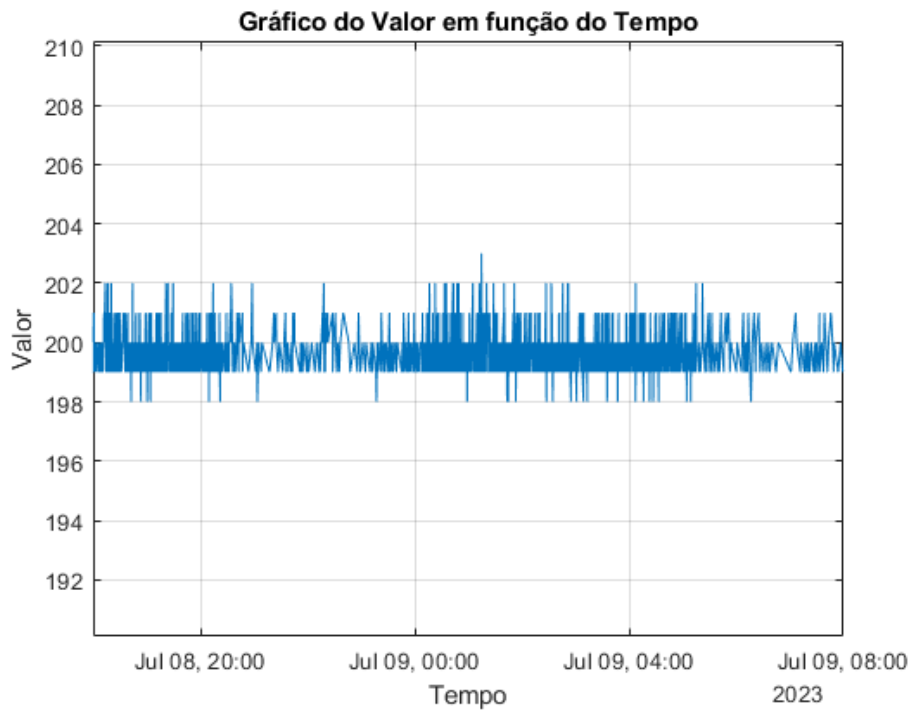


Figura 25 – Gráfico da variação da potência transmitida em função do tempo entre a noite do dia 8 de julho de 2023 e a manhã do dia 9 de julho de 2023.

26 e 27 mostram a variação das medições realizadas ao longo do dia 12 de julho de 2023, onde os valores medidos foram condizentes com os relatados pela equipe de manutenção.

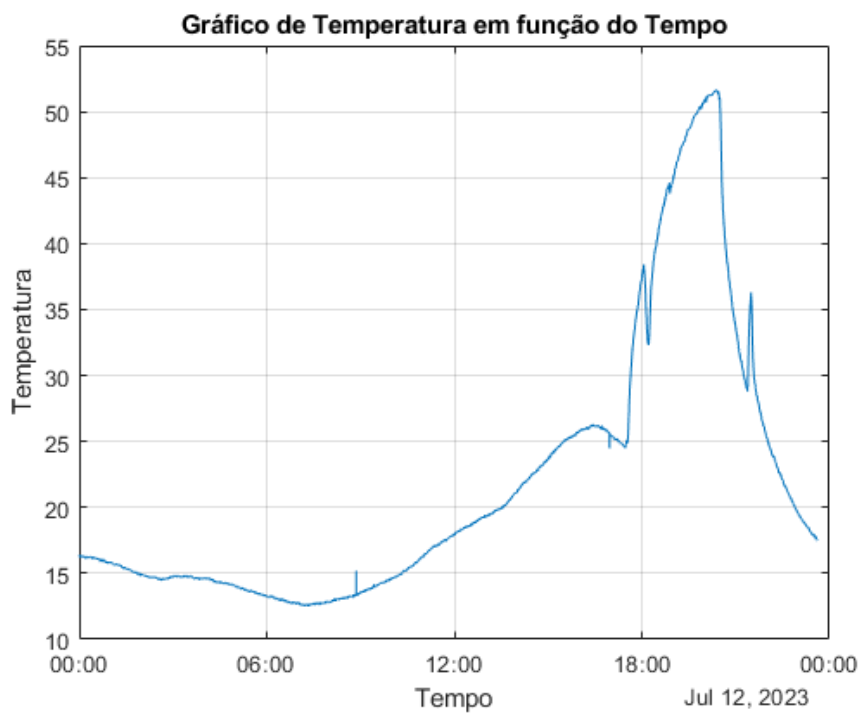


Figura 26 – Gráfico do sensor de temperatura em função do tempo no dia 12/07/2023.

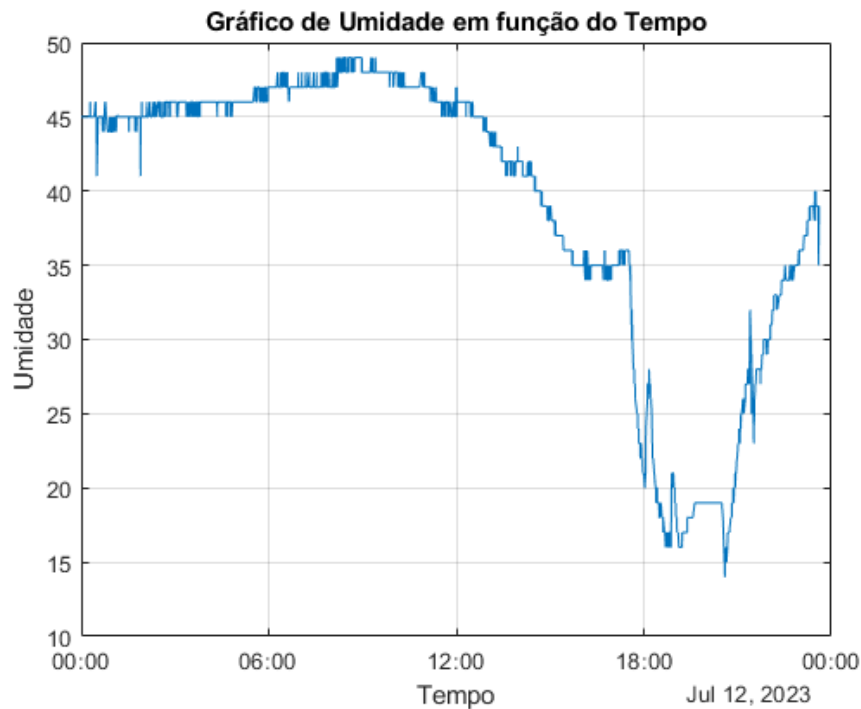


Figura 27 – Gráfico do sensor de umidade em função do tempo no dia 12/07/2023.

4.4 Monitoramento de Entradas Analógicas

As entradas analógicas foram testadas com uma fonte de bancada controlável e calibradas durante a realização dos testes, porém, ao instalar no ambiente real, o sensor de tensão AC da rede elétrica apresentou problemas. Portanto, não foi possível realizar medidas no ambiente para coletar dados de tensão analógica.

4.5 Monitoramento de Entradas Digitais

As entradas digitais foram testadas com a ajuda de um sensor de presença infravermelho passivo (PIR), que permaneceu durante a maior parte dos testes com o *status* 0, indicando que não havia ninguém no abrigo. Houve uma mudança nesse *status* no dia 10 de julho de 2023, quando houve uma manutenção no aparelho de ar-condicionado. Dessa forma, o sensor alarmou presença durante o período em que a equipe esteve no local, como pode ser visto na Figura 28, validando os testes executados.

4.6 Envio de Comandos

Como visto na Figura 18, foi desenvolvido um módulo de *reset*, que quando acionado pela interface envia um comando SSH para a estação de telemetria e reinicia o receptor. Isto é bem útil, pois em muitos casos é necessário realizar uma viagem somente

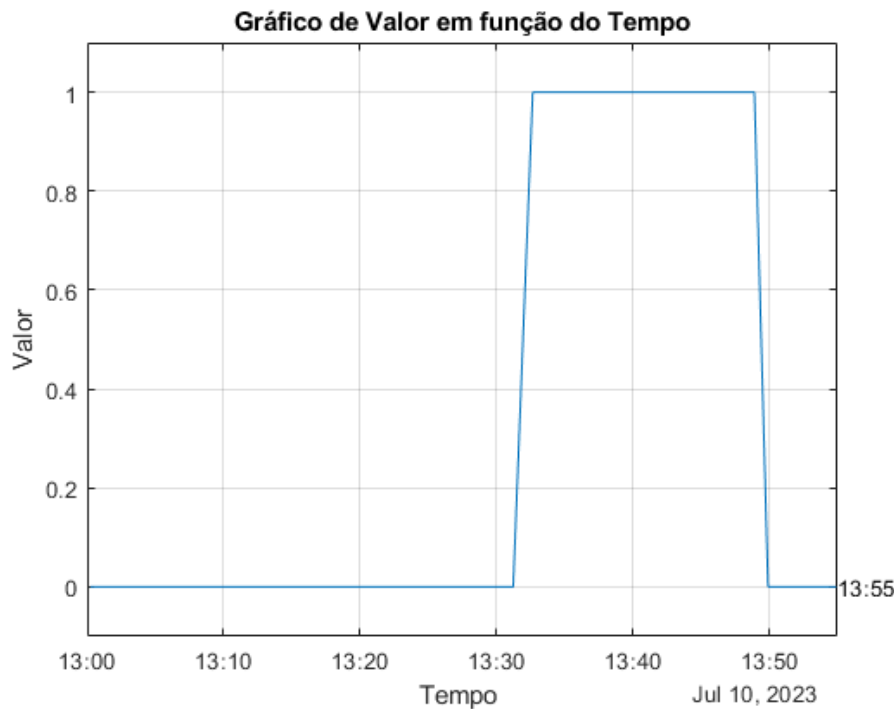


Figura 28 – Gráfico do sensor de presença em função do tempo no dia 10/07/2023.

para reiniciar o equipamento para o caso de algum travamento. Durante o período de testes foi realizado com sucesso um comando de *reset* para reiniciar o receptor, e após isso o sistema foi reestabelecido.

4.7 Conexão e Envio dos Dados

A conexão de dados foi feita com um modem 3G ZTE MF180S, utilizando um *chip* da operadora Vivo e configurado através do próprio sistema operacional Ubuntu Mate, logo, pode ser aprimorada para outras tecnologias como o 4g ou 5g apenas realizando a troca do modem. Porém, por conta de limitações da tecnologia 3G e da interferência gerada pelo transmissor de TV, por vezes a conexão se tornou instável, onde durante os dias de teste houve um dia inteiro em que o equipamento ficou fora do ar, aguardando o retorno do sinal por algum tempo.

Caso o aparelho esteja desconectado da internet, os dados salvos ficam armazenados no banco de dados, e ao voltar o dispositivo envia todos os dados armazenados de uma vez para que haja registros no servidor.

5 Conclusões e Passos Futuros

5.1 Considerações Finais

A necessidade de monitoramento remoto em emissoras de TV é de grande importância, por conta do seu alto custo de implementação e manutenção. O manuseio dos equipamentos deve ser feito com cuidado, sendo sempre monitorados. O sistema foi desenvolvido utilizando um *Raspberry Pi 3 model B+* para realizar o processamento e o envio dos dados. Foi conectado também um *Arduino Nano* para realizar a leitura de entradas analógicas, além de uma placa com relés para reiniciar equipamentos, um modem 3G para conexão com a internet, uma placa multiplexadora para expandir a quantidade de entradas analógicas, alguns periféricos para expandir as conexões do *Raspberry Pi* para o painel, uma fonte de 5V para alimentar o sistema e duas ventoinhas para arrefecimento. Todo o sistema foi montado em um gabinete de *rack* 1U, de fácil instalação em qualquer lugar. Também foi adicionada uma *webcam* para monitorar o ambiente, um sensor de temperatura e umidade do tipo DHT11, um sensor de presença, um *switch ethernet* e um módulo de *reset* para reiniciar equipamentos.

O custo total foi em torno de 1400 reais, o que se torna uma solução viável, visto que o valor do deslocamento por vezes pode superar o custo do equipamento, além disso, é necessário um custo fixo com a manutenção do servidor e da internet móvel que pode chegar a casa dos 150 reais mensais.

Com base nos testes, foi possível obter resultados satisfatórios, onde o sistema respondeu bem aos dados inseridos. Utilizando vários sensores em um ambiente real, foi possível armazenar os dados e acessá-los através da interface gráfica, sendo possível visualizar os dados com o tempo, executar as funções propostas inicialmente, e monitorar uma retransmissora numa cidade que não possuía nenhum equipamento do tipo.

5.2 Passos Futuros

Entre os próximos passos, sugere-se a melhoria da comunicação com a internet, que pode ser feita através de um modem com suporte a tecnologias mais recentes como o 4G ou 5G. Além disso, pode-se posicionar uma antena externa fora do abrigo para evitar interferências eletromagnéticas.

Outro ponto interessante é a melhoria da rede VPN utilizada, que só permite dois acessos, podendo futuramente utilizar um serviço pago que consiga suportar outros acessos simultâneos. Além disso, pode-se usar um servidor mais potente para reduzir o

tempo de latência ao realizar uma consulta no banco de dados e também um domínio na internet para tornar o uso mais fácil.

Por fim, um ultimo ponto a ser implementado é o redirecionamento dos IPs dos receptores e transmissores conectados a estação de telemetria para a VPN, de maneira que seja possível acessar os equipamentos mesmo a distância, tornando a manutenção dos dispositivos mais fácil.

Referências

- Agência Nacional de Telecomunicações (Anatel). *Televisão Digital*. 2015. Disponível em: <<https://www.gov.br/anatel/pt-br/regulado/radiodifusao/televisao-digital>>. Citado na página 19.
- ARDUINO. *Arduino Nano — Arduino Official Store*. 2008. Disponível em: <<https://store.arduino.cc/products/arduino-nano>>. Citado 2 vezes nas páginas 11 e 30.
- BARROS, E.; CAVALCANTE, S. Introdução aos sistemas embarcados. *Artigo apresentado na Universidade Federal de Pernambuco-UFPE*, p. 36, 2010. Citado na página 23.
- CARDOSO, M. A. C. Tecnologias 3g: uma visão prática. 2008. Citado na página 30.
- CARRO, L.; WAGNER, F. Capítulo 2 das jornadas de atualização em informática.”. *IN: XXII JAI*, p. 45–94, 2003. Citado na página 23.
- COMUNICAÇÕES, M. das. *Emissoras de Rádio e TV têm até 16 de novembro para solicitar parcelamento de pagamento de débitos vencidos — Ministério das Comunicações*. 2022. Disponível em: <<https://www.gov.br/mcom/pt-br/noticias/2022/outubro/emissoras-de-radio-e-tv-tem-30-dias-para-solicitar-parcelamento-de-pagamento-de-outorgas>>. Citado na página 24.
- DATE, C. J. *Introdução a sistemas de bancos de dados*. [S.l.]: Elsevier Brasil, 2004. Citado na página 31.
- Instituto Brasileiro de Geografia e Estatística (IBGE). *Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal 2021*. 2022. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101963_informativo.pdf>. Citado na página 19.
- KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet: Uma Abordagem Top-Down*. 6. ed. [S.l.]: Pearson, 2014. Citado 4 vezes nas páginas 11, 26, 27 e 28.
- MATTOS, A. N. de. *Telemetria e conceitos relacionados*. [S.l.]: Alessandro Nicoli de Mattos, 2004. Citado na página 25.
- MAURO, D.; SCHMIDT, K. *Essential SNMP: Help for System and Network Administrators*. [S.l.]: "O'Reilly Media, Inc.", 2005. Citado na página 26.
- NANO, A. *Arduino nano. A MOBICON Company*, 2018. Citado na página 30.
- PI, R. *Buy A raspberry pi 3 model B+*. 2018. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>>. Citado na página 29.
- PI, R. *Raspberry Pi 3 Model B+ – Raspberry Pi*. 2018. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>>. Citado 2 vezes nas páginas 11 e 29.

PISKA, C. M.; PEREIRA, M. C.; NASCIMENTO, M. do. Telesite: monitoramento e gerência de estações remotas de tv e telecom. Universidade Tecnológica Federal do Paraná, 7 2018. Disponível em: <<http://repositorio.utfpr.edu.br:8080/jspui/handle/1/9710>>. Citado na página 20.

ROCHA, L. C. da et al. Índice de popularidade das linguagens de programação e frameworks front-end e back-end nas fábricas de software da região de belo horizonte. *Computação & Sociedade*, v. 1, n. 1, 2019. Citado na página 32.

SILVA, R. P. Telemetria de estações retransmissoras de tv. Universidade Estadual Paulista (Unesp), 2022. Citado 4 vezes nas páginas 11, 24, 25 e 26.

TANENBAUM, A. S. *Redes de Computadores*. 5. ed. [S.l.]: Pearson, 2011. Citado 2 vezes nas páginas 26 e 31.