



**BRENDA RAYANE SANTOS E OTÁVIO RIBEIRO
MATIOLI**

**DESENVOLVIMENTO E APLICAÇÃO DE SOLUÇÃO PARA
COMUNICAÇÃO EM USINAS FOTOVOLTAICAS**

LAVRAS – MG

2023

BRENDA RAYANE SANTOS E OTÁVIO RIBEIRO MATIOLI

**DESENVOLVIMENTO E APLICAÇÃO DE SOLUÇÃO PARA COMUNICAÇÃO EM
USINAS FOTOVOLTAICAS**

Trabalho de Conclusão de Curso apresentada à
Universidade Federal de Lavras, como parte das
exigências do Curso de Engenharia de Controle
e Automação na modalidade de Concepção
Básica, para a obtenção do título de Bacharel.

Prof.Dr. Márcio Montagnana Vicente Leme

Orientador

LAVRAS – MG

2023

BRENDA RAYANE SANTOS E OTÁVIO RIBEIRO MATIOLI

**DESENVOLVIMENTO E APLICAÇÃO DE SOLUÇÃO PARA COMUNICAÇÃO EM
USINAS FOTOVOLTAICAS**

Trabalho de Conclusão de Curso apresentada à
Universidade Federal de Lavras, como parte das
exigências do Curso de Engenharia de Controle
e Automação na modalidade de Concepção
Básica, para a obtenção do título de Bacharel.

APROVADA em 18 de Dezembro de 2023.

Prof.Dr. Márcio Montagnana Vicente Leme	UFLA
Prof.Dr. Arthur de Miranda Neto	UFLA
Álvaro Henrique Alves Maia	UFLA

Prof.Dr. Márcio Montagnana Vicente Leme
Orientador

**LAVRAS – MG
2023**

AGRADECIMENTOS

Agradeço a todos os que colaboraram de alguma forma ao desenvolvimento deste trabalho, desde professores que colaboraram com nossa formação, aos pais que estiveram nos suportando em nossas dificuldades. Destacamos um agradecimento especial aos professores Carlos Eduardo Castilla Alvarez e Marcio Montagnana Vicente Leme que nos deram a oportunidade de desenvolver este projeto e somar ao nosso conhecimento e formação como engenheiros.

RESUMO

Atualmente a busca por energias renováveis mostra-se constante, a sua utilização gera fontes quase infinitas de energia no qual causam menor degradação ao meio ambiente em sua aplicação. A energia fotovoltaica é uma das propostas de geração de energia que possui potencial de crescimento e dominância no mercado. O presente trabalho visa realizar, em formato de concepção básica, a análise e descrição do processo de desenvolvimento de uma solução de comunicação industrial, com foco específico em usinas fotovoltaicas. A solução foi concebida por meio da integração de um processo decisório fundamentado em pesquisas laboratoriais, visando identificar as soluções com o melhor custo-benefício e possibilidade de aplicação. Esse processo foi combinado com o desenvolvimento de um código em Python para a captura de dados utilizando o protocolo Modbus, empregando padrões de comunicação RS-485 e TCP/IP. Além disso, implementou-se a funcionalidade de envio dos dados para um banco de dados online, proporcionando a capacidade de consulta e acesso remoto. Todo o desenvolvimento foi orientado pela busca de uma solução adequada a ser uma nova alternativa presente no mercado. Nesse contexto, a proposta deste trabalho é oferecer uma alternativa funcional e acessível, contribuindo para os processos relacionados à monitoramento de usinas fotovoltaicas.

Palavras-chave: Usinas fotovoltaica; Protocolo Modbus; Banco de dados online; Comunicação Industrial

ABSTRACT

Currently, the search for renewable energies remains constant, and their utilization generates nearly infinite sources of energy that cause less environmental degradation in their application. Photovoltaic energy is one of the proposed energy generation methods that has the potential for growth and dominance in the market. This work aims to perform, in a basic conceptual format, the analysis and description of the development process of an industrial communication solution, with a specific focus on photovoltaic plants. The solution was conceived through the integration of a decision-making process based on laboratory research, aiming to identify solutions with the best cost-benefit and application possibilities. This process was combined with the development of Python code for data capture using the Modbus protocol, employing RS-485 and TCP/IP communication standards. Additionally, the functionality of sending data to an online database was implemented, providing the capability for querying and remote access. The entire development was guided by the search for a suitable solution to be a new alternative in the market. In this context, the proposal of this work is to offer a functional and affordable alternative, contributing to processes related to the monitoring of photovoltaic plants.

Keywords: Photovoltaic power plants; Modbus protocol; Online database; Industrial communication

LISTA DE FIGURAS

Figura 2.1 – Mensagem Modbus	11
Figura 2.2 – Arduino Mega	17
Figura 2.3 – Nobreak	17
Figura 2.4 – DPS	18
Figura 2.5 – Conversor rs485	19
Figura 2.6 – ESP8266	19
Figura 2.7 – Protótipo de interface homem-máquina	20
Figura 4.1 – Fluxo da alternativa 1	26
Figura 4.2 – Fluxo da alternativa 2	27
Figura 4.3 – Fluxo da alternativa 3	28
Figura 6.1 – Placas Fotovoltaicas	33
Figura 6.2 – Inversor	34
Figura 6.3 – Cabos com blindagem e conectores	34
Figura 7.1 – Fluxo de dados	39
Figura 7.2 – Login	39
Figura 7.3 – Tabela com os dados	40
Figura 7.4 – Continuação da tabela com os dados	40
Figura 7.5 – Gráficos da Potência	41
Figura 7.6 – Gráficos da Tensão	41
Figura 7.7 – Gráficos da Corrente	42
Figura 7.8 – Gráficos da Energia	42

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Identificação do problema	8
1.2	Objetivos	9
2	Fundamentação	10
2.1	Modbus RTU	10
2.1.1	RS232	11
2.1.2	RS485	12
2.2	Modbus TCP	14
2.3	Raspberry	15
2.4	Arduino	16
2.5	Nobreak	17
2.6	DPS	18
2.7	Conversor RS485	18
2.8	ESP8266	19
2.9	Python	19
2.10	Sistema SCADA	20
2.11	Ferramentas de monitoramento	20
2.12	Inversores	21
2.13	Combiners Box	21
2.14	Estação solarimétrica	22
2.15	<i>Trackers</i>	22
3	DEFINIÇÃO DO PROBLEMA	24
3.1	Captação de dados	24
3.2	Criação de uma interface homem máquina	24
3.3	Alimentação de um banco de dados com informações captadas	25
3.4	Desenvolvimento de uma interface acessível em diferentes localidades	25
3.5	União das funcionalidades e aplicação em usinas	25
4	SOLUÇÕES PROPOSTAS	26
4.1	Alternativa 1: Raspberry e Python para coleta dos dados	26
4.2	Alternativa 2: Arduino para coleta dos dados	27
4.3	Alternativa 3: Utilização de um sistema SCADA comercial	28

5	Análise das soluções	30
5.1	Custos	30
5.2	Capacidade	31
6	Implementação do Sistema	32
6.1	Materiais e metodologia	32
6.1.1	Sistema fotovoltaico	32
6.1.1.1	Placas fotovoltaicas	33
6.1.1.2	Inversor	33
6.1.2	Conexão e instalação do sistema	34
6.1.3	Extração dos dados	35
6.1.3.1	Captação dos dados dados inversores	35
6.1.3.2	Captação dos dados da estação solarimétrica	36
6.1.3.3	Captação dos dados dos Trackers	36
6.1.4	Tratamento dos dados	37
6.1.5	Envio dos dados para o banco de dados	37
7	Resultados	39
8	CONCLUSÃO	43
9	Apêndice	47

1 INTRODUÇÃO

1.1 Identificação do problema

A energia solar emergiu nos últimos anos com intuito de trazer energia limpa e renovável como solução para reduzir os impactos ambientais associados às mudanças climáticas, contribuindo para uma sociedade mais sustentável (BONDARIK; PILATTI; HORST, 2018). Dessa forma, o número de usinas fotovoltaicas tem aumentado consideravelmente, impulsionado pelo processo de conversão da luz solar em eletricidade. Nesse processo ocorre a conversão da luz solar em energia elétrica através de células fotovoltaicas.

A eletricidade gerada por essas células é inicialmente produzida como corrente contínua (CC). Para torná-la compatível com as redes elétricas convencionais, é necessário um dispositivo essencial conhecido como inversor. Ele desempenha um papel crucial, convertendo corrente contínua em corrente alternada (CA). Junto a ele é utilizado combiner box, neste equipamento contém dispositivos de segurança para que não ocorra sobrecarga ou curto-circuito no inversor (MODEL, 2021). Posteriormente, essa energia elétrica convertida é direcionada para a rede elétrica ou para o consumo local.

Para monitorar e registrar a quantidade de energia produzida pela usina fotovoltaica, bem como a quantidade de energia consumida da rede, utiliza-se um medidor de energia bidirecional. Esse dispositivo é fundamental para acompanhar o desempenho do sistema, que registra tanto a energia gerada quanto a energia recebida da rede (MACHADO; MIRANDA, 2015).

Além das *combiners box* e dos inversores, uma usina fotovoltaica conta também com a presença de uma estação meteorológica que possui sensores para obter a velocidade do vento, umidade do ar, radiação e temperatura local. Outro equipamento importante são os *trackers*, que são equipamentos motorizados que utiliza apenas um eixo para seguir parcialmente o sol. Dessa forma, as placas solares conseguem aproveitar mais a incidência solar (GUARNIERI et al., 2017).

No entanto, a eficiência e a confiabilidade dessas usinas dependem crucialmente da coleta e análise de dados precisos e abrangentes. Nesse contexto, a importância dos dados, em tempo real, para o monitoramento de usinas fotovoltaicas é inegável, uma vez que desempenham um papel fundamental na otimização do desempenho, na manutenção preventiva e na verificação e identificação de falhas.

As usinas fotovoltaicas costumam adotar inversores, *combiners box* e uma estação meteorológica, podendo eles ser equipados com o protocolo de comunicação Modbus RTU. Além disso, as usinas podem ser equipadas com *trackers*, que utilizam como protocolo de comunicação o Modbus TCP-IP. Estes são protocolos de comunicações amplamente utilizados na estruturação de mensagens para a transferência de dados, abrangendo informações tanto discretas quanto analógicas, entre dispositivos microprocessados. O protocolo Modbus integra mecanismos para identificação e correção de erros de transmissão. Esse protocolo opera com um modelo de comunicação mestre-escravo/ cliente-servidor, em que um único dispositivo mestre/cliente tem o privilégio de iniciar as comunicações, enquanto os dispositivos escravos/servidor respondem prontamente fornecendo os dados solicitados pelo mestre ou executando tarefas específicas conforme solicitado. Essa abordagem é essencial para assegurar uma comunicação robusta e confiável entre os dispositivos (FERREIRA, 2018).

Essa tecnologia desempenha um papel substancial na implementação de sistemas SCADA (*Supervisory Control and Data Acquisition*), os quais capacitam a aquisição, monitoramento e rastreamento de dados de forma eficiente e precisa.

No entanto, extrair esses dados gera desafios, sendo o principal deles, a estabilidade da extração, pois como os inversores não possuem um banco de dados, então o dado não lido é um dado perdido. Deste modo, a estabilidade de um sistema de extração de dados é um ponto crucial para o processo de extração.

1.2 Objetivos

O presente trabalho possui como objetivo realizar uma análise de sistemas de captação de dados em usinas fotovoltaicas. Realizando uma análise de custos voltadas a aplicação básica em laboratório.

Tem como objetivo secundário o monitoramento de uma bancada de trabalho com funcionalidades voltadas para a produção de energia fotovoltaica, para isso busca-se desenvolver e implementar um sistema de captação de dados do inversor e armazená-los em um banco de dados, gerando a possibilidade análises e ampliação para usinas fotovoltaicas reais.

2 FUNDAMENTAÇÃO

O presente capítulo busca descrever e apresentar as bases de conhecimento voltadas ao desenvolvimento da solução proposta pelo trabalho. São explorados conhecimentos voltados ao protocolo de comunicação, padrão de comunicação e equipamentos utilizados.

2.1 Modbus RTU

O protocolo Modbus é amplamente utilizado e até os dias atuais como um dos padrões de comunicação dominantes na indústria. Foi desenvolvido pela empresa Modicon, líder no mercado de Controladores Lógico Programáveis (CLP), em 1979 (THOMAS, 2008). O objetivo do Modbus é estabelecer comunicação entre dispositivos, estabelecendo uma relação Mestre-Escravo ou Cliente-Servidor, termos que foram cunhados no início do protocolo e são utilizados até hoje para descrever esse tipo de comunicação. Esse modelo de comunicação pode ser aplicado a dois ou mais dispositivos, desde que apenas um deles seja designado como mestre, e o número de escravos é limitado a 247 dispositivos (SOUZA, 2008).

Nesse modelo de comunicação, cada dispositivo envolvido desempenha um papel específico. O mestre é geralmente um dispositivo controlador, responsável por tomar decisões ao longo do processo industrial, como um painel de controle, um computador industrial ou micro-controladores. Os escravos, por outro lado, incluem sensores, atuadores e outros participantes do processo industrial. O mestre não requer um endereço, enquanto os escravos são identificados por endereços que variam de 1 a 247. O endereço 0 é reservado para transmissões de difusão (*Broadcast*), destinadas a todos os dispositivos na comunicação (THOMAS, 2008).

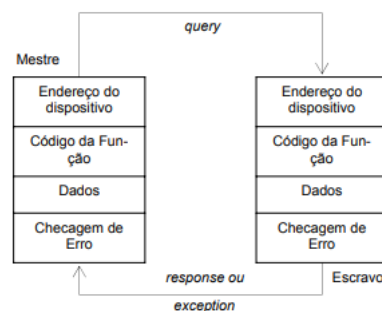
O modelo de comunicação Modbus é eficiente, pois os papéis dos dispositivos são claramente definidos. O mestre é responsável pelo envio principal de informações e é o único dispositivo capaz de iniciar a comunicação. Os escravos respondem apenas às solicitações do mestre (HUIJSING et al., 2008). Para ilustrar essa situação, considere um termômetro que monitora a temperatura de um processo industrial. Dentro desse sensor, podem ser definidos limites de operação. Se o termômetro for configurado como um escravo na comunicação e detectar uma temperatura acima do limite, ele não pode informar o controlador do processo (mestre) a menos que o mestre envie uma solicitação de leitura da temperatura para o escravo.

O Modbus é caracterizado por sua robustez e eficiência, em grande parte devido ao uso de um mapa de registradores. Esse mapa pré-determina códigos para a comunicação, agilizando

a troca de dados entre dispositivos e reduzindo o risco de erros causados por solicitações mal compreendidas. Portanto, ao estabelecer uma comunicação Modbus entre dispositivos, o programador responsável pela implementação da rede deve dominar o mapa de registradores para garantir uma comunicação eficiente na rede (SOUZA, 2008).

A troca de dados entre os participantes da rede Modbus ocorre por meio de dois modos de transmissão: ASCII (*American Code for Information Interchange*) e RTU (*Remote Terminal Unit*). O programador pode escolher o modo de transmissão ao iniciar o protocolo. O modo ASCII utiliza pacotes de 10 bits e tem a vantagem de tolerar intervalos prolongados entre o envio de dados na mesma mensagem, prevenindo erros em situações de comunicação mais lenta. Por outro lado, o modo RTU emprega o envio de palavras no formato hexadecimal e requer 11 bits por mensagem, com 8 bits dedicados aos dados (um a mais do que no modo ASCII), proporcionando uma comunicação mais densa e eficiente. A escolha do modo de transmissão depende das necessidades do processo de comunicação (SOUZA, 2008).

Figura 2.1 – Mensagem Modbus



Fonte: Nascimento e Lucena (2003).

2.1.1 RS232

O RS-232 é um padrão de protocolo de comunicação serial, que busca padronizar a comunicação entre um terminal de dados ou em inglês *Data Terminal Equipment* (DTE), e um comunicador de dados, também chamado de *Data Communication Equipment* (DCE), esse modelo de comunicação foi padronizado pela empresa *Eletronics Industries Association* (EIA) em 1969 (THE... , 2023). A Modicon realizou sua primeira implementação do protocolo Modbus usando o padrão de comunicação RS-232-C realizando comunicação ponto a ponto nos seus CLPs. Deste modo, todos os envolvidos na comunicação necessitariam de portas seriais para que a comunicação pudesse ser realizada, cada um dos escravos deveriam ter uma porta se-

rial, enquanto o mestre deveria ter o número de portas seriais suficientes para atender todos os escravos.

Dentre as vantagens de se utilizar o padrão de comunicação RS232, temos uma grande flexibilidade, visto que diversos equipamentos eletrônicos possuem como configuração padrão uma porta de comunicação serial, possibilitando sua aplicação em equipamentos que variam de computadores a equipamentos de automação industrial (DIFFERENCE... , 2023).

A velocidade de transmissão que é utilizada na comunicação RS-232 é variável, podendo ser utilizado em aplicações que necessitam de baixa a alta velocidade de transmissão.

Dentre as desvantagens do padrão de comunicação RS-232 tem-se a situação citada anteriormente, em que um dispositivo utilizado como mestre na rede de comunicação, necessitaria de portas seriais o suficiente para atender todos os escravos, gerando assim uma limitação para utilização deste padrão de comunicação. Visto que em uma grande companhia em que necessitava-se controlar uma grande quantidade de equipamentos, o controlador deveria possuir uma quantidade muito grande de portas, o que geralmente ultrapassava as limitações de espaço e geravam a necessidade de compra de novos equipamentos. A comunicação utilizando o padrão RS-232 mostrava-se precisa, porém trazia consigo outra limitação, a distância entre os equipamentos, os cabos de comunicação serial por padrão tem um pequeno tamanho (15 metros), para aplicações de rotina e em indústrias de pequena escala, podem parecer adequadas, porém em indústrias de grande porte e de setores específicos gera a necessidade da utilização de um controlador muito próximo da região de atuação dos equipamentos que fornecerão os dados (DTE). Em ambientes industriais como usinas de geração de energia solar, o ambiente de atuação dos inversores costuma ter uma temperatura inapropriada para computadores industriais, além de serem expostos a fenômenos climáticos, como chuva, ventos, etc. Deste modo, tem-se mais uma limitação para o uso deste padrão.

2.1.2 RS485

O RS485 é um padrão de comunicação serial de dados, desenvolvido para uso e aplicação em redes industriais. Este padrão de comunicação foi desenvolvido pela *Electronic Industry Association* em 1983. Estabeleceu-se um meio de comunicação que se diferenciava do RS232 na forma de comunicação e mostrava-se capaz de resolver alguns de seus "problemas", tais como a curta distância que o padrão de comunicação permitia, a capacidade de resistir a inter-

ferências eletromagnéticas (ponto de grande utilidade em ambientes industriais). Deste modo, criou-se um padrão que se assemelhava ao RS232, porém funcionava com a utilização de um par de fios; os sinais poderiam ser captados através das diferenças de tensão existentes entre esse par de fios (MARAIS, 2008).

Por meio do modo de transmissão diferencial, o padrão de comunicação apresentou alguns pontos positivos e alguns pontos negativos em relação a padrões já existentes. Como o padrão de comunicação se baseia em transmissão diferencial, ao utilizar o padrão deve-se atentar apenas às características elétricas especificadas, modos de operação em rede. No entanto, o padrão de comunicação não especifica ou recomenda o uso de protocolos, apesar da base de estudo do presente trabalho se basear em comunicações relacionadas ao protocolo Modbus. O padrão de comunicação RS485 oferece extrema versatilidade quanto a protocolos de comunicação. Outra de suas vantagens, como citado anteriormente, é sua resistência a interferências eletromagnéticas, o que em ambientes industriais traz enorme confiabilidade e, em diversos casos, viabiliza o padrão de comunicação para realização de certos processos de trocas de mensagem. A construção deste padrão de comunicação também permite que até 32 dispositivos estejam conectados ao mesmo barramento, possibilitando a utilização de controladores que ocupam menos espaço físico dentro de um ambiente industrial ou de controle de uma planta industrial. A distância que é possível atingir utilizando este tipo de comunicação representa um de seus pontos positivos, possibilitando que a comunicação utilizando o padrão RS485 ocorra em até 1200 metros de distância entre os dispositivos, sanando alguns dos problemas citados anteriormente voltados à presença de estações de controle em ambientes hostis.(DIFFERENCE. . . , 2023).

Na contraparte dos pontos citados anteriormente, a comunicação pelo padrão RS485 ocorre necessariamente com pares de cabos, o que leva à necessidade de cabeamento específico para realização da comunicação. Além disso, quando a comunicação ocorre em ambientes externos, observa-se a necessidade de cabos blindados e preparados para manter a comunicação, gerando custos diretamente ligados à aplicação deste padrão de comunicação. Quando falamos a respeito da distância máxima suportada pelo padrão de comunicação, devemos observar também que, à medida que a distância entre os dispositivos que se comunicam aumenta, a velocidade de comunicação reduz, podendo a mesma variar entre 10 megabits por segundo, para distâncias de até 12 metros entre os terminais, a taxas de 100 kilobits por segundo, em distâncias próximas aos 1200 metros (DIFFERENCE. . . , 2023).

O padrão de comunicação RS485 ainda possui a flexibilidade de disponibilizar 3 modos de transmissão a serem escolhidos para utilização, sendo eles o *Simplex*, modo no qual a transmissão ocorre em sentido unidirecional, sem a capacidade de retorno do receptor da mensagem; modo *Half-duplex*, em que ambos os dispositivos envolvidos na comunicação podem transmitir e receber mensagens, mas essas ações não podem ser realizadas ao mesmo tempo por diferentes dispositivos. Já o modo *Full-duplex* permite que tanto o dispositivo transmissor quanto o receptor possam enviar e receber mensagens ao mesmo tempo, gerando assim versatilidade para quem utiliza o padrão de comunicação escolher entre o modo que mais atende suas necessidades (WOJAHN, 2023).

2.2 Modbus TCP

O protocolo Modbus, devido à versatilidade de sua aplicação, pode ainda utilizar a base de comunicação desenvolvida para suportar o protocolo TCP/IP (protocolo de comunicação utilizado em redes de internet).

O protocolo TCP (*Transmission Control Protocol*) é um protocolo de comunicação criado para utilização na camada de transporte de dados, situando-se entre a camada de rede e a camada de aplicativo. Geralmente, isso cria a necessidade do protocolo ser analisado juntamente com o protocolo de rede IP. O protocolo TCP é orientado a conexão e tem como principal objetivo criar, na camada de transporte, uma conexão entre dois TCPs para possibilitar controle de fluxo e de erros no nível de transporte (XUAN; YONGZHONG, 2019).

Toda essa questão de conexão entre dois dispositivos TCP traz ao protocolo de comunicação questões voltadas à segurança dos dados transmitidos. Pode-se dizer que a conexão TCP cria um tubo imaginário entre os participantes da comunicação e, com isso, permite comunicação segura e eficaz entre os envolvidos, com verificações de erros e a passagem de um fluxo de bytes, diferenciando-se dos padrões de comunicação concorrentes em termos de aplicações. Apesar do benefício relacionado à confiabilidade dos dados, em comparação com o protocolo UDP (*User Datagram Protocol*), o protocolo TCP apresenta troca de dados mais lenta. A numeração dos bytes e os processos de verificação geram tal lentidão comparada a outros protocolos mais simples (OLAYA; LÓPEZ; MORENO, 2004).

Em combinação a esses fatores, para conectar dois dispositivos de maneira única e evitar confusões, usa-se em conjunto com o protocolo TCP os endereços IP, que podem ser tratados

como um número de identificação de máquinas, sendo único e permitindo que dispositivos possam ser reconhecidos em uma dada rede de comunicação. O endereço IP consiste em 32 bits, o que permite a numeração de 4.294.967.296 dispositivos diferentes. Apesar de existir teoricamente, isso não ocorre nem mesmo para a rede de internet que utiliza esse protocolo. Sendo assim, para uma rede de comunicação industrial, o TCP em conjunto com o IP oferece possibilidade de conexão entre dispositivos mais do que suficiente para a maioria das indústrias existentes (OLAYA; LÓPEZ; MORENO, 2004).

Somando as bases teóricas desenvolvidas a respeito do TCP e do IP, podemos citar que o Modbus TCP une essas qualidades e adapta ao seu padrão de comunicação a forma de atuar do TCP, permitindo a criação de uma rede de comunicação com dispositivos que usam a identificação IP e o modo de comunicação com fluxo de dados constantemente testado, utilizado pelo TCP. Este modelo de comunicação pode ser utilizado utilizando o cabeamento Ethernet e fibra óptica, que é padrão para a comunicação TCP em aplicações de internet. O modelo Modbus TCP ainda carrega consigo a estrutura de mestre e escravo, discutida anteriormente, o que organiza a comunicação de forma padrão para redes que possuem um controlador entre seus dispositivos e permite maior adaptabilidade seguindo os padrões de comunicação comumente utilizados.

2.3 Raspberry

O Raspberry Pi foi desenvolvido pela Raspberry Pi Foundation, uma organização sem fins lucrativos do Reino Unido. Seu principal objetivo era promover o ensino de programação e informática para crianças, especialmente em escolas públicas. No entanto, devido ao seu grande sucesso, o pequeno computador de placa única (Single Board Computer ou SBC) conquistou uma ampla variedade de usos em projetos, incluindo automação e aplicações industriais (JUCÁ; PEREIRA, 2018).

A fundação lançou vários modelos de Raspberry Pi, incluindo o Raspberry Pi 5, Raspberry Pi Zero 2W, Raspberry Pi 4 Model B, Raspberry Pi 3 Model A+, Raspberry Pi 3 Model B+, Raspberry Pi 3 Model B, Raspberry Pi 1 Model B+, Raspberry Pi 1 Model A+, Raspberry Pi Zero W e Raspberry Pi Zero (ABOUT... , 2023b).

Quando foi lançado, o Raspberry Pi foi considerado o menor computador do mundo, com o tamanho de um cartão de crédito. Apesar de seu tamanho compacto, ele é capaz de rea-

lizar todas as tarefas de um computador convencional, como reprodução de vídeos, navegação na internet e processamento de textos. Além disso, é uma ferramenta poderosa para aprender a programar em linguagens como C, Python e Java.

O microcomputador possui fontes de alimentação, comunicação serial e pinos de entrada e saída. Desta forma, ele também tem a capacidade de interagir com o mundo exterior, utilizando sensores e atuadores, o que possibilita o desenvolvimento de projetos de sistemas embarcados (ABOUT. . . , 2023b).

O Raspberry Pi não possui BIOS ou configurações de setup tradicionais. Todas as configurações relacionadas ao hardware e ao processo de inicialização são feitas em um arquivo chamado 'config.txt', que está localizado na raiz do cartão SD.

Ele é compatível com várias distribuições Linux, incluindo Debian (Raspbian), Fedora e Arch Linux. Cada uma dessas distribuições tem suas próprias características e vantagens. Por exemplo:

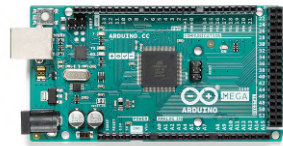
- Raspbian (baseado no Debian): É a distribuição oficial da Raspberry Pi Foundation e é otimizada para o Raspberry Pi. Possui uma ampla gama de pacotes e é uma escolha popular para iniciantes.
- Fedora: Oferece uma experiência Linux mais voltada para desktop, com suporte a várias aplicações e recursos modernos.
- Arch Linux: É conhecido por ser altamente personalizável e adequado para usuários avançados que desejam um controle total sobre sua instalação.

2.4 Arduino

O Arduino é uma plataforma open source de prototipagem eletrônica que é amplamente utilizada para projetos interativos utilizando sensores, atuadores, botões dentre outros componentes eletrônicos. O controle da placa é realizada através de um conjunto de instruções enviadas ao microcontrolador. Dessa forma, é utilizado a linguagem de programação do arduino baseada em Wiring, sendo ela um estrutura de código aberto para microcontroladores (WIRING, 2023), e a IDE(Integrated Development Environment - Ambiente de Desenvolvimento Integrado) do Arduino para realizar o desenvolvimento do conjunto de intruções.

O microcontrolador surgiu no Ivrea Interaction Design Institute com o principal intuito para alunos que não tinham conhecimentos prévios em programação e eletrônica. Porém, com a chegada em uma comunidade mais ampla ele se adaptou para os novos desafios. Atualmente, é empregado para aplicações IoT(Internet of Things), ambientes embarcados, automação residencial, robôs, extração de dados utilizando o protocolo de comunicação Modbus, dentre outras aplicações (ABOUT. . . , 2023a).

Figura 2.2 – Arduino Mega



Fonte: Arduino, 2023

2.5 Nobreak

O nobreak é um dispositivo de proteção para equipamentos eletrônicos. Ele protege e mantém o funcionamento dos dispositivos em períodos com oscilação ou ausência de energia elétrica. Devido à sua bateria interna, o nobreak consegue manter dispositivos eletrônicos em funcionamento mesmo quando não há energia disponível. Além disso, o nobreak atua como um estabilizador, evitando desligamentos inesperados dos equipamentos (CARVALHO, 2013).

A utilização desse dispositivo em conjunto com equipamentos que possuem circuitos sensíveis contribui para preservar a vida útil e a qualidade desses equipamentos por um período mais longo.

Figura 2.3 – Nobreak



Fonte: Dell, 2023

2.6 DPS

Os DPS, ou Dispositivos de Proteção contra Surtos, são equipamentos utilizados para garantir a segurança de circuitos elétricos e eletrônicos. Eles desempenham um papel crucial na proteção de circuitos contra surtos de energia, que podem ser gerados por diversas causas em usinas fotovoltaicas, como raios, surtos de energia comuns em áreas rurais, defeitos em outros equipamentos presentes nas usinas, entre outros. Portanto, a utilização de um DPS para proteger o sistema de controle da usina é de extrema importância. Esse dispositivo ajuda a prevenir danos causados por picos de tensão, garantindo o funcionamento seguro e confiável dos equipamentos elétricos e eletrônicos na usina fotovoltaica (O... , 2023).

Figura 2.4 – DPS



Fonte: Soprano, 2023

2.7 Conversor RS485

Os conversores RS485 são dispositivos que permitem contornar as limitações da conexão com o padrão RS485, possibilitando a conversão desse sinal para USB, RS232 ou outros padrões de comunicação. Esse dispositivo torna-se útil quando *HUBs* ou microcomputadores utilizados para realizar o controle de equipamentos industriais necessitam dessa comunicação. Dessa forma, ao utilizar um conversor, é possível aproveitar efetivamente o padrão de comunicação RS485 em redes industriais, garantindo uma interface adequada para a integração de dispositivos em sistemas de controle industrial (INC., 2017).

Figura 2.5 – Conversor rs485

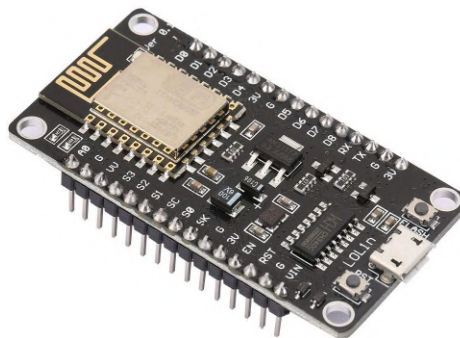


Fonte: Autores(2023)

2.8 ESP8266

O ESP8266 é um módulo, de baixo custo, System on Chip wi-fi que está no mercado desde 2014 e é produzido pela empresa Espressif Systems. Idealmente utilizado para desenvolvimentos IoT(Internet of Things- Internet das Coisas). Vários sistemas utilizam ele como módulo wifi para o arduino apesar que possuir processamento próprio (MEHTA, 2015).

Figura 2.6 – ESP8266



Fonte: NODEMCU, 2023

2.9 Python

O python é uma linguagem de programação amplamente utilizada no mercado, pode ser classificada como uma linguagem de alto nível e orientada a objetos. Por sua simplicidade e facilidade de uso, o Python é uma das linguagens mais utilizadas para aplicações e permite integrações com diversos sistemas e dispositivos (BORGES, 2014).

2.10 Sistema SCADA

Os sistemas SCADA (*Supervisory Control and Data Acquisition*) são considerados fundamentais para o monitoramento de processos industriais nos dias de hoje. Esses sistemas possuem uma combinação de ferramentas físicas e softwares que, quando unidos, possibilitam a extração e visualização de dados industriais em tempo real (MÜLLER et al., 2017).

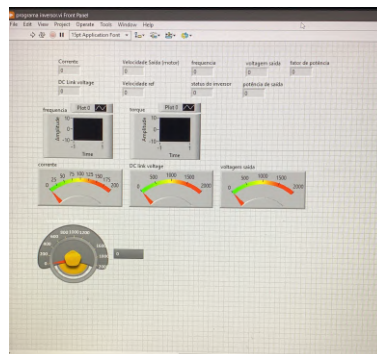
Os sistemas SCADA têm três componentes principais: uma rede de comunicação, um conjunto de equipamentos utilizados para coletar e enviar informações (podemos exemplificar como sensores), e um software de supervisão. Este software pode ter sua utilização limitada localmente ou estar disponível online. Dessa forma, o sistema SCADA oferece a qualquer aplicação industrial uma solução ideal para o seu acompanhamento (SANTOS; MARINATO, 2021). No entanto, é importante destacar que os sistemas SCADA costumam vir acompanhados de um alto custo de implementação, manutenção e utilização.

2.11 Ferramentas de monitoramento

No mercado, é possível encontrar ferramentas que proporcionam o desenvolvimento simplificado de um ambiente de monitoramento. Essas ferramentas geralmente contam com uma estrutura de programação gráfica, o que facilita o acesso para pessoas que não têm contato próximo com linguagens de programação. Ao permitirem a criação desses ambientes, essas ferramentas possibilitam o uso de protocolos de comunicação para a obtenção de dados e, conseqüentemente, sua apresentação em unidades de monitoramento (DZIALOSZYNSKI, 2015).

Estas soluções costumam apresentar custos médios para sua aplicação de forma direta, porém limitam o usuário e de forma indireta podem gerar grandes custos de implementação.

Figura 2.7 – Protótipo de interface homem-máquina



Fonte: Autores (2023)

2.12 Inversores

Os inversores utilizados em usinas de energia fotovoltaica são equipamentos utilizados para converter corrente contínua, gerada pelas placas fotovoltaicas, em corrente alternada, que será conectada a rede de distribuição. Tais equipamentos mostram-se um dos elos mais importantes da produção de energia fotovoltaica, carregando consigo diversas funções e diversos dados que podem ser captados e lidos, buscando-se monitorar o processo e encontrar pontos de força e de melhoria na produção da energia. Em conexão com estes equipamentos podem-se obter dados de corrente e tensão de entrada e saída, temperatura dos equipamentos, potência gerada, Energia gerada, qualidade da energia, alarmes relacionados ao funcionamento das placas e do próprio inversor(MODEL, 2021).

Por serem equipamentos de uso industrial, geralmente contam com a comunicação padrão RS485, com broadcast ativo para o primeiro endereço, baud rate de 9600 bit/s, 8 bits de dados e 1 bit de parada, e modo RTU, dentre os dados para cada um dos endereços lidos, pode-se encontrar dados em diferentes formatos, sendo eles *Unsigned int 16 bit* (inteiros que variam de 0 a 65535), *Signed int 16 bit* inteiros que variam de -32767 a +32767, *Unsigned int 32 bit* e *Signed int 32 bit*, sendo a leitura dos dados de 32 bits feita em duas séries de dados separadas na seguinte ordem de exemplo (ABCD) se transmitira C, D, A, B. Apesar de esse ser um padrão encontrado na maioria dos equipamentos do mercado, pode haver variações de acordo com a marca e modelo do equipamento (CO., 2018).

2.13 Combiners Box

As combiner box são equipamentos posicionados entre as placas solares e os inversores para proteger a parte de corrente contínua da ligação e por receberem corrente das placas e direcionar aos inversores geram informações que também podem ser consultadas geralmente via modbus RS485, de modo geral os equipamentos são da mesma marca do inversor, gerando um mesmo formato de dados e de consulta (MODEL, 2021).

2.14 Estação solarimétrica

A estação solarimétrica é uma das estruturas presentes nas usinas e tem como função abrigar diversos sensores, incluindo o anemômetro (sensor que mede a velocidade do vento), sensores de temperatura, sensor de radiação solar horizontal, sensor de radiação solar inclinada, sensor de radiação solar albedo (reflexão da radiação solar), sensor de umidade e sensor de precipitação. Usualmente estes sensores possuem comunicação serial com um dispositivo registrador de dados em campo (PONTES, 2021).

O dispositivo tem a capacidade de registrar dados por um período de tempo e pode ter conexão por meio do protocolo Modbus RS485 com uma central de controle que consulta os dados de todos os sensores conectados a ele. Cada sensor é conectado a uma entrada digital, analógica ou virtual (PONTES, 2021).

Os dados têm especificações para realização de leitura ou escrita e podem ser obtidos como binários, *Word High*, *Word Low*, inteiros de 16 bits com sinal, pontos flutuantes (*single precision*) e inteiros de 32 bits sem sinal. Cada um desses dados deve receber um tratamento adequado em sua leitura (LTDA., 2018).

2.15 Trackers

Os *Trackers* são motores responsáveis pela movimentação das placas solares, ajustando sua posição com base no movimento do sol ao longo do dia. Esses dispositivos precisam receber informações relacionadas à posição desejada e, como resultado, fornecer informações sobre seu ângulo, temperatura, condições de funcionamento, entre outros (GUARNIERI et al., 2017). Conforme os manuais dos equipamentos, são utilizadas algumas especificações para a coleta de dados.

A comunicação com os trackers, nos dispositivos pesquisados pelo mercado, ocorre por meio de dois protocolos. O primeiro protocolo utilizado é o Zigbee, um protocolo de rede sem fio configurado pela empresa de trackers na instalação. Esse protocolo transmitia informações da chamada TCU (Tracker Control Unit) para a NCU (Network Control Unit). O dispositivo NCU geralmente se conecta à central de controle por meio de Modbus TCP/IP, para conexão utiliza-se o ip da NCU e a porta 502, padrão do TCP, permitindo o acesso aos dados dos trackers (TECHNOLOGIES, 2017).

Os dados podem ser acessados por meio de endereços, geralmente dados de maior interesse são: o ângulo atual do tracker (em radianos) e o ângulo ideal em que ele deveria estar (também em radianos). Esses dados segundo manual do fabricante do tracker pesquisado se encontram em endereços numerados de 30180 até 31854. Cada um desses endereços de referência correspondem a um dado de um dos trackers (escravos) na comunicação. Outros dados de interesse para comunicação costumam ser um total de 10 alarmes, sendo eles posição segura, modo sleep, nível crítico e baixo de bateria, botão de parrada, baixa velocidade do eixo de rotação, bloqueio do eixo de rotação e sobrecorrente, sua leitura é realizada e quando o bit captado registra 0, os alarmes estão desligados, quando se capta um bit 1 os alarmes encontram-se ativos (TECHNOLOGIES, 2017).

3 DEFINIÇÃO DO PROBLEMA

3.1 Captação de dados

A atividade desenvolvida, na qual se baseia a construção deste documento de concepção básica, foi gerada a partir de um processo de pesquisa em laboratório pelos discentes. No laboratório, o primeiro objetivo era captar dados de geração de energia fotovoltaica. A pesquisa se iniciava como um desafio: a implementação de formas de captação de dados de equipamentos encontrados em usinas de energia fotovoltaica, gerando assim um ambiente que permitiria a utilização destes tipos de dados para análise, controle e possíveis atividades de automação.

O desafio de captação de dados pode ser observado como uma demanda simples, pois, como mencionado na seção Sistema SCADA, já existem soluções prontas no mercado que realizam esse tipo de demanda. Deste modo, a intenção da presente pesquisa, além da coleta de dados, estaria ligada à utilização de um conhecimento referente à extração de dados desses equipamentos. Isso envolve o entendimento das possibilidades dos padrões de comunicação aplicáveis a esta situação, bem como dos hardwares que podem ser utilizados, incluindo controladores, cabeamento, conectores, entre outros.

Todo esse processo de conhecimento e testes de ferramentas, equipamentos e aplicações teriam como objetivo a capacitação da equipe designada para a realização deste trabalho, assim como também o desenvolvimento de um senso crítico relacionado a realizar a melhor escolha possível da combinação de equipamentos e práticas para alcançar o objetivo estabelecido para a pesquisa.

3.2 Criação de uma interface homem máquina

Durante o processo de desenvolvimento exploratório da captação de dados de equipamentos utilizados em usinas fotovoltaicas, surgiu a necessidade de leitura dos dados captados. Isso permitiria que os envolvidos no processo de desenvolvimento pudessem interpretar de forma crítica e entender o funcionamento dos sensores atrelados aos equipamentos utilizados. Assim, surgiu a necessidade do desenvolvimento de uma forma de conversão e visualização dos dados para valores que se mostrassem adequados à compreensão humana.

Concomitantemente a essa necessidade de compreensão dos dados extraídos, surgiu uma visão atrelada à parte comercial. Essa ideia estava relacionada à visualização dos dados através

de uma interface gráfica e de forma simplificada. O objetivo era permitir o acesso aos dados coletados para as pessoas interessadas, gerando também a possibilidade de análise de desempenho dos equipamentos monitorados, detecção de erros, melhorias em projetos de usinas fotovoltaicas e aplicação de modelos de inteligência artificial para detecção de parâmetros visuais.

3.3 Alimentação de um banco de dados com informações captadas

Para criar uma forma adequada de expor os dados, desenvolvendo análises e proporcionando insights e soluções aos proprietários dos dados, surgiu a necessidade de registrar esses dados em um local de armazenamento. Isso possibilitaria a consulta dos dados, a geração de gráficos históricos, análises estatísticas e a análise de sazonalidade, entre outras funcionalidades.

De maneira geral, surgiu então uma nova necessidade: a busca por um banco de dados com capacidade suficiente para registrar grandes quantidades de informação. Isso gera um local de consulta confiável e adequado para a realização das análises mencionadas.

3.4 Desenvolvimento de uma interface acessível em diferentes localidades

Unindo-se à necessidade de um meio de consulta dos dados extraídos, posteriormente surgiu a ideia de acessar esses dados de qualquer localidade. Buscava-se então acompanhar os dados extraídos, a frequência, a qualidade e a precisão, gerando assim a demanda de migrar os recursos anteriores para o espaço web. Isso permitiria que qualquer pessoa com as devidas credenciais de acesso pudesse ter acesso aos dados de qualquer lugar, sendo o único limitante o acesso à internet.

3.5 União das funcionalidades e aplicação em usinas

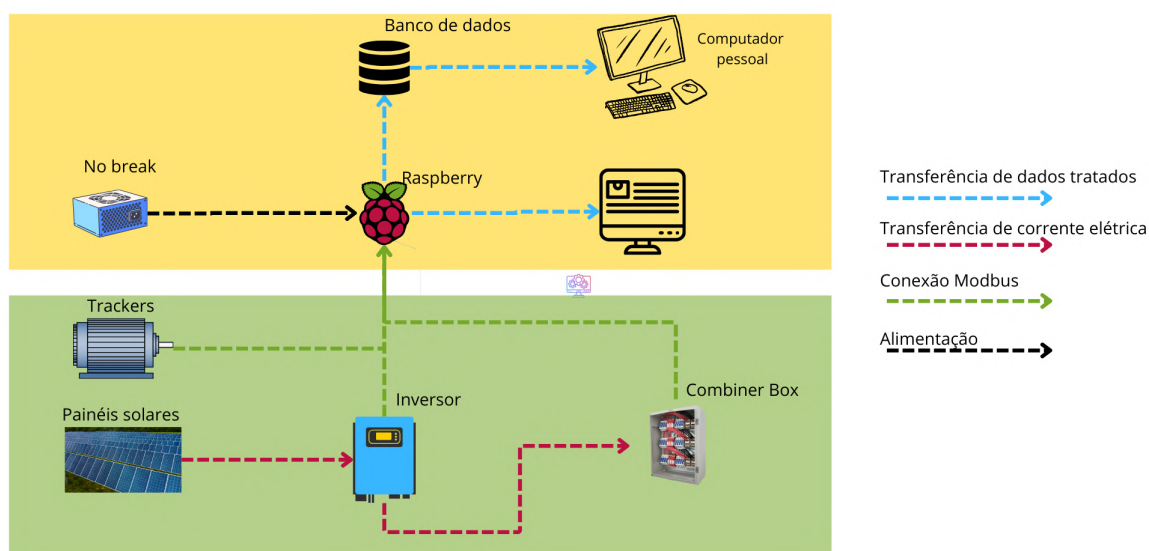
Após uma série de demandas pontuadas, houve a necessidade de unir todas as soluções para aplicá-las nos locais de destino, buscando-se então um produto flexível em termos de adaptação para diferentes tipos de usinas, levando em consideração variáveis como tamanho, temperatura, recursos disponíveis, localização, entre outros.

4 SOLUÇÕES PROPOSTAS

Levando em consideração o problema descrito no capítulo 3, foram desenvolvidas três alternativas para realizar o desenvolvimento do produto. De forma geral, a solução busca atender a uma usina fotovoltaica completa em termos de equipamentos.

4.1 Alternativa 1: Raspberry e Python para coleta dos dados

Figura 4.1 – Fluxo da alternativa 1



Fonte: Autores (2023)

A Figura 4.1 ilustra o fluxo da Alternativa 1, delineando o sistema padrão de usinas em verde. A proposta, destacada em amarelo, apresenta uma solução para a extração e disponibilização remota e local dos dados provenientes dos inversores, combiners e trackers em uma usina fotovoltaica.

O processo tem início nos equipamentos como inversores, *combiner box* e *trackers*, tais dispositivos são conectados ao dispositivo de controle utilizando padrões de comunicação RS485 ou Modbus TCP/IP. Entre os dispositivos em campo e o controlador que atua na captura de dados, pode haver um Dispositivo de Proteção contra Surtos (DPS) para prevenir danos causados por picos elétricos. O sinal é encaminhado ao módulo eletrônico RS485, responsável por converter o sinal serial 485 para USB. Este sinal, então, é transmitido para um dispositivo Raspberry Pi do modelo Pi 4 B, opção disponível no laboratório que atende as demandas, como

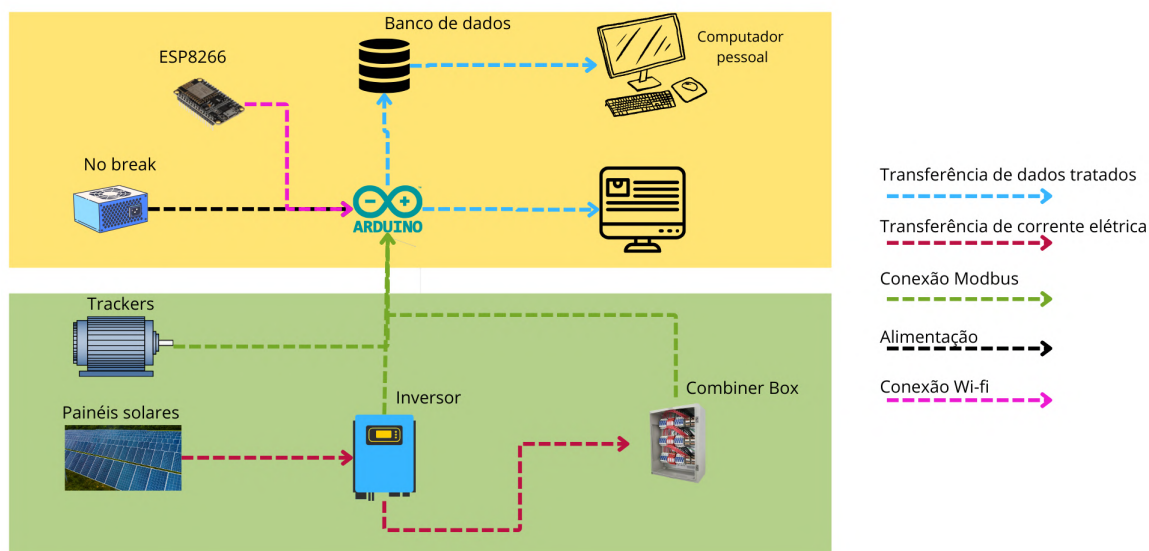
processamento Broadcom BCM2711, memória RAM de até 8G, conexão com wi-fi e quatro portas USB. O microcontrolador é responsável pela captura dos dados enviados pelos inversores e *combiners box*. No caso dos trackers, o sinal TCP/IP passa por uma Unidade de Controle de Rede (NCU) antes de ser direcionado para o Raspberry Pi.

Posteriormente, os dados são submetidos a um processo de tratamento antes de serem encaminhados para o banco de dados, onde são armazenados para posterior consumo e disponibilização em uma interface remota, para obter o acompanhamento da usina.

Esse sistema integrado visa não apenas garantir a segurança elétrica por meio do DPS, mas também a coleta e a visualização de dados, tanto localmente quanto remotamente. A sincronização eficiente entre as etapas do processo fortalece a capacidade de monitoramento da usina fotovoltaica, proporcionando uma gestão mais eficaz e aprimorando a tomada de decisões futuras

4.2 Alternativa 2: Arduino para coleta dos dados

Figura 4.2 – Fluxo da alternativa 2



Fonte: Autores (2023)

A Figura 4.2 apresenta o fluxo da Alternativa 2. Similar à primeira alternativa, o processo de transferência de dados tem início na conexão entre os dispositivos e o controlador. Esta conexão, exige que os dados passem pelos conversores RS485, nos quais o sinal em RS485 é

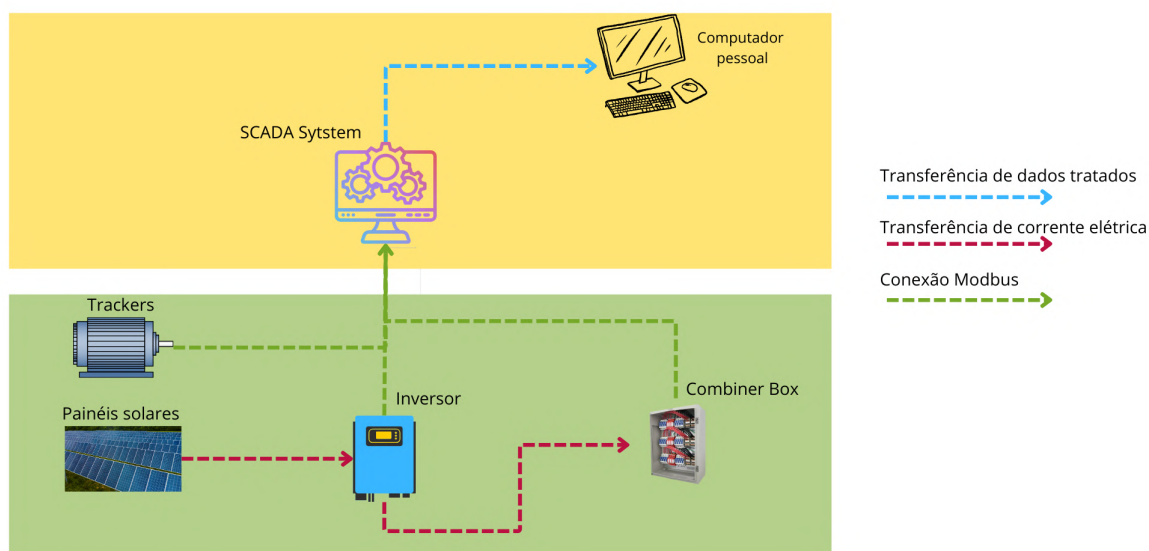
convertido para USB. Notavelmente, no Arduino Mega (opção disponível para o desenvolvimento no laboratório), os pinos RX/TX são empregados para receber esses dados, embora seja possível configurar outros pinos do Arduino para funcionarem como receptores de sinal serial.

Posteriormente, os dados são encaminhados para o Arduino, onde são interpretados, tratados e, em seguida, enviados para o banco de dados. Para viabilizar essa conexão, é essencial contar com acesso à internet, e para esse propósito, é empregado um módulo ESP8266 Mega (opção disponível para o desenvolvimento no laboratório). A presença desse componente assegura a conectividade necessária para a transmissão dos dados.

Com os dados armazenados no banco de dados, torna-se possível o acesso por meio de uma interface remota. Essa abordagem não apenas centraliza a gestão e interpretação dos dados localmente, mas também oferece a flexibilidade de acesso remoto para análise e monitoramento.

4.3 Alternativa 3: Utilização de um sistema SCADA comercial

Figura 4.3 – Fluxo da alternativa 3



Fonte: Autores (2023)

A Figura 4.3 apresenta o fluxograma referente à terceira solução que foi proposta. Entre as 3 propostas, esta seria a de mais simples aplicação, visto que a mesma se baseia na utilização de um sistema SCADA já pronto do mercado para aplicação nas plantas das usinas. Sendo assim, a solução em si já nos ofereceria os requisitos para conexão. Sua aplicação deveria ser

utilizada em computador industrial unido a um *HUB* para conexões, e a aplicação utilizado, assim como a base de dados e a fonte de acesso remoto aos indicadores, estariam sendo disponibilizados pela empresa do sistema contratado.

O maior desafio desta etapa se encontraria no desenvolvimento do padrão visual de monitoramento das usinas, bem como na otimização para menor utilização de *tags* de monitoramento possível para cada uma das usinas em que o serviço fosse aplicado. Isso porque, como o serviço tem como premissa a redução de custos e o pagamento por sistemas SCADA no mercado é feito com base no número de *tags* de monitoramento utilizadas, a otimização de um menor número delas, trazendo o máximo de informação, seria parte importante de sua aplicação.

5 ANÁLISE DAS SOLUÇÕES

Para analisar as soluções apresentadas, foram utilizados alguns critérios, sendo eles o custo que envolve a aplicação da solução. Visto que a aplicação das soluções poderia ser voltada para um ambiente comercial. Outro parâmetro analisado seria a capacidade de cada uma das soluções. Um dos objetivos da pesquisa era buscar uma solução que atendesse a todo o mercado, sendo capaz de atender desde pequenas usinas que contassem com baixíssimos números de equipamentos monitorados até mesmo grandes usinas. Deste modo, segue-se uma análise aprofundada de cada um dos pontos.

5.1 Custos

A alternativa com menores custos para a aplicação é a utilização do Arduino juntamente com o módulo ESP8266, pois esses equipamentos costumam ser utilizados em ambientes de pesquisa e contam com baixo custo de aplicação. Atualmente, o Arduino Mega tem um custo de R\$ 150,00, enquanto o módulo ESP8266 é encontrado por cerca de R\$ 40,00. A Parte de cabeamento e conversores envolveriam um custo adicional de aproximadamente R\$ 600,00. Por fim, como parte de proteção para o sistema, incluindo o nobreak e os DPS, seria gerado um custo de R\$ 600,00, já os custos com banco de dados, para nível laboratorial, foram desconsiderados, dado que esse seria local. Assim, totaliza um valor próximo a R\$ 1390,00 para a aplicação da solução.

Para a solução um temos o Raspberry Pi. Este equipamento oferece um custo no mercado atualmente próximo aos R\$ 1600,00, para os conversores e cabeamento, assim como para a solução com arduino seriam gastos em torno de R\$ 600,00 e como parte de proteção para o sistema, incluindo o nobreak e os DPS, seria gerado um custo de R\$ 600,00, já os custos com banco de dados, para nível laboratorial, foram desconsiderados, dado que esse seria local. Assim, totaliza totalizando um custo de R\$ 2800,00 para a solução.

Já a terceira solução, que engloba o sistema SCADA uma solução pronta no mercado, poderia variar de valor de usina para usina, pois ela leva em consideração o número de equipamentos de uma pequena usina. Realizadas breves pesquisas no mercado em 2021 a aplicação da solução por um sistema SCADA simples contaria com mensalidades em torno de R\$ 15.000,00, podendo chegar a valores muito maiores de acordo com a variação de tamanho da usina e da empresa que forneceria a aplicação do sistema utilizado.

5.2 Capacidade

Em termos de capacidade para serem aplicados em usinas de diferentes tamanhos, o sistema SCADA se destaca nesse quesito. Os sistemas SCADA estão no mercado há anos, possuindo conhecimento e uma gama de testes de aplicação em diversos tipos de indústrias, desde indústrias de pequeno porte até indústrias líderes em seus segmentos. Dessa forma, o sistema SCADA ofereceria soluções testadas que se adequariam a qualquer aplicação escolhida.

A solução utilizando Raspberry Pi em testes mostrou-se muito capaz e flexível. Por se tratar de um microcomputador com elevada capacidade de processamento, pode-se afirmar que o Raspberry Pi possui capacidade suficiente para suprir diversas usinas analisadas e realizar a coleta de dados, além do envio para a nuvem.

O Arduino, por ser um equipamento voltado para testes laboratoriais, possui suas limitações. Em ambiente comercial, isso gerava uma certa preocupação em relação ao seu desempenho. O Arduino possui formas limitadas de comunicação que poderiam ser supridas por conversores ou ampliação do número de portas. No entanto, a utilização de mais equipamentos que não têm sua construção feita em conjunto com o Arduino poderia gerar erros com maior facilidade. Além disso, surge um problema significativo quando se pensa no Arduino se conectando à internet, já que o mesmo precisaria de um equipamento constantemente conectado a si para ter acesso à rede e realizar o trabalho. Dessa forma, o Arduino fica em último lugar nesse critério.

6 IMPLEMENTAÇÃO DO SISTEMA

Analisando as alternativas propostas, bem como as suas vantagens e desvantagens, optou-se pela alternativa 1. Pois o custo apresentado é passível para aplicação a nível laboratorial e ele possui requisitos técnicos que são suficientes para extrair, tratar e realizar o envio dos dados.

Solução	Ranking	
	Custos	Capacidade
Raspberry pi 4 B + Python	2	2
Arduino + Esp 8266	1	3
Sistema Scada comercial	3	1

6.1 Materiais e metodologia

No presente capítulo serão apresentados materiais e a metodologia utilizada para o desenvolvimento do projeto a nível laboratorial.

6.1.1 Sistema fotovoltaico

Com a determinação da área da presente pesquisa, viu-se a necessidade de um sistema fotovoltaico para a geração de energia, o mesmo foi construído pela equipe de pesquisa em laboratório na universidade. A primeira decisão considerou o tipo de sistema a ser utilizado, podendo-se fazer o uso do sistema diretamente ligado a rede, híbrido ou isolado (utilizando-se de baterias para o acúmulo da energia gerada), tendo em vista o local de realização do projeto, a UFLA já possui diversas instalações fotovoltaicas, o que indica um local preparado para a implementação de um sistema *On Grid* ou diretamente ligado a rede, deste modo esta foi a implementação escolhida.

Todo o processo foi desenvolvido no laboratório de combustíveis e combustão, localizado no prédio da ABI (Área Básica de Ingresso) da UFLA. Enquanto os equipamentos para o desenvolvimento dos testes e do sistema foram providenciados pela Universidade, coordenado pelos professores Carlos Eduardo Castilla Alvarez e Márcio Montagnana Vicente Leme.

6.1.1.1 Placas fotovoltaicas

Para a composição da bancada de trabalho em laboratório utilizou-se 3 placas fotovoltaicas da marca Trina Solar, modelo Honey. Tais placas foram conectadas em série e como suporte utilizava-se uma estrutura própria desenvolvida pelo grupo de pesquisa exclusivamente para realização de testes. Tal estrutura apresenta angulação fixa de 24 graus em relação ao solo, possui também rodas que permitem movimenta-la e posiciona-la em local adequado.

Figura 6.1 – Placas Fotovoltaicas

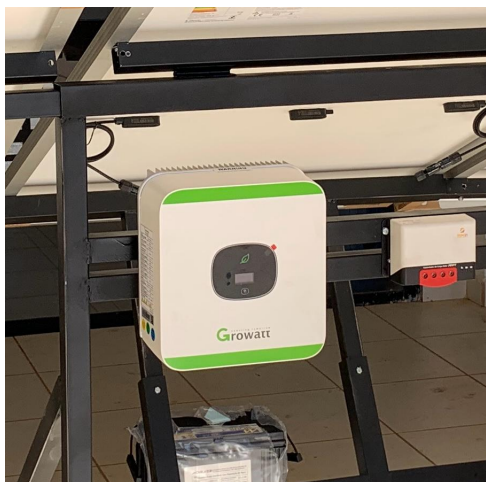


Fonte: Autores (2021)

6.1.1.2 Inversor

Juntamente com as placas fotovoltaicas deve-se utilizar um inversor que seja capaz de converter a energia de corrente contínua (CC) gerada pelo conjunto de placas em energia de corrente alternada (AC), para isso a equipe utilizou de um inversor da marca Growatt, modelo MIC 1500TL-X, o equipamento possui capacidade de suportar tensão de entrada de 500 Volts, corrente de entrada máxima de 13 amperes, operando em temperaturas entre -25°C e 60°C . O inversor ainda teria capacidade de operar de forma híbrida, porém tal artifício não foi explorado em pesquisa.

Figura 6.2 – Inversor



Fonte: Autores (2021)

6.1.2 Conexão e instalação do sistema

Para realizar a conexão física do sistema, utilizou cabos específicos para a conexão RS485 utilizados no inversor, em conjunto com o conversor de RS485 para USB, como mostrado na figura 2.6. Estes são cabos de cobre que possuem blindagem eletrostática, especialmente projetados para comunicação RS485.

Figura 6.3 – Cabos com blindagem e conectores



Fonte: Autores (2021)

A conexão é estabelecida entre o Inversor e o controlador, mas para a ampliação e aplicação em usinas pode ser ligado a linhas de comunicação. Cada linha é conectada a um DPS, que, por sua vez, utilizando os cabos para RS485, é conectado aos módulos conversores. Em seguida, esses mesmos cabos são utilizados para conectar os módulos ao Raspberry Pi.

6.1.3 Extração dos dados

Optou-se por utilizar a linguagem de programação Python para realizar as extrações. Essa escolha foi motivada por ser uma linguagem de maior domínio pela equipe, podendo ser utilizada outras linguagens desde que consiga realizar os requisitos da solução proposta. Além disso, Python conta com uma comunidade ativa de desenvolvedores, proporcionando uma ampla gama de recursos e suporte.

Outro ponto relevante é a presença de bibliotecas Modbus, que oferecem uma interface fácil para interagir com dispositivos Modbus, simplificando significativamente a implementação.

6.1.3.1 Captação dos dados dados inversores

Para realizar a extração, desenvolveu-se uma função capaz de obter dados de acordo com as especificações do protocolo Modbus RTU. Como parâmetros para a função, são fornecidos o ID dos escravos, a porta USB, e os números do registro inicial e final. Inicialmente, essa função estabelece uma conexão Modbus cliente. Nesse processo, passaram-se como parâmetros o tipo de protocolo físico, que, neste caso, é o RTU, a porta USB da linha, o timeout (período máximo de espera). Após testes, foi determinado um valor ideal para o timeout de 0.25 segundos. Por fim, foi definido o baudrate de 9600, a velocidade de transmissão de dados, conforme especificado pelo fornecedor nas seções 2.11 e 2.12 deste trabalho.

Uma vez estabelecida a conexão Modbus entre o Raspberry e os inversores, a extração dos dados é realizada por meio da função *read_input_registers* da biblioteca PyModbus. Nessa função, são passados os valores de início e fim do registro a serem lidos, os quais foram especificados nas seções 2.11 e 2.12 deste trabalho.

Abaixo segue um passo a passo para realizar a extração do dados dos inversores .

1. Inclui a biblioteca que implementa o protocolo Modbus.

2. Utiliza a função da biblioteca para incluir os parâmetros do protocolo, como o método a ser utilizado (RTU), a porta de conexão USB, timeout e o baudrate.
3. Inicia a conexão com o protocolo.
4. Utiliza a função de comando especificada pelo equipamento, como *input register*, para incluir os endereços dos dados a serem coletados e o ID do escravo.
5. Incluir um loop variando de acordo com o ID do escravo e dos registros dos dados.

6.1.3.2 Captação dos dados da estação solarimétrica

A captação dos dados da estação solarimétrica é bem semelhante a extração dos inversores, diferenciando-se pelo tipo de leitura, esta por sua vez é do tipo *holding* que é possível ler um ou mais registros por vez.

6.1.3.3 Captação dos dados dos Trackers

A captação de dados aplicadas a tracker foi uma das funções desenvolvidas para o código buscando atender os padrões de mercado, como em laboratório não existia este equipamento para trabalho e testes, buscou-se desenvolver um código de comunicação básica voltada ao Modbus TCP/IP, no qual se capitariam dados dos escravos de forma sequencial salvando tais dados em vetores.

Tal função contava com a conexão TCP/IP por meio de um ip de rede e foi programada para realizar extrações em *Holding Registers*, a função ainda conta com a capacidade de traduzir dados obtidos em binários para decimais, facilitando a leitura e o tratamento.

1. Inclui a biblioteca que implementa o protocolo Modbus.
2. Utiliza a função da biblioteca para incluir os parâmetros do protocolo, como o endereço de IP do dispositivo.
3. Inicia a conexão com o protocolo.
4. Utiliza a função de comando especificada pelo equipamento, como *holding registers*, para incluir os endereços dos dados a serem coletados e o ID do escravo.
5. Incluir um loop variando de acordo com o ID do escravo e dos registros dos dados.

6.1.4 Tratamento dos dados

De maneira geral, os dados são extraídos em vetores, o que facilitou o seu tratamento. Conforme especificado nas seções 2.11, 2.12 e 2.13, a capacidade máxima de armazenamento de cada registro é de 16 bits. Dessa forma, alguns registros, como potência total, potência reativa, tempo total, energia total, dentre outros são representados em dois registros, devido ao fato de serem números muito grandes. Portanto, foi necessário converter esses números em bits, unindo-os para formar um registro de 32 bits, que posteriormente era convertido em inteiro.

Quanto aos registros de 16 bits, eles já eram fornecidos como inteiros, sendo necessário apenas conhecer o nome do dado associado.

6.1.5 Envio dos dados para o banco de dados

Com os dados coletados a próxima etapa é encaminhá-los para o banco de dados. Para isso é realizada a conexão com o banco de dados exemplificada a baixo. A conexão é feita de maneira similar para todos os envios visto que o banco de dados foi construído de forma a receber os dados de cada um dos equipamentos para diferentes tabelas.

1. Inclui a biblioteca para conexão com o banco de dados.
2. Utiliza a função para conectar com o banco de dados.
3. Na função para realizar a conexão é necessário passar como parâmetro o host, usuário, senha e a base de dados.
4. Utiliza o comando com SQL para inserir os dados no banco de dados.

Em seguida é realizada a inserção desses dados no banco de dados.

Após o processo de envio dos dados coletados ao banco de dados, a equipe ainda se encarregou em desenvolver um site localmente, de modo a expor as capacidades que o sistema possui, possibilitando o acesso aos dados coletados de forma remota.

Nesta etapa, os discentes contribuíram para garantir o pleno funcionamento da integração da base de dados com o site desenvolvido, além do desenvolvimento do site como uma interface que possibilitasse a leitura dos dados por qualquer indivíduo. Como resultado desse

processo, foi gerada uma página de relatório de dados da usina teste que se atualiza em intervalos próximos a 5 minutos, podendo ter variações impostas pelo desenvolvimento do código de coleta de dados. Na bancada de testes utilizada pelos discentes por conta do baixo número de equipamentos e de dados coletados, o registro de dados no banco de dados e consequentemente coleta de dados por parte do site poderia ser feita em um intervalo muito baixo de tempo, chegando a quase tempo real.

7 RESULTADOS

Tendo em vista os códigos gerados e a possibilidade de extração de dados de acordo com os equipamentos presentes em laboratório, pôde-se realizar testes para adequação do código e gerar os resultados esperados. Como discutido anteriormente, o conjunto de hardware e software escolhidos foi o Raspberry Pi 4, em conjunto com a linguagem de programação Python. Visando atender às necessidades em laboratório, o protocolo de comunicação explorado foi o RS-485, necessitando ainda da utilização de conversores de RS-485 para USB. Sabendo destas necessidades voltadas à implementação, aplicou-se o código voltado para a captação de dados no ambiente de laboratório, utilizando-se do inversor disponível para a bancada de trabalho. O processo é composto pelo seguinte fluxo.

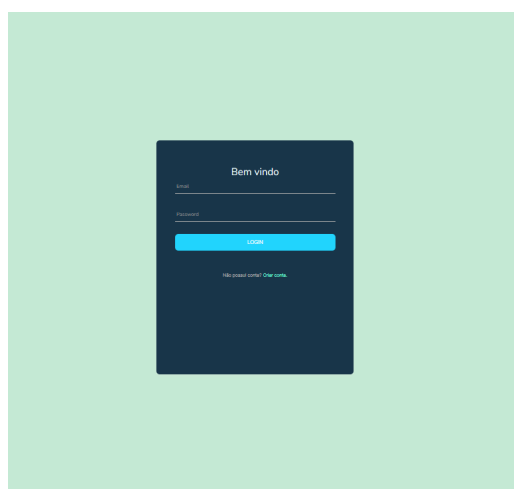
Figura 7.1 – Fluxo de dados



Fonte: Autores (2023)

Os dados foram captados, tratados e expostos em um site de exemplo, hospedado localmente, de modo que se tornasse possível realizar os testes das funções desenvolvidas para a implementação. A representação do site em seu funcionamento pode ser observada a seguir.

Figura 7.2 – Login



Fonte: Autores (2023)

Após o usuário realizar o seu login na página, ele consegue acesso aos dados de potência, tensão, corrente e energia gerada, tudo isso em forma gráfica e em tabela para uma melhor visualização dos dados.

Figura 7.3 – Tabela com os dados

Data	Potência	Frequência	Tensão	Corrente	Potência CC	Tensão CC
2023-06-29T13:15:08.0...	674.4	60	227.4	3.2	688.2	103.2
2023-06-29T13:15:09.0...	684	59.97	227.6	3.1	694.5	99.6
2023-06-29T13:15:10.0...	687	59.97	227.4	3.1	700.9	99.5
2023-06-29T13:15:11.0...	681.7	59.97	227.4	3.1	697.3	96.9
2023-06-29T13:15:13.0...	675.6	59.97	227.4	3.1	689.4	96.7
2023-06-29T13:15:14.0...	681.9	59.97	227.3	3.2	696.4	99.5
2023-06-29T13:15:15.0...	686.7	59.96	227.2	3.1	700.7	99.6
2023-06-29T13:15:16.0...	687.3	59.96	227.2	3.1	701	99.6
2023-06-29T13:15:17.0...	683.8	59.96	227.3	3.1	696.9	103.1

Fonte: Autores (2023)

Figura 7.4 – Continuação da tabela com os dados

Corrente	Potência Diária	Potência Total	Energia Diária	Energia Diária CC
7.3	0.5	30.2	0.4	30.7
7.3	0.5	30.2	0.4	30.7
7.3	0.5	30.2	0.4	30.7
7.5	0.5	30.2	0.4	30.7
7.3	0.5	30.3	0.5	30.7
7.3	0.5	30.3	0.5	30.7
7.3	0.5	30.3	0.5	30.7
6.9	0.5	30.3	0.5	30.7
7.3	0.5	30.3	0.5	30.7
7.3	0.5	30.3	0.5	30.7
7.3	0.5	30.3	0.5	30.7

Fonte: Autores (2023)

A imagem acima mostra uma tabela com os dados coletados a cada intervalo de tempo. Nela, é possível obter de modo abrangente todos os dados extraídos. Dessa forma, é possível realizar análises que incluem todas as variáveis coletadas

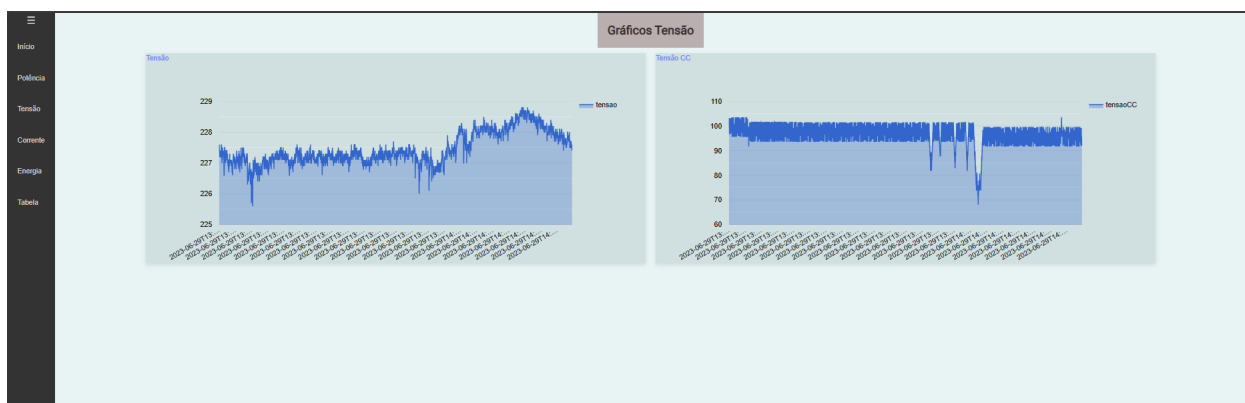
Com os gráficos da potência, tensão, corrente e energia, é possível monitorar o desempenho do sistema, verificar variações inesperadas, analisar os dados para realizar manutenções preventivas, verificar a eficiência do inversor, verificar problemas com os módulos e a sua eficiência entre outras análises. Assim, esses dados trazem confiabilidade operacional e a sustentabilidade do sistema.

Figura 7.5 – Gráficos da Potência



Fonte: Autores (2023)

Figura 7.6 – Gráficos da Tensão



Fonte: Autores (2023)

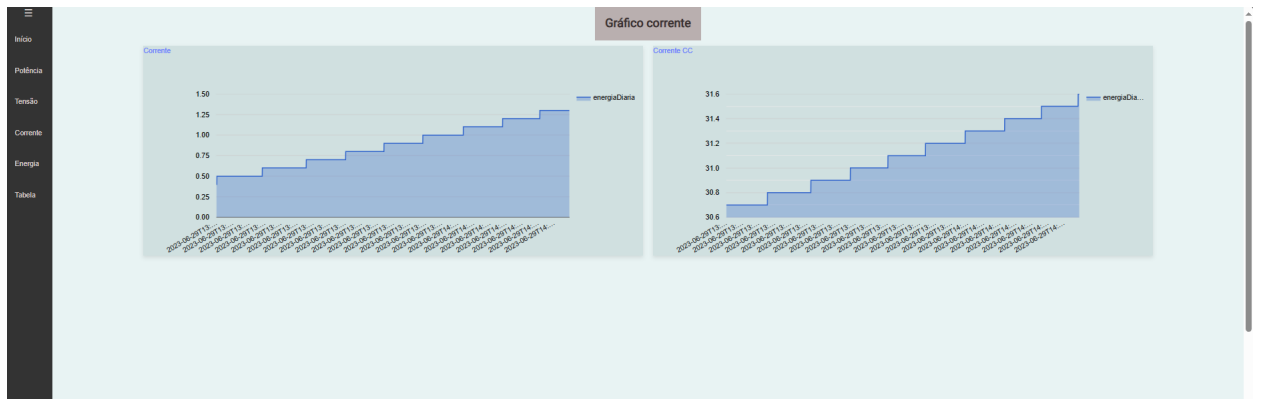
Outro ponto relevante é para avaliar a produção total de energia ao longo do tempo. Isso não apenas permite a verificação do cumprimento de metas de geração, mas também facilita a identificação de padrões sazonais e tendências que podem influenciar as estratégias de operação e manutenção.

Figura 7.7 – Gráficos da Corrente



Fonte: Autores (2023)

Figura 7.8 – Gráficos da Energia



Fonte: Autores (2023)

8 CONCLUSÃO

Com o decorrer deste processo de desenvolvimento, evidenciou-se a viabilidade de conjugar ferramentas simples e acessíveis para conceber soluções proveitosas e funcionais. É notável que, mediante uma utilização apropriada, é possível criar soluções que cumprem com seus objetivos. A solução em que os discentes desenvolveram revela-se útil a nível de pesquisa e pode evoluir para caminhar e acessar ambientes comerciais, desde que as devidas adequações sejam realizadas. Isso abre novas alternativas para um mercado que, até então, conta com soluções direcionadas aos sistemas SCADAs comerciais.

Apesar do sucesso no processo de aplicação da solução diversas dificuldades foram encontradas em cada ponto de aplicação das soluções, sendo elas em primeiro lugar voltadas aos equipamentos para a extração de dados, tendo a pesquisa se iniciado em inversores de motor, posteriormente os discentes enfrentaram dificuldades voltadas a instalação do sistema, bem como os testes voltados ao funcionamento dos padrões de comunicação. No qual os docentes que acompanharam o trabalho encontraram-se extremamente solícitos e dispostos a ajudar e colaborar. Após o sistema instalado e as conexões realizadas, surgiram-se dificuldades voltadas ao desenvolvimento do software, no qual pode ser sanada devido a linguagem escolhida e seu vasto acervo informativo na internet, bem como o acesso ao manual dos equipamentos.

Ao longo da pesquisa para o desenvolvimento desta solução de comunicação industrial voltada para aplicação em usinas fotovoltaicas, tornou-se possível, primeiramente, o desenvolvimento de habilidades analíticas por parte dos discentes envolvidos. Além disso, contribuiu para a formação de atributos básicos necessários a um engenheiro de controle e automação, explorando e integrando conceitos estudados na universidade relacionados a instalações elétricas, redes de comunicação industrial, instrumentação, programação aplicada, entre outros. Essa experiência proporcionou uma abordagem prática e interdisciplinar, enriquecendo o aprendizado dos envolvidos e contribuindo para as pesquisas no campo da engenharia industrial.

REFERÊNCIAS BIBLIOGRÁFICAS

ABOUT Arduino. 2023. Acessado em 8 de Novembro de 2023. Disponível em: <<https://www.arduino.cc/en/about>>.

ABOUT us. 2023. Acessado em 8 de Novembro de 2023. Disponível em: <<https://www.raspberrypi.org/about/>>.

BONDARIK, R.; PILATTI, L. A.; HORST, D. J. Uma visão geral sobre o potencial de geração de energias renováveis no brasil. **Interciencia**, Asociación Interciencia Venezuela interciencia@gmail.com, v. 43, n. 10, p. 680–688, 2018.

BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. [S.l.]: Novatec Editora, 2014.

CARVALHO, F. M. d. Método alternativo para sintonia de múltiplos controladores ressonantes aplicados em sistemas ininterruptos de energia (nobreak). 2013.

CO., L. G. N. E. T. **Growatt Inverter Modbus RTU Protocol**. *I.Shiyan*, 2018.

DIFFERENCE Between RS232 and RS485 : Advantages and Disadvantages. 2023. Acessado em 8 de Novembro de 2023. Disponível em: <<https://www.elprocus.com/difference-between-rs232-and-rs485-advantages-and-disadvantages/>>.

DZIALOSZYNSKI, B. S. Introdução à programação em ambiente labview. **São Paulo:[sn]**, 2015.

FERREIRA, I. V. Uma arquitetura para a aplicação da internet das coisas industrial. Universidade Estadual Paulista (Unesp), 2018.

FRANÇA, J. L. et al. **Manual Para Normalização de Publicações Técnico-científicas**. 5. ed. Belo Horizonte: UFMG, 2001.

GOLDEMBERG, J.; LUCON, O. Energia e meio ambiente no brasil. **Estudos avançados**, SciELO Brasil, v. 21, p. 7–20, 2007.

GUARNIERI, M. V. et al. Usinas solares fotovoltaicas com seguimento em um eixo no brasil: aspectos da construção, custos e expectativa de desempenho. 2017.

HUITSING, P. et al. Attack taxonomies for the modbus protocols. **International Journal of Critical Infrastructure Protection**, Elsevier, v. 1, p. 37–44, 2008.

INC., S. L. **SINGLE-CHIP USB-TO-UART BRIDGE**. [S.l.], 2017.

JUCÁ, S.; PEREIRA, R. Aplicações práticas de sistemas embarcados linux utilizando raspberry pi. **PoD Editoria**, 2018.

LTDA., N. P. E. **MANUAL DE COMUNICAÇÃO MODBUS V1.7x**. [S.l.], 2018.

MACHADO, C. T.; MIRANDA, F. de S. Energia solar fotovoltaica: Uma breve revisão. In: . [s.n.], 2015. Disponível em: <<https://api.semanticscholar.org/CorpusID:169391183>>.

MARAIS, H. Rs-485/rs-422 circuit implementation guide. **AN-960 Application Note**. ANALOG DEVICES, 2008.

MEHTA, M. Esp8266: A breakthrough in wireless sensor networks and internet of things. **International Journal of Electronics and Communication Engineering & Technology**, v. 6, n. 8, p. 7–11, 2015.

MELO, G. H. S. **Implementação de um sistema de controle supervisorio e aquisição de dados para usinas de geração distribuída voltada à energia solar fotovoltaica**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2023.

MODEL, D. A. **Análise do desempenho de uma usina fotovoltaica: valores de projeto versus valores reais**. Dissertação (B.S. thesis), 2021.

MÜLLER, M. L. et al. Sistema supervisorio e aquisição de dados de uma usina de geração fotovoltaica. Florianópolis, SC., 2017.

NASCIMENTO, J. M. A. do; LUCENA, P. B. de. Protocolo modbus. **Redes para Automação Industrial, Universidade do Rio Grande do Norte**, 2003.

O que é DPS ou Dispositivo de Proteção contra surtos? 2023. Acessado em 8 de Novembro de 2023. Disponível em: <<https://www.se.com/br/pt/work/local-content/dps-dispositivo-protecao-contrasurtos/>>.

OLAYA, A. F. R.; LÓPEZ, A. B.; MORENO, F. G. G. Implementación de una red modbus/tcp. **Ingeniería y Competitividad**, v. 6, n. 2, p. 35–44, 2004.

PONTES, W. M. C. S. Implantação de estação solarimétrica e de sistema supervisorio com scadabr e plataforma iot em usina fotovoltaica na unilab-ce. 2021.

RASPBERRY Pi Zero. 2023. Acessado em 8 de Novembro de 2023. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-zero/>>.

RELLA, R. Energia fotovoltaica no brasil. **Revista de Iniciação Científica**, v. 15, n. 1, p. 28–38, 2017.

SANTOS, F. T. d.; MARINATO, M. M. Sistema supervisorio. SANTOS, Fabrício Teixeira dos Santos MARINATO, Matheus Mariano, 2021.

SOUZA, V. A. O protocolo modbus. **Available in:< http://www. cerne-tec. com. br/Modbus. pdf> Acesso**, v. 11, 2008.

TECHNOLOGIES, I. A. **Technical tracker Specification**. [S.l.], 2017.

THE RS232 STANDARD. 2023. Acessado em 8 de Novembro de 2023. Disponível em: <https://www.camiresearch.com/Data_Com_Basics/RS232_standard.html#anchor1154232>.

THOMAS, G. Introduction to the modbus protocol. **The Extension**, v. 9, n. 4, p. 1–4, 2008.

WIRING. 2023. Acessado em 8 de Novembro de 2023. Disponível em: <<https://wiring.org.co/>>.

WOJAHN, N. N. Testador de cabo digital com informação de temperatura e umidade utilizado em sistemas de termometria com comunicação tia/eia-485. Instituto Federal de Educação, Ciência e Tecnologia Farroupilha, 2023.

XUAN, L.; YONGZHONG, L. Research and implementation of modbus tcp security enhancement protocol. In: IOP PUBLISHING. **Journal of Physics: Conference Series**. [S.l.], 2019. v. 1213, n. 5, p. 052058.

9 APÊNDICE

1 - Função para extrair dados do inversor e da *combiner box*

```

from pymodbus.client.sync import ModbusSerialClient as ModbusClient
def run_sync_dados_linhas(id, porta, register_ini, register_fim):
    # Instancia um objeto modbus com os par metros do
    # inversor ou da combiner
    client = ModbusClient
        (method='rtu',
        port=porta,
        timeout=0.25,
        baudrate=9600)
    client.strict = False
    # Inicia a conexao
    client.connect()
try:
    # Utiliza a funcao de leitura dos dados pra serem extraidos
    dados = client.read_input_registers
        (register_ini,
        register_fim,
        unit=id)
    except Exception as e:
        print('Erro ao extrair os dados')
        raise e
try:
    a = dados.registers
    except Exception as e:
        return []
    # Fecha a conex o
    client.close()
return a

```

2 - Função para extrair dados da estação

```

from pymodbus.client.sync import ModbusSerialClient as ModbusClient
def run_sync_estacao(id, porta, register_ini, qtd):

```

```

# Instancia um objeto modbus com os par metros da esta o
client = ModbusClient
    (method='rtu',
    port=porta,
    timeout=0.25,
    baudrate=9600)
client.strict=False
# Inicia a conexao
client.connect()
try:
    # Utiliza a funcao de leitura dos dados pra serem extraidos
    dados = client.read_holding_registers
        (register_ini,
        qtd,
        unit=id)
except Exception as e:
    print('Erro ao extrair os dados')
    raise e
try:
    a = dados.registers
except Exception as e:
    print('Erro ao extrair os dados')
    return []
# Fecha a conex o
client.close()
return a

```

3 - Função para armazenar os dados do inversor no banco de dados

```

import mysql.connector
from mysql.connector import Error
def mysql(user, password, database, host):
    try:
        # Inicia a conex o com o banco
        banco = mysql.connector.connect

```

```

(host=host,
user=user,
passwd=password,
database=database)
cursor = banco.cursor()
#Insere no banco
comando_SQL = '''INSERT INTO inversor (
    energiaDiaria,
    energiaCC,
    potenciaTotal,
    tensao,
    corrente,
    correnteCC,
    potenciaDiaria,
    potencia,
    frequencia,
    tensao,
    tensaoCC,
    hora)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)'''
dados = ( temperaturaAr,
    temperaturaModulo
    vento,
    umidade,
    refletida,
    hora,
    data)
print(dados)
# Para cada variavel inserido um valor
cursor.execute(comando_SQL, dados)
banco.commit()
cursor.close()
banco.close()
print("Conexao finalizada")

```

```

except Error as erro:
    print("Falha na comunicacao: {}".format(erro))
finally:
    try:
        if(banco.is_connected()):
            cursor.close()
            banco.close()
            print("Conexao finalizada")
            return 1
        else:
            return 0
    except Exception:
        print("nao conectou no DB")

```

4 - Função para armazenar os dados da estação no banco de dados

```

import mysql.connector
from mysql.connector import Error
def mySQL_estacao(user, password, database, host):
    try:
        # Inicia a conexão com o banco
        banco = mysql.connector.connect
            (host=host,
            user=user,
            passwd=password,
            database=database)
        cursor = banco.cursor()
        #Insere no banco
        comando_SQL = '''INSERT INTO estacao (
            temperaturaAr,
            temperaturaModulo
            vento,
            umidade,
            refletida,
            hora)
            VALUES (%s, %s, %s, %s, %s, %s)'''

```

```
dados =( temperaturaAr ,
          temperaturaModulo
          vento ,
          umidade ,
          refletida ,
          hora)

print(dados)

# Para cada variavel      inserido um valor
cursor.execute(comando_SQL, dados)

banco.commit()
cursor.close()
banco.close()

print("Conexao finalizada")

except Error as erro:
    print("Falha na comunicacao: {}".format(erro))

finally:
    try:
        if(banco.is_connected()):
            cursor.close()
            banco.close()
            print("Conexao finalizada")
            return 1
        else:
            return 0

except Exception:
    print("nao conectou no DB")
```