



ANDRÉ CORTEZ

**DESENVOLVIMENTO DE UMA PLATAFORMA SCADA
PARA SISTEMAS FOTOVOLTAICOS UTILIZANDO O
PROTOCOLO DE COMUNICAÇÃO RS-485**

LAVRAS – MG

2023

ANDRÉ CORTEZ

**DESENVOLVIMENTO DE UMA PLATAFORMA SCADA PARA SISTEMAS
FOTOVOLTAICOS UTILIZANDO O PROTOCOLO DE COMUNICAÇÃO RS-
485**

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do curso de Engenharia de Controle e Automação, para a obtenção do título de Bacharel.

Prof. Dr. Carlos Eduardo Castilla Alvarez
Orientador

**LAVRAS – MG
2023**

AGRADECIMENTOS

Gostaria de agradecer a Deus, que me deu forças para enfrentar todos os obstáculos encontrados durante o curso.

Aos meus pais, Sônia Margarete Quassi Cortez e Gilberto Garcia Cortez, sem eles me apoiando e incentivando a me tornar uma pessoa cada vez mais esforçada e focada em seus objetivos não chegaria até aqui. A minha irmã e sobrinha, Priscila Caroline Cortez e Sophia, que sempre estiveram ao meu lado mesmo distante. A minha tia e meus avós, Sandra, Amélia e Waldomir, mesmo estando longe sempre deram conselhos para eu nunca desistir.

À minha namorada, Carolina Salles Freire, que sempre me apoiou e esteve ao meu lado nos melhores e piores momentos do meu curso, incentivando e dando conselhos para seguir em frente, foi essencial nesta caminhada.

Ao meu orientador, Carlos Eduardo Castilla Alvarez, desde o momento que nos conhecemos sempre se mostrou prestativo e companheiro, e foi de grande ajuda para a conclusão do meu curso e início de carreira no mercado de trabalho junto da empresa MVPower, que me deu a oportunidade de estagiar durante um ano. Além de todos os outros professores que tive contato na universidade, com certeza foram essenciais para minha carreira pessoal.

Agradeço a oportunidade que a Universidade Federal de Lavras me forneceu durante esses anos de graduação e ao Departamento de Automática.

Aos meus amigos do curso e de fora, em especial ao Arthur Souza, Iris Oliveira, Gabriel Dani, esses com certeza nunca esquecerei, foram minha segunda família em Lavras.

Ao Núcleo de Estudos do Laboratório de Mobilidade Terrestre, o qual me permitiu desenvolver Soft e Hard Skills, me tornando uma pessoa mais capacitada para o mercado de trabalho.

Por fim, agradeço a banca por ter aceitado o convite de participar e contribuir para este trabalho.

RESUMO

A energia renovável é uma prioridade global e o Brasil apresenta grande potencial no aproveitamento de energia solar. Nos últimos anos o Brasil tem-se destacado na instalação de sistemas fotovoltaicos, porém há um desafio a ser enfrentado, como a aquisição e monitoramento de dados gerados pelas usinas solares. Logo, objetivou-se com este trabalho por meio do protocolo Modbus e a linguagem de programa Python, desenvolver um sistema de monitoramento e aquisição de dados de usinas fotovoltaicas. Os componentes do sistema fotovoltaico foram montados de acordo com as instruções do fornecedor. O protocolo Modbus RS-485 foi utilizado para a aquisição de dados do inversor conectado às placas fotovoltaicas. Em seguida, realizou-se a configuração do sistema operacional para a linguagem de programação Python e utilização do conversor RS485-USB. O supervisório foi desenvolvido no ambiente de desenvolvimento integrado (IDE) Android Studio, por meio da linguagem de programação Flutter. Por meio deste supervisório e da captação de dados, foi possível adquirir e monitorar os dados provenientes do inversor, os quais foram armazenados e apresentados com sucesso em uma interface para dispositivos móveis, tornando-se um sistema SCADA (Supervisory Control And Data Acquisition) com execução em tempo real, permitindo a utilização em sistemas fotovoltaicos com a comunicação RS485 para supervisão remota.

Palavras Chaves: Energia Fotovoltaica, SCADA, Sistema Supervisório, Python, Flutter.

ABSTRACT

Renewable energy is a global priority and Brazil has great potential in harnessing solar energy. In recent years, Brazil has stood out in the installation of photovoltaic systems, but there is a challenge to be faced, such as the acquisition and monitoring of data generated by solar plants. Therefore, the objective of this work, through the Modbus protocol and the Python program language, was to develop a monitoring and data acquisition system for photovoltaic plants. The photovoltaic system components were assembled according to the supplier's instructions. The Modbus RS-485 protocol was used to acquire data from the inverter connected to the photovoltaic panels. Then, the operating system was configured for the Python programming language and the RS485-USB converter was used. The supervisory was developed in the integrated development environment (IDE) Android Studio, through the Flutter programming language. Through this supervisory and data capture, it was possible to acquire and monitor data from the inverter, which were successfully stored and presented in an interface for mobile devices, becoming a SCADA (Supervisory Control And Data Acquisition) system with execution in real time, allowing the use in photovoltaic systems with RS485 communication for remote supervision.

Key-words: Photovoltaic Energy, SCADA, Supervisory System, Python, Flutter.

SUMÁRIO

1. INTRODUÇÃO.....	7
2. OBJETIVOS.....	9
2.1. OBJETIVO GERAL	9
2.2. OBJETIVOS ESPECÍFICOS	9
3. REFERENCIAL TEÓRICO	10
3.1. A RENOVABILIDADE DA MATRIZ ENERGÉTICA BRASILEIRA	10
3.2. ENERGIA FOTOVOLTAICA	11
3.2.1. EFEITO FOTOVOLTAICO	11
3.2.2. TIPOS DE SISTEMAS FOTOVOLTAICOS.....	13
3.3. O AVANÇO DA ENERGIA SOLAR NO BRASIL	15
3.4. SISTEMA DE SUPERVISÃO PARA USINA FOTOVOLTAICA	17
3.4.1. CONTROLE SUPERVISÓRIO E AQUISIÇÃO DE DADOS.....	18
3.4.2. PROTOCOLO MODBUS.....	20
3.5. PYTHON	21
4. MATERIAL E MÉTODOS	22
4.1. COMPONENTES DO SISTEMA FOTOVOLTAICO.....	22
4.2. PROTOCOLO DE COMUNICAÇÃO	23
4.3. CONFIGURAÇÃO DO SISTEMA OPERACIONAL E LINGUAGEM DE PROGRAMAÇÃO.....	25
4.4. CAPTAÇÃO DOS DADOS.....	26
4.5. SUPERVISÓRIO	27
5. RESULTADOS E DISCUSSÃO	28
5.1. BANCADA DE TESTES PARA O SUPERVISÓRIO.....	28
5.2. CONVERSOR RS485-USB E SISTEMA OPERACIONAL.....	29
5.3. COMUNICAÇÃO ENTRE SUPERVISÓRIO E INVERSOR.....	30
5.3.1. RECEBIMENTO DOS DADOS FOTOVOLTAICOS	34
5.4. INTERFACE HOMEM MÁQUINA (IHM).....	35
6. CONCLUSÃO	38
7. TRABALHOS FUTUROS	39
REFERÊNCIAS BIBLIOGRÁFICAS	40
APÊNDICE	43

1. INTRODUÇÃO

A busca por fontes de energia renováveis tem se tornado uma prioridade global, visando mitigar os impactos ambientais causados pela geração de energia a partir de combustíveis fósseis (DE OLIVEIRA *et al.*, 2018). Segundo o Ministério de Minas e Energia (MME), o Brasil possui cerca de 49% de distribuição de energia renovável no país no ano de 2023, sendo a maior parte dela em biomassa de cana e hidrelétrica. Nesse contexto, o Brasil se destaca no aproveitamento da energia renovável, principalmente a energia solar, devido às suas características geográficas favoráveis, em que boa parte do território nacional se encontra perto da linha do equador (MARTINS *et al.* 2017).

A energia fotovoltaica começou a apresentar altos índices de crescimento no país, totalizando apenas em 2022 um valor de 13 Gigawatts (GWs) de potência gerada (MAURICIO, 2022). Os tipos de sistemas que podem ser construídos a partir do uso de placas fotovoltaicas são diversos, como no uso residencial, por meio de sistemas isolados, fornecendo assim energia para eletrodomésticos, iluminação e aquecimento de água (CÂMARA, 2011). Outro sistema que pode ser utilizado em residência ou lugares mais isolados é o sistema híbrido, fornecendo assim a mesma característica do sistema isolado, porém com a vantagem de se conectar a outros sistemas de fornecimento de energia (BRANDÃO, 2013). Por fim, o sistema conectado à rede e comumente mais utilizado na construção de parques solares, apresenta a grande vantagem de diminuir a utilização de combustíveis fósseis, além de gerar créditos para consumidores que tiverem sistemas de geração distribuídas, diminuindo o custo gasto em eletricidade (JARDIM *et al.* 2004).

O Brasil mesmo obtendo vantagem territorial para uso da energia solar, demorou para seguir com o crescimento desta fonte renovável. Contudo há esforços governamentais, como sanções tributárias, centros de pesquisas e empresas privadas para alavancar o uso de painéis fotovoltaicos para geração de energia elétrica (DA SILVA; ARAUJO, 2022). Porém há outro obstáculo a ser enfrentado, a aquisição e monitoramento dos dados gerados por essas usinas solares. Segundo a pesquisa de Melo (2020), o sistema de geração distribuída pode apresentar falhas silenciosas e ficar inoperante até que seja detectada no local de instalação, para isso é comum a utilização de sistemas de captação e supervisão de dados de forma remota.

Para que seja possível monitorar as usinas solares de forma remota é necessário a utilização de protocolos de comunicação industriais, geralmente os mais usados nos equipamentos presentes nessas usinas são os protocolos RS485, RS232 e TCP/IP. O protocolo Modbus, que engloba todos os citados anteriormente, permite que os dados sejam enviados e

recebidos com diferentes formas de valores, sejam eles numéricos, discretos ou analógicos. Essa vantagem permite a integração com diversos equipamentos e linguagens de programação disponíveis no mercado (ZHANG *et al.*, 2022).

Diante disso, o protocolo Modbus apresenta diversas vantagens relacionadas à transmissão de dados, tornando-se uma opção altamente vantajosa para a utilização em inversores, controladores e medidores de energia, todos esses equipamentos essenciais em sistemas fotovoltaicos. Além disso, o Modbus proporciona uma alta velocidade de envio e recebimento dos dados, permite comunicação eficiente em longas distâncias e demonstra uma notável resistência a interferências eletromagnéticas. Essas características fazem dele um aliado indispensável e altamente confiável quando aplicado em conjunto com usinas fotovoltaicas (KUMAR; SETHURAMAN, 2018).

Com o protocolo de comunicação permitindo a integração com diversos sistemas, a linguagem de programação Python se destaca como uma aliada importante nesse cenário. Sua simplicidade, versatilidade e eficiência, aliadas à sua capacidade de funcionar em diferentes plataformas, a tornam uma escolha ideal para os sistemas disponíveis no mercado (SRINATH, 2017). Além disso, a linguagem possui uma comunidade ativa e é amplamente utilizada, conforme evidenciado por um relatório do GitHub (2022), sendo também open-source, permitindo modificações e uso sem restrições.

Em suma, o monitoramento e aquisição de dados em usinas fotovoltaicas desempenham um papel fundamental ao assegurar a integridade dos sistemas e evitar falhas ou perdas de eficiência. Além disso, a disponibilidade de um sistema SCADA acessível e robusto viabiliza sua aplicação em usinas de menor porte, tornando-se uma opção competitiva em relação aos sistemas de monitoramento e aquisição de dados mais dispendiosos (MELO, 2020).

2. OBJETIVOS

2.1. Objetivo Geral

O objetivo deste trabalho foi desenvolver uma aplicação para aquisição e supervisão de dados provenientes de inversores utilizados em conjunto com placas solares.

2.2. Objetivos específicos

Os objetivos específicos deste trabalho foram:

1. Realizar o estudo de protocolos de comunicação utilizado em inversores;
2. Desenvolver o algoritmo para captação de dados dos equipamentos;
3. Desenvolver uma interface gráfica para apresentação dos dados das placas fotovoltaicas;
4. Fazer a integração entre captação de dados e supervisor, a fim de deixar a aplicação em tempo real.

3. REFERENCIAL TEÓRICO

A seguir, apresenta-se uma revisão de literatura sobre a evolução da matriz energética brasileira, com ênfase na energia fotovoltaica. Serão abordados aspectos relevantes sobre o uso de sistemas inteligentes (SCADA, IoT), incluindo linguagens de programação e protocolos de comunicação utilizados nos equipamentos de geração de energia.

3.1. A Renovabilidade da Matriz Energética Brasileira

O aumento significativo no consumo de energia, tanto no setor industrial quanto residencial, levou o Brasil a buscar investimentos em fontes de energia renovável como forma de suprir a demanda e enfrentar a escassez de recursos fósseis (DE OLIVEIRA *et al.*, 2018). Energia renovável é aquela obtida a partir de recursos naturais que são naturalmente reabastecidos, como a luz solar, o vento, a água e a biomassa, ao contrário dos combustíveis fósseis, que são finitos e poluentes. A matriz energética brasileira destaca-se internacionalmente pela expressiva participação nas fontes renováveis. De acordo com o Ministério de Minas e Energia (MME), essas fontes representaram 48% da matriz energética brasileira, enquanto na matriz energética mundial esse percentual foi de apenas 15% (SÍNTESE, 2021).

As principais fontes renováveis presentes na matriz energética brasileira são: biomassa de cana (19,1%), hidráulica (12,6%), lenha e carvão vegetal (8,9%) e outras renováveis (7,7%), conforme mostrado na Figura 1. Estas outras renováveis incluem a energia eólica, a energia solar e a energia geotérmica. Nos últimos anos, a energia eólica e a energia solar têm apresentado um crescimento expressivo no país, graças aos incentivos do Governo Federal e à redução dos impostos contidos nos equipamentos. Em 2020, a capacidade instalada em energia solar fotovoltaica cresceu 66% no país e a energia eólica representou 10,9% da matriz elétrica brasileira (SÍNTESE, 2021).

Figura 1: Distribuição das fontes de energias disponíveis no Brasil



Fonte: SÍNTESE (2021, p. 16)

A elevada participação das fontes renováveis na matriz energética brasileira traz diversos benefícios para o país, como a redução na emissão de gases de efeito estufa, diversificação da matriz energética e geração de empregos. Desse modo, o uso de outros meios energéticos, abandonando aos poucos os combustíveis fósseis contribui com diversos fatores, sejam eles ambientais ou sociais (ALBUQUERQUE; FAGUNDEZ; MOSMANN, 2019).

O Brasil vem investindo em políticas de incentivo, como a redução de impostos em usinas renováveis, incentivando na diminuição da construção de novas usinas não renováveis. Além disso, o governo incentiva a construção de usinas menores em locais mais distantes dos grandes centros, onde pequenas comunidades possam se beneficiar de energia limpa (GONZÁLEZ; SILVA; MASIP-MACÍA, 2021).

A transição que vinha ocorrendo de maneira gradual tornou-se mais acelerada no ano de 2022, evidenciando que o Brasil está trilhando um caminho de transformação em sua matriz energética. Especificamente, destaca-se o avanço significativo na implantação de sistemas de energia solar, o que reduz a dependência em relação às hidroelétricas e outras fontes não renováveis. Esta mudança tem como objetivo mitigar os riscos de escassez de energia diante das alterações climáticas (DE SOUZA; FERREIRA, 2022).

3.2. Energia Fotovoltaica

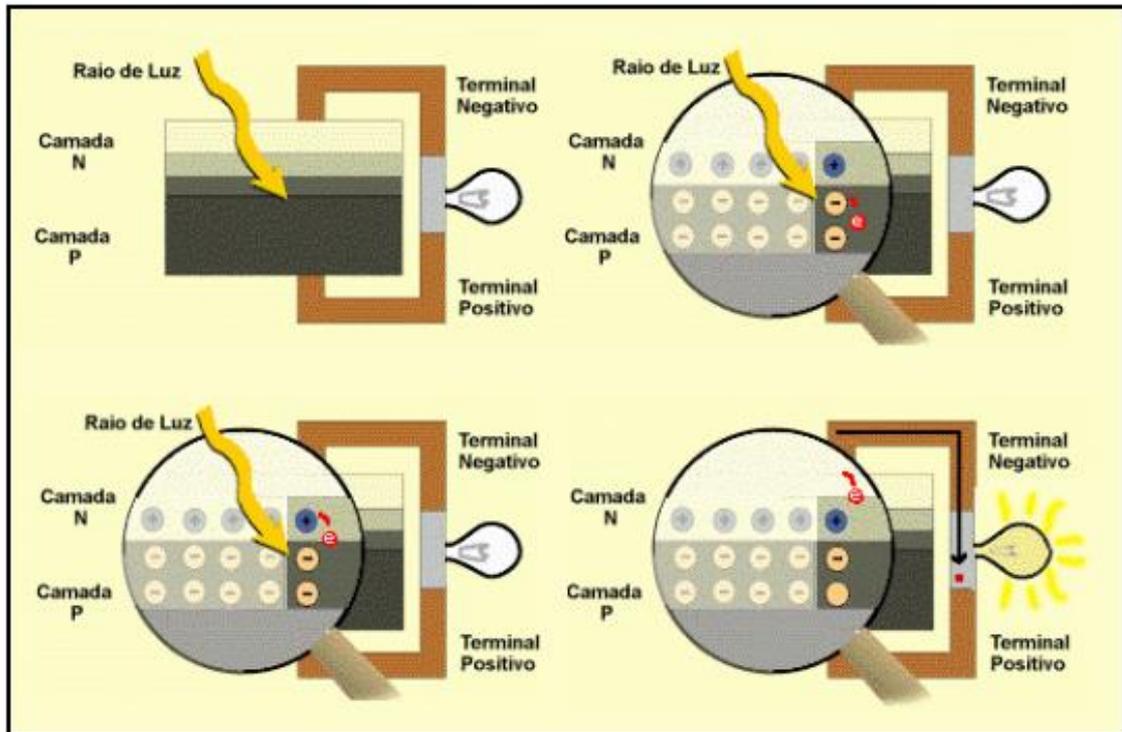
Segundo Portal Solar (2020), a energia fotovoltaica é uma forma de gerar eletricidade a partir da luz do sol, aproveitando o efeito fotovoltaico, que foi descoberto pelo físico francês Alexandre Edmond Becquerel em 1839. O efeito fotovoltaico consiste na conversão direta da luz em corrente elétrica por meio de materiais semicondutores, como o silício. A primeira célula fotovoltaica foi criada pelo inventor norte-americano Charles Fritts em 1883, usando selênio revestido de ouro. Desde então, a tecnologia evoluiu muito, tornando-se mais eficiente, barata e sustentável.

3.2.1. Efeito fotovoltaico

O efeito fotovoltaico é um fenômeno físico que ocorre nas células fotovoltaicas, que são dispositivos eletrônicos fabricados com materiais semicondutores. O material mais comumente utilizado nestas células é o silício, que normalmente é encontrado na forma de areia. Como o silício é um mal condutor, ocorre um processo denominado dopagem, acrescentando-se outros elementos para tornar o silício um material do tipo N e P, mais conhecido como silício tipo N e silício tipo P (DO NASCIMENTO, 2004).

Esta junção PN ilustrada na Figura 2, quando exposta a fótons, que são partículas que transportam a energia contida em radiações eletromagnéticas, acabam gerando pares de elétrons. A partir da geração desses pares, ocorre a aceleração de corrente elétrica e o deslocamento dessas cargas, mais conhecido como efeito fotovoltaico (CRESESB, 2006).

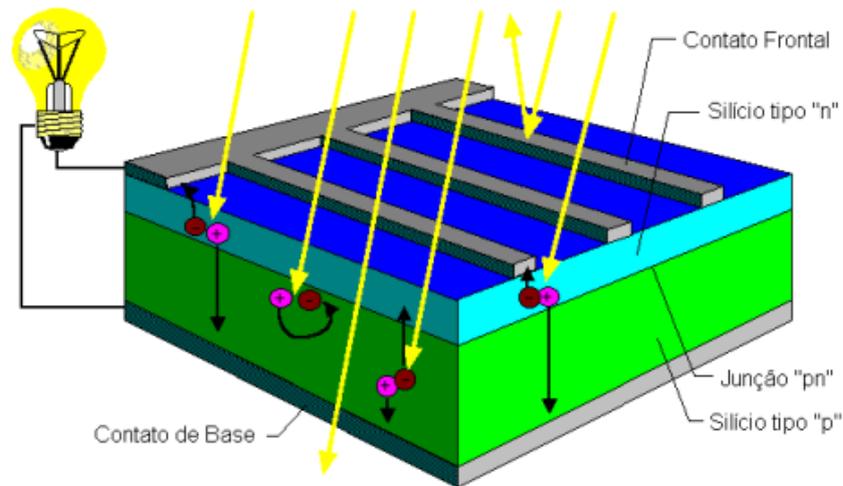
Figura 2 – Efeito fotovoltaico na junção PN



Fonte: CRESESB (2006, p. 13)

Por meio da ligação de um condutor externo, como um fio, e durante a incidência de raios solares, haverá o fluxo de elétrons (corrente elétrica) e será possível gerar eletricidade. Uma célula fotovoltaica não armazena energia elétrica, esta é fundamental para gerar esse fluxo de corrente elétrica quando há raios solares incidentes. O efeito fotovoltaico é a base do funcionamento das células fotovoltaicas (Figura 3), que convertem a energia da luz solar diretamente em eletricidade (SILVA; ABREU, 2017).

Figura 3 – Corte transversal de uma célula fotovoltaica



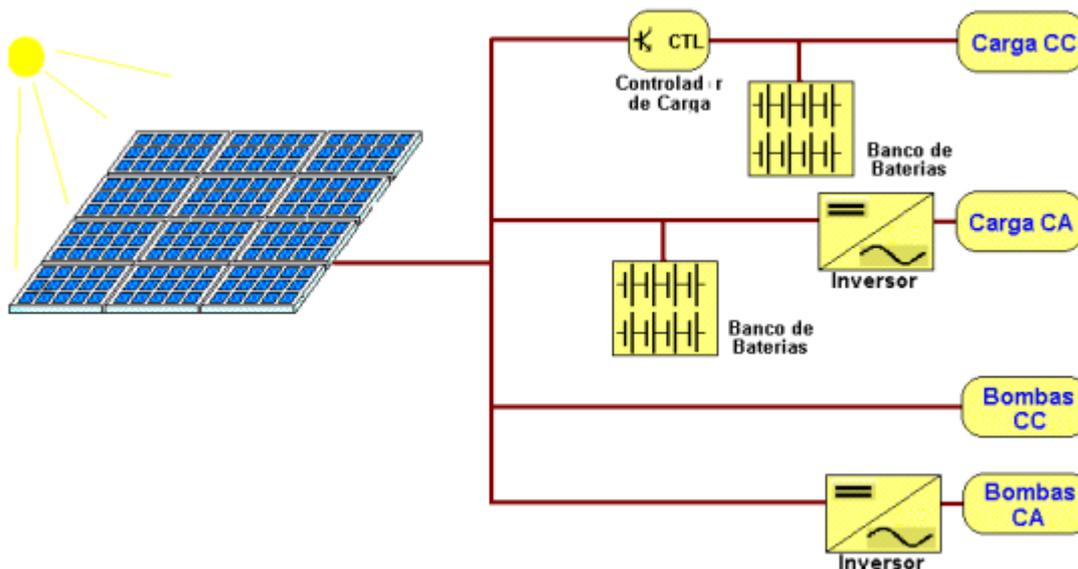
Fonte: CRESESB (2006, p. 13)

3.2.2. Tipos de sistemas fotovoltaicos

Segundo CRESESB (2006), configuração básica de um sistema fotovoltaico consiste em quatro componentes principais: painéis solares, controladores de carga, inversores e baterias. Os painéis solares são responsáveis por transformar a energia solar em eletricidade de corrente contínua (CC). Os controladores de carga regulam a tensão e a corrente que chegam às baterias, evitando sobrecargas ou descargas excessivas. Os inversores convertem a eletricidade de CC em corrente alternada (CA), que é a forma mais comum de energia utilizada em residências e estabelecimentos comerciais. As baterias armazenam a energia gerada pelos painéis solares para que possa ser usada quando não há sol ou quando a demanda é maior do que a oferta.

Sistemas fotovoltaicos isolados (Figura 4) são sistemas que não estão conectados à rede elétrica e dependem de baterias para armazenar a energia gerada pelos painéis solares. Esses sistemas são indicados para locais remotos ou de difícil acesso, onde a conexão com a rede é inviável ou muito cara. Sistemas fotovoltaicos isolados podem fornecer energia para diversas aplicações, como iluminação, telecomunicações, bombeamento de água e refrigeração (CÂMARA, 2011).

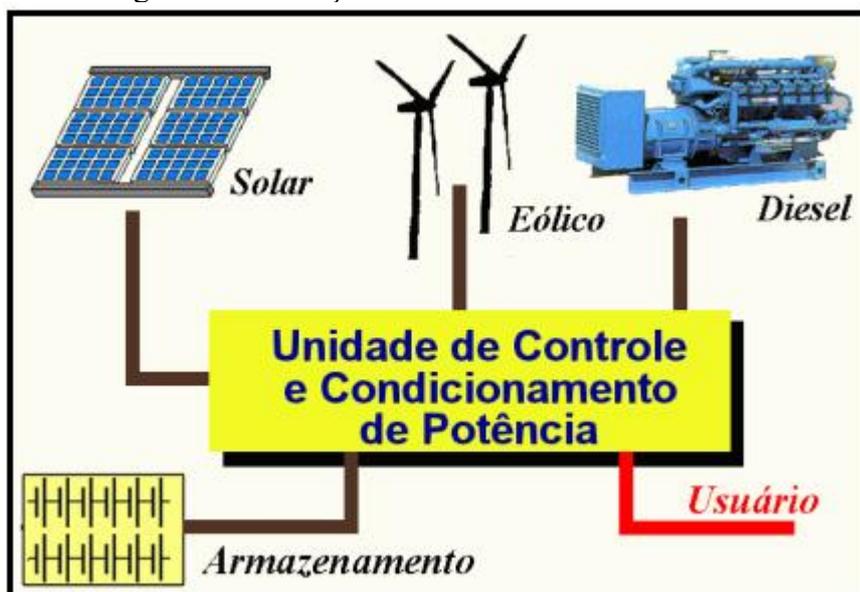
Figura 4 – Ilustração de sistema fotovoltaico isolado



Fonte: CRESESB (2006, p. 22)

Os sistemas fotovoltaicos híbridos (Figura 5) são aqueles que combinam a geração de energia solar com outra fonte de energia ou com o armazenamento em baterias. Estes sistemas permitem aproveitar melhor a energia solar e garantir o fornecimento de eletricidade em caso de falhas na rede ou durante a noite, tornando o sistema mais flexível, se comparado ao sistema fotovoltaico isolado. Os sistemas fotovoltaicos híbridos podem ser de diferentes tipos, dependendo da sua conexão com a rede elétrica e da fonte de energia complementar. Além disso, podem ser conectados a outros sistemas de geração de energia, como eólico, ou ligados a baterias para armazenamento de carga (BRANDÃO, 2013).

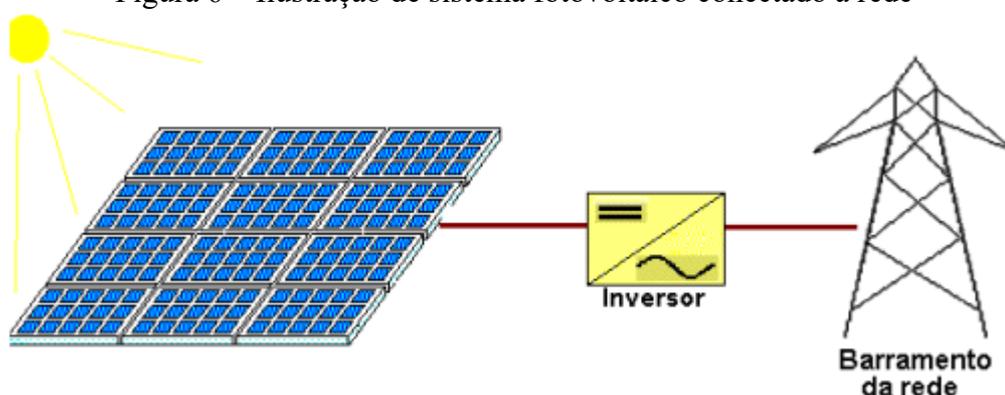
Figura 5 – Ilustração de sistema fotovoltaico híbrido



Fonte: CRESESB (2006, p. 22)

Segundo Jardim *et al.* (2004) os sistemas fotovoltaicos interligados à rede (Figura 6) são sistemas que convertem a energia solar em eletricidade e a injetam na rede elétrica. A vantagem desse tipo de sistema é que ele pode reduzir a dependência de combustíveis fósseis e contribuir para a diversificação da matriz energética. Além disso, os sistemas fotovoltaicos interligados à rede podem gerar créditos de energia para os consumidores que possuem sistemas de geração distribuída. O arranjo de placas solares é conectado em inversores, os quais garantem a qualidade e fornecimento de energia ideal para a rede. Estes inversores convertem a energia gerada, tensão e corrente contínua (CC), para tensão e corrente alternada (CA) utilizada na rede de distribuição.

Figura 6 – Ilustração de sistema fotovoltaico conectado à rede



Fonte: CRESESB (2006, p. 23)

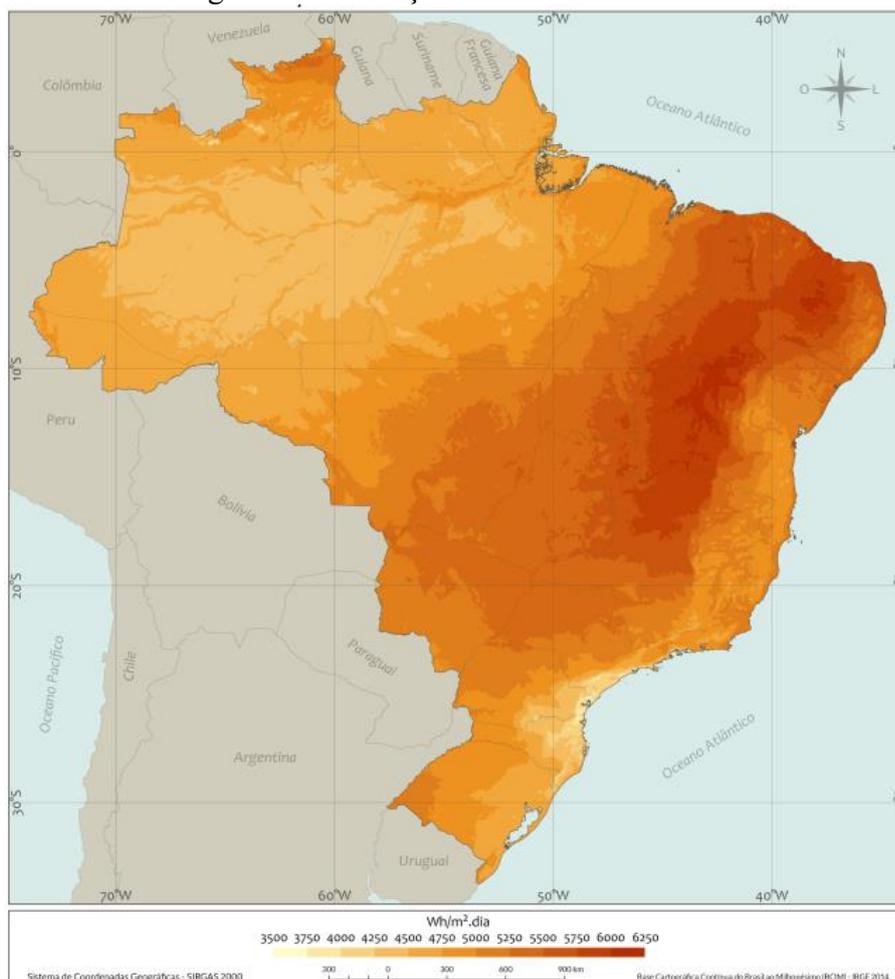
3.3. O avanço da energia solar no Brasil

A energia solar é uma fonte altamente vantajosa, caracterizada por sua natureza limpa, renovável e seu potencial estratégico para impulsionar o desenvolvimento sustentável no Brasil. O país tem registrado progressos significativos nos últimos anos tanto na ampliação da capacidade instalada, quanto na produção de energia elétrica proveniente da radiação solar. Somente no ano de 2022, usinas de grande porte e sistemas distribuídos de menor escala totalizaram 13 Gigawatts (GW) de usinas solares instaladas, sendo que 8,7 GW correspondem a micro e minigeração distribuídas (MAURICIO, 2022). É importante destacar que a ascensão das pequenas usinas de geração distribuída tem desempenhado um papel fundamental no crescimento do setor solar no Brasil, resultado de uma diretriz estabelecida em 2012 pela Agência Nacional de Energia Elétrica (ANEEL), a qual autorizou a interconexão de pequenas usinas geradoras de energia à rede elétrica nacional (RIGO *et al.*, 2019).

O Brasil oferece condições extremamente propícias para a implantação de fontes de energia renováveis, especialmente no que se refere à energia eólica e solar. Isto deve-se à sua localização geográfica privilegiada, atravessando a linha do equador e seu clima tropical, que

garante uma incidência significativa de radiação solar ao longo do ano (Figura 7). Estes fatores posicionam o país de forma favorável para o desenvolvimento dessas fontes de energia em ampla escala territorial. No entanto, mesmo apresentando estas vantagens, o Brasil demorou para dar os primeiros passos no desenvolvimento de parques solares, contudo há esforços de universidades, centros de pesquisas e ações governamentais para alavancar o avanço tecnológico nesta área (DA SILVA; ARAUJO, 2022).

Figura 7 – Irradiação média anual no Brasil

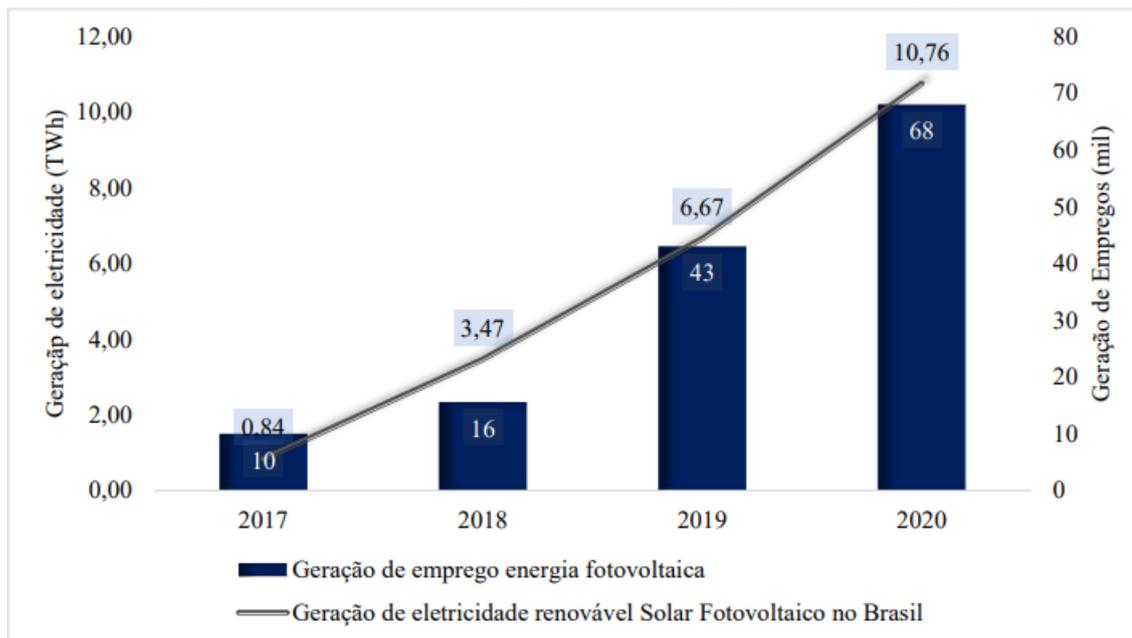


Fonte: MARTINS *et al.* (2017, p. 40)

Além de impulsionar significativamente a geração de energia elétrica renovável, o Brasil tem experimentado um notável crescimento na empregabilidade do setor nos últimos anos. Entre 2017 e 2020, o número de profissionais empregados nessa área aumentou de 10 mil para 68 mil pessoas, refletindo um avanço expressivo. No entanto, esse aumento no emprego não acompanhou proporcionalmente o crescimento da energia solar, que apresentou um crescimento superior a 500% durante o mesmo período, conforme mostrado na Figura 8. Dentre os possíveis fatores que contribuíram para essa disparidade, destaca-se a dependência de importação de

painéis solares, o que resulta em uma redução da produção local e, conseqüentemente, impacta negativamente na geração de empregos (DOS SANTOS, 2023).

Figura 8 – Geração de energia renovável x geração de emprego.



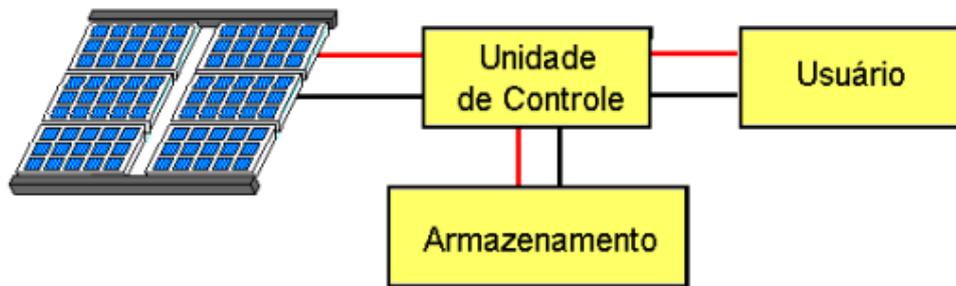
Fonte: DOS SANTOS (2023, p. 34).

3.4. Sistema de supervisão para usina fotovoltaica

Segundo CRESESB (2014), um sistema de geração distribuída baseado em energia fotovoltaica é composto por diferentes elementos, como: bloco gerador, que inclui os módulos fotovoltaicos e seus componentes; bloco de conversão de energia, que envolve o inversor de frequência, dispositivos de proteção, supervisão e controle; bloco de armazenamento, em alguns casos, com baterias para situações de falta de energia elétrica (Figura 9).

Embora seja considerado altamente confiável, o sistema de geração distribuída pode apresentar desafios na detecção de falhas devido ao seu difícil acesso, quando instalado em áreas remotas, e operação de forma silenciosa. Para lidar com estas dificuldades, é comum a implementação de sistemas de supervisão e controle que visam otimizar a operação. A monitoração e avaliação dos parâmetros elétricos, como a quantidade de energia produzida, desempenham um papel fundamental para garantir um funcionamento eficiente e confiável do sistema (MELO *et al.*, 2020).

Figura 9 – Configuração básica de um sistema fotovoltaico



Fonte: CRESESB (2006, p. 21)

A supervisão e o controle de um sistema de usina fotovoltaica são cruciais para garantir seu desempenho, eficiência e confiabilidade ao longo do tempo. Através de um sistema de supervisão, é possível monitorar em tempo real as variáveis elétricas e operacionais do sistema, permitindo a detecção precoce de falhas, identificação de áreas de baixo rendimento e implementação de ações corretivas efetivas (PORTAL SOLAR S.A, 2020).

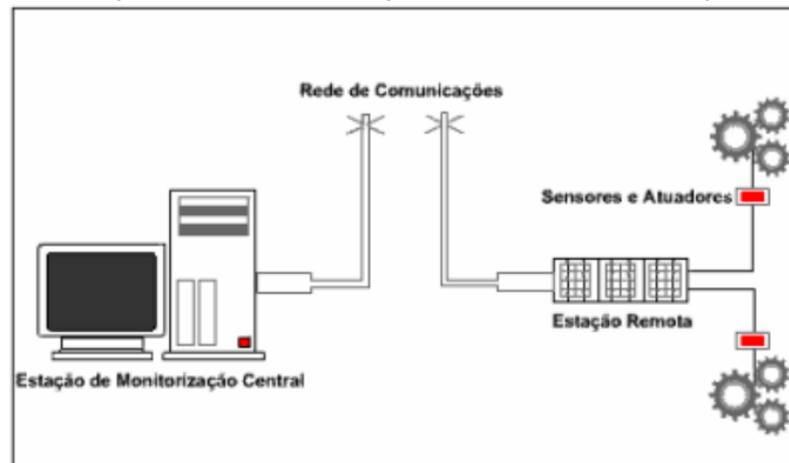
Além disso, a supervisão contínua dos parâmetros elétricos, como tensão, corrente, frequência e potência, proporciona uma visão abrangente do desempenho do sistema e facilita a identificação de qualquer desvio em relação aos valores esperados. Logo, medidas preventivas podem ser tomadas para otimizar a produção de energia e prolongar a vida útil dos componentes do sistema (DOS SANTOS; MARINATO; DOS SANTOS, 2021).

3.4.1. Controle Supervisório e Aquisição de Dados

SCADA é a sigla em inglês para Supervisory Control and Data Acquisition, ou seja, controle supervisório e aquisição de dados. Trata-se de um sistema que permite monitorar e controlar remotamente processos industriais, como a geração de energia elétrica a partir de fontes renováveis. Um sistema SCADA (Figura 10) é composto por quatro elementos principais: sensores, controladores, unidades terminais remotas (UTRs) e estações mestras (DANEELS; SALTER, 1999).

Os sensores são dispositivos que medem variáveis físicas, como corrente, tensão, temperatura e radiação solar. Os controladores são dispositivos que executam algoritmos de controle e lógica para atuar sobre os equipamentos do processo, como inversores, disjuntores, válvulas. As UTRs são dispositivos que coletam os dados dos sensores e dos controladores, transmitindo-os para as estações mestras, computadores que armazenam, processam e exibem as informações em telas gráficas (DANEELS; SALTER, 1999).

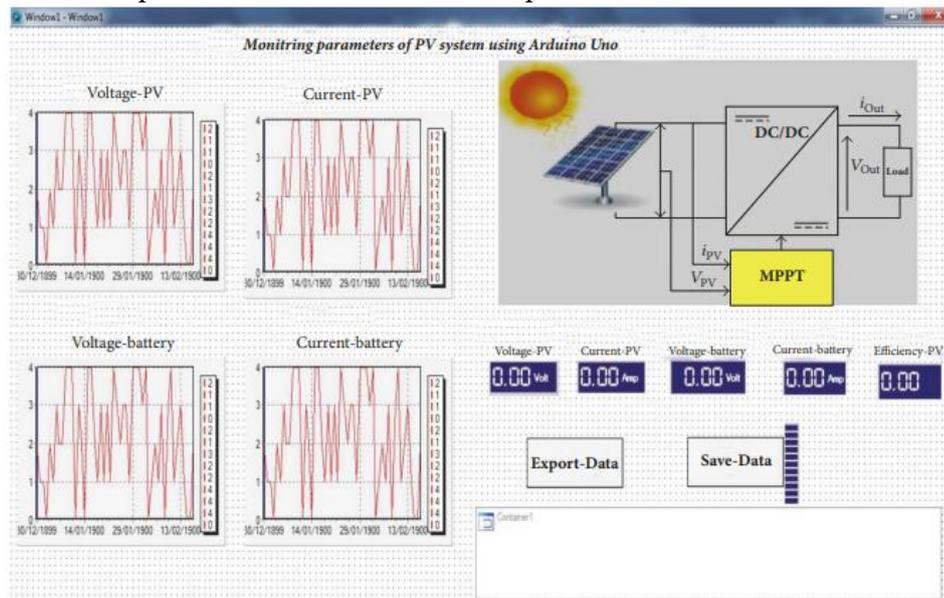
Figura 10 – Comunicação de um sistema supervisorio com uma estação remota (SCADA).



Fonte: DOS SANTOS; MARINATO; DOS SANTOS (2021, p.8)

Um sistema SCADA aplicado a um sistema fotovoltaico (Figura 11) tem como objetivo principal otimizar o desempenho e a eficiência da geração de energia solar. Para isso, o sistema SCADA monitora continuamente os parâmetros elétricos dos módulos e das strings fotovoltaicas, bem como as condições ambientais e meteorológicas. Com um sistema de supervisão e controle adequado, é possível detectar e corrigir anomalias, falhas e perdas no sistema fotovoltaico, aumentando assim a disponibilidade, confiabilidade e rentabilidade da planta. Além disso, permite integrar o sistema fotovoltaico à rede elétrica, cumprindo os requisitos técnicos e regulatórios exigidos pelo operador do sistema (ALLAFI; IQBAL, 2018).

Figura 11 – Exemplo de uma interface SCADA para monitoramento de usina fotovoltaica



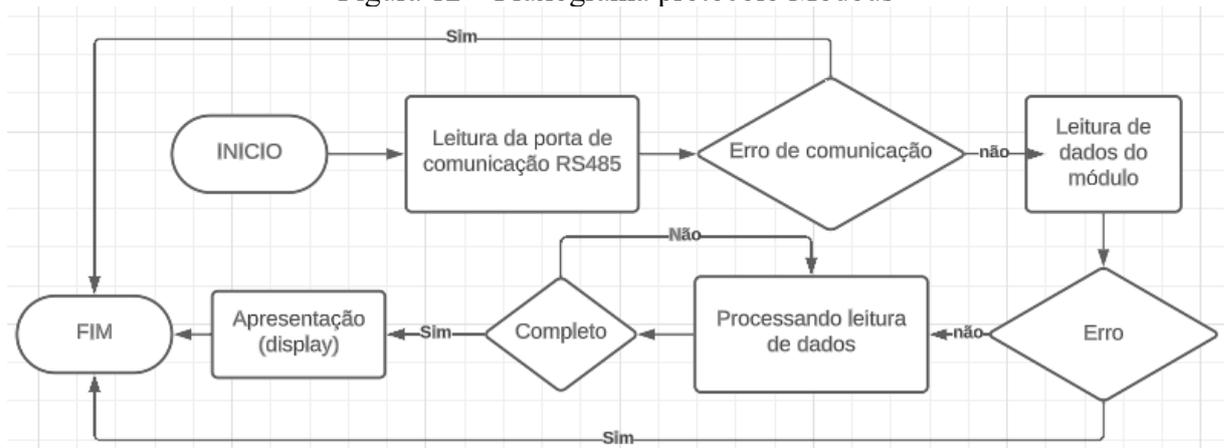
Fonte: ALLAFI; IQBAL (2018, p. 6)

3.4.2. Protocolo MODBUS

O protocolo Modbus (Figura 12) é um dos mais utilizados na indústria 4.0, pois permite a comunicação rápida e eficiente entre dispositivos de automação e de campo. O protocolo Modbus pode operar em diferentes meios físicos, como RS-232, RS-485 ou Ethernet TCP/IP, e em diferentes modos de transmissão, como Modbus TCP ou Modbus RTU (ZHANG *et al.*, 2022).

O protocolo Modbus utiliza uma estrutura de mensagem baseada na arquitetura mestre-escravo ou cliente-servidor, onde o mestre ou cliente envia uma requisição aos escravos ou servidores, que respondem com os dados solicitados ou uma confirmação. Este é capaz de enviar dados numéricos, analógicos ou discretos. Além disso, é compatível com diversos equipamentos de diferentes fabricantes, além de ser uma solução simples, econômica e abrangente para a integração de dispositivos na indústria 4.0 (ZHANG *et al.*, 2022).

Figura 12 – Fluxograma protocolo Modbus



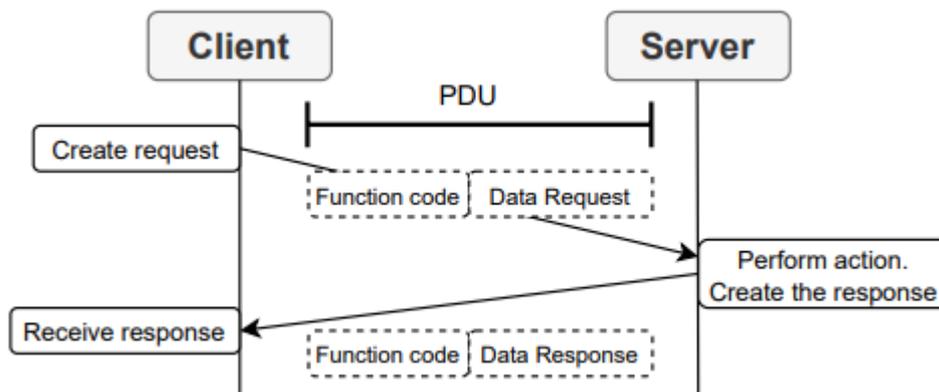
Fonte: Modificado de ZHANG *et al.* (2022, p. 754)

A comunicação Modbus RTU (RS-485 e RS-232) é realizada de forma serial e permite a transmissão de dados entre vários dispositivos em uma rede. Ela é amplamente utilizada em sistemas fotovoltaicos para monitorar e controlar os inversores, os controladores de carga e os medidores de energia. O protocolo oferece vantagens como alta velocidade, longa distância, baixo ruído e resistência à interferência eletromagnética (KUMAR; SETHURAMAN, 2018).

Segundo RODRÍGUEZ-PÉREZ *et al.* (2022), a comunicação Modbus TCP, representada na Figura 13, é uma implementação do protocolo Modbus baseado em TCP/IP. Este protocolo é muito utilizado na indústria para interligar dispositivos de controle e de supervisão, como a interface homem máquina (IHM). O Modbus TCP permite a troca de dados numéricos ou discretos entre os dispositivos, seguindo uma estrutura de mensagem no formato cliente-servidor, ou seja, o cliente solicita ao servidor por meio do código da aplicação e o

servidor processa o dado recebido pelo cliente e retorna com a ação ou resposta desejada, também é amplamente utilizada na indústria 4.0.

Figura 13 – Comunicação simplificada do protocolo Modbus TCP



Fonte: RODRÍGUEZ-PÉREZ *et al.* (2022, p. 2)

3.5. Python

Segundo Srinath (2017), a linguagem Python tem se tornado cada vez mais popular nos últimos anos, graças à sua simplicidade, versatilidade e eficiência. Python é uma linguagem de alto nível, interpretada e multiplataforma, que permite desenvolver desde aplicações web e desktop até inteligência artificial e ciência de dados. O Python também possui uma grande comunidade de desenvolvedores e uma vasta biblioteca de módulos e pacotes que facilitam a integração com outras tecnologias e ferramentas. Esta linguagem, segue o princípio da legibilidade e da elegância do código, o que torna a sua aprendizagem e manutenção mais fáceis e agradáveis.

O Python possui uma versatilidade tão abrangente que consegue se comportar tanto como um sistema back-end como um sistema front-end, com isto é possível desenvolver interfaces gráficas e sistemas de aquisição de dados em uma só aplicação. A versatilidade permite executar o mesmo código em diversos sistemas operacionais, sejam eles, Linux, Mac ou Windows. Além disso, o código open source permite que qualquer pessoa altere e construa suas aplicações do zero, utilizando uma vasta biblioteca de pacotes que facilitam a implementação de novas aplicações (SRINATH, 2017).

4. MATERIAL E MÉTODOS

A seguir será apresentado os materiais e métodos utilizados para desenvolvimento do trabalho, citando desde os componentes utilizados, como inversor, placas fotovoltaicas, até os softwares utilizados para desenvolvimento do sistema de supervisão.

4.1. Componentes do Sistema Fotovoltaico

Inicialmente decidiu-se o tipo de sistema (conectado à rede, híbrido ou isolado) e seus componentes a serem utilizados, como apresentado por Jardim *et al.* (2004). Os sistemas conectados à rede podem contribuir para a diminuição do gás carbônico gerado na atmosfera, levando em consideração que a Universidade Federal de Lavras (UFLA) é uma instituição que contribui com a preservação ambiental, esse sistema foi o escolhido. Para a bancada de testes do sistema supervisorio utilizou-se um inversor (Growatt) conectado à rede, três placas solares (Honey – Trina Solar) e um conjunto de disjuntores (disjuntor de carga CC e disjuntor de carga CA). Todos esses equipamentos foram fornecidos pela UFLA, e estavam disponíveis no laboratório de combustíveis e combustão no prédio da ABI (Area Básica de Ingressos).

O inversor utilizado possui tensão máxima de entrada de 500 Volts, corrente máxima de entrada de 13 Amperes e potência máxima de 1500 Watts, e sua temperatura de operação entre $-25\text{ }^{\circ}\text{C}$ – $60\text{ }^{\circ}\text{C}$, este pode operar tanto em modo híbrido, que seria carregando uma bateria e conectado à rede, ou somente conectado à rede. Neste caso foi utilizado somente conectado à rede (Tabela 1).

Tabela 1 – Especificações do Inversor

Modelo	MIC 1500TL-X
Tensão Máxima	500 d.c.V
Faixa de tensão PV	50-500 d.c.V
Corrente de curto circuito PV (Isc)	16 d.cA
Corrente c.c. Máxima	13 d.cA
Potência CA Máxima	1500 w
Potência VA Máxima	1500 VA
Tensão CA Nominal	220 a.c.V
Corrente CA Máxima	7.1 a.cA
Frequência de saída Nominal	60 Hz
Faixa de Temperatura de Operação	-25°C a +60°C

Fonte: Do autor (2023)

As placas fotovoltaicas (Honey – Trina Solar) utilizadas possuem eficiência de 20,7% e geram entre 360 – 380 W, sendo empregadas três placas solares do modelo TSM-DE08M(II).

De acordo com Dogado (2018), quando as placas fotovoltaicas são conectadas em série, a tensão é somada enquanto a corrente permanece a mesma em todos os módulos. Com base

nessa análise e levando em consideração as especificações das placas e do inversor utilizados, foi decidido ligar as placas em série (Figura 14). Essa escolha deve-se ao fato de que, caso as três placas atinjam o máximo de potência gerada, o inversor será capaz de captar essa energia e distribuí-la de forma eficiente na rede elétrica.

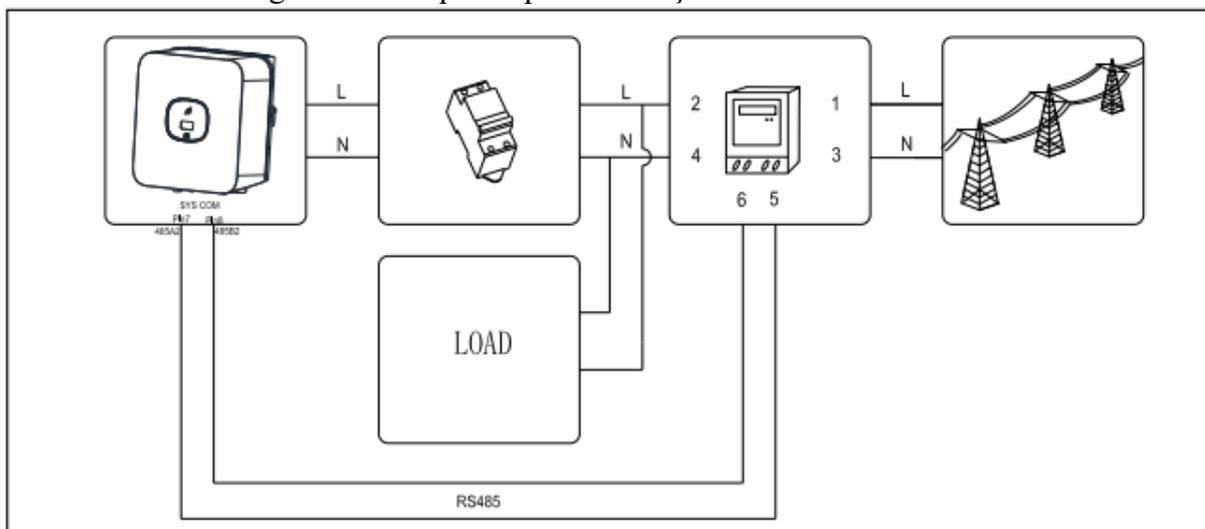
Figura 14 – Placas fotovoltaicas conectadas em série



Fonte: Do autor (2023)

Com a utilização de cabos de energia, seguiu-se a instrução do manual do inversor da Growatt (Figura 15) para união dos componentes citados, desse modo o sistema foi montado e conectado à rede elétrica da UFLA.

Figura 15 – Esquema para instalação do inversor Growatt



Fonte: GROWATT (2020, p. 2)

4.2. Protocolo de Comunicação

Para a aquisição de dados do inversor, foi necessário verificar junto do seu manual e equipamento, quais protocolos eram suportados por ele. Diante disso, utilizou-se o protocolo Modbus RS-485 para aquisição de dados, visto que o modelo disponível na UFLA possui a porta COM (Figura 16) para comunicação entre mestre-escravo.

Figura 16 – Entrada de comunicação (COM) do inversor



Fonte: Do autor (2023)

Após, ligou-se os cabos ao suporte proveniente do inversor nos respectivos locais de comunicação (pino 3, laranja e 4, preto), como mostrado na Figura 16 e 17. Em seguida, para a conexão dos cabos no computador foi utilizado um conversor USB, que recebe os dados via RS-485 e converte para USB, podendo assim ser conectado a qualquer sistema operacional, seja Linux, Mac ou Windows.

Figura 17 – Pinos de conexão da comunicação do inversor.

COM Port Pin Definitions		
COM Port	Definition	
1	NA	This pin is no signal
2	NA	
3	RS 485A1	Signal for communication
4	RS 485B1	
5	CT-P	Signal for export limitation (Optional)
6	CT-N	
7	RS 485A2	Signal for smart meter
8	RS 485B2	

Fonte: GROWATT (2020)

4.3. Configuração do Sistema Operacional e Linguagem de Programação

Para que fosse possível utilizar o conversor RS485-USB (Figura 18), foi necessário instalar os drivers do dispositivo em um computador com o sistema operacional Windows. Utilizou-se para instalação dos drivers o programa chamado Driver Booster (versão 10.5). O driver instalado para funcionamento do conversor foi o “*USB to UART Bridge Controller*”.

Figura 18 – Conversor RS485-USB



Fonte: Do autor (2023)

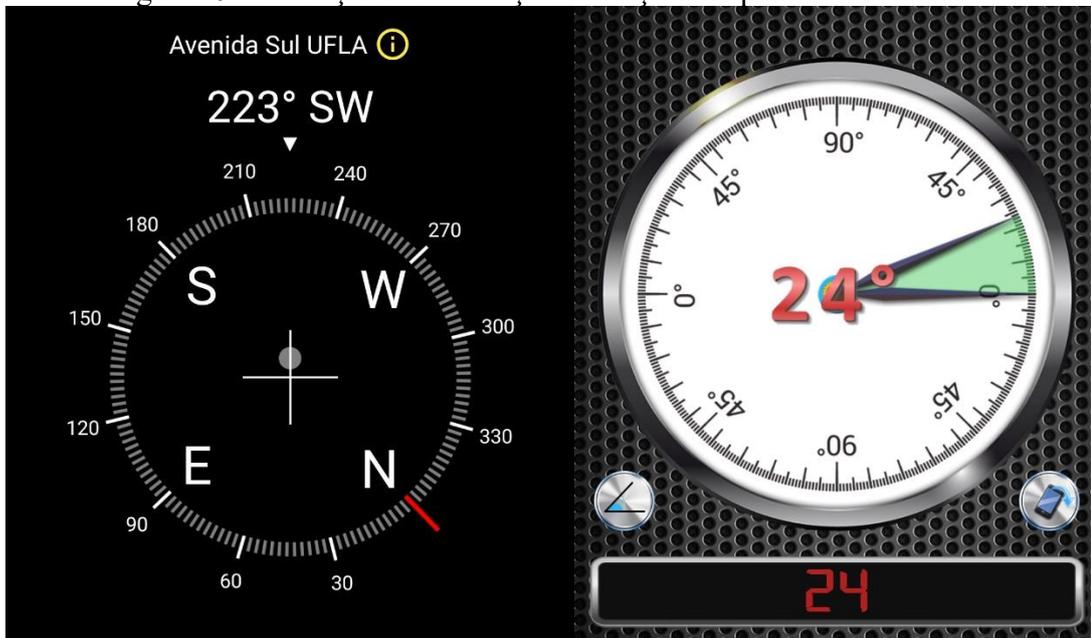
Após configurar o USB no computador, realizou-se o download e instalação de um ambiente de desenvolvimento integrado (IDE - *integrated development environment*), o VSCode (versão 1.80).

Para a aquisição dos dados por meio do USB utilizou-se uma linguagem de programação, neste caso o Python (versão 3.11). Como a IDE utilizado é muito versátil, por meio dele foram instalados o Python e as dependências que foram necessárias para o desenvolvimento da aplicação. Nesta linguagem foi possível realizar as trocas de dados com o inversor por meio de uma biblioteca chamada PyModbus (versão 2.5.3). Apenas fornecendo poucos dados, como o cliente (porta de comunicação) e os números dos registros do inversor coletados (encontrados no Manual/Datasheet), junto da lógica de programação, foi capaz de adquirir os dados provenientes do sistema fotovoltaico.

4.4. Captação dos Dados

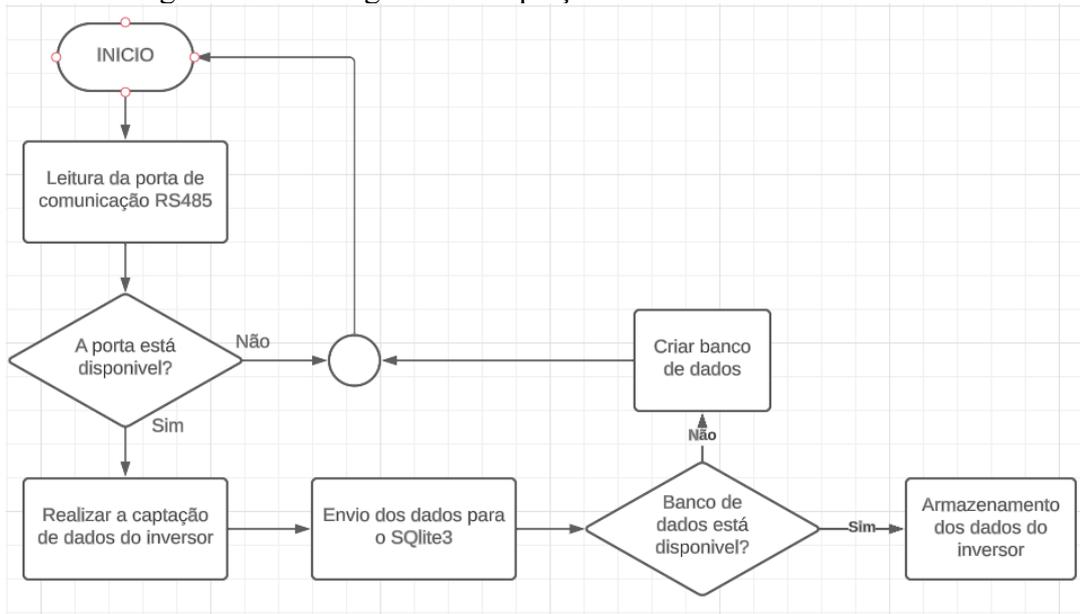
Para que o supervisor apresente os valores no banco de dados, estes foram coletados, na UFLA, no dia 29 de junho de 2023. Esses dados foram coletados em torno de 2 horas, e armazenados no banco de dados Sqlite3 para uso posteriormente na execução da interface desenvolvida para ser o supervisor do equipamento. Em relação as placas fotovoltaicas, estas ficaram inclinadas cerca de 24 graus em relação a horizontal e apontadas para noroeste, neste dia o sol estava com incidência direta sobre as placas, como pode ser visto na Figura 14. Os aplicativos usados foram, Medidor de ângulo e Digital Compass, ambos disponíveis na Google Play Store.

Figura 19 – Medição da inclinação e direção das placas fotovoltaicas



Fonte: Do autor (2023)

Figura 20 – Fluxograma da captação e armazenamento de dados

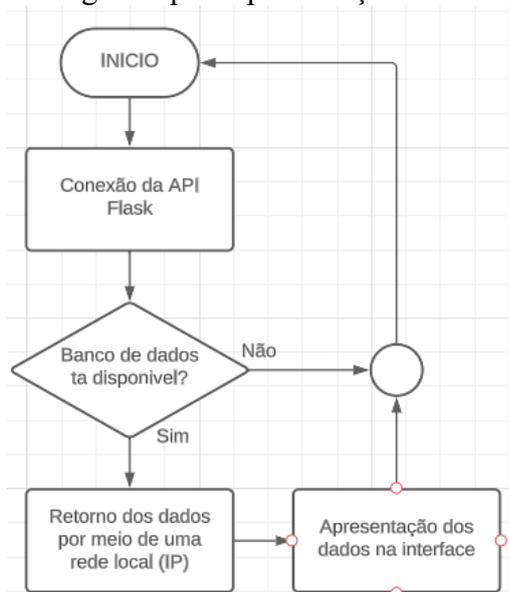


Fonte: Do autor (2023)

4.5. Supervisório

Para o desenvolvimento do supervisório foi aplicada a metodologia de Melo (2020) com modificações. Neste caso, a fim de se obter um sistema mais robusto utilizou-se o Flutter (versão 3.10), que apresenta vantagens como uma boa documentação para desenvolvimento e fácil integração entre aplicações, enquanto o Flask (Python framework) foi essencial para se integrar junto a ele, mostrando todos os dados necessários em um aplicativo móvel, desktop ou web. Para esse aplicativo foi utilizado a IDE Android Studio, sendo configurado o framework Flutter, possibilitando a programação do sistema supervisório.

Figura 21 – Fluxograma para apresentação de dados na interface



Fonte: Do autor (2023)

5. RESULTADOS E DISCUSSÃO

Neste capítulo será apresentado como os equipamentos mencionados foram montados e o ambiente de desenvolvimento configurado. Além disso, será explicado como os dados foram coletados e os testes realizados para garantir a confiabilidade na troca de informações entre o sistema desenvolvido e os equipamentos.

5.1. Bancada de Testes para o Supervisório

A fim de viabilizar a implementação de um sistema SCADA capaz de monitorar e coletar dados fotovoltaicos, os equipamentos foram conectados à rede elétrica da UFLA e verificou-se a ocorrência da captação de energia proveniente da luz solar. Conforme mostrado na Figura 22, observa-se o sistema operando, visto que o inversor dispõe de um display que exibe diversas variáveis e uma luz que representa se há geração de energia, neste caso estava verde mostrando que o sistema estava funcionando corretamente. Por meio desse display, analisou se o sistema estava gerando energia das placas solares e a quantidade de energia que estava sendo enviada à rede elétrica, uma vez que o sistema implementado é do tipo conectado à rede.

Figura 22 – Parte Frontal do Inversor



Fonte: Do autor (2023)

Conforme demonstrado na Figura 17, o inversor apresenta uma entrada para comunicação RS-485.

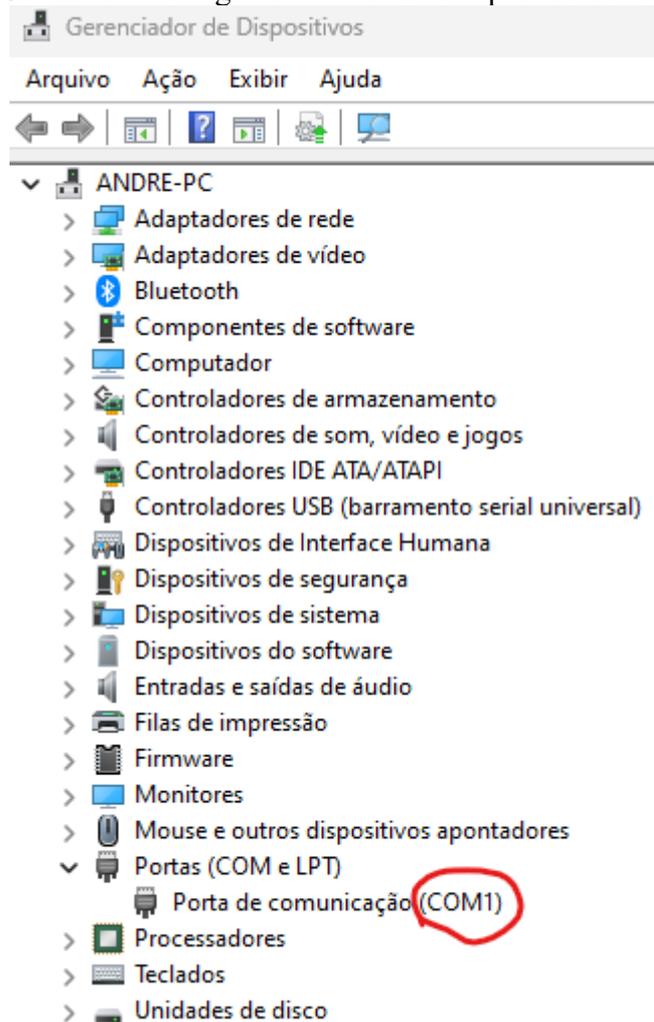
A vantagem de utilizar esse protocolo reside na despreocupação com relação à distância física. Independentemente da distância (0 a 1200 metros) do equipamento responsável pela leitura, a comunicação permanecerá confiável e rápida (KUMAR; SETHURAMAN, 2018). Diferentemente de uma comunicação Ethernet TCP/IP (MODBUS TCP), que, embora exija menos equipamentos para operar e não apresente problemas de conexão do cabo, visto que as extremidades possuem os mesmos padrões de entrada, pode estar sujeita a ataques de rede, pois requer um cliente-servidor em tempo real para a comunicação. No caso do RS-485, não há necessidade de utilizar protocolos de internet, apenas a definição de mestre-escravo.

5.2. Conversor RS485-USB e Sistema Operacional

Para a captação dos dados provenientes do inversor foi necessário adaptar sua saída de dados utilizando um conversor, o qual possui a função de modificar a saída RS485 para a forma serial USB, essa adaptação garante que uma ampla gama de dispositivos compatíveis com a interface USB possa receber e processar os dados do inversor. Ao fornecer essa capacidade de conexão, o conversor abre um leque de possibilidades para a análise e visualização dos dados coletados (Figura 18).

O sistema operacional utilizado na captação de dados para o supervisório foi o Windows 11, este sistema permitiu realizar as devidas configurações e testes com o conversor USB, por meio da ferramenta já presente no sistema operacional, o gerenciador de dispositivos. Essa ferramenta permitiu verificar as portas COM disponíveis no sistema, identificando aquela que seria utilizada posteriormente para a requisição de dados do supervisório, como mostrado na Figura 23.

Figura 23 – Interface de gerenciamento de dispositivos do Windows.



Fonte: Do autor (2023)

A utilização do conversor RS485-USB aliado ao sistema operacional Windows 11 trouxe benefícios significativos para a captação de dados do inversor. Essa abordagem permitiu uma integração eficaz entre os dispositivos envolvidos, garantindo a disponibilidade e a precisão das informações necessárias para o monitoramento e controle do sistema fotovoltaico. O uso do gerenciador de dispositivos no Windows 11 proporcionou uma configuração simplificada e confiável do conversor, viabilizando a obtenção dos dados do inversor pelo sistema supervisorio.

5.3. Comunicação entre Supervisorio e Inversor

Devido à ampla compatibilidade do protocolo RS485 com diversas linguagens de programação, optou-se por utilizar o Python para estabelecer a comunicação e transmitir informações entre o inversor e o supervisorio. Para esse propósito, a biblioteca PyModbus foi empregada. Os dados resultantes foram armazenados em um banco de dados SQLite3, uma

solução leve e eficiente adequada para aplicações com necessidades de armazenamento relativamente modestas. Esse banco de dados oferece recursos para manipulação e consulta de dados, além de ser amplamente suportado e facilmente integrado com o Python, para a leitura dos dados armazenados foi utilizado a API (*Application Platform Interface*) Flask e visualização o Flutter, como pode ser demonstrado pela Figura 24.

Figura 24 – Diagrama representando a comunicação entre o Inversor e o Supervisor



Fonte: Do autor (2023)

Ao contrário do trabalho realizado por Melo (2020), que utilizou uma plataforma pré-desenvolvida com funcionalidades de armazenamento e visualização de dados, o Blynk, neste trabalho teve o objetivo de ser executado em qualquer sistema operacional e ser modular, se adaptando com facilidade com outras ferramentas e equipamentos, incluindo até mesmo um módulo Raspberry Pi com sistema Linux, visto que o Blynk possui limitação por não ser código aberto e sendo necessário desembolsar gastos financeiros para utilização da plataforma, o Python foi essencial nesta escolha, sendo uma linguagem open-source e com ampla compatibilidade entre outras linguagens de programação e comunicação disponíveis no mercado.

O código desenvolvido utilizou bibliotecas específicas para implementar a comunicação de forma confiável e eficaz. Essa comunicação permitiu a obtenção contínua dos dados do inversor, como potência gerada, tensão de saída e frequência da rede. Essas informações foram então armazenadas de maneira organizada e com índices, tudo foi configurado e manipulado diretamente com o uso do Python. Uma parcela do código mostrando como foi efetuada essa manipulação dos dados está apresentada na Figura 25, como pode ser observado a criação de variáveis no banco de dados foi muito simples utilizando a linguagem Python, resultando em um sistema robusto e simplificado para armazenamento dos dados. Todos os códigos apresentados em seguida estarão em apêndice para consulta.

Figura 25 – Código para criação da base de dados com suas variáveis.

```
1 import sqlite3
2 from datetime import datetime
3 # Criando conexão com o banco de dados
4 conn = sqlite3.connect('tcc.db')
5 # Criando a tabela "dados_inversor"
6 conn.execute('''CREATE TABLE dados_inversor
7                 (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
8                  pot DOUBLE(11),
9                  freq DOUBLE(11),
10                 tensao DOUBLE(11),
11                 corrente DOUBLE(11),
12                 potenciaCC DOUBLE(11),
13                 tensaoCC DOUBLE(11),
14                 correnteCC DOUBLE(11),
15                 potDiarria DOUBLE(11),
16                 potTotal DOUBLE(11),
17                 energDiarria DOUBLE(11),
18                 energDiaCC DOUBLE(11),
19                 datahora TIMESTAMP DEFAULT CURRENT_TIMESTAMP);''')
20 # Salvando as mudanças no banco de dados e fechando a conexão
21 conn.commit()
22 conn.close()
```

Fonte: Do autor (2023)

Para possibilitar a visualização dos dados coletados de forma amigável, foi criada uma API utilizando o framework Flask, o código dessa API está representado na Figura 26. O Flask é uma estrutura minimalista em Python para desenvolvimento de aplicações web. Por meio dessa API, os dados armazenados no banco de dados foram expostos por meio de endpoints (pontos de acesso via endereço web), os quais foram projetados para fornecer os dados em um formato adequado para serem consumidos pelo supervisor.

Figura 26 – Algoritmo da API em Flask (Python)

```
1  from flask import Flask, jsonify
2  import sqlite3
3
4  app = Flask(__name__)
5
6  @app.route('/dados')
7  def dados():
8      conn = sqlite3.connect('tcc.db')
9      cursor = conn.cursor()
10     app.logger.debug('Executando consulta SQL...')
11     cursor.execute('SELECT * FROM dados_inversor ORDER BY datahora DESC LIMIT 30')
12     dados = cursor.fetchall()
13     app.logger.debug(f'Retornado {len(dados)} registros.')
14     conn.close()
15     # Transforma a lista de tuplas em uma lista de dicionários
16     dados_dict = []
17     for row in dados:
18         dado = {}
19         dado['id'] = row[0]
20         dado['pot'] = row[1]
21         dado['freq'] = row[2]
22         dado['tensao'] = row[3]
23         dado['corrente'] = row[4]
24         dado['potenciaCC'] = row[5]
25         dado['tensaoCC'] = row[6]
26         dado['correnteCC'] = row[7]
27         dado['potDiaria'] = row[8]
28         dado['potTotal'] = row[9]
29         dado['energDiaria'] = row[10]
30         dado['energDiaCC'] = row[11]
31         dado['datahora'] = row[12]
32         dados_dict.append(dado)
33     return jsonify(dados_dict)
```

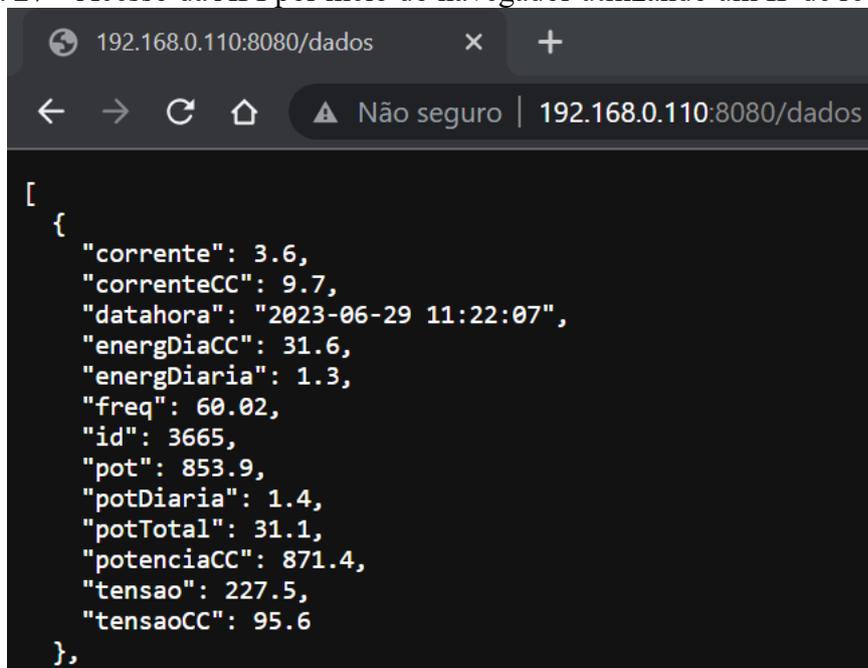
Fonte: Do autor (2023)

O formato utilizado na troca de informações entre a aplicação back-end (API) e o front-end foi o JSON (JavaScript Object Notation), este possui uma ampla adoção e suporte em diversas plataformas e linguagens de programação. Sua estrutura baseada em pares de chave-valor torna a troca de dados entre o supervisor e a aplicação mais flexível e fácil de ser interpretada. Além disso, o JSON oferece suporte a diferentes tipos de dados, como strings, números, arrays e objetos, permitindo a representação adequada dos dados coletados do inversor. Para utilização do JSON foi necessário importar o componente presente na biblioteca do Flask chamado de jsonify, a partir dele os dados representados nas colunas do banco de dados foram transformados de uma lista de tuplas para uma lista de dicionários, sendo acessados por meio de chave-valor.

A partir da criação da API em Flask, com suporte ao formato JSON, e a junção com o banco de dados, o sistema proporcionou uma interface eficiente e amigável para a troca de

informações entre o supervisor e a aplicação. Essa abordagem permitiu a visualização dos dados coletados de maneira flexível, adaptável e em tempo real, sendo acessível por meio de um IP de rede local como mostrado na Figura 27.

Figura 27 – Acesso da API por meio do navegador utilizando um IP de rede local



Fonte: Do autor (2023)

5.3.1. Recebimento dos Dados Fotovoltaicos

O código desenvolvido para o recebimento dos dados do inversor consiste em uma série de rotinas e bibliotecas específicas (Figura 28). Através do uso da biblioteca ModbusSerialClient, do PyModbus, é estabelecida a conexão com o inversor utilizando o método de comunicação RS485. O código implementa a leitura dos registradores do inversor, obtendo informações como potência, tensão, corrente, energia gerada, frequência da rede, entre outros parâmetros relevantes. Os dados obtidos são armazenados em variáveis e posteriormente inseridos no banco de dados SQLite3 por meio de uma função dedicada. Essa função realiza a inserção dos dados na tabela "dados_inversor" de forma organizada, incluindo a data e hora da obtenção dos dados. Dessa forma, é possível registrar a evolução dos parâmetros ao longo do tempo e disponibilizá-los para análise posteriormente no supervisor.

Figura 28 – Parte do código para recebimento de dados do Inversor.

```
conn = sqlite3.connect('tcc.db')
def sqlite3(vetor):
    try:
        cursor = conn.cursor()

        comando_SQL = "INSERT INTO dados_inversor (pot, freq, tensao, corrente, potenciaCC, tensaoCC,
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"

        datahora = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        dados = tuple(vetor) + (datahora,)

        cursor.execute(comando_SQL, dados)
        conn.commit()
    except sqlite3.Error as erro:
        print("Falha na comunicação: {}".format(erro))
    finally:
        if cursor:
            cursor.close()
            print("Conexão finalizada")

def run_sync_inversor(id):
    client = ModbusClient(method='rtu', port='COM7', timeout=0.5, baudrate=9600)
    client.connect()
    inversor = client.read_input_registers(2,61,unit=id)
    try:
        a = inversor.registers
    except Exception as e:
        return []
    client.close()
    return a
```

Fonte: Do autor (2023)

A implementação do código de recebimento dos dados do inversor, aliada ao uso do banco de dados SQLite3, proporciona uma solução eficiente e simplificada para o armazenamento das informações relevantes ao sistema de monitoramento. A escolha da linguagem Python, juntamente com a biblioteca PyModbus, permitiu a comunicação confiável e eficaz com o inversor, enquanto o SQLite3 ofereceu recursos de manipulação e consulta de dados necessários para atender às necessidades do sistema.

5.4. Interface Homem Máquina (IHM)

A interface de monitoramento foi desenvolvida utilizando o framework Flutter, em que possibilitou o monitoramento em dispositivos móveis de forma agradável e intuitiva. A aplicação foi projetada com o objetivo de fornecer informações relevantes sobre o desempenho do sistema de energia fotovoltaica.

A tela principal da interface exibe de forma clara e concisa os valores em tempo real da potência gerada pelo sistema, a frequência e a energia acumulada ao longo do dia. Essas

informações foram obtidas por meio da comunicação com a API que retornava os dados armazenados no banco de dados. Para que o aplicativo retornasse esses dados foi necessário fazer uma requisição HTTP para o IP local que foi demonstrado na Figura 24. A partir da conexão com esse IP foi possível mostrar os valores na tela do aplicativo como pode ser visto na Figura 29.

Figura 29 – Tela inicial do aplicativo desenvolvido



Fonte: Do autor (2023)

Outras telas foram desenvolvidas para visualização dos dados, essas telas podem ser acessadas pelo botão de Menu (Figura 30). As telas adicionais fornecem informações complementares sobre o sistema de energia fotovoltaica. Uma das telas exibe um gráfico de potência em função do tempo, permitindo uma análise visual da geração de energia ao longo do tempo. Esse gráfico é atualizado em tempo real conforme novos dados são recebidos. A outra tela é uma tabela que apresenta dados mais detalhados sobre o sistema, como a potência, energia

gerada diária, a corrente e a tensão. Essa tabela permite uma visualização mais aprofundada e análise dos valores registrados em diferentes momentos.

Figura 30 – Diagrama exibindo demais telas do aplicativo



Fonte: Do autor (2023)

Além da interface mobile, vale ressaltar que as telas e funcionalidades desenvolvidas para o aplicativo móvel também podem ser facilmente exibidas em um navegador de internet, graças à versatilidade do framework Flutter. Com a opção de compilação para Flutter Web, a aplicação tem a possibilidade de mostrar todas as informações relevantes sobre o desempenho do sistema de energia fotovoltaica diretamente de um navegador. Isso proporciona flexibilidade de execução da aplicação, permitindo que possa ser utilizada em qualquer plataforma que tenha acesso a um navegador, como por exemplo o Google Chrome.

6. CONCLUSÃO

A partir do desenvolvimento do sistema de aquisição e supervisão de dados, foi possível obter um ambiente completo de monitoramento de sistema fotovoltaico a partir da utilização de linguagens de programação (Python e Flutter) e o protocolo de comunicação (RS-485). Os equipamentos montados funcionaram de maneira correta, transformando a energia solar em elétrica, por meio do inversor. Além disso, o modelo de inversor utilizado permitiu o uso do protocolo de comunicação RS-485, facilitando a adaptação do sistema de aquisição de dados, tornando um sistema completo, seguro e compatível com diversos sistemas operacionais. O desenvolvimento de uma aplicação back-end, utilizando o Flask e Python, foram eficazes para o armazenamento e recuperação de dados, de forma que puderam ser visualizados pela interface de supervisão mobile.

Através da aplicação SCADA que foi desenvolvida, torna-se possível a sua utilização em conjunto com sistemas fotovoltaicos, abrangendo tanto usinas fotovoltaicas como sistemas de geração particulares. Esse sistema também é aplicável a outros projetos que utilizem inversores e tenham comunicação RS-485. Ele permite, de forma remota, o monitoramento e aquisição de dados, proporcionando um conjunto robusto, econômico e altamente eficiente.

Uma das grandes vantagens desse sistema é a utilização de linguagens de programação sem custo adicional, o que o torna ainda mais acessível. Além disso, as linguagens utilizadas permitem adaptações conforme a quantidade de equipamentos e variáveis presentes no sistema, oferecendo assim uma maior flexibilidade de personalização. Essa característica é especialmente útil para atender às necessidades específicas de diferentes projetos.

Portanto, a aplicação SCADA desenvolvida representa uma solução tecnologicamente avançada. Sua versatilidade, juntamente com sua eficiência e baixo custo de implementação, torna-o uma escolha promissora e vantajosa para diferentes aplicações no campo da energia fotovoltaica.

7. TRABALHOS FUTUROS

De modo geral o desenvolvimento dessa plataforma chegou ao seu objetivo final, para trabalhos futuros, vale adaptar esse sistema a outros protocolos de comunicação, como o Modbus TCP/IP, permitindo a execução e comunicação em mais equipamentos disponíveis no mercado. Além disso, outra sugestão seria implementar inteligência artificial para identificar problemas de forma automática em sistemas fotovoltaicos, como falhas elétricas, mecânicas ou equipamento danificado. Visto que a linguagem Python possui um grande mercado em Machine Learning e contém diversas bibliotecas para análise de dados, como a NumPy e Pandas. Por fim, um aprimoramento que pode ser feito, é a junção de todos os algoritmos (captação, armazenamento e supervisão) e adaptar em um módulo de Raspberry Pi com um monitor externo, obtendo desse modo um pequeno protótipo SCADA, sem a necessidade de um desktop ou notebook.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALBUQUERQUE, L.; FAGUNDEZ, G. T.; MOSMANN, M. P. Litigância climática como instrumento indutor da descarbonização da matriz energética brasileira. **Revista Videre**, v. 11, n. 22, p. 154-170, 2019.
- ALLAFI, I.; IQBAL, T. Low-cost SCADA system using arduino and reliance SCADA for a stand-alone photovoltaic system. **Journal of Solar Energy**, v. 2018, p. 1-8, 2018.
- BRANDÃO, D. I. **Sistema de geração fotovoltaico multifuncional**. 2013, 135 p. Dissertação (Mestrado em Engenharia) – Universidade Estadual Paulista Julio de Mesquita Filho, Bauru, 2013.
- CÂMARA, C. F. **Sistemas Fotovoltaicos conectados à rede elétrica**. 2011, 67 p. Monografia (Pós-graduação em Formas Alternativas de Energia) - Universidade Federal de Lavras, Lavras, 2011.
- CRESESB, Centro de Referência para Energia Solar e Eólica de Salvo Brito. **Manual de engenharia para sistemas fotovoltaicos**. Rio de Janeiro, 2014.
- CRESESB, Centro de Referência para Energia Solar e Eólica de Salvo Brito. **Energia Solar Princípios e Aplicações**. Rio de Janeiro, 2006.
- DA SILVA, H. M. F.; ARAÚJO, F. J. C. Energia solar fotovoltaica no Brasil: uma revisão bibliográfica. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, v. 8, n. 3, p. 859-869, 2022.
- DANEELS, A.; SALTER, W. What is SCADA? **International Conference on Accelerator and Large Experimental Physics Control Systems**, p. 339 – 343, 1999.
- DE OLIVEIRA, A. P. M. *et al.* Análise técnica e econômica de fontes de energia renováveis. **The Journal of Engineering and Exact Sciences**, v. 4, n. 1, p. 0163-0169, 2018.
- DE SOUZA, M.; FERREIRA, L. G. Sustainability Transition Challenges in the Brazilian Energy Sector. **CEBRI-Revista: Brazilian Journal of International Affairs**, n. 4, p. 104-127, 2022.
- DOGADO, L. E. **Análise do comportamento de módulos fotovoltaicos quando conectados em série e paralelo**. 2018. 49 f. Trabalho de Conclusão de Curso (Tecnologia em Automação Industrial) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2018.

DO NASCIMENTO, C. A. **Princípio de funcionamento da célula fotovoltaica**. 2004, 21 p. Dissertação Pós-graduação Lato-sensu em Fontes alternativas de energia – Universidade Federal de Lavras, Lavras, 2004.

DOS SANTOS, F. T.; MARINATO, M. M; DOS SANTOS, W. C. **Sistema supervisorio**. Rede de Ensino de Vitória, 41 p., 2021.

DOS SANTOS, V. V. **O crescimento da geração de energia solar fotovoltaica pela perspectiva da geração de empregos (2017-2022)**. Monografia (Bacharel em Ciências Econômicas) - Universidade Federal de Uberlândia, Uberlândia, 40 p., 2023.

GITHUB, **The top programming languages. 2022**. Disponível em: <https://octoverse.github.com/2022/top-programming-languages>. Acesso em 07/07/2023

GONZÁLEZ, A. B. P.; SILVA, B. J.; MASIP-MACÍA, Y. Transición energética en América Latina y el Caribe: diálogos inter y transdisciplinarios en tiempos de pandemia por Covid-19. **Líder: revista labor interdisciplinaria de desarrollo regional**, n. 39, p. 33-61, 2021.

GROWATT, **MIC 750 Manual de Instalação. 2020**. Disponível em: https://www.ginverter.com/upload/file/MIC_750-3300TL-X_Quick_Guide_EN202201.pdf. Acesso em 16/06/2023.

JARDIM, C. S. *et al.* O potencial dos sistemas fotovoltaicos interligados à rede elétrica em áreas urbanas: dois estudos de caso. **Proceedings of the 5th Encontro de Energia no Meio Rural**, 2004.

KUMAR, S.; SETHURAMAN, C. Instrumentation for Solar Photovoltaic System Efficiency Monitoring through Modbus Protocol. **In: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)**. IEEE, p. 232-240, 2018.

MARTINS, F. R. *et al.* **Atlas brasileiro de energia solar**. 2. ed, São José dos Campos: INPE. Disponível em: <http://doi.org/10.34024/978851700089> 80p., 2017.

MAURICIO, C. E. S. **Análise de fatores que proporcionam a redução de desempenho em sistemas fotovoltaicos**: um estudo de caso. Monografia (Graduação em Engenharia Elétrica) - Universidade Federal Rural do Semiárido, Mossoró, 43 p., 2022.

MELO, A. R. *et al.* **Desenvolvimento de um sistema supervisor aplicado em uma usina de geração de energia solar fotovoltaica.** Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal Rural do Semiárido, Mossoró, 80p., 2020.

PORTAL SOLAR S.A. **Novo medidor de energia da Elsys permite monitoramento simultâneo do consumo e geração de sistemas fotovoltaicos.** 2020. Disponível em: <https://www.portalsolar.com.br/noticias/tecnologia/equipamentos-fv/novo-medidor-de-energia-da-elsys-permite-monitoramento-simultaneo-do-consumo-e-geracao-de-sistemas-fotovoltaicos>. Acesso em: 27/05/2023.

RIGO, P. D. *et al.* Is the success of small-scale photovoltaic solar energy generation achievable in Brazil? **Journal of Cleaner Production**, v. 240, p. 118243, 2019.

RODRÍGUEZ-PÉREZ, N. *et al.* Scalability Evaluation of a Modbus TCP Control and Monitoring System for Distributed Energy Resources. **In: 2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)**. IEEE, p. 1-6, 2022.

SILVA, B.; ABREU, E. J. **Análise de viabilidade da implantação de um micro usina fotovoltaica em residência unifamiliar.** Trabalho de conclusão de curso – Faculdades Integradas de Caratinga, Caratinga, 53 p., 2017.

SÍNTESE, B. Relatório, ano base 2020. **Ministério de Minas e Energia.** 2021 Disponível em: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/balanco-energetico-nacional-2021>. Acesso em, 10 de maio .2023.

SRINATH, K. R. Python—the fastest growing programming language. **International Research Journal of Engineering and Technology**, v. 4, n. 12, p. 354-357, 2017.

ZHANG, W. *et al.* Data monitoring platform based on Modbus protocol. **In: 2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST)**. IEEE, p. 754-758, 2022

APÊNDICE

1 - Código para criação da base de dados em SQLite3

```
import sqlite3
from datetime import datetime

# Criando conexão com o banco de dados
conn = sqlite3.connect('tcc.db')

# Criando a tabela "dados_inversor"
conn.execute("""CREATE TABLE dados_inversor
              (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
              pot DOUBLE(11),
              freq DOUBLE(11),
              tensao DOUBLE(11),
              corrente DOUBLE(11),
              potenciaCC DOUBLE(11),
              tensaoCC DOUBLE(11),
              correnteCC DOUBLE(11),
              potDiaria DOUBLE(11),
              potTotal DOUBLE(11),
              energDiaria DOUBLE(11),
              energDiaCC DOUBLE(11),
              datahora TIMESTAMP DEFAULT CURRENT_TIMESTAMP);""")

# Salvando as mudanças no banco de dados e fechando a conexão
conn.commit()
conn.close()
```

2 – Código da API em Flask para acessar o banco de dados SQLite3

```
from flask import Flask, jsonify
import sqlite3

app = Flask(__name__)

@app.route('/dados')
def dados():
    conn = sqlite3.connect('tcc.db')
    cursor = conn.cursor()
    app.logger.debug('Executando consulta SQL...')
    cursor.execute('SELECT * FROM dados_inversor ORDER BY datahora DESC LIMIT 30')
    dados = cursor.fetchall()
    app.logger.debug(f'Retornado {len(dados)} registros.')
    conn.close()
    # Transforma a lista de tuplas em uma lista de dicionários
    dados_dict = []
    for row in dados:
        dado = {}
```

```

    dado['id'] = row[0]
    dado['pot'] = row[1]
    dado['freq'] = row[2]
    dado['tensao'] = row[3]
    dado['corrente'] = row[4]
    dado['potenciaCC'] = row[5]
    dado['tensaoCC'] = row[6]
    dado['correnteCC'] = row[7]
    dado['potDiaria'] = row[8]
    dado['potTotal'] = row[9]
    dado['energDiaria'] = row[10]
    dado['energDiaCC'] = row[11]
    dado['datahora'] = row[12]
    dados_dict.append(dado)
return jsonify(dados_dict)

```

```

@app.route('/dadoTelaInicial')
def dadoTelaInicial():
    conn = sqlite3.connect('tcc.db')
    cursor = conn.cursor()
    app.logger.debug('Executando consulta SQL...')
    cursor.execute('SELECT * FROM dados_inversor ORDER BY id DESC LIMIT 1')
    dados = cursor.fetchone()
    app.logger.debug(f'Retornado 1 registro.')
    conn.close()
    # Transforma a tupla em um dicionário
    dados_dict = []
    dado = {}
    dado['id'] = dados[0]
    dado['pot'] = dados[1]
    dado['freq'] = dados[2]
    dado['tensao'] = dados[3]
    dado['corrente'] = dados[4]
    dado['potenciaCC'] = dados[5]
    dado['tensaoCC'] = dados[6]
    dado['correnteCC'] = dados[7]
    dado['potDiaria'] = dados[8]
    dado['potTotal'] = dados[9]
    dado['energDiaria'] = dados[10]
    dado['energDiaCC'] = dados[11]
    dado['datahora'] = dados[12]
    dados_dict.append(dado)
    return jsonify(dados_dict)

```

```

if __name__ == '__main__':
    app.run(debug=True)

```

3 – Código para captação de dados do inversor

```
from pymodbus.client.sync import ModbusSerialClient as ModbusClient
from datetime import datetime
import time
import mysql.connector
from mysql.connector.errors import Error
from mysql.connector import Error
import sqlite3

conn = sqlite3.connect('tcc.db')

def sqlite3(vetor):
    try:
        cursor = conn.cursor()

        comando_SQL = "INSERT INTO dados_inversor (pot, freq, tensao, corrente,
potenciaCC, tensaoCC, correnteCC, potDiaria, potTotal, energDiaria, energDiaCC, datahora)
\
VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?)"

        datahora = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        dados = tuple(vetor) + (datahora,)

        cursor.execute(comando_SQL, dados)
        conn.commit()
    except sqlite3.Error as erro:
        print("Falha na comunicação: {}".format(erro))
    finally:
        if cursor:
            cursor.close()
            print("Conexão finalizada")

def run_sync_inversor(id):
    client = ModbusClient(method='rtu', port='COM7', timeout=0.5, baudrate=9600)
    client.connect()
    inverter = client.read_input_registers(2,61,unit=id)
    try:
        a = inverter.registers
    except Exception as e:
        return []
    client.close()
    return a

if __name__ == "__main__":
    while(True):
        inversor = run_sync_inversor(1)
        # Dados dos registros do inversor
        try:
            potenciaCC = inversor[0]/10
```

```

tensaoCC = inversor[1]/10
correnteCC = inversor[2]/10
pot = inversor[34]/10
freq = inversor[35]/100
tensao = inversor[36]/10
corrente = inversor[37]/10
energDiaria = inversor[52]/10
potTotal = inversor[54]/10
potDiaria = inversor[58]/10
energDiaCC = inversor[60]/10
except Exception as e:
    print('>>> erroInversor <<<<')
print('Dados do inversor:')
print(pot,freq,tensao,corrente,potenciaCC,tensaoCC,correnteCC,potDiaria,potTotal,energ
Diaria,energDiaCC)
vetor = [pot,freq, tensao, corrente, potenciaCC, tensaoCC,correnteCC, potDiaria,
potTotal, energDiaria, energDiaCC]
sqlite3(vetor)
time.sleep(1)
conn.close()

```