

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

ANA JULIA AGUIAR TAGLIASSACHI

Estudo de APIs e a Integração de Sistemas Microprocessados
para a Conexão de Inversores de Potência à Internet da
Energia

São Carlos
2024

ANA JULIA AGUIAR TAGLIASSACHI

Estudo de APIs e a Integração de Sistemas Microprocessados
para a Conexão de Inversores de Potência à Internet da
Energia

Monografia apresentada ao Curso
de Engenharia da Computação,
da Escola de Engenharia de
São Carlos da Universidade
de São Paulo, como parte dos
requisitos para obtenção do Título
de Engenheira da Computação

Orientador: Prof. Dr. Augusto
Matheus dos Santos Alonso

VERSÃO CORRIGIDA

São Carlos
2024

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

A282e Aguiar-Tagliassachi, Ana Julia
 Estudo de APIs e a integração de sistemas
microprocessados para a conexão de inversores de
potência à internet da energia / Ana Julia Aguiar-
Tagliassachi; orientadora Augusto Matheus dos Santos
Alonso. São Carlos, 2024.

Monografia (Graduação em Engenharia de
Computação) -- Escola de Engenharia de São Carlos e
Instituto de Ciências Matemáticas e de Computação da
Universidade de São Paulo, 2024.

1. Internet da Energia. 2. Inversores Inteligentes.
3. Fontes de Energias Renováveis. 4. Inversores
Eletrônicos de Potência. I. Título.

FOLHA DE APROVAÇÃO

Nome: Ana Julia Aguiar Tagliassachi

Título: “Estudo de APIs e a Integração de Sistemas Microprocessados para a Conexão de Inversores de Potência à Internet da Energia”

Trabalho de Conclusão de Curso defendido em 02 / 12 / 2024.

Comissão Julgadora:

Resultado:

Prof. Dr. Augusto Matheus dos Santos Alonso -
Orientador - SEL/EESC/USP

APROVADO

Prof. Dr. Fabio Romano Lofrano Dotto
SEL/EESC/USP

APROVADO

Prof. Adjunto Wesley Angelino de Souza
UTFPR/Campus Cornélio Procopio

APROVADO

Coordenador pela EESC do Curso Interunidades
Engenharia de Computação:

Prof. Dr. Valdir Grassi Junior - EESC/USP

*Dedico este trabalho à comunidade científica e ao valor da disseminação do conhecimento,
na esperança de abrir portas para novas oportunidades.*

AGRADECIMENTOS

Concluir o TCC e a jornada acadêmica é uma grande realização, fruto de muito esforço e dedicação. Esses cinco anos na instituição me transformaram profundamente, moldando-me na pessoa que sou hoje. Sou grata a cada desafio, cada aprendizado e cada conquista ao longo desse caminho.

Primeiramente, agradeço aos meus pais, Silvia Helena Aguiar Tagliassachi e José Mario Tagliassachi, por sempre me apoiarem e incentivarem nessa jornada. Como promotores dedicados da educação, me ensinaram a valorizar o estudo e sempre estiveram ao meu lado, para que eu nunca desistisse do desafio da graduação.

Também gostaria de agradecer ao meu namorado, João Augusto Leite, por todo apoio e companheirismo ao longo desse processo. Mesmo distante, do outro lado do Atlântico, nunca deixou de se fazer presente, sendo meu suporte nos momentos mais difíceis e celebrando comigo nos mais felizes, sempre com muito amor e carinho.

Aos amigos que tornaram todos os dias mais leves e divertidos, amizades que desejo levar para a vida inteira. Em especial, agradeço ao Rafael Dalonso Schultz e ao Daniel Martins Arrais, que estiveram ao meu lado em cada momento, dividindo risos e dificuldades ao longo desses cinco anos. Em apenas cinco anos, construímos uma amizade que carrega a profundidade de muitas vidas.

Agradeço, ainda, ao meu orientador, Prof. Dr. Augusto Matheus dos Santos Alonso, por todo o apoio e confiança no meu potencial. Ele abriu portas e me ofereceu oportunidades que transformaram não apenas minha trajetória acadêmica, mas minha vida. Em apenas um ano, impactou profundamente minha graduação, sempre me impressionando com seu empenho e dedicação aos alunos e ao seu trabalho.

Por fim, expresso minha profunda gratidão à FAPESP pelo financiamento da minha pesquisa de Iniciação Científica e pela oportunidade de intercâmbio na Itália, que ampliou minha visão de mundo e transformou minha vida de maneiras que a graduação, sozinha, não poderia proporcionar. Agradeço também à Dra. Elisabetta Tedeschi por me receber de braços abertos e por sua generosidade e carinho, que permanecem até hoje.

Sou grata pelos anos vividos na Universidade de São Paulo (USP), na Escola de Engenharia de São Carlos (EESC) e no Instituto de Ciências Matemáticas e de Computação (ICMC). Cada aprendizado e conhecimento adquirido me tornaram uma pessoa melhor, e, hoje, saio dessa instituição uma nova mulher!

RESUMO

TAGLIASSACHI, A. J. A. **Estudo de APIs e a Integração de Sistemas Microprocessados para a Conexão de Inversores de Potência à Internet da Energia.** 2024. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

As energias renováveis são um tema central no cenário mundial atual, e, nesse contexto, surge a ideia da Internet da Energia, que tem como objetivo principal integrar e criar novos consumidores e produtores de energia em uma rede inteligente, que se comunica e atende de forma eficiente às necessidades de seus usuários. Este estudo propõe, a partir da compreensão da Internet da Energia, a implementação de um protótipo que utiliza um inversor eletrônico de potência conectado a uma plataforma microprocessada, integrando-se a APIs que fornecem dados meteorológicos em tempo real, possibilitando o ajuste dinâmico do inversor. Para viabilizar o projeto, foram usados o DSP TMS320F28035PNT, disponível na plataforma eletrônica DRV830x-HC-C2-KIT da Texas Instruments, e uma plataforma microprocessada acessível, a Raspberry Pi 4. Algoritmos em Python foram desenvolvidos para conectar a Raspberry Pi à internet e coletar dados meteorológicos em tempo real, além de implementar um algoritmo inteligente para processar esses dados. O protótipo foi montado com comunicação serial UART, validada pelo envio de comandos entre dispositivos e pelo processamento de dados meteorológicos, que determinam o despacho ou armazenamento de energia pelo inversor. A abordagem mostrou flexibilidade ao permitir a substituição das plataformas utilizadas, a expansão para diferentes APIs e a personalização dos algoritmos de decisão, demonstrando o potencial da integração entre tecnologias modernas e a Internet da Energia. Dessa forma, esta pesquisa buscou simular uma rede que implementa a Internet da Energia, explorando a integração de tecnologias atuais com as iniciativas globais de expansão das energias limpas.

Palavras-chave: Internet da Energia, Inversores Inteligentes, Fontes de Energias Renováveis, Inversores Eletrônicos de Potência.

ABSTRACT

TAGLIASSACHI, A. J. A. **Study of APIs and the Integration of Microprocessed Systems for Connecting Power Inverters to the Energy Internet.** 2024. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

Renewable energies are a central theme in the current global landscape, and, in this context, the concept of the Energy Internet arises, with its main objective being to integrate and create new energy consumers and producers within an intelligent network that communicates and efficiently meets its users' needs. This study proposes, starting from an understanding of the Energy Internet, the implementation of a prototype that uses a power electronic inverter connected to a microprocessor platform, integrating with APIs that provide real-time meteorological data, enabling the dynamic adjustment of the inverter. To enable this project, the TMS320F28035PNT DSP, available in the DRV830x-HC-C2-KIT electronic platform from Texas Instruments, was used, along with an accessible microprocessor platform, the Raspberry Pi 4. Python algorithms were developed to connect the Raspberry Pi to the internet for real-time meteorological data collection, as well as to implement an intelligent algorithm for processing the data. The prototype was assembled with UART serial communication, validated by sending commands between devices and processing meteorological data to determine the dispatch or storage of energy by the inverter. The approach demonstrated flexibility by allowing the substitution of the platforms used, expansion to different APIs, and customization of decision algorithms, showcasing the potential of integrating modern technologies with the Energy Internet. Thus, this research aimed to simulate a network implementing the Energy Internet, exploring the integration of current technologies with global initiatives for expanding clean energy sources.

Keywords: Energy Internet, Smart Inverters, Renewable Energy Sources, Power Electronic Inverters.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação esquemática da Internet da Energia.	22
Figura 2 – Representação da Internet da Energia, demonstrando a conexão entre a produção de energias limpas e renováveis e a produção de energia elétrica para diversos dispositivos como carros elétricos, celulares e computadores	25
Figura 3 – Plataforma experimental a ser considerada neste projeto proposto, considerando a placa eletrônica de um IEP comercial a ser integrada com um sistema microprocessado conectado à IE	31
Figura 4 – Gráfico de barras comparativo entre as APIs e <i>Accuweather</i> para os parâmetros de temperatura, umidade e cobertura de nuvens na cidade de São Paulo	37
Figura 5 – Gráfico de barras comparativo entre as APIs e <i>Accuweather</i> para os parâmetros de temperatura, umidade e cobertura de nuvens na cidade de São Carlos	37
Figura 6 – Gráfico de barras comparativo entre as APIs e <i>Accuweather</i> para os parâmetros de temperatura, umidade e cobertura de nuvens na cidade de Roma	38
Figura 7 – Gráfico do tipo box-plot comparativo para o tempo de resposta de cada uma das APIs	41
Figura 8 – Fluxograma do processamento realizado pelo algoritmo inteligente programado	49
Figura 9 – Planejamento do fluxograma do protótipo	50
Figura 10 – Configuração do protótipo montado	51
Figura 11 – Vista detalhada da montagem do protótipo	52
Figura 12 – Onda senoidal mostrada no CCS.	53
Figura 13 – Resultado exibido no terminal da Raspberry Pi.	54

LISTA DE TABELAS

Tabela 1 – Compilado de informações sobre os diferentes tipos de comunicação serial: UART, SPI, I2C e CAN	30
Tabela 2 – Descrição detalhada sobre as características de cada uma das APIs . .	34
Tabela 3 – Comparação entre limitações dos planos gratuitos e valores dos planos profissionais das APIs	40
Tabela 4 – Resultados de 10 consultas de cada API na Raspberry Pi da previsão de tempo (em milisegundos)	41

LISTA DE ABREVIATURAS E SIGLAS

FER	Fontes de Energia Renovável
IEP	Inversor Eletrônico de Potência
DSP	Processadores de Sinal Digital
IE	Internet da Energia
IoT	Internet das Coisas
PROINFA	Programa de Incentivo às Fontes Alternativas
CC	Corrente Contínua
CA	Corrente Alternada
DSP	Digital Signal Processor
LAFAPE	Laboratório de Fontes Alternativas e Processamento de Energia
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
CAN	Controller Area Network
UART	Universal Asynchronous Receiver-Transmitter
TI	Texas Instruments
PV	Fotovoltaico
DNI	Irradiação Direta Normal
DHI	Irradiação Difusa Horizontal
GHI	Irradiação Global Horizontal
CSS	Code Composer Studio
API	Interface de Programação de Aplicação
INMET	Instituto Nacional de Meteorologia do Brasil
GFS	Sistema Global de Previsão

SUMÁRIO

1	INTRODUÇÃO	21
2	OBJETIVOS	23
2.1	Objetivo Geral	23
2.2	Objetivos Específicos	23
3	CONTEÚDO TEÓRICO	25
3.1	Internet da Energia	25
3.2	Inversores Eletrônicos de Potência	27
3.3	Plataformas Microprocessadas	28
3.4	Técnicas de Comunicação Serial	29
4	ESCOLHA E ESTUDO DAS APIS	33
4.1	Precisão e Cobertura	35
4.2	Custo	39
4.3	Tempo de Resposta	40
4.4	Usabilidade	42
4.5	Conclusão	43
4.6	Observação sobre APIs de dados solares	43
5	IMPLEMENTAÇÃO DO PROTÓTIPO	45
5.1	Desenvolvimento do protótipo no IEP	45
5.2	Implementação do protótipo na plataforma microprocessada	45
5.2.1	Desenvolvimento do Algoritmo de Decisão	46
5.3	Integração entre plataformas	48
6	RESULTADOS	51
6.1	Procedimento Operacional do Protótipo	52
6.1.1	Fluxo de Entrada do Usuário	52
6.1.2	Processamento de Dados Obtidos via API	53
7	CONCLUSÕES	55
	REFERÊNCIAS	57

1 INTRODUÇÃO

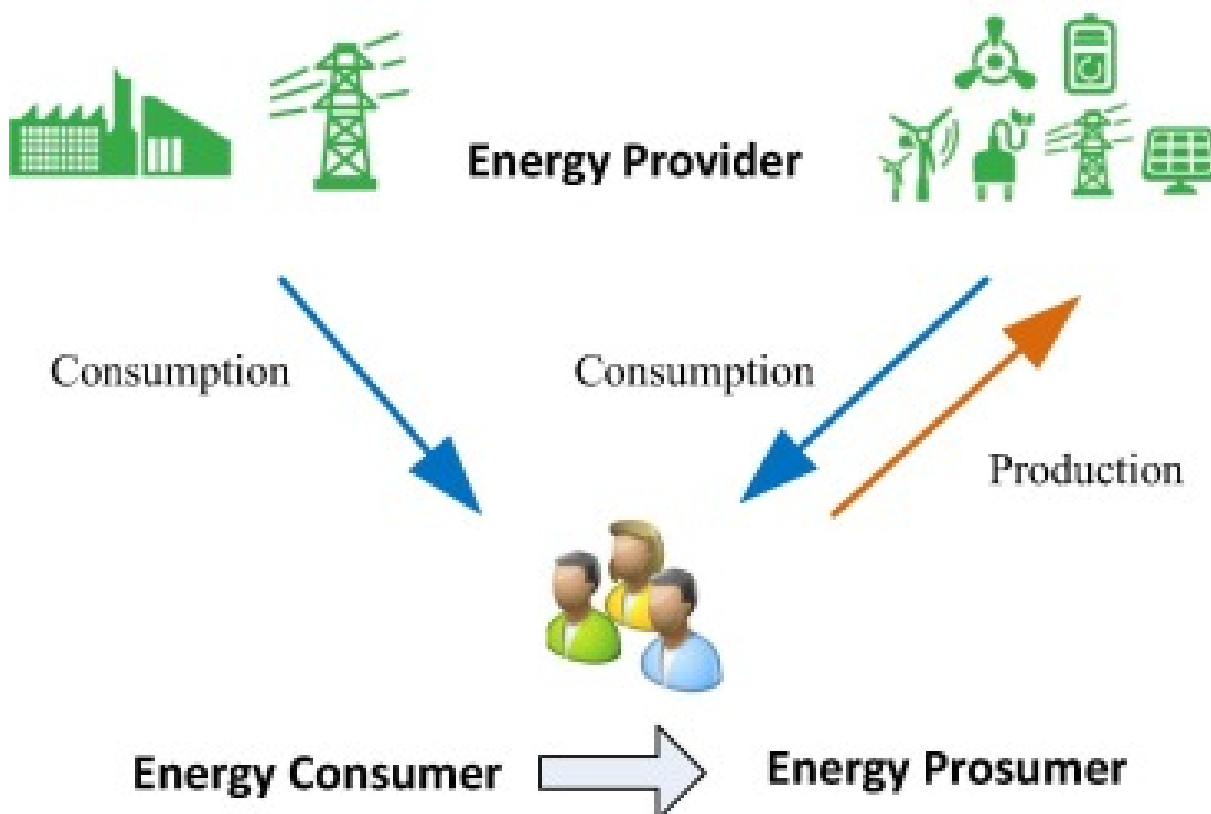
Ao longo dos últimos anos, tem-se observado uma crescente tendência global para a adoção de práticas que visam promover um futuro energético sustentável, impulsionada pelas diversas vantagens socioeconômicas geradas por essas iniciativas (ARAÚJO et al., 2022). Em âmbito mundial, fontes de energia renováveis (FER) estão sendo amplamente utilizadas para substituir as fontes convencionais não renováveis, que dependem de recursos naturais esgotáveis (IPAKCHI, 2009; NREL, 2022). Para que essa transição ocorra de maneira eficiente, também se fazem necessários investimentos em tecnologias eletrônicas capazes de viabilizar a distribuição da energia gerada por essas FERs por meio da rede elétrica (ANGADI et al., 2021; TANG; YANG; BLAABJERG, 2022).

Os inversores eletrônicos de potência (IEPs) desempenham um papel crucial nesse processo de transição. Como a geração de energia em sistemas fotovoltaicos ocorre em corrente contínua, o uso de inversores é fundamental para converter essa energia e integrá-la à rede elétrica, que opera com corrente alternada (ANGADI et al., 2021). Além de atuar na conversão de energia, os IEPs também oferecem diversas outras funcionalidades, como a regulação de tensão, a melhoria da qualidade da energia na rede e o aumento da eficiência energética (ARBAB-ZAVAR et al., 2019; MARAFÃO, F. B. et al., 2015). Devido à sua relevância, esses dispositivos são conhecidos como "inversores inteligentes", embora seu potencial seja limitado pelas restrições computacionais dos sistemas embarcados que os gerenciam (MIRAFZAL; ADIB, 2020).

Os algoritmos responsáveis pela conversão de energia são implementados em sistemas embarcados, como os processadores de sinal digital (DSPs), amplamente utilizados devido ao seu baixo custo e facilidade de produção (SI et al., 2021). Entretanto, a limitada capacidade de processamento desses dispositivos evidencia a necessidade de sua integração com sistemas microprocessados independentes, que ofereçam maior conectividade e poder de processamento, permitindo o controle remoto e a interação com outros dispositivos (IEEE. . . , 2018; SERBAN et al., 2020).

O conceito de Internet da Energia (IE) surge como uma evolução natural nesse contexto, assemelhando-se à descentralização ocorrida na indústria da computação entre as décadas de 1980 e 1990 (HUANG et al., 2011). A tendência atual é descentralizar as redes energéticas, promovendo a interoperabilidade entre dispositivos e permitindo a coleta e análise de dados em tempo real para aprimorar as operações da rede elétrica (CHENG et al., 2019; JOSEPH; BALACHANDRA, 2020). Nesse sentido, os estudos sobre a IE têm motivado investigações sobre a incorporação de plataformas computacionais mais flexíveis e poderosas nos IEPs, fortalecendo o papel desses dispositivos na rede inteligente. A Figura 1 ilustra o conceito, em que a descentralização da rede elétrica permite que consumidores de energia também atuem como "prosumidores"—produtores e consumidores de energia simultaneamente.

Figura 1 – Representação esquemática da Internet da Energia.



Fonte: (ZHOU; YANG; SHAO, 2016a)

A integração entre a IE e os IEPs possibilita uma otimização na gestão da rede elétrica. Um exemplo dessa integração é o uso de dados meteorológicos e informações sobre a demanda energética para ajustar a produção de energia em tempo real (HUANG et al., 2011; XIA et al., 2020). Dessa forma, o estudo da integração entre a plataforma eletrônica de um IEP e um sistema microprocessado com acesso à IE ganha relevância no campo das redes elétricas inteligentes. Este projeto, alinhado a essa perspectiva, visa aprofundar os estudos sobre IEPs e IE, com foco na integração de plataformas computacionais para viabilizar a implementação desse sistema.

Recentemente, com a crescente disponibilidade de APIs que oferecem dados climáticos e informações em tempo real, abre-se um novo campo de estudo sobre como essas ferramentas podem ser utilizadas para melhorar o desempenho dos IEPs. APIs de fontes meteorológicas globais permitem a obtenção de informações atualizadas que, quando integradas a sistemas de inversores inteligentes, oferecem a possibilidade de ajustes automáticos na operação do sistema conforme variações nas condições climáticas (WAVHALE et al., 2020). Este projeto também inclui a análise e comparação de diferentes APIs, avaliando sua precisão, cobertura geográfica e desempenho, com o objetivo de identificar as melhores opções para integrar essas tecnologias aos IEPs e otimizar o controle da rede elétrica.

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver um sistema embarcado que conecte um IEP a uma plataforma microprocessada, com o objetivo de monitorar e controlar a transformação energética de fontes renováveis alternativas. O sistema visa otimizar a eficiência operacional e permitir a análise de dados em tempo real, utilizando APIs conectadas à internet que fornecem informações meteorológicas e climáticas em tempo real.

O estudo também tem como objetivo ampliar o conhecimento científico relacionado à produção de energias limpas e sustentáveis, enquanto aprofunda a compreensão sobre eletrônica e programação em plataformas inteligentes. Além disso, os resultados práticos serão comparados com os resultados teóricos, proporcionando uma avaliação mais completa do desempenho e da eficácia do sistema desenvolvido.

2.2 Objetivos Específicos

Este projeto tem como objetivo investigar as tecnologias atuais relacionadas à IE, explorando suas possibilidades e vantagens no contexto contemporâneo. O principal foco é a integração entre plataformas, visando simular um protótipo funcional de IE. Além disso, busca-se estudar as plataformas eletrônicas de IEP, analisando suas funcionalidades e limitações, bem como estratégias para superá-las ou otimizá-las por meio do uso de plataformas microprocessadas.

A integração das plataformas será realizada por meio de uma rede que viabilizará a comunicação serial entre os dispositivos. Para isso, os protocolos de comunicação serial existentes serão analisados detalhadamente. O protótipo resultante será submetido a testes para avaliar seu desempenho e a eficácia do sistema integrado.

Também serão investigadas APIs online que fornecem informações meteorológicas globais e dados solares, com o objetivo de maximizar o potencial do projeto. Entre as opções disponíveis, algumas APIs serão selecionadas e comparadas com base em suas principais características, buscando identificar a mais adequada para o desenvolvimento de um sistema inteligente e integrado que conecte os usuários da IE.

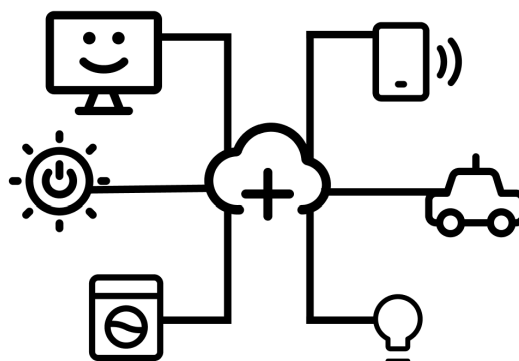
3 CONTEÚDO TEÓRICO

3.1 Internet da Energia

É inegável a movimentação global em direção a energias limpas e sustentáveis, impulsionada pela inovação tecnológica, com o objetivo de substituir as fontes de energia não renováveis e poluentes (GARCIA et al., 2014; HANSEN; SATO; RUEDY, 2012; SEDDON et al., 2016). A queima de combustíveis fósseis, como carvão e petróleo, tem causado grandes impactos ambientais, incluindo a poluição e o aquecimento global, além de ser uma fonte de energia finita (BACH, 1981; WILLIAMS, 1993). No contexto atual, a eletrificação extensiva baseada em energias limpas surge como uma das soluções mais promissoras para enfrentar esses desafios, sendo especialmente relevante no cenário de crise energética e ambiental (HUSSAIN et al., 2020). Dentro desse panorama, surge o conceito da IE, que visa integrar dispositivos, sensores e infraestrutura energética em uma rede interconectada e orientada por dados.

A IE, muitas vezes comparada à IoT, se diferencia pela sua aplicação no setor energético, permitindo uma produção descentralizada e distribuída de energia (HIMEUR et al., 2024). A comunicação ponto a ponto entre dispositivos gera uma malha interconectada de energia, com consumidores que também podem ser produtores, os chamados "prosumidores" (ZHOU; YANG; SHAO, 2016b). Com o uso de tecnologias digitais como blockchain, big data e mineração de dados, a IE possibilita que pequenos sistemas de geração de energia, como painéis solares e turbinas eólicas, sejam integrados à rede elétrica, transformando a maneira como a energia é gerada, consumida e comercializada. Uma ilustração da IE pode ser vista na Figura 2.

Figura 2 – Representação da Internet da Energia, demonstrando a conexão entre a produção de energias limpas e renováveis e a produção de energia elétrica para diversos dispositivos como carros elétricos, celulares e computadores



Fonte: Elaborado pela autora (2024)

Rifkin (2011) define o conceito de IE com base em quatro características principais: as fontes de energia devem ser predominantemente renováveis; a IE deve possibilitar a integração de um grande número de sistemas de energia renovável distribuída e dispositivos de armazenamento; também deve permitir o fluxo bidirecional de energia e comunicação; e deve incentivar a eletrificação dos sistemas de transporte (ZHIHONG et al., 2018).

A IE pode ser vista como uma transformação integradora no setor energético, capaz de viabilizar a transição da produção de energia em larga escala e centralizada para uma proliferação de pequenos sistemas de geração distribuída (JOSEPH; BALACHANDRA, 2020). Destaca-se como uma tecnologia emergente que cria uma plataforma onde os prosumidores podem produzir, utilizar e até comercializar os excedentes de eletricidade gerados em suas próprias propriedades (ZHOU; YANG; SHAO, 2016b).

Uma proposta para o futuro da produção de energia envolve a criação de uma rede elétrica descentralizada, onde cada indivíduo tenha dispositivos geradores de energia, como painéis solares e turbinas eólicas, juntamente com mecanismos de armazenamento dessa energia. Para isso, toda a infraestrutura energética precisa ser repensada e modificada de um modelo centralizado para um modelo distribuído (CHENG et al., 2019).

O Brasil, por sua vez, apresenta um cenário propício para a expansão de energias renováveis. Com uma matriz energética predominantemente renovável, o país se destaca no uso de energia hidrelétrica, biomassa, eólica e, mais recentemente, solar (MINISTÉRIO DE MINAS E ENERGIA, 2022). A expansão da geração distribuída, especialmente com o incentivo de políticas públicas como o PROINFA e o modelo de compensação de energia, tem acelerado o crescimento da energia solar fotovoltaica no país (LIMA et al., 2020).

Esse contexto global e brasileiro evidencia a relevância da IE como uma perspectiva para o futuro energético. A possibilidade de transformar o setor de energia por meio da interconectividade entre dispositivos e a utilização de fontes renováveis representa uma revolução no modo de produção e distribuição de eletricidade. A integração de IEPs em plataformas microprocessadas conectadas à internet, como proposta neste projeto, permitirá a criação de um sistema inteligente e autônomo que pode monitorar e controlar o consumo e a produção de energia de maneira eficiente (CHO; YAMAZAKI; HAHN, 2009).

Com essa infraestrutura tecnológica, a gestão da energia será otimizada, permitindo que consumidores acompanhem, em tempo real, a geração de energia local, o consumo de eletricidade, e até o preço da energia. No futuro, prevê-se que sistemas inteligentes, com o auxílio de algoritmos de inteligência artificial, assumam o controle autônomo da produção e do consumo de energia, melhorando a qualidade do fornecimento e tornando o sistema mais eficiente e acessível para todos.

Em suma, a implementação da IE representa uma solução inovadora que não apenas aborda as questões ambientais urgentes, mas também otimiza o uso e a gestão de energia de forma mais inteligente e distribuída. Isso resultará em uma rede energética mais sustentável, equitativa e eficiente, alinhada às necessidades tecnológicas e ambientais do

futuro.

3.2 Inversores Eletrônicos de Potência

A transição global para fontes de energia limpas e renováveis, como solar e eólica, avança de maneira expressiva, conforme discutido anteriormente. No entanto, um dos principais desafios técnicos é que a energia solar gera eletricidade em corrente contínua (CC), enquanto a energia eólica e as redes elétricas, assim como a maioria dos dispositivos elétricos, operam em corrente alternada (CA) (ANGADI et al., 2021). Nesse contexto, os IEPs desempenham um papel fundamental ao realizar a conversão entre CC e CA, garantindo uma transferência eficiente de energia entre diferentes fontes renováveis e a rede elétrica (HOSSEINZADEH et al., 2021). Além disso, em sistemas de armazenamento, como baterias que armazenam energia em CC, a conversão para CA é necessária para liberar essa energia de forma compatível com a rede elétrica ou para consumo local (STEIGERWALD, 2001).

Entretanto, o papel dos IEPs vai além da simples conversão entre CA e CC. Eles são fundamentais no controle do fluxo de energia, na detecção de falhas e na capacidade de desconexão, se necessário, assegurando uma operação confiável da rede elétrica (ARBAB-ZAVAR et al., 2019). A importância desses inversores cresce ainda mais no contexto da IE, onde dispositivos inteligentes precisam tomar decisões autônomas. Um inversor inteligente, por exemplo, pode ajustar seu funcionamento com base em medições locais e informações externas, operando como o núcleo de controle da rede, garantindo segurança, controle e eficiência (MIRAFZAL; ADIB, 2020).

Os IEPs modernos integram tecnologias avançadas, como microcontroladores, microprocessadores, sensores e algoritmos de controle. Essas estruturas permitem que os inversores se adaptem dinamicamente às condições da rede, aumentando sua eficiência e confiabilidade (BOWES; MOUNT, 1981).

Neste projeto, foi utilizada uma plataforma eletrônica comercialmente disponível para representar o IEP. A plataforma, desenvolvida pela *Texas Instruments* e denominada DRV830x-HC-C2-KIT (DRV830X-HC-C2-KIT... , 2014), consiste em um inversor trifásico controlado por DSP, modelo TMS320F28035PNT. Este DSP tem capacidade de comunicação serial, permitindo sua integração a uma plataforma microprocessada responsável pela conexão com a internet, possibilitando a implementação da IE em pequena escala para o trabalho. Além disso, o DRV830x-HC-C2-KIT foi escolhido por sua compatibilidade com aplicações de pequena escala, devido à sua configuração compacta e alta eficiência, o que facilita a realização de protótipos de maneira prática e econômica. Sua robustez e confiabilidade como plataforma comercialmente disponível asseguram consistência nos testes e replicabilidade dos resultados. Por fim, sua documentação abrangente e o suporte técnico fornecido pela *Texas Instruments* oferecem uma base sólida para o desenvolvimento e depuração do

sistema.

Visto que o papel fundamental do IEP no protótipo foi esclarecido, o próximo passo é selecionar a plataforma microprocessada mais adequada para o projeto. Além disso, é pertinente compreender sua função no processo de implementação da rede inteligente desenhada para esse projeto.

3.3 Plataformas Microprocessadas

Dado que foi definida a escolha do DSP da *Texas Instruments*, iniciaram-se os estudos para selecionar a plataforma microprocessada mais adequada ao propósito do projeto. Esta plataforma será responsável por interconectar o sistema à Internet, viabilizando a conexão essencial para a criação de um protótipo capaz de simular a IE.

A escolha recaiu sobre a Raspberry Pi 4, uma plataforma não apenas disponível no Laboratório de Fontes Alternativas e Processamento de Energia (LAFAPE), onde a pesquisa tomou lugar, mas também de fácil acesso ao público em geral. A decisão foi baseada em sua simplicidade operacional, suporte a um sistema operacional robusto, facilidade de conexão à Internet (utilizando o *Raspberry Pi Imager* (RASPBerry PI LTD., 2023)), familiaridade da pesquisadora com a plataforma, escalabilidade, vasta documentação consolidada, e suporte a uma ampla variedade de linguagens de programação.

Além disso, a Raspberry Pi é amplamente reconhecida no contexto da IoT, onde desempenha um papel crucial na interconexão de dispositivos, facilitando a comunicação entre eles e a Internet. Ela possibilita a troca de dados entre dispositivos locais e a rede global (MARAGATHAM; BALASUBRAMANIE; VIVEKANANDHAN, 2021). A plataforma também é conhecida por seu baixo consumo de energia, um fator essencial em projetos voltados à eficiência energética, além de sua capacidade de realizar processamento local de dados, reduzindo a dependência de servidores externos.

A versatilidade da Raspberry Pi em aplicações científicas e tecnológicas é notável, sendo frequentemente utilizada em monitoramento em tempo real de consumo energético e controle de sistemas embarcados, o que reforça sua adequação a projetos voltados para otimização da gestão e eficiência energética (MUDALIAR; SIVAKUMAR, 2020). Sua compatibilidade com protocolos padrão de comunicação e sua capacidade de operar em redes cabeadas ou sem fio tornam-na ideal para aplicações de integração com dispositivos como o DSP escolhido.

No contexto deste projeto, a adoção da Raspberry Pi é estratégica, pois seu principal objetivo é conectá-la à Internet para coletar dados em tempo real de fontes como serviços meteorológicos confiáveis e APIs disponíveis na internet, permitindo a integração desses dados com o DSP. Essa integração visa otimizar os processos de produção e conversão de energia, além de garantir flexibilidade para futuras expansões do sistema.

A fim de viabilizar essa conexão, é necessário desenvolver algoritmos na Raspberry Pi,

responsáveis pelo pré-processamento dos dados coletados da Internet. A proposta envolve uma abordagem computacional para processar e analisar os dados meteorológicos, transmitindo essas informações, já tratadas, ao DSP. Essa abordagem inclui o desenvolvimento de um algoritmo inteligente, que será detalhado ao longo do relatório.

Nesse contexto, a plataforma microprocessada atua como intermediária entre a fonte de dados em tempo real online e o DSP. Para isso, será necessário definir tanto o método de comunicação serial dessas informações quanto a fonte dos dados. Essas próximas etapas da pesquisa serão analisadas e discutidas nas seções subsequentes.

3.4 Técnicas de Comunicação Serial

A comunicação serial é uma parte essencial para a integração entre diferentes plataformas em projetos eletrônicos, como o descrito no contexto deste trabalho. Existem diversas técnicas de comunicação disponíveis, cada uma com características específicas que as tornam adequadas para diferentes cenários. Entre as mais comuns, destacam-se SPI, I2C, CAN e UART, que serão estudadas e analisadas ao longo deste projeto.

O SPI é um protocolo de comunicação projetado para transmitir dados rapidamente entre um dispositivo mestre e um ou mais dispositivos escravos. Ele utiliza quatro conexões principais: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCLK (Serial Clock), e SS (Slave Select). Essas linhas permitem uma comunicação full-duplex, ou seja, mestre e escravo podem enviar e receber dados ao mesmo tempo. No entanto, o SPI é limitado a curtas distâncias e não possui um protocolo padrão, exigindo configurações específicas de software para cada dispositivo conectado ao barramento, o que pode dificultar sua implementação em sistemas mais complexos (WOOTTON, 2016).

Já o I2C, projetado pela Philips Semiconductor, é amplamente usado em sistemas embarcados devido à sua simplicidade e capacidade de conectar múltiplos dispositivos em um mesmo barramento com apenas duas linhas: SDA (dados) e SCL (clock). Apesar de ser um protocolo popular em aplicações com distâncias maiores e diversos periféricos, sua taxa de transferência de dados é consideravelmente mais baixa, o que pode se tornar um empecilho em projetos que demandam maior velocidade (KALINSKY; KALINSKY, 2002).

Por outro lado, o CAN, desenvolvido pela Bosch, é um protocolo robusto e amplamente utilizado em aplicações automotivas e industriais. Ele é capaz de lidar com a comunicação em ambientes ruidosos, oferecendo alta confiabilidade e controle de erros. A característica multi-master e o sistema de priorização de mensagens tornam o CAN extremamente eficiente em redes com múltiplos dispositivos que requerem alta sincronização. No entanto, o uso do CAN pode ser excessivamente complexo e, para projetos que envolvem um número limitado de dispositivos e baixa taxa de transmissão de dados, essa complexidade pode ser um fator de desvantagem (CORRIGAN, 2002).

Diante dessas três opções, a comunicação UART surge como uma solução mais simples

e eficiente para o contexto do projeto. O UART utiliza apenas dois fios para a transmissão e recepção de dados — Tx e Rx — o que facilita sua implementação. Além disso, a simplicidade da UART, que dispensa a necessidade de sincronização de clock como no SPI ou I2C, a torna ideal para aplicações em que a simplicidade e a eficiência são prioridades (LADDHA; THAKARE, 2013). Embora o UART não seja ideal para redes complexas com muitos dispositivos, sua implementação direta e de baixo custo computacional a torna a escolha perfeita para aplicações simples, como a comunicação entre a Raspberry Pi e o DSP no projeto atual, em que os dispositivos estarão lado a lado conectados por fios.

A comunicação foi estabelecida para transmitir um valor numérico representando o pico de uma onda senoidal da Raspberry Pi para o DSP. Esses dados foram enviados em formato binário simples, ao invés de caracteres ASCII, o que demonstra a utilização de um formato direto e eficiente. A configuração de transmissão adotada foi de 9600 baud, e controle de fluxo por hardware. O valor transmitido tinha como finalidade permitir ao DSP modificar o formato da onda senoidal gerada. Essa abordagem reforça a adequação do UART para este cenário, uma vez que a comunicação baseou-se na transmissão de dados discretos e objetivos, eliminando a necessidade de protocolos mais complexos.

Dado que os dados a serem transmitidos no projeto são relativamente simples (como inteiros ou comandos binários), a escolha por uma comunicação UART reflete a necessidade de uma abordagem enxuta e eficaz. Comparado às opções mais sofisticadas, como SPI e CAN, o UART oferece uma solução menos complexa, que se alinha melhor às demandas específicas do projeto. A seguir, pode ser observada na Tabela 1 com um compilado das observações feitas a respeito de cada uma das técnicas de comunicação.

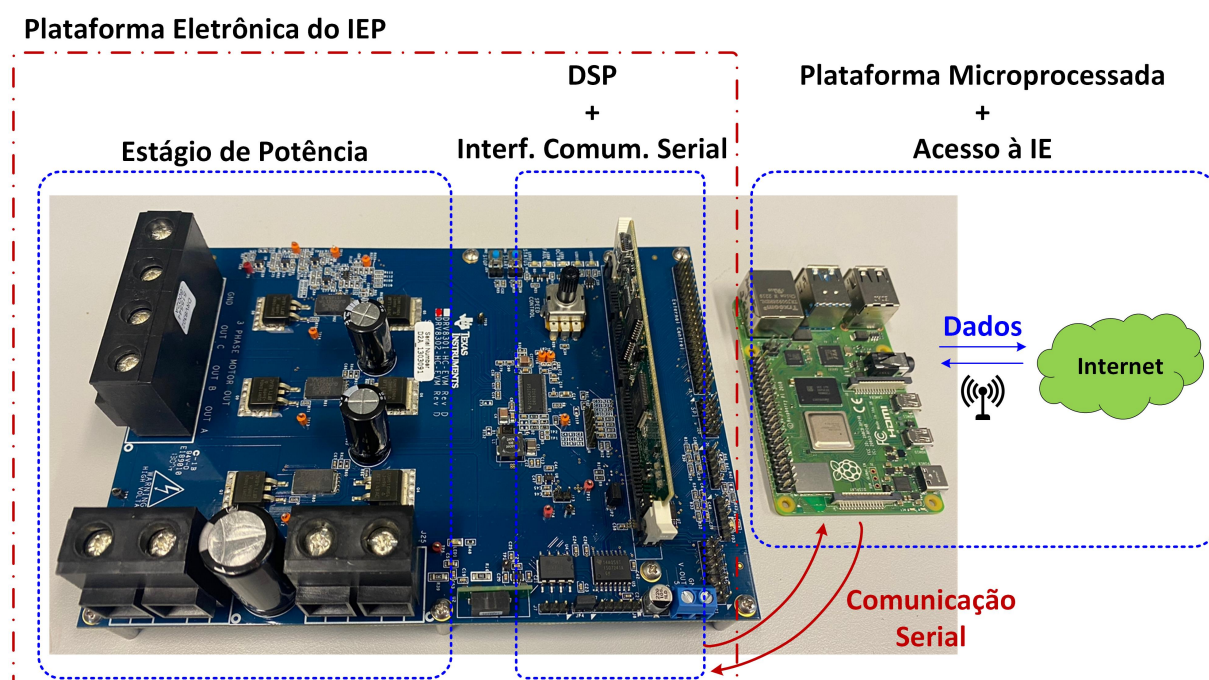
Tabela 1 – Compilado de informações sobre os diferentes tipos de comunicação serial: UART, SPI, I2C e CAN

Protocolo	UART	SPI	I2C	CAN
Complexidade	Complexidade baixa	Complexidade cresce com a complexidade do dispositivo	Fácil alinhar múltiplos dispositivos	Complexidade Moderada
Velocidade	Lento	Alta velocidade	Velocidade moderada	Alta velocidade
Número de dispositivos	No máximo dois	Vários podem ser conectados numa mesma linha de barramento	Até 127 dispositivos	Pode suportar vários na mesma rede
Número de fios	Dois	No mínimo quatro	Dois	Dois
Duplex	Full duplex	Full duplex	Half Duplex	Full duplex

A partir das comparações entre técnicas realizadas, foi possível chegar na conclusão de que a comunicação UART é a escolha mais adequada, proporcionando simplicidade, baixo custo de implementação e a eficiência necessária para a integração entre as plataformas utilizadas no desenvolvimento.

O objetivo pretendido ao construir o protótipo com os dispositivos mencionados pode ser visualizado na Figura 3. Nesta representação, observa-se a conexão do IEP à plataforma microprocessada independente através de comunicação serial. Propõe-se a comunicação do tipo UART entre o DSP e a Raspberry Pi, onde esta última atua como intermediária na interação com a internet e processamento analítico dos dados.

Figura 3 – Plataforma experimental a ser considerada neste projeto proposto, considerando a placa eletrônica de um IEP comercial a ser integrada com um sistema microprocessado conectado à IE



Fonte: Elaborado pelo Prof. Dr. Augusto Matheus dos Santos Alonso

Após definir as plataformas que serão conectadas à internet e a forma como essa conexão será estabelecida, o passo seguinte consiste na escolha da API mais adequada para o projeto. Nesse contexto, foi realizada uma análise detalhada e aprofundada das APIs disponíveis online, avaliando seus prós e contras, com o objetivo de tomar uma decisão informada e precisa, considerando todos os fatores necessários para o desenvolvimento do protótipo final que concretiza o projeto.

4 ESCOLHA E ESTUDO DAS APIS

As interfaces de programação de aplicação (APIs) emergem como elementos essenciais no cenário tecnológico atual, atuando como pontes que conectam diferentes sistemas de software. Elas possibilitam a comunicação e a troca de informações de maneira padronizada entre diversas aplicações (SCHEUERMANN, 2019).

No contexto desta pesquisa, as APIs desempenham um papel fundamental como facilitadoras do processo de coleta de dados meteorológicos, utilizados para o treinamento de um algoritmo inteligente. Sua implementação simplifica consideravelmente a obtenção dessas informações, uma vez que permitem a integração fácil e eficiente de chamadas em diversas linguagens de programação, possibilitando a conexão com todo o sistema de maneira prática.

Além do desenvolvimento do protótipo para a implementação de um modelo de IE, um dos principais objetivos deste projeto é a análise comparativa de APIs que fornecem informações meteorológicas em tempo real, com abrangência global. É importante destacar que, além dessas informações, os dados solares em tempo real são cruciais para o projeto, já que fatores como a radiação solar podem influenciar diretamente o desempenho de sistemas fotovoltaicos. Portanto, o monitoramento dessas condições é essencial para otimizar a operação desses dispositivos.

Inicialmente, foram investigadas e analisadas as principais APIs disponíveis no mercado. Com base em revisões de usuários, estudos de pesquisadores e nos objetivos específicos do projeto, selecionaram-se cinco APIs meteorológicas e uma API especializada em dados solares. A análise das APIs meteorológicas será baseada em critérios como precisão, cobertura, custo, tempo de resposta e usabilidade, fatores essenciais para determinar a API mais adequada à integração no sistema desenvolvido. É importante destacar que a API de dados solares não será avaliada sob os mesmos critérios, pois os dados oferecidos por ela diferem significativamente dos das APIs meteorológicas, tornando qualquer comparação direta imprópria.

Antes de iniciar a comparação, é fundamental apresentar brevemente cada uma das APIs selecionadas: OpenWeather (OpenWeatherMap, 2024), Weatherapi.com (Weatherapi, 2024), Tomorrow.io (Tomorrow.io, 2024), Open-Meteo (Open-Meteo, 2024), PirateWeather (Alexander Rey, 2024), e Solcast (Solcast, 2024). A Tabela 2 resume as principais características de cada uma delas. Os limites de chamadas indicados na tabela correspondem às versões gratuitas das APIs.

A API *OpenWeather* é amplamente utilizada em serviços web e aplicativos móveis, fornecendo dados meteorológicos baseados em estações de radar, aeroportos e outras fontes oficiais. Musah (2022), por exemplo, comparou os dados fornecidos por essa API com os dados do INMET e descobriu que as leituras de temperatura são bastante similares, embora os dados de umidade não apresentem a mesma precisão. Além disso, a API também

Tabela 2 – Descrição detalhada sobre as características de cada uma das APIs

Nome	Características Importantes	Descrição adicional	Limite de chamadas
OpenWeather	A versão gratuita inclui o clima atual com temperatura, cobertura de nuvens, umidade, velocidade do vento e muitas outras informações.	Há uma versão paga que inclui radiação solar; documentação extensiva está disponível.	1000 chamadas/dia para os dados meteorológicos em tempo real gratuitos; 0,11 euros por chamada para a funcionalidade de radiação solar.
Weatherapi.com	Também inclui informações sobre o clima em <i>real-time</i> gratuitamente.	Documentação extensiva.	1000000 chamadas/dia.
Tomorrow.io	Dados meteorológicos em tempo real gratuitos incluindo previsão de precipitação, intensidade de chuva e neve.	Clima baseado em satélite; documentação abrangente; inclui dados solares.	500 chamadas/dia.
Open-meteo	API de previsão do tempo, sem dados em tempo real.	Não é necessário chave de API, pronto para uso; fácil de usar; código aberto.	10000 chamadas/dia.
Pirate Weather	Oferece informações de meteorologia em tempo real e dados históricos.	Segue o estilo da descontinuada API <i>Dark Sky</i> da Apple e é <i>open source</i> , tem como principal objetivo manter o legado da API; é código aberto.	10000 chamadas/mês.
Solcast	Fornecer informações de previsão, irradiação solar, energia fotovoltaica e dados meteorológicos em tempo real e históricos.	Apresenta uma API de graça exclusiva para pesquisadores contendo as informações solares.	600 chamadas/minuto.

utiliza aprendizado de máquina para melhorar seus modelos de previsão, tornando-se ainda mais complexa e robusta.

A *WeatherAPI* foi desenvolvida para o serviço Energia Consulting Oy, uma subsidiária do Energia Group Oy, com o objetivo de fornecer informações meteorológicas com base em código postal, período de tempo e resolução especificada. Desenvolvida em C# com .NET Core, essa API se destaca pelo uso do banco de dados InfluxDB, ideal para armazenar

séries temporais meteorológicas devido à sua capacidade de lidar com grandes volumes de dados em tempo real com alta eficiência. O InfluxDB é otimizado para consultas rápidas e inserções de dados de séries temporais, como as informações meteorológicas que se atualizam constantemente, o que facilita a resposta rápida às solicitações dos consumidores autorizados (PAAPPANEN, 2017).

A *Tomorrow.io* é uma plataforma que se destaca por fornecer dados meteorológicos utilizando tecnologias de ponta, como radar e inteligência artificial. Sua API permite a integração de dados meteorológicos em diversos projetos. No estudo de Velosa (2023), foi utilizada para obter dados de vento com resolução horária, que foram aplicados na simulação de produção eólica na plataforma PROCSIM. Além disso, no trabalho de pesquisa de Yanqui e Merchan (2023), a *Tomorrow.io* foi integrada a um sistema de irrigação inteligente, onde seus dados meteorológicos alimentaram um modelo de previsão para otimizar o controle do sistema no trabalho.

A API *Open-Meteo* é um serviço gratuito que hospeda dados meteorológicos e permite o envio e download de séries temporais meteorológicas e hidrometeorológicas. Com uma interface acessível e um sistema robusto, os dados são disponibilizados via página web e JSON, o que facilita a integração com serviços externos. A simplicidade do serviço e a ausência da necessidade de chaves de API o tornam popular entre desenvolvedores (KOZANIS et al., 2013).

A *Pirate Weather API* oferece dados de previsão meteorológica com base em fontes como o GFS, entre outras. No projeto de pesquisa de Valente (2022), essa API foi utilizada para alimentar um painel Grafana com previsões diárias de até quatro dias e previsões horárias de até 48 horas, abrangendo parâmetros como temperatura, umidade, vento e precipitação. Sua capacidade de fornecer dados precisos a curto prazo a torna uma ferramenta versátil para diversos tipos de análise meteorológica.

Solcast se destaca por fornecer dados solares históricos e previsões de irradiação solar globalmente. Utilizando imagens de satélites meteorológicos e modelos numéricos, suas previsões são atualizadas a cada 10-15 minutos. Solcast é especialmente relevante para pesquisadores, oferecendo uma quantidade substancial de dados gratuitos, acessíveis através de sua API e website (BRIGHT, 2019).

É importante destacar que, a partir deste ponto, as comparações serão focadas principalmente nas APIs de dados meteorológicos. Ao final, será feita uma observação sobre a API Solcast, que, por não fornecer dados meteorológicos como as demais, não pode ser comparada de forma equivalente.

4.1 Precisão e Cobertura

Após a introdução de cada uma das APIs, com base na exploração de artigos já existentes, iniciou-se uma análise mais detalhada com foco nas necessidades específicas

do projeto em desenvolvimento.

Primeiramente, foi realizada uma análise da precisão e cobertura de cada API, capturando dados meteorológicos com previsão de 3 dias para três cidades de interesse no projeto: São Paulo, São Carlos e Roma. Essas localidades foram escolhidas para representar diferentes perspectivas de desempenho das APIs, considerando o contraste entre o sul e o norte globais, além de fornecer uma comparação entre o interior e grandes centros urbanos no Brasil.

Em seguida, os resultados obtidos foram comparados com uma fonte meteorológica de alta confiabilidade. Esta abordagem permite verificar a acurácia das APIs em relação a sites meteorológicos bem estabelecidos e também comparar sua cobertura em diferentes regiões do planeta, observando se há variações de desempenho entre os hemisférios norte e sul.

Apesar da escolha da *Accuweather* (ACCUWEATHER, 2024) como fonte meteorológica nesta pesquisa, considerando sua precisão e confiabilidade para previsões de até 10 dias (HANSEN, 2007), os resultados obtidos indicaram que sua acurácia pode variar dependendo da localização analisada. Isso é particularmente relevante ao considerar regiões como Brasil e Itália, cujas condições climáticas podem apresentar desafios específicos para modelos globais de previsão (FRONDUTI, 2020). Nesse sentido, uma abordagem alternativa poderia envolver a utilização de estações meteorológicas locais fixas, que, por estarem mais ajustadas às características regionais, poderiam oferecer previsões mais acuradas para os parâmetros de interesse, como temperatura, umidade e cobertura de nuvens.

Essa limitação da *Accuweather* abre espaço para estudos futuros que aprofundem a análise de diferentes fontes de dados meteorológicos e suas respectivas acurácias. Investigar quais APIs ou estações locais se mostram mais adequadas para regiões específicas, como o Brasil e a Itália, poderia contribuir significativamente para o avanço de pesquisas e aplicações em meteorologia, energia renovável e outras áreas que dependem de previsões climáticas precisas.

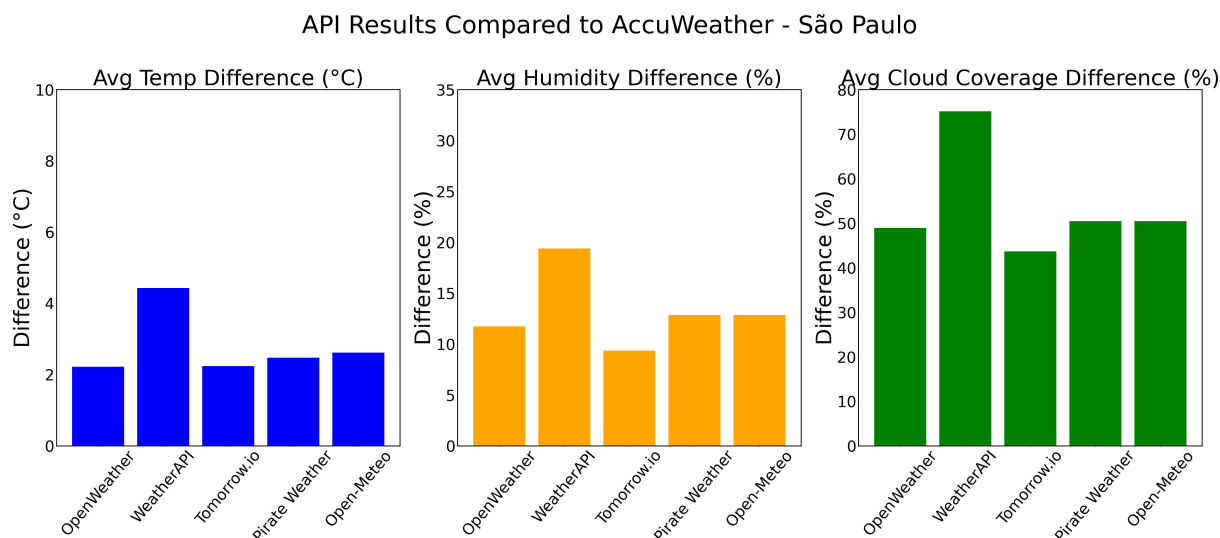
Dessa forma, os dados foram capturados para cada uma das três cidades e comparados em termos de percentual de diferença média entre os resultados obtidos pelas APIs em relação aos da *Accuweather*. As representações gráficas dos resultados para São Paulo, São Carlos e Roma são apresentadas nas Figuras 4, 5 e 6, respectivamente.

A partir dos gráficos gerados e dos resultados numéricos obtidos, foi possível tirar conclusões a respeito do desempenho das APIs nas questões de cobertura e acurácia dos parâmetros.

Em São Paulo, a API com menor diferença na temperatura é a OpenWeather, com um desvio de 2,21 °C, seguida da Tomorrow.io, com 2,24 °C. Em relação à umidade, a Tomorrow.io também apresenta o menor desvio, de 9,36%, seguida pela OpenWeather, com 11,73%. Para a cobertura de nuvens, a Tomorrow.io lidera com um desvio de 43,68%, seguida de perto pela OpenWeather, com 48,93%.

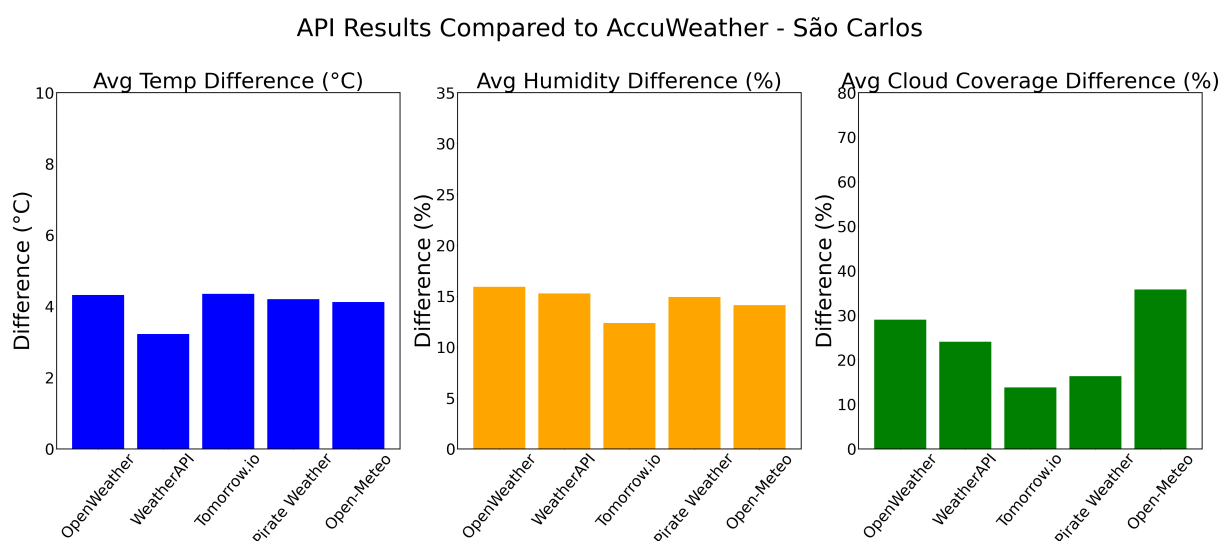
Em São Carlos, a WeatherAPI tem o menor desvio de temperatura (3,22 °C), seguida

Figura 4 – Gráfico de barras comparativo entre as APIs e *Accuweather* para os parâmetros de temperatura, umidade e cobertura de nuvens na cidade de São Paulo



Fonte: Elaborado pela autora (2024)

Figura 5 – Gráfico de barras comparativo entre as APIs e *Accuweather* para os parâmetros de temperatura, umidade e cobertura de nuvens na cidade de São Carlos



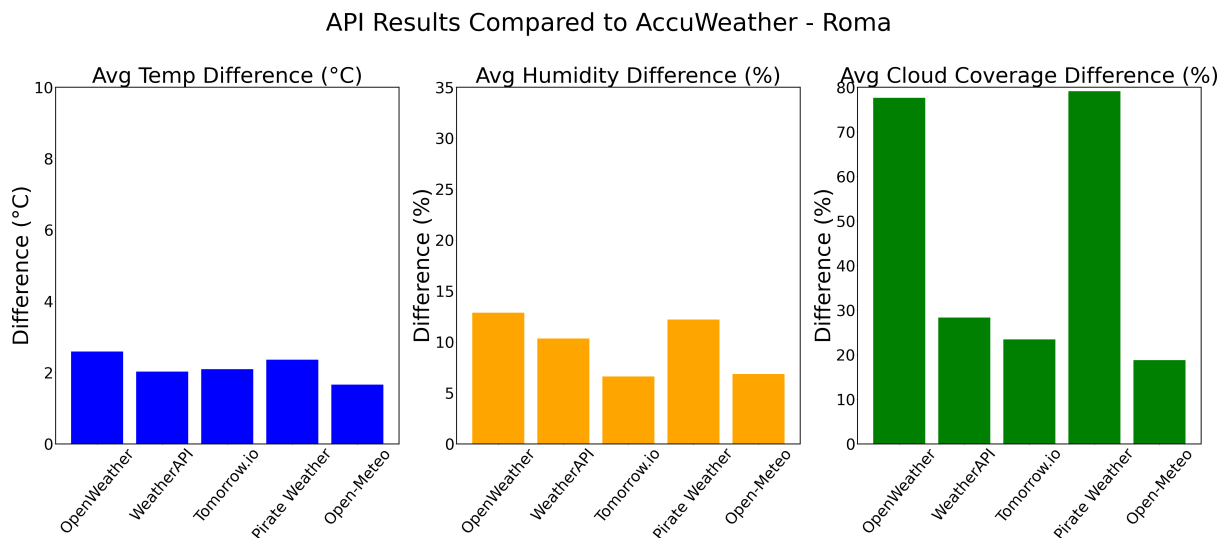
Fonte: Elaborado pela autora (2024)

da Open-Meteo (4.12 °C). Quanto à umidade, a Tomorrow.io novamente se destaca, com um desvio de 12,38%. Para a cobertura de nuvens, a Tomorrow.io possui a vantagem, apresentando o menor desvio de 13,79%.

Em Roma, a Open-Meteo tem a menor diferença de temperatura (1.66 °C), seguida pela WeatherAPI (2.02 °C). A Tomorrow.io se destaca na umidade, com um desvio de 6,62%, enquanto a Open-Meteo lidera na cobertura de nuvens, com 18,8%, seguida de perto pela Tomorrow.io, com 23,43%.

De maneira geral, a Tomorrow.io apresenta boa acurácia em várias cidades, com

Figura 6 – Gráfico de barras comparativo entre as APIs e *Accuweather* para os parâmetros de temperatura, umidade e cobertura de nuvens na cidade de Roma



Fonte: Elaborado pela autora (2024)

destaque para umidade e cobertura de nuvens, além de bons resultados em temperatura em São Paulo e Roma. A OpenWeather também se sobressai em São Paulo, sendo relativamente precisa em temperatura e umidade. A Open-Meteo mostra a melhor acurácia em Roma, especialmente para temperatura e cobertura de nuvens.

Ao comparar a cobertura entre a Itália e o Brasil (São Paulo e São Carlos), observamos que Roma apresenta desvios menores em temperatura e umidade para várias APIs, com a Open-Meteo mostrando excelente desempenho na cidade italiana, com apenas 1,66 °C de desvio em temperatura e 18,8% em cobertura de nuvens. Em contraste, em São Carlos, as diferenças de temperatura são maiores para todas as APIs, com uma média de 4.0 °C. Isso sugere que a cobertura das APIs é mais precisa na Itália, especialmente em relação a parâmetros como temperatura e cobertura de nuvens. São Carlos apresenta maiores desvios em quase todos os parâmetros, com a temperatura média divergindo mais em comparação a São Paulo e Roma, indicando que a cobertura das APIs é menos precisa em cidades menores ou menos metropolitanas. Em São Paulo, os desvios são intermediários; a precisão na cobertura de nuvens é menor (especialmente para a WeatherAPI, com desvio de 75%), mas a temperatura e a umidade são razoavelmente precisas.

A precisão das previsões parece ser mais robusta em regiões metropolitanas maiores e em países desenvolvidos, como a Itália (Roma). São Paulo, sendo uma cidade grande, tem melhor cobertura do que São Carlos, embora apresente maiores desvios em alguns parâmetros. Essa diferença pode estar relacionada à infraestrutura de coleta de dados e à qualidade dos modelos meteorológicos usados pelas APIs em diferentes regiões.

Ao comparar os três parâmetros (temperatura, umidade e cobertura de nuvens), a temperatura tende a ser o parâmetro mais preciso entre as APIs, com desvios relativamente

menores em todas as cidades. Isso indica que os modelos meteorológicos conseguem prever temperatura com maior confiabilidade, possivelmente devido à grande quantidade de dados históricos disponíveis e à natureza mais direta da previsão de temperatura. A umidade apresenta mais variação, com algumas APIs, como a Tomorrow.io, tendo bom desempenho em várias cidades, enquanto outras, como a WeatherAPI, mostram maiores desvios. A cobertura de nuvens, por sua vez, é o parâmetro mais desafiador de prever com precisão, como observado nos desvios mais altos, especialmente em São Paulo e Roma. Em Roma, por exemplo, as diferenças na cobertura de nuvens chegaram a quase 80% para a Pirate Weather, e em São Paulo, todas as APIs tiveram desvios superiores a 40%.

Alguns possíveis motivos para essas variações incluem o fato de que a previsão de temperatura pode ser mais confiável devido à maior disponibilidade de sensores e estações meteorológicas que captam dados de temperatura com frequência. A temperatura também é um fenômeno mais contínuo e menos variável em curtos períodos, o que facilita a modelagem. A umidade, no entanto, é mais sensível a condições microclimáticas e depende de fatores como vegetação e presença de corpos d'água, o que pode dificultar previsões precisas, especialmente em áreas urbanas. A cobertura de nuvens é afetada por fatores mais complexos, como ventos em diferentes altitudes, frentes atmosféricas e fenômenos locais, tornando a previsão desse parâmetro mais desafiadora e explicando os desvios mais altos observados em todas as cidades.

Concluindo, a Tomorrow.io se destaca como a API com maior precisão geral, especialmente para umidade e cobertura de nuvens. Roma possui a cobertura mais precisa, sugerindo que as APIs podem ter melhor desempenho em áreas metropolitanas europeias. A temperatura é o parâmetro mais bem previsto, enquanto a cobertura de nuvens é o mais desafiador, provavelmente devido à complexidade dos fatores que influenciam as nuvens.

4.2 Custo

Para comparar os custos das APIs e identificar a opção mais vantajosa, as informações mais relevantes foram organizadas na Tabela 3. A coluna de comparação inclui planos profissionais, uma vez que cada API apresenta os custos de seus planos pagos de forma distinta. Todos os valores foram padronizados e convertidos para dólares com a taxa de câmbio atual, visando manter a consistência dos dados.

Vale ressaltar que as APIs pagas geralmente oferecem diferentes planos com preços variados, mais acessíveis do que os apresentados na tabela. É importante mencionar que duas das APIs analisadas não possuem valores claramente divulgados: Pirate Weather, por ser gratuita, embora com recursos limitados; e Tomorrow.io, cuja oferta é voltada para soluções empresariais e exige contato direto com a empresa para orçamentos personalizados, o que ficou fora do escopo desta análise.

Observa-se que os planos gratuitos de OpenWeather e Weatherapi oferecem um número

Tabela 3 – Comparação entre limitações dos planos gratuitos e valores dos planos profissionais das APIs

Nome	Requisições/Mês (Plano Gratuito)	Valor da API Profissional
OpenWeather	1.000.000	≈ 480.30 dólares/mês (1 bilhão de chamadas/mês)
Weatherapi	1.000.000	35 dólares/mês (10 milhões de chamadas/mês)
Tomorrow.io	500 requisições/dia (até 25 por hora, 3 por segundo)	Não informado publicamente
Open-Meteo	Até 10.000 (uso recomendado)	≈ 105.06 dólares/mês (5 milhões de chamadas/mês)
Pirate Weather	10.000	API completamente gratuita

significativamente maior de chamadas em comparação com as demais. As APIs restantes, embora mais restritivas, refletem diferentes modelos de negócio: Tomorrow.io é voltada para o setor comercial, enquanto Open-Meteo e Pirate Weather seguem princípios de código aberto, alinhando-se ao propósito da pesquisa. Nos planos pagos, OpenWeather apresenta o custo mais alto, porém com o maior volume de chamadas disponíveis.

4.3 Tempo de Resposta

Para a comparação de tempo de resposta, um simples programa que captura a previsão do tempo para um dia em São Paulo, utilizando cada uma das APIs, foi implementado e os resultados estão compilados na Tabela 4. Foram realizadas 10 medições consecutivas para cada API, permitindo o cálculo da média, do desvio padrão e a construção do gráfico do tipo *box plot*, apresentado na Figura 7.

Essa abordagem, embora eficiente para estabelecer uma análise inicial, apresenta limitações, uma vez que os dados foram coletados em um único dia e em uma sequência temporal específica. Essa metodologia pode não capturar variações que ocorrem em diferentes dias ou em horários distintos ao longo do dia, que poderiam influenciar os tempos de resposta das APIs, seja devido a flutuações na demanda ou a condições de infraestrutura dos servidores.

Dessa forma, recomenda-se que trabalhos futuros explorem levantamentos de dados mais abrangentes, incluindo coletas realizadas em múltiplos dias e em diferentes horários. Essa diversificação temporal contribuiria para uma análise mais robusta e representativa, permitindo compreender melhor as variações de desempenho das APIs em diferentes cenários de uso.

A análise dos tempos de resposta das APIs revela diferenças significativas em termos de consistência e rapidez, conforme mostrado na Tabela 4 e no gráfico presente na Figura

Tabela 4 – Resultados de 10 consultas de cada API na Raspberry Pi da previsão de tempo (em milissegundos)

API Name	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7	Time 8	Time 9	Time 10
OpenWeather	429.31	412.78	428.91	366.16	393.18	477.60	440.84	451.12	353.49	373.26
WeatherAPI	772.27	111.45	341.70	362.71	111.93	135.01	390.47	362.75	343.87	373.26
Open-Meteo	1047.12	727.92	669.41	755.88	973.67	599.49	624.83	614.14	606.32	602.97
PirateWeather	848.64	468.07	474.00	460.50	462.04	467.57	537.21	464.95	468.93	501.83
Tomorrow.io	673.67	510.75	480.93	521.98	397.27	481.20	439.13	478.51	485.99	433.12

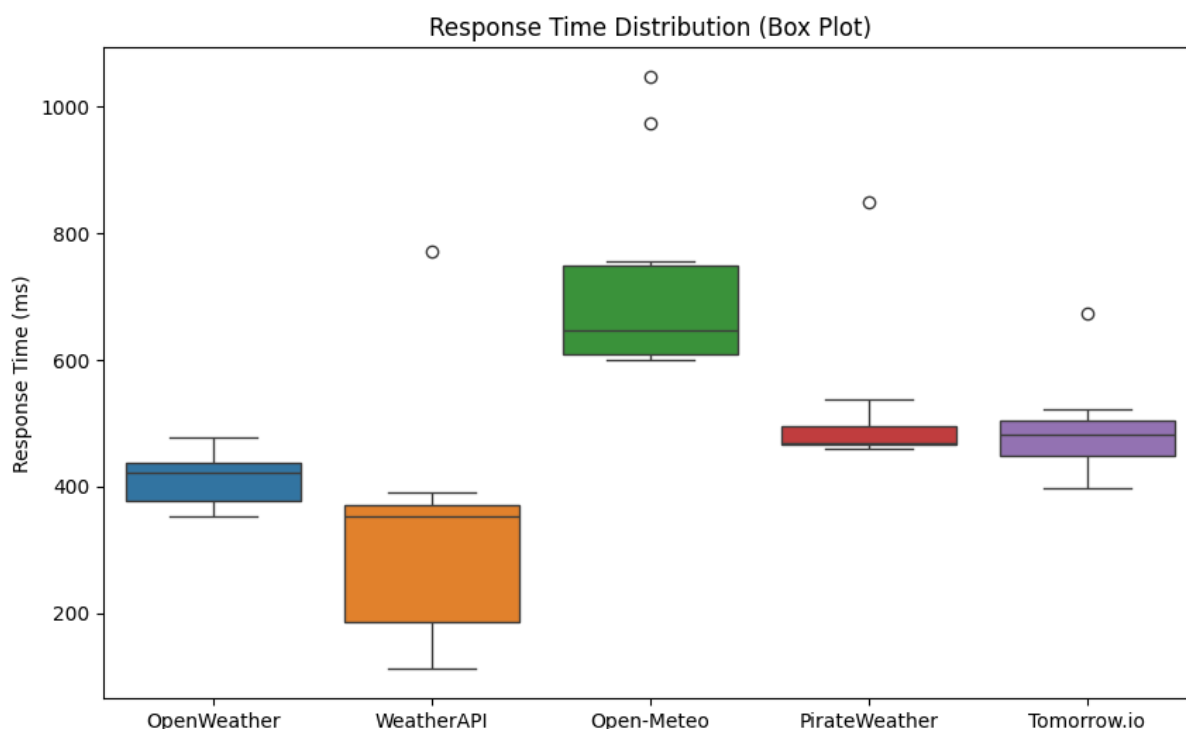


Figura 7 – Gráfico do tipo box-plot comparativo para o tempo de resposta de cada uma das APIs

7. Dentre as APIs avaliadas, a OpenWeather destaca-se pela confiabilidade e baixa variabilidade em seus tempos de resposta, com uma média de 412.66 ms e um desvio padrão de apenas 38.25 ms. A faixa estreita dos valores de resposta e a ausência de outliers no box plot sugerem uma estrutura de backend eficiente e consistente, capaz de oferecer previsibilidade e estabilidade para aplicações que exigem rapidez e baixa latência.

Em contrapartida, a WeatherAPI, apesar de apresentar o menor tempo médio de resposta (330.54 ms), enfrenta alta variabilidade, evidenciada pelo desvio padrão de 183.56 ms e pela ampla faixa de valores. Essa inconsistência é visível no box plot, onde se destacam outliers que sugerem problemas ocasionais de sobrecarga ou flutuações na infraestrutura. A WeatherAPI, portanto, demonstra um desempenho instável, o que pode prejudicar sua utilização em cenários que demandem previsibilidade e tempos de resposta regulares.

A Open-Meteo apresenta o maior tempo médio de resposta (722.17 ms) e uma variação considerável, com um desvio padrão de 153.83 ms. As dispersão nos tempos de resposta e a

presença de outliers no box plot podem indicar fatores como localização geográfica distante dos servidores em relação a São Paulo ou uma infraestrutura menos otimizada. Além disso, por ser uma API open source, é possível que não disponha de recursos financeiros suficientes para aprimorar sua infraestrutura, o que se reflete em um desempenho mais inconsistente e tempos de resposta mais elevados.

A PirateWeather também apresenta uma variabilidade moderada, com uma média de 515.37 ms e um desvio padrão de 113.37 ms, sugerindo uma estrutura de backend que enfrenta picos ocasionais de carga. No box plot, outliers mostram que, embora a média de resposta esteja em um nível aceitável, a API pode apresentar tempos elevados em determinados momentos, possivelmente devido à sua infraestrutura limitada.

Por fim, a Tomorrow.io combina um tempo médio de resposta competitivo (490.25 ms) com uma variabilidade relativamente baixa (desvio padrão de 70.65 ms). A faixa de valores moderada e a ausência de outliers no box plot indicam que a API possui uma infraestrutura robusta e bem desenvolvida, refletindo o elevado padrão de qualidade associado aos produtos da Tomorrow.io. Essa consistência torna a API uma boa opção para aplicações que necessitam de confiabilidade e rapidez, ainda que com um custo possivelmente elevado.

Em resumo, OpenWeather e Tomorrow.io se destacam pela consistência nos tempos de resposta, tornando-se as melhores opções para quem busca previsibilidade. Em contraste, embora a WeatherAPI apresente uma média de resposta rápida, sua alta variabilidade torna os tempos de resposta menos confiáveis. Open-Meteo e PirateWeather, por sua vez, têm tempos de resposta mais elevados e irregulares, o que limita seu uso em sistemas que exigem estabilidade.

4.4 Usabilidade

A API OpenWeather se destaca por ser intuitiva e bem documentada. Sua documentação é clara, tornando simples tanto para iniciantes quanto para desenvolvedores experientes recuperar dados meteorológicos. Em comparação, o WeatherAPI é menos intuitivo, exigindo mais esforço para entender, apesar de ainda ter uma boa documentação, sinto a falta demais detalhes e uma documentação mais organizada.

O Tomorrow.io, embora conte com um site abrangente e repleto de terminologia detalhada e recursos variados, apresenta sérias limitações no quesito usabilidade, especialmente no que se refere à documentação da API. Apesar de incluir exemplos de código, a documentação carece de profundidade e clareza, tornando difícil compreender as possibilidades e limitações do serviço. Além disso, a navegação no site é pouco intuitiva, dificultando a localização de informações relevantes sobre a API, que, por sua vez, são escassas e pouco claras. Outro obstáculo significativo são as cotas restritas de requisições, frequentemente acompanhadas de problemas como timeouts, o que representa um desafio para estudos comparativos envolvendo múltiplas APIs. Ademais, a falta de transparência nos preços, que

só são disponibilizados mediante contato direto, complica ainda mais a tomada de decisões.

Por outro lado, o Open-Meteo oferece uma experiência de usuário direta ao fornecer trechos de código prontos para uso em várias linguagens, tornando-o especialmente acessível para não programadores. No entanto, sua estrutura de dados difere significativamente das outras APIs, o que complica os esforços para generalizar ou modificar o código para casos de uso mais complexos. Embora seja fácil de usar, essa inconsistência dificulta a comparação dos resultados entre as APIs.

O Pirate Weather se destaca em termos de documentação e facilidade de uso. Ele é bem estruturado e intuitivo, mas suas limitações em termos de volume de chamadas reduzem sua aplicabilidade para uso mais amplo.

4.5 Conclusão

Após a análise das cinco principais dimensões de cada uma das APIs de previsão meteorológica, observa-se que OpenWeather e Tomorrow.io destacam-se como as mais robustas, potentes e precisas. No entanto, considerando os requisitos específicos deste projeto e os objetivos definidos, OpenWeather revela-se a escolha mais adequada para o desenvolvimento do TCC. Sua acessibilidade econômica e excelente usabilidade tornam-na preferível em comparação com Tomorrow.io, cujo perfil altamente comercial e gama extensa de funcionalidades avançadas excede as necessidades acadêmicas e experimentais deste estudo. Como o propósito é acadêmico e não comercial, a escolha recai sobre uma API que, ao mesmo tempo que oferece qualidade e precisão, apresenta um custo viável para o contexto de pesquisa.

4.6 Observação sobre APIs de dados solares

Também é importante destacar a API Solcast, que, embora inicialmente considerada para este projeto, não se encaixa diretamente nas comparações realizadas. Solcast, em sua versão gratuita para pesquisadores, fornece dados específicos sobre irradiação solar e produção fotovoltaica, informações cruciais para este projeto, que visa à otimização da produção de energias renováveis, como a solar. Ainda assim, a Solcast não contempla variáveis meteorológicas abrangentes, que são igualmente essenciais para analisar o desempenho de painéis solares e outras aplicações energéticas no projeto.

A API da Solcast, além de ser acessível e voltada para estudantes, oferece dados de grande relevância. Contudo, ao compará-la com outras APIs, percebe-se que o uso da OpenWeather agrega mais valor, visto que sua versão paga oferece dados solares detalhados a um custo de aproximadamente 0,1 GBP por chamada. Essas informações incluem irradiação e dados meteorológicos variados, além de modelos detalhados que superam até mesmo os oferecidos pela Solcast em alguns aspectos.

Portanto, com o apoio do orientador, Dr. Augusto Matheus Alonso, optou-se pelo uso da OpenWeather para abranger todo o projeto, assegurando a aquisição de dados necessários para os testes e desenvolvimento dos protótipos deste estudo.

5 IMPLEMENTAÇÃO DO PROTÓTIPO

5.1 Desenvolvimento do protótipo no IEP

Esta seção descreve o processo de construção dos algoritmos no DSP TMS320F28035-PNT, utilizado para avaliar os dados recebidos da internet, que, por sua vez, podem ser usados para ajustar o funcionamento do IEP. Foi realizado o estudo de como configurá-la em conjunto com a plataforma embarcada, permitindo a comunicação serial. Essa etapa envolveu a análise detalhada da documentação técnica da TI e a compreensão do funcionamento da comunicação UART.

A comunicação serial entre o DSP e a Raspberry Pi foi baseada em um exemplo fornecido pela plataforma *controlSUITE* da TI (CONTROLSUITE. . . , 2024). O módulo ePWM foi utilizado para controlar um LED e atualizar variáveis no ambiente de desenvolvimento *Code Composer Studio* (CCS), que monitoram a entrada de dados. A implementação original foi adaptada para processar os caracteres recebidos pelo DSP, convertendo-os em uma forma de onda senoidal exibida no CCS. Essa onda simula a corrente processada pelo IEP, representando uma injeção hipotética de energia na rede elétrica. Dessa forma, ao receber dados da Raspberry Pi, o IEP ajusta dinamicamente a corrente, demonstrando a integração eficaz entre os dispositivos.

O código implementado permite que o microcontrolador receba e processe dados da Raspberry Pi por meio da porta serial, retornando-os ao mesmo canal após o processamento. Esta configuração simplifica a comunicação entre os dispositivos, permitindo a troca de dados sem a necessidade de sincronização adicional.

Adicionalmente, foi programada uma interrupção que aciona o ePWM, controlando o LED e atualizando a função senoidal. A interrupção também processa o valor recebido pela comunicação serial, utilizando bibliotecas de ponto flutuante, uma vez que o DSP opera em ponto fixo. Esse valor é então modulado para ajustar a forma de onda senoidal.

O código descrito implementado pode ser acessado através do Github ou acessando a url <https://github.com/anajuliatagliassachi/DSP-code>

5.2 Implementação do protótipo na plataforma microprocessada

Para a plataforma microprocessada, como mencionado anteriormente na Seção 3, optou-se por utilizar a Raspberry Pi 4, com o sistema operacional *Raspberry Pi OS Lite*, visando garantir leveza e compatibilidade com os requisitos do projeto (OPERATING. . . , 2024). Durante a execução da pesquisa, a conexão à internet foi feita utilizando um ponto de acesso via celular, já que a rede Wi-Fi da universidade onde a pesquisa foi realizada não estava disponível.

5.2.1 Desenvolvimento do Algoritmo de Decisão

Após a definição das plataformas de implementação e da API a ser utilizada, a próxima etapa do processo envolveu a escolha do algoritmo de tomada de decisão a ser aplicado no projeto.

Embora inicialmente tenha sido considerada a implementação de algoritmos inteligentes, como redes neurais, árvores de decisão e regressões lineares, as limitações práticas levaram à adoção de um algoritmo especializado. A aplicação de uma rede neural robusta, por exemplo, exigiria não apenas uma base de dados extensa e confiável, mas também informações específicas alinhadas aos objetivos do projeto. No entanto, a indisponibilidade de uma fonte de dados adequada, aliada às limitações das APIs utilizadas, inviabilizou a implementação de modelos avançados de inteligência artificial. Assim, optou-se por desenvolver um algoritmo especializado, projetado para realizar uma análise direta e eficiente dos dados meteorológicos fornecidos pela API, a fim de gerar instruções práticas para o sistema IEP e otimizar sua operação.

Diante disso, uma nova fase de investigação começou, focando na revisão da literatura. O objetivo dessa revisão foi entender como as informações disponíveis fornecidas pela API escolhida poderiam ser utilizadas de forma eficaz em relação à produção de energia solar. Em essência, o foco passou a ser compreender como as condições climáticas em tempo real poderiam impactar o funcionamento dos painéis solares. O objetivo final era explorar como essas variáveis poderiam teoricamente influenciar o inversor inteligente, que seria controlado pela Raspberry Pi.

Primeiramente, o desempenho dos módulos PV é significativamente influenciado pela temperatura de operação, com a eficiência diminuindo conforme a temperatura dos painéis aumenta. A eficiência dos painéis solares PV é afetada tanto pela irradiação quanto pela temperatura dos painéis. Enquanto a radiação solar inclui energia radiante e térmica, a geração de energia PV é particularmente impactada pela energia radiante solar. A temperatura da célula, que é determinada por diversos fatores físicos e ambientais, é fundamental para o desempenho elétrico e térmico dos módulos PV em condições reais (GOK et al., 2020).

Em ambientes quentes e ensolarados, como é geralmente o caso do Brasil, a temperatura se destaca como a principal causa de perdas de energia em sistemas PV. Dentre os fatores ambientais que influenciam a produção fotovoltaica, a irradiância é o mais significativo. Diferentes materiais respondem de forma única à ampla gama de comprimentos de onda do espectro solar, afetando os valores de produção com base na seletividade da absorção da faixa de energia nas células PV (do Nascimento et al., 2020).

A escolha de 25 °C como limite de temperatura foi fundamentada na análise das condições típicas de operação de módulos fotovoltaicos e no impacto que a temperatura ambiente exerce sobre sua eficiência. Em torno de 25 °C, os módulos atingem uma faixa ideal de

desempenho, enquanto temperaturas mais elevadas podem provocar superaquecimento das células, reduzindo a eficiência energética devido ao aumento da resistência elétrica interna. Além disso, esse valor reflete uma média aproximada entre as temperaturas médias anuais registradas no Brasil e na Itália, que giram em torno de 30 °C e 19 °C, respectivamente. Essa escolha equilibra a simplicidade do algoritmo com a necessidade de levar em conta as variações ambientais que impactam diretamente o desempenho dos sistemas fotovoltaicos.

Embora a velocidade do vento seja considerada em alguns modelos de classificação de energia devido ao seu impacto na temperatura dos módulos e na eficiência, surgem desafios ao obter valores representativos, já que a velocidade do vento varia bastante dependendo do ambiente local. Apesar dos debates sobre a inclusão da velocidade do vento, o consenso predominante nas pesquisas sugere que a radiação solar e a temperatura ambiente são os parâmetros meteorológicos mais relevantes para classificar e estimar a produção de energia fotovoltaica (FV) (DIAS et al., 2017).

Além disso, uma consideração crucial foi a análise do índice de cobertura de nuvens. Como a API gratuita não fornece explicitamente informações sobre uma característica vital—especificamente detalhes sobre irradiação solar—uma solução alternativa foi explorada. Embora se reconheça que a cobertura de nuvens não equivale a medições diretas de irradiação solar e não tenha a mesma precisão, foi tomada a decisão de implementar um mecanismo que avalie a cobertura de nuvens. Esta abordagem foi escolhida com base no entendimento de que a cobertura de nuvens pode oferecer *insights* sobre a irradiação solar potencial que o painel solar pode experimentar (PARK et al., 2021).

Finalmente, foi realizado um estudo aprofundado sobre a melhor forma de utilizar o índice de irradiação solar. A API de irradiação solar oferece dois tipos de relatórios: *Cloudy Sky* e *Clear Sky*. Ambos relatórios incluem os índices: DNI, DHI e GHI. Todos os índices são fornecidos com resolução temporal horária.

O índice GHI abrange tanto a radiação direta quanto a difusa em um plano horizontal na superfície da Terra (QUASCHNING, 2003). O componente DNI é especificamente definido como a irradiância solar recebida diretamente dos raios solares em todo o espectro solar (BLANC et al., 2014). No contexto dos sistemas FV, uma compreensão abrangente dos recursos de GHI desempenha um papel essencial na previsão precisa da produção de energia solar (LOPES et al., 2018). Além disso, esse conhecimento ajuda a mitigar possíveis flutuações de energia que possam surgir nas injeções de energia na rede elétrica (PASCUAL et al., 2015).

Para aumentar a precisão neste contexto, o algoritmo *Clear Sky* será utilizado. O programa incorpora um mecanismo para avaliar a cobertura de nuvens, distinguindo entre condições superiores ou inferiores a 50%. Essa distinção é crucial, pois uma cobertura de nuvens excessiva tende a reduzir a produtividade. Em seguida, no caso de céu limpo, o algoritmo avalia os índices GHI. Índices GHI favoráveis indicam condições ótimas para a produção de energia, resultando na geração e transmissão de energia para a rede.

Existem muitas outras variáveis, como os aspectos econômicos do despacho de energia, que poderiam ser consideradas no algoritmo. Entretanto, devido à grande complexidade de integrar essas variáveis, elas não foram incluídas, já que o foco principal deste projeto é desenvolver um motor de dados online para otimizar a operação local do inversor em estudo.

O limite de 500 W m^{-2} para o índice GHI foi selecionado com base em referências técnicas que definem esse valor como um limiar mínimo para que sistemas fotovoltaicos operem de forma eficiente em condições de radiação solar moderada. Em cenários onde a irradiância está abaixo desse valor, a produção de energia tende a ser insuficiente para compensar as perdas associadas à conversão e ao armazenamento, tornando mais eficiente a opção de reter energia para usos futuros. Essa abordagem busca garantir que o sistema atue em condições economicamente viáveis, evitando o despacho de energia em períodos de baixa produtividade.

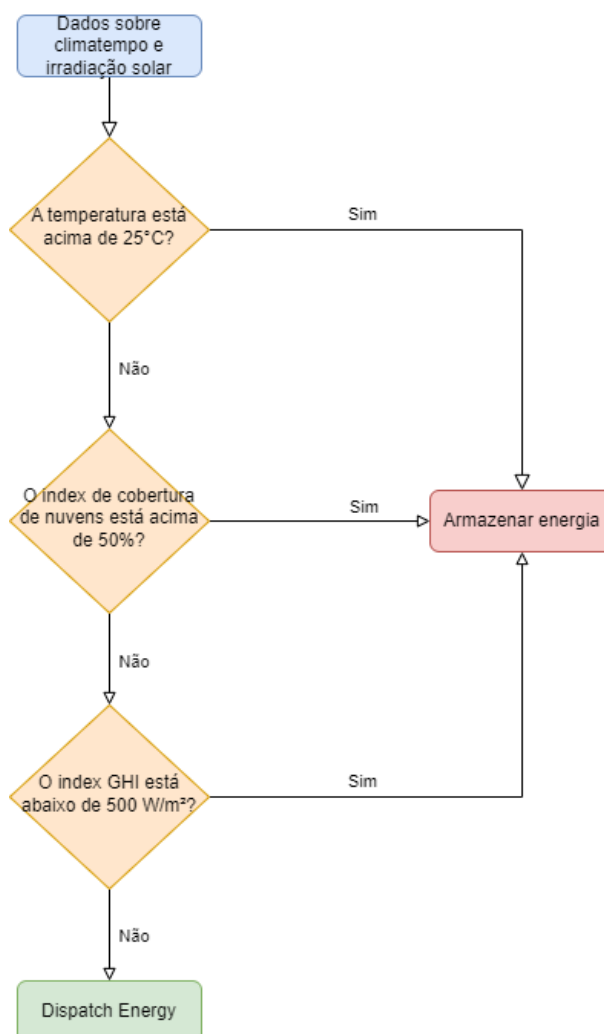
Considerando os pontos mencionados anteriormente, o fluxograma apresentado na Figura 8 foi projetado como base para o algoritmo de tomada de decisão implementado. Assim, para temperaturas superiores a 25°C , o dispositivo é instruído a armazenar energia. Caso contrário, se a temperatura estiver abaixo desse limite, o algoritmo avalia se o índice de nebulosidade ultrapassa 50%. Se sim, o dispositivo armazena energia; caso contrário, verifica-se se o índice GHI está abaixo de 500 W m^{-2} . No caso de o índice GHI registrar um valor abaixo desse limite, o dispositivo novamente é instruído a armazenar energia. Caso contrário, o dispositivo pode proceder com o despacho de energia para a rede.

No desenvolvimento atual do projeto, a simulação do processamento dos dados recebidos via internet resulta na alteração de uma *flag* para cada um dos casos finais do fluxograma, sendo definida como "1" ou "0". Esse processo gera os seguintes textos de saída: "Despachar energia para o grid" ou "Armazene energia. As condições não são favoráveis para a operação completa do painel solar", respectivamente. A versão atual do algoritmo é rudimentar e, em caso de implementação real como um algoritmo de tomada de decisão, seria necessário considerar diversas outras variáveis, como o estado atual de armazenamento ou despacho de energia, além da orientação do painel PV e outros detalhes complexos. Apesar de sua simplicidade, o algoritmo implementado cumpre adequadamente sua função designada, oferecendo uma interpretação simplificada dos aspectos essenciais na produção de energia solar. O código implementado para esse fluxo pode ser observado no Anexo - A.

5.3 Integração entre plataformas

O último passo para finalizar a implementação do protótipo foi estabelecer a comunicação entre as placas. Após a implementação do algoritmo no DSP, a etapa seguinte foi desenvolver o algoritmo responsável pelo envio de dados a partir da Raspberry Pi para

Figura 8 – Fluxograma do processamento realizado pelo algoritmo inteligente programado



Fonte: Elaborado pela autora (2024)

completar a comunicação entre os dispositivos.

Parte central desse código foi o módulo de comunicação serial disponível em Python. Esse módulo permite configurar a porta de comunicação serial da Raspberry Pi com uma taxa de transmissão de 9600 bauds.

A lógica de comunicação serial foi estruturada para duas possíveis aplicações: o primeiro cenário envolve o processamento automático de dados recebidos via Internet pela Raspberry Pi, que são então transmitidos ao DSP; e o segundo cenário oferece a possibilidade de o usuário inserir manualmente um valor, que será utilizado para controlar o funcionamento do IEP presente no DSP.

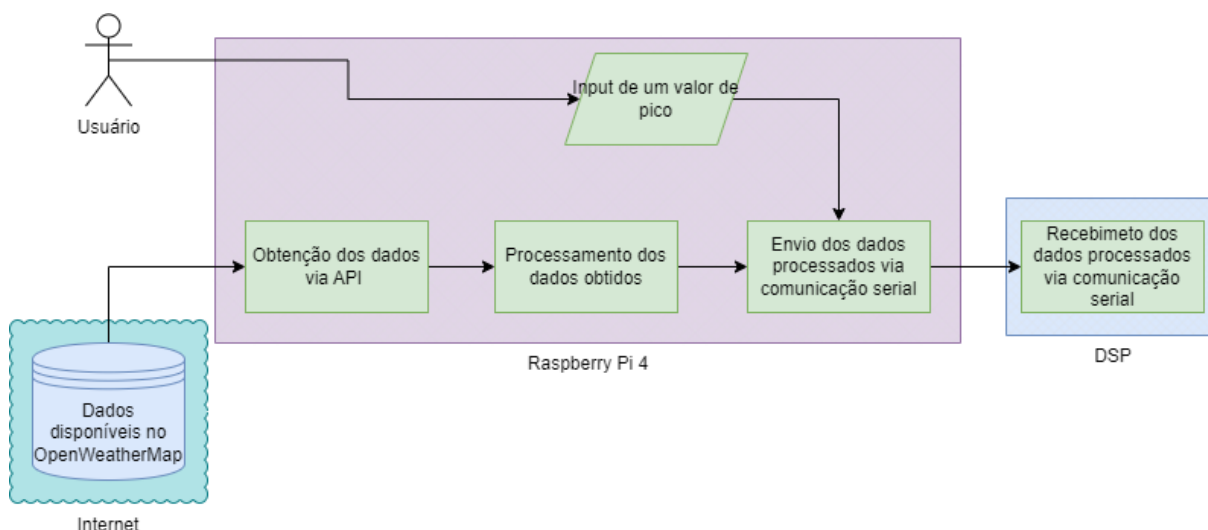
No caso dessa segunda aplicação, o código permite que o usuário forneça um valor que simula uma corrente de pico. Esse valor é convertido para o formato binário e enviado via porta serial para ser processado pelo DSP. A comunicação segue um ciclo contínuo, onde, após cada valor enviado, o sistema aguarda um breve intervalo antes de aceitar novos dados.

Por fim, a porta serial é fechada automaticamente caso não ocorram erros, garantindo o término seguro da comunicação. É dessa forma que a comunicação entre as placas foi desenhada e implementada, utilizando o protocolo UART.

Para assegurar o funcionamento adequado da comunicação, foi realizado o mapeamento dos pinos tanto na Raspberry Pi quanto no DSP. Na Raspberry Pi, os pinos GPIO 14 e 15 foram designados para transmissão e recepção de dados (TXD e RXD, respectivamente). Já no DSP, foi mapeado o pino GPIO 28, correspondente ao SCIRX-A, responsável pela recepção de dados enviados pela Raspberry Pi.

Enfim, com todas as etapas concluídas, o fluxograma da comunicação estabelecida foi delineado, conforme ilustrado na Figura 9.

Figura 9 – Planejamento do fluxograma do protótipo

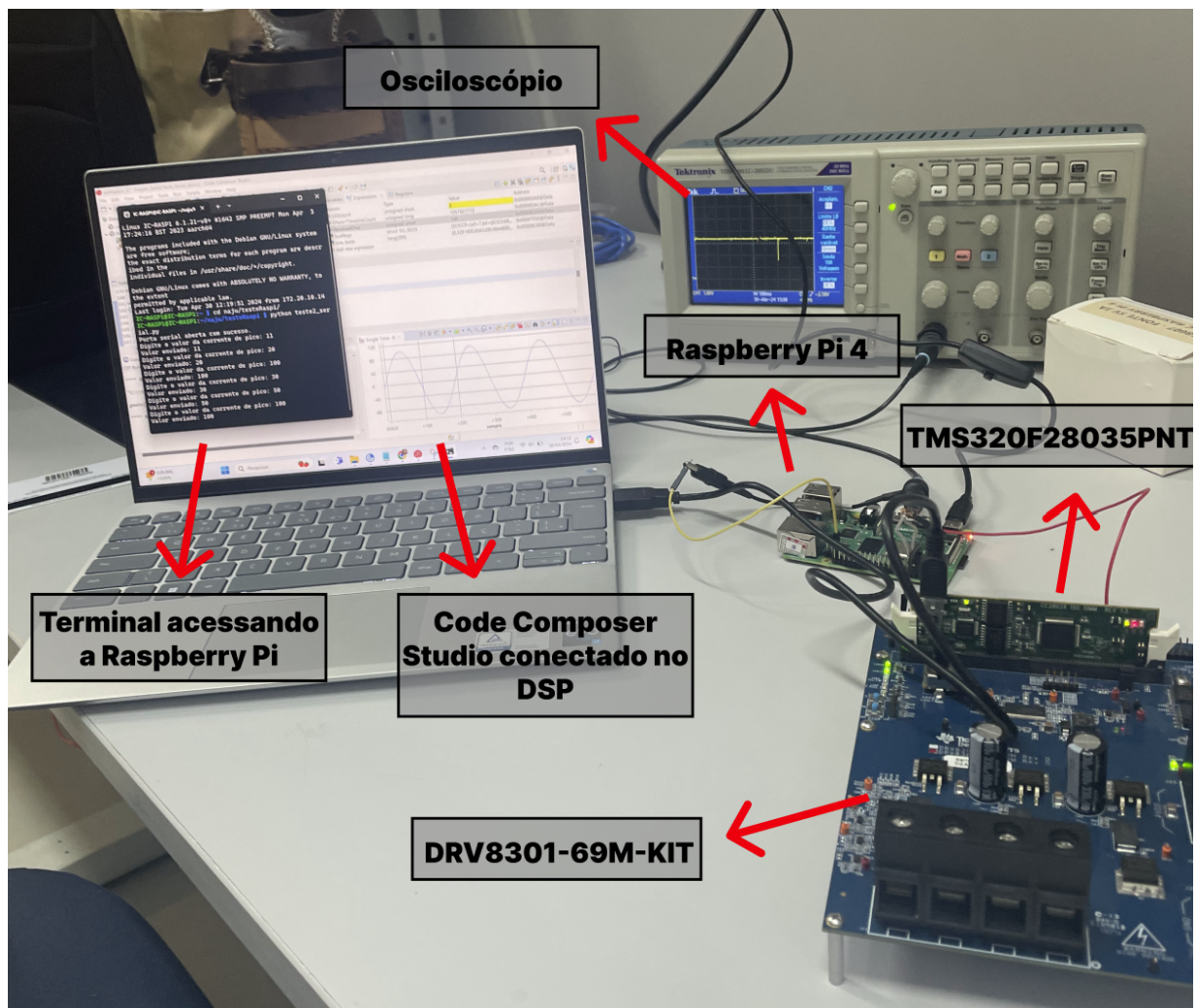


Fonte: Elaborado pela autora (2024)

6 RESULTADOS

Para validar o funcionamento do projeto conforme o esperado, foi construído um protótipo utilizando a Raspberry Pi 4, a plataforma eletrônica, um computador (que acessa tanto a Raspberry quanto o DSP), e um osciloscópio. A configuração desse protótipo pode ser vista nas Figuras 10,11. Vale destacar que tanto a Raspberry Pi quanto o computador estavam conectados à rede 4G do celular da estudante, embora esse aspecto não esteja ilustrado nas figuras.

Figura 10 – Configuração do protótipo montado



Fonte: elaborado pela autora (2024)

Além disso, é importante enfatizar que a escolha da comunicação via UART no protótipo proporciona maior flexibilidade, permitindo a utilização de plataformas diferentes da Raspberry Pi. Isso acontece porque o DSP está preparado para receber uma variável que ajusta a operação do IEP.

Conforme o fluxograma apresentado na Figura 9, dois casos de estudo foram explorados com base nas funcionalidades implementadas: a inserção de um comando pelo usuário no

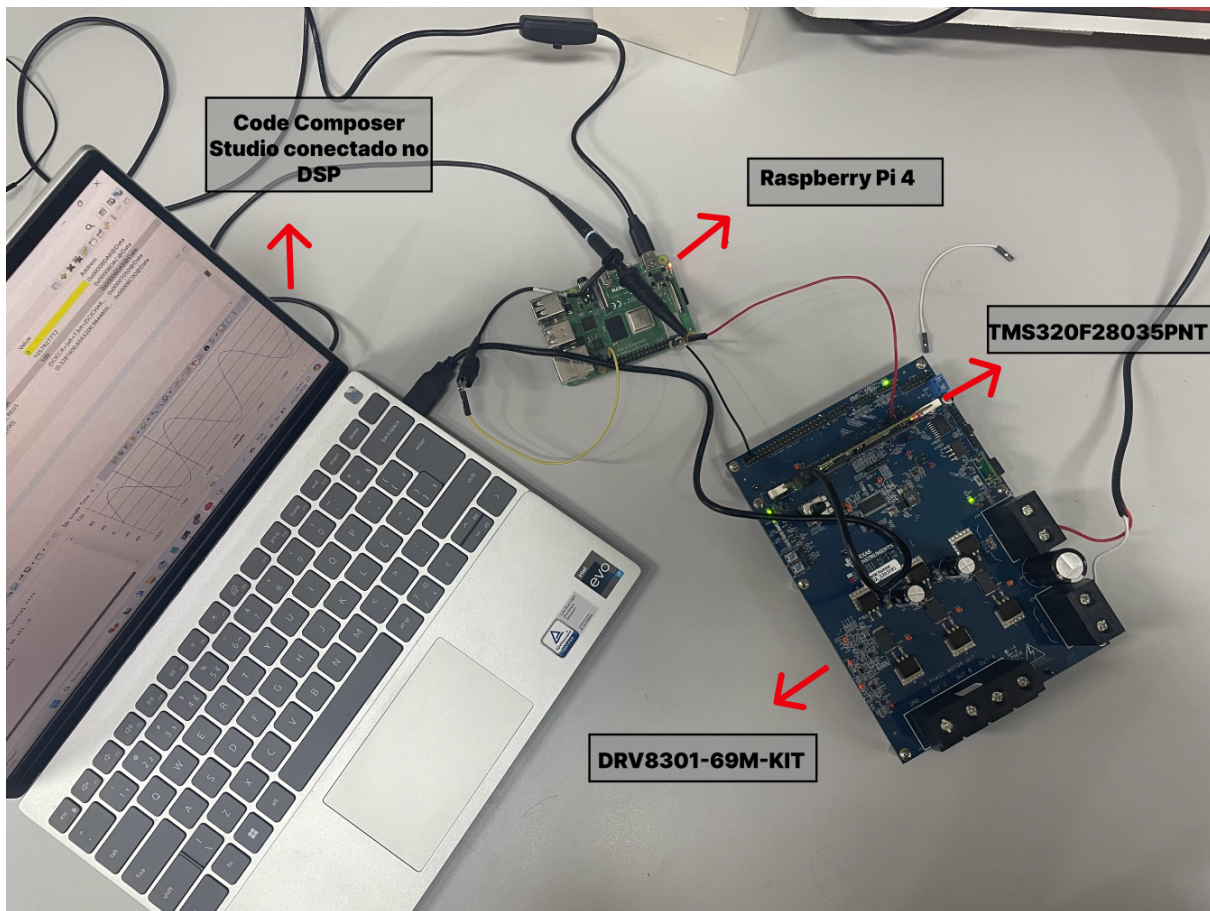


Figura 11 – Vista detalhada da montagem do protótipo

Fonte: elaborado pela autora (2024)

fluxo do código e a coleta e processamento de dados obtidos diretamente da internet por meio de uma API.

6.1 Procedimento Operacional do Protótipo

O acesso remoto à Raspberry Pi, por meio do terminal do computador, era feito via SSH, desde que ambos estivessem conectados à mesma rede. O DRV8301-69M-KIT era ligado à tomada, alimentando todos os componentes da placa, enquanto o DSP era conectado ao computador por USB. O código em C, desenvolvido e acessado pelo CCS, foi carregado e executado no DSP, enquanto o código Python, programado diretamente no terminal usando o editor nano, rodava na Raspberry Pi.

6.1.1 Fluxo de Entrada do Usuário

No primeiro fluxo, representado pela Figura 10, o usuário pode inserir um valor via *input*, simulando uma corrente de pico. Esse sinal pode ser observado na tela do osciloscópio,

validando o envio do comando para outro dispositivo. O sinal gerado pela Raspberry Pi e recebido pelo DSP altera a amplitude da onda senoidal exibida no CCS.

Na Figura 12, é mostrado um exemplo de saída do código. Inicialmente, a onda de frequência 60 Hz tem valor de pico de 50 A, e, após a simulação de um *input* de 100 A na Raspberry Pi, a amplitude da onda se ajusta para 100 A. Essa variação é observada na figura, evidenciando que o valor de corrente inserido foi transmitido corretamente via UART para o DSP, que ajusta a corrente processada pelo IEP, demonstrando a funcionalidade da comunicação serial entre os dispositivos.

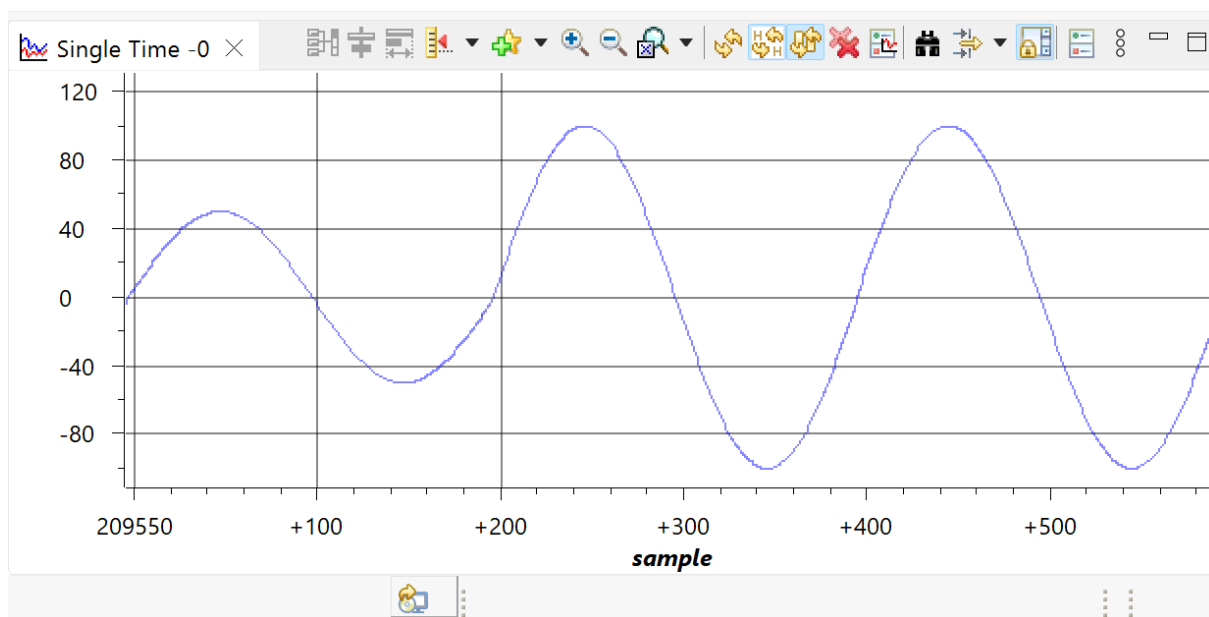


Figura 12 – Onda senoidal mostrada no CCS.

6.1.2 Processamento de Dados Obtidos via API

No segundo fluxo, a Raspberry Pi acessa a Internet, onde os dados meteorológicos são processados por um algoritmo de tomada de decisão. Dependendo dos dados processados, o sistema decide se o IEP deve despachar energia para o inversor ou armazená-la. Se o resultado do processamento for “1”, o IEP deverá despachar energia ao inversor. Se o resultado for “0”, a energia deve ser armazenada. Esse algoritmo é executado periodicamente a cada hora, e a Raspberry Pi é capaz de acessar qualquer banco de dados ou API online, reforçando seu poder de processamento.

Um exemplo de saída do algoritmo inteligente rodando na Raspberry Pi pode ser observado na Figura 13. O programa avalia condições meteorológicas tanto em Trento, na Itália, quanto em São Carlos, no Brasil, locais onde diferentes partes do estudo foram conduzidas. A opção por Trento deve-se ao fato de a pesquisa ter sido desenvolvida naquela região, que oferece um cenário específico de condições climáticas para validação do modelo proposto. A saída do programa que indica *“Store energy. The conditions are not favorable*

for the full operation of the solar panel." gera um sinal serial correspondente de "0" para o DSP. Caso contrário, a saída indicaria *"Dispatch energy to the grid."*

```
Weather report:
City: Province of Trente
Country: IT
Current Temperature: 3.98°C
Cloudiness: 0%
Max Temperature: 6.36°C
Min Temperature: 1.96°C
Clear Sky GHI at 16:00 in Province of Trente: 31.79 W/m²

Store energy. The conditions are not favorable for the full operation of the solar panel.

Weather report:
City: São Carlos
Country: BR
Current Temperature: 30.17°C
Cloudiness: 62%
Max Temperature: 30.17°C
Min Temperature: 30.17°C
Clear Sky GHI at 12:00 in São Carlos: 1118.41 W/m²

Store energy. The conditions are not favorable for the full operation of the solar panel.
```

Figura 13 – Resultado exibido no terminal da Raspberry Pi.

Por fim, é importante destacar a flexibilidade da abordagem de integração entre o IEP e a Internet neste projeto:

- A Raspberry Pi facilita o acesso a qualquer banco de dados ou API disponível na internet, o que permite aumentar a complexidade dos algoritmos de tomada de decisão empregados na gestão do IEP;
- Devido ao uso da comunicação UART, outras plataformas podem ser utilizadas no lugar da Raspberry Pi, já que o DSP apenas interpreta os dados disponíveis nos seus pinos de comunicação;
- Da mesma forma, outras plataformas de controle local do IEP poderiam substituir o DSP, pois a integração do sistema à internet é feita pelo sistema microprocessado com maior poder de computação, no caso, a Raspberry Pi;
- Além disso, o algoritmo de tomada de decisão pode ser facilmente ajustado para atender a diferentes necessidades do operador, tornando a implementação flexível para qualquer IEP, incluindo conversores CC-CC usados em microrredes de corrente contínua.

7 CONCLUSÕES

Este trabalho demonstrou que a integração de uma plataforma contendo um IEP com uma plataforma microprocessada, a Raspberry Pi 4, é viável. A plataforma foi conectada à internet e integrada a uma API de dados meteorológicos em tempo real, além de utilizar um algoritmo inteligente desenvolvido durante o experimento. Esse desenvolvimento contribuiu para ampliar as pesquisas do grupo LFAPE na EESC, resultando em um protótipo versátil para estudos de interoperabilidade de IEPs e sua conexão com a Internet da Energia.

Além disso, foi realizada uma análise detalhada das APIs meteorológicas disponíveis, comparando-as em termos de precisão, custo, usabilidade e tempo de resposta. Essa análise, complementada por uma revisão de artigos relevantes, apontou a OpenWeather como a API mais adequada para a criação de um projeto que visa integrar dados relevantes para a implementação de energias limpas e a criação de um protótipo que simula a IE. Observou-se, por exemplo, que a OpenWeather apresentou menor desvio em temperatura para São Paulo (2.21°C), enquanto a Tomorrow.io destacou-se na previsão de umidade e cobertura de nuvens em diversas cidades, com desvios de até 9,36% e 43,68%, respectivamente.

Esse trabalho ressaltou a importância dos IEPs inteligentes no contexto das mudanças climáticas e da demanda crescente por fontes de energia mais sustentáveis. O projeto demonstrou como a tecnologia, em conjunto com componentes elétricos estabelecidos, pode contribuir para um futuro mais sustentável e integrado.

A comunicação eficiente entre a Raspberry Pi e a plataforma eletrônica, somada ao processamento de dados pelo algoritmo inteligente, ofereceu uma alternativa automatizada e eficiente para processos anteriormente manuais, permitindo ao usuário também o controle manual da placa e maior autonomia sobre o IEP.

O estudo impulsionou não apenas a integração tecnológica, mas também a exploração de novas técnicas de comunicação, a análise de plataformas disponíveis no mercado e a implementação prática de algoritmos simplificados.

Assim, o projeto não apenas alcançou seus objetivos científicos como trabalho de conclusão de curso, mas também proporcionou percepções valiosas sobre o papel das tecnologias de tempo real e da IE na sustentabilidade. Enfatizou como ações direcionadas podem contribuir para desacelerar as mudanças climáticas, incentivando a inovação e promovendo uma rede colaborativa entre usuários de IE, beneficiando não apenas o Brasil, mas o mundo.

Como propostas para trabalhos futuros, sugere-se expandir o projeto com a automação da otimização dos IEPs, utilizando dados em tempo real para aprimorar a eficiência energética e implementar uma inteligência artificial mais robusta, capaz de considerar fatores adicionais, como a orientação de painéis fotovoltaicos e o estado atual do IEP, se está despachando ou armazenando energia. Essas melhorias transformariam o protótipo em um sistema avançado de gestão de energia inteligente, alinhado às demandas da Internet da

Energia e aos requisitos de ambientes energéticos complexos.

REFERÊNCIAS

- ACCUWEATHER. *AccuWeather - Weather for Life*. 2024. <https://www.accuweather.com/>. Acessado em: 25 de outubro de 2024.
- Alexander Rey. *Pirate Weather*. 2024. <https://pirateweather.net/en/latest/>. Accessed: October 1, 2024.
- ANGADI, S. et al. Comprehensive review on solar, wind and hybrid wind-pv water pumping systems-an electrical engineering perspective. *CPSS Transactions on Power Electronics and Applications*, v. 6, n. 1, p. 1–19, 2021.
- ARAÚJO, R. S. de et al. Fontes de energias renováveis: pesquisas, tendências e perspectivas sobre as práticas sustentáveis. *Research, Society and Developmen*, 2022.
- ARBAB-ZAVAR, B. et al. Smart inverters for microgrid applications: A review. *Energies*, v. 12, n. 5, 2019. ISSN 1996-1073. Disponível em: <https://www.mdpi.com/1996-1073/12/5/840>.
- BACH, W. Fossil fuel resources and their impacts on environment and climate. *International Journal of Hydrogen Energy*, v. 6, n. 2, p. 185–201, 1981. ISSN 0360-3199. Disponível em: <https://www.sciencedirect.com/science/article/pii/0360319981900070>.
- BLANC, P. et al. Direct normal irradiance related definitions and applications: The circumsolar issue. *Solar Energy*, v. 110, p. 561–577, 2014. ISSN 0038-092X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0038092X14004824>.
- BOWES, S.; MOUNT, M. Microprocessor control of pwm inverters. *IEE Proceedings B (Electric Power Applications)*, v. 128, p. 293–305, 1981. Disponível em: <https://digital-library.theiet.org/doi/abs/10.1049/ip-b.1981.0053>.
- BRIGHT, J. M. Solcast: Validation of a satellite-derived solar irradiance dataset. *Solar Energy*, Elsevier, v. 189, p. 435–449, 2019.
- CHENG, L. et al. Energy internet access equipment integrating cyber-physical systems: Concepts, key technologies, system development, and application prospects. *IEEE Access*, v. 7, p. 23127–23148, 2019.
- CHO, H. S.; YAMAZAKI, T.; HAHN, M. Determining location of appliances from multi-hop tree structures of power strip type smart meters. *IEEE Transactions on Consumer Electronics*, v. 55, n. 4, p. 2314–2322, 2009.
- CONTROLSUITE Brochure. 2024. <https://www.ti.com/lit/ml/sprb199a/sprb199a.pdf>. (Accessed on 04/29/2024).
- CORRIGAN, S. *Introduction to the Controller Area Network (CAN)*. [S.l.], 2002.
- DIAS, C. L. de A. et al. Performance estimation of photovoltaic technologies in brazil. *Renewable Energy*, v. 114, p. 367–375, 2017. ISSN 0960-1481. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0960148117306468>.
- do Nascimento, L. R. et al. Performance assessment of solar photovoltaic technologies under different climatic conditions in brazil. *Renewable Energy*, v. 146, p. 1070–1082, 2020. ISSN 0960-1481. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0960148119310006>.

DRV830X-HC-C2-KIT Hardware Reference Guide. [S.l.], 2014.

FRONDUTI, A. *How Accurate Are Weather Forecasts? Comparing the Most Popular Weather Forecasting Services*. 2020. Relatório Final, Earth Space Sciences, 9th Grade.

GARCIA, R. A. et al. Multiple dimensions of climate change and their implications for biodiversity. *Science*, v. 344, 2014.

GOK, A. et al. The influence of operating temperature on the performance of bipv modules. *IEEE Journal of Photovoltaics*, PP, p. 1–8, 06 2020.

HANSEN, B. U. *Internet Weather Forecast Accuracy*. 2007. OmniNerd.com. Acessado em 19 de outubro de 2013. Disponível em: http://www.omninerd.com/articles/Internet_Weather_Forecast_Accuracy.

HANSEN, J.; SATO, M.; RUEDY, R. Perception of climate change. *Proc. Natl. Acad. Sci.*, v. 109, p. 14726–14727, E2415–E2423, 2012.

HIMEUR, Y. et al. Edge ai for internet of energy: Challenges and perspectives. *Internet of Things*, v. 25, p. 101035, 2024. ISSN 2542-6605. Disponível em: <https://www.sciencedirect.com/science/article/pii/S254266052300358X>.

HOSSEINZADEH, N. et al. Voltage stability of power systems with renewable-energy inverter-based generators: A review. *Electronics*, v. 10, n. 2, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/2/115>.

HUANG, A. Q. et al. The future renewable electric energy delivery and management (freedm) system: The energy internet. *Proceedings of the IEEE*, v. 99, n. 1, p. 133–148, 2011.

HUSSAIN, H. M. et al. What is energy internet? concepts, technologies, and future directions. *IEEE Access*, v. 8, p. 183127–183145, 2020.

IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces. *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, p. 1–138, 2018.

IPAKCHI, F. A. A. Grid of the future. *IEEE Power and Energy Magazine*, v. 7, 2009.

JOSEPH, A.; BALACHANDRA, P. Smart grid to energy internet: A systematic review of transitioning electricity systems. *IEEE Access*, v. 8, p. 215787–215805, 2020.

KALINSKY, D.; KALINSKY, R. Introduction to serial peripheral interface. *Embedded Systems Programming*, 2002.

KOZANIS, S. et al. openmeteo.org: A web service for the dissemination of free meteorological data. In: HELMIS, C. G.; NASTOS, P. T. (Ed.). *Advances in Meteorology, Climatology and Atmospheric Physics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 203–208. ISBN 978-3-642-29172-2.

LADDHA, N. R.; THAKARE, A. P. International journal of advanced research in computer science and software engineering. 2013.

LIMA, M. et al. Renewable energy in reducing greenhouse gas emissions: Reaching the goals of the paris agreement in brazil. *Environmental Development*, v. 33, p. 100504, 2020. ISSN 2211-4645. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2211464520300191>.

LOPES, F. M. et al. Short-term forecasts of ghi and dni for solar energy systems operation: assessment of the ecmwf integrated forecasting system in southern portugal. *Solar Energy*, v. 170, p. 14–30, 2018. ISSN 0038-092X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0038092X18304729>.

MARAFÃO, F. B. et al. Multi-task control strategy for grid-tied inverters based on conservative power theory. *The Institute of Engineering and Technology*, 2015.

MARAGATHAM, T.; BALASUBRAMANIE, P.; VIVEKANANDHAN, M. Iot based home automation system using raspberry pi 4. *IOP conference series*, v. 1055, n. 1, p. 012081–012081, 2 2021. Disponível em: <https://iopscience.iop.org/article/10.1088/1757-899X/1055/1/012081/meta>.

MINISTÉRIO DE MINAS E ENERGIA. *BALANÇO ENERGÉTICO NACIONAL*. [S.l.], 2022.

MIRAFZAL, B.; ADIB, A. On grid-interactive smart inverters: Features and advancements. *IEEE Access*, v. 8, p. 160526–160536, 2020.

MUDALIAR, M. D.; SIVAKUMAR, N. Iot based real time energy monitoring system using raspberry pi. *Internet of Things*, v. 12, p. 100292, 2020. ISSN 2542-6605. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2542660520301244>.

MUSAH, A. et al. An evaluation of the openweathermap api versus inmet using weather data from two brazilian cities: Recife and campina grande. *Data*, v. 7, n. 8, 2022. ISSN 2306-5729. Disponível em: <https://www.mdpi.com/2306-5729/7/8/106>.

NREL. *From the Bottom Up: Designing a Decentralized Power System: Cross-Discipline Team Envisions the Grid of the Future*. [S.l.], 2022.

Open-Meteo. *Open-Meteo*. 2024. Acesso em: 12 out. 2024. Disponível em: <https://open-meteo.com/>.

OpenWeatherMap. *OpenWeatherMap API*. 2024. Acesso em: 12 out. 2024. Disponível em: <https://openweathermap.org/api>.

OPERATING system images – Raspberry Pi. 2024. <https://www.raspberrypi.com/software/operating-systems/>. (Accessed on 04/30/2024).

PAAPPANEN, M. *Weather API*. 28 p. Bachelor's thesis — Turku University of Applied Sciences, December 2017.

PARK, S. et al. Prediction of solar irradiance and photovoltaic solar energy product based on cloud coverage estimation using machine learning methods. *Atmosphere*, v. 12, n. 3, 2021. ISSN 2073-4433. Disponível em: <https://www.mdpi.com/2073-4433/12/3/395>.

PASCUAL, J. et al. Energy management strategy for a renewable-based residential microgrid with generation and demand forecasting. *Applied Energy*, v. 158, p. 12–25, 2015. ISSN 0306-2619. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0306261915009733>.

QUASCHNING, V. The sun as an energy resource. *Renewable Energy World*, v. 6, n. 5, p. 90–93, 2003.

RASPBERRY PI LTD. *Raspberry Pi OS – Raspberry Pi*. 2023. Acesso em: 13 out. 2023. Disponível em: <https://www.raspberrypi.com/software/>.

RIFKIN, J. *The Third Industrial Revolution: How Lateral Power Is Transforming Energy, the Economy, and the World Hardcover*. [S.l.]: St. Martin's Press, 2011.

SCHEUERMANN, I. Mosaico weather api: desenvolvimento de web api's em r para acesso a dados de previsão do tempo. *Artigo de conclusão de curso (Bacharel em Computação). Curso de Computação. Universidade de Passo Fundo, Passo Fundo, RS*, 2019.

SEDDON, R. et al. Sensitivity of global terrestrial ecosystems to climate variability. *Nature, Nature Portfolio*, v. 531, n. 7593, p. 229–232, 2 2016. Disponível em: <https://www.nature.com/articles/nature16986#citeas>.

SERBAN, I. et al. Communication requirements in microgrids: A practical survey. *IEEE Access*, v. 8, p. 47694–47712, 2020.

SI, Y. et al. A high performance communication architecture for a smart micro-grid testbed using customized edge intelligent devices (eids) with spi and modbus tcp/ip communication protocols. *IEEE Open Journal of Power Electronics*, v. 2, p. 2–17, 2021.

Solcast. *Solcast: a DNV Company*. 2024. Acesso em: 12 out. 2024. Disponível em: <https://www.solcast.com/>.

STEIGERWALD, R. Power electronic converter technology. *Proceedings of the IEEE*, v. 89, n. 6, p. 890–897, 2001.

TANG, Z.; YANG, Y.; BLAABJERG, F. Power electronics: The enabling technology for renewable energy integration. *CSEE Journal of Power and Energy Systems*, v. 8, n. 1, p. 39–52, 2022.

Tomorrow.io. *Tomorrow.io*. 2024. Acesso em: 12 out. 2024. Disponível em: <https://www.tomorrow.io/>.

VALENTE, A. et al. A lorawan iot system for smart agriculture for vine water status determination. *Agriculture*, v. 12, n. 10, 2022. ISSN 2077-0472. Disponível em: <https://www.mdpi.com/2077-0472/12/10/1695>.

VELOSA, N. et al. Procsim: An open-source simulator to generate energy community power demand and generation scenarios. *Energies*, MDPI, v. 16, n. 4, p. 1611, 2023.

WAVHALE, V. D. et al. Weather data forecast and analytics. *INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY (IRJET)*, 2020.

Weatherapi. *Weatherapi*. 2024. Acesso em: 12 out. 2024. Disponível em: <https://www.weatherapi.com>.

WILLIAMS, A. Role of fossil fuels in electricity generation and their environmental impact. *IEE Proceedings A (Science, Measurement and Technology)*, v. 140, p. 8–12, 1993. Disponível em: <https://digital-library.theiet.org/doi/abs/10.1049/ip-a-3.1993.0003>.

WOOTTON, C. Serial peripheral interface (spi). In: _____. *Samsung ARTIK Reference: The Definitive Developers Guide*. Berkeley, CA: Apress, 2016. p. 335–349. ISBN 978-1-4842-2322-2. Disponível em: https://doi.org/10.1007/978-1-4842-2322-2_21.

XIA, K. et al. A real-time monitoring system based on zigbee and 4g communications for photovoltaic generation. *CSEE Journal of Power and Energy Systems*, v. 6, n. 1, p. 52–63, 2020.

YANQUI, E. F. C.; MERCHAN, K. J. P. *Sistema de Riego Inteligente usando Machine Learning*. Proyecto Integrador — Escuela Superior Politécnica del Litoral, Guayaquil - Ecuador, 2023.

ZHIHONG, J. et al. Energy internet - a new driving force for sustainable urban development. *Energy Procedia*, v. 152, p. 1206–1211, 2018. ISSN 1876-6102. Cleaner Energy for Cleaner Cities. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1876610218307148>.

ZHOU, K.; YANG, S.; SHAO, Z. Energy internet: The business perspective. *Applied Energy*, v. 178, p. 212–222, 2016. ISSN 0306-2619. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0306261916308273>.

ZHOU, K.; YANG, S.; SHAO, Z. Energy internet: The business perspective. *Applied Energy*, v. 178, p. 212–222, 2016. ISSN 0306-2619. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0306261916308273>.

ANEXO

ANEXO A - CÓDIGO IMPLEMENTADO NA RASPBERRY PI

```

1 import requests
2 from datetime import datetime, timedelta
3 import csv
4 import time
5
6 # Function to get weather data for general weather information
7 def get_weather(api_key, latitude, longitude, timezone):
8     base_url = "http://api.openweathermap.org/data/2.5/weather"
9     params = {
10         "lat": latitude,
11         "lon": longitude,
12         "tz": timezone,
13         "appid": api_key,
14         "units": "metric", # Specify units as Celsius
15     }
16
17     response = requests.get(base_url, params=params)
18
19     if response.status_code == 200:
20         data = response.json()
21         return data
22     else:
23         print(f"Error: {response.status_code}")
24         return None
25
26 # Function to print weather information
27 def print_weather(weather_data, clear_sky_ghi, current_hour_solar):
28     print("\nWeather report:")
29     print("City: {}".format(weather_data['name']))
30     print("Country: {}".format(weather_data['sys']['country']))
31     print("Current Temperature: {} C".format(weather_data['main']['temp']))
32     print("Cloudiness: {}%".format(weather_data['clouds']['all']))
33     print("Max Temperature: {} C".format(weather_data['main']['temp_max']))
34     print("Min Temperature: {} C".format(weather_data['main']['temp_min']))
35     #print(f"Clear Sky DNI at {current_hour_solar}:00 in {weather_data['name']}: {clear_sky_dni_trento} W/m^2")
36     print(f"Clear Sky GHI at {current_hour_solar}:00 in {weather_data['name']}: {clear_sky_ghi} W/m^2")
37
38
39 # Function to save weather information to a text file
40 def save_to_file(weather_data, file_path):

```

```

41     with open(file_path, 'w') as file:
42         file.write("City: {}\n".format(weather_data['name']))
43         file.write("Country: {}\n".format(weather_data['sys']['country']))
44         file.write("Current Temperature: {} C\n".format(weather_data['main']
45 ]['temp']))
46         file.write("Cloudiness: {}%\n".format(weather_data['clouds']['all']
47 ]))
48         file.write("Max Temperature: {} C\n".format(weather_data['main']['temp_max']))
49         file.write("Min Temperature: {} C\n".format(weather_data['main']['temp_min']))
50         print(f"Data saved to {file_path}")
51
52 # Function to save weather information to a CSV file
53 def save_to_csv(weather_data, file_path):
54     with open(file_path, 'w', newline='') as csvfile:
55         fieldnames = ['City', 'Country', 'Current Temperature ( C)', '
56 Cloudiness (%)', 'Max Temperature ( C)', 'Min Temperature ( C)', '
57 Timestamp']
58         writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
59
60         writer.writeheader()
61         writer.writerow({
62             'City': weather_data['name'],
63             'Country': weather_data['sys']['country'],
64             'Current Temperature ( C)': weather_data['main']['temp'],
65             'Cloudiness (%)': weather_data['clouds']['all'],
66             'Max Temperature ( C)': weather_data['main']['temp_max'],
67             'Min Temperature ( C)': weather_data['main']['temp_min'],
68         })
69
70 # Function to perform weather classification
71 def classification(data_weather, data_solar):
72     if data_weather['main']['temp'] > 25:
73         return 0
74     else:
75         if data_weather['clouds']['all'] > 50:
76             return 0
77         else:
78             if data_solar < 500:
79                 return 0
80             else:
81                 return 1
82
83 # Function to print weather classification
84 def print_classification(status):
85     if status == 0:

```

```

82         print("\nStore energy. The conditions are not favorable for the
full operation of the solar panel.")
83     else:
84         print("\nDispatch energy to the grid.")
85
86 # Function to get solar data for specific solar information
87 def get_solar_data(api_key, latitude, longitude, date):
88     base_url = "https://api.openweathermap.org/energy/1.0/solar/data"
89     params = {
90         "lat": latitude,
91         "lon": longitude,
92         "appid": api_key,
93         "date": date,
94         "units": "metric",
95     }
96
97     response = requests.get(base_url, params=params)
98
99     if response.status_code == 200:
100         data = response.json()
101         return data
102     else:
103         print(f"Error: {response.status_code}")
104         return None
105
106 # Function to get Clear Sky GHI for a specific hour
107 def get_clear_sky_ghi(data, current_hour):
108     hourly_data = next((item for item in data['irradiance']['hourly'] if
item['hour'] == current_hour), None)
109
110     if hourly_data:
111         clear_sky_ghi = hourly_data['clear_sky']['ghi']
112         return clear_sky_ghi
113     else:
114         print(f"No data found for hour {current_hour}")
115         return None
116
117 # Function to get Clear Sky DNI for a specific hour
118 def get_clear_sky_dni(data, current_hour):
119     hourly_data = next((item for item in data['irradiance']['hourly'] if
item['hour'] == current_hour), None)
120
121     if hourly_data:
122         clear_sky_dni = hourly_data['clear_sky']['dni']
123         return clear_sky_dni
124     else:
125         print(f"No data found for hour {current_hour}")

```

```

126         return None
127
128 #Function to get location time considering timezone offset
129 def get_location_time(data):
130     timezone_offset = data.get('tz_offset', 0) # Offset in seconds
131     current_utc_time = datetime.utcnow()
132     location_time = current_utc_time + timedelta(seconds=timezone_offset)
133     return location_time
134
135 # Function to get solar information and evaluate the output
136 def get_info(data, api_key_solar, latitude, longitude):
137     # Extract date and time from the weather API response
138     api_time_utc = datetime.utcfromtimestamp(data['dt'])
139     api_time_local = api_time_utc + timedelta(seconds=data['timezone'])
140
141     #Use the local time obtained from the weather API for solar data
142     request
143     date_solar = api_time_local.date()
144
145     # Get solar data for Trento
146     data_solar = get_solar_data(api_key_solar, latitude, longitude,
147     date_solar)
148
149     if data_solar is not None:
150         # Get Clear Sky DNI for the current hour
151         #clear_sky_dni = get_clear_sky_dni(data_solar, api_time_local.hour)
152
153         #Get Clear Sky GHI data for the current hour
154         clear_sky_ghi = get_clear_sky_ghi(data_solar, api_time_local.hour)
155
156         if clear_sky_ghi is not None:
157             # Print general weather information
158             print_weather(data, clear_sky_ghi, api_time_local.hour)
159             # Perform weather classification
160             status = classification(data, clear_sky_ghi)
161             # Print weather classification
162             print_classification(status)
163
164 # Main function
165 def main():
166     # General weather API key
167     api_key_weather = 'e611a2afc532e306be926cd1cf085606'
168     # Solar data API key
169     api_key_solar = '6bbeffa72df02a49d9e30097b96ff15b'
170
171     # Shared coordinates for Trento, Italy
172     # Exactly from the building I am working from: CREATE-ET FBK

```