

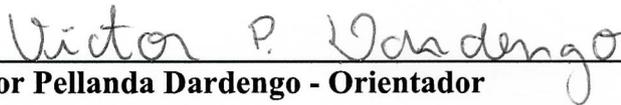
RENATA BANDEIRA GOMES

**DASHBOARD INTERATIVO PARA MONITORAMENTO DE USINAS
FOTOVOLTAICAS: ANÁLISE COMPARATIVA E GESTÃO DE
DESEMPENHO**

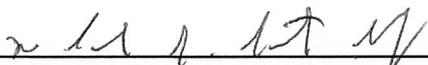
Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 - Projeto de Engenharia II e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 28 de janeiro de 2025.

COMISSÃO EXAMINADORA



Prof. Dr. Victor Pellanda Dardengo - Orientador
Universidade Federal de Viçosa



Prof. Dr. José Carlos da Costa Campos - Membro
Universidade Federal de Viçosa



Prof. Dr. Mauro de Oliveira Prates – Membro
Universidade Federal de Viçosa

Dashboard Interativo para Monitoramento de Usinas Fotovoltaicas: Análise Comparativa e Gestão de Desempenho

1st Renata Bandeira Gomes
Departamento de Engenharia Elétrica
Universidade Federal de Viçosa
Viçosa, Brasil
renata.bandeira@ufv.br

2nd Victor Pellanda Dardengo
Departamento de Engenharia Elétrica
Universidade Federal de Viçosa
Viçosa, Brasil
victor.dardengo@ufv.br

Resumo—Diante do crescimento exponencial do setor solar no Brasil, a maximização do desempenho de sistemas fotovoltaicos torna-se uma necessidade para justificar investimentos e otimizar o retorno energético e financeiro. Fatores como sombreamento, acúmulo de sujeira, defeitos e degradação dos painéis impactam diretamente a produção, demandando soluções eficazes de monitoramento e análise. Este trabalho apresenta o desenvolvimento de um dashboard interativo em Python, projetado para o monitoramento de usinas fotovoltaicas. A ferramenta possibilita análises comparativas entre diferentes usinas por meio de gráficos e tabelas, auxiliando na identificação de falhas, na avaliação do desempenho financeiro e na otimização das operações.

Palavras-chave—dashboard interativo, usinas fotovoltaicas, monitoramento, desempenho

I. INTRODUÇÃO

Em um século, a temperatura média da Terra registrou um aumento de cerca de 0,5°C. Projeções científicas sugerem que esse valor pode chegar a 4°C até o final do século atual [1]. Diante dessa condição, severas alterações climáticas podem ser desencadeadas e essas consequências tem se tornado uma preocupação mundial.

Uma das causas para esse desequilíbrio climático se justifica com a emissão de gases de efeito estufa (GEE) para a atmosfera. A maior parte das atividades humanas depende de uma energia proveniente da queima de combustíveis fósseis. Entende-se como exemplos desse tipo de fonte de energia o carvão mineral, os derivados de petróleo e o gás natural [1]. A queima desses combustíveis gera a emissão dos GEE para a atmosfera. Perante esse contexto, a transição energética surge como uma aliada no combate à emissão dos gases poluentes [2]. Se mostrando uma solução viável para a redução da dependência de combustíveis fósseis e mitigação dos impactos ambientais associados à geração de energia.

Entre as tecnologias que estão sendo utilizadas no processo da transição energética, está a tecnologia proveniente das células fotovoltaicas. Essa inovação baseia-se no efeito fotovoltaico, sendo este a conversão da luz solar em energia elétrica [3]. A eletricidade gerada através desse fenômeno também é conhecida, mais comumente, como energia solar fotovoltaica.

A energia renovável derivada do sol tem apresentado uma expansão notável no Brasil, destacando-se como uma das principais fontes sustentáveis em crescimento no país. Em 2023, a geração solar atingiu 50,6 TWh, representando um aumento de 68,1% em relação ao ano anterior, enquanto sua capacidade instalada alcançou 37.843 MW, registrando uma expansão de 54,8% [4]. A evolução da fonte solar no Brasil pode ser analisada através da Fig. 1. Essa trajetória de crescimento reforça o papel da energia solar como uma das protagonistas na diversificação e renovação da matriz energética brasileira.

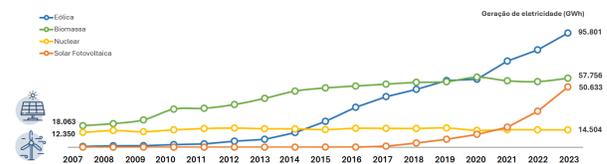


Fig. 1. Evolução histórica da geração de energia elétrica por fontes renováveis. [4].

No contexto global, o Brasil destaca-se como um dos principais líderes na incorporação de fontes renováveis à matriz energética, com ênfase na energia solar. Em 2022, essa diferença é evidenciada pela Fig. 2. O país apresenta uma taxa de renovabilidade de 86,1% na oferta interna de energia, valor significativamente superior à média mundial de 29,5% [5]. Essa discrepância expressiva reflete os esforços nacionais voltados à transição energética e à exploração do vasto potencial solar brasileiro, favorecido por elevados índices de radiação ao longo do ano [3].

A regulamentação do setor também desempenhou um papel crucial para o crescimento da energia solar no Brasil. Em 17 de abril de 2012, foi promulgada a Resolução Normativa ANEEL n° 482/2012, autorizando os consumidores a produzirem sua própria energia elétrica por meio de fontes renováveis ou cogeração qualificada [6]. Além disso, permitiu-se o envio do excedente à rede de distribuição, viabilizando a compensação futura com o consumo [7]. A partir dessa resolução, observou-

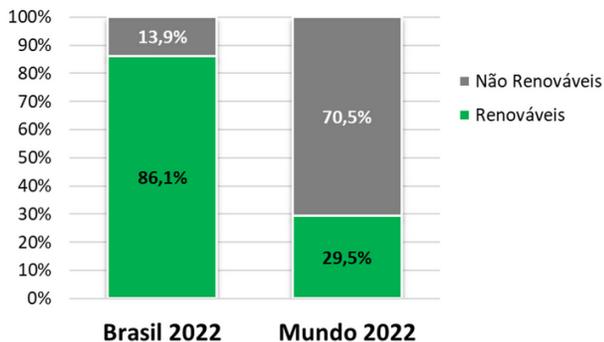


Fig. 2. Comparativo entre fontes renováveis e não renováveis na geração de energia elétrica no Brasil e no mundo em 2022 [5].

se um aumento na aquisição de sistemas solares, com um crescimento exponencial especialmente a partir de 2017, como mostrado nas Figuras 1 e 3.

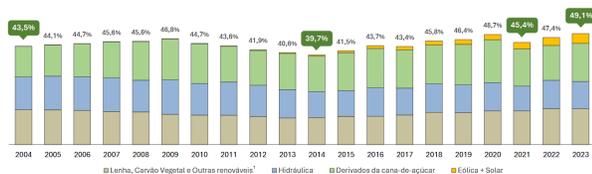


Fig. 3. Evolução da Energia Solar no Brasil de 2004 a 2023 [4].

Diante desse cenário de crescimento, a energia solar não apenas ganha espaço na matriz energética brasileira, mas também contribui para a sustentabilidade ambiental, a segurança energética e a redução de emissões de gases de efeito estufa. A Fig. 4 apresenta a matriz elétrica brasileira em 2023, com 89,2% de fontes renováveis, reforçando o papel crucial da energia solar nesse resultado [5].

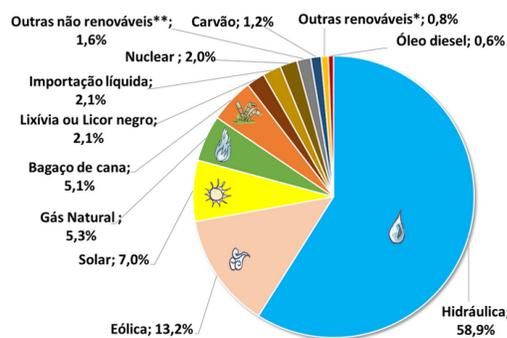


Fig. 4. Matriz energética brasileira em 2023 [5].

Tendo em vista essa tendência de crescimento ao longo dos anos, garantir o máximo desempenho dos sistemas fotovoltaicos tornou-se essencial para justificar os investimentos contínuos e maximizar o retorno energético e financeiro. Além disso, a variabilidade de fatores como radiação solar, condições climáticas e possíveis falhas no sistema pode im-

pactar diretamente na eficiência da geração de energia. Nesse cenário, o desenvolvimento de um sistema de monitoramento de desempenho surge como uma ferramenta indispensável, permitindo não apenas a identificação rápida de anomalias, mas também a análise detalhada da eficiência e da produtividade ao longo do tempo, promovendo uma gestão mais eficaz da geração solar [2].

Perante esse contexto, o presente trabalho tem como objetivo o desenvolvimento de uma dashboard de monitoramento de usinas fotovoltaicas. No qual será possível acompanhar o desempenho, diário, anual e mensal dos sistemas escolhidos, além disso realizar análises precisas sobre as falhas do sistema e o impacto financeiro gerado em suas ocorrências.

Por fim, a ferramenta possibilita a realização de comparações entre diferentes sistemas fotovoltaicos, permitindo analisar simultaneamente seus desempenhos e identificar oportunidades de melhoria, como ajustes operacionais ou ações de manutenção. Essa análise comparativa também destaca o impacto de fatores ambientais, como variações de radiação solar, temperatura e sombreamento, além de aspectos tecnológicos, especialmente em usinas que utilizam diferentes configurações ou equipamentos. Dessa forma, é possível avaliar qual tecnologia apresenta melhor desempenho e compreender como esses elementos influenciam a produção, contribuindo para a otimização dos sistemas e a maximização da eficiência energética.

II. METODOLOGIA

A partir do contexto previamente discutido, esta seção apresenta o embasamento teórico e as ferramentas utilizadas no desenvolvimento prático deste trabalho.

A. Estruturação e Construção do Sistema

1) *Aspectos Técnicos: Linguagem de Programação, Bibliotecas e Ferramentas:* Para a criação do dashboard, foi escolhida a linguagem de programação Python. Além de ser uma ferramenta gratuita e apresentar uma comunidade grande e ativa de computação científica e análise de dados [8], ela se destaca como uma das linguagens de programação mais utilizadas na análise e visualização de dados devido à sua simplicidade, versatilidade e extensiva biblioteca de ferramentas. Sua sintaxe intuitiva permite que profissionais de diversas áreas, mesmo com pouca experiência em programação, possam rapidamente aprender e aplicar técnicas de análise de dados.

Ao longo dos anos esse sistema de programação veio ganhando cada vez mais robustez e ao contrário de linguagens como R, Python é muito generalista [9]. Uma das principais vantagens de Python é a vasta coleção de bibliotecas específicas para manipulação e visualização de dados [8]. Bibliotecas como *Pandas* e *Polars* facilitam o tratamento e a análise de grandes volumes de informações, enquanto *Matplotlib* oferece recursos abrangentes para criar visualizações gráficas de alta qualidade. Além disso, ferramentas como *Plotly* e *Dash* permitem o desenvolvimento de dashboards interativos, tornando as análises mais dinâmicas e acessíveis [8].

Além de todas as outras vantagens já mencionadas, dashboards desenvolvidos em Python podem ser executados em qualquer sistema operacional, tornando a distribuição mais acessível. Neste trabalho, para a criação do painel de visualização e análise, as bibliotecas *Dash*, *Pandas*, *Polars* e *Plotly* foram fundamentais, enquanto notebooks e arquivos *.py* garantiram uma estrutura de código organizada e eficiente.

A biblioteca *Dash* foi utilizada como a base para o desenvolvimento do dashboard. Projetada para criar aplicações web interativas, o *Dash* combina Python com tecnologias como HTML e CSS, permitindo que desenvolvedores se concentrem na lógica e nos dados, sem a necessidade de conhecimentos avançados de *front-end*. Sua capacidade de integrar controles interativos, como dropdowns, sliders e gráficos dinâmicos, foi essencial para criar uma interface amigável e responsiva, permitindo que os usuários explorassem os dados de maneira intuitiva e livre.

Para realizar um tratamento e manipular os dados de entrada, garantindo a sua apresentação de forma clara e organizada, o *Pandas* desempenhou um papel crucial. Essa biblioteca, amplamente utilizada em ciência de dados, oferece ferramentas robustas para leitura, organização e transformação de DataFrames. Através dela, tornou-se possível realizar operações como agrupamentos, cálculos estatísticos e filtragem de informações, etapas fundamentais para que os dados pudessem ser visualizados de forma eficiente e otimizada no dashboard.

Nos casos em que se demandou uma performance maior, especialmente ao lidar com grandes volumes de informações, o *Polars* foi empregado como uma alternativa ao *Pandas*. Essa biblioteca, construída em Rust, é conhecida por sua alta velocidade e eficiência em operações complexas, proporcionando uma solução ideal para manipulação de grandes volumes de informações [10]. A combinação de *Pandas* e *Polars* garantiu flexibilidade no tratamento dos dados, equilibrando facilidade de uso e desempenho.

Para a criação de gráficos interativos e visualmente atraentes, o *Plotly* foi a ferramenta escolhida. Esse pacote possibilita a geração de mais de 40 gráficos distintos e personalizados em Python [11]. Totalmente integrado ao *Dash*, ele permitiu a construção de gráficos de linhas e barras com um alto grau de personalização. Além de sua estética refinada, a capacidade de interação dos gráficos, como zoom, seleção de dados e destaque de elementos, agrega um valor significativo à utilização e exploração do painel informativo por parte dos usuários.

O desenvolvimento também foi facilitado pela utilização de notebooks e arquivos *.py*. Os notebooks foram especialmente úteis nas etapas iniciais, e durante o desenvolvimento, quando era necessário explorar os dados, manipular DataFrames com a criação de novas colunas e testar soluções de forma incremental, com a vantagem de visualizar resultados instantaneamente. Posteriormente, o código final foi organizado em um arquivo *.py*, garantindo modularidade e reuso de funções, além de facilitar a manutenção e uma expansão futura do projeto.

Com a sinergia entre essas ferramentas, foi possível criar

um dashboard interativo, eficiente e altamente personalizável. Adequado para a análise e comparação de dados de usinas fotovoltaicas. A combinação de bibliotecas poderosas e uma estrutura de desenvolvimento bem definida exemplifica o potencial de Python na criação de soluções visuais que conectam dados e usuários de maneira prática e inovadora.

2) *Arquitetura do Dashboard*: O layout do dashboard é dividido em duas abas principais: "*Monitoring Dashboard*", na qual se concentra o painel principal, voltada para a visualização de gráficos e métricas, e "*Detailed Failure and Financial Analysis*", que oferece uma análise detalhada das falhas e impactos financeiros. O design da interface tem por objetivo adotar um estilo moderno com um esquema de cores escuras, proporcionando uma boa legibilidade com elementos contrastantes, como texto branco sobre fundo escuro.

A interface do painel principal foi projetada com base em quatro quadrantes principais, complementados por uma barra lateral à esquerda contendo quatro menus dropdowns, como pode ser observado através da Fig. 5. Os setores centrais apresentam um gráfico de linhas, um gráfico de barras e duas tabelas, cada uma acompanhada por um botão associado.

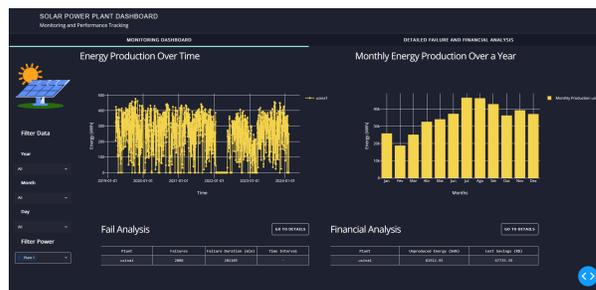


Fig. 5. Exibição padrão do dashboard.

O gráfico de linha, apresentado na Fig. 6, presente no primeiro quadrante do painel interativo, tem o objetivo de descrever a produção de potência e energia de uma usina fotovoltaica ao longo do tempo. Uma das maiores vantagens desse diagrama é a capacidade de mostrar tendências com clareza. Como as linhas conectam os pontos de dados sequenciais, torna-se fácil identificar padrões de produção, como o aumento gradual ao amanhecer, o pico ao meio-dia, e a queda ao entardecer, além de variações sazonais ou mudanças ao longo de dias e meses. Isso permite uma análise intuitiva do comportamento da usina.

Além disso, o gráfico de linha é uma excelente escolha para comparações temporais. Ao exibir múltiplas séries de dados em uma única visualização, como a produção em diferentes dias, semanas ou meses, é possível identificar variações significativas entre períodos, permitindo uma análise detalhada. Essa característica é particularmente útil para diagnósticos, já que anomalias, como quedas inesperadas de produção, tornam-se rapidamente visíveis e podem ser investigadas.

A segunda área principal da ferramenta de visualização exibe um gráfico de barras que apresenta, de forma exclusiva, a produção de uma ou mais usinas solares ao longo

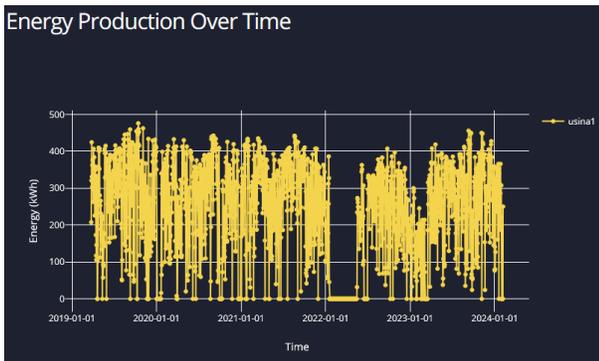


Fig. 6. Gráfico de linha presente no primeiro quadrante do dashboard.

dos anos, conforme ilustrado na Fig. 7. Diferentemente da visualização anterior, que permite uma análise em diferentes escalas temporais, como anos, meses ou dias, este gráfico foca em uma perspectiva anual, oferecendo uma visão consolidada da produção ao longo do tempo.

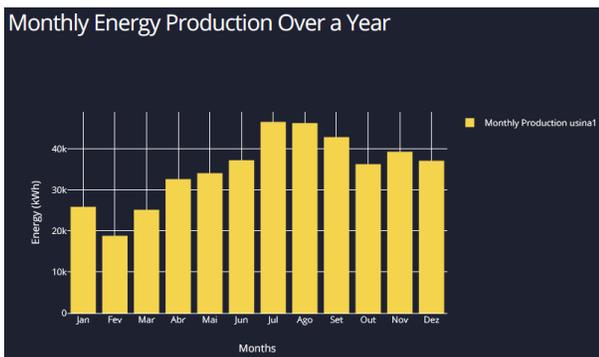


Fig. 7. Gráfico de barras localizado no segundo quadrante do dashboard.

Para expor a produção anual de uma usina solar, a ilustração em barras, oferece diversas vantagens. Esse tipo de visualização destaca comparações diretas entre meses, tornando evidente o crescimento, a estabilidade ou possíveis quedas na produção ao longo do ano. Além disso, as barras fornecem uma representação clara e intuitiva, facilitando a análise por parte de diferentes públicos, incluindo aqueles sem conhecimento técnico. Essa simplicidade visual é ideal para identificar rapidamente os meses de maior ou menor produtividade, permitindo uma avaliação eficiente do desempenho da usina em períodos mais longos.

A tabela, situada no terceiro quadrante do dashboard, é utilizada exclusivamente para exibir de forma clara e didática as falhas nos sistemas solares monitorados, conforme mostrado na Fig. 8. Entende-se por falhas os períodos de tempo, em diversas escalas temporais, em que a produção de energia se torna nula, ou seja, em que o sistema se encontra inoperante ou há uma falha na comunicação da usina com o sistema de monitoramento. Vale ressaltar que a produção considerada é limitada ao intervalo de 6:30 às 17:30h, de modo que a ausência de produção durante o período noturno não é

classificada como uma falha.

Plant	Failures	Failure Duration (min)	Time Interval
usina1	2008	202105	-

Fig. 8. Tabela de análise de falhas posicionada no terceiro quadrante do dashboard.

Junto à tabela de análise de falhas, há um botão projetado para exibir, de maneira detalhada, os erros mensais ou anuais, caso o usuário deseje aprofundar a análise. Ao ser acionado, o botão redireciona o usuário para a aba secundária, "Detailed Failure and Financial Analysis". Nesta seção é possível verificar as informações detalhadas sobre as intercorrências mensais ou anuais. Essa funcionalidade foi implementada com o objetivo de fornecer ao usuário uma visão mais abrangente das falhas em um intervalo de tempo maior, sem sobrecarregar o dashboard principal com excesso de informações.

Na quarta seção principal do painel, está localizada uma tabela dedicada à análise financeira que pode ser observada através da Fig.9. Essa tabela apresenta, com base nas falhas identificadas, a quantidade de energia não produzida, expressa em kWh, e o valor monetário correspondente à economia não alcançada, expresso em reais. Além disso, a tabela conta com um botão associado, que direciona o usuário para a aba secundária da ferramenta, permitindo uma análise mais detalhada dos impactos financeiros ao longo de períodos de tempo mais extensos.

Plant	Unproduced Energy (kWh)	Lost Savings (R\$)
usina1	81912.95	67735.39

Fig. 9. Tabela de análise financeira situada no terceiro quadrante do dashboard.

Por fim, os dropdowns na barra lateral esquerda têm como objetivo permitir que o usuário escolha a quantidade de usinas a ser analisada simultaneamente, além de definir o período de tempo a ser monitorado. Assim, as seleções feitas nos menus suspensos controlam todas as outras exibições principais do dashboard.

B. Processamento e Análise de Dados

1) *Coleta de Dados*: Inicialmente, os dados utilizados nesse estudo, foram obtidos através de um *web scraping*. Esse processo corresponde à extração de dados de sites da internet de forma automatizada, utilizando ferramentas ou scripts para acessar o conteúdo das páginas e coletar informações estruturadas.

A obtenção dos dados foi realizada através da plataforma web GDASH. Uma solução digital desenvolvida para apoiar integradores solares na otimização da tomada de decisões na

gestão de ativos de energia renovável, utilizando análise e inteligência de dados [12].

Foi desenvolvido um código em linguagem Python com o objetivo de realizar a extração de dados diários de uma usina fotovoltaica específica. O sistema analisado é constituído por 228 módulos fotovoltaicos de 340 W cada, totalizando uma potência instalada de 77,52 kWp, e conta com um inversor de 60 kW.

A instalação está localizada no município de Caraíva, no estado de Minas Gerais. A operação da usina teve início em março de 2019, enquanto o monitoramento de dados começou em 25 de março de 2019. A última data considerada para a extração dos dados foi 6 de fevereiro de 2024.

O script foi desenvolvido para realizar o login automatizado na plataforma, acessar a usina específica, selecionar um dia por vez e, em seguida, baixar automaticamente um arquivo .csv diretamente do site. O formato do arquivo e as informações extraídas podem ser observados na Tabela 1.

TABELA I
REPRESENTAÇÃO DE ARQUIVO .CSV EXTRAÍDO NO WEB SCRAPING.

Data	Hora	Potência (kW)
01-01-2023	07:20	7.576
01-01-2023	07:25	7.796
01-01-2023	07:30	7.038

Conforme apresentado na Tabela 1, os dados foram coletados diariamente e incluem informações referentes à data, hora e geração de potência instantânea. Observa-se que a potência foi registrada em intervalos de 5 minutos, garantindo uma amostragem regular e detalhada. Ao todo, foram extraídos 1.547 arquivos no formato .csv, totalizando um volume de 9,87 MB de dados.

2) *Banco de Dados*: Após a extração dos dados, foi necessário construir uma base de dados unificada e bem estruturada. Para isso, foi desenvolvido um novo script em Python, executado em um ambiente Jupyter Notebook. Esse código realizou a integração de todos os arquivos relacionados à usina, resultando na construção de uma base de dados consistente e consolidada.

Além disso, para preparar essa base para aplicações no dashboard, tornou-se essencial a adição de mais uma coluna. Sendo esta a de Energia (kWh). Para isto, partiu-se da operação mostrada na Equação 1.

$$\text{Energia} = \text{Potência} \times \text{Tempo} \quad (1)$$

Com isso, a partir da coluna "Potência (kW)", foi gerada a coluna "Energia (kWh)". Como as medições de potência acontecem diariamente a cada 5 minutos, para calcular a energia o tempo considerado foi igual 5/60. Logo, a operação foi feita conforme a Equação 2.

$$\text{Energia} = \text{Potência} \times \frac{5}{60} \quad (2)$$

Essas duas grandezas são essenciais para compreender o consumo e a geração de energia elétrica. A potência, medida

em quilowatts (kW), representa a taxa de consumo ou geração de energia em um dado momento. Por outro lado, a energia, expressa em quilowatt-hora (kWh), indica a quantidade total de energia consumida ou gerada ao longo de um período de tempo. A base de dados, com a adição da nova coluna, pode ser visualizada na Tabela 2.

TABELA II
REPRESENTAÇÃO DA BASE DE DADOS DEPOIS DA ADIÇÃO DA COLUNA "ENERGIA (KWH)".

Data	Hora	Potência (kW)	Energia (kWh)
2020-01-01	05:15	0.22	0.018
2020-01-01	05:20	0.371	0.031
2020-01-01	05:25	0.523	0.523

Tendo em vista que um dos objetivos deste dashboard é possibilitar a comparação entre dois sistemas distintos, para atender a esse propósito, foi criada uma usina fotovoltaica fictícia baseada no sistema real. A criação dessa usina foi realizada aplicando-se um fator multiplicativo de 1,4 às colunas "Potência (kW)" e "Energia (kWh)" da base de dados original. Esse procedimento resultou em uma nova base de dados com um incremento de 40% nos valores originais, permitindo que as diferenças entre os sistemas fossem visualizadas de forma mais clara na interface a ser desenvolvida.

Após esse processo, os dois conjuntos de dados foram integrados em um único banco de informações. Para diferenciar os sistemas, foi criada uma nova coluna denominada "Usina", na qual o sistema real foi identificado como "Usina 1" e o sistema fictício como "Usina 2".

Observou-se que, durante falhas na produção de energia, os dados correspondentes a essas ocorrências não eram registrados, resultando em lacunas na base de dados. Para solucionar esse problema e prevenir inconsistências futuras na análise, foi desenvolvido um script adicional capaz de identificar e tratar essas ausências de forma eficiente.

A Fig. 10 ilustra como o gráfico de linhas apresentava os dados antes do tratamento das falhas. É importante notar que, embora as falhas tenham ocorrido, a ausência de contabilização na base de dados impediu que o gráfico indicasse produção zerada durante esses períodos. Em vez disso, o gráfico exibia apenas o ponto correspondente ao último dia de produção antes da falha e o primeiro dia de retomada, sem evidenciar o intervalo em que as falhas ocorreram.

A elaboração do código baseou-se na premissa de que o sistema opera, em média, das 6h30 às 17h30. O objetivo do programa foi preencher dados ausentes dentro desse intervalo, considerando a frequência de registros a cada 5 minutos. Quando identificadas lacunas nesse período, os dados faltantes foram preenchidos com o valor '0' nas colunas "Potência (kW)" e "Energia (kWh)", representando falhas na produção.

Após todo o tratamento realizado, a base de dados foi preparada para utilização na construção do dashboard. A Tabela 3 apresenta a estrutura dela no momento em que o desenvolvimento do código do dashboard foi iniciado.

3) *Pré-Processamento dos Dados*: No processo de desenvolvimento do algoritmo para o dashboard, identificou-

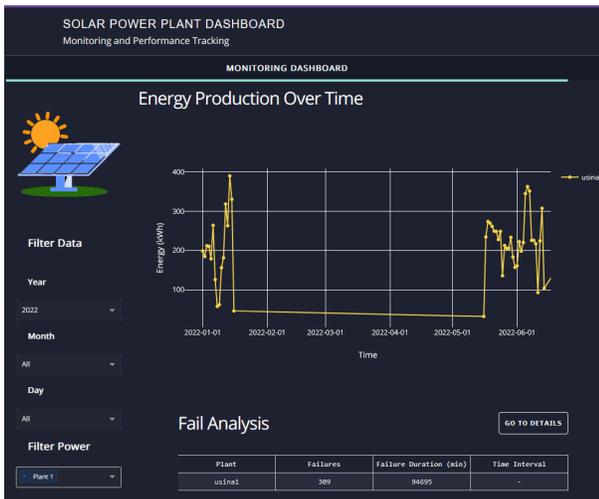


Fig. 10. Painel de monitoramento antes do tratamento das falhas.

TABELA III
REPRESENTAÇÃO DA ESTRUTURA DA BASE DE DADOS ANTES DO DESENVOLVIMENTO DO DASHBOARD.

Data	Hora	Potência (kW)	Energia (kWh)	Usina
2019-03-25	06:30	0.0	0.0	usina2
2019-03-25	06:30	0.0	0.0	usina1
2019-03-25	06:35	0.0	0.0	usina2

se a necessidade de realizar manipulações adicionais nos dados. Para viabilizar uma análise financeira mais precisa, foi necessário criar uma "curva média" de potência específica para cada usina. Essa curva serve como referência para estimar a quantidade de energia que deveria ter sido produzida em períodos de falha, permitindo uma análise mais detalhada das perdas de geração associadas a esses eventos.

Para isso, um novo programa utilizando o ambiente Jupyter Notebook foi elaborado. O código em questão realiza cálculos e manipulações para analisar dados de potência gerada pelas duas usinas fotovoltaicas analisadas. Primeiramente, agrupa os dados por horário e usina para calcular a potência média gerada em cada combinação, inserindo esses valores no DataFrame original.

Além disso, o script calcula a energia média gerada em intervalos de 5 minutos com base na potência média, adicionando essa métrica como uma nova coluna no DataFrame. Por fim, os dados processados são exportados para um arquivo CSV, formando assim uma nova base de dados com duas novas colunas que pode ser observada através da Fig. 11.

usina	data	Hora	Potência (kW)	Energia (kWh)	potencia_media	energia_media
usina2	2019-03-25	06:30:00	0.0	0.0	3.21	2.67
usina1	2019-03-25	06:30:00	0.0	0.0	1.83	1.53
usina2	2019-03-25	06:35:00	0.0	0.0	3.78	3.15
usina1	2019-03-25	06:35:00	0.0	0.0	2.16	1.80
usina2	2019-03-25	06:40:00	0.0	0.0	4.38	3.65

Fig. 11. Representação da nova base de dados durante o desenvolvimento do dashboard.

Outro tratamento precisou ser feito para construir as interfaces para análise de falhas e financeira. Para identificar as

falhas de maneira mais ágil, um novo dataframe, este chamado de "failures_financial_df" foi criado.

Sendo assim, outro algoritmo foi criado utilizando o ambiente Jupyter Notebook. O programa, em questão, analisa falhas operacionais em usinas fotovoltaicas, identificando períodos sem geração de energia e calculando as perdas econômicas associadas. Primeiro, a base de dados total é carregada de um arquivo csv, com conversão e manipulação das colunas de data e hora, além da criação de colunas auxiliares como ano, mês e dia. Em seguida, o dataset é filtrado para considerar apenas o horário operacional de 06h30min da manhã até às 17h30min da tarde.

Para cada usina e dia, o código detecta períodos de falha quando a potência ou a energia gerada são iguais a zero. Ele registra o início, fim e duração das falhas, calculando a energia que deixou de ser produzida nesses intervalos. Com base no ano do registro, aplica-se uma tarifa de energia específica para determinar a perda financeira associada. Os valores adotados para a tarifa variam conforme o ano: 0,70 para 2019, 0,75 para 2020, 0,80 para 2021, 0,85 para 2022, 0,90 para 2023 e 0,95 para 2024.

Os resultados, contendo detalhes sobre falhas, energia não produzida e perdas econômicas, são consolidados em um arquivo .csv final que será carregado no código principal do dashboard. Uma representação dele pode ser observada através da Fig. 12. Por fim, a Fig. 13 exibe um fluxograma detalhado de todo o processamento de dados realizado para confecção do painel de monitoramento.

usina	dia	mes	ano	data_inicio	data_fim	Tempo da Falha (min)	Intervalo de Horário	energia_nao_produzida	tarifa	economia_perdida
usina2	25	3	2019	2019-03-25 06:30:00	2019-03-25 11:05:00	275.0	06:30:00 - 11:05:00	157.84	0.7	110.49
usina2	25	3	2019	2019-03-25 11:20:00	2019-03-25 12:40:00	80.0	11:20:00 - 12:40:00	50.7	0.7	35.59
usina2	26	3	2019	2019-03-26 13:25:00	2019-03-26 16:00:00	158.0	13:25:00 - 16:00:00	138.35	0.7	96.45
usina2	26	3	2019	2019-03-26 14:20:00	2019-03-26 14:25:00	5.0	14:20:00 - 14:25:00	4.91	0.7	3.44
usina2	3	4	2019	2019-04-03 10:00:00	2019-04-03 10:10:00	10.0	10:00:00 - 10:10:00	9.25	0.7	6.48

Fig. 12. Representação do novo DataFrame "failures_financial_df".



Fig. 13. Fluxograma do tratamento de dados.

4) Implementação de Interfaces Gráficas e Visualizações:

A interface do dashboard foi construída com base em um template disponível no site *Dash Enterprise App Gallery*, uma plataforma fornecida pela *Plotly*. Essa plataforma oferece uma coleção de aplicativos e templates prontos, desenvolvidos com o framework *Dash*, os quais servem como pontos de partida para a personalização e implementação de interfaces gráficas interativas com visualizações complexas de dados [13].

Com base nesse template, a aplicação desenvolvida neste trabalho inicia sua construção com a criação de um objeto *Dash*. Feito isso, os arquivos .csv contendo as bases de dados são carregados utilizando o *Pandas*. As colunas de data e hora são convertidas para o formato adequado, e novas colunas, como ano, mês e dia, são extraídas das datas para facilitar a filtragem dos dados. Além disso, valores duplicados são

eliminados, garantindo que as informações exibidas sejam únicas e precisas.

A construção da interface é enriquecida por diversos componentes interativos, como os menus suspensos, que permitem ao usuário filtrar os dados exibidos de maneira flexível. São utilizados quatro listas suspensas: uma para selecionar o ano, outra para escolher o mês, uma terceira para o dia, e uma exclusiva para definir a quantidade de usinas a serem monitoradas. Importante destacar que as escolhas feitas nesses dropdowns impactam diretamente todo o dashboard, pois determinam tanto o intervalo temporal da análise quanto o número de usinas a serem acompanhadas simultaneamente.

Na região central da interface de controle, estão posicionados dois gráficos e duas tabelas. O primeiro gráfico, localizado no primeiro quadrante, é um gráfico de linha. O eixo vertical desse gráfico exibe os valores de 'Potência (kW)' ou 'Energia (kWh)', dependendo da seleção feita nos dropdowns. Se o usuário selecionar o intervalo de tempo diário, o gráfico exibirá a potência instantânea, enquanto para intervalos de tempo maiores, como mês ou ano, será exibida a energia acumulada. O eixo horizontal representa a dimensão temporal, variando entre dias, meses ou anos, conforme a seleção do usuário. Em resumo, esse gráfico mostra a produção de energia ao longo do tempo, ajustada conforme o intervalo temporal e a quantidade de sistemas selecionados.

A função `"update_line_chart"` foi criada com o objetivo de atualizar o gráfico de linhas e as duas tabelas de análise, baseando-se nos filtros selecionados pelo usuário, como usinas, ano, mês e dia. O primeiro passo da função consiste na aplicação de filtros sobre o DataFrame `"failures_financial_df"`, ajustando os dados de acordo com os parâmetros fornecidos. Esses filtros permitem que o usuário selecione um conjunto específico de dados de produção e falhas, por exemplo, de uma usina ou para um ano e mês determinados. A seguir, a função verifica se a data foi completamente selecionada, o que determina a forma de agrupamento dos dados.

Em caso de seleção incompleta de data, os dados de falhas são agrupados por usina e são somadas a duração das falhas e o número de ocorrências. Além disso, a função agrupa também os dados financeiros, calculando a energia não produzida e a economia perdida para cada usina. Quando todos os componentes da data, ano, mês e dia, são selecionados, os dados são apresentados sem agrupamento, permitindo uma análise mais detalhada. Esse agrupamento flexível permite ao usuário visualizar tanto dados gerais quanto específicos, dependendo da granularidade da seleção.

A tabela de análise de falhas apresenta informações cruciais sobre as intercorrências ocorridas em cada usina. As colunas da tabela incluem a usina, o número de falhas, calculado a partir dos dados filtrados, o tempo total de falha em minutos e o intervalo de horário em que as falhas ocorreram. Se a data não for completamente selecionada, os dados são agregados por usina e as falhas são somadas, fornecendo uma visão geral das falhas. Caso o usuário tenha selecionado uma data específica, a tabela é exibida com os registros detalhados das

falhas, permitindo uma análise mais precisa.

A tabela de análise financeira fornece informações sobre o impacto financeiro das falhas. As colunas incluem a usina, a energia não produzida, em kWh, e a economia perdida, em reais. Assim como na tabela de falhas, quando a data não é completamente selecionada, os dados financeiros são agrupados por usina, permitindo uma visão agregada. Para seleções de data mais específicas, os valores financeiros são apresentados de forma individualizada. As tabelas são estilizadas para proporcionar uma experiência visual agradável, com uma formatação que facilita a leitura e a interpretação dos dados pelo usuário e que não sobrecarregue visualmente a interface do painel de monitoramento como um todo.

A função `"update_histogram_monthly"` foi desenvolvida para gerar e atualizar o gráfico de barras que mostra a produção de energia mensal das usinas selecionadas durante um ano específico. Ela começa filtrando o DataFrame global `"df"` de acordo com as usinas e o ano escolhidos. Caso a seleção do menu suspenso "Ano" seja "All", todos os dados são utilizados. O DataFrame filtrado é então agrupado por mês e usina, somando a energia produzida em kWh.

O gráfico de barras é construído para cada usina selecionada, utilizando a produção mensal de energia. As cores das barras são alternadas entre as usinas, com uma paleta predefinida no código. O gráfico é configurado para exibir as barras lado a lado, o que facilita a comparação entre as usinas para cada mês. As colunas do eixo horizontal representam os meses do ano, de janeiro a dezembro, e o eixo vertical mostra a energia total produzida em kWh.

O layout do gráfico é personalizado para garantir uma boa visualização. As legendas são exibidas para identificar as usinas de acordo com a seleção, e os rótulos dos meses são apresentados de forma clara no eixo horizontal. A função retorna a figura final do gráfico, que é utilizada no painel principal.

III. RESULTADOS

A Fig. 5 apresentou a interface principal do dashboard. Observa-se que a ferramenta está localizada na aba denominada `"Monitoring Dashboard"`, com todos os menus suspensos relacionados ao tempo configurados como "All". Esta exibição foi definida como a configuração padrão, de modo que, ao inicializar a aplicação, esta tela é exibida.

A Fig. 14 demonstra a mesma configuração ilustrada na Fig. 5, com a diferença de que, nesta instância, foram selecionadas duas usinas na lista suspensa `"Filter Power"`. Ressalta-se que essa seleção impactou toda a interface de controle, resultando na atualização dos gráficos e das tabelas para exibirem os dados referentes às usinas 1 e 2.

Através da Fig. 15, é possível notar o comportamento do painel quando é selecionada uma data completa. Nesse exemplo, os menus suspensos foram configurados para analisar o dia 01/01/2024. Nesse período, foram registradas duas falhas, as quais estão representadas no gráfico de barras e em ambas as tabelas. Uma das tabelas destaca de forma mais evidente o momento e a duração das falhas, enquanto a outra enfatiza os



Fig. 14. Interface de controle com duas usinas selecionadas.



Fig. 17. Representação de uma dia sem falhas na produção.

impactos financeiros associados. A Fig. 16 apresenta o mesmo dia, porém com a seleção de duas usinas.

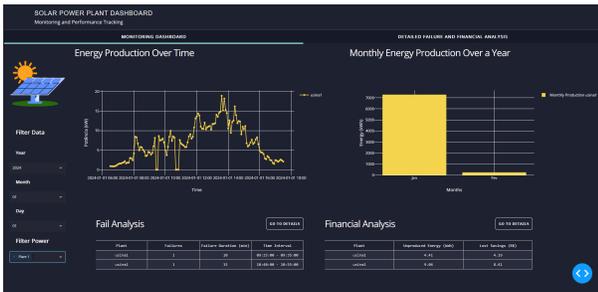


Fig. 15. Exemplo da seleção de uma data completa com falhas.

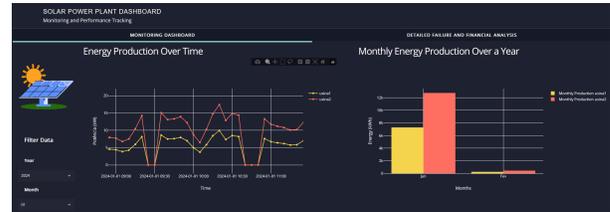


Fig. 18. Gráfico de linhas com zoom aplicado.

A Fig. 19 mostra a funcionalidade "Box Select" aplicada ao gráfico de barras. Essa ferramenta interativa permite selecionar um subconjunto de dados em um gráfico usando uma caixa retangular desenhada com o mouse. Essa seleção possibilita uma análise visual melhor e sem a necessidade de realizar filtros manuais.



Fig. 16. Exemplo da seleção de uma data completa com falhas para dois sistemas.

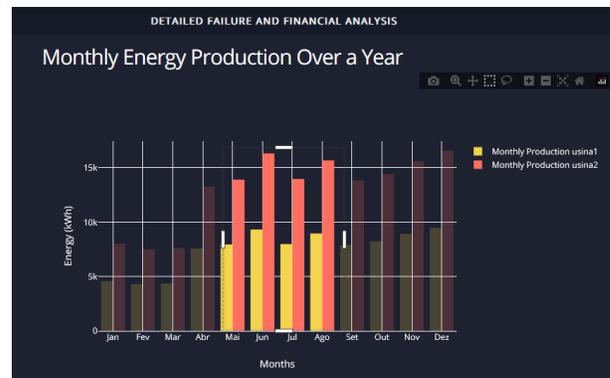


Fig. 19. Gráfico de barras com o "Box Selection" aplicado.

Um dia sem falhas, exemplificando uma curva que demonstra uma produção constante e esperada de um sistema solar, é exibido na Fig. 17. Nota-se que ambas as tabelas estão vazias, uma vez que sua exibição de dados está condicionada à ocorrência de intercorrências nas usinas.

A Figuras 18 e 19 ilustram algumas das funcionalidades oferecidas pela biblioteca *Plotly* para a construção e interação com gráficos. No exemplo da Fig. 18 foi aplicado um zoom no gráfico de linhas, permitindo uma visualização mais detalhada e precisa dos momentos em que ocorreram falhas. Essa funcionalidade é especialmente útil para análises mais refinadas, pois possibilita ao usuário explorar áreas específicas do gráfico, identificar padrões ou anomalias com maior clareza e melhorar a compreensão do comportamento do sistema em intervalos específicos.

Agora, ao realizar uma análise mensal, observa-se que o comportamento da interface de controle se altera. As tabelas exibem informações mais condensadas, apresentando de forma agrupada a quantidade de falhas, a energia não produzida e a economia perdida. A Fig. 20 expõe esse comportamento.

Para uma análise anual, nota-se que o comportamento da interface de controle apresenta características semelhantes. As tabelas consolidam ainda mais as informações, exibindo de forma agrupada o total de falhas ocorridas ao longo do ano, a energia acumulada que deixou de ser produzida e o montante total da economia não alcançada. A Fig. 21 ilustra esse comportamento, evidenciando como a interface adapta os



Fig. 20. Análise da produção mensal.

dados para proporcionar uma visão mais ampla e estratégica do desempenho do sistema ao longo do período anual.



Fig. 21. Análise da produção anual.

Observa-se que o ano representado na Fig. 21 apresentou um número significativo de falhas, incluindo períodos de total inatividade durante três meses consecutivos. Para uma análise mais detalhada das intercorrências, o usuário pode selecionar qualquer um dos botões vinculados às tabelas, que o direcionará para a aba secundária "Detailed Failure and Financial Analysis". A interface dessa tela é apresentada na Fig. 22.

Fault	Incidents	Failure Duration (min)	Time Interval	Operational Energy (kWh)	Cost Savings (R\$)
Operational	10	120	2022-12-01	10000	1000
Operational	15	180	2022-12-02	15000	1500
Operational	20	240	2022-12-03	20000	2000
Operational	25	300	2022-12-04	25000	2500
Operational	30	360	2022-12-05	30000	3000
Operational	35	420	2022-12-06	35000	3500
Operational	40	480	2022-12-07	40000	4000
Operational	45	540	2022-12-08	45000	4500
Operational	50	600	2022-12-09	50000	5000
Operational	55	660	2022-12-10	55000	5500
Operational	60	720	2022-12-11	60000	6000
Operational	65	780	2022-12-12	65000	6500
Operational	70	840	2022-12-13	70000	7000
Operational	75	900	2022-12-14	75000	7500
Operational	80	960	2022-12-15	80000	8000
Operational	85	1020	2022-12-16	85000	8500
Operational	90	1080	2022-12-17	90000	9000
Operational	95	1140	2022-12-18	95000	9500
Operational	100	1200	2022-12-19	100000	10000
Operational	105	1260	2022-12-20	105000	10500
Operational	110	1320	2022-12-21	110000	11000
Operational	115	1380	2022-12-22	115000	11500
Operational	120	1440	2022-12-23	120000	12000
Operational	125	1500	2022-12-24	125000	12500
Operational	130	1560	2022-12-25	130000	13000
Operational	135	1620	2022-12-26	135000	13500
Operational	140	1680	2022-12-27	140000	14000
Operational	145	1740	2022-12-28	145000	14500
Operational	150	1800	2022-12-29	150000	15000
Operational	155	1860	2022-12-30	155000	15500
Operational	160	1920	2022-12-31	160000	16000

Fig. 22. Interface da aba "Detailed Failure and Financial Analysis".

Nesta aba, encontram-se dois menus adicionais que permitem ao usuário selecionar o mês e o ano a serem analisados. Com base nas opções escolhidas, é gerada uma tabela detalhada contendo informações sobre todas as falhas diárias ocorridas no período selecionado, incluindo a duração de cada falha e seus respectivos impactos financeiros. A falha com maior duração no mês selecionado, ou seja, o dia em que o tempo de inatividade foi mais extenso, é destacada visualmente em vermelho. Esse destaque facilita a rápida identificação da falha mais crítica, permitindo ao usuário concentrar sua análise nos eventos de maior impacto.

Após a apresentação das principais funcionalidades do dashboard, é importante destacar que, devido à sua estrutura inicial baseada em um conjunto de dados com apenas três colunas, o projeto demonstra uma alta escalabilidade. Sua construção, a partir de informações simples, permite que ele seja facilmente adaptado e expandido para usinas com históricos de dados maiores, bem como para um número elevado de usinas, viabilizando sua aplicação em diferentes cenários e ampliando suas possibilidades de uso.

IV. CONCLUSÃO

O desenvolvimento deste trabalho teve como objetivo a construção de um dashboard interativo para monitoramento de usinas fotovoltaicas, utilizando a linguagem de programação Python. O contexto histórico apresentado mostrou a evolução da energia solar no Brasil, desde seu início até sua atual expansão exponencial, ressaltando a relevância desse tipo de energia no cenário energético nacional.

O dashboard desenvolvido se destacou pela sua capacidade de análise comparativa simultânea de diferentes usinas fotovoltaicas, proporcionando uma ferramenta eficaz para o monitoramento e gestão desses sistemas. A integração de gráficos de linha e barra, juntamente com tabelas de análise de falhas e financeira, permitiu uma visualização clara e detalhada dos dados coletados, oferecendo insights valiosos para a tomada de decisões.

A simplicidade e eficácia do dashboard, construído a partir de dados básicos como data, hora e potência (kW) produzida diariamente, demonstraram o potencial das ferramentas de visualização de dados para transformar informações brutas em análises significativas. Um dos pontos fortes deste projeto é sua escalabilidade. Tendo partido de informações simples, o dashboard pode ser expandido para incluir novos dados e funcionalidades. Essa característica garante que a ferramenta possa evoluir e se adaptar às necessidades crescentes do setor de energia solar.

Para trabalhos futuros, algumas sugestões incluem a adição de novos indicadores de desempenho, como impacto ambiental, custo-benefício e comparações com outras fontes de energia renovável. Além disso, a previsão de produção de energia utilizando técnicas de aprendizado de máquina para prever a produção futura com base em dados históricos e condições meteorológicas seria de grande valia para complementar as informações já existentes no painel.

Assim, o desenvolvimento deste projeto mostrou-se um passo significativo na direção de uma maior eficiência e eficácia no monitoramento de usinas fotovoltaicas, reforçando o papel da tecnologia e da ciência de dados na promoção de práticas mais sustentáveis e inovadoras no setor energético.

REFERÊNCIAS

- [1] Empresa de Pesquisa Energética, "Mudanças Climáticas e Transição Energética," disponível em: <https://www.epe.gov.br/pt/abcedenergia/clima-energia>, acesso em: 21 jan. 2025.
- [2] M. Andrade, "Desenvolvimento de um sistema de baixo custo para aquisição de dados e monitoramento web de usinas fotovoltaicas," Ifg.edu.br, 2022.

- [3] R. R. Biggi, "O uso da luz solar como fonte de energia elétrica através de sistema fotovoltaico - SF," Lavras, 2013.
- [4] Empresa de Pesquisa Energética, "Relatório Síntese 2024," disponível em: <https://www.epe.gov.br/pt>, acesso em: 14 jan. 2025.
- [5] Empresa de Pesquisa Energética, "Matriz Energética," disponível em: <https://www.epe.gov.br/pt/abcdenergia/matriz-energetica-e-eletrica>, acesso em: 21 jan. 2025.
- [6] Agência Nacional de Energia Elétrica, "Geração Distribuída," disponível em: <https://www.gov.br/aneel/pt-br/assuntos/geracao-distribuida>, acesso em: 21 jan. 2025.
- [7] L. G. Rocha, M. O. Medeiros, e D. Diego, "Modelagem da previsão de geração de energia solar e análise econômica da implantação de painéis fotovoltaicos no IFNMG Campus Teófilo Otoni," *Brazilian Journal of Production Engineering*, pp. 70–84, 28 maio 2021.
- [8] W. McKinney, *Python for Data Analysis*. O'Reilly Media, 2017.
- [9] Silva e Silva, "Linguagem de Programação Python," *Tecnologias em Projeção*, vol. 10, no. 1, pp. 55–71, 2019.
- [10] R. Ribeiro, "Manipulando dados com Pandas, Polars e PySpark," Medium, disponível em: https://medium.com/@rafaelrlima_54619/manipulando-dados-com-pandas-polars-e-pyspark-b1c444e95815, acesso em: 13 jan. 2025.
- [11] G. Datt, "Visualização de Dados Interativa com Plotly: Guia Prático em Python," Medium, disponível em: <https://medium.com/@guilhermedatt/visualiza%C3%A7%C3%A3o-de-dados-interativa-com-plotly-guia-pr%C3%A1tico-em-python-f46cef6065f2>, acesso em: 13 jan. 2025.
- [12] GDASH, "Sobre Nós," disponível em: <https://gdash.io/sobre/>, acesso em: 13 jan. 2025.
- [13] Plotly, "Dash Gallery - Portal," disponível em: <https://dash.gallery/Portal/?search=>, acesso em: 21 jan. 2025.