



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

ESTUDO COMPARATIVO ENTRE MÉTODOS DE MEDIÇÃO DA IRRADIÂNCIA SOLAR DIFUSA APLICADOS AO LEVANTAMENTO DE RECURSO SOLAR EM NATAL (RN)

João Vitor Batista Silva

Relatório de Iniciação
Científica do programa
PIBIC, orientado pelo Dr.
André Rodrigues
Gonçalves e coorientado
pelo Dr. George Santos
Marinho.

URL do documento original:

INPE
Natal
2025

Silva, João.

Estudo comparativo entre métodos de medição da irradiância solar difusa aplicados ao levantamento de recurso solar em Natal (RN)

João Vitor Batista Silva - Natal: INPE, 2025.

Relatório de Iniciação Científica -
Instituto Nacional de Pesquisas Espaciais, Natal, 2025.
Orientador: André Rodrigues Gonçalves.

1. Energia solar. 2. Instrumentação Meteorológica. 3. Qualificação de dados.

I. Estudo comparativo entre métodos de medição da irradiância solar difusa aplicados ao levantamento de recurso solar em Natal (RN)

CDU

Internacionais de Catalogação na Publicação



Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.
This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License

“Portanto, quer comais quer bebais, ou façais qualquer outra coisa, fazei tudo para glória de Deus.”

1Coríntios 10:31 - Bíblia Sagrada ACF

Primeiramente a Deus, à minha família, aos amigos, à minha namorada, que estiveram ao meu lado me apoiando, e aos colegas de trabalho, por toda ajuda e incentivo.

AGRADECIMENTOS

Agradeço ao meu orientador Dr. André Rodrigues Gonçalves e ao coordenador pelo Dr. George Santos Marinho, agradeço também pelos ensinamentos ao Dr. Francisco Raimundo da Silva, pela grandiosa orientação e ensinamentos que me proporcionaram. Também aos técnicos Tércio e Edmilson pela grande bagagem que me foi repassada, ao meu colega de laboratório e de curso Renan, a toda equipe do INPE-COENE pela cooperação e ajuda ao longo do projeto e ao CNPq/CAPES pelo financiamento da bolsa.

RESUMO

O presente trabalho tem como objetivo validar e analisar dados de irradiância solar global (GHI) e difusa (DHI) dos meses de junho a Dezembro de 2024, obtidos pela estação solarimétrica da rede SONDA, operada pelo LAVAT/INPE em Natal-RN, com foco na consistência e confiabilidade das medições utilizadas em estudos climáticos e aplicações em energia solar. Para isso, foram utilizados protocolos de validação recomendados pelo CGCT/INPE, baseados nos critérios da rede BSRN, aplicando algoritmos específicos para variáveis radiométricas. O processamento dos dados foi realizado em Python por meio da plataforma Google Colab, com leitura direta dos arquivos hospedados em nuvem. As análises envolveram testes de qualidade física dos dados, identificação de inconsistências e substituição de valores inválidos por NaN, além da criação de colunas corrigidas. Foram estudadas as relações entre a fração difusa (K_d) e o índice de claridade (K_t), com aplicação de filtros baseados no ângulo zenital e na irradiância mínima. Também foi feita uma análise de dispersão para detecção de valores suspeitos e avaliação horária da distribuição da DHI. O trabalho ainda descreve a disposição dos equipamentos da estação e os principais problemas técnicos enfrentados, como falhas na conexão de rede, domos sujos, e o rastreador solar SOLYS2 inoperante, que impactaram a coleta de dados. Os resultados indicam boa qualidade nas medições de GHI e a necessidade de correções mais rigorosas nos dados de DHI. Este estudo contribui para o aprimoramento do controle de qualidade dos dados radiométricos e reforça a importância da manutenção adequada dos equipamentos para garantir a continuidade e a confiabilidade das séries temporais de irradiância solar.

Palavras-chave: 1- Energia solar. 2- Instrumentação Meteorológica. 3- Irradiância difusa.

ABSTRACT

The present work aims to validate and analyze global horizontal irradiance (GHI) and diffuse horizontal irradiance (DHI) data from June to December 2024, collected by the solarimetric station of the SONDA network, operated by LAVAT/INPE in Natal-RN, focusing on the consistency and reliability of the measurements used in climate studies and solar energy applications. For this purpose, validation protocols recommended by CCST/INPE were applied, based on the BSRN network criteria, using specific algorithms for radiometric variables. Data processing was carried out using Python on the Google Colab platform, with direct access to files hosted in the cloud. The analyses involved physical quality tests, identification of inconsistencies, and replacement of invalid values with NaN, in addition to the creation of corrected columns. The relationship between the diffuse fraction (K_d) and the clearness index (K_t) was studied, applying filters based on the solar zenith angle and minimum irradiance thresholds. A dispersion analysis was also conducted to detect suspicious values and assess the hourly distribution of DHI. The work also describes the layout of the station's equipment and the main technical issues encountered, such as network connectivity failures, dirty radiometer domes, and the inoperability of the SOLYS2 solar tracker, all of which affected data collection. The results indicate good quality in GHI measurements and the need for more rigorous corrections in DHI data. This study contributes to improving the quality control of radiometric data and highlights the importance of proper equipment maintenance to ensure the continuity and reliability of solar irradiance time series.

Keywords: 1- Solar energy. 2- Meteorological instrumentation. 3- Data qualification.

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 1 - Equipamentos Bancada Leste.....	10
Tabela 2 - Equipamentos Bancada Leste.....	11
Tabela 3 - Equipamentos Bancada Sul.....	12
Tabela 4 - Equipamentos Bancada Leste.....	13
Tabela 5 - Resultados de RMSE e Viés dos modelos para K_t e K_d	25

LISTA DE FIGURAS E GRÁFICOS

Pág.

Figura 1 - Foto da Face Norte - Local onde a Câmera AllSky deve ficar, juntamente do abrigo ventilado (Autoria própria).....	10
Figura 2 - Foto da Bancada Leste (Autoria própria).....	11
Figura 3 - Foto da Bancada Oeste (Autoria própria).....	12
Figura 4 - Foto da Bancada Oeste (Autoria própria).....	13
Figura 5 - SOLYS2 Sun Tracker (KIPP & ZONEN, 2024).....	15
Gráfico 1 - Testes de qualidade de GHI (autoria própria).....	16
Gráfico 2 - Testes de qualidade de DHI (autoria própria).....	18
Gráfico 3 - Relação de GHI vs DHI retirando dados suspeitos (autoria própria).....	19
Gráfico 4 - Relação KT vs KD sem os dados suspeitos (autoria própria).....	21
Gráfico 5 - KT vs KD com curva de Erbs (autoria própria).....	22
Gráfico 6 - KT vs KD com Ajuste Polinomial (autoria própria).....	23
Gráfico 7 - KT vs KD com Random Forest (autoria própria).....	24
Gráfico 8 - Distribuição pontual com densidade da radiação difusa por horas do dia (autoria própria).....	26
Gráfico 9 - Calendário Solar (autoria própria).....	27

LISTA DE SIGLAS E ABREVIATURAS

INPE	Instituto Nacional de Pesquisas Espaciais
SONDA	Sistema de Organização Nacional de Dados Ambientais
LAVAT	Laboratório de Variáveis Ambientais
LABREN	Laboratório de Modelagem e Estudos de Recursos Renováveis de Energia

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO.....	1
1.1 A história do Laboratório de Variáveis Ambientais Tropicais - LAVAT.....	1
1.2 Desenvolvimento da pesquisa.....	2
1.3 Justificativa da Pesquisa.....	2
1.4 Objetivo Geral.....	3
1.4.1 Objetivos Específicos:.....	3
2 DESENVOLVIMENTO.....	4
2.1 Revisão bibliográfica.....	4
2.2 Materiais e Métodos.....	6
2.2.1 DHI e GHI.....	7
2.2.1.1 Radiação Solar Global Horizontal (GHI).....	7
2.2.1.2 Radiação Solar Difusa Horizontal (DHI).....	7
2.2.2 Validação dos dados.....	7
2.2.2.1 Importância da Validação.....	7
2.2.2.2 Método de Validação.....	8
2.2.2.2.1 Testes para GHI (Radiação Global).....	8
2.2.2.2.2 Testes para DHI (Radiação Difusa).....	9
2.2.3 Relação entre a fração difusa da radiação solar e o índice de claridade no ano de 2015.....	9
2.2.4 Disposição dos equipamentos nas bancadas.....	10
2.2.4.1 Face Norte.....	10
2.2.4.2 Bancada Leste.....	10
2.2.4.3 Bancada Sul.....	11
2.2.4.4 Bancada Oeste.....	12
2.2.5 Problemas encontrados nos equipamentos.....	13
2.2.5.1 Conexão de rede.....	14
2.2.5.2 Câmera inoperante.....	14
2.2.5.3 Rastreador inoperante.....	14
3 RESULTADOS.....	16
3.1 Testes de qualidade de GHI.....	16
3.2 Testes de qualidade de DHI.....	17
3.3 Análise de Dispersão GHI vs DHI com Detecção de Valores Suspeitos.....	18
3.4 Análise da Relação KT (Índice de Claridade) vs KD (Fração Difusa).....	20
3.5 Erbs para Análise do Comportamento do kt e do kd.....	22
3.6 Ajuste polinomial (grau 4) para Análise do Comportamento do kt e do kd.....	23
3.7 Random forest para Análise do Comportamento do kt e do kd.....	23
3.8 Comparação do RMSE e viés de Erbs, Ajuste Polinomial e Random Forest.....	25
3.9 Distribuição da densidade da Radiação Difusa (DHI) por hora do dia.....	26

3.10 Intensidade da radiação solar de Junho a Dezembro de 2024.....	27
4 CONCLUSÃO.....	28
5 REFERÊNCIAS BIBLIOGRÁFICAS.....	29
ANEXO.....	31

1 INTRODUÇÃO

Estudar o comportamento de variáveis ambientais faz jus ao nome do laboratório LAVAT, que tem como nome por extenso “Laboratório de Variáveis Ambientais Tropicais”.

1.1 A história do Laboratório de Variáveis Ambientais Tropicais - LAVAT

De acordo com o Centro Regional do Nordeste do INPE (2012) que hoje é a COENE, as pesquisas de Ozônio no Brasil tiveram início em 1974 com o *Dr. Yogeshwar Sahai* quando este trouxe o primeiro Espectrofotômetro de Dobson que foi instalado na Cidade de Cachoeira Paulista-SP e o segundo foi instalado em Natal, em novembro de 1978, tendo como operador o técnico Francisco Raimundo da Silva.

O Laboratório de Ozônio em Natal foi criado em 1978 pelo pesquisador *Dr. Volker Kirchhoff*, e se tornou a primeira estação de ozônio estabelecida nos trópicos que fornece dados contínuos para pesquisadores nacionais e internacionais. O objetivo inicial do laboratório foi dar apoio aos lançamentos de balões ozoniossondas, especialmente na preparação, calibração e redução dos dados, realizados no Centro de Lançamento de Foguetes da Barreira do Inferno, sendo hoje mais abrangente.

Foram instalados: um espectrofotômetro de *Dobson* (1978), um espectrofotômetro de *Brewer* (1994) e um radiômetro de GUV (1998) para estudo da Camada de Ozônio e radiação ultravioleta e PAR (Radiação fotossinteticamente ativa).

A partir de 1995, os balões passaram a ser lançados na sede do INPE/CRN (Campus da Universidade Federal do Rio Grande do Norte).

Em 2001, com a construção do laboratório em Barra de Maxaranguape/RN, os lançamentos foram realocados para lá, devido ao crescimento urbano de Natal e da necessidade de se obter amostras de ar não poluído.

Em 2006 foi criado o LAVAT - Laboratório de Variáveis Ambientais Tropicais, com maior abrangência e que atualmente abriga todas as atividades que eram conduzidas, até então, pelo Laboratório de Ozônio.

A partir daí, foi instalada uma estação meteorológica (2006) e uma estação solarimétrica (2007) na torre da caixa d'água do INPE-Natal-RN.

A partir de setembro de 2013, os balões voltaram a ser lançados na sede do INPE/CRN (Campus da Universidade Federal do Rio Grande do Norte).

Em 2024 a estação de Caicó-RN foi desativada, assim alguns sensores foram realocados em Natal.

Atualmente o LAVAT conta somente com uma instalação física. A sede do LAVAT em Natal conta com 04 salas denominadas: Operação, Apoio técnico, Instrumentação e Supervisão. Sua ampliação prevê a construção de uma sala para Pesquisa. No entanto, apesar dos avanços significativos na tecnologia de ozoniosondagem, ainda existem desafios substanciais a serem superados, especialmente no que diz respeito ao processamento e análise dos dados coletados.

1.2 Desenvolvimento da pesquisa

Esta presente pesquisa de iniciação científica aspira avaliar a qualidade dos dados da estação solarimétrica da rede SONDA localizada na COENE-INPE, mantendo-a operante e implantando novos equipamentos de medição..

1.3 Justificativa da Pesquisa

A importância de desenvolver uma metodologia robusta, eficaz e precisa para a medição da irradiância solar difusa é inquestionável no contexto da energia solar. Segundo o plano de trabalho do projeto "Estudo comparativo entre métodos de medição da irradiância solar difusa aplicados ao levantamento de recurso solar em Natal (RN)", coordenado pelo INPE-CRN, a decomposição adequada das componentes direta e difusa da radiação solar é crucial para a estimativa da geração de energia em usinas fotovoltaicas. O uso de técnicas como o piranômetro com anel de sombreamento e sistemas de rastreamento solar são centrais nesse processo, proporcionando dados essenciais para melhorar os algoritmos de correção e, conseqüentemente, a confiabilidade das medições. Este projeto visa, portanto, não apenas aprimorar os métodos existentes, mas também desenvolver novos critérios de qualidade e algoritmos

que aumentem a precisão e a aplicabilidade das medições de irradiância solar difusa, contribuindo para a compreensão da variabilidade da irradiância solar difusa no Brasil.

1.4 Objetivo Geral

O objetivo geral é o estudo comparativo entre a medição de irradiância solar difusa na unidade INPE-CRN em Natal-RN por duas técnicas: i) piranômetro com anel de sombreamento e ii) piranômetro montado sobre rastreador solar sombreado por esferas. Estes resultados permitirão a revisão dos algoritmos de correção e propor melhorias de modo a aumentar a confiabilidade da decomposição da irradiância solar e subsidiar a validação de modelos computacionais de levantamento do recurso solar no futuro.

1.4.1 Objetivos Específicos:

- 1- Avaliação estatística descritiva das medições pelos diferentes métodos e de suas diferenças;
- 2- Estudo de novos critérios de qualidade para correção de viés gerado pelo anel de sombreamento através de análises de regressão com parâmetros ambientais e astronômicos;
- 3- Aperfeiçoamento do algoritmo para controle de qualidade dos dados de irradiância difusa coletados por aneis de sombreamento;
- 4- Avaliação de modelos empíricos de parametrização da fração difusa.

2 DESENVOLVIMENTO

2.1 Revisão bibliográfica

Uma boa estimativa de irradiação solar é um fator crítico para o desempenho dos sistemas fotovoltaicos (PV). A localização geográfica e as condições atmosféricas desempenham papéis significativos na determinação da quantidade de energia solar que pode ser aproveitada. No contexto do Brasil, que está situado em baixas latitudes, a avaliação do potencial de irradiação solar é essencial para maximizar o rendimento das instalações PV. Esta revisão bibliográfica examina diversas metodologias e estudos para estimar irradiação solar difusa,.

Cronemberger et al. (2012) destacam a importância da avaliação do potencial de irradiação solar para aplicações fotovoltaicas em edifícios localizados em latitudes baixas, fazendo um estudo de caso para o Brasil. Eles abordam a variabilidade espacial e temporal da irradiação solar e discutem métodos para otimizar a captação de energia solar em ambientes urbanos, estimando a radiação difusa pelo índice de claridade.

Dal Pai et al. (2016) desenvolveram o método *MEO shadowring* para medir a irradiação solar difusa, propondo correções baseadas na cobertura do céu. Este método é crucial para garantir medições precisas de irradiação difusa, que é uma componente importante da irradiação total recebida por sistemas PV. Eles também exploram como diferentes condições atmosféricas podem influenciar essas medições.

O manual da *Kipp & Zonen* para o anel de sombra CM 121 detalha os princípios de funcionamento e a metodologia de uso do anel de sombra combinado com um piranômetro para medir a irradiação solar difusa. Este dispositivo é particularmente útil para medições em qualquer latitude, sendo essencial para garantir que o piranômetro permaneça na sombra, permitindo a medição apenas da irradiação difusa.

O manual da *Kipp & Zonen* discute a necessidade de ajustes regulares nas barras deslizantes para acompanhar a mudança na declinação solar ao longo do ano. Isso garante a precisão das medições, considerando que o anel

de sombra também intercepta parte da irradiação difusa, que deve ser corrigida matematicamente.

Mehler et al. (2010) exploram a determinação do ângulo de inclinação e orientação ótimos para arrays fotovoltaicos, destacando a importância de ajustar os sistemas PV para maximizar a captação de energia ao longo do ano. Eles utilizam modelos matemáticos para calcular os ângulos ideais de inclinação, considerando fatores como latitude e variabilidade sazonal da irradiação solar.

O Atlas Brasileiro de Energia Solar, compilado por Pereira et al. (2017), fornece uma análise abrangente dos recursos solares no Brasil. Este atlas é uma ferramenta essencial para planejadores e engenheiros que buscam implementar sistemas PV em diferentes regiões do país. Ele inclui mapas detalhados de irradiação solar, dados históricos e projeções que ajudam a identificar as áreas com maior potencial para geração de energia solar.

O Sistema de Organização Nacional de Dados Ambientais (SONDA) é outra fonte crucial de dados sobre irradiação solar e outras variáveis meteorológicas no Brasil. Disponibiliza dados que podem ser utilizados para validar modelos de irradiação solar e para o planejamento de sistemas PV.

A otimização da captação de energia solar em baixas latitudes, como no Brasil, requer uma boa compreensão das variáveis de irradiação solar e especificamente das metodologias de medição da irradiação difusa. Os estudos e recursos abordados nesta revisão fornecem uma base inicial para o desenvolvimento e a implementação de bons métodos para estimativa da irradiância no plano de captação dos sistemas fotovoltaicos. A combinação de dados empíricos, como os fornecidos pelo Atlas Brasileiro de Energia Solar e pelo sistema SONDA, com avanços metodológicos, como o método *MEO shadowring* e as diretrizes para o uso do anel de sombra CM 121, permite uma abordagem complementar trazendo mais rigor para as estimativas da fração difusa do potencial de energia solar no Brasil.

2.2 Materiais e Métodos

A validação dos dados usados foi feita usando o protocolo do SONDA para variáveis radiométricas, conforme descrito na página oficial de validação do CCST/INPE (SONDA, 2024).

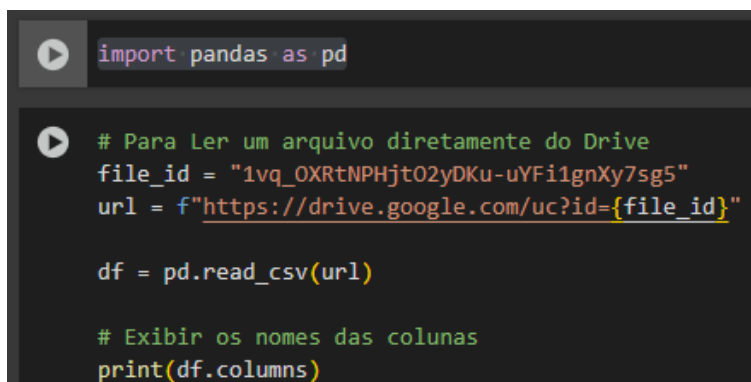
A linguagem de programação utilizada foi *Python*, com os códigos desenvolvidos e executados no ambiente de *Jupyter Notebook* por meio da plataforma *Google Colab*. A escolha do *Google Colab* se justifica pela sua característica de ser uma plataforma baseada em nuvem, que permite o desenvolvimento, execução e compartilhamento de códigos de forma acessível e independente do sistema operacional, podendo ser utilizado em diferentes dispositivos, como computadores, tablets e sistemas como *Windows*, *Linux* ou *MacOS*.

Esse ambiente online favorece significativamente o trabalho colaborativo, uma vez que os orientadores e colaboradores podem acessar, revisar e executar os códigos de qualquer localidade, necessitando apenas de conexão com a internet, além de guardar um histórico de alterações.

Os dados foram trabalhados em *.csv* e em *.dat* usando as bibliotecas *numpy*, *matplotlib*, *seaborn*, *pandas*, *scipy* e *pvlb*.

Os códigos para processamento e para plot de gráfico no anexo A.

Para facilitar a leitura dos dados, o arquivo fonte foi hospedado no *Google Drive*, não sendo necessário baixar nada em máquina local, o Colab lê diretamente o arquivo via link como pode ser observado na imagem X abaixo.

A screenshot of a Jupyter Notebook cell with a dark background. The code is written in Python and uses the pandas library to read a CSV file from Google Drive. The code includes comments in Portuguese. The first line imports pandas as 'pd'. The second line is a comment. The third line assigns a file_id. The fourth line constructs a URL using the file_id. The fifth line reads the CSV file into a DataFrame named 'df'. The sixth line is a comment. The seventh line prints the columns of the DataFrame.

```
import pandas as pd

# Para Ler um arquivo diretamente do Drive
file_id = "1vq_OXRtNPHjt02yDKu-uYFi1gnXy7sg5"
url = f"https://drive.google.com/uc?id={file_id}"

df = pd.read_csv(url)

# Exibir os nomes das colunas
print(df.columns)
```

Figura 1 - Código feito para ler o arquivo e deixá-lo em um dataframe. (Autoria própria)

2.2.1 DHI e GHI

2.2.1.1 Radiação Solar Global Horizontal (GHI)

A GHI (Global Horizontal Irradiance) é a radiação solar total incidente em uma superfície horizontal na superfície terrestre, composta pela soma da radiação direta (DNI) e difusa (DHI) (SONDA, 2024). Sua fórmula é dada por:

$$GHI = DNI \cdot \cos(\theta_z) + DHI$$

Onde:

- *DNI* (Direct Normal Irradiance): Radiação direta do Sol (W/m²).
- θ_z Ângulo zenital solar (graus)
- *DHI* (Diffuse Horizontal Irradiance): Radiação difusa (W/m²).

2.2.1.2 Radiação Solar Difusa Horizontal (DHI)

A DHI é a componente da radiação solar espalhada pela atmosfera, medida com um piranômetro equipado com anel de sombreamento para bloquear a radiação direta (MARTÍNEZ-DURÁ et al., 2020).

2.2.2 Validação dos dados

A validação de dados de radiação solar é uma etapa crítica para garantir a confiabilidade de medições usadas em projetos de energia solar, estudos climáticos e modelagem atmosférica.

2.2.2.1 Importância da Validação

É fundamental realizar a validação dos dados a fim de evitar erros sistemáticos que comprometam a qualidade das análises. Dados não validados podem conter falhas de medição, como sombreamento inadequado, sujeira nos sensores ou mau funcionamento dos equipamentos.

Além disso, é necessário assegurar a consistência física dos valores registrados. Por exemplo, os dados de irradiância solar devem respeitar limites teóricos estabelecidos, como o fato de a irradiância global horizontal (GHI) não poder ultrapassar a radiação extraterrestre incidente.

A utilização de dados não validados pode gerar consequências significativas em diversas áreas. No setor de energia solar, pode levar ao subdimensionamento de usinas fotovoltaicas, comprometendo a eficiência e a capacidade de geração de energia. Em estudos climáticos, o uso de dados com erros pode resultar em análises imprecisas, afetando a compreensão de padrões meteorológicos e tendências de longo prazo. Além disso, projetos energéticos baseados em informações incorretas estão sujeitos a prejuízos financeiros consideráveis, seja por investimentos mal direcionados, estimativas equivocadas de produção ou falhas no planejamento técnico.

2.2.2.2 Método de Validação

O método de validação utilizado foi o do SONDA, que se encontra no website oficial, no endereço <https://sonda.ccst.inpe.br/infos/validacao.html>, esse método prevê alguns algoritmos para diversas variáveis ambientais diferentes, para o estudo foram utilizados os quatro algoritmos de validação para variáveis Radiométricas. O processo de validação dos dados obtidos pelas estações SONDA baseia-se na estratégia de controle de qualidade de dados adotada pela BSRN (Baseline Surface Radiation Network)(SONDA, 2024).

2.2.2.2.1 Testes para GHI (Radiação Global)

Algoritmo 1: Limites Físicos

$$0 \leq GHI \leq 1.2 \cdot G0h$$

- $G0h$ = Radiação extraterrestre (calculada com pvlib).
- Fator 1.2: Considera efeitos de reflexão e dispersão atmosférica.

Um exemplo da aplicação desse algoritmo é: Se $G0h = 1000W/m^2$, GHI válida deve estar entre 0 e $1200W/m^2$.

Algoritmo 2: Relação $GHI-DHI$

$$GHI \geq DHI$$

Fisicamente, a radiação difusa (DHI) não pode ser maior que a global (GHI).

2.2.2.2.2 Testes para DHI (Radiação Difusa)

Algoritmo 3: Limite Superior para DHI

$$DHI \leq 0.8 \cdot G_{0h}$$

Em condições extremas (ex.: nuvens espessas), DHI pode atingir até 80% de G_{0h} .

Algoritmo 4: Consistência com Ângulo Zenital

$$DHI \leq GHI \cdot \cos(\theta_z)^{-1}$$

θ_z = Ângulo zenital (evita valores irreais quando o Sol está baixo).

2.2.3 Relação entre a fração difusa da radiação solar e o índice de claridade no ano de 2015

A relação entre a fração difusa (K_d) e o índice de claridade (K_t) é calculada a partir de medições de radiação solar global e difusa.

$$K_t = \frac{G_h}{G_{0h}}$$

G_h é a radiação solar global medida na superfície.

G_{0h} é a radiação solar extraterrestre horizontal.

$$K_d = \frac{G_d}{G_h}$$

G_d é a radiação solar difusa medida na superfície.

G_h é a radiação solar global medida na superfície.

A fração difusa da radiação solar (K_d) e o índice de claridade (K_t) são parâmetros cruciais na análise da radiação solar que atinge a superfície terrestre. Vários estudos destacam a correlação negativa entre K_d e K_t . No artigo "*Evaluation of diffuse fraction and diffusion coefficient using statistical analysis*", é demonstrado que à medida que K_t aumenta, indicando condições mais claras, K_d tende a diminuir, uma vez que mais luz solar atinge diretamente o solo (RAJU et al., 2018). Este estudo utiliza análises estatísticas para quantificar essa relação e fornecer coeficientes de difusão precisos para diferentes condições atmosféricas.

2.2.4 Disposição dos equipamentos nas bancadas

A estação solarimétrica da rede SONDA foi instalada no ano de 2007 e é operada pelo LAVAT desde então, sendo feitas as manutenções pelos técnicos do laboratório, a estação contava com diversos equipamentos e sensores para medir as variáveis ambientais.

2.2.4.1 Face Norte

Equipamento	Quantidade	Função
Câmera AllSky	1 (Uma)	Imagem de céu com 180 ° (Cobertura de nuvens)
Abrigo Ventilado	1 (Um)	Proteger o sensor de temperatura de receber radiação solar e chuva diretamente.

Tabela 1 - Equipamentos Bancada Leste

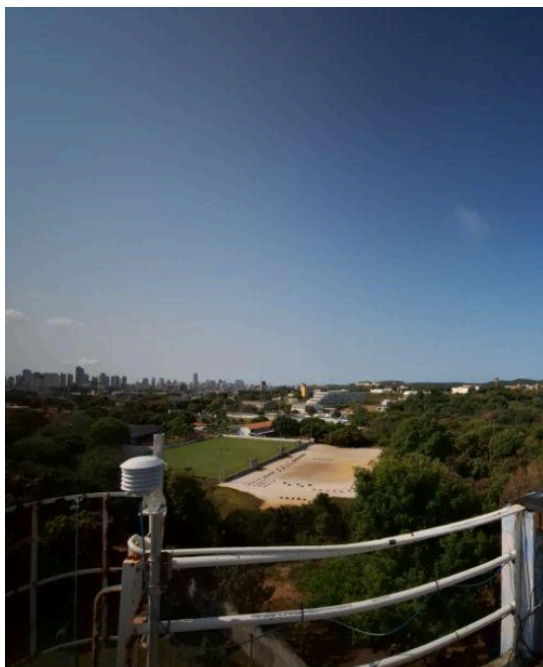


Figura 1 - Foto da Face Norte - Local onde a Câmera AllSky deve ficar, juntamente do abrigo ventilado (Autoria própria)

2.2.4.2 Bancada Leste

Equipament	Quantidad	Função
------------	-----------	--------

o	e	
Radiação Global	1 (Um)	Medir a radiação na superfície exposta ao sol (W/m ²)
Radiação Difusa	1 (Um)	Medir a radiação na sombra (W/m ²)
Radiação Onda Longa	1 (Um)	Medir a radiação de onda longa (W/m ²)
Caixa DataLogger	1 (Um)	Coletar, converter e registrar os sinais elétricos em grandezas físicas
Anemômetro	1 (Um)	Medir a velocidade (Em m/s) e a direção dos ventos (Em graus °)

Tabela 2 - Equipamentos Bancada Leste

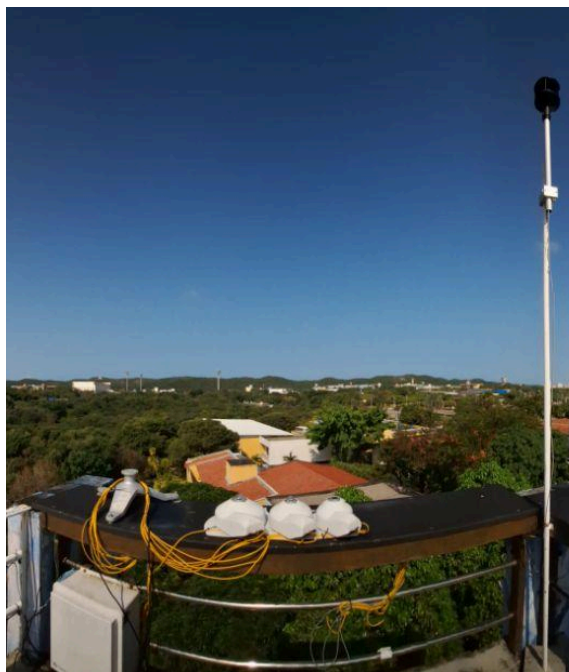


Figura 2 - Foto da Bancada Leste (Autoria própria)

2.2.4.3 Bancada Sul

Equipament o	Quantidad e	Função
Pluviômetro	1 (Um)	Medir a quantidade de precipitação líquida

		em um determinado período de tempo (mm).
Para-raio	1 (Um)	Proteger estruturas, equipamentos e sensores contra os efeitos destrutivos dos raios.
Termo-higrômetro	1 (Um)	Medição de temperatura do ar em abrigo (°C)

Tabela 3 - Equipamentos Bancada Sul



Figura 3 - Foto da Bancada Oeste (Autoria própria)

2.2.4.4 Bancada Oeste

Equipamento	Quantidade	Função
Radiação Difusa	1 (Um)	Medir a radiação na sombra (W/m^2)
Radiação Global e Par	1 (Um)	Medir a radiação na superfície exposta ao sol (W/m^2)
Luminosidade Difusa	1 (Um)	Medir a luminosidade na sombra (Lúmens)

Luminosidade Global	1 (Um)	Medir a luminosidade na superfície exposta ao sol (Lúmens)
---------------------	--------	--

Tabela 4 - Equipamentos Bancada Leste

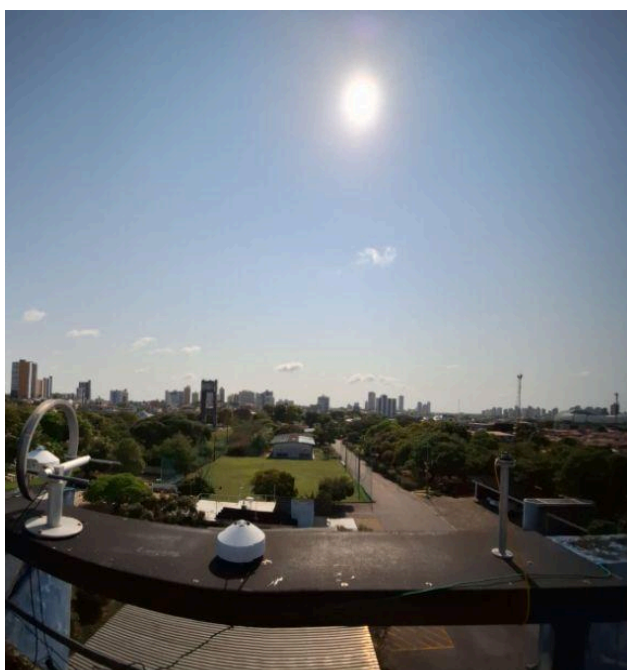


Figura 4 - Foto da Bancada Oeste (Autoria própria)

2.2.5 Problemas encontrados nos equipamentos

Durante a execução do projeto, foram identificados diversos problemas relacionados à infraestrutura que impactaram significativamente a disponibilidade dos dados. Entre os principais problemas encontrados estavam a interface de rede intermitente, que causou instabilidades na transmissão e coleta dos dados; a câmera *all-sky*, que apresentou falhas de *firmware*, comprometendo a captura de imagens; e os domos sujos dos radiômetros, que prejudicaram a qualidade das medições. A quebra das correias do rastreador solar *SOLYS2* também comprometeu a disponibilidade dos dados. Todos esses fatores, isoladamente e em conjunto, levaram a uma redução na disponibilidade e qualidade dos dados coletados.

2.2.5.1 Conexão de rede

Antes da atualização do *DataLogger* CR10 para o CR1000X, a estação estava com frequentes quedas de conectividade, foram feitos diversos resets na estação com o objetivo de reaver a conexão com a rede do INPE, porém sem êxito na retomada da conectividade até então.

De acordo com as medições feitas anteriormente totalizando 49 metros e seguindo as recomendações do Dr. Francisco Raimundo da Silva, Técnico no LAVAT, que sugeriu o cabo de rede categoria CAT-6, foram instalados os novos cabos, sendo um para o CR1000X e um para a câmera *AllSky*.

Os conectores RJ45 foram instalados em *in loco*, após a passagem dos cabos pelos dutos, evitando sujeira e danos físicos aos conectores.

Após ser realizada a instalação dessa nova infraestrutura de cabos a conexão foi retomada com sucesso.

2.2.5.2 Câmera inoperante

Após solicitado, a câmera que estava apresentando falhas de *firmware* e comprometendo a coleta das imagens, foi retirada e enviada para o LABREN (Laboratório de Modelagem e Estudos de Recursos Renováveis de Energia), no INPE São José dos Campos para reparo.

2.2.5.3 Rastreador inoperante

O rastreador *SOLYS2 Sun Tracker* remanejado da estação de Caicó-RN, que foi descontinuada, foi trazido a Natal-RN com o objetivo de melhorar os equipamentos instalados. O equipamento em questão é igual ao mostrado na figura 7 a seguir.



Figura 5 - SOLYS2 Sun Tracker (KIPP & ZONEN, 2024)

O rastreador apresentou defeito em sua correia, sendo necessário o envio para o LABREN (Laboratório de Modelagem e Estudos de Recursos Renováveis de Energia), no INPE São José dos Campos para reparo.

O equipamento ainda se encontra inoperante aguardando reparo para ser trazido de volta a Natal.

3 RESULTADOS

3.1 Testes de qualidade de GHI

A verificação da qualidade dos dados de irradiância global horizontal (GHI), representados pela variável `glo_avg_2`, foi realizada com base em dois testes comparativos com a irradiância extraterrestre (G_0), calculada a partir da posição solar no local da medição (Natal-RN). O Teste 1 considerou limites físicos esperados para GHI, restringindo seus valores ao intervalo de 0 a $1,2 \times G_0$. Já o Teste 2 aplicou limites mais permissivos, permitindo valores até $1,5 \times G_0$, a fim de detectar casos extremos ainda plausíveis. Ambos os testes tiveram como objetivo identificar registros fisicamente inconsistentes, substituindo-os por NaN em uma nova coluna corrigida (`glo_avg_2_corrigido`), como o gráfico 1 apresenta.

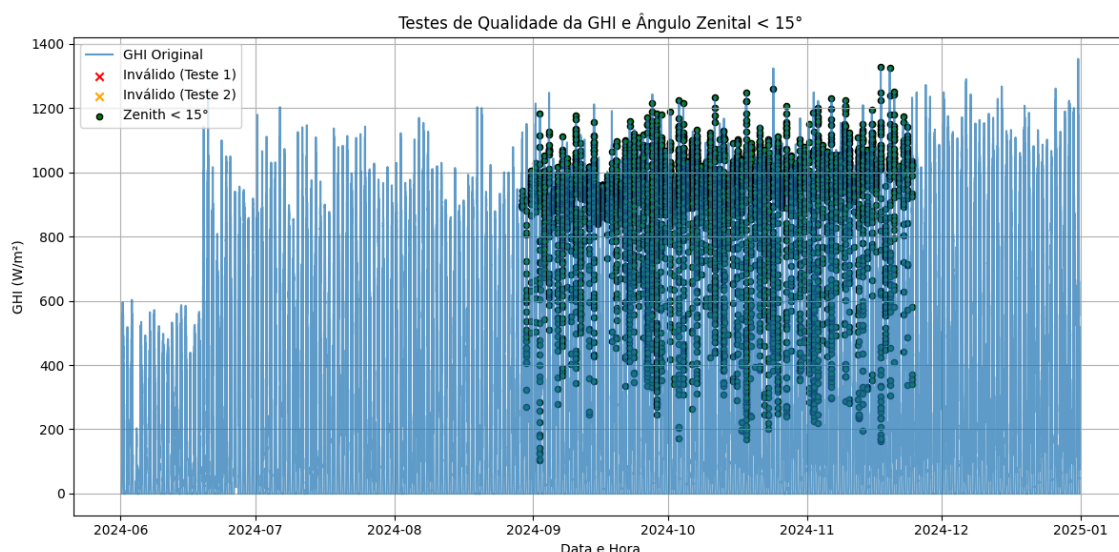


Gráfico 1 - Testes de qualidade de GHI (autoria própria)

Como resultado, dos dados analisados, foi identificado um total de 148.808 valores ausentes (NaN) antes mesmo da aplicação dos testes, o que representa uma parcela considerável de lacunas na série temporal. No entanto, nenhum valor presente nos dados foi classificado como inválido por qualquer um dos critérios aplicados. Isso indica que, apesar da quantidade de dados faltantes, os valores registrados estão em total conformidade com os limites físicos e extremos considerados, demonstrando alta consistência e confiabilidade na porção disponível dos dados de GHI.

3.2 Testes de qualidade de DHI

O teste de qualidade aplicado aos dados de irradiância difusa horizontal (DHI) teve como objetivo identificar medições inválidas ou inconsistentes com base em critérios físicos e empíricos reconhecidos. Três algoritmos foram utilizados: o primeiro aplica limites físicos máximos para o DHI com base na irradiância extraterrestre (G_0), no cosseno do ângulo zenital solar e em um limite de 85° para o zenite; o segundo é uma versão mais permissiva desses limites, permitindo a detecção de valores extremos ainda possíveis; e o terceiro compara o DHI com a irradiância global horizontal (GHI), assumindo que DHI não pode exceder GHI em mais de 50 W/m^2 . Esses testes ajudam a validar a integridade dos dados medidos em campo, considerando as condições reais da atmosfera e do posicionamento solar.

A análise considerou um total de 308.028 registros, dos quais 135.048 (43,8%) estavam ausentes (NaN) desde o início, indicando lacunas significativas no conjunto de dados. O primeiro teste de qualidade identificou 58.904 valores como inválidos (19,1% do total), sugerindo que quase um quinto dos dados disponíveis apresentaram valores fisicamente inconsistentes. O segundo teste, com critérios mais tolerantes, invalidou 56.487 registros (18,3%), mostrando leve concordância com o primeiro. Já o terceiro teste, baseado na comparação com a GHI, detectou apenas 46 inconsistências (0,0%), indicando que a grande maioria dos valores de DHI estavam dentro do esperado em relação à irradiância global. Como resultado, foi criada uma nova série corrigida (`dif_avg_2_corrigido`), na qual os valores reprovados no primeiro teste foram substituídos por NaN, permitindo a continuidade de análises com maior confiabilidade, o gráfico 2 apresenta isso.

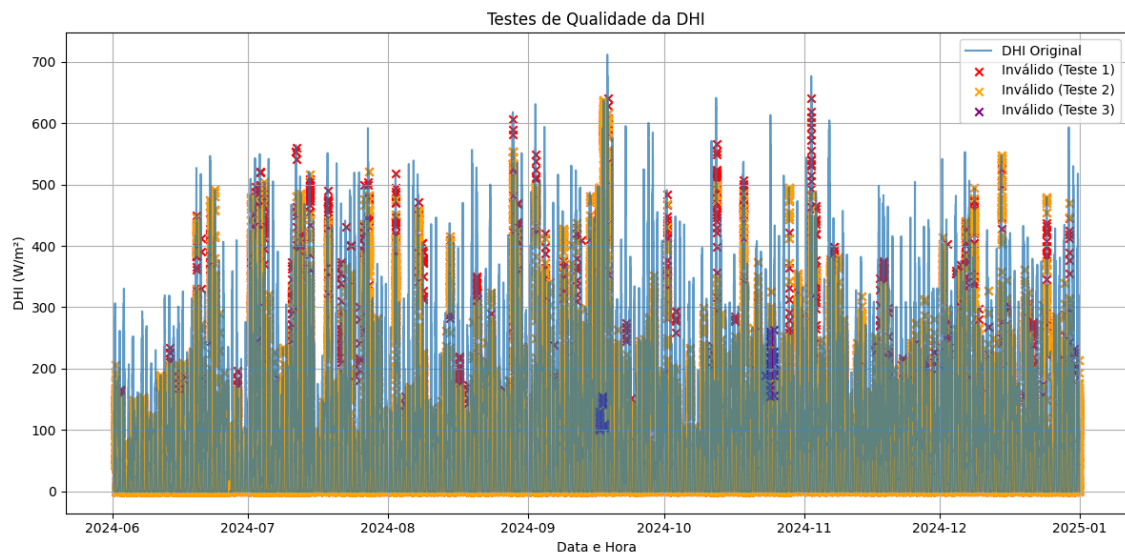


Gráfico 2 - Testes de qualidade de DHI (autoria própria)

Como resultado numérico para ter uma visão clara da qualidade dos dados analisados, como saída em texto foram obtidos os seguintes dados: dos 308.028 registros totais, 135.048 (43,8%) estavam ausentes (NaN) desde o início, indicando falhas ou lacunas na coleta. O Teste 1, baseado em limites físicos rigorosos, identificou 58.904 valores inválidos (19,1%), enquanto o Teste 2, mais permissivo, encontrou 56.487 (18,3%), revelando consistência entre os dois critérios. O Teste 3, que compara DHI com GHI, detectou apenas 46 inconsistências (0,0%), sugerindo que a maioria dos dados de DHI está compatível com a irradiância global. Como resultado, foi criada a coluna `dif_avg_2_corrigido`, na qual os dados inválidos do Teste 1 foram substituídos por NaN, permitindo análises futuras com maior confiabilidade.

3.3 Análise de Dispersão GHI vs DHI com Detecção de Valores Suspeitos

O objetivo desta etapa é verificar a consistência física entre os valores medidos de irradiância global horizontal (GHI) e irradiância difusa horizontal (DHI), identificando registros que apresentem valores suspeitos. Para isso, foram aplicados critérios como: DHI maior que GHI, valores negativos de DHI e valores de DHI muito elevados (acima de 1000 W/m²) quando o ângulo zenital solar é menor ou igual a 85 graus. Esses casos indicam possíveis erros de medição ou problemas nos sensores. O gráfico gerado mostra a dispersão dos

dados de GHI versus DHI, coloridos pelo ângulo zenital para facilitar a visualização da posição solar. Os dados suspeitos são destacados em vermelho com um marcador em "x", e linhas de referência são incluídas para indicar limites teóricos, como a igualdade entre GHI e DHI e um limite de 80% da irradiância global para o componente difuso. Essa análise visual permite identificar facilmente os pontos que não seguem as relações físicas esperadas, o que é fundamental para garantir a qualidade dos dados utilizados em estudos subsequentes. A relação de GHI com DHI após validação e correção dos dados pode ser observada no gráfico abaixo

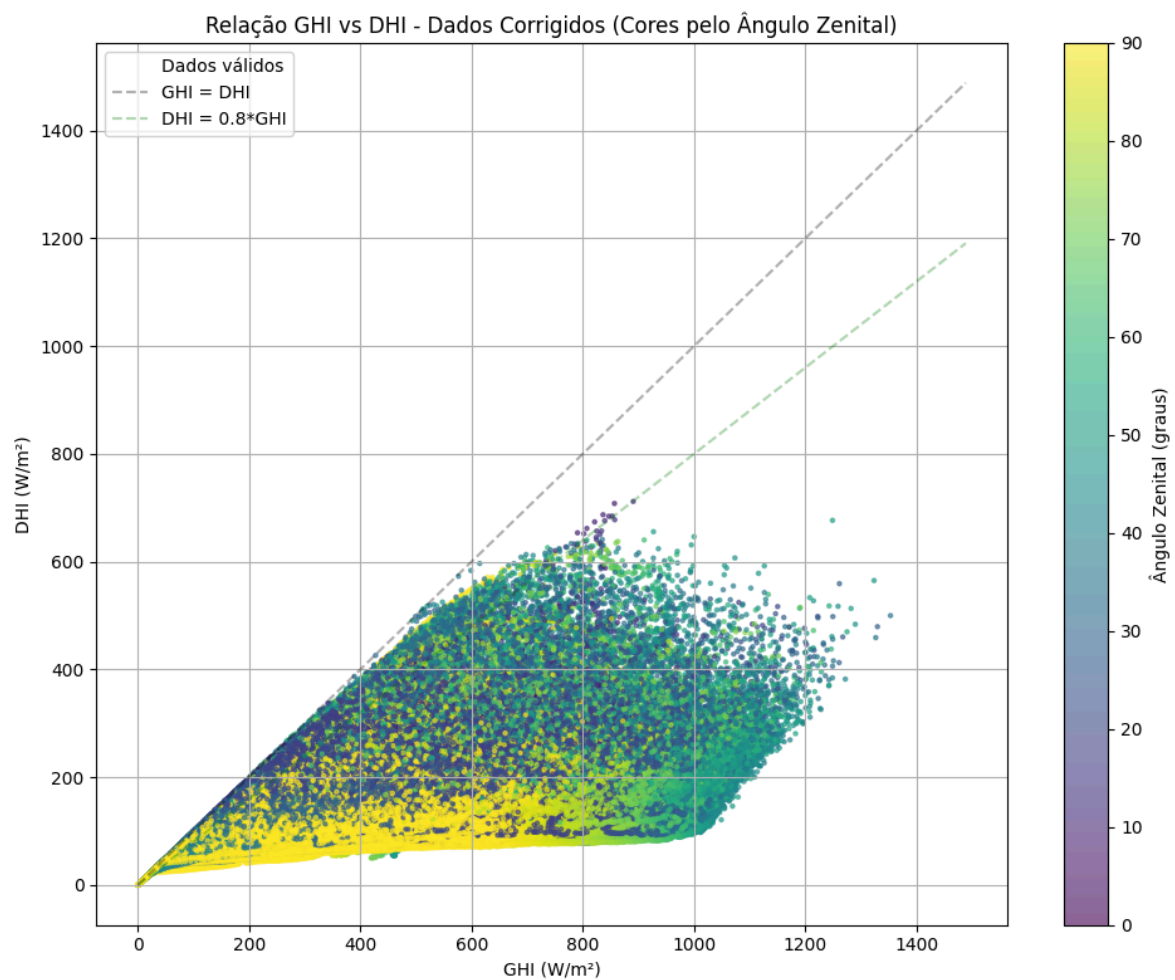


Gráfico 3 - Relação de GHI vs DHI retirando dados suspeitos (autoria própria)

3.4 Análise da Relação KT (Índice de Claridade) vs KD (Fração Difusa)

A última análise foca na relação entre o índice de claridade (KT), que é a razão entre a irradiância global horizontal medida e a irradiância extraterrestre, e a fração difusa (KD), que é a proporção da irradiância difusa em relação à global. Estes parâmetros são indicadores importantes das condições atmosféricas e do estado do céu, sendo amplamente utilizados para caracterizar a nebulosidade e a transparência da atmosfera. Para garantir uma análise confiável, foram filtrados apenas os dados com ângulo zenital menor ou igual a 75 graus e com valores de GHI superiores a 10 W/m², eliminando assim registros de baixa radiação ou ângulos solares críticos que podem gerar distorções. No gráfico resultante, a dispersão dos pontos é apresentada colorida pelo ângulo zenital, enquanto os valores suspeitos identificados anteriormente são destacados em vermelho. Linhas de referência são incluídas para marcar limites típicos, como KD igual a 1 e KT igual a 0,8, auxiliando na interpretação dos dados.

A relação entre a fração difusa (K_d) e o índice de claridade (K_t) é calculada a partir de medições de radiação solar global e difusa.

$$K_t = \frac{G_h}{G_{0h}}$$

G_h é a radiação solar global medida na superfície.

G_{0h} é a radiação solar extraterrestre horizontal.

$$K_d = \frac{G_d}{G_h}$$

G_d é a radiação solar difusa medida na superfície.

G_h é a radiação solar global medida na superfície.

Essa análise possibilita avaliar a coerência dos dados medidos em relação às expectativas físicas e detectar potenciais inconsistências que possam comprometer estudos posteriores.

Excluindo os valores suspeitos obtemos o gráfico abaixo.

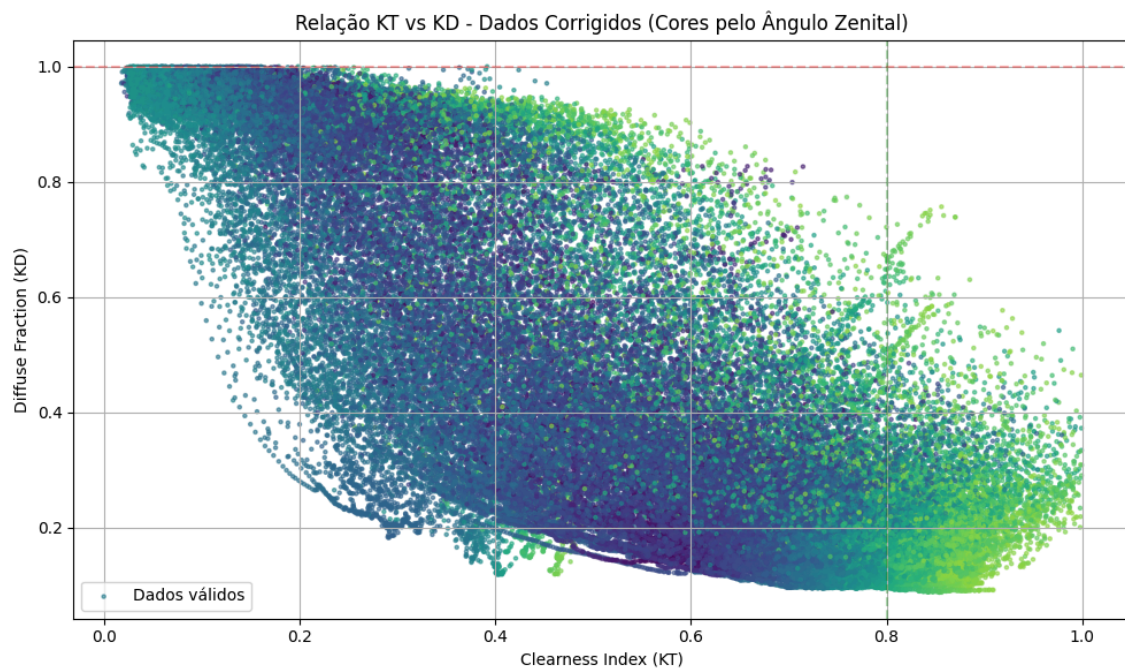


Gráfico 4 - Relação K_T vs K_D sem os dados suspeitos (autoria própria)

Eixo X (Índice de Claridade, K_t): Representa a claridade atmosférica, variando de 0 a 1. Valores baixos indicam baixa claridade (mais nublado), e valores altos indicam alta claridade (mais ensolarado).

Eixo Y (Fração Difusa, K_d): Representa a fração de radiação solar que é difusa. Valores altos indicam que uma grande parte da radiação é difusa, enquanto valores baixos indicam que a maior parte da radiação é direta.

Região de Alta Fração Difusa: Quando K_t é baixo, K_d é alto, significando que, em condições de baixa claridade, a maior parte da radiação solar é difusa.

Queda Rápida: Conforme K_t aumenta, K_d começa a cair rapidamente. Isso indica que, à medida que o índice de claridade aumenta (o céu fica mais claro), a fração de radiação que é difusa diminui rapidamente.

Estabilização: Para valores mais altos de K_t , a queda de K_d desacelera e eventualmente se estabiliza. Isso sugere que, em condições muito claras, a fração difusa se estabiliza em um valor baixo.

Quando o índice de claridade K_t é superior a 0,8, observa-se um aumento na fração de radiação difusa, o que indica a presença de mais

nuvens. Esse fenômeno ocorre principalmente quando há nuvens esparsas, cujas bordas brancas refletem a radiação em direção ao solo, aumentando a quantidade de radiação difusa e, conseqüentemente, o valor da radiação global.

Assim, um céu com nuvens esparsas pode gerar mais energia solar do que um céu completamente claro. Esse efeito é especialmente comum em atmosferas tropicais.

3.5 Erbs para Análise do Comportamento do k_t e do k_d

É um modelo empírico tradicional utilizado para estimar a fração difusa (K_D) com base no índice de claridade (K_T), geralmente ajustado a partir de observações experimentais. Apesar de simples e amplamente usado, pode não se adaptar bem a dados locais específicos.

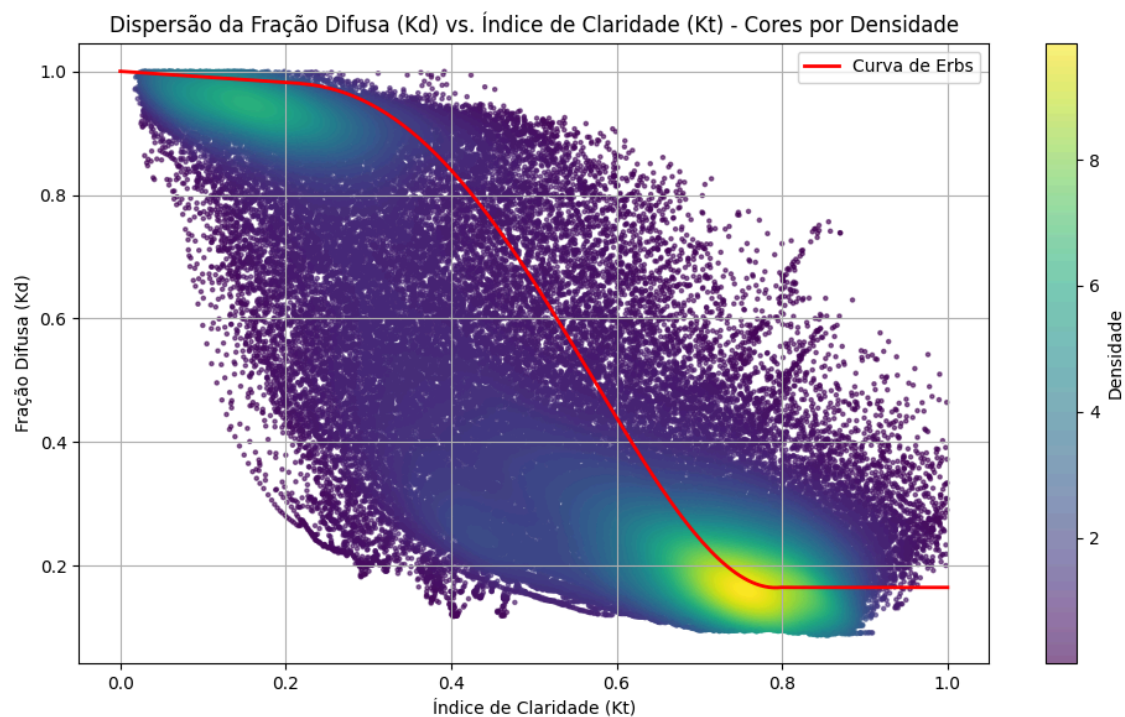


Gráfico 5 - K_T vs K_D com curva de Erbs (autoria própria)

3.6 Ajuste polinomial (grau 4) para Análise do Comportamento do kt e do kd

Trata-se de um ajuste matemático usando um polinômio de quarto grau. Ele modela a relação $KT \times KD$ de forma contínua e suave, capturando bem curvaturas presentes nos dados.

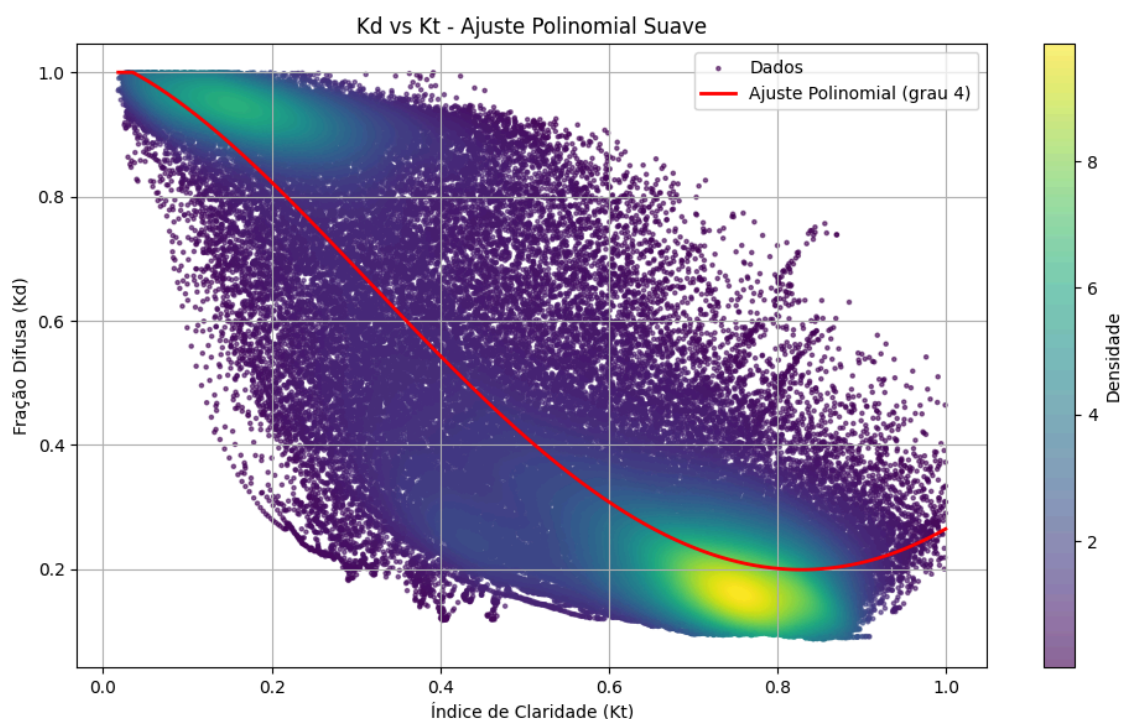


Gráfico 6 - KT vs KD com Ajuste Polinomial (autoria própria)

3.7 Random forest para Análise do Comportamento do kt e do kd

O uso de métodos de aprendizado de máquina (machine learning) tem se consolidado como uma ferramenta poderosa na análise de dados ambientais, especialmente por sua capacidade de modelar relações complexas e não lineares. Um desses métodos é o Random Forest, ou Floresta Aleatória, que consiste em um conjunto de árvores de decisão combinadas para produzir uma predição mais robusta e confiável.

No contexto da análise da relação entre o índice de claridade (Kt) e a fração difusa (Kd), o Random Forest é utilizado para visualizar o comportamento dos dados e entender como variações em Kt influenciam os

valores de K_d . Ao treinar o modelo com os dados observados, é possível obter uma superfície de predição que revela padrões e regiões de maior variabilidade.

O gráfico gerado pelo Random Forest mostra um comportamento não determinístico e inconstante da relação K_t vs K_d e isso tem uma explicação física. Isso significa que valores semelhantes de K_t podem resultar em diferentes valores de K_d , o que contraria modelos determinísticos como a curva ERBS. Essa variabilidade é explicada principalmente pela dinâmica atmosférica, já que a distribuição das nuvens varia constantemente ao longo do tempo e do espaço. Em diferentes condições atmosféricas, um mesmo valor de K_t pode estar associado a diferentes composições de radiação difusa e direta, resultando em diferentes valores de K_d .

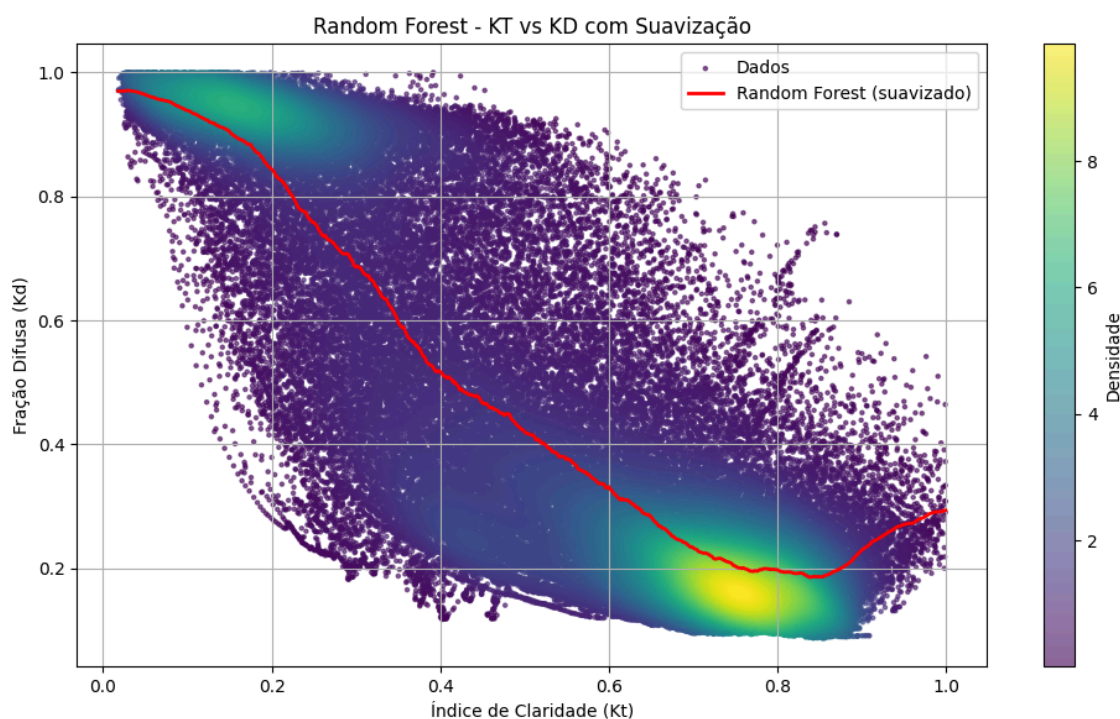


Gráfico 7 - K_T vs K_D com Random Forest (autoria própria)

Portanto, o uso do Random Forest se adequa a casos como este onde a relação entre K_t e K_d não é única e exata, mas sim influenciada por diversos

fatores ambientais, sendo um método útil para representar essa incerteza e complexidade natural dos dados.

3.8 Comparação do RMSE e viés de Erbs, Ajuste Polinomial e Random Forest

RMSE (Erro Quadrático Médio da Raiz) é uma medida que mostra o quanto, em média, os valores previstos por um modelo se afastam dos valores reais. Ele é bastante sensível a erros grandes: se o modelo cometer um erro grande em poucos pontos, o RMSE tende a aumentar bastante. Quanto menor for o RMSE, melhor o desempenho do modelo — isso indica que as previsões estão mais próximas da realidade.

Viés (ou bias) mostra se o modelo tem uma tendência a sempre errar para mais ou para menos. Se o viés for positivo, significa que o modelo está, em média, superestimando os valores reais (prevendo valores maiores do que são de fato). Se o viés for negativo, o modelo tende a subestimar (prevê valores menores). O ideal é que o viés esteja próximo de zero, o que indica ausência de tendência sistemática de erro.

Na tabela 5 encontram os resultados de RMSE e Viés para os 3 modelos apresentados acima.

	Erbs	Polinomial	Random Forest
RMSE	0.4428	0.1581	0.1509
Viés	0.1491	-0.0000	0.0000

Tabela 5 - Resultados de RMSE e Viés dos modelos para Kt e Kd

Na comparação entre os três métodos avaliados — modelo empírico de Erbs, ajuste polinomial e Random Forest —, observou-se que tanto o ajuste polinomial quanto o modelo Random Forest apresentaram melhor desempenho na estimativa da fração difusa (KD) a partir do índice de claridade (KT), quando comparados ao modelo tradicional de Erbs. O valor do RMSE foi significativamente menor nos modelos ajustados (0,1581 para o polinomial e 0,1509 para o Random Forest), indicando que suas previsões foram mais

próximas dos valores reais. Além disso, ambos apresentaram viés praticamente nulo, demonstrando que não houve tendência sistemática de superestimação ou subestimação nos resultados. Em contraste, o modelo de Erbs apresentou maior erro (RMSE = 0,4428) e viés positivo (0,1491), evidenciando uma tendência de superestimar os valores de KD. Esses resultados indicam que os métodos baseados em dados locais, especialmente o Random Forest, são mais adequados para representar a variabilidade da relação entre KT e KD na região estudada.

3.9 Distribuição da densidade da Radiação Difusa (DHI) por hora do dia

Para entender quais horas em que os dados mais se concentram ao longo dos dias foi feito um gráfico de densidade, no qual o eixo vertical é a Radiação Difusa em W/m^2 e o eixo horizontal são as horas do dia 0h-20h, o comportamento de DHI durante os dias pode ser observado no gráfico 7.

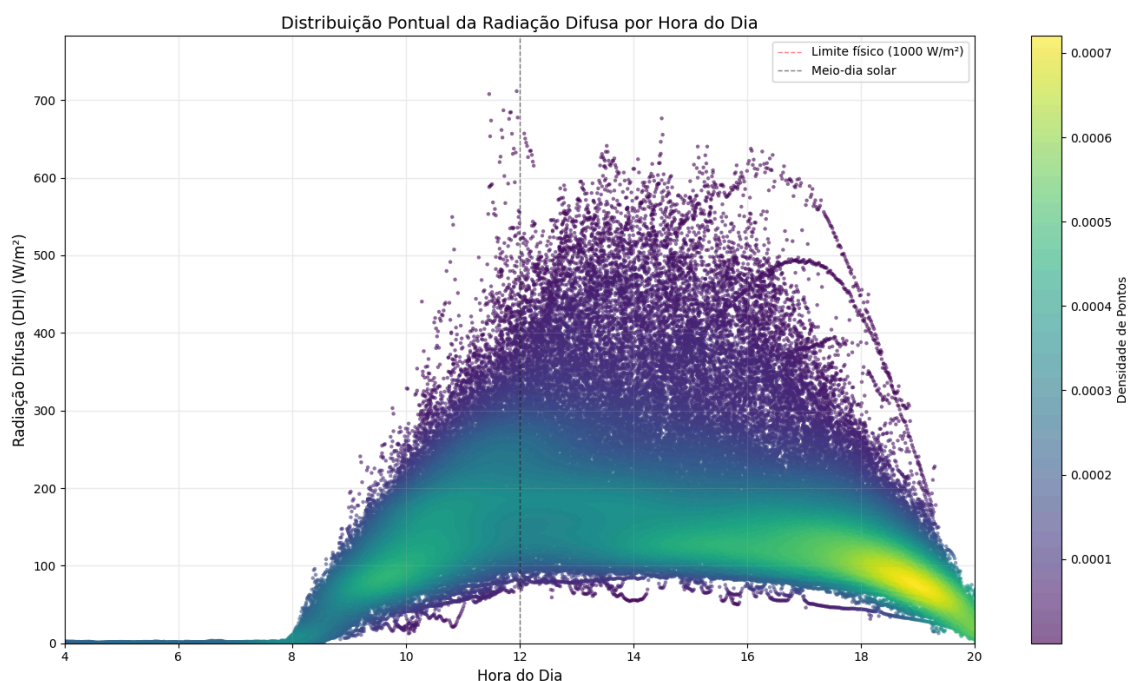


Gráfico 8 - Distribuição pontual com densidade da radiação difusa por horas do dia (autoria própria)

O gráfico apresenta os dados em horário Zulu (UTC). Considerando que em Natal a hora local está no fuso UTC-3, observa-se que a maior concentração de dados ocorre entre 18h e 20h UTC, o que corresponde ao período entre 15h e 17h no horário local. Essa densidade está na faixa de 100

a 300 W/m^2 , isso é um bom indicador da qualidade dos dados de DHI, embora ainda existam pontos suspeitos, esses não foram detectados pelos testes de qualidade, esse comportamento anômalo pode ser observado nas duas gaussianas entre as 14h e 20h que tem comportamento diferente da nuvem de pontos.

3.10 Intensidade da radiação solar de Junho a Dezembro de 2024

Para os natalenses, a intensidade do sol é percebida no dia a dia de forma intuitiva na pele — mesmo sem estudos formais, é amplamente reconhecido que a radiação solar costuma ser elevada na maior parte dos dias. Com o objetivo de compreender de forma mais precisa a distribuição da irradiância solar ao longo dos dias do segundo semestre, foi elaborado um Calendário Solar, apresentado no Gráfico 8.

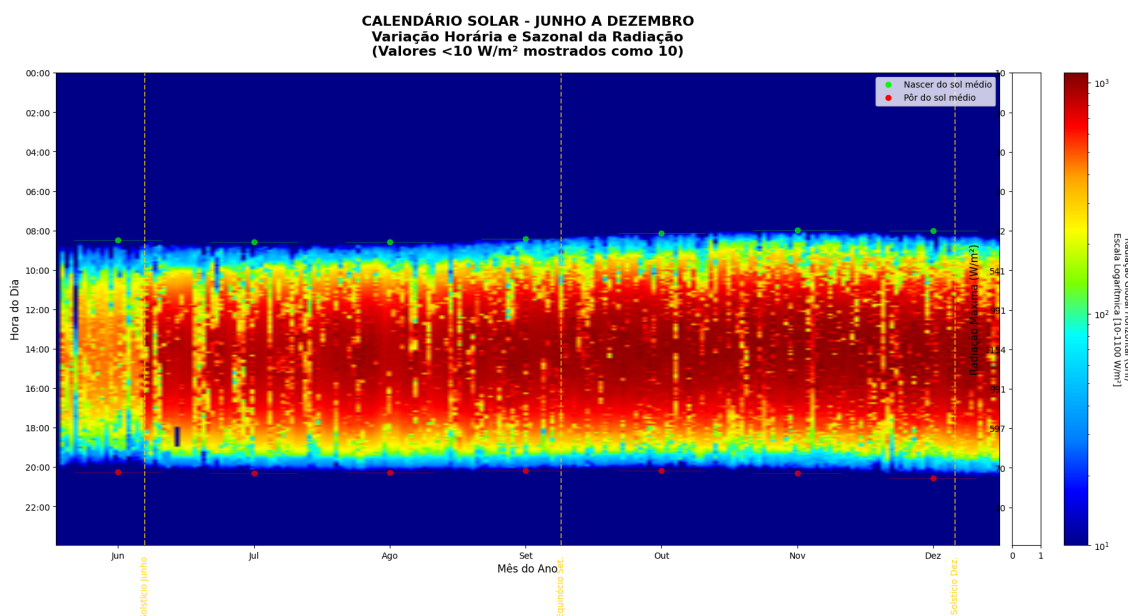


Gráfico 9 - Calendário Solar (autoria própria)

O gráfico evidencia um padrão de alta intensidade de radiação ao longo dos dias, o que segundo Martins et al. (2004), se deve à alta frequência de céu claro e ao baixo número de dias chuvosos na região Nordeste.

4 CONCLUSÃO

O desenvolvimento do projeto proporcionou um aperfeiçoamento significativo no conhecimento sobre instrumentos de medição e coleta de dados meteorológicos, além de promover uma sólida experiência prática em infraestrutura de rede, análise e ciência de dados, bem como na aplicação de técnicas estatísticas.

Durante a fase de análise, foram desenvolvidos algoritmos em uma plataforma remota, o que possibilitou o contato direto com ambientes de computação em nuvem e o tratamento de grandes volumes de dados, agregando valor ao aprendizado sobre escalabilidade e desempenho computacional.

A análise permitiu observar com clareza problemas recorrentes relacionados à angulação solar sobre o sensor de radiação difusa. Essa limitação reforça a importância da utilização de um rastreador solar, que assegura maior confiabilidade nas medições e facilita a comparação entre diferentes conjuntos de dados.

Contudo, a comparação com os dados obtidos por meio do rastreador solar ainda não foi possível, uma vez que o equipamento encontra-se em manutenção no INPE, em São José dos Campos - SP. Sua ausência compromete a capacidade de detectar falhas sutis no alinhamento do anel de sombreamento, o que já foi identificado como uma possível fonte de erro.

Adicionalmente, foi identificado um número expressivo de dados suspeitos ao longo da série temporal analisada, possivelmente decorrentes de falhas de posicionamento, obstruções no sensor ou condições atmosféricas atípicas. Esse achado reforça a necessidade de sistemas complementares de validação e controle de qualidade dos dados coletados.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ABDULLAH, Qassim A. *Relationship Between Standard Deviation and Root Mean Square Error (RMSE)*. Penn State: GEOG 892: Geospatial Applications of Unmanned Aerial Systems (UAS). Disponível em: <https://www.e-education.psu.edu/geog892/node/704>. Acesso em: 6 ago. 2024.

CRONEMBERGER, J.; CAAMAÑO-MARTÍN, E.; SÁNCHEZ, S. V. Assessing the solar irradiation potential for solar photovoltaic applications in buildings at low latitudes – Making the case for Brazil. *Energy and Buildings*, v. 55, p. 264-272, 2012.

DAL PAI, A. et al. MEO shadowring method for measuring diffuse solar irradiance: Corrections based on sky cover. *Renewable Energy*, v. 99, p. 754-763, 2016.

DRAPER, N. R.; SMITH, H. *Applied Regression Analysis*. John Wiley & Sons, 1998.

ERBS, D. G.; KLEIN, S. A.; DUFFIE, J. A. Estimation of the Diffuse Radiation Fraction for Hourly, Daily and Monthly-Average Global Radiation. *Solar Energy*, v. 28, n. 4, p. 293-302, 1982.

HUTCH. *RMSE vs Standard deviation in population*. Cross Validated. Disponível em: <https://stats.stackexchange.com/questions/270053/rmse-vs-standard-deviation-in-population>. Acesso em: 6 ago. 2024.

IQBAL, Muhammad. *An Introduction to Solar Radiation*. Elsevier, 2012.

KIPP & ZONEN. *Shadow Ring CM 121 Manual*. 2004.

KIPP & ZONEN. *Solys 2 sun tracker*. 2024. Disponível em: <https://www.kippzonen.com/Product/20/SOLYS2-Sun-Tracker>. Acesso em: 21 jun. 2024.

MARTÍNEZ-DURÁ, J. et al. *New Models for Separating Hourly Diffuse and Direct Components of Global Solar Radiation*. Springer, 2020. Disponível em: <https://link.springer.com/article/10.1007/s00477-020-01855-0>. Acesso em: 01 jul. 2024.

MARTINS, F. R. et al. The Role of Solar Energy in the Northeast of Brazil: The Use of Satellite Data to Determine Solar Radiation. In: *COSPAR Scientific Assembly*, 35., 2004, Paris. Anais [...]. Paris: COSPAR, 2004. Disponível em:

https://labren.ccst.inpe.br/producao/eventos/COSPAR2004_Paris_COSPAR04-A-00738_FRMartins_etal.pdf. Acesso em: 10 jul. 2025.

MCARTHUR, L. J. B. Baseline Surface Radiation Network (BSRN). Operations Manual. WMO/TD-No. 1274, WCRP/WMO, 2005.

MEHLERI, E. D.; ZERVAS, P. L.; SARIMVEIS, J. A.; MARKATOS, N. C. Determination of optimal tilt angle and orientation for solar photovoltaic arrays. *Renewable Energy*, v. 35, p. 2468-2475, 2010.

Montgomery, D. C.; Runger, G. C. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2010.

PEREIRA, E. B. et al. Atlas Brasileiro de Energia Solar. 2. ed. São José dos Campos, SP: INPE, 2017.

RAJU, A. S. et al. *Evaluation of Diffuse Fraction and Diffusion Coefficient Using Statistical Analysis*. Springer, 2018. Disponível em: <https://link.springer.com/article/10.1007/s13201-018-0790-8>. Acesso em: 01 jul. 2024.

SONDA – Sistema de Organização Nacional de Dados Ambientais. Disponível em: <http://sonda.ccst.inpe.br/>. Acesso em: setembro de 2023.

SONDA. *Validação.* Disponível em: <https://sonda.ccst.inpe.br/infos/validacao.html>. Acesso em: 07 ago. 2024.

TP-Link. *TL-SG1024.* Disponível em: https://static.tp-link.com/TL-SG1024_UN_11.0_02_1499780179532m.jpg. Acesso em: 20 jun. 2024.

Wackerly, D. D.; Mendenhall, W.; Scheaffer, R. L. *Mathematical Statistics with Applications*. Cengage Learning, 2014.

Anexo A

Códigos do Jupyter Notebook

Este anexo contém os principais trechos de código utilizados no desenvolvimento do trabalho.

Código 1 — Processamento dos Dados

```
1 import pandas as pd
2 # Para Ler um arquivo diretamente do Drive
3 file_id = "1vq_OXRtNPHjt02yDKu-uYFi1gnXy7sg5"
4 url = f"https://drive.google.com/uc?id={file_id}"
5 df = pd.read_csv(url)
6
7 # Exibir os nomes das colunas
8 print(df.columns)
9
10 # Convertendo a coluna 'data' para datetime
11 df['data'] = pd.to_datetime(df['data'], format='%Y-%m-%d', errors='
    coerce')
12
13 # Filtrando apenas o ano de 2024
14 df_filtrado = df[(df['data'] >= '2024-06-01') & (df['data'].dt.year ==
    2024)]
15
16 # Substituindo "NAN" por valores NaN reais
17 df_filtrado.replace("NAN", pd.NA, inplace=True)
18
19 import numpy as np
20 # Definindo as colunas a serem verificadas
21 colunas_para_verificar = [
22     'glo_avg_2', 'glo_std_2', 'glo_min_2', 'glo_max_2',
23     'dif_avg_2', 'dif_std_2', 'dif_min_2', 'dif_max_2'
24 ]
25
26 # Convertendo as colunas para numerico, erros serao substituidos por NaN
27 for coluna in colunas_para_verificar:
28     df_filtrado[coluna] = pd.to_numeric(df_filtrado[coluna], errors='
        coerce')
29
30 # Substituindo os valores negativos por NaN
31 df_filtrado[colunas_para_verificar] = df_filtrado[colunas_para_verificar
    ].where(df_filtrado[colunas_para_verificar] >= 0)
32 df_filtrado.head(10) # Exibe as primeiras 10 linhas
```

Listing 1: Importacao de dados e bibliotecas

Código 2 — Cálculo do ângulo Zenital usando a PVLIB

```
1 import pandas as pd
2 import numpy as np
3 import pvlib
```

```

4 from pvlib import solarposition, irradiance
5 from pvlib.location import Location as PVLibLocation
6
7 # 1. CONFIGURACAO DA LOCALIZACAO (NATAL/RN)
8 # =====
9 latitude = -5.836750      # Latitude de Natal
10 longitude = -35.206478    # Longitude de Natal
11 altitude = 56             # Altitude (metros)
12
13
14 # Use PVLibLocation to create the location object
15 local = PVLibLocation(
16     latitude,
17     longitude,
18     tz='America/Fortaleza', # Fuso horario de Natal
19     altitude=altitude,
20     name='Natal' # Nome opcional para identificacao
21 )
22
23 # Combine 'data' e 'hora' em uma coluna datetime
24 # Convert 'data' and 'hora' to strings before concatenating
25 df_filtrado['datetime'] = pd.to_datetime(df_filtrado['data'].astype(str)
26     + ' ' + df_filtrado['hora'].astype(str))
27
28 # Check if the datetime column is already timezone-aware
29 if not df_filtrado['datetime'].dt.tz:
30     # If not, localize it to 'America/Fortaleza'
31     df_filtrado['datetime'] = df_filtrado['datetime'].dt.tz_localize('
32         America/Fortaleza')
33 else:
34     # If already timezone-aware, print a message or handle accordingly
35     print("The 'datetime' column is already timezone-aware.")
36
37 # Define como indice do DataFrame
38 df_filtrado.set_index('datetime', inplace=True)
39
40 # Calcula posicao solar (zenite, azimuth, etc.)
41 # Ensure 'Temperatura_Datalogger' column is numeric
42 df_filtrado['Temperatura_Datalogger'] = pd.to_numeric(df_filtrado['
43     Temperatura_Datalogger'], errors='coerce')
44
45 solar_pos = solarposition.get_solarposition(
46     time=df_filtrado.index,
47     latitude=latitude,
48     longitude=longitude,
49     altitude=altitude,
50     temperature=df_filtrado['Temperatura_Datalogger'], # Opcional: usa
51     temperatura local
52     pressure=None # Pode ser fornecido se disponivel
53 )
54
55 # Adiciona ao DataFrame original
56 df_filtrado['zenital'] = solar_pos['apparent_zenith'] # Angulo zenital
57     (graus)
58 df_filtrado['azimutal'] = solar_pos['azimuth'] # Angulo azimutal
59     (graus)
60
61 # Radiacao extraterrestre (W/m2)

```

```

56 df_filtrado['radiacao_extraterrestre'] = irradiance.get_extra_radiation(
    df_filtrado.index)

```

Listing 2: Cálculo do ângulo Zenital usando a PVLIB

Código 3 — Validando GHI

```

1  import pandas as pd
2  import numpy as np
3  import pvlib
4  from pvlib.location import Location
5  import matplotlib.pyplot as plt
6
7  # --- Localiza o ---
8  latitude = -5.836750      # Natal
9  longitude = -35.206478
10 altitude = 56
11 site = Location(latitude=latitude, longitude=longitude, altitude=
    altitude, tz='America/Sao_Paulo')
12
13 # --- Convers o de data/hora ---
14 df_filtrado['datetime'] = pd.to_datetime(df_filtrado['data'].astype(str)
    + ' ' + df_filtrado['hora'].astype(str))
15
16 # Garantir timezone correto
17 if not df_filtrado['datetime'].dt.tz:
18     df_filtrado['datetime'] = df_filtrado['datetime'].dt.tz_localize('
    America/Sao_Paulo')
19 else:
20     df_filtrado['datetime'] = df_filtrado['datetime'].dt.tz_convert('
    America/Sao_Paulo')
21
22 # Remove indices duplicados antes de setar o index para evitar o erro
23 df_filtrado = df_filtrado.drop_duplicates(subset=['datetime']).set_index
    ('datetime')
24
25 # --- Posi o solar ---
26 solar_position = site.get_solarposition(df_filtrado.index)
27 df_filtrado['zenith'] = solar_position['zenith']
28
29
30 # --- Posi o solar ---
31 solar_position = site.get_solarposition(df_filtrado.index)
32 df_filtrado['zenith'] = solar_position['zenith']
33
34 # --- Irradia o extraterrestre (G0) ---
35 df_filtrado['G0'] = pvlib.irradiance.get_extra_radiation(df_filtrado.
    index)
36
37 # --- Marcar ngulos zenitais < 15 ---
38 df_filtrado['zenith_menor_15'] = df_filtrado['zenith'] < 15
39
40 # --- Teste 1: GHI v lido entre 0 e 1.2 * G0 ---

```

```

41 df_filtrado['GHI_valido_teste1'] = (df_filtrado['glo_avg_2'].fillna(0)
    >= 0) & (df_filtrado['glo_avg_2'].fillna(0) <= 1.2 * df_filtrado['G0',
    ])
42 df_filtrado.loc[df_filtrado['glo_avg_2'].isna(), 'GHI_valido_teste1'] =
    np.nan
43
44 # --- Teste 2: GHI v lido entre 0 e 1.5 * G0 ---
45 df_filtrado['GHI_valido_teste2'] = (df_filtrado['glo_avg_2'].fillna(0)
    >= 0) & (df_filtrado['glo_avg_2'].fillna(0) <= 1.5 * df_filtrado['G0',
    ])
46 df_filtrado.loc[df_filtrado['glo_avg_2'].isna(), 'GHI_valido_teste2'] =
    np.nan
47
48 # --- Marcar valores inv lidos ---
49 df_filtrado['GHI_invalido_teste1'] = df_filtrado['GHI_valido_teste1'].eq
    (False) & df_filtrado['glo_avg_2'].notna()
50 df_filtrado['GHI_invalido_teste2'] = df_filtrado['GHI_valido_teste2'].eq
    (False) & df_filtrado['glo_avg_2'].notna()
51
52 # --- Substituir valores inv lidos por NaN (opcional) ---
53 df_filtrado['glo_avg_2_corrigido'] = df_filtrado['glo_avg_2'].where(
    df_filtrado['GHI_valido_teste1'].fillna(True), np.nan)
54
55 # --- Visualiza o ---
56 plt.figure(figsize=(12, 6))
57 plt.plot(df_filtrado.index, df_filtrado['glo_avg_2'], label='GHI
    Original', alpha=0.7)
58
59 # Pontos inv lidos (teste 1)
60 plt.scatter(df_filtrado.loc[df_filtrado['GHI_invalido_teste1'].fillna(
    False)].index,
61             df_filtrado.loc[df_filtrado['GHI_invalido_teste1'].fillna(
    False), 'glo_avg_2'],
62             color='red', label='Inv lido (Teste 1)', marker='x')
63
64 # Pontos inv lidos (teste 2)
65 plt.scatter(df_filtrado.loc[df_filtrado['GHI_invalido_teste2'].fillna(
    False)].index,
66             df_filtrado.loc[df_filtrado['GHI_invalido_teste2'].fillna(
    False), 'glo_avg_2'],
67             color='orange', label='Inv lido (Teste 2)', marker='x')
68
69 # Pontos com zenith < 15
70 plt.scatter(df_filtrado.loc[df_filtrado['zenith_menor_15']].index,
71             df_filtrado.loc[df_filtrado['zenith_menor_15'], 'glo_avg_2'
    ],
72             color='green', label='Zenith < 15 ', marker='o', s=20,
    edgecolors='black')
73
74 plt.xlabel('Data e Hora')
75 plt.ylabel('GHI (W/m )')
76 plt.title('Testes de Qualidade da GHI e ngulo Zenital < 15 ')
77 plt.legend()
78 plt.grid()
79 plt.tight_layout()
80 plt.show()
81
82 # --- Estat sticas ---

```

```

83 print("Valores NaN encontrados:", df_filtrado['glo_avg_2'].isna().sum())
84 print("Valores inv lidos no Teste 1:", df_filtrado['GHI_invalido_teste1
    '].sum())
85 print("Valores inv lidos no Teste 2:", df_filtrado['GHI_invalido_teste2
    '].sum())
86 print("Total de valores com ngulo zenital < 15 :", df_filtrado['
    zenith_menor_15'].sum())

```

Listing 3: Validacao para GHI

Código 4 — Validando DHI

```

1  import pandas as pd
2  import numpy as np
3  import pvlib
4  from pvlib.location import Location
5  import matplotlib.pyplot as plt
6
7  # Configuracao do local
8  latitude = -5.836750      # Latitude de Natal
9  longitude = -35.206478    # Longitude de Natal
10 altitude = 56             # Altitude (metros)
11
12 # Criar localizacao
13 site = Location(latitude=latitude, longitude=longitude, altitude=
    altitude, tz='America/Sao_Paulo')
14
15 # SOLUcao PARA OS WARNINGS:
16 # 1. Criar uma copia explicita do DataFrame para evitar
    SettingWithCopyWarning
17 df_validacao = df_filtrado.copy()
18
19 # 2. Converter data/hora para string e depois para datetime
20 try:
21     # Ensure 'data' is in date string format (YYYY-MM-DD)
22     df_validacao['data_str'] = df_validacao['data'].dt.strftime('%Y-%m-%
        d')
23     # Ensure 'hora' is in time string format (HH:MM:SS)
24     df_validacao['hora_str'] = df_validacao['hora'].dt.strftime('%H:%M:%
        S')
25
26     # Concatenate the string representations and convert to datetime
27     df_validacao['datetime'] = pd.to_datetime(
28         df_validacao['data_str'] + ' ' + df_validacao['hora_str'],
29         format='%Y-%m-%d %H:%M:%S'
30     )
31
32     # Remove the temporary string columns
33     df_validacao = df_validacao.drop(columns=['data_str', 'hora_str'])
34
35     # Remove duplicates if they exist
36     df_validacao = df_validacao.drop_duplicates(subset=['datetime'])
37
38     # Verify and adjust timezone
39     if not df_validacao['datetime'].dt.tz:

```

```

40     df_validacao['datetime'] = df_validacao['datetime'].dt.
        tz_localize('America/Sao_Paulo')
41     else:
42         df_validacao['datetime'] = df_validacao['datetime'].dt.
            tz_convert('America/Sao_Paulo')
43
44     # Set datetime as index (removes issues with duplicates)
45     df_validacao = df_validacao.set_index('datetime')
46
47 except Exception as e:
48     print(f"Erro ao processar datas: {e}")
49     raise
50
51 # Calcular o angulo zenital solar
52 try:
53     # Ensure the index is a DatetimeIndex before passing to
        get_solarposition
54     if not isinstance(df_validacao.index, pd.DatetimeIndex):
55         df_validacao.index = pd.to_datetime(df_validacao.index, utc=
            True).tz_convert('America/Sao_Paulo')
56
57     solar_position = site.get_solarposition(df_validacao.index)
58     df_validacao['zenith'] = solar_position['apparent_zenith']
59 except Exception as e:
60     print(f"Erro ao calcular posicao solar: {e}")
61     raise
62
63 # Calcular a irradiacao extraterrestre (G0)
64 df_validacao['G0'] = pvlib.irradiance.get_extra_radiation(df_validacao.
    index)
65
66 # Contar valores NaN antes dos testes
67 nan_count = df_validacao['dif_avg_2'].isna().sum()
68
69 ## Algoritmo 1: Limites fisicos para DHI
70 cos_zenith = np.cos(np.radians(df_validacao['zenith']))
71 df_validacao['DHI_max_teste1'] = 0.95 * df_validacao['G0'] * (cos_zenith
    **1.2) + 50
72 df_validacao['DHI_valido_teste1'] = (df_validacao['dif_avg_2'].fillna(0)
    >= 0) & \
73     (df_validacao['dif_avg_2'].fillna(0)
        <= df_validacao['DHI_max_teste1'])
    & \
74     (df_validacao['zenith'] <= 85)
75 df_validacao.loc[df_validacao['dif_avg_2'].isna(), 'DHI_valido_teste1']
    = np.nan
76
77 ## Algoritmo 2: Limites extremos para DHI
78 df_validacao['DHI_max_teste2'] = 1.2 * df_validacao['G0'] * (cos_zenith
    **1.2) + 100
79 df_validacao['DHI_valido_teste2'] = (df_validacao['dif_avg_2'].fillna(0)
    >= 0) & \
80     (df_validacao['dif_avg_2'].fillna(0)
        <= df_validacao['DHI_max_teste2'])
    & \
81     (df_validacao['zenith'] <= 85)
82 df_validacao.loc[df_validacao['dif_avg_2'].isna(), 'DHI_valido_teste2']
    = np.nan

```



```

83
84 ## Algoritmo 3: Relacao entre DHI e GHI (quando disponivel)
85 if 'glo_avg_2' in df_validacao.columns:
86     df_validacao['DHI_valido_teste3'] = (df_validacao['dif_avg_2'].
87         fillna(0) <= (df_validacao['glo_avg_2'].fillna(0) + 50))
88     df_validacao.loc[df_validacao['dif_avg_2'].isna(), '
89         DHI_valido_teste3'] = np.nan
90 else:
91     print("AVISO: Coluna 'glo_avg_2' nao encontrada. Teste 3 nao foi
92         aplicado.")
93
94 # Marcar valores invalidos
95 df_validacao['DHI_invalido_teste1'] = df_validacao['DHI_valido_teste1'].
96     eq(False) & df_validacao['dif_avg_2'].notna()
97 df_validacao['DHI_invalido_teste2'] = df_validacao['DHI_valido_teste2'].
98     eq(False) & df_validacao['dif_avg_2'].notna()
99
100 if 'DHI_valido_teste3' in df_validacao.columns:
101     df_validacao['DHI_invalido_teste3'] = df_validacao['
102         DHI_valido_teste3'].eq(False) & df_validacao['dif_avg_2'].notna()
103
104 # Substituir valores invalidos por NaN
105 df_validacao['dif_avg_2_corrigido'] = df_validacao['dif_avg_2'].where(
106     df_validacao['DHI_valido_teste1'].fillna(True), np.nan)
107
108 # Visualizacao
109 plt.figure(figsize=(12, 6))
110 plt.plot(df_validacao.index, df_validacao['dif_avg_2'], label='DHI
111     Original', alpha=0.7)
112
113 # Plotar pontos invalidos
114 colors = ['red', 'orange', 'purple']
115 tests = ['DHI_invalido_teste1', 'DHI_invalido_teste2', '
116     DHI_invalido_teste3']
117 labels = ['Inv lido (Teste 1)', 'Invalido (Teste 2)', 'Inv lido (Teste
118     3)']
119
120 for color, test, label in zip(colors, tests, labels):
121     if test in df_validacao.columns:
122         invalid_data = df_validacao[df_validacao[test]]
123         plt.scatter(invalid_data.index, invalid_data['dif_avg_2'], color
124             =color, label=label, marker='x')
125
126 plt.xlabel('Data e Hora')
127 plt.ylabel('DHI (W/m2)')
128 plt.title('Testes de Qualidade da DHI')
129 plt.legend()
130 plt.grid()
131 plt.tight_layout()
132 plt.show()
133
134 # Resumo estatistico
135 print("\nResumo dos Testes de Qualidade para DHI:")
136 print(f"Total de registros: {len(df_validacao)}")
137 print(f"Valores NaN encontrados: {nan_count} ({nan_count/len(
138     df_validacao):.1%})")
139

```

```

128 for i, test in enumerate(['DHI_invalido_teste1', 'DHI_invalido_teste2',
129     'DHI_invalido_teste3'], 1):
130     if test in df_validacao.columns:
131         count = df_validacao[test].sum()
132         print(f"Valores invalidos no Teste {i}: {count} ({count/len(
133             df_validacao):.1%})")
134
135 print("\nDados corrigidos salvos na coluna 'dif_avg_2_corrigido'")

```

Listing 4: Validação para DHI

Código 5 — Gráfico de Análise de valores suspeitos

```

1 import pandas as pd
2 import numpy as np
3 import pvlib
4 from pvlib.location import Location
5 import matplotlib.pyplot as plt
6 from matplotlib import colors
7
8 # Configuracao do local
9 latitude = -5.836750
10 longitude = -35.206478
11 altitude = 56
12 site = Location(latitude=latitude, longitude=longitude, altitude=
13     altitude, tz='America/Sao_Paulo')
14
15 ## 1. Verificacao do calculo de G0 (Irradiacao Extraterrestre)
16 # Calcular G0 com duas abordagens para comparacao
17 df_validacao['G0_pvlib'] = pvlib.irradiance.get_extra_radiation(
18     df_validacao.index)
19 df_validacao['G0_manual'] = 1367 * (1 + 0.033 * np.cos(np.radians(360 *
20     (df_validacao.index.dayofyear - 3)/365)))
21
22 # Plotar comparacao
23 plt.figure(figsize=(12, 5))
24 plt.plot(df_validacao.index, df_validacao['G0_pvlib'], label='G0 (pvlib)
25     ')
26 plt.plot(df_validacao.index, df_validacao['G0_manual'], '--', label='G0
27     (calculado manual)')
28 plt.title('Comparacao do calculo de G0')
29 plt.ylabel('Irradiacao Extraterrestre (W/m2)')
30 plt.legend()
31 plt.grid()
32 plt.tight_layout()
33 plt.show()
34
35 # Verificar diferencas
36 diff = (df_validacao['G0_pvlib'] - df_validacao['G0_manual']).abs()
37 print(f"\nDiferenca maxima entre metodos G0: {diff.max():.4f} W/m2")
38 print(f"Diferenca media: {diff.mean():.4f} W/m2")
39
40 ## 2. Analise de Dispersao GHI vs DHI com Valores Suspeitos
41 plt.figure(figsize=(10, 8))
42
43 # Criar mascara para valores suspeitos de DHI

```

```

39 mask_suspeito = (
40     (df_validacao['dif_avg_2'] > df_validacao['glo_avg_2']) | # DHI >
        GHI
41     (df_validacao['dif_avg_2'] < 0) | # DHI
        negativo
42     (df_validacao['zenith'] <= 85) & (df_validacao['dif_avg_2'] > 1000)
        # Valores muito altos
43 )
44
45 # Plotar dispersao com cores por angulo zenital
46 norm = colors.Normalize(vmin=0, vmax=90)
47 sc = plt.scatter(
48     df_validacao['glo_avg_2'],
49     df_validacao['dif_avg_2'],
50     c=df_validacao['zenith'],
51     cmap='viridis',
52     norm=norm,
53     alpha=0.6,
54     label='Dados normais'
55 )
56
57 # Destacar valores suspeitos
58 plt.scatter(
59     df_validacao.loc[mask_suspeito, 'glo_avg_2'],
60     df_validacao.loc[mask_suspeito, 'dif_avg_2'],
61     color='red',
62     marker='x',
63     s=50,
64     label='Dados suspeitos'
65 )
66
67 # Linhas de referencia
68 max_val = max(df_validacao['glo_avg_2'].max(), df_validacao['dif_avg_2']
69     ].max()) * 1.1
69 plt.plot([0, max_val], [0, max_val], 'k--', alpha=0.3, label='GHI = DHI',
70     )
70 plt.plot([0, max_val], [0, 0.8*max_val], 'g--', alpha=0.3, label='DHI =
71     0.8*GHI')
72
73 plt.xlabel('GHI (W/m2)')
74 plt.ylabel('DHI (W/m2)')
75 plt.title('Relacao GHI vs DHI (Cores pelo angulo Zenital)')
76 plt.colorbar(sc, label='angulo Zenital (graus)')
77 plt.legend()
78 plt.grid()
79 plt.tight_layout()
80 plt.show()
81
82 ## 3. Estatisticas dos Valores Suspeitos
83 print("\nEstatisticas dos Dados Suspeitos:")
84 print(f"Total de registros: {len(df_validacao)}")
85 print(f"Registros suspeitos: {mask_suspeito.sum()} ({mask_suspeito.mean
86     ()*100:.1f}%)")
87
88 if mask_suspeito.any():
89     print("\nCaracteristicas dos registros suspeitos:")
90     print(df_validacao.loc[mask_suspeito, ['glo_avg_2', 'dif_avg_2', '
91         zenith']].describe())

```

```

89
90     # Plotar serie temporal dos valores suspeitos
91     plt.figure(figsize=(12, 6))
92     plt.plot(df_validacao.index, df_validacao['dif_avg_2'], 'b-', alpha
93             =0.3, label='DHI')
94     plt.scatter(
95         df_validacao.loc[mask_suspeito].index,
96         df_validacao.loc[mask_suspeito, 'dif_avg_2'],
97         color='red',
98         label='Valores suspeitos'
99     )
100     plt.title('Serie Temporal com Valores Suspeitos Destacados')
101     plt.ylabel('DHI (W/m )')
102     plt.legend()
103     plt.grid()
104     plt.tight_layout()
105     plt.show()
106
107     ## 4. Relacao KT (Clearness Index) vs KD (Diffuse Fraction)
108     df_validacao['KT'] = df_validacao['glo_avg_2'] / df_validacao['G0_pvlib']
109     # indice de claridade
110     df_validacao['KD'] = df_validacao['dif_avg_2'] / df_validacao['glo_avg_2']
111     # Fracao difusa
112
113     # Filtrar para angulos zenitais razoaveis e GHI > 0
114     mask_analysis = (df_validacao['zenith'] <= 75) & (df_validacao['glo_avg_2'] > 10)
115     df_analysis = df_validacao.loc[mask_analysis].copy()
116
117     # Plotar relacao esperada KT vs KD
118     plt.figure(figsize=(10, 6))
119     plt.scatter(
120         df_analysis['KT'],
121         df_analysis['KD'],
122         c=df_analysis['zenith'],
123         cmap='viridis',
124         norm=norm,
125         alpha=0.6
126     )
127
128     # Destacar valores suspeitos
129     plt.scatter(
130         df_analysis.loc[mask_suspeito & mask_analysis, 'KT'],
131         df_analysis.loc[mask_suspeito & mask_analysis, 'KD'],
132         color='red',
133         marker='x',
134         s=50,
135         label='Dados suspeitos'
136     )
137
138     # Linhas de referencia
139     plt.axhline(y=1, color='r', linestyle='--', alpha=0.3)
140     plt.axvline(x=0.8, color='g', linestyle='--', alpha=0.3)
141     plt.xlabel('Clearness Index (KT)')
142     plt.ylabel('Diffuse Fraction (KD)')
143     plt.title('Relacao KT vs KD (Cores pelo angulo Zenital)')
144     plt.colorbar(sc, label='angulo Zenital (graus)')
145     plt.grid()

```

```

143 plt.tight_layout()
144 plt.show()

```

Listing 5: Plotagem dos dados com valores suspeitos

Código 6 — Gráfico de Análise KT vs KD

```

1 import pandas as pd
2 import numpy as np
3 import pvlib
4 from pvlib.location import Location
5 import matplotlib.pyplot as plt
6 from matplotlib import colors
7
8 # Configura o do local
9 latitude = -5.836750
10 longitude = -35.206478
11 altitude = 56
12 site = Location(latitude=latitude, longitude=longitude, altitude=
    altitude, tz='America/Sao_Paulo')
13
14 ## 4. Relação KT (Clearness Index) vs KD (Diffuse Fraction) - Dados
    Corrigidos
15 df_corrigido['KT'] = df_corrigido['glo_avg_2'] / (df_corrigido['G0_pvlib
    ']*np.cos(np.radians(df_corrigido['zenith']))) # índice de
    claridade
16 df_corrigido['KD'] = df_corrigido['dif_avg_2'] / df_corrigido['glo_avg_2
    ']' # Fração difusa
17
18 # Filtrar dados confiáveis com KT e KD entre 0 e 1
19 mask_analysis = (
20     (df_corrigido['zenith'] <= 75) &
21     (df_corrigido['glo_avg_2'] > 10) &
22     (df_corrigido['KT'].notnull()) &
23     (df_corrigido['KD'].notnull()) &
24     (df_corrigido['KT'] >= 0) & (df_corrigido['KT'] <= 1) &
25     (df_corrigido['KD'] >= 0) & (df_corrigido['KD'] <= 1)
26 )
27
28 df_analysis = df_corrigido.loc[mask_analysis].copy()
29
30
31 # Plotar relação esperada KT vs KD
32 plt.figure(figsize=(10, 6))
33 plt.scatter(
34     df_analysis['KT'],
35     df_analysis['KD'],
36     c=df_analysis['zenith'],
37     cmap='viridis',
38     norm=norm,
39     alpha=0.6,
40     s=5,
41     label='Dados válidos'
42 )
43
44 # Destacar valores suspeitos residuais

```

```

45 if mask_suspeito_residual.any():
46     plt.scatter(
47         df_analysis.loc[mask_suspeito_residual & mask_analysis, 'KT'],
48         df_analysis.loc[mask_suspeito_residual & mask_analysis, 'KD'],
49         color='red',
50         marker='x',
51         s=50,
52         label='Dados suspeitos residuais'
53     )
54
55 # Linhas de referência
56 plt.axhline(y=1, color='r', linestyle='--', alpha=0.3)
57 plt.axvline(x=0.8, color='g', linestyle='--', alpha=0.3)
58 plt.xlabel('Clearness Index (KT)')
59 plt.ylabel('Diffuse Fraction (KD)')
60 plt.title('Relação KT vs KD - Dados Corrigidos (Cores pelo ângulo
61         Zenital)')
62 plt.colorbar(sc, label='ângulo Zenital (graus)')
63 plt.legend()
64 plt.grid()
65 plt.tight_layout()
66 plt.show()

```

Listing 6: Gráfico de Análise KT vs KD

Código 7 — Gráfico de Análise KT vs KD com Erbs

```

1 import matplotlib.pyplot as plt
2 import seaborn as sns
3 import numpy as np
4 from scipy.stats import gaussian_kde
5
6 # Filtro já feito
7 df_kt_kd = df_analysis.copy()
8
9 # Função da curva de Erbs
10 def erbs_diffuse_fraction(Kt):
11     if Kt <= 0.22:
12         return 1 - 0.09 * Kt
13     elif Kt <= 0.8:
14         return 0.9511 - 0.1604 * Kt + 4.388 * Kt**2 - 16.638 * Kt**3 +
15             12.336 * Kt**4
16     else:
17         return 0.165
18
19 # Dados para densidade
20 x = df_kt_kd['KT'].values
21 y = df_kt_kd['KD'].values
22
23 # Calcular densidade 2D
24 xy = np.vstack([x, y])
25 z = gaussian_kde(xy)(xy)
26
27 # Ordenar para desenhar os pontos mais densos por cima
28 idx = z.argsort()
29 x, y, z = x[idx], y[idx], z[idx]

```

```

29
30 # Plot
31 plt.figure(figsize=(10, 6))
32 sc = plt.scatter(x, y, c=z, s=5, cmap='viridis', alpha=0.6)
33 plt.xlabel('Índice de Claridade (Kt)')
34 plt.ylabel('Fração Difusa (Kd)')
35 plt.title('Dispersão da Fração Difusa (Kd) vs. Índice de Claridade (Kt) - Cores por Densidade')
36 plt.grid(True)
37
38 # Curva de Erbs
39 erbs_kt_range = np.linspace(0, 1, 200)
40 erbs_kd_values = [erbs_diffuse_fraction(kt) for kt in erbs_kt_range]
41 plt.plot(erbs_kt_range, erbs_kd_values, color='red', label='Curva de Erbs', linewidth=2)
42
43 # Colorbar e legenda
44 cbar = plt.colorbar(sc)
45 cbar.set_label('Densidade')
46 plt.legend()
47 plt.tight_layout()
48 plt.show()

```

Listing 7: Gráfico de Análise KT vs KD com Erbs

Código 8 — Gráfico de Análise KT vs KD com Ajuste Polinomial

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from numpy.polynomial import Polynomial
4 from scipy.stats import gaussian_kde
5
6 # Dados (já filtrados)
7 x = df_kt_kd['KT'].values
8 y = df_kt_kd['KD'].values
9
10 # Densidade
11 xy = np.vstack([x, y])
12 z = gaussian_kde(xy)(xy)
13 idx = z.argsort()
14 x, y, z = x[idx], y[idx], z[idx]
15
16 # Ajuste polinomial (grau 4 geralmente suficiente)
17 grau = 4
18 coef = np.polyfit(x, y, grau)
19 p = np.poly1d(coef)
20
21 # Curva ajustada
22 x_fit = np.linspace(min(x), max(x), 500)
23 y_fit = p(x_fit)
24
25 # Forçar os limites entre 0 e 1
26 y_fit = np.clip(y_fit, 0, 1)
27

```

```

28 # Plot
29 plt.figure(figsize=(10, 6))
30 sc = plt.scatter(x, y, c=z, s=5, cmap='viridis', alpha=0.6, label='Dados
    ')
31 plt.plot(x_fit, y_fit, color='red', lw=2, label=f'Ajuste Polinomial (
    grau {grau})')
32
33 plt.xlabel(' ndice de Claridade (Kt)')
34 plt.ylabel('Fra o Difusa (Kd)')
35 plt.title('Kd vs Kt - Ajuste Polinomial Suave')
36 plt.grid(True)
37 cbar = plt.colorbar(sc)
38 cbar.set_label('Densidade')
39 plt.legend()
40 plt.tight_layout()
41 plt.show()
42
43 # Coeficientes do polin mio (opcional)
44 print("Coeficientes do polin mio (grau {}):".format(grau))
45 print(p)
46
47 from sklearn.metrics import mean_squared_error
48
49 # Previs es nos mesmos pontos de x originais
50 y_pred = p(x)
51
52 # For ar limites [0, 1] como feito na curva
53 y_pred = np.clip(y_pred, 0, 1)
54
55 # RMSE
56 rmse = np.sqrt(mean_squared_error(y, y_pred))
57
58 # Vi s (bias)
59 bias = np.mean(y_pred - y)
60
61 # Exibir resultados
62 print(f"RMSE: {rmse:.4f}")
63 print(f"Vi s: {bias:.4f}")

```

Listing 8: Gráfico de Análise KT vs KD com Ajuste Polinomial

Código 9 — Gráfico de Análise KT vs KD com Random Forest

```

1 import matplotlib.pyplot as plt
2 import seaborn as sns
3 import numpy as np
4 from scipy.stats import gaussian_kde
5 from scipy.ndimage import uniform_filter1d
6 from sklearn.ensemble import RandomForestRegressor
7
8 # Filtro j feito
9 df_kt_kd = df_analysis.copy()
10
11 # Dados

```



```

12 x = df_kt_kd['KT'].values
13 y = df_kt_kd['KD'].values
14
15 # Densidade 2D para colorir os pontos
16 xy = np.vstack([x, y])
17 z = gaussian_kde(xy)(xy)
18
19 # Ordenar por densidade
20 idx = z.argsort()
21 x, y, z = x[idx], y[idx], z[idx]
22
23 # --- Random Forest ---
24 rf = RandomForestRegressor(
25     n_estimators=100,
26     min_samples_leaf=20,
27     random_state=42
28 )
29
30 # Treina modelo (precisa de reshape)
31 X_train = x.reshape(-1, 1)
32 y_train = y
33 rf.fit(X_train, y_train)
34
35 # Espa o suavizado de KT para prever KD
36 x_fit = np.linspace(min(x), max(x), 500).reshape(-1, 1)
37 y_fit = rf.predict(x_fit)
38
39 # Suaviza o da curva prevista
40 from scipy.ndimage import uniform_filter1d
41 y_fit_smooth = uniform_filter1d(y_fit, size=30) # aumente o size para
    suavizar mais
42
43 # --- Plot ---
44 plt.figure(figsize=(10, 6))
45 sc = plt.scatter(x, y, c=z, s=5, cmap='viridis', alpha=0.6, label='Dados
    ')
46 plt.plot(x_fit, y_fit_smooth, color='red', lw=2, label='Random Forest (
    suavizado)')
47
48 plt.xlabel('ndice de Claridade (Kt)')
49 plt.ylabel('Fra o Difusa (Kd)')
50 plt.title('Random Forest - KT vs KD com Suaviza o')
51 plt.grid(True)
52 cbar = plt.colorbar(sc)
53 cbar.set_label('Densidade')
54 plt.legend()
55 plt.tight_layout()
56 plt.show()
57
58 from sklearn.metrics import mean_squared_error
59
60 # Previs es no mesmo espa o dos dados originais (para calcular erro)
61 y_pred = rf.predict(X_train)
62
63 # RMSE
64 rmse = np.sqrt(mean_squared_error(y_train, y_pred))
65
66 # Vi s (m dia do erro)

```

```

67 bias = np.mean(y_pred - y_train)
68
69 # Exibir
70 print(f"RMSE: {rmse:.4f}")
71 print(f"Vi s: {bias:.4f}")

```

Listing 9: Gráfico de Análise KT vs KD com Random Forest

Código 10 — Gráfico da Distribuição Pontual da Radiação Difusa por Hora do Dia

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3  from scipy.stats import gaussian_kde
4  # Preparar os dados
5  hours = df_corrigido.index.hour + df_corrigido.index.minute/60
6  dhi_values = df_corrigido['dif_avg_2'].values
7  valid_mask = ~np.isnan(dhi_values)
8
9  # Calcular a densidade dos pontos
10 xy = np.vstack([hours[valid_mask], dhi_values[valid_mask]])
11 z = gaussian_kde(xy)(xy)
12
13 # Criar o grafico
14 plt.figure(figsize=(14, 8))
15
16 # Plotar cada ponto com cor baseada na densidade
17 sc = plt.scatter(
18     hours[valid_mask],
19     dhi_values[valid_mask],
20     c=z,
21     cmap='viridis',
22     alpha=0.6,
23     s=10,
24     edgecolors='none'
25 )
26
27 # Configuracoes do grafico
28 plt.title('Distribuicao Pontual da Radiacao Difusa por Hora do Dia',
29     fontsize=14)
30 plt.xlabel('Hora do Dia', fontsize=12)
31 plt.ylabel('Radiacao Difusa (DHI) (W/m2)', fontsize=12)
32 cbar = plt.colorbar(sc)
33 cbar.set_label('Densidade de Pontos')
34
35 # Linhas de referencia
36 plt.axhline(y=1000, color='r', linestyle='--', alpha=0.5, linewidth=1,
37     label='Limite fisico (1000 W/m2)')
38 plt.axvline(x=12, color='k', linestyle='--', alpha=0.5, linewidth=1,
39     label='Meio-dia solar')
40
41 # Ajustar eixos
42 plt.xlim(4, 20)
43 plt.ylim(0, min(1200, np.nanmax(dhi_values)*1.1))

```

```

42 plt.legend()
43 plt.grid(alpha=0.2)
44 plt.tight_layout()
45 plt.show()

```

Listing 10: Distribuição Pontual da Radiação Difusa por Hora do Dia

Código 11 — Gráfico do CALENDÁRIO SOLAR

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3  from matplotlib.colors import LogNorm, LinearSegmentedColormap
4  from mpl_toolkits.axes_grid1 import make_axes_locatable
5  from pvlib.solarposition import get_solarposition
6
7  # Coordenadas da esta o
8  latitude = -5.795
9  longitude = -35.209
10
11 # Calcular nascer/p r do sol m dio por m s
12 monthly_sun = {}
13 for month in range(1, 13):
14     month_data = df_corrigido[df_corrigido.index.month == month]
15     if not month_data.empty:
16         elevations = get_solarposition(month_data.index, latitude,
17                                         longitude)['elevation']
18         sunrise = month_data[elevations > 0].index.min()
19         sunset = month_data[elevations > 0].index.max()
20         monthly_sun[month] = (
21             sunrise.hour + sunrise.minute / 60,
22             sunset.hour + sunset.minute / 60
23         )
24
25 # Preparar matriz de radia o
26 df_corrigido['Dia_Ano'] = df_corrigido.index.dayofyear
27 df_corrigido['Hora_Decimal'] = df_corrigido.index.hour + df_corrigido.
28     index.minute / 60
29 horas = np.linspace(0, 23.958, 288)
30 dias = np.arange(152, 366) # Apenas de junho at dezembro
31 rad_matrix = np.full((len(horas), len(dias)), np.nan)
32
33 for i, dia in enumerate(dias):
34     dia_data = df_corrigido[df_corrigido['Dia_Ano'] == dia]
35     if not dia_data.empty:
36         for j, hora in enumerate(horas):
37             hora_mask = (abs(dia_data['Hora_Decimal'] - hora) < 0.042)
38             if hora_mask.any():
39                 rad_matrix[j, i] = dia_data.loc[hora_mask, 'glo_avg_2'].
40                     median()
41
42 rad_matrix = np.nan_to_num(rad_matrix, nan=10)
43
44 # Plot
45 fig, ax = plt.subplots(figsize=(20, 10))
46
47 # Heatmap

```

```

45 colors = ["#00008B", "#0000FF", "#0080FF", "#00BFFF", "#00FFFF",
46           "#80FF00", "#FFFF00", "#FFA500", "#FF0000", "#800000"]
47 cmap = LinearSegmentedColormap.from_list("rad", colors, N=1024)
48 im = ax.imshow(rad_matrix, aspect='auto', cmap=cmap,
49               norm=LogNorm(vmin=10, vmax=1100),
50               extent=[152, 365, 23.958, 0], interpolation='bilinear')
51
52 # Marcadores horizontais mensais (junho a dezembro)
53 month_positions = np.linspace(166, 350, 7) # Posi o m dia dos meses
54   de junho a dezembro
55 meses = ['Jun', 'Jul', 'Ago', 'Set', 'Out', 'Nov', 'Dez']
56
57 for i, month in enumerate(range(6, 13)): # Junho (6) a Dezembro (12)
58     if month in monthly_sun:
59         sunrise, sunset = monthly_sun[month]
60         sunrise_adj = min(sunrise + 3, 23.958)
61         sunset_adj = min(sunset + 3, 23.958)
62         pos = month_positions[i]
63         ax.plot(pos, sunrise_adj, 'o', color='lime', markersize=6, alpha
64               =0.7)
65         ax.plot(pos, sunset_adj, 'o', color='red', markersize=6, alpha
66               =0.7)
67         ax.hlines(y=[sunrise_adj, sunset_adj], xmin=pos - 10, xmax=pos +
68               10,
69               colors=['lime', 'red'], linestyle='--', alpha=0.3,
70               linewidths=1)
71
72 # Eixo Y
73 ax.set_yticks(np.arange(0, 24, 2))
74 ax.set_yticklabels([f"{h:02d}:00" for h in range(0, 24, 2)])
75 ax.set_ylabel('Hora do Dia', fontsize=12)
76 ax.set_ylim(23.958, 0)
77
78 # Eixo X
79 ax.set_xticks(month_positions)
80 ax.set_xticklabels(meses)
81 ax.set_xlabel('M s do Ano', fontsize=12)
82
83 # Legenda dos pontos
84 ax.plot([], [], 'o', color='lime', label='Nascer do sol m dio')
85 ax.plot([], [], 'o', color='red', label='P r do sol m dio')
86 ax.legend(loc='upper right')
87
88 # Eixo secund rio com radia o m xima
89 def hora_to_rad(h):
90     idx = np.abs(horas - h).argmin()
91     return np.max(rad_matrix[idx, :])
92
93 divider = make_axes_locatable(ax)
94 ax_right = divider.append_axes("right", size="3%", pad=0.2)
95 ax_right.set_ylabel('Radia o M xima (W/m )', fontsize=12)
96 ax_right.set_yticks(np.arange(0, 24, 2))
97 ax_right.set_yticklabels([f"{hora_to_rad(h):.0f}" for h in np.arange(0,
98   24, 2)])
99 ax_right.set_ylim(23.916, 0)

```

```

97 # Barra de cores
98 cbar = fig.colorbar(im, ax=ax, pad=0.02)
99 cbar.set_label('Radiação Global Horizontal (GHI)\nEscala Logarítmica
    [10-1100 W/m2 ]',
100               rotation=270, labelpad=25)
101
102 # Título
103 ax.set_title('CALENDÁRIO SOLAR - JUNHO A DEZEMBRO\nVariação Horária
    e Sazonal da Radiação\n(Valores <10 W/m2 mostrados como 10)',
104             fontsize=16, pad=20, fontweight='bold')
105
106 # Eventos astronômicos (a partir de junho)
107 events = {
108     'Solstício Junho': 172,
109     'Equinócio Set.': 266,
110     'Solstício Dez.': 355
111 }
112 for name, day in events.items():
113     ax.axvline(x=day, color='gold', linestyle='--', alpha=0.7)
114     ax.text(day, 24.5, name, rotation=90, ha='center', va='top',
115            color='gold', fontsize=10)
116
117 plt.tight_layout()
118 plt.show()

```

Listing 11: CALENDÁRIO SOLAR