



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  
2024

---

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, 2024.

Relatório de Iniciação Científica -  
Instituto Nacional de Pesquisas Espaciais, Natal, 2024.  
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

*“Pois onde estiver o seu tesouro, aí também estará o seu coração”.*

*Mateus 6:21 - Bíblia Sagrada NVI*

Primeiramente a Deus, minha família, aos colegas de trabalho e aos amigos 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 comparar duas técnicas de medição da irradiância solar difusa (DHI) na unidade do COENE-INPE em Natal-RN: o uso de piranômetro com anel de sombreamento e o piranômetro sobre rastreador solar sombreado por esferas. A pesquisa justifica-se pela necessidade de métodos precisos para a medição da irradiância solar difusa, fator crucial para a estimativa de geração de energia em usinas fotovoltaicas. Para isso, foram coletados dados ao longo de 2023/2024, de irradiância difusa e global dentre outras variáveis meteorológicas na estação solarimétrica instalada em Natal-RN. O estudo enfrentou atrasos por problemas técnicos relacionados a um rastreador solar inoperante. Este evento limitou as análises aos dados coletados pelo anel de sombreamento, onde foram analisadas a fração difusa da radiação solar e o índice de claridade. Uma análise estatística das medições e a implementação de novos critérios de qualidade também foram realizados visando a melhoria dos algoritmos de correção e a confiabilidade dos dados. Os resultados mostram as séries temporais de DHI ajustadas conforme fatores de correção hemisférica do instrumento e um ajuste adequado a um modelo de partição simples de Erbs. Espera-se que com a recuperação do rastreador solar as comparações previstas sejam realizadas no próximo ciclo. Este estudo contribui para o avanço das técnicas de medição de irradiância solar difusa, fundamentais para a otimização de sistemas de energia solar, dentre outras aplicações para a sociedade.

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

## **ABSTRACT**

The present report aims to compare two techniques for measuring diffuse horizontal irradiance (DHI) at the COENE-INPE unit in Natal-RN: the use of a pyranometer with a shading ring and a pyranometer on a solar tracker shaded by spheres. The research is justified by the need for accurate methods for measuring diffuse solar irradiance, which is crucial for estimating energy generation in photovoltaic plants. Data were collected throughout 2023/2024, including diffuse and global irradiance and other meteorological variables at the solarimetric station installed in Natal-RN. The study faced delays due to technical problems related to an inoperative solar tracker. This event limited the analyses to the data collected by the shading ring, where the diffuse fraction of solar radiation and the clearness index were analyzed. A statistical analysis of the measurements and the implementation of new quality criteria were also carried out to improve correction algorithms and data reliability. The results show the time series of DHI adjusted according to the instrument's hemispherical correction factors and adequately fitted to a simple Erbs partition model. It is expected that with the recovery of the solar tracker, the planned comparisons will be carried out in the next cycle. This study contributes to the advancement of diffuse solar irradiance measurement techniques, essential for optimizing solar energy systems, among other applications for society.

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

## LISTA DE TABELAS

	<u>Pág.</u>
Tabela 1 - Equipamentos Bancada Leste.....	8
Tabela 2 - Equipamentos Bancada Leste.....	8
Tabela 3 - Equipamentos Bancada Sul.....	9
Tabela 4 - Equipamentos Bancada Leste.....	10
Tabela 5 - Tentativas de conexão.....	11
Tabela 6 - Ajuste de curva, desvio padrão e RMSE para o modelo de Erbs no gráfico de dispersão (autoria própria).....	20



## LISTA DE FIGURAS E GRÁFICOS

	<u>Pág.</u>
Figura 1 - Foto da Face Norte - Câmera AllSky (Autoria própria).....	8
Figura 2 - Foto da Bancada Leste (Autoria própria).....	9
Figura 3 - Foto da Bancada Oeste (Autoria própria).....	9
Figura 4 - Foto da Bancada Oeste (Autoria própria).....	10
Figura 5 - Print LoggerNet (autoria própria).....	12
Figura 6 - imagem do switch TL-SG1024 (TP-Link, 2024).....	12
Figura 7 - SOLYS2 Sun Tracker (KIPP & ZONEN, 2024).....	13
Figura 8: Rastreador solr Solys2 com correias danificadas (autoria de Dr. André Rodrigues).....	14
Gráfico 1 - Quantidade de dados coletados por ano (autoria própria).....	15
Gráfico 2 - Quantidade de dados coletados por ano (autoria própria).....	16
Gráfico 3 - Histograma das irradiâncias GHI (acima) e DHI (abaixo) (autoria própria)	17
Gráfico 4 - Curva de Erbs para a dispersão da fração difusa no ano de 2015 (autoria própria).....	18
Gráfico 5 - Dispersão da fração difusa no ano de 2015 com curva ajustada de Erbs (autoria própria).....	18
Gráfico 6 - Ampliação do gráfico 5 (autoria própria).....	19

## **LISTA DE SIGLAS E ABREVIATURAS**

INPE Instituto Nacional de Pesquisas Espaciais

SONDA Sistema de Organização Nacional de Dados Ambientais

LAVAT Laboratório de Variáveis Ambientais

## 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 Relação entre a fração difusa da radiação solar e o índice de claridade no ano de 2015.....	6
2.2.2 Modelo de Erbs (1982).....	7
2.2.3 Regressão logística.....	7
2.2.4 Disposição dos equipamentos nas bancadas.....	7
2.2.4.1 Face Norte.....	7
2.2.4.2 Bancada Leste.....	8
2.2.4.3 Bancada Sul.....	9
2.2.4.4 Bancada Oeste.....	10
2.2.5 Problemas encontrados nos equipamentos.....	11
2.2.5.1 Conexão de rede.....	11
2.2.5.2 Cabo de rede.....	12
2.2.5.3 Rastreador inoperante.....	13
3 RESULTADOS.....	15
3.1 Conjunto de dados coletados ao longo dos anos 2007-2022.....	15
3.2 Distribuição de dados de Difusa e Global.....	16
4 CONCLUSÃO.....	21
5 REFERÊNCIAS BIBLIOGRÁFICAS.....	22
ANEXO.....	24

## 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).

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 usando o protocolado sonda, conforme descrito na página oficial de validação do CCST/INPE (SONDA, 2024).

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

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

### 2.2.1 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.

Outro estudo significativo é "*New Models for Separating Hourly Diffuse and Direct Components of Global Solar Radiation*", que propõe diversos

modelos para estimar a radiação solar difusa com base na radiação global. Esse estudo enfatiza a importância de prever  $K_d$  a partir de  $K_t$  e valida diferentes métodos para essa previsão (MARTÍNEZ-DURÁ et al., 2020).

A medição da Irradiância Solar Global Horizontal (GHI) em Natal permite a aplicação do modelo de Erbs (1982) para a estimativa da fração de radiação difusa. Com essa estimativa, é possível calcular a irradiância no plano de captação dos painéis solares (*PoA: Plane of Array*). O modelo de Erbs também é aplicável em diversas áreas, como agricultura, iluminação e arquitetura bioclimática, entre outras.

### 2.2.2 Modelo de Erbs (1982)

O modelo de Erbs é uma maneira de correlacionar a fração de radiação difusa ( $K_d$ ) com o índice de claridade ( $K_t$ ). É uma ferramenta importante em meteorologia e energia solar, pois ajuda a estimar a quantidade de radiação solar que chega à superfície terrestre de forma difusa, o que é crucial para o dimensionamento de sistemas de energia solar.

### 2.2.3 Regressão logística

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

$P(x)$  é a fração difusa ( $K_d$ )

$x$  é o índice de claridade ( $K_t$ )

$\beta_0$  e  $\beta_1$  são parâmetros ajustados para melhor descrever a relação entre  $K_d$  e  $K_t$ .

### 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)

Tabela 1 - Equipamentos Bancada Leste



Figura 1 - Foto da Face Norte - Câmera AllSky (Autoria própria)

#### 2.2.4.2 Bancada Leste

Equipamento	Quantidade	Função
Sensor de Luminosidade	4 (Quatro)	Medir a duração do dia através da Luminosidade (Lúmens)
Termo-higrômetro	1 (Um)	Medição de temperatura do ar em abrigo (°C)
Caixa DataLogger	1 (Um)	Coletar, converter e registrar os sinais elétricos em grandezas físicas

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.

Tabela 3 - Equipamentos Bancada Sul

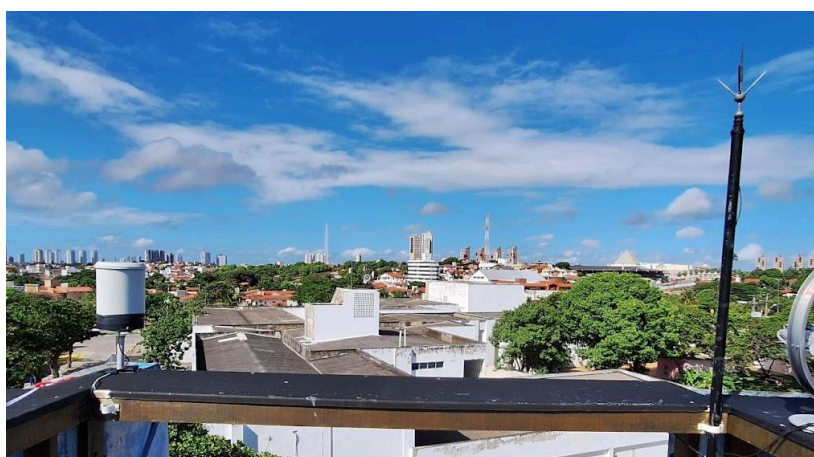


Figura 3 - Foto da Bancada Oeste (*Autoria própria*)

#### 2.2.4.4 Bancada Oeste

Equipamento	Quantidade	Função
Radiação Difusa	1 (Um)	Medir a radiação na sombra ( $\text{W/m}^2$ )
Radiação Global e Par	1 (Um)	Medir a radiação na superfície exposta ao sol ( $\text{W/m}^2$ )
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

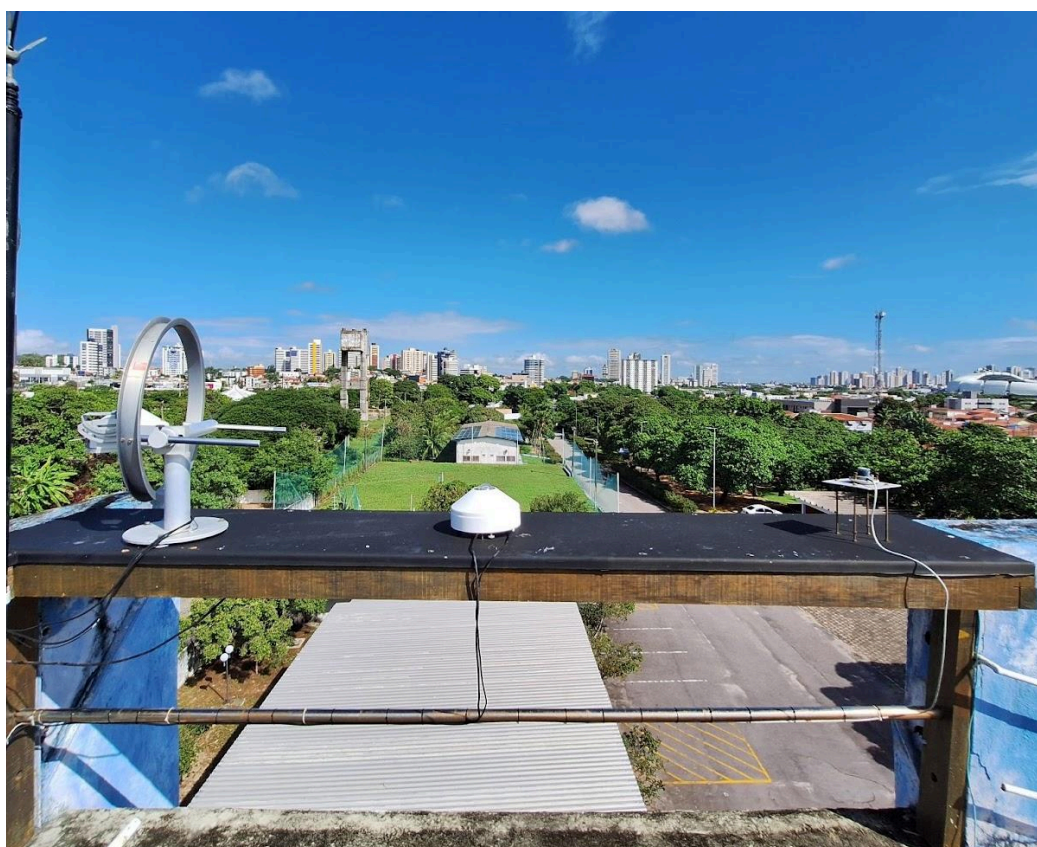


Figura 4 - Foto da Bancada Oeste (Autoria própria)

## 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. Além disso, o anel desalinhado resultou em erros na coleta de dados, enquanto o datalogger com falhas no relógio interno afetou a sincronização e o registro preciso dos dados. 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, segue abaixo a tabela 5 com as últimas manutenções feitas antes da implementação do novo *DataLogger* com o objetivo de retomada de conexão:

Data	Conexão
06/05/2024	Falhou
13/05/2024	Falhou
20/05/2024	Falhou

Tabela 5 - Tentativas de conexão

Até o dia 23/05/2024 às 14:18, ainda não foi possível conectar a estação, a figura 5 mostra o print da tentativa de conexão.



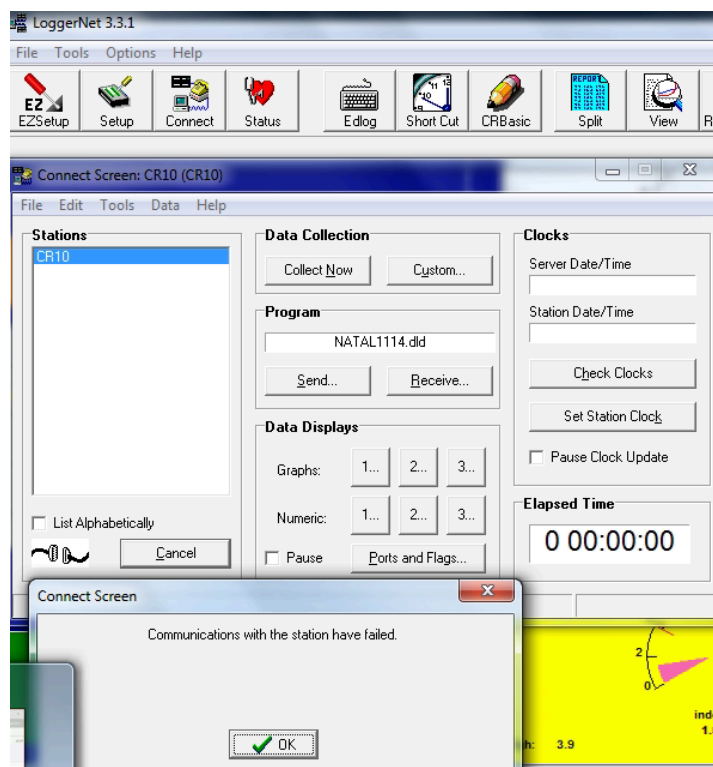


Figura 5 - Print LoggerNet (autoria própria)

### 2.2.5.2 Cabo de rede

Segundo servidores mais antigos, o cabo de rede que conectava o switch de rede conforme mostrado na figura 6 com a imagem do switch TL-SG1024 (TP-Link, 2024), até a estação está a quase duas décadas sem ser trocado, uma das possibilidades para a perda de conexão é decorrente do ressecamento e envelhecimento deste material, visto que está exposto a tanto tempo aos efeitos ambientais.



Figura 6 - imagem do switch TL-SG1024 (TP-Link, 2024)

Portanto foi proposto e sugerido uma troca do antigo cabo para um novo da categoria CAT6, para ter melhor qualidade de sinal e de durabilidade.

Foi feita uma medição da distância que o cabo deve percorrer, essas distâncias foram de 36 metros horizontais e 13 metros verticais, totalizando 49

metros. Portanto devemos adquirir um cabo de 60 metros para termos margem de segurança.

O cabo sugerido pelo Dr. Francisco Raimundo da Silva, Técnico no LAVAT foi o CAT-6, e o ideal seria que fosse blindado para atenuar possíveis ruídos pela radiação solar.

### 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 7 - SOLYS2 Sun Tracker (KIPP & ZONEN, 2024)

O equipamento após ser içado e instalado na bancada leste apresentou falha, a função de se movimentar horizontalmente não estava sendo executada, comprometendo portanto a funcionalidade do equipamento. Foi diagnosticado que houve rompimento das correias internas que movimentam as engrenagens, provavelmente ocorrido durante o traslado conforme Figura 8. O equipamento foi remetido para o INPE-SJC para manutenção corretiva.



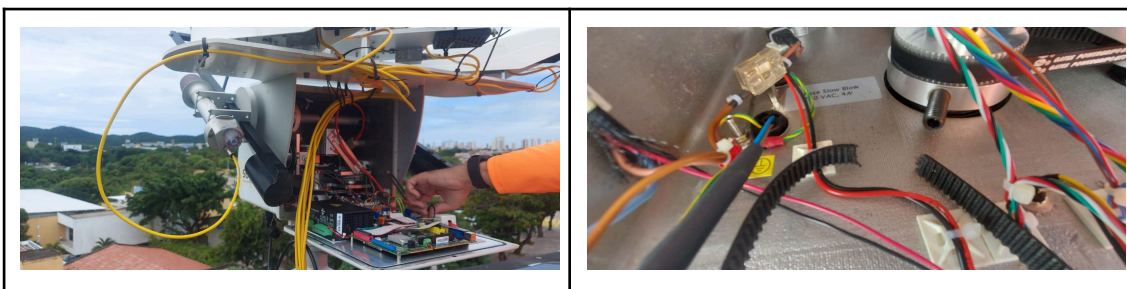


Figura 8: Rastreador solr Solys2 com correias danificadas (autoria de Dr. André Rodrigues).

O *Solys 2*, desenvolvido pela *KIPP & ZONEN*, é um rastreador solar avançado projetado para monitoramento da radiação solar. Utiliza sensores de alta precisão para seguir o movimento do sol ao longo do dia, garantindo que os sistemas de medição e os sensores meteorológicos obtenham dados precisos de radiação solar global e direta. Ideal para aplicações em estações meteorológicas, esta tecnologia ajuda a otimizar a coleta de dados meteorológicos e ambientais em diversas condições climáticas (*KIPP & ZONEN*, 2024).

### 3 RESULTADOS

#### 3.1 Conjunto de dados coletados ao longo dos anos 2007-2022

Foi desenvolvido um código que utiliza *Python* com as bibliotecas *pandas*, *matplotlib* e *tkinter* para realizar a análise de dados de um arquivo CSV. Primeiramente, o usuário seleciona uma pasta onde está localizado o arquivo CSV através de uma interface gráfica. Após a seleção, o código carrega o arquivo CSV, filtrando e limpando os dados para manter apenas a coluna que representa o ano. Em seguida, são contabilizados os dados por ano e um gráfico de barras é gerado para visualizar a distribuição desses dados ao longo dos anos.

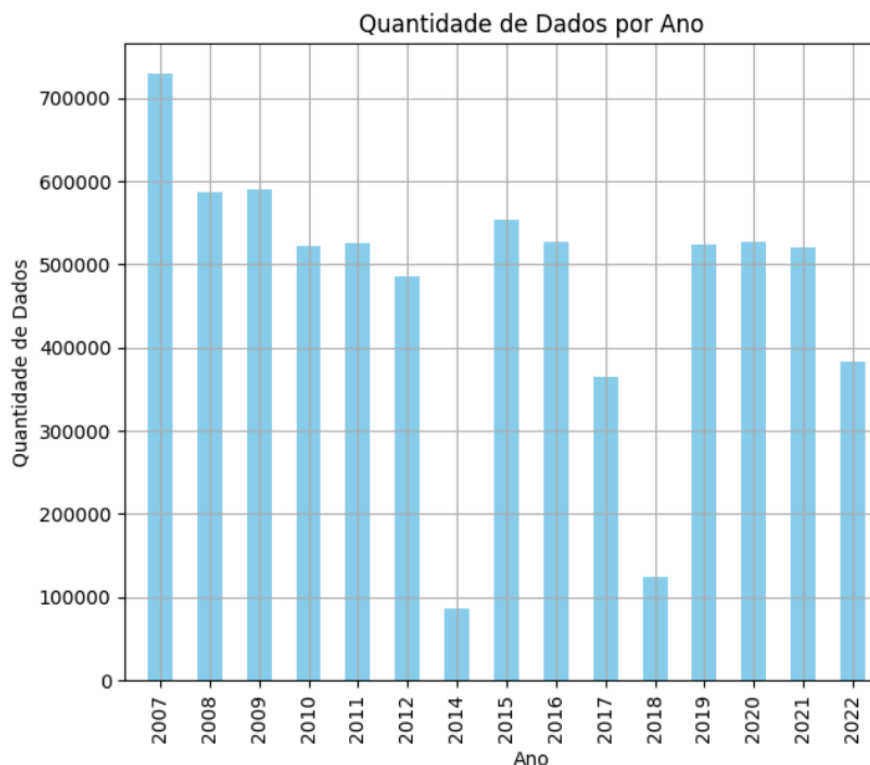


Gráfico 1 - Quantidade de dados coletados por ano (autoria própria)

Com base no gráfico, podemos concluir que a quantidade de dados medidos anualmente não é constante. Há anos sem nenhum dado registrado, como 2013, e outros com poucos dados coletados, como 2014 e 2018, que possuem aproximadamente 100 mil dados cada.

### 3.2 Distribuição de dados de Difusa e Global

Para a análise da distribuição, foi necessário reprocessar os dados, filtrando apenas as informações relevantes para a observação. Inicialmente, todos os dados foram lidos do arquivo CSV, e em seguida, foi gerado um novo arquivo contendo apenas as colunas de interesse: *Year*, *Day*, *Min*, *Global\_AVG* e *Diffuse\_AVG*. Esse procedimento foi adotado para reduzir o custo de processamento, que estava excessivamente elevado ao trabalhar com o conjunto de dados completo. A justificativa para essa abordagem foi focar na observação específica do comportamento dos dados de *Global\_AVG* e *Diffuse\_AVG*, garantindo eficiência no processamento e análise dos dados.

Semelhantemente do restante dos dados, a quantidade coletada anualmente não é constante, como mostrado no gráfico anterior. No Gráfico 2, podemos observar apenas a quantidade de dados de *Global\_AVG* e *Diffuse\_AVG* coletados anualmente, sendo 2013 o único ano sem dados no intervalo de 2007 a 2022. E no Gráfico 3 um histograma mostrando a frequência absoluta dos valores de irradiância global e difusa no mesmo período.

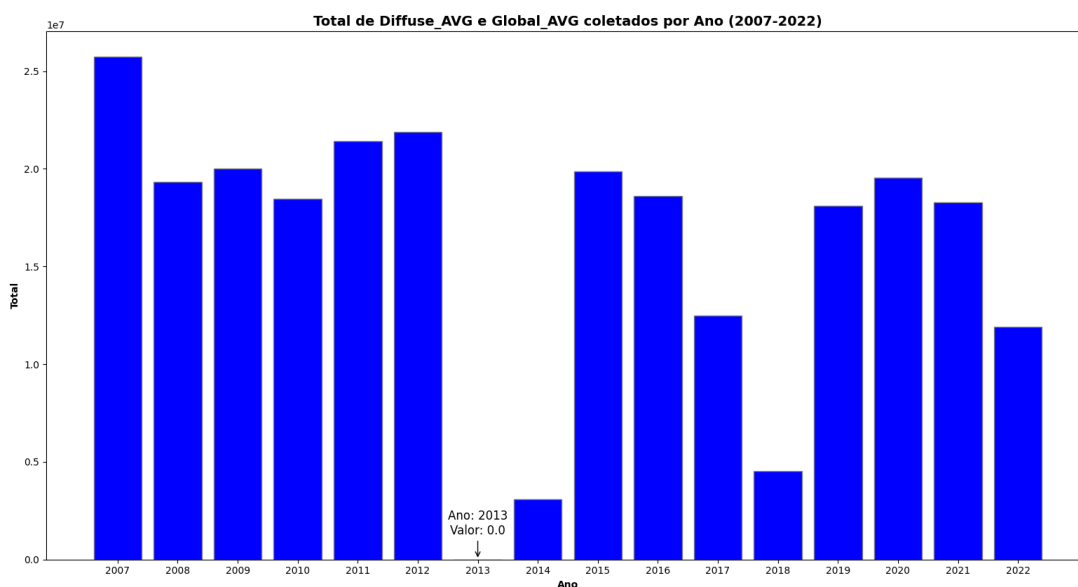


Gráfico 2 - Quantidade de dados coletados por ano (autoria própria)

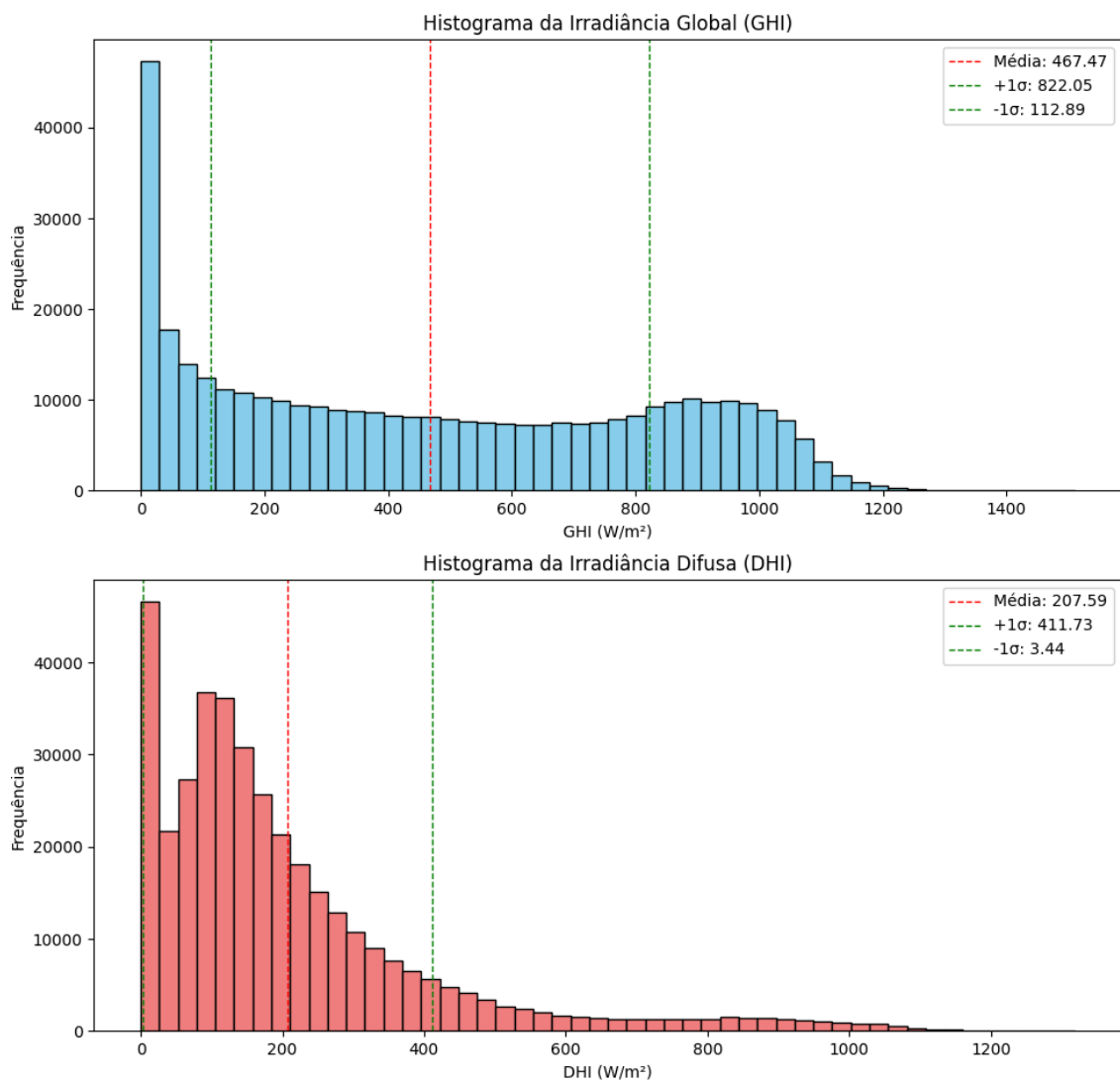


Gráfico 3 - Histograma das irradiâncias GHI (acima) e DHI (abaixo) (autoria própria)

No gráfico 4 e 5, vemos a relação entre a fração difusa ( $K_d$ ) e o índice de claridade ( $K_t$ ), sendo o primeiro o modelo de Erbs e o segundo a dispersão com a curva de Erbs (ERBS et al., 1982).

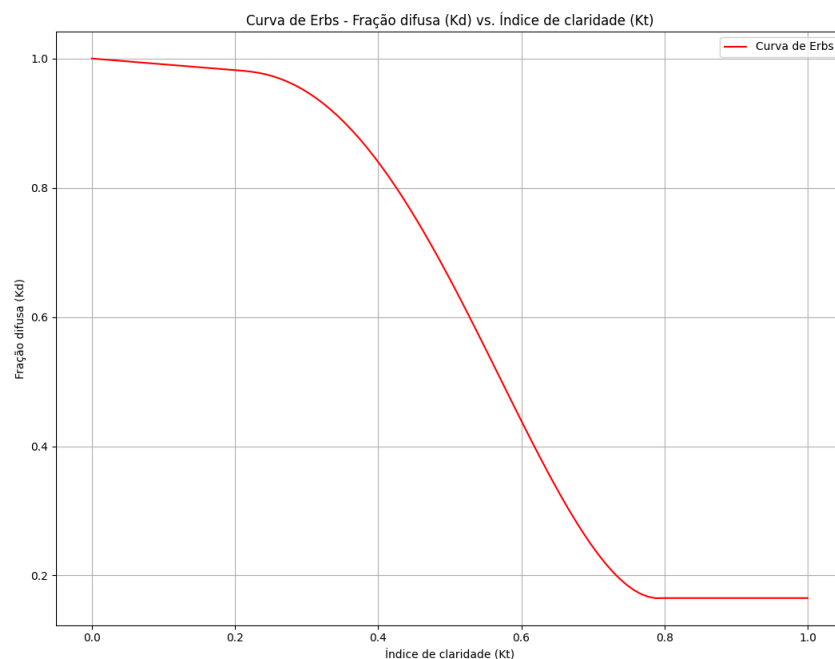


Gráfico 4 - Curva de Erbs para a dispersão da fração difusa no ano de 2015 (autoria própria)

No gráfico 5 temos o modelo de dispersão utilizando o modelo de Erbs, pode-se observar o comportamento da fração difusa em relação ao índice de claridade.

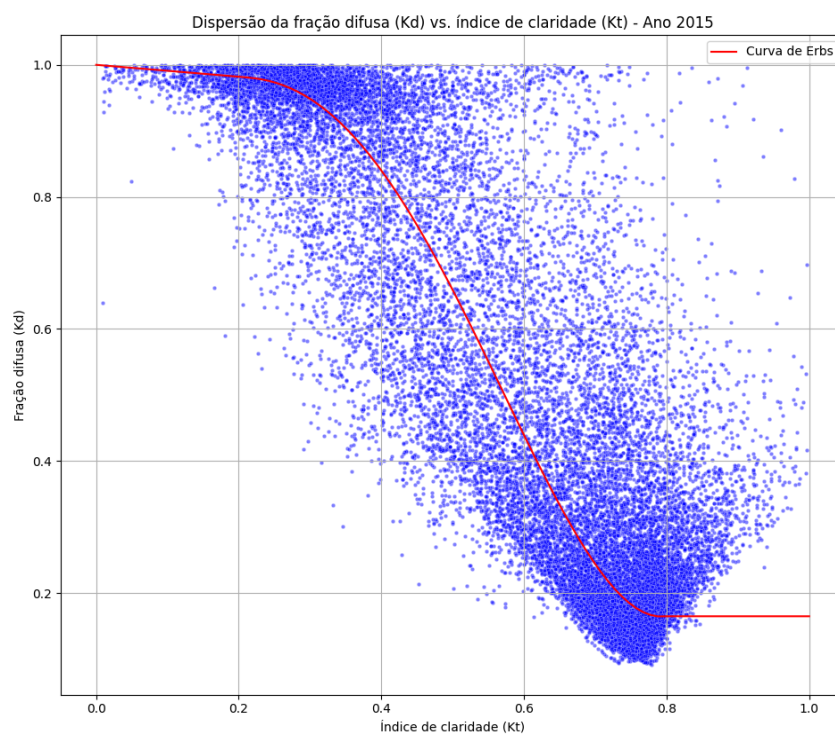


Gráfico 5 - Dispersão da fração difusa no ano de 2015 com curva ajustada de Erbs (autoria própria)

**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. Esse comportamento pode ser observado no gráfico 6 que é uma ampliação do setor inferior direito do gráfico 5.

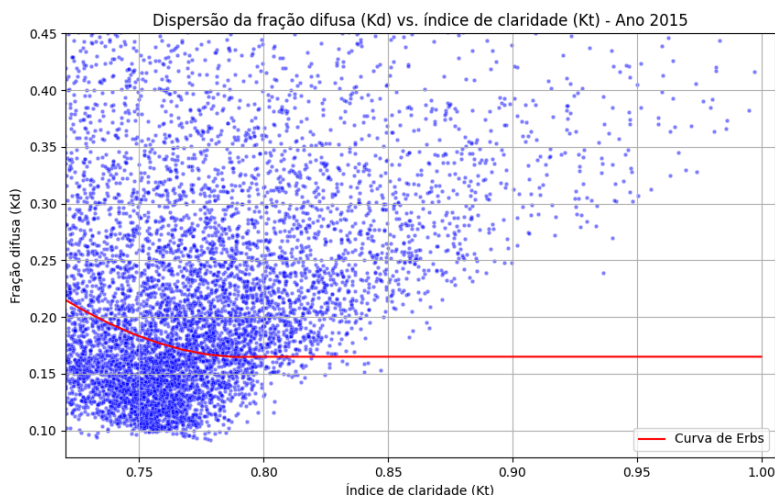


Gráfico 6 - Ampliação do gráfico 5 (autoria própria)

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.

### 3.2.1 Comparando Modelo de Erbs com os dados observado

Observando os parâmetros do modelo de Erbs (Gráfico 4 e 5) que estão na tabela 6, foi possível constatar que a Raiz quadrada do erro-médio (RMSE) no modelo foram ambos 0,155. Enquanto  $\beta_0$  e  $\beta_1$  são 9,47 e -13,005 respectivamente.

Parâmetro	Valor
beta_0	9.47
beta_1	-13.005
Desvio padrão dos resíduos	0.155
RMSE	0.155

Tabela 6 - Ajuste de curva, desvio padrão e RMSE para o modelo de Erbs no gráfico de dispersão (autoria própria)

Quando o desvio padrão (SD) e o erro quadrático médio (RMSE) têm o mesmo valor, isso sugere que não há viés no conjunto de dados e que os erros são puramente aleatórios. Isso ocorre porque o desvio padrão mede a dispersão dos valores em torno da média, refletindo apenas os erros aleatórios, enquanto o RMSE mede as diferenças entre os valores previstos por um modelo e os valores observados, incluindo tanto erros aleatórios quanto sistemáticos.

Se ambos têm o mesmo valor, significa que o erro médio (viés) é zero, ou seja, os erros são distribuídos aleatoriamente sem qualquer desvio sistemático. Esse fenômeno pode ser observado em análises de precisão geoespacial, onde a remoção de um viés sistemático dos dados resulta em valores iguais para SD e RMSE, indicando que os erros restantes são puramente aleatórios (ABDULLAH, 2024; HUTCH, 2024).

## **4 CONCLUSÃO**

O projeto permitiu o valioso aprendizado sobre instrumentos de medição e coleta de dados meteorológicos, também permitiu trabalhar com infraestrutura de rede, visto que foi feito novo cabeamento e conectores.

Foram desenvolvidos algoritmos para o tratamento dos dados com o objetivo de tornar a análise mais simples, isso trouxe uma excelente experiência relacionada a tratamento de dados massivos.

Foi possível testar uma parametrização para fração difusa sobre os dados coletados, que eventualmente poderá ser otimizada na sequência do trabalho através de uma função logística.

Não foi possível ser realizada a comparação dos dados com o rastreador solar, pois o mesmo será recuperado e então instalado para permitir que o estudo seja concluído até o fim do ano de 2024.



## 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.

**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](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

### Anexo A - Códigos utilizados para processamento de dados

A seguir, é apresentado o código utilizado para realizar o reprocessamento de dados com o objetivo de reduzir o custo de processamento a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas e tkinter.

Código 1: Reprocessamento dos dados de Year, Day, Min, Global\_AVG e Diffuse\_AVG

```
```python
import pandas as pd
import tkinter as tk
from tkinter import filedialog, simpledialog

# Inicializar o Tkinter
root = tk.Tk()
root.withdraw()

# Selecionar a pasta onde está o arquivo CSV
pasta = filedialog.askdirectory()

# Definir o caminho do arquivo
arquivo = f"{pasta}/arquivo_final_date_vld.csv"

# Solicitar ao usuário que insira o ano desejado
ano = simpledialog.askstring("Input", "Digite o ano que deseja filtrar (ex: 2023):")

# Carregar o arquivo CSV, especificando o delimitador e a linha do cabeçalho
try:
    data = pd.read_csv(arquivo, delimiter=',', header=0)
    print("Arquivo carregado com sucesso")
except Exception as e:
    print(f"Erro ao carregar o arquivo: {e}")
    exit()

# Exibir as primeiras linhas do dataframe para verificação
print("Primeiras linhas do dataframe:")
print(data.head())

# Verificar as colunas disponíveis no dataframe
print("Colunas disponíveis no dataframe:")
print(data.columns)
```

```

# Verificar os tipos de dados das colunas
print("Tipos de dados das colunas:")
print(data.dtypes)

# Filtrar apenas as colunas necessárias e remover linhas com dados
faltantes em 'Global_AVG' e 'Diffuse_AVG'
try:
    data_filtered = data[['Year', 'Day', 'Min', 'Global_AVG',
'Diffuse_AVG']].copy()
    data_filtered['Global_AVG'] =
pd.to_numeric(data_filtered['Global_AVG'], errors='coerce')
    data_filtered['Diffuse_AVG'] =
pd.to_numeric(data_filtered['Diffuse_AVG'], errors='coerce')
    data_filtered = data_filtered.dropna(subset=['Global_AVG',
'Diffuse_AVG'])
except KeyError as e:
    print(f"Erro ao filtrar colunas: {e}")
    exit()

# Exibir as primeiras linhas do dataframe filtrado para verificação
print("Primeiras linhas do dataframe filtrado:")
print(data_filtered.head())

# Verificar os valores únicos na coluna 'Year'
print("Valores únicos na coluna 'Year':")
print(data_filtered['Year'].unique())

# Filtrar os dados para o ano inserido pelo usuário
data_filtered = data_filtered[data_filtered['Year'] == int(ano)]

# Verificar se há linhas após o filtro do ano
print(f"Número de linhas após o filtro do ano {ano}:
{len(data_filtered)}")

# Caso não haja linhas, encerrar o script
if len(data_filtered) == 0:
    print(f"Nenhum dado encontrado para o ano de {ano}.")
    exit()

# Converter 'Min' para inteiro
data_filtered['Min'] = data_filtered['Min'].astype(int)

# Filtrar para intervalos de 10 minutos
data_filtered = data_filtered[data_filtered['Min'] % 10 == 0]

# Exibir as primeiras linhas do dataframe após a filtragem

```

```
print("Primeiras linhas do dataframe após a filtragem:")
print(data_filtered.head())

# Salvar os dados filtrados em um novo arquivo CSV com delimitador
especificado
output_file = f"{pasta}/csv_processado{ano}.csv"
data_filtered.to_csv(output_file, index=False, sep=',')

print(f"Dados processados foram salvos em: {output_file}")
```

A seguir, é apresentado o código utilizado para realizar a correção de dados de radiação difusa a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas, numpy e tkinter.

Código 2: Correção de radiação difusa

```
```python
import pandas as pd
import numpy as np
import tkinter as tk
from tkinter import filedialog

# Função para inicializar Tkinter e selecionar arquivo
def selecionar_arquivo(titulo):
    root = tk.Tk()
    root.withdraw()
    arquivo = filedialog.askopenfilename(title=titulo, filetypes=[('CSV
Files', '*.csv')])
    return arquivo

# Função para calcular declinação solar
def calc_declination(day_of_year):
    return -23.45 * np.cos(2 * np.pi * (day_of_year + 10) / 365)

# Função para corrigir os dados de difusa
def corrige_difusa_anel(diffdata, lat):
    F = np.array([
        [1.1, 1.11, 1.12, 1.12, 1.13, 1.13, 1.14, 1.14, 1.14],
        [1.11, 1.11, 1.12, 1.13, 1.13, 1.13, 1.14, 1.14, 1.14],
        [1.11, 1.12, 1.12, 1.13, 1.13, 1.13, 1.14, 1.14, 1.14],
        [1.11, 1.12, 1.12, 1.13, 1.13, 1.14, 1.14, 1.14, 1.14],
        [1.12, 1.12, 1.13, 1.13, 1.13, 1.14, 1.14, 1.14, 1.14],
        [1.12, 1.12, 1.13, 1.13, 1.13, 1.14, 1.14, 1.13, 1.13],
        [1.12, 1.13, 1.13, 1.13, 1.13, 1.14, 1.13, 1.13, 1.13],
        [1.12, 1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.13],
        [1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.12],
        [1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.12, 1.12],
        [1.13, 1.13, 1.13, 1.13, 1.13, 1.13, 1.12, 1.12, 1.11],
        [1.13, 1.13, 1.13, 1.13, 1.13, 1.12, 1.12, 1.11, 1.11],
        [1.13, 1.13, 1.13, 1.13, 1.13, 1.12, 1.12, 1.11, 1.1],
        [1.13, 1.13, 1.13, 1.13, 1.12, 1.12, 1.11, 1.11, 1.1],
        [1.13, 1.13, 1.13, 1.13, 1.12, 1.11, 1.11, 1.1, 1.1],
        [1.13, 1.13, 1.12, 1.12, 1.11, 1.11, 1.1, 1.1, 1.1],
        [1.13, 1.13, 1.12, 1.12, 1.11, 1.1, 1.1, 1.1, 1.1],
        [1.12, 1.12, 1.12, 1.11, 1.11, 1.1, 1.1, 1.1, 1.1]
    ])
    ])
```

```

diffdata['declinacao'] = diffdata['Day'].apply(calc_declination)
diffdata['dx'] = diffdata['Global_AVG'] / diffdata['Diffuse_AVG']
diffdata['dx'] = diffdata['dx'].clip(lower=0, upper=1)

        nlinha      = np.where(diffdata['Diffuse_AVG'] > 45,
np.minimum((diffdata['Diffuse_AVG'] - 45) // 5, 16), 0).astype(int)
        ncoluna = np.minimum((diffdata['dx'] * 8).astype(int), 8)

        rad_correcao = F[nlinha, ncoluna]
        diffdata['Diffuse_CORRIGIDA'] = np.where(diffdata['Diffuse_AVG'] > 0,
diffdata['Diffuse_AVG'] * rad_correcao, np.nan)

    return diffdata

# Função principal para selecionar arquivos e corrigir dados
def main():
    arquivo_processado = selecionar_arquivo('Selecione o arquivo
processado (processado_2022.csv)')
    if not arquivo_processado:
        print("Nenhum arquivo processado selecionado.")
        return

    try:
        diffdata = pd.read_csv(arquivo_processado)

        diffdata['Day'] = pd.to_datetime(diffdata['Day']).dt.dayofyear
        diffdata = diffdata.dropna(subset=['Day'])

        if 'Diffuse_AVG' in diffdata.columns:
            diffdata['Diffuse_AVG'] =
diffdata['Diffuse_AVG'].astype(float)
        if 'Global_AVG' in diffdata.columns:
            diffdata['Global_AVG'] = diffdata['Global_AVG'].astype(float)

        lat = -5.836749 # Latitude correta

        diffdata_corrigida = corrige_difusa_anel(diffdata, lat)

        # Salvar o arquivo corrigido
        arquivo_difusa_corrigida =
filedialog.asksaveasfilename(defaultextension=".csv", filetypes=[("CSV
files", "*.csv")],
title="Salvar arquivo de difusa corrigida",
initialfile="difusa_corrigida_2016.csv")
        if arquivo_difusa_corrigida:

```

```
        diffdata_corrigida.to_csv(arquivo_difusa_corrigida,
index=False)
        print(f"Arquivo salvo em: {arquivo_difusa_corrigida}")

    except Exception as e:
        print(f"Erro ao processar dados: {e}")

# Executar a função principal
if __name__ == '__main__':
    main()
```



## Anexo B - Códigos para plot de gráficos

A seguir, é apresentado o código utilizado para visualizar quantidade dados coletados por ano a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas, matplotlib e tkinter.

Código 1: Análise e Visualização de Dados coletados por ano

```
```python
import pandas as pd
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter import filedialog

# Inicializar o Tkinter
root = tk.Tk()
root.withdraw()

# Selecionar a pasta onde está o arquivo CSV
pasta = filedialog.askdirectory(title='Selecione a pasta onde está o
arquivo CSV')

# Definir o caminho do arquivo
arquivo = f"{pasta}/csv_processado_2007_2022.csv"

# Carregar o arquivo CSV
try:
    data = pd.read_csv(arquivo, delimiter=',', header=0)
    print("Arquivo carregado com sucesso")
except Exception as e:
    print(f"Erro ao carregar o arquivo: {e}")
    exit()

# Converter colunas para numéricas e remover linhas com dados faltantes
try:
    data['Year'] = pd.to_numeric(data['Year'], errors='coerce')
    data['Diffuse_AVG'] = pd.to_numeric(data['Diffuse_AVG'],
errors='coerce')
    data['Global_AVG'] = pd.to_numeric(data['Global_AVG'],
errors='coerce')
    data = data.dropna(subset=['Year'])
    data = data[(data['Year'] >= 2007) & (data['Year'] <= 2022)]
except KeyError as e:
    print(f"Erro ao processar dados: {e}")
    exit()

# Contar a quantidade de dados por ano para Diffuse_AVG e Global_AVG
```

```

years = list(range(2007, 2023))
counts_diffuse = []
counts_global = []

for year in years:
    counts_diffuse.append(data[data['Year'] ==
year]['Diffuse_AVG'].notna().sum())
    counts_global.append(data[data['Year'] ==
year]['Global_AVG'].notna().sum())

# Preparar os dados para o gráfico de pizza
labels_diffuse = [f"{year} (Diffuse)" for year in years]
labels_global = [f"{year} (Global)" for year in years]

sizes = counts_diffuse + counts_global
labels = labels_diffuse + labels_global

# Plotar o gráfico de pizza
fig, ax = plt.subplots(figsize=(10, 10), subplot_kw=dict(aspect="equal"))

wedges, texts, autotexts = ax.pie(sizes, labels=labels,
autopct='%1.1f%%',
startangle=90, counterclock=False)

for wedge in wedges:
    wedge.set_edgecolor('black')

ax.set_title('Quantidade de Dados de Diffuse_AVG e Global_AVG por Ano
(2007-2022)')

plt.setp(autotexts, size=10, weight="bold")
plt.show()

```

A seguir, é apresentado o código utilizado para realizar a análise e visualização de dados a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas, matplotlib e tkinter.

#### Código 2: Análise e Visualização de Dados

```
```python
import pandas as pd
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter import filedialog
import numpy as np

# Inicializar o Tkinter
root = tk.Tk()
root.withdraw()

# Selecionar a pasta onde está o arquivo CSV
pasta = filedialog.askdirectory(title='Selecione a pasta onde está o
arquivo CSV')

# Definir o caminho do arquivo
arquivo = f"{pasta}/csv_processado_2007_2022.csv"

# Carregar o arquivo CSV
try:
    data = pd.read_csv(arquivo, delimiter=',', header=0)
    print("Arquivo carregado com sucesso")
except Exception as e:
    print(f"Erro ao carregar o arquivo: {e}")
    exit()

# Converter colunas para numéricas e remover linhas com dados faltantes
try:
    data['Year'] = pd.to_numeric(data['Year'], errors='coerce')
    data['Diffuse_AVG'] = pd.to_numeric(data['Diffuse_AVG'],
errors='coerce')
    data['Global_AVG'] = pd.to_numeric(data['Global_AVG'],
errors='coerce')
    data = data.dropna(subset=['Year'])
    data = data[(data['Year'] >= 2007) & (data['Year'] <= 2022)]
except KeyError as e:
    print(f"Erro ao processar dados: {e}")
    exit()

# Calcular a soma total de Diffuse_AVG e Global_AVG por ano
```

```

years = list(range(2007, 2023))
totals = []

for year in years:
    total_diffuse = data[data['Year'] == year]['Diffuse_AVG'].sum()
    total_global = data[data['Year'] == year]['Global_AVG'].sum()
    total = total_diffuse + total_global
    totals.append(total)

# Plotar o gráfico de barras
fig, ax = plt.subplots(figsize=(12, 8))

# Plotar as barras
ax.bar(years, totals, color='b', edgecolor='grey')

# Adicionar labels, título e legendas
ax.set_xlabel('Ano', fontweight='bold')
ax.set_ylabel('Total', fontweight='bold')
ax.set_title('Total de Diffuse_AVG e Global_AVG coletados por Ano (2007-2022)', fontsize=14, weight='bold')
ax.set_xticks(years)

# Encontrar o ano com o menor valor de soma total
min_total_index = np.argmin(totals)
min_total_year = years[min_total_index]
min_total_value = totals[min_total_index]

# Adicionar anotação com o menor valor de soma total
ax.annotate(f'Ano: {min_total_year}\nValor: {min_total_value:.1f}',
            xy=(min_total_year, min_total_value), xytext=(min_total_year,
min_total_value + 0.05 * max(totals)),
            arrowprops=dict(facecolor='black', arrowstyle='->'),
            fontsize=12, ha='center')

# Mostrar o gráfico
plt.show()

```

A seguir, é apresentado o código utilizado para realizar plot de dados de difusa global com o objetivo de reduzir o custo de processamento a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas, matplotlib, numpy e tkinter.

Código 3: Histograma com desvios de valores para Global\_AVG e Diffuse\_AVG

```
```python
import pandas as pd
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter import filedialog
import numpy as np

# Inicializar o Tkinter
root = tk.Tk()
root.withdraw()

# Selecionar a pasta onde está o arquivo CSV
pasta = filedialog.askdirectory(title='Selecione a pasta onde está o
arquivo CSV')

# Definir o caminho do arquivo
arquivo = f"{pasta}/csv_processado_2007_2022.csv"

# Carregar o arquivo CSV
try:
    data = pd.read_csv(arquivo, delimiter=',', header=0)
    print("Arquivo carregado com sucesso")
except Exception as e:
    print(f"Erro ao carregar o arquivo: {e}")
    exit()

# Converter colunas para numéricas e remover linhas com dados faltantes
try:
    data['Year'] = pd.to_numeric(data['Year'], errors='coerce')
    data['Diffuse_AVG'] = pd.to_numeric(data['Diffuse_AVG'],
errors='coerce')
    data['Global_AVG'] = pd.to_numeric(data['Global_AVG'],
errors='coerce')
    data = data.dropna(subset=['Year'])
    data = data[(data['Year'] >= 2007) & (data['Year'] <= 2022)]
except KeyError as e:
    print(f"Erro ao processar dados: {e}")
    exit()
```

```

# Calcular a soma total de Diffuse_AVG e Global_AVG por ano
years = list(range(2007, 2023))
totals = []

for year in years:
    total_diffuse = data[data['Year'] == year]['Diffuse_AVG'].sum()
    total_global = data[data['Year'] == year]['Global_AVG'].sum()
    total = total_diffuse + total_global
    totals.append(total)

# Plotar o gráfico de barras
fig, ax = plt.subplots(figsize=(12, 8))

# Plotar as barras
ax.bar(years, totals, color='b', edgecolor='grey')

# Adicionar labels, título e legendas
ax.set_xlabel('Ano', fontweight='bold')
ax.set_ylabel('Total', fontweight='bold')
ax.set_title('Total de Diffuse_AVG e Global_AVG coletados por Ano (2007-2022)', fontsize=14, weight='bold')
ax.set_xticks(years)

# Encontrar o ano com o menor valor de soma total
min_total_index = np.argmin(totals)
min_total_year = years[min_total_index]
min_total_value = totals[min_total_index]

# Adicionar anotação com o menor valor de soma total
ax.annotate(f'Ano: {min_total_year}\nValor: {min_total_value:.1f}',
            xy=(min_total_year, min_total_value), xytext=(min_total_year,
min_total_value + 0.05 * max(totals)),
            arrowprops=dict(facecolor='black', arrowstyle='->'),
            fontsize=12, ha='center')

# Mostrar o gráfico
plt.show()

# Plotar histogramas para GHI e DHI
fig, axs = plt.subplots(2, 1, figsize=(12, 10))

# Histograma da GHI (Global_AVG)
axs[0].hist(data['Global_AVG'].dropna(), bins=50, color='skyblue',
edgecolor='black')
axs[0].set_title('Histograma da Irradiância Global (GHI)')
axs[0].set_xlabel('GHI (W/m²)')

```

```

axs[0].set_ylabel('Frequência')
ghi_mean = data['Global_AVG'].mean()
ghi_std = data['Global_AVG'].std()
axs[0].axvline(ghi_mean, color='red', linestyle='dashed', linewidth=1,
label=f'Média: {ghi_mean:.2f}')
axs[0].axvline(ghi_mean + ghi_std, color='green', linestyle='dashed',
linewidth=1, label=f'+1σ: {ghi_mean + ghi_std:.2f}')
axs[0].axvline(ghi_mean - ghi_std, color='green', linestyle='dashed',
linewidth=1, label=f'-1σ: {ghi_mean - ghi_std:.2f}')
axs[0].legend()

# Histograma da DHI (Diffuse_AVG)
axs[1].hist(data['Diffuse_AVG'].dropna(), bins=50, color='lightcoral',
edgecolor='black')
axs[1].set_title('Histograma da Irradiância Difusa (DHI)')
axs[1].set_xlabel('DHI (W/m²)')
axs[1].set_ylabel('Frequência')
dhi_mean = data['Diffuse_AVG'].mean()
dhi_std = data['Diffuse_AVG'].std()
axs[1].axvline(dhi_mean, color='red', linestyle='dashed', linewidth=1,
label=f'Média: {dhi_mean:.2f}')
axs[1].axvline(dhi_mean + dhi_std, color='green', linestyle='dashed',
linewidth=1, label=f'+1σ: {dhi_mean + dhi_std:.2f}')
axs[1].axvline(dhi_mean - dhi_std, color='green', linestyle='dashed',
linewidth=1, label=f'-1σ: {dhi_mean - dhi_std:.2f}')
axs[1].legend()

plt.tight_layout()
plt.show()

```

A seguir, é apresentado o código utilizado para realizar a análise e visualização de dados de um gráfico de dispersão a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas, numpy, seaborn, matplotlib e tkinter.

Código 4: Análise e Visualização de Dados de dispersão

```
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tkinter as tk
from tkinter import filedialog

# Inicializar o Tkinter
root = tk.Tk()
root.withdraw()

# Selecionar a pasta onde está o arquivo .dat
pasta = filedialog.askdirectory(title='Selecione a pasta onde está o
arquivo .dat')

# Definir o caminho do arquivo .dat
arquivo_dat = f"{pasta}/difusa_corrigida_com_geometria_solar.dat"

# Carregar o arquivo .dat
data_filtered = pd.read_csv(arquivo_dat, delimiter=',') # Ajuste o
delimitador conforme necessário

# Filtrar os dados para o ano de 2015
data_filtered = data_filtered[data_filtered['Year'] == 2015]

# Verificar e corrigir a falha de data (remover repetição dos primeiros
20 dias)
data_filtered = data_filtered.drop_duplicates(subset=['Year', 'Day',
'Min'])

# Calcular a elevação solar (Elev) se necessário
if 'Elev' not in data_filtered.columns:
    latitude = -5.836749 # Latitude correta
```



```

longitude = -35.206500 # Longitude correta

def calculate_solar_elevation(row, latitude, longitude):
    day_of_year = int(row['Day']) # Dia do ano (1 a 365)
    solar_time = int(row['Min']) / 60 # Tempo solar (horas)
    declination = 23.45 * np.sin(np.radians(360 * (284 + day_of_year)
/ 365))
    hour_angle = 15 * (solar_time - 12)
    solar_zenith = np.degrees(np.arccos(
        np.sin(np.radians(latitude)) *
np.sin(np.radians(declination)) +
        np.cos(np.radians(latitude)) *
np.cos(np.radians(declination)) * np.cos(np.radians(hour_angle))
    ))
    solar_elevation = 90 - solar_zenith
    return solar_elevation

data_filtered['Elev'] =
data_filtered.apply(calculate_solar_elevation, axis=1, args=(latitude,
longitude))

# Calcular a radiação extraterrestre (E0) se necessário
if 'E0' not in data_filtered.columns:
    data_filtered['E0'] = 1367 * (1 + 0.033 * np.cos(np.radians(360 *
data_filtered['Day'] / 365)))

# Calcular o índice de claridade (Kt)
data_filtered['Kt'] = data_filtered['Global_AVG'] / data_filtered['E0']

# Calcular a fração difusa (Kd)
data_filtered['Kd'] = data_filtered['Diffuse_CORRIGIDA'] /
data_filtered['Global_AVG']

# Excluir dados com Elev < 5 graus
data_filtered = data_filtered[data_filtered['Elev'] >= 5]

# Excluir dados suspeitos (Difusa = Global) para intervalos > 600 W/m² e
difusa > 650 W/m²
data_filtered = data_filtered[~((data_filtered['Global_AVG'] > 600) &
(data_filtered['Global_AVG'] == data_filtered['Diffuse_AVG']))]

```

```

data_filtered = data_filtered[data_filtered['Diffuse_AVG'] <= 650]

# Filtrar apenas os dados válidos para o gráfico de dispersão
filtered_data = data_filtered[(data_filtered['Kd'].notnull()) &
(data_filtered['Kd'] >= 0) & (data_filtered['Kd'] <= 1) &
(data_filtered['Kt'].notnull()) & (data_filtered['Kt'] >= 0) &
(data_filtered['Kt'] <= 1)]

# Função de cálculo da fração difusa segundo o modelo de Erbs
def erbs_diffuse_fraction(Kt):
    if Kt <= 0.22:
        Kd = 1 - 0.09 * Kt
    elif Kt <= 0.8:
        Kd = 0.9511 - 0.1604 * Kt + 4.388 * Kt**2 - 16.638 * Kt**3 +
12.336 * Kt**4
    else:
        Kd = 0.165
    return Kd

# Criar o gráfico de dispersão usando seaborn
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Kt', y='Kd', data=filtered_data, s=10, color='blue',
alpha=0.5)
plt.xlabel('Índice de claridade (Kt)')
plt.ylabel('Fração difusa (Kd)')
plt.title('Dispersão da fração difusa (Kd) vs. índice de claridade (Kt) -
Ano 2015')
plt.grid(True)

# Plotar a curva de Erbs
erbs_kt_range = np.linspace(0, 1, 100)
erbs_kd_values = [erbs_diffuse_fraction(kt) for kt in erbs_kt_range]
plt.plot(erbs_kt_range, erbs_kd_values, color='red', label='Curva de
Erbs')

plt.legend()
plt.show()

```

A seguir, é apresentado o código utilizado para realizar a análise e visualização de dados de um gráfico de erbs a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas, matplotlib e tkinter.

Código 5: Análise e Visualização de Modelo de Erbs

```
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tkinter as tk
from tkinter import filedialog

# Inicializar o Tkinter
root = tk.Tk()
root.withdraw()

# Selecionar a pasta onde está o arquivo .dat
pasta = filedialog.askdirectory(title='Selecione a pasta onde está o
arquivo .dat')

# Definir o caminho do arquivo .dat
arquivo_dat = f"{pasta}/difusa_corrigida_com_geometria_solar.dat"

# Carregar o arquivo .dat
data_filtered = pd.read_csv(arquivo_dat, delimiter=',') # Ajuste o
delimitador conforme necessário

# Filtrar os dados para o ano de 2015
data_filtered = data_filtered[data_filtered['Year'] == 2015]

# Verificar e corrigir a falha de data (remover repetição dos primeiros
20 dias)
data_filtered = data_filtered.drop_duplicates(subset=['Year', 'Day',
'Min'])

# Calcular a elevação solar (Elev) se necessário
if 'Elev' not in data_filtered.columns:
    latitude = -5.836749 # Latitude correta
    longitude = -35.206500 # Longitude correta
```

```

def calculate_solar_elevation(row, latitude, longitude):
    day_of_year = int(row['Day']) # Dia do ano (1 a 365)
    solar_time = int(row['Min']) / 60 # Tempo solar (horas)
    declination = 23.45 * np.sin(np.radians(360 * (284 + day_of_year)
/ 365))
    hour_angle = 15 * (solar_time - 12)
    solar_zenith = np.degrees(np.arccos(
        np.sin(np.radians(latitude)) *
np.sin(np.radians(declination)) +
        np.cos(np.radians(latitude)) *
np.cos(np.radians(declination)) * np.cos(np.radians(hour_angle))
    ))
    solar_elevation = 90 - solar_zenith
    return solar_elevation

data_filtered['Elev'] =
data_filtered.apply(calculate_solar_elevation, axis=1, args=(latitude,
longitude))

# Calcular a radiação extraterrestre (E0) se necessário
if 'E0' not in data_filtered.columns:
    data_filtered['E0'] = 1367 * (1 + 0.033 * np.cos(np.radians(360 *
data_filtered['Day'] / 365)))

# Calcular o índice de claridade (Kt)
data_filtered['Kt'] = data_filtered['Global_AVG'] / data_filtered['E0']

# Calcular a fração difusa (Kd)
data_filtered['Kd'] = data_filtered['Diffuse_CORRIGIDA'] /
data_filtered['Global_AVG']

# Excluir dados com Elev < 5 graus
data_filtered = data_filtered[data_filtered['Elev'] >= 5]

# Excluir dados suspeitos (Difusa = Global) para intervalos > 600 W/m² e
difusa > 650 W/m²
data_filtered = data_filtered[~((data_filtered['Global_AVG'] > 600) &
(data_filtered['Global_AVG'] == data_filtered['Diffuse_CORRIGIDA']))]
data_filtered = data_filtered[data_filtered['Diffuse_CORRIGIDA'] <= 650]

```

```

# Filtrar apenas os dados válidos para o gráfico de dispersão
filtered_data = data_filtered[(data_filtered['Kd'].notnull()) &
(data_filtered['Kd'] >= 0) & (data_filtered['Kd'] <= 1) &
(data_filtered['Kt'].notnull()) & (data_filtered['Kt'] >= 0) &
(data_filtered['Kt'] <= 1)]

# Função de cálculo da fração difusa segundo o modelo de Erbs
def erbs_diffuse_fraction(Kt):
    if Kt <= 0.22:
        Kd = 1 - 0.09 * Kt
    elif Kt <= 0.8:
        Kd = 0.9511 - 0.1604 * Kt + 4.388 * Kt**2 - 16.638 * Kt**3 +
12.336 * Kt**4
    else:
        Kd = 0.165
    return Kd

# Calcular a curva de ajuste usando o modelo de Erbs
params = np.polyfit(filtered_data['Kt'], filtered_data['Kd'], 4)

# Calcular resíduos e RMSE
filtered_data['Kd_estimated'] = np.polyval(params, filtered_data['Kt'])
filtered_data['Residual'] = filtered_data['Kd'] -
filtered_data['Kd_estimated']
std_residual = np.std(filtered_data['Residual'])
rmse = np.sqrt(np.mean(filtered_data['Residual']**2))

# Criar dataframe com os resultados
resultados = pd.DataFrame({
    'Parâmetro': ['beta_0', 'beta_1', 'Desvio padrão dos resíduos',
'RMSE'],
    'Valor': [round(params[0], 3), round(params[1], 3),
round(std_residual, 3), round(rmse, 3)]
})

# Mostrar resultados em formato tabular com matplotlib
fig, ax = plt.subplots(figsize=(8, 4))
ax.axis('off')
ax.axis('tight')

```

```

ax.table(cellText=resultados.values,          colLabels=resultados.columns,
cellLoc='center', loc='center')

plt.title('Resultados da Curva Erbs Teórica e Estatísticas de Erro')
plt.show()

# Criar o gráfico da curva de Erbs
plt.figure(figsize=(10, 6))
erbs_kt_range = np.linspace(0, 1, 100)
erbs_kd_values = [erbs_diffuse_fraction(kt) for kt in erbs_kt_range]

# Plotar a curva de Erbs
plt.plot(erbs_kt_range, erbs_kd_values, color='red', label='Curva de Erbs')
plt.xlabel('Índice de claridade (Kt)')
plt.ylabel('Fração difusa (Kd)')
plt.title('Curva de Erbs - Fração difusa (Kd) vs. Índice de claridade (Kt)')
plt.grid(True)

plt.legend()
plt.show()

```

A seguir, é apresentado o código utilizado para plotar a tabela com os dados para o modelo de Erbs a partir de um arquivo CSV. O código foi desenvolvido em Python utilizando as bibliotecas pandas, matplotlib e tkinter.

Código 5: Tabela do Modelo de Erbs

```
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tkinter as tk
from tkinter import filedialog

# Inicializar o Tkinter
root = tk.Tk()
root.withdraw()

# Selecionar a pasta onde está o arquivo .dat
pasta = filedialog.askdirectory(title='Selecione a pasta onde está o
arquivo .dat')

# Definir o caminho do arquivo .dat
arquivo_dat = f"{pasta}/difusa_corrigida_com_geometria_solar.dat"

# Carregar o arquivo .dat
data_filtered = pd.read_csv(arquivo_dat, delimiter=',') # Ajuste o
delimitador conforme necessário

# Filtrar os dados para o ano de 2015
data_filtered = data_filtered[data_filtered['Year'] == 2015]

# Verificar e corrigir a falha de data (remover repetição dos primeiros
20 dias)
data_filtered = data_filtered.drop_duplicates(subset=['Year', 'Day',
'Min'])

# Calcular a elevação solar (Elev) se necessário
if 'Elev' not in data_filtered.columns:
    latitude = -5.836749 # Latitude correta
    longitude = -35.206500 # Longitude correta
```

```

def calculate_solar_elevation(row, latitude, longitude):
    day_of_year = int(row['Day']) # Dia do ano (1 a 365)
    solar_time = int(row['Min']) / 60 # Tempo solar (horas)
    declination = 23.45 * np.sin(np.radians(360 * (284 + day_of_year)
/ 365))
    hour_angle = 15 * (solar_time - 12)
    solar_zenith = np.degrees(np.arccos(
        np.sin(np.radians(latitude)) *
np.sin(np.radians(declination)) +
        np.cos(np.radians(latitude)) *
np.cos(np.radians(declination)) * np.cos(np.radians(hour_angle))
    ))
    solar_elevation = 90 - solar_zenith
    return solar_elevation

data_filtered['Elev'] =
data_filtered.apply(calculate_solar_elevation, axis=1, args=(latitude,
longitude))

# Calcular a radiação extraterrestre (E0) se necessário
if 'E0' not in data_filtered.columns:
    data_filtered['E0'] = 1367 * (1 + 0.033 * np.cos(np.radians(360 *
data_filtered['Day'] / 365)))

# Calcular o índice de claridade (Kt)
data_filtered['Kt'] = data_filtered['Global_AVG'] / data_filtered['E0']

# Calcular a fração difusa (Kd)
data_filtered['Kd'] = data_filtered['Diffuse_CORRIGIDA'] /
data_filtered['Global_AVG']

# Excluir dados com Elev < 5 graus
data_filtered = data_filtered[data_filtered['Elev'] >= 5]

# Excluir dados suspeitos (Difusa = Global) para intervalos > 600 W/m² e
difusa > 650 W/m²
data_filtered = data_filtered[~((data_filtered['Global_AVG'] > 600) &
(data_filtered['Global_AVG'] == data_filtered['Diffuse_CORRIGIDA']))]
data_filtered = data_filtered[data_filtered['Diffuse_CORRIGIDA'] <= 650]

```



```

# Filtrar apenas os dados válidos para o gráfico de dispersão
filtered_data = data_filtered[(data_filtered['Kd'].notnull()) &
(data_filtered['Kd'] >= 0) & (data_filtered['Kd'] <= 1) &
(data_filtered['Kt'].notnull()) & (data_filtered['Kt'] >= 0) &
(data_filtered['Kt'] <= 1)]

# Função de cálculo da fração difusa segundo o modelo de Erbs
def erbs_diffuse_fraction(Kt):
    if Kt <= 0.22:
        Kd = 1 - 0.09 * Kt
    elif Kt <= 0.8:
        Kd = 0.9511 - 0.1604 * Kt + 4.388 * Kt**2 - 16.638 * Kt**3 +
12.336 * Kt**4
    else:
        Kd = 0.165
    return Kd

# Calcular a curva de ajuste usando o modelo de Erbs
params = np.polyfit(filtered_data['Kt'], filtered_data['Kd'], 4)

# Calcular resíduos e RMSE
filtered_data['Kd_estimated'] = np.polyval(params, filtered_data['Kt'])
filtered_data['Residual'] = filtered_data['Kd'] -
filtered_data['Kd_estimated']
std_residual = np.std(filtered_data['Residual'])
rmse = np.sqrt(np.mean(filtered_data['Residual']**2))

# Criar dataframe com os resultados
resultados = pd.DataFrame({
    'Parâmetro': ['beta_0', 'beta_1', 'Desvio padrão dos resíduos',
'RMSE'],
    'Valor': [round(params[0], 3), round(params[1], 3),
round(std_residual, 3), round(rmse, 3)]
})

# Mostrar resultados em formato tabular com matplotlib
fig, ax = plt.subplots(figsize=(8, 4))
ax.axis('off')
ax.axis('tight')

```

```

ax.table(cellText=resultados.values, colLabels=resultados.columns,
cellLoc='center', loc='center')

plt.title('Resultados da Curva Erbs Teórica e Estatísticas de Erro')
plt.show()

# Criar o gráfico de dispersão usando seaborn
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Kt', y='Kd', data=filtered_data, s=10, color='blue',
alpha=0.5)
plt.xlabel('Índice de claridade (Kt)')
plt.ylabel('Fração difusa (Kd)')
plt.title('Dispersão da fração difusa (Kd) vs. índice de claridade (Kt) -
Ano 2015')
plt.grid(True)

# Plotar a curva de Erbs
erbs_kt_range = np.linspace(0, 1, 100)
erbs_kd_values = [erbs_diffuse_fraction(kt) for kt in erbs_kt_range]
plt.plot(erbs_kt_range, erbs_kd_values, color='red', label='Curva de
Erbs')

plt.legend()
plt.show()

```