

**FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA**  
**DATA SCIENCE**

**RM 98119 – CESAR OLIVEIRA GOES**  
**RM 97885 – FIAMA DOS SANTOS TRAJANO**



**GLOBAL SOLUTIONS**  
**GREEN ENERGY**

## Sumário

<b>1. NOME DO PROJETO.....</b>	<b>3</b>
<b>2. CONTEXTUALIZAÇÃO DO PROBLEMA.....</b>	<b>4</b>
<b>3. SOLUÇÃO DO PROBLEMA.....</b>	<b>5</b>
<b>4. PROPOSTA DA SOLUÇÃO (COMO SERÁ RESOLVIDO).....</b>	<b>6</b>
<b>5. BENEFÍCIOS ESPERADOS.....</b>	<b>7</b>
<b>6. IMPACTO DA SOLUÇÃO.....</b>	<b>8</b>
<b>7. TECNOLOGIAS NECESSÁRIAS.....</b>	<b>9</b>
<b>7.1 ARQUITETURA DA SOLUÇÃO.....</b>	<b>12</b>
7.1.1 Coleta de Dados: Onde olhamos os dados de energia.....	13
7.1.2 Ingestão de Dados.....	13
7.1.3 Tratamento e Pré processamento de dados.....	16
7.1.4 Modelo de Machine Learning e Deep Learning para previsão.....	17
7.1.5 Armazenamento de Resultados.....	24
7.1.6 Visualização de Dados e Insights.....	28
7.1.6.1 KPI's e Métricas.....	29
7.1.6.2 Dicionário de Dados/Metadado.....	30
7.1.6.3 Painéis e Gráficos.....	31
7.1.7 Etapa Avulsas na Arquitetura.....	43
<b>8. CÁLCULO DE CUSTOS, ROIS E BENEFÍCIOS DA SOLUÇÃO.....</b>	<b>50</b>
<b>9. LINKS PARA SOLUÇÃO/PROTÓTIPO.....</b>	<b>52</b>
<b>10. CONCLUSÃO.....</b>	<b>53</b>
<b>REFERÊNCIAS.....</b>	<b>54</b>

## 1. Nome do Projeto

O nome da solução SolarWindIQ combina dois pilares da energia renovável: o sol (representada pela energia solar) e o vento (pela energia eólica), representados pelas palavras "Solar" e "Wind". O "IQ" simboliza a inteligência e a capacidade analítica da solução, destacando o uso de dados e tecnologia avançada para otimizar a geração de energia solar e eólica. O produto se propõe a oferecer previsões precisas e insights baseados em dados meteorológicos em tempo real, melhorando a eficiência operacional. Com isso, o SolarWindIQ permite uma gestão mais inteligente e sustentável da produção de energia renovável. O propósito que queremos atingir reflete o nosso compromisso com a inovação melhorada e sustentabilidade no setor energético.



Logo do SolarWindIQ

Cores:

- Verde remete a sustentabilidade que é o nosso compromisso com o planeta;
- Laranja que remete a energia, alegria e disposição, pois nossas soluções entregam e agregam valor ao negócio dos nossos clientes.

Simbolos:

- Folha: Remete ao crescimento, novas ideias e sustentabilidade;
- Raio: Remete a Inovação, energia, impacto e a rapidez em solucionar problemas

As cores unificadas com os símbolos e o que somos quando se trata de inovação e entregar bons resultados.

Slogan: **Dados Inteligentes. Insights Precisos para um futuro sustentável.**

Nosso Slogan reflete a proposta central do **SolarWind IQ**, que utiliza dados de forma estratégica e tecnologia inteligente para otimizar a geração de energia renovável. Destacando-se pela importância de transformar grandes volumes de dados em informações valiosas, proporcionando previsões precisas para melhorar a eficiência operacional e apoiar decisões mais sustentáveis. Auxiliando empresas a promover um futuro mais verde e eficiente por meio da energia solar e eólica, tomando decisões baseadas em dados graças aos modelos preditivos e IA generativa.

## **2. Contextualização do problema**

O problema que decidimos enfrentar é o crescente aumento pela demanda de energia no mundo e como tornar a produção e oportunidades cada vez mais sustentáveis. Dessa forma nos debruçamos em como a Ciência de Dados pode contribuir para um Futuro Energético Sustentável.

Um dos maiores desafios do século XXI, se não o maior, é a crescente demanda por energia elétrica para sustentar nossa sociedade altamente tecnológica e como suprir essa demanda de forma sustentável para combater a crescente ameaça das mudanças climáticas.

Estudos realizados pela Agência Internacional de Energia apontam que no Brasil, a demanda por energia elétrica irá crescer em média 2,5% nos próximos dois anos, e dessa forma, surge a questão, como suprir essa crescente demanda de forma sustentável?

Essa mesma agência, estima que a produção Eólica e Solar no Brasil deva crescer cerca de 50% como forma de suprir essa crescente demanda e impulsionar novos mercados.

Países ao redor do mundo estão observando a Energia Renovável como a grande solução para seus problemas climáticos, sociais e ambientais, onde encontros internacionais como a COP29, sendo realizada entre o dia 11/11/2024 e 22/11/2024 surgem como ponto de discussão de ideias e negócios para impulsionar o desenvolvimento sustentável.



Usina Fotovoltaica no Mato Grosso do Sul Fonte: <https://www.rcn67.com.br/economia/paranaiba-e-cassilandia-terao-a-maior-fazenda-de-energia-solar-de-ms>

Países como Noruega, Colômbia e Canadá vem se destacando como países com alto investimento e metas ambiciosas para tornar grande parte de sua matriz energética sustentável, porém, nenhum desses países possui o potencial de diversificação energética como o Brasil.

Em 2023 o Brasil atingiu uma produção de energia elétrica limpa recorde na sua história, com cerca de 90% de sua matriz vindo de energias renováveis, porém, com a crescente demanda e mudanças climáticas como longos períodos sem chuvas em locais de produção hidrelétrica, podem fazer com que o cenário se altere, sendo utilizado cada vez mais usinas de combustíveis fósseis para compensar como termelétrica à carvão.

Segundo a FIEMG, até 2035 o Brasil tem condições de banir de vez as termelétricas substituindo-as por usinas de energia limpa, como solar e eólica.

Sendo assim, como a ciência de dados pode ajudar o Brasil como um todo a alcançar uma matriz energética mais sustentável e limpa, contribuindo para a melhoria da vida da população, garantindo a segurança energética e combatendo o aquecimento global reduzindo a emissão de gases?

### **3. Solução do problema**

Nossa solução consiste em desenvolver uma plataforma de previsão de potencial energético no Brasil, por meio de machine learning e redes neurais, mais especificamente, previsão de capacidade energética solar e eólica.

Essa solução traria mais precisão, visão e agilidade para auxiliar empresas e governos a tomarem ações estratégicas quanto à produção de energia solar e eólica, por

exemplo, locais com grande potencial indicados pelo nosso modelo de previsão podem ser explorados economicamente para construção de usinas solares e eólicas para aumentarem a produção de energia limpa e até mesmo substituir termelétricas que suprem alguma demanda daquela região específica.

Além disso, essa plataforma trará insights valiosos para estudos e compreensão climáticos e ambientais dessas regiões. Por exemplo, pode-se descobrir enorme potencial em região A e que a construção de uma usina ali pode desenvolver as cidades ao redor. Outro exemplo, seria que a região B possui um potencial menor comparando com a C, porém, na B já há usinas fotovoltaicas, então um estudo de viabilidade para mover a usina ou até desativá-la poderia ser aplicado.

De forma geral, nossa solução irá preencher a lacuna e melhorar os estudos de potencial energético das mais diversas regiões do Brasil, em uma metáfora, seria como nosso projeto fosse um cardápio de um restaurante com as mais diversas opções de potencial energético para as empresas, governo e agências estudarem para implementar usinas de energia sustentável, no caso eólicas e solares.

#### **4. Proposta de solução (como será resolvido)**

A proposta de solução visa auxiliar o mercado de energia elétrica através de uma plataforma inovadora que combina coleta de dados de diversas fontes, processamento inteligente e geração de insights acionáveis. A plataforma (Definir um Nome) fornecerá informações precisas e em tempo real sobre o potencial de produção energética solar e eólico.

A solução será dividida em cinco etapas principais:

##### **Coleta de Dados:**

- **Dados públicos:** Coleta de dados sobre produção e capacidade energéticas passados disponíveis em sites de agências do governo como ANEEL e ONS e utilizando APIs públicas.
- **Dados meteorológicos:** Dados de histórico meteorológicos de sites com API'S públicas e abertas

##### **Processamento e Tratamento de Dados:**

- **Limpeza e pré-processamento:** Os dados coletados serão limpos, padronizados, normalizados e preparados para análise e implementação dos modelos de machine learning e redes neurais.

- **Treinamento de modelos:** Modelos de Machine Learning e Redes neurais serão treinadas com os dados coletados para identificar com precisão os melhores pontos para produção e geração de energia eólica e solar.
- **Análise de dados:** Algoritmos avançados serão utilizados para analisar os dados coletados, identificando padrões e previsões.

#### **Armazenamento:**

- **Banco de dados seguro:** Os dados coletados e processados serão armazenados em um banco de dados seguro e acessível, garantindo a integridade e confidencialidade das informações.

#### **Geração de Insights:**

- **Visualização de dados:** Dashboards interativos e relatórios detalhados serão criados para apresentar os insights gerados pela análise dos dados.
- **Identificação de áreas com potencial:** As áreas que apresentarem excelente potencial energético, serão identificadas com precisão, permitindo a direcionamento de ações.
- **Tendências e previsões:** O sistema será capaz de identificar tendências e prever cenários futuros, auxiliando na definição de estratégias de longo prazo.

#### **Tomada de Decisões e Ações:**

- **Informações para políticas públicas:** Os insights gerados pelo sistema fornecerão informações valiosas para a definição de políticas públicas eficazes para o desenvolvimento sustentável e geração de energia limpa e renovável.
- **Apoio a ações de empresas do ramo energético:** Empresas que atuam no Setor de Energia terão acesso a dados e ferramentas para embasar suas ações e otimizar seus recursos.

### **5. Benefícios esperados**

#### **Agilidade na Obtenção de Insights:**

- **Ferramenta prática e amigável:** Visualize as áreas de forma precisa com potencial energético através de gráficos e mapas interativos.
- **Insights completos:** Obtenha estimativas de quantidade a ser produzida, previsão de movimento dos ventos e intensidade solar e impacto ambiental e econômico.

#### **Precisão na Localização:**

- **Coleta de dados de diversas fontes:** Dados tratados e cruzados por latitude e longitude para informar com exatidão a localização.

### **Parcerias Público-Privadas:**

- **Ferramenta facilitadora:** A plataforma auxilia na formação de parcerias entre governos, instituições e iniciativa privada para o desenvolvimento energético sustentável.
- **Ações conjuntas:** Amplificação do impacto na busca por soluções para um problema global.

### **Grande Base de Dados para Estudos:**

- **Apoio à pesquisa:** Ferramenta valiosa para pesquisas sobre energias renováveis, auxiliando no desenvolvimento de soluções.
- **Evolução contínua:** Dados da plataforma alimentam pesquisas e estudos, permitindo seu aprimoramento constante.

## **6. Impacto da solução**

A plataforma terá um impacto significativo tanto como ferramenta de prospecção de oportunidades de geração de energia solar e eólica, quanto fomentar o desenvolvimento de geração de energia renováveis nas mais diversas áreas do território brasileiro.

### **Precisão no monitoramento e otimização de geração de energia:**

- **Localização precisa:** Por meio do modelo de previsão, pode ser indicado com precisão locais com enorme potencial de geração de energia, classificando essas regiões de acordo com seu potencial de geração energética para eólico e solar.

### **Fomento ao Desenvolvimento de Geração Elétrica Renovável:**

**Maior previsibilidade e garantias:** Assim como a empresa Casa dos Ventos utiliza de modelos de machine learning para prever a intensidade dos ventos e assim ativar suas usinas nos melhores horários, nossa ferramenta agirá de forma semelhante, porém, por meio de localização geográfica precisa, dessa forma, empresas geradoras de energia eólica e solar podem tomar decisões mais otimizadas.

- **Desenvolvimento econômico local:** O aumento da geração de energia solar e eólica impulsionará a economia local, gerando renda e oportunidades para as cidades e regiões onde as plantas serão instaladas promovendo o desenvolvimento sustentável.



### **Redução de Custos:**

- **Mais assertividade e menos gastos:** Com uma maior assertividade nos locais para instalação, otimizando a geração de energia, os custos de operação e instalação diminuem.



## Promoção da preservação do meio ambiente:

- **Energia limpa e sustentável:** A plataforma contribuirá para a redução de geração de energia altamente poluente ou não renovável, como termelétricas por exemplo, pois, ao dar uma visão ampla e precisa de pontos com excelente potencial energético, torna-se mais amplo o leque de instalação de usinas eólicas ou solares, dessa forma, fomentando produção renovável e limpa de energia elétrica, emitindo menos CO2, e impactando minimamente o meio ambiente.

7. Tecnologias Necessárias	
A seguir, segue a lista de ferramentas que utilizamos para o desenvolvimento dessa solução.	
Data Ingestion	
	<p><b>Azure Data Factory:</b> O Azure Data Factory é um serviço de integração de dados na nuvem que permite criar, agendar e gerenciar fluxos de trabalho para mover e transformar dados em larga escala, conectando diversas fontes de dados. Ele é ideal para processos de ETL (Extract, Transform, Load) e pipelines de dados complexos, possibilitando análises em tempo real e suporte a cenários de big data.</p>
Data Preparation and Cleaning	
	<p><b>Databricks:</b> O Databricks é uma plataforma unificada de análise de dados que combina processamento de dados em larga escala e nessa fase utilizamos para o armazenamento dos dados coletados pelo Operador Nacional do Sistema Elétrico (ONS), Agência Nacional de Energia Elétrica (ANEEL) e dados meteorológicos, como insolação e velocidade do vento. Esses dados são obtidos a partir de fontes públicas e APIs especializadas, como OpenWeatherMap e WeatherAPI. Eles são armazenados na primeira camada da nossa arquitetura Medallion, denominada L1.</p>
Productive Modeling	



**Python:** O pré-processamento e o processamento dos dados serão realizados no ambiente Python Colab, utilizando técnicas de deep learning e machine learning. Bibliotecas nativas do Python, como NumPy e Pandas, auxiliarão na manipulação dos dados.



**Tensorflow Keras:** TensorFlow é uma biblioteca de software de código aberto criada pelo Google para computação numérica, aprendizado de máquina em grande escala, deep learning e outras cargas de trabalho de análise estatística e preditiva. Ele facilita a criação e o treinamento de modelos de aprendizado de máquina, como redes neurais artificiais, para diversas tarefas, como:

- **Previsão de séries temporais:** Previsão de eventos futuros com base em dados históricos.
- **Deteção de anomalias:** Identificação de padrões incomuns em dados.

### Results Storage and Processing



**SGBD Oracle Data modeler:** Oracle Data Modeler para estruturar a modelagem das tabelas utilizando da arquitetura Medallion. Pensando nisso, garantimos uma organização eficiente e escalável dos dados. Na camada L1, serão armazenados os dados brutos, enquanto na L2 ocorrerá o processamento e a limpeza dos dados. Já na L3, teremos os dados refinados e preparados para análise avançada. Essa abordagem permite maior flexibilidade e governança, otimizando o fluxo de dados ao longo das camadas.



**Databricks:** Nesta fase de armazenamento dos dados nas camadas L2 e L3, o foco é transformar e otimizar os dados para análises mais profundas e insights acionáveis. Na L2, os dados passam por um processo de limpeza e enriquecimento, garantindo sua qualidade e confiabilidade. Já na L3, aplicamos modelos e transformações avançadas, preparando os dados para análises detalhadas e relatórios, com ênfase na performance e na escalabilidade, aproveitando ao máximo as funcionalidades do Databricks.

## Orquestração



**Apache Airflow:** Ferramenta de orquestração de fluxos de dados e a automação e o controle eficiente dos processos. Com sua abordagem baseada em DAGs (Grafos Acíclicos Dirigidos), é possível agendar, monitorar e gerenciar tarefas de forma flexível, garantindo que os dados sejam processados na ordem correta e dentro dos prazos estabelecidos. Isso proporciona maior eficiência, confiabilidade e transparência na execução dos fluxos de dados.

## Monitoring and Backup




**Microsoft Azure:** Microsoft Azure é uma plataforma de computação em nuvem que oferece uma ampla gama de serviços para construção, implantação e gerenciamento de aplicativos e dados.

No monitoramento usamos o Azure Blob Storage no formato Parquet para armazenar cópias periódicas dos dados e modelos treinados. Essa abordagem garante a segurança dos dados, permitindo a rápida recuperação em caso de falhas ou perda de informações. Mesmo com a falta de um storage nativo do Databricks para esse fim, essa estratégia assegura a continuidade dos serviços e a integridade dos dados.



**Databricks:** Databricks Built-in Monitoring, é uma ferramenta que permite monitorar continuamente o desempenho da arquitetura. Através de alertas, o sistema detecta falhas ou quedas no desempenho dos modelos e pode iniciar automaticamente o processo de retrain com novos dados meteorológicos ou imagens da infraestrutura, mantendo a precisão das previsões. A funcionalidade pode ser utilizada na modalidade paga, proporcionando maior controle e automação no gerenciamento de dados e modelos.

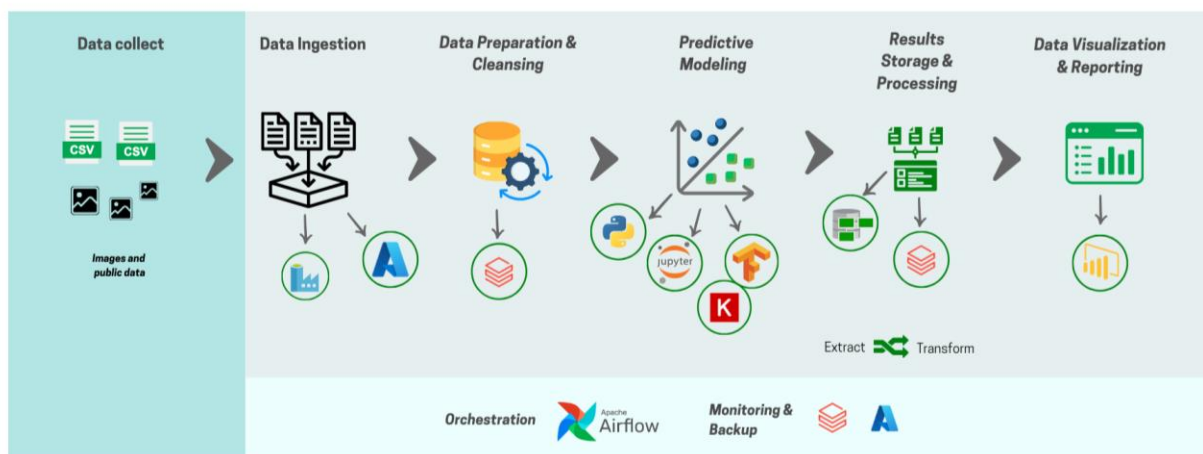
Ambiente de Visualização	
	<p><b>Microsoft Power BI:</b> Utilizaremos o Power BI para criação de dashboards, insights e a exibição tudo integrado à nossa plataforma, em outras palavras, o Power BI é onde todos os dados serão ingeridos para geração de insights, gráficos e dashboards.</p>

## 7.1 Arquitetura da solução

A arquitetura da **SolarWindIQ** visa otimizar para a geração de energia renovável, tendo em vista a incorporação de dados meteorológicos, históricos de produção e modelos preditivos avançados para proporcionar insights em tempo real e otimizar a manutenção dos sistemas de energia solar e eólica. Nossa solução abrange desde a ingestão de dados até a visualização de resultados, passando por tratamento de dados, treinamento de modelos de machine learning e deep learning e a entrega de insights valiosos para os tomadores de decisão.

Nossa arquitetura proporciona uma estratégia que abrange a redução de custo com processamento de dados das GPUs dos modelos, provedor e ferramentas de orquestração dos fluxos de dados, pois normalmente essas são as etapas onde mais se obtêm gastos dentro das áreas de tecnologia.

Segue abaixo nosso modelo de arquitetura



Outra vantagem é a utilização da prática de *arquitetura medallion*, que centraliza todos os dados num Lake House na etapa 3, o objetivo da utilização é aprimorar de forma contínua e progressiva a estrutura e a qualidade dos dados à medida que eles transitam por cada camada da arquitetura, passando de Bronze para Prata e, por fim, para Ouro e assim permitindo melhor controle das áreas de dados envolvidas e também custos, ao invés de usar vários provedores, sendo fácil de administrar, monitorar e efetividade na resolução de problemas.

### 7.1.1 Coleta de Dados: Onde olhamos os dados de energia

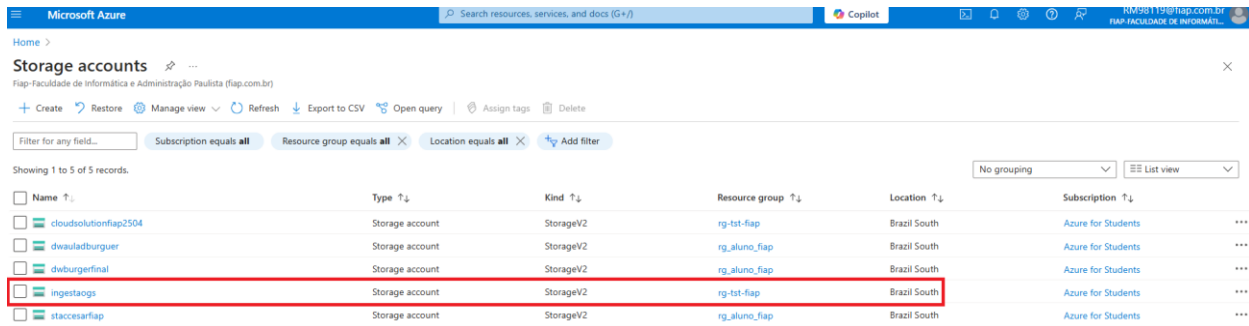
O primeiro passo da arquitetura envolve a coleta de dados provenientes de fontes públicas e externas relevantes para o setor de energia renovável. Esses dados incluem informações do Operador Nacional do Sistema Elétrico (ONS), Agência Nacional de Energia Elétrica (ANEEL) e CCEE e outras fontes, que fornecem dados sobre a operação e desempenho das usinas de energia. Além disso, dados meteorológicos, como insolação e velocidade do vento, são coletados de fontes públicas. Nessa etapa os dados são essenciais para a previsão precisa dos modelos nas etapas a diante.

### 7.1.2 Ingestão de Dados

Nessa etapa, a ingestão de dados é realizada através de ferramentas como Azure Data Factory, que automatiza a captura e transferência de dados para um repositório centralizado. Os dados são então armazenados em um Lake House, garantindo que o

sistema tenha capacidade de armazenar grandes volumes de dados de forma escalável e segura.

## Storage Account



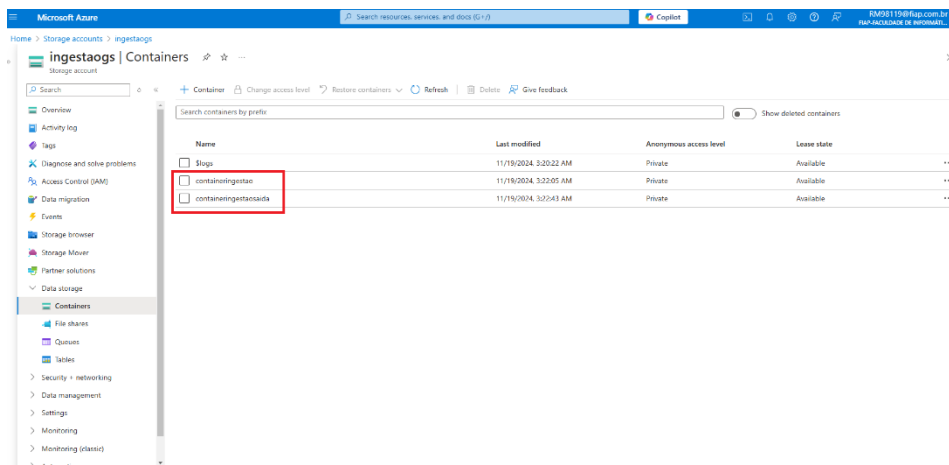
Microsoft Azure

Storage accounts

Showing 1 to 5 of 5 records.

Name	Type	Kind	Resource group	Location	Subscription
cloudsolutionflap2504	Storage account	StorageV2	rg-tst-flap	Brazil South	Azure for Students
dwsuladburguer	Storage account	StorageV2	rg_aluno_flap	Brazil South	Azure for Students
dwebgerfinal	Storage account	StorageV2	rg_aluno_flap	Brazil South	Azure for Students
ingestaogs	Storage account	StorageV2	rg-tst-flap	Brazil South	Azure for Students
staccsarflap	Storage account	StorageV2	rg_aluno_flap	Brazil South	Azure for Students

## Container de Origem e Saída



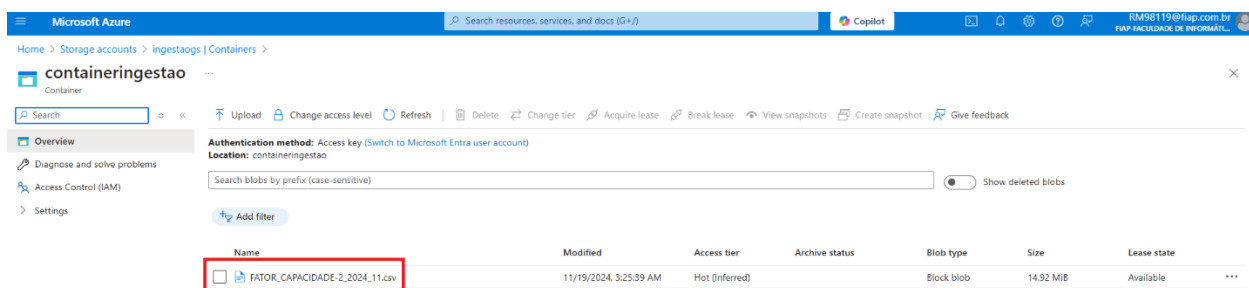
Microsoft Azure

ingestaogs | Containers

Search containers by prefix

Name	Last modified	Anonymous access level	Lease state
blobs	11/19/2024, 3:20:22 AM	Private	Available
containeringestao	11/19/2024, 3:20:05 AM	Private	Available
containeringestao_saida	11/19/2024, 3:20:43 AM	Private	Available

## Exemplo de um dos datasets dentro do container de origem



Microsoft Azure

containeringestao

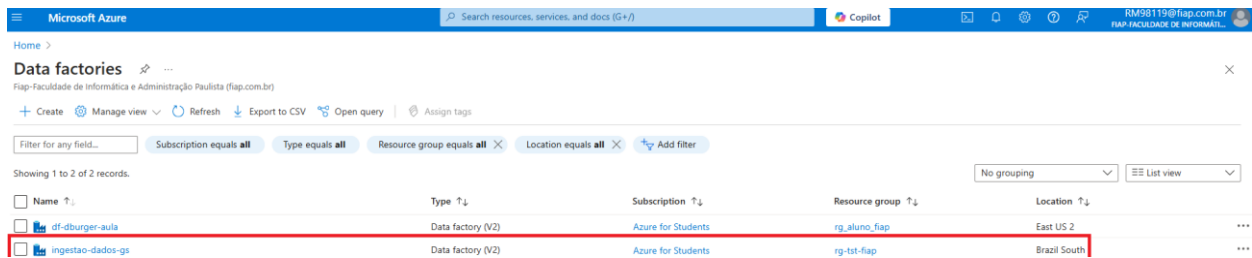
Authentication method: Access key (Switch to Microsoft Entra user account)

Location: containeringestao

Search blobs by prefix (case-sensitive)

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
FATOR_CAPACIDADE-2_2024_11.csv	11/19/2024, 3:25:39 AM	Hot (Inferred)		Block blob	14.92 MiB	Available

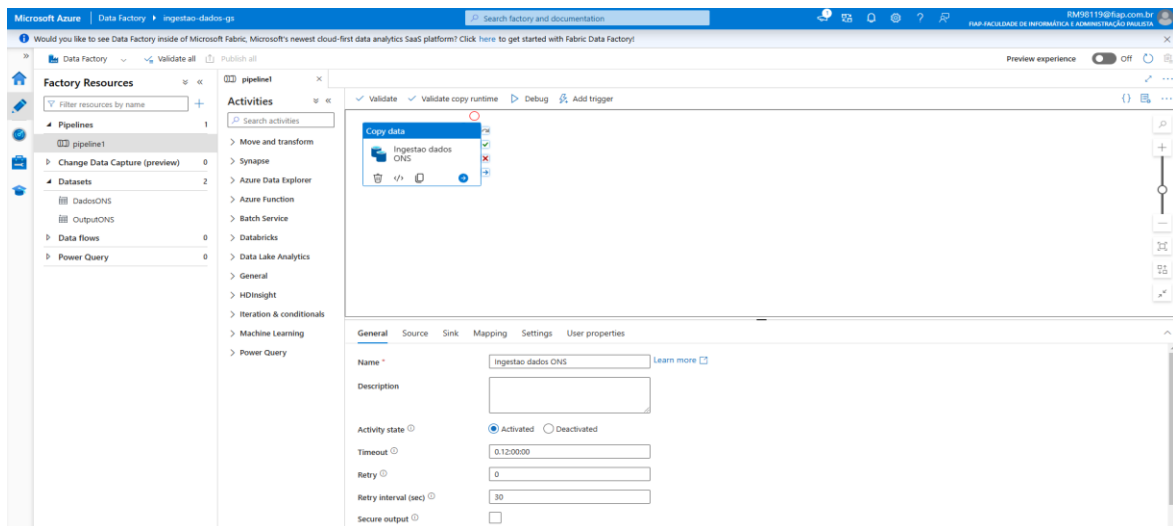
## Data Factory para ingestão de dados



The screenshot shows the Microsoft Azure Data Factory console. At the top, there's a search bar and a 'Copilot' button. Below that, the 'Data factories' section is active, showing a list of factories. A red box highlights the 'ingestao-dados-gs' factory, which is a Data factory (V2) located in Brazil South, under the rg-tst-fiap resource group.

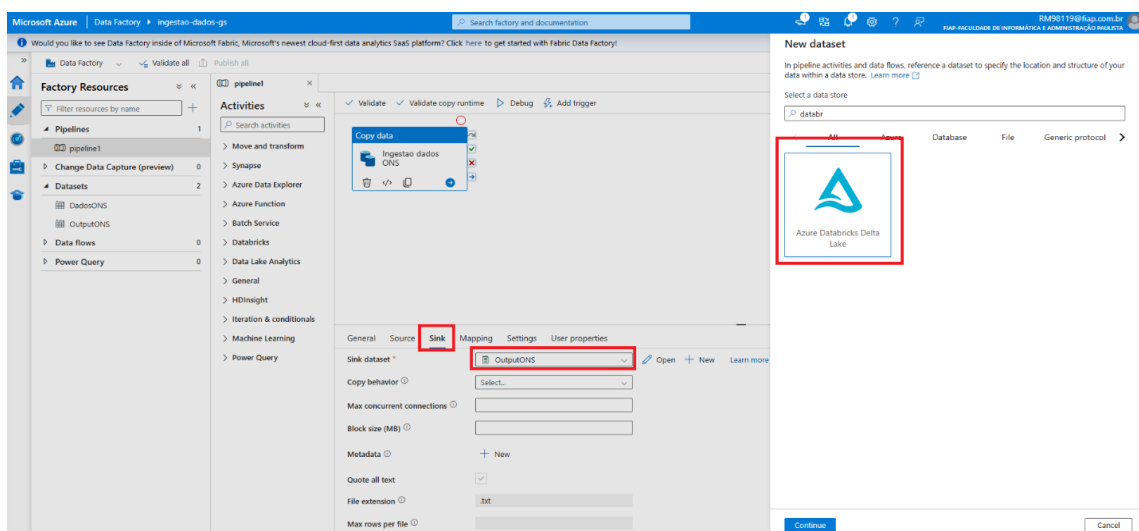
Name	Type	Subscription	Resource group	Location
df-dburger-aula	Data factory (V2)	Azure for Students	rg_aluno-fiap	East US 2
ingestao-dados-gs	Data factory (V2)	Azure for Students	rg-tst-fiap	Brazil South

## Exemplo dos datasets e pipeline de ingestão no Data Factory Studio



The screenshot shows the Data Factory Studio interface. On the left, the 'Factory Resources' pane shows a list of pipelines, datasets, and data flows. The 'pipeline1' is selected. In the center, the 'Activities' pane shows a 'Copy data' activity named 'Ingestao dados ONS'. The 'General' tab is active, showing the activity's configuration: Name 'Ingestao dados ONS', Description, Activity state 'Activated', Timeout '0:12:00:00', Retry '0', and Retry interval '30'.

## Depois da ingestão dos dados, destino é o Databricks Delta Lake



The screenshot shows the 'Sink' tab of the 'Copy data' activity in Data Factory Studio. The 'Sink dataset' is set to 'OutputONS'. A red box highlights the 'Sink' tab and the 'OutputONS' dataset. On the right, the 'New dataset' pane is open, showing the 'Azure Databricks Delta Lake' icon, which is also highlighted with a red box.

### 7.1.3 Tratamento e Pré-Processamento de dados

Uma vez que os dados são ingeridos, o tratamento e a preparação são realizados utilizando Azure Databricks e Python. O tratamento envolve a limpeza dos dados para remover inconsistências e valores ausentes, além de normalização e transformação para garantir que os dados estejam prontos para o treinamento de modelos, dentro do Lake House usando Arquitetura Medallion. O uso de Python permite personalizar o processo de tratamento de dados, proporcionando flexibilidade na aplicação de técnicas específicas de processamento.

```
Pré Processamento

1 # Verificação de valores ausentes
2 print(df.isnull().sum())

id_subistema      0
nom_subistema     0
id_estado         0
nom_estado        0
nom_pontoconexao  0
nom_localizacao   66528
val_latitudesecoletora 3984
val_longitudesecoletora 3984
val_latitudepontoconexao 7104
val_longitudepontoconexao 7104
nom_modalidadeoperacao 0
nom_tipousina     0
nom_usina_conjunto 0
id_ons            0
ceg              0
din_instante     0
val_geracaoprogramada 6072
val_geracaoverificada 0
val_capacidadeinstalada 0
val_fatorcapacidade 0
dtype: int64

[7] 1 # Remover ou preencher valores ausentes (se necessário)
    2 df = df.dropna() # Remove registros com valores nulos

[8] 1 # Remover colunas desnecessárias
    2 df = df.drop(['ceg', 'din_instante'], axis=1)

[9] 1 # Aplicar One-Hot Encoding
    2 categorical_columns = ['id_subistema', 'nom_subistema', 'id_estado', 'nom_estado',
    3 | | | | | | | | | | 'nom_pontoconexao', 'nom_localizacao', 'nom_modalidadeoperacao',
    4 | | | | | | | | | | 'nom_tipousina', 'nom_usina_conjunto', 'id_ons']
    5
    6 df_encoded = pd.get_dummies(df, columns=categorical_columns, drop_first=True)

[10] 1 # Selecionar colunas numéricas
    2 numerical_columns = ['val_latitudesecoletora', 'val_longitudesecoletora',
    3 | | | | | | | | | | 'val_latitudepontoconexao', 'val_longitudepontoconexao',
    4 | | | | | | | | | | 'val_capacidadeinstalada']
    5
    6 scaler = StandardScaler()
    7 df_encoded[numerical_columns] = scaler.fit_transform(df_encoded[numerical_columns])
```



### 7.1.4 Modelo de Machine Learning e Deep Learning para previsão

A modelagem preditiva é o coração da solução. Utilizando ferramentas como TensorFlow e Keras, são treinados modelos de machine learning e redes neurais (NLP) para prever a geração de energia renovável com base em dados meteorológicos e históricos de produção, para rodar os modelos será utilizada ambiente python. Esses modelos permitem a previsão de cenários futuros, ajudando as empresas a otimizarem o consumo e a produção, além de oferecer insights para planejamento de manutenção. Modelos generativos também são aplicados para criar cenários sintéticos, ampliando a capacidade preditiva sob condições climáticas variadas e gerando imagens para detectar anomalias em painéis solares e turbinas eólicas.

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.ensemble import RandomForestRegressor
6 from sklearn.metrics import mean_squared_error
7 import tensorflow as tf
8 from tensorflow.keras.models import Sequential
9 from tensorflow.keras.layers import Dense
10 import matplotlib.pyplot as plt
11 import seaborn as sns
12 import joblib
```

```
[4] 1 # Lista com os caminhos dos arquivos CSV
2 arquivos_csv = ['FATOR_CAPACIDADE-2_2024_11.csv', 'FATOR_CAPACIDADE-2_2024_10.csv', 'FATOR_CAPACIDADE-2_2024_09.csv']
3
4 # Ler e concatenar todos os arquivos
5 df = pd.concat([pd.read_csv(arquivo, sep=';') for arquivo in arquivos_csv], ignore_index=True)
6
7 # Verificar o DataFrame final
8 df.head(5)
```

	id_subsistema	nom_subsistema	id_estado	nom_estado	nom_pontoconexao	nom_localizacao	val_latitudeseletores	val_longitudeseletores	val_latitudepontoconexao	val_longitudepontoconexao	nom_modalidadeoperacao	nom_s
0	N	Norte	MA	MARANHAO	MIRANDA II 500kVA	NaN	-2.727222	-42.596389	-3.566219	-44.535478	Conjunto de Usinas	
1	NE	Nordeste	BA	BAHIA	IGAPORA II - 230 kV (B)	Interior	-14.102794	-42.609369	-14.030300	-42.624600	Conjunto de Usinas	
2	NE	Nordeste	BA	BAHIA	U SOBRADINHO - 500 kV (A)	Interior	-9.751812	-41.006198	-9.436083	-40.829167	Conjunto de Usinas	
3	NE	Nordeste	BA	BAHIA	MORRO CHAPEU2 - 230 kV (A)	Interior	-10.970000	-41.228000	-11.714700	-41.398900	Conjunto de Usinas	
4	NE	Nordeste	BA	BAHIA	OUROLANDIA II - 230 kV (A)	Interior	-11.119977	-41.290773	-10.874500	-41.339900	Conjunto de Usinas	

```

1 # Estatísticas descritivas
2 print(df.describe())

```

	val_latitudesecoleitora	val_longitudesecoleitora	\
count	387552.000000	387552.000000	
mean	-10.116329	-40.681855	
std	6.768870	4.465724	
min	-33.473520	-55.753600	
25%	-13.310839	-42.608611	
50%	-7.963652	-40.574048	
75%	-5.271972	-36.814212	
max	-2.727222	-35.447243	

	val_latitudepontoconexao	val_longitudepontoconexao	\
count	384432.000000	384432.000000	
mean	-10.051228	-40.618810	
std	6.680283	4.455356	
min	-33.473520	-55.753600	
25%	-12.583053	-42.614000	
50%	-8.033014	-40.300463	
75%	-5.377281	-36.985295	
max	-2.927658	-35.299061	

	val_geracaoprogramada	val_geracaoverificada	val_capacidadeinstalada	\
count	385464.000000	391536.000000	391536.000000	
mean	82.148278	83.037200	220.820831	
std	100.833379	109.265201	194.475976	
min	0.000000	0.000000	2.000000	
25%	13.500000	9.204000	90.000000	
50%	50.000000	45.932000	163.701000	
75%	110.500000	109.816250	277.800000	
max	927.000000	963.601000	1535.146800	

	val_fatorcapacidade
count	391536.000000
mean	0.379982
std	0.302463
min	0.000000
25%	0.079232
50%	0.360864
75%	0.630453
max	3.325608

### Pré Processamento

```

1 # Verificação de valores ausentes
2 print(df.isnull().sum())

```

id_sistema	0
nom_sistema	0
id_estado	0
nom_estado	0
nom_pontoconexao	0
nom_localizacao	66528
val_latitudesecoleitora	3984
val_longitudesecoleitora	3984
val_latitudepontoconexao	7104
val_longitudepontoconexao	7104
nom_modalidadeoperacao	0
nom_tiposina	0
nom_usina_conjunto	0
id_ons	0
ceg	0
din_instante	0
val_geracaoprogramada	6072
val_geracaoverificada	0
val_capacidadeinstalada	0
val_fatorcapacidade	0

dtype: int64

```

[7] 1 # Remover ou preencher valores ausentes (se necessário)
    2 df = df.dropna() # Remove registros com valores nulos

```

```

[8] 1 # Remover colunas desnecessárias
    2 df = df.drop(['ceg', 'din_instante'], axis=1)

```

```

[9] 1 # Aplicar One-Hot Encoding
2 categorical_columns = ['id_subsistema', 'nom_subsistema', 'id_estado', 'nom_estado',
3                       'nom_pontoconexao', 'nom_localizacao', 'nom_modalidadeoperacao',
4                       'nom_tipousina', 'nom_usina_conjunto', 'id_ons']
5
6 df_encoded = pd.get_dummies(df, columns=categorical_columns, drop_first=True)

[10] 1 # Selecionar colunas numéricas
2 numerical_columns = ['val_latitudesecoletora', 'val_longitudesecoletora',
3                     'val_latitudepontoconexao', 'val_longitudepontoconexao',
4                     'val_capacidadeinstalada']
5
6 scaler = StandardScaler()
7 df_encoded[numerical_columns] = scaler.fit_transform(df_encoded[numerical_columns])

```

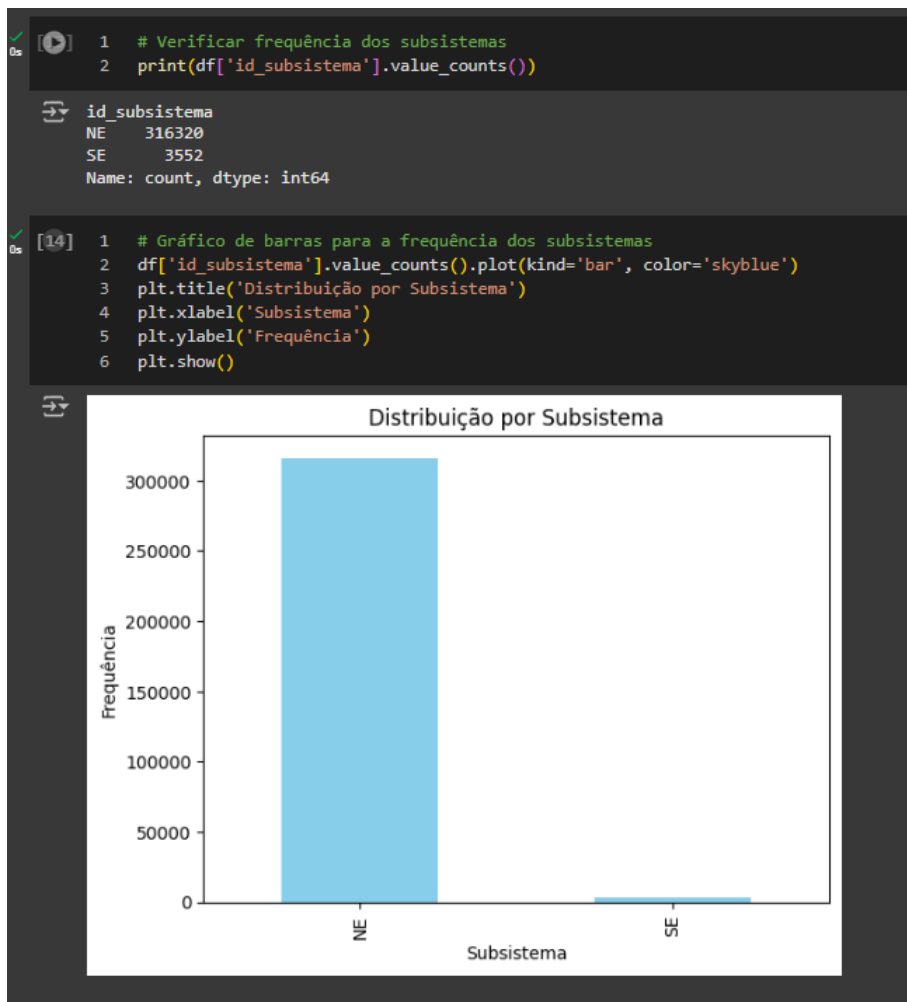
## Rápida Análise Exploratória

```

1 # Matriz de correlação com One-Hot Encoding aplicado
2 correlation_matrix = df_encoded.corr()

```





Modelo de ML e Deep Learning

```

[15] 1 # Dividir em X (features) e y (target)
2 X = df_encoded.drop('val_geracaoverificada', axis=1)
3 y = df_encoded['val_geracaoverificada']

[16] 1 # Divisão em treino e teste
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[17] 1 # Treinamento do modelo Random Forest
2 model_rf = RandomForestRegressor(n_estimators=100, random_state=42)
3 model_rf.fit(X_train, y_train)

```

RandomForestRegressor

RandomForestRegressor(random\_state=42)

```

[18] 1 # Fazer previsões no conjunto de teste
2 y_pred = model_rf.predict(X_test)

[19] 1 # Avaliar o modelo
2 rmse = np.sqrt(mean_squared_error(y_test, y_pred))
3 print(f"RMSE Random Forest: {rmse}")

```

RMSE Random Forest: 1.2528122829774282

```

1 # Modelo sequencial para rede neural
2 model_nn = Sequential([
3     Dense(64, activation='relu', input_dim=X_train.shape[1]),
4     Dense(32, activation='relu'),
5     Dense(1) # Saída contínua para regressão
6 ])

```

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an 'input\_shape'/'input\_dim' argument to a layer. When using Sequential models, prefer using an 'Input(shape)' object as the first layer in the model instead.  
super().\_\_init\_\_(activity\_regularizer=activity\_regularizer, \*\*kwargs)

```
1 # Compilação
2 model_nn.compile(optimizer='adam', loss='mean_squared_error')
```

```
1 # Treinamento
2 model_nn.fit(X_train, y_train, validation_split=0.2, epochs=50, batch_size=32)
```

6398/6398 ————— 12s 2ms/step - loss: 16.0146 - val\_loss: 5.4144  
Epoch 23/50  
6398/6398 ————— 12s 2ms/step - loss: 14.3312 - val\_loss: 9.2233  
Epoch 24/50  
6398/6398 ————— 12s 2ms/step - loss: 18.9229 - val\_loss: 165.8202  
Epoch 25/50  
6398/6398 ————— 12s 2ms/step - loss: 20.6110 - val\_loss: 11.8055  
Epoch 26/50  
6398/6398 ————— 11s 2ms/step - loss: 13.1842 - val\_loss: 7.5818  
Epoch 27/50  
6398/6398 ————— 21s 2ms/step - loss: 10.1916 - val\_loss: 3.3726  
Epoch 28/50  
6398/6398 ————— 12s 2ms/step - loss: 9.0462 - val\_loss: 4.5849  
Epoch 29/50  
6398/6398 ————— 12s 2ms/step - loss: 14.3755 - val\_loss: 20.3709  
Epoch 30/50  
6398/6398 ————— 23s 2ms/step - loss: 15.0724 - val\_loss: 2.2605  
Epoch 31/50  
6398/6398 ————— 12s 2ms/step - loss: 10.0792 - val\_loss: 3.3706  
Epoch 32/50  
6398/6398 ————— 14s 2ms/step - loss: 12.8171 - val\_loss: 27.1776  
Epoch 33/50  
6398/6398 ————— 18s 2ms/step - loss: 10.7875 - val\_loss: 2.4656  
Epoch 34/50  
6398/6398 ————— 21s 2ms/step - loss: 13.7080 - val\_loss: 3.1210  
Epoch 35/50  
6398/6398 ————— 12s 2ms/step - loss: 12.4848 - val\_loss: 2.6339  
Epoch 36/50  
6398/6398 ————— 20s 2ms/step - loss: 11.1319 - val\_loss: 2.5837  
Epoch 37/50  
6398/6398 ————— 13s 2ms/step - loss: 9.3930 - val\_loss: 1.7447  
Epoch 38/50  
6398/6398 ————— 13s 2ms/step - loss: 8.6308 - val\_loss: 2.9178  
Epoch 39/50  
6398/6398 ————— 13s 2ms/step - loss: 10.5624 - val\_loss: 10.7721  
Epoch 40/50  
6398/6398 ————— 20s 2ms/step - loss: 10.1717 - val\_loss: 1.2494  
Epoch 41/50  
6398/6398 ————— 15s 2ms/step - loss: 9.5904 - val\_loss: 1.2971  
Epoch 42/50  
6398/6398 ————— 18s 2ms/step - loss: 9.5461 - val\_loss: 4.4665  
Epoch 43/50  
6398/6398 ————— 12s 2ms/step - loss: 8.2437 - val\_loss: 9.8311  
Epoch 44/50  
6398/6398 ————— 12s 2ms/step - loss: 8.2147 - val\_loss: 7.0739  
Epoch 45/50  
6398/6398 ————— 12s 2ms/step - loss: 6.7656 - val\_loss: 1.6083  
Epoch 46/50  
6398/6398 ————— 20s 2ms/step - loss: 7.2886 - val\_loss: 4.8216  
Epoch 47/50  
6398/6398 ————— 21s 2ms/step - loss: 7.8597 - val\_loss: 6.7324  
Epoch 48/50  
6398/6398 ————— 13s 2ms/step - loss: 9.9163 - val\_loss: 5.7456  
Epoch 49/50  
6398/6398 ————— 12s 2ms/step - loss: 7.5806 - val\_loss: 1.1945  
Epoch 50/50  
6398/6398 ————— 21s 2ms/step - loss: 6.4658 - val\_loss: 0.8586  
<keras.src.callbacks.history.History at 0x7c59957be110>

```

[23] 1 # Avaliação
2 y_pred_rm = model_m.predict(X_test)
3 print(f"RMSE Neural Network: (np.sqrt(mean_squared_error(y_test, y_pred_rm)))")

20000/20000 3s 1ms/step
RMSE Neural Network: 0.9355335324158309

[24] 1 # Salvar os modelos
2 joblib.dump(model_rf, 'modelo_rf.pkl') # Random Forest
3 model_nn.save('modelo_nn.h5') # Rede Neural

WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.save_model(model)'. This file format is considered legacy. We recommend using instead the native Keras format, e.g. 'model.save('my_model.keras')' or 'keras.saving.save_model(model, 'my_model.keras')'.

[25] 1 # Adicionar previsões ao Dataframe
2 y_pred_full = model_rf.predict(X)
3 df_encoded['val_geracaoverificada_prevista'] = y_pred_full

[26] 1 # Conferir resultados
2 print(df_encoded[['val_geracaoverificada', 'val_geracaoverificada_prevista']].head())

val_geracaoverificada val_geracaoverificada_prevista
1 80.495 80.76214
2 151.749 152.18284
3 129.911 129.97982
4 346.934 346.92183
5 152.568 152.49725

```

```

[27] 1 df_encoded.head(5)
2

val_latitudeseletores val_longitudeseletores val_latitudepontoconexao val_longitudepontoconexao val_geracaoprogramada val_geracaoverificada val_capacidadeinstalada val_fatorcapacidade id_subistema_86 nom_subistema_Sudeste/Centro-Oeste ... id_ens_CDU
1 -1.719128 -1.207602 -1.704312 -1.213711 77.0 80.495 -0.209677 0.479994 False False ...
2 -0.502068 -0.628851 -0.398132 -0.562542 181.5 151.749 0.157479 0.663558 False False ...
3 -0.842020 -0.708022 -1.045965 -0.769173 135.0 129.911 -0.397500 0.951729 False False ...
4 -0.884772 -0.731584 -0.807088 -0.747775 337.0 346.934 0.975048 0.951888 False False ...
5 -1.217481 -1.111386 -1.215046 -1.111525 137.5 152.568 -0.165792 0.871821 False False ...

5 rows x 484 columns

[28] 1 df_encoded.shape
(319872, 484)

```

```

[29] 1 # Adicionar a coluna 'val_geracaoverificada_prevista' do df_encoded ao df
2 df[['val_geracaoverificada_prevista'] = df_encoded['val_geracaoverificada_prevista']]
3
4 df_final = df
5
6 # Verificar se a coluna foi adicionada corretamente
7 df_final.head(5)
8

id_estado nom_estado nom_pontoconexao nom_localizacao val_latitudeseletores val_longitudeseletores val_latitudepontoconexao val_longitudepontoconexao nom_modalidadeoperacao nom_tiposusina nom_usina_conjunto id_ens val_geracaoprogramada val_geracaoverificada val_geracaoverificada_prevista
BA BAVBA IGAPORAII - 230 kV (B) Interior -14.102794 -42.609369 -14.030300 -42.624600 Conjunto de Usinas Eólica Conj Açaçis CJU_BAARA 77.0
BA BAVBA U SOBRADINHO - 600 kV (A) Interior -9.751812 -41.005198 -9.436083 -40.829167 Conjunto de Usinas Eólica Conj Atzema CJU_BAARI 181.5
BA BAVBA MORRO CHAPEL2 - 230 kV (A) Interior -10.970000 -41.228000 -11.714700 -41.398900 Conjunto de Usinas Eólica Conj Babônia CJU_BABAB 135.0
BA BAVBA OURILANDIA II - 230 kV (A) Interior -11.119977 -41.296773 -10.874500 -41.339900 Conjunto de Usinas Eólica Conj Babilônia Sul CJU_BABBS 337.0
BA BAVBA BROT MACALBAS - 230 kV (A) Interior -12.309409 -42.342846 -12.309409 -42.342846 Conjunto de Usinas Eólica Conj Brotas de Macalbas CJU_BABMC 137.5

```

```

[30] 1 # Remover duplicatas mantendo a linha com o maior valor na coluna 'capacidade'
2 df_final = df_final.loc[df.groupby('nom_usina_conjunto')['val_geracaoverificada_prevista'].idxmax()]
3 df_final.head(5)

id_subistema nom_subistema id_estado nom_estado nom_pontoconexao nom_localizacao val_latitudeseletores val_longitudeseletores val_latitudepontoconexao val_longitudepontoconexao nom_modalidadeoperacao nom_tiposusina nom_usina_conjunto
65320 NE Nordeste RN GRANDE DO NORTE ACU II - 230 kV (A) Litoral -5.115283 -36.384401 -5.593044 -36.908622 Tipo I Eólica Alegria I F
197096 NE Nordeste RN GRANDE DO NORTE ACU II - 230 kV (A) Litoral -5.097279 -36.402501 -5.593044 -36.908622 Tipo I Eólica Alegria II F
154933 NE Nordeste RN GRANDE DO NORTE ACU II - 138 kV (A) Litoral -5.546084 -37.826396 -5.593044 -36.908622 Tipo II-B Solar Assi V R
219394 NE Nordeste CE ACARAU II - 230 kV (A) Litoral -2.841442 -40.866143 -2.927658 -40.114352 Tipo II-B Eólica Cataventos Acarau I CI
93490 NE Nordeste BA BAVBA PINDAI II - 230 kV (A) Interior -14.353933 -42.575842 -14.415433 -42.607767 Conjunto de Usinas Eólica Conj ABI I CJU_

Próximas etapas: Gerar código com df_final Ver gráficos recomendados New interactive sheet + Código + Texto

[31] 1 df_final.shape
(189, 19)

[32] 1 df_final.to_csv('df_final.csv', index=False)

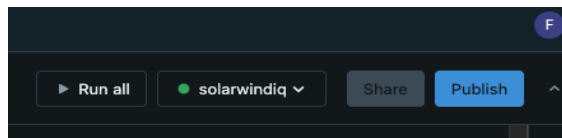
```

### 7.1.5 Armazenamento de Resultados

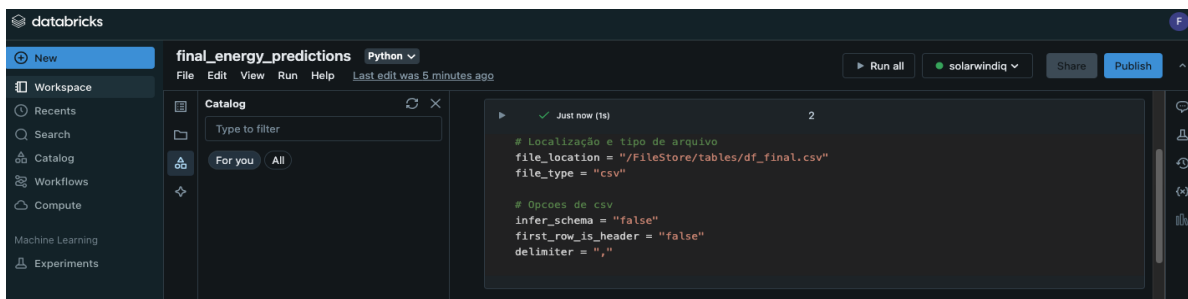
Os resultados dos modelos preditivos são armazenados em uma arquitetura de Lake house, que combina as vantagens de um Data Lake com as capacidades de processamento de um Data Warehouse dentro do Databricks aplicado às práticas da arquitetura medallion já citadas anteriormente. Isso garante a flexibilidade necessária para armazenar dados em grande escala e, ao mesmo tempo, possibilita consultas rápidas e eficientes, mantendo os dados organizados, acessíveis e prontos para serem plotados na ferramenta de visualização de dados.

Carregamento da tabela dos resultados gerados pelo modelo no databricks utilizando python, utilizando o **Databricks Community Edition**:

- Seleccionamos a máquina do projeto:



- Localizamos o carregamento da base:





- Carregamos a base dos resultados:

Run all solarwindiq Share Publish

Just now (2s) 3 Python

```
# Carregar o arquivo no DataFrame
df = spark.read.format(file_type)\
    .option("inferSchema", infer_schema)\
    .option("header", "true")\
    .option("sep", delimiter)\
    .load(file_location)

# Exibir os dados para verificar os nomes das colunas
display(df)
```

(2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [id\_subistema: string, nom\_subistema: string ... 17 more fields]

Table +

	id_subistema	nom_subistema	id_estado	nom_estado	nom_ponti
1	NE	Nordeste	RN	RIO GRANDE DO NOR...	ACU II - 230 kV
2	NE	Nordeste	RN	RIO GRANDE DO NOR...	ACU II - 230 kV
3	NE	Nordeste	RN	RIO GRANDE DO NOR...	ACU II - 138 kV
4	NE	Nordeste	CE	CEARA	ACARAU II - 230 kV
5	NE	Nordeste	BA	BAHIA	PINDAI II - 230 kV
6	NE	Nordeste	BA	BAHIA	IGAPORA III - 230 kV
7	NE	Nordeste	CE	CEARA	ACARAU II - 66 kV

- Criamos uma tabela temporária:

Run all solarwindiq Share Publish

1 minute ago (<1s) 4

```
# Criar uma Temp View
temp_table_name = "df_final_csv"
df.createOrReplaceTempView(temp_table_name)
```

- No primeiro bloco, criamos a tabela permanente e a carregamos a partir da tabela temporária e no segundo temos a consulta a tabela permanente, em seguida publicamos:

▶ Run all

solarwindiq ▼

Share

Publish

▶ 2 minutes ago (6s)

5

```
# Nome da tabela permanente
permanent_table_name = "final_energy_predictions"

# Salvar como tabela permanente
df.write.format("parquet").saveAsTable(permanent_table_name)
```

▶ (1) Spark Jobs

▶ Just now (5s)

6

SQL

🗑️

🔍

⋮

```
%sql
SELECT * FROM final_energy_predictions
```

▶ (1) Spark Jobs

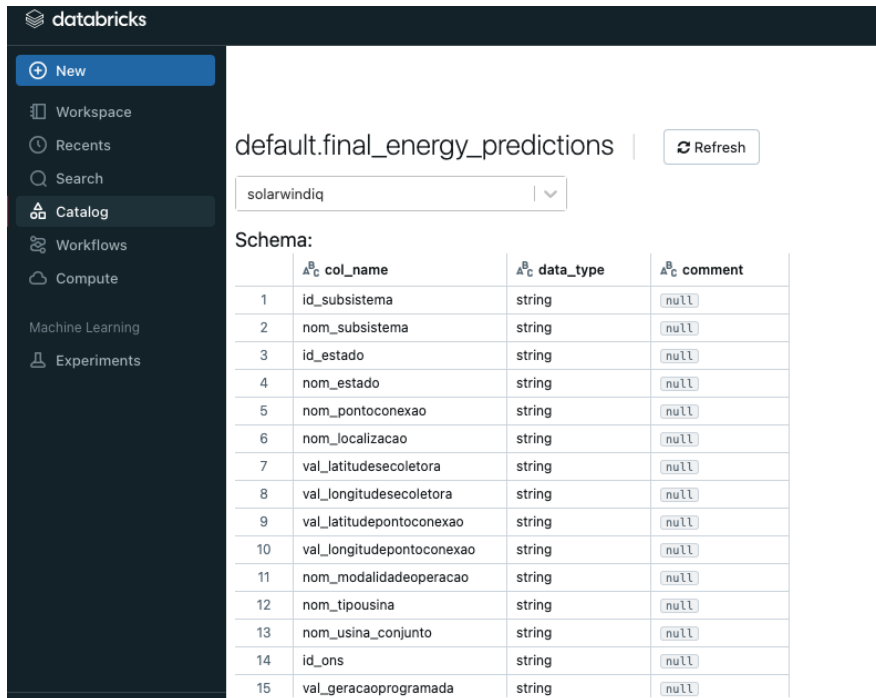
▶ \_sqldf: pyspark.sql.dataframe.DataFrame = [id\_subsistema: string, nom\_subsistema: string ... 17 more fields]

Table ▼ +

🔍 🔍 🗑️

	id_subsistema	nom_subsistema	id_estado	nom_estado	nom_pontu
1	NE	Nordeste	RN	RIO GRANDE DO NOR...	ACU II - 230 kV
2	NE	Nordeste	RN	RIO GRANDE DO NOR...	ACU II - 230 kV
3	NE	Nordeste	RN	RIO GRANDE DO NOR...	ACU II - 138 kV
4	NE	Nordeste	CE	CEARA	ACARAU II - 230 kV

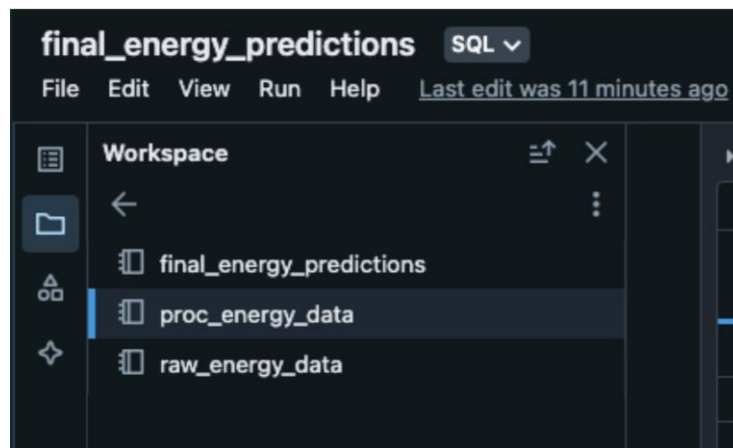
- Quando consultamos a tabela a mesma está dessa forma no default do catálogo:



The screenshot shows the Databricks interface with the 'Catalog' tab selected. The table 'default.final\_energy\_predictions' is displayed, filtered by 'solarwindiq'. The schema table below lists 15 columns, all of which are of type 'string' and have a 'null' comment.

	col_name	data_type	comment
1	id_subistema	string	null
2	nom_subistema	string	null
3	id_estado	string	null
4	nom_estado	string	null
5	nom_pontoconexao	string	null
6	nom_localizacao	string	null
7	val_latitudesecoleitora	string	null
8	val_longitudesecoleitora	string	null
9	val_latitudepontoconexao	string	null
10	val_longitudepontoconexao	string	null
11	nom_modalidadeoperacao	string	null
12	nom_tipousina	string	null
13	nom_usina_conjunto	string	null
14	id_ons	string	null
15	val_geracaoprogramada	string	null

- No final pensando no modelo de arquitetura medallion com a exemplo de energia a SolarWindIQ estaria dessa forma no SQL Editor:



### 7.1.6 Visualização de Dados e Insights

A entrega dos resultados ao usuário final é feita através de dashboards e relatórios interativos no Power BI. O Power BI permite que os dados e as previsões geradas pelos modelos sejam visualizados de forma clara e dinâmica, facilitando a interpretação dos resultados por parte dos gestores. Além disso, a visualização de anomalias geradas pelas redes neurais ajuda a identificar áreas que precisam de manutenção preventiva, otimizando a performance do sistema de energia renovável.

Pensando de forma eficaz e estratégica que possa atingir o público-alvo da **SolarWindIQ**, foi desenvolvida a persona abaixo, na qual se encaixa com os objetivos dessa solução:

Link de visualização da imagem: [https://www.canva.com/design/DAGWsgCg8-Y/KvZbIWweKiHkgKhGillCTw/view?utm\\_content=DAGWsgCg8-Y&utm\\_campaign=designshare&utm\\_medium=link&utm\\_source=editor](https://www.canva.com/design/DAGWsgCg8-Y/KvZbIWweKiHkgKhGillCTw/view?utm_content=DAGWsgCg8-Y&utm_campaign=designshare&utm_medium=link&utm_source=editor)

**Ana Clara Martins**

Idade: 37 Anos  
Endereço: Avenida Ninolopis, 345  
Porto Alegre - RS  
Ocupação: Engenheira de Energia Renovável

Ana Clara Martins é uma engenheira apaixonada por construir um futuro sustentável. Formada em Engenharia de Energia Renovável pela PUCRS, ela combina seu conhecimento técnico com uma visão prática para enfrentar os desafios da transição energética. Durante sua formação, Ana Clara desenvolveu expertise em tecnologias como energia solar e eólica, além de uma base sólida em gestão de sistemas energéticos. Focada em inovação, ela utiliza IA, análise e visualização de dados para enfrentar desafios como intermitência na geração, detecção de falhas e previsão de demandas, garantindo eficiência, redução de custos e impacto ambiental positivo, alinhando sua trajetória profissional à construção de um futuro mais limpo e equilibrado.

**NECESSIDADES**

- Soluções que ofereçam dados atualizados sobre geração e desempenho de infraestrutura.
- Ferramentas baseadas em IA para prever falhas e otimizar o planejamento de manutenção.
- Modelos analíticos robustos que ajudem a alinhar geração de energia com a demanda projetada.

**PAIN POINTS**

- Dificuldade em gerenciar a variabilidade de fontes renováveis, como solar e eólica, impactando a estabilidade, perdas operacionais e o planejamento.
- Falta de ferramentas eficazes para detectar rapidamente problemas em painéis solares ou turbinas eólicas.
- Incertezas na previsão de demanda e produção, levando a decisões menos precisas e aumento de desperdícios.

**MOTIVAÇÕES**

- Curiosa
- Orientada a Dados
- Racional
- Análítica

**INTERESSES**

- Explorar novas tecnologias para melhorar a eficiência e o impacto ambiental das operações.
- Interesse em dashboards intuitivos e insights estratégicos que suportem decisões fundamentadas.
- Adotar soluções disruptivas que alinhem eficiência energética com avanços em sustentabilidade.

**TECNOLOGIAS**

WhatsApp, Instagram, X, Office 365, slack, SAP

*"Insights são sempre bem-vindos, pois nos permitem aproveitar a tecnologia e a inovação para transformar os desafios energéticos em oportunidades para um futuro mais sustentável e equilibrado."*  
Ana Clara Martins

Descrição:

- Nome: Ana Clara Martins
- Idade: 37 Anos
- Endereço: Avenida Nilópolis, 345 - Porto Alegre - RS
- ocupação: Engenheira de Energia Renovável

Descrição: Ana Clara Martins é uma engenheira apaixonada por construir um futuro sustentável. Formada em Engenharia de Energia Renovável pela PUCRS, ela combina

seu conhecimento técnico com uma visão prática para enfrentar os desafios da transição energética. Durante sua formação, Ana Clara desenvolveu expertise em tecnologias como energia solar e eólica, além de uma base sólida em gestão de sistemas energéticos.

Focada em inovação, ela utiliza IA, análise e visualização de dados para enfrentar desafios como intermitência na geração, detecção de falhas e previsão de demandas, garantindo eficiência, redução de custos e impacto ambiental positivo, alinhando sua trajetória profissional à construção de um futuro mais limpo e equilibrado.

#### Necessidades:

- Soluções que ofereçam dados atualizados sobre geração e desempenho de infraestrutura.
- Ferramentas baseadas em IA para prever falhas e otimizar o planejamento de manutenção.
- Modelos analíticos robustos que ajudem a alinhar geração de energia com a demanda projetada.

#### Interesses:

- Explorar novas tecnologias para melhorar a eficiência e o impacto ambiental das operações.
- Interesse em dashboards intuitivos e insights estratégicos que suportem decisões fundamentadas.
- Adotar soluções disruptivas que alinhem eficiência energética com avanços em sustentabilidade.

#### Pain Points:

- Dificuldade em gerenciar a variabilidade de fontes renováveis, como solar e eólica, impactando a estabilidade, perdas operacionais e o planejamento.
- Falta de ferramentas eficazes para detectar rapidamente problemas em painéis solares ou turbinas eólicas.
- Incertezas na previsão de demanda e produção, levando a decisões menos precisas e aumento de desperdícios.

#### 7.1.6.1 KPI'S e Métricas

Abaixo seguem KPI's e Métricas:

- **Taxa de Fator de Geração Prevista:** O percentual previsto comparando previsão de geração x capacidade prevista.
- **Percentual por Subsistema:** A proporção de geração prevista por subsistema da rede nacional e que geram energia Eólica e Solar.

- **Previsão de Geração Solar e Eólica:** Volume previsto de MWh por ponto/localização geográfica.
- **Classificação por Localização Geográfica:** Precisão exata dos locais com potencial gerador, faz com que a análise e comparação entre eles seja eficiente.

#### 7.1.6.2 Dicionário de Dados/Metadados

Dataset 1: df\_final.xlsx

COLUNAS	DESCRIÇÃO
id_subsistema	Sigla do Subsistema de Energia do Brasil
nom_subsistema	Nome do Subsistema de Energia do Brasil
id_estado	Sigla do Estado (UF)
nom_estado	Nome do Estado
nom_pontoconexao	Nome do Ponto de Conexão com a Rede Elétrica
nom_localizacao	Nome da Localização do conjunto
val_latitudese coletora	Latitude do Conjunto
val_longitudese coletora	Longitude do Conjunto
val_latitudepontoconexao	Latitude Ponto de Conexão
val_longitudepontoconexao	Longitude Ponto de Conexão
nom_modalidadeoperacao	Nome do tipo de operação da usina
nom_tipousina	Nome do tipo da geração da usina
nom_usina_conjunto	Nome da usina
id_ons	Identificador no ONS
val_geracaoprogramada	Geração Programada em MWh
val_geracaoverificada	Valor Gerado em MWh
val_capacidadeinstalada	Capacidade de geração em MWw
val_fatorcapacidade	Valor da divisão da geração verificada pela capacidade, quanto mai próximo de 1 mais eficiente a usina é
val_geracaoverificada_prevista	Valor previsto por modelo de machine learning e deep learning para produção do próximo mês

Dataset 2: preco\_mensal

COLUNAS	DESCRIÇÃO
SUL	Valor Médio dos últimos 3 meses do subsistema SUL



Gráfico de Colunas Clusterizado:



Gráfico demonstrando o total do potencial de geração prevista por subsistema e por tipo (eólica e solar).

Tabela:

**Tabela Comparativa por Estado**

Estado	Geração Prevista (MWm)	Fator Capacidade Prev.	Capacidade Instalada
PIAUI	6.149,40	0,98	6.980,75
PARAIBA	1.196,88	0,93	1.284,93
PERNAMBUCO	1.700,53	0,93	1.771,03
CEARA	2.879,56	0,92	3.068,92
MINAS GERAIS	1.793,06	0,91	2.058,13
BAHIA	11.993,89	0,91	13.301,48
RIO GRANDE DO NORTE	9.926,43	0,89	11.272,23
<b>Total</b>	<b>35.639,75</b>	<b>0,91</b>	<b>39.737,46</b>

O Bahia possui a maior geração prevista (11,9 mil MWh) e excelente fator de capacidade (0,91), mostrando liderança em energia renovável.

Tabela Comparativa por Estado, demonstrando a geração prevista, fator de capacidade prevista e capacidade instalada por estado. Essa visualização permite uma visão mais generalista por Estado, identificando o potencial de cada um.



## Gráfico de Barras Clusterizado:



Gráfico demonstrando Geração Prevista por Estado e tipo de usina, demonstrando o potencial de cada estado para cada tipo de geração.

## Mapa:



Mapa destacando os 7 estados com maior potencial eólico e solar.

## Filtros:

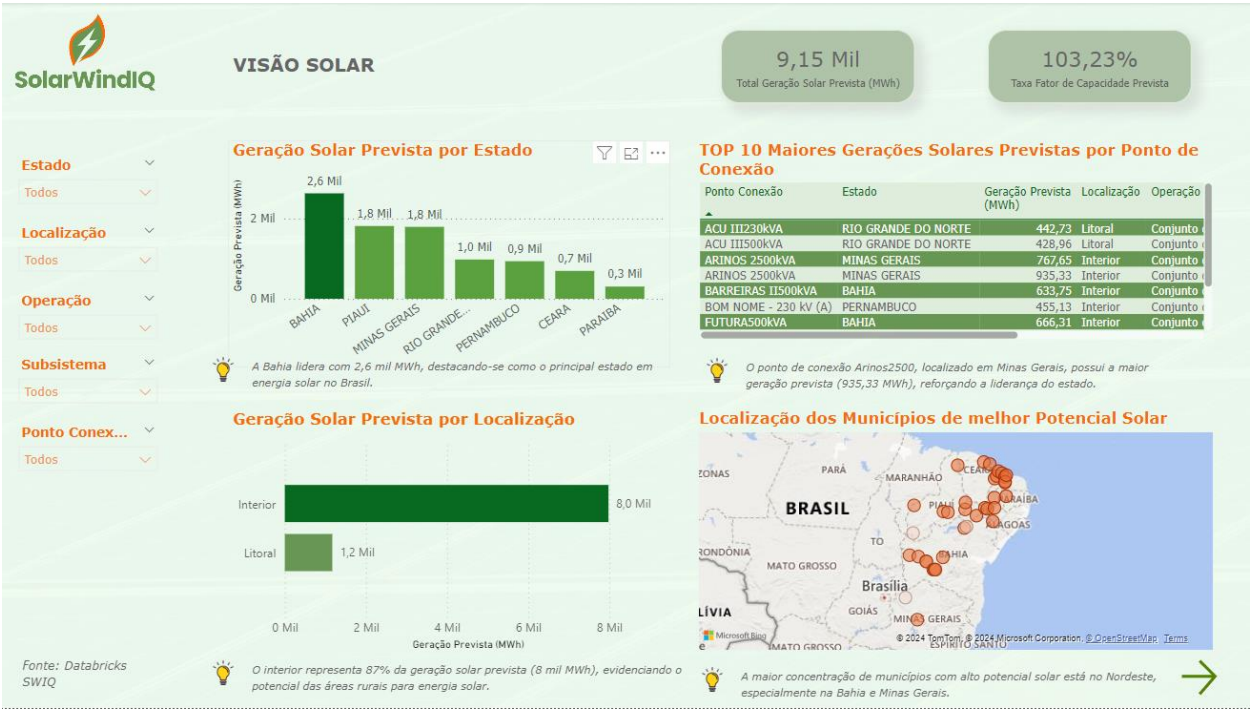
**Estado** ▼  
Todos

**Subsistema** ▼  
Todos

**Tipo Usina** ▼  
Todos

Filtros por Estado, Subsistema e Tipo Usina para facilitar a interação e análise dos dados.

Painel 2: Visão Solar



Painel focado no aspecto da geração solar, possibilitando análise mais granulares desse tipo de geração.

Cards:



Cards indicando o total de geração solar prevista por MWh e a taxa de fator de capacidade prevista, que excede em 3,23% as capacidades instaladas, mostrando o enorme potencial.

Gráfico de Colunas Clusterizado:



Gráfico demonstrando em ordem decrescente os estados com maior potencial de geração solar prevista em MWh, destacando-se em verde escuro o Estado da Bahia.

Tabela:

Ponto Conexão	Estado	Geração Prevista (MWh)	Localização	Operação
ACU III230kVA	RIO GRANDE DO NORTE	442,73	Litoral	Conjunto
ACU III500kVA	RIO GRANDE DO NORTE	428,96	Litoral	Conjunto
ARINOS 2500kVA	MINAS GERAIS	767,65	Interior	Conjunto
ARINOS 2500kVA	MINAS GERAIS	935,33	Interior	Conjunto
BARREIRAS II500kVA	BAHIA	633,75	Interior	Conjunto
BOM NOME - 230 kV (A)	PERNAMBUCO	455,13	Interior	Conjunto
FUTURA500kVA	BAHIA	666,31	Interior	Conjunto

O ponto de conexão Arinos2500, localizado em Minas Gerais, possui a maior geração prevista (935,33 MWh), reforçando a liderança do estado.

Tabela Comparativa por Ponto de Conexão, demonstrando o Estado, a geração prevista, a localização, tipo de operação, subsistema, latitude e longitude.

Gráfico de Barras Clusterizado:



Gráfico indicando qual região, litoral ou interior, possui mais potencial de geração previsto. Nota-se que o interior possui uma enorme diferença do litoral.

Mapa:



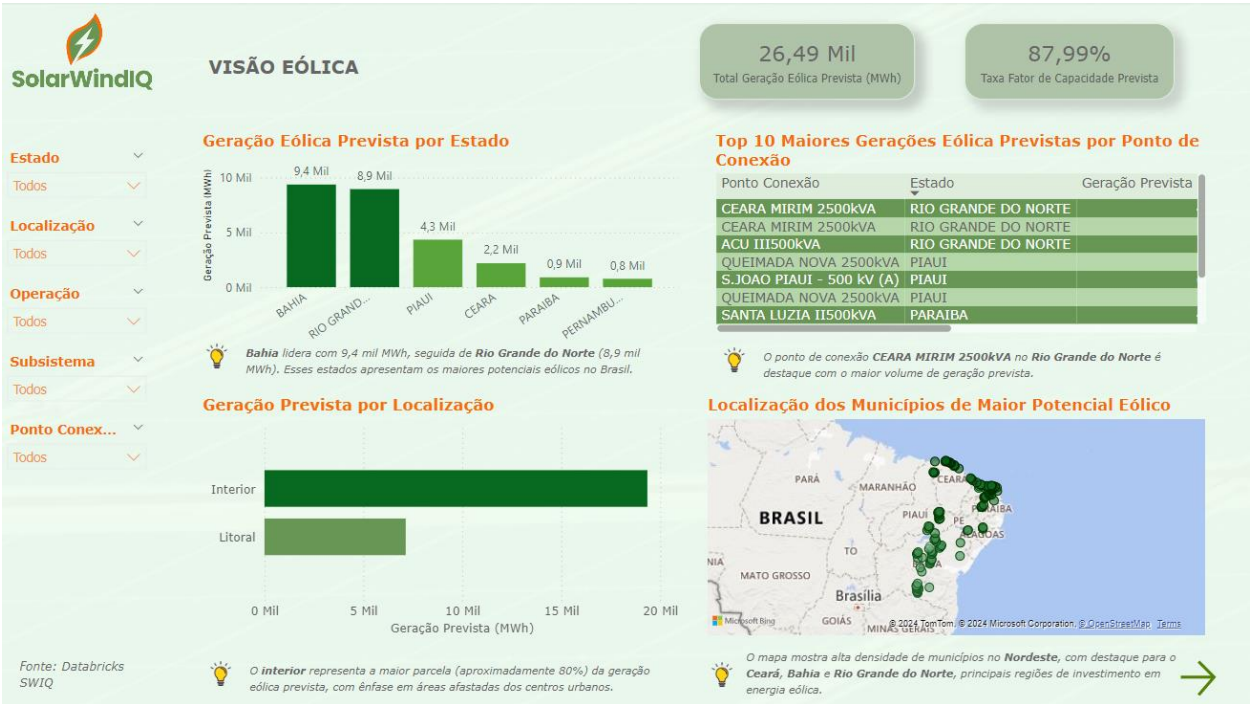
Mapa indicando os municípios com melhor potencial solar. Tal visão pode ser extremamente útil para planejamento energético e econômico de regiões.

Filtros:

<b>Estado</b>	<b>Subsistema</b>
Todos	Todos
<b>Localização</b>	<b>Ponto Conex...</b>
Todos	Todos
<b>Operação</b>	
Todos	

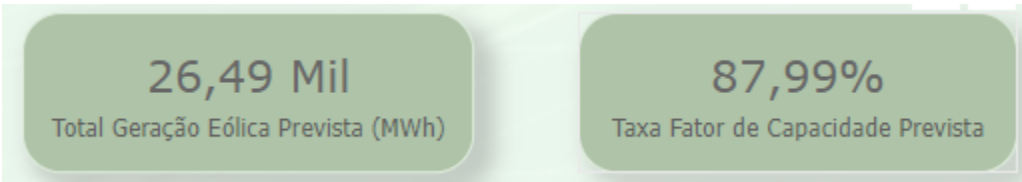
Filtros por Estado, Localização, Operação, Subsistema e Pontos de Conexão

Painel 3: Visão Eólica



Painel focado no aspecto da geração eólica, possibilitando análise mais granulares desse tipo de geração.

Cards:



Cards indicando o total de geração solar prevista por MWh e a taxa de fator de capacidade prevista, demonstrando que ainda há cerca de 12% para ainda ser explorado.

Gráfico de Colunas Clusterizado:



Gráfico demonstrando em ordem decrescente os estados com maior potencial de geração eólica prevista em MWh, destacando em verde escuro o Estado da Bahia e Rio Grande do Norte.

Tabela:

**Top 10 Maiores Gerações Eólica Previstas por Ponto de Conexão**

Ponto Conexão	Estado	Geração Prevista
CEARA MIRIM 2500kVA	RIO GRANDE DO NORTE	
CEARA MIRIM 2500kVA	RIO GRANDE DO NORTE	
ACU III500kVA	RIO GRANDE DO NORTE	
QUEIMADA NOVA 2500kVA	PIAUI	
S.JOAO PIAUI - 500 kV (A)	PIAUI	
QUEIMADA NOVA 2500kVA	PIAUI	
SANTA LUZIA II500kVA	PARAIBA	

*O ponto de conexão CEARA MIRIM 2500kVA no Rio Grande do Norte é destaque com o maior volume de geração prevista.*

Tabela Comparativa por Ponto de Conexão, demonstrando o Estado, a geração prevista, a localização, tipo de operação, subsistema, latitude e longitude.



Gráfico de Barras Clusterizado:



Gráfico indicando qual região, litoral ou interior, possui mais potencial de geração previsto. Nota-se que o litoral ganha mais potencial comparando com o litoral da visão solar.

Mapa:



Mapa indicando os municípios com melhor potencial solar. Tal visão pode ser extremamente útil para planejamento energético e econômico de regiões.

Filtros:

Estado

Todos

Localização

Todos

Operação

Todos

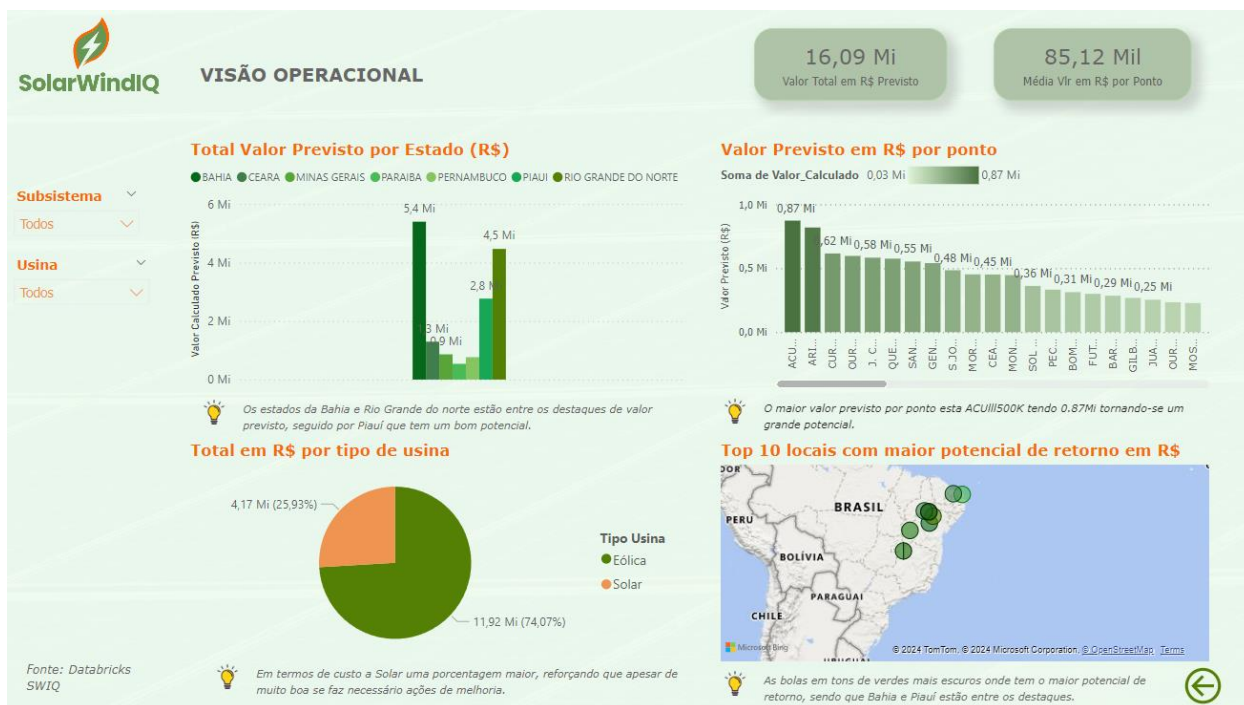
Subsistema

Todos

Ponto Conexão

Todos

#### Painel 4: Visão Operacional



Este painel permite uma visão voltada e focada para os custos e retorno dos investimentos. Os cálculos e métricas desse painel foram feitos baseados na média de PLD (Preço de Liquidez) divulgados pela CCEE diariamente. Dessa forma, foi possível calcular uma previsão de retorno de investimento e lucratividade para a geração de energia eólica e solar.



Cards:



Cards demonstrando o valor total previsto em R\$ considerando o potencial de geração e a média de valor em R\$ por ponto/localização.

Gráfico de Colunas Clusterizado:



Gráfico demonstrando a potencial rentabilidade por Estado, note que Bahia e Rio Grande do Norte demonstram maiores potenciais.

Gráfico de Colunas Clusterizado:



Gráfico demonstrando o valor previsto por ponto/localização em ordem decrescente.

Gráfico de Pizza:

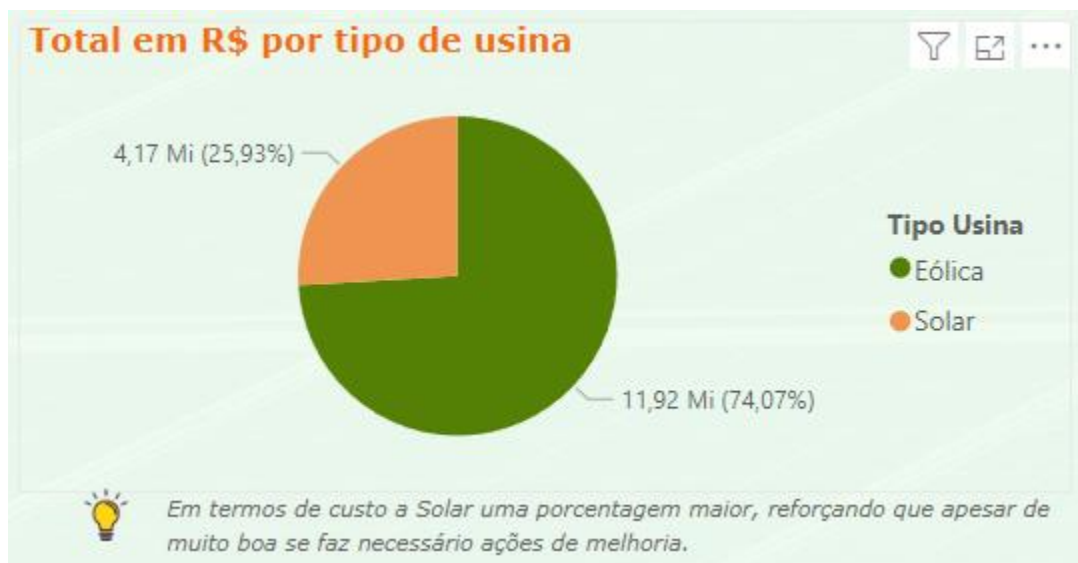


Gráfico demonstrando o total previsto em R\$ por tipo de usina/geração. O gráfico em pizza facilita a visão de dois atributos.

Mapa:



Mapa destacando os 10 maiores pontos/locais com maior potencial de retorno de investimento.

Filtros:

**Subsistema** ▾

Todos ▾

**Usina** ▾

Todos ▾

Filtros por Subsistema e Tipo de Usina

### 7.1.7 Etapas avulsas na Arquitetura:

- **Monitoramento e Automação**

Para garantir que o sistema funcione de forma eficiente, o Databricks Built-in Monitoring é utilizado para monitorar continuamente o desempenho da arquitetura e. Alertas são configurados para detectar falhas ou quedas no desempenho dos modelos, e o sistema pode acionar um processo de retrain automático com novos dados meteorológicos ou imagens da infraestrutura para manter a precisão das previsões ao longo do tempo, podendo ser utilizada na modalidade paga.

- **Backup e Recuperação de Dados**

Realizar o backup é uma etapa crucial, pois, ao lidarmos com dados históricos, é fundamental garantir que, em caso de indisponibilidade, haja uma cópia segura desses dados. Para isso, utilizamos o Databricks SQL, integrando com o *Azure Blob Storage* via o formato Parquet, que permite armazenar cópias periódicas dos dados e modelos treinados. Essa estratégia assegura que, em situações de falha ou perda de dados, o sistema possa ser rapidamente restaurado, garantindo a integridade das informações e a continuidade dos serviços. Além disso, ela considera a segurança, caso a plataforma fique inativa, já que o Databricks não oferece um storage próprio para esse fim.

- **Modelagem das tabelas da arquitetura medallion**

Para obter o histórico e entender a estrutura das tabelas, utilizamos o Oracle Data Modeler, uma ferramenta open-source da Oracle para modelagem de dados. Nas camadas L2 e L3, adotamos o modelo Star Schema, que é amplamente reconhecido como uma das melhores práticas para organizar e visualizar dados. Na camada L1, mantemos os dados brutos, sem transformações, seguindo um modelo de dados transacional, diretamente extraído das fontes originais durante a ingestão.

- **Orquestração de dados - Airflow:**

Visando a orquestração e otimização dos custos da SolarWindIQ, escolhemos uma ferramenta que oferece flexibilidade e escalabilidade. Utilizando o Airflow, conseguimos adaptar os processos de forma dinâmica às demandas, assegurando maior eficiência e agilidade na entrega de soluções.

- **Instalação e DAGs:**

Atualização das dependências do sistema:

```
Last metadata expiration check: 3:09:07 ago on Fri 20 Sep 2024 07:22:03 PM GMT.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
kernel-uek	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	2.5 M
kernel-uek-core	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	61 M
kernel-uek-devel	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	21 M
kernel-uek-modules	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	69 M
Upgrading:				
audit	x86_64	3.1.2-1.0.1.el8	ol8_baseos_latest	264 k
audit-libs	x86_64	3.1.2-1.0.1.el8	ol8_baseos_latest	124 k
bind-export-libs	x86_64	32:9.11.36-16.el8_10.2	ol8_baseos_latest	1.1 M
bind-libs	x86_64	32:9.11.36-16.el8_10.2	ol8_appstream	176 k
bind-libs-lite	x86_64	32:9.11.36-16.el8_10.2	ol8_appstream	1.2 M
bind-license	noarch	32:9.11.36-16.el8_10.2	ol8_appstream	104 k
bind-utils	x86_64	32:9.11.36-16.el8_10.2	ol8_appstream	453 k
bpftool	x86_64	5.15.0-210.163.7.el8uek	ol8_UEKR7	3.2 M
bubblewrap	x86_64	0.4.0-2.el8_10	ol8_baseos_latest	50 k
ca-certificates	noarch	2024.2.69.v5.0.303-80.0.el8_10	ol8_baseos_latest	981 k
cloud-init	noarch	23.4-7.0.1.el8_10.7	ol8_appstream	1.3 M
curl	x86_64	7.61.1-34.el8_10.2	ol8_baseos_latest	352 k
firewalld	noarch	0.9.11-8.0.1.el8_10	ol8_baseos_latest	509 k
firewalld-filesystem	noarch	0.9.11-8.0.1.el8_10	ol8_baseos_latest	78 k
glibc	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	2.2 M
glibc-common	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	1.0 M
glibc-devel	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	90 k
glibc-gconv-extra	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	1.6 M
glibc-headers	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	495 k
glibc-langpack-en	x86_64	2.28-251.0.2.el8_10.4	ol8_baseos_latest	834 k
initscripts	x86_64	10.00.18-1.0.2.el8	ol8_baseos_latest	339 k
iproute	x86_64	6.8.0-2.el8_10	ol8_UEKR7	873 k
iproute-tc	x86_64	6.8.0-2.el8_10	ol8_UEKR7	451 k

```
3.64.181.164.115 (opc)
initscripts-10.00.18-1.0.2.el8.x86_64
iproute-tc-6.8.0-2.el8_10.x86_64
krb5-devel-1.18.2-29.0.1.el8_10.x86_64
libcurl-7.61.1-34.el8_10.2.x86_64
libipa_hbac-2.9.4-4.0.1.el8_10.x86_64
libkadm5-1.18.2-29.0.1.el8_10.x86_64
libsss_certmap-2.9.4-4.0.1.el8_10.x86_64
libsss_idmap-2.9.4-4.0.1.el8_10.x86_64
mdadm-4.2-14.0.4.el8_10.x86_64
nss-3.101.0-7.el8_8.x86_64
nss-softokn-freebl-3.101.0-7.el8_8.x86_64
nss-util-3.101.0-7.el8_8.x86_64
openssh-clients-8.0p1-25.0.1.el8_10.x86_64
pcp-5.3.7-22.0.1.el8_10.x86_64
pcp-doc-5.3.7-22.0.1.el8_10.noarch
pcp-pmda-dm-5.3.7-22.0.1.el8_10.x86_64
pcp-pmda-openmetrics-5.3.7-22.0.1.el8_10.x86_64
pcp-system-tools-5.3.7-22.0.1.el8_10.x86_64
platform-python-setuptools-39.2.0-8.el8_10.noarch
python3-bind-32:9.11.36-16.el8_10.2.noarch
python3-firewall-0.9.11-8.0.1.el8_10.noarch
python3-libdnf-0.63.0-20.0.1.el8_10.x86_64
python3-setuptools-39.2.0-8.el8_10.noarch
python3-sssdconfig-2.9.4-4.0.1.el8_10.noarch
python3-oci-sdk-2.134.0-1.el8.x86_64
source-highlight-3.1.8-18.el8_10.x86_64
sssd-ad-2.9.4-4.0.1.el8_10.x86_64
sssd-common-2.9.4-4.0.1.el8_10.x86_64
sssd-ipa-2.9.4-4.0.1.el8_10.x86_64
sssd-krb5-2.9.4-4.0.1.el8_10.x86_64
sssd-ldap-2.9.4-4.0.1.el8_10.x86_64
sssd-proxy-2.9.4-4.0.1.el8_10.x86_64
Installed:
kernel-uek-5.15.0-210.163.7.el8uek.x86_64
kernel-uek-devel-5.15.0-210.163.7.el8uek.x86_64
iproute-6.8.0-2.el8_10.x86_64
jose-10-2.el8_10.3.x86_64
krb5-libs-1.18.2-29.0.1.el8_10.x86_64
libdnf-0.63.0-20.0.1.el8_10.x86_64
libjose-10-2.el8_10.3.x86_64
libsss_autofs-2.9.4-4.0.1.el8_10.x86_64
libsss_idmap-2.9.4-4.0.1.el8_10.x86_64
libsss_sudo-2.9.4-4.0.1.el8_10.x86_64
network-scripts-10.00.18-1.0.2.el8.x86_64
nss-softokn-3.101.0-7.el8_8.x86_64
nss-sysinit-3.101.0-7.el8_8.x86_64
openssh-8.0p1-25.0.1.el8_10.x86_64
openssh-server-8.0p1-25.0.1.el8_10.x86_64
pcp-conf-5.3.7-22.0.1.el8_10.x86_64
pcp-libs-5.3.7-22.0.1.el8_10.x86_64
pcp-pmda-nfsclient-5.3.7-22.0.1.el8_10.x86_64
pcp-selinux-5.3.7-22.0.1.el8_10.x86_64
pcp-zeroconf-5.3.7-22.0.1.el8_10.x86_64
python3-audit-3.1.2-1.0.1.el8.x86_64
python3-dnf-plugin-ulininfo-0.3-3.module+el8.10.0+90274+07ba55de.noarch
python3-hawkey-0.63.0-20.0.1.el8_10.x86_64
python3-pcp-5.3.7-22.0.1.el8_10.x86_64
python3-setuptools-wheel-39.2.0-8.el8_10.noarch
python3-urllib3-1.24.2-8.el8_10.noarch
sos-4.7.2-2.0.1.el8_10.noarch
sssd-2.9.4-4.0.1.el8_10.x86_64
sssd-client-2.9.4-4.0.1.el8_10.x86_64
sssd-common-pac-2.9.4-4.0.1.el8_10.x86_64
sssd-kcm-2.9.4-4.0.1.el8_10.x86_64
sssd-krb5-common-2.9.4-4.0.1.el8_10.x86_64
sssd-nfs-idmap-2.9.4-4.0.1.el8_10.x86_64
wget-1.19.5-12.0.1.el8_10.x86_64
Complete!
```

- Instalação das dependências necessárias do airflow:

```
Last metadata expiration check: 3:14:25 ago on Fri 20 Sep 2024 07:22:03 PM GMT.
Package gcc-8.5.0-22.0.1.el8_10.x86_64 is already installed.
Package python36-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 is already installed.
Package python3-pip-9.0.3-24.el8.noarch is already installed.
Dependencies resolved.
=====
Package                                Architecture      Version                                Repository          Size
=====
Installing:
git                                     x86_64            2.43.5-1.el8_10                       ol8_appstream       91 k
libpq-devel                            x86_64            13.11-1.el8                            ol8_appstream       98 k
python3-devel                           x86_64            3.6.8-39.module+el8.10.0+90274+07ba55de ol8_appstream       15 k
Installing dependencies:
git-core                                x86_64            2.43.5-1.el8_10                       ol8_appstream       11 M
git-core-doc                            noarch            2.43.5-1.el8_10                       ol8_appstream       3.1 M
libpq                                    x86_64            13.11-1.el8                            ol8_appstream       198 k
perl-Error                              noarch            1:0.17025-2.el8                       ol8_appstream        46 k
perl-Git                                 noarch            2.43.5-1.el8_10                       ol8_appstream        78 k
perl-TermReadKey                         x86_64            2.37-7.el8                             ol8_appstream        40 k
platform-python-devel                   x86_64            3.6.8-62.0.1.el8_10                   ol8_appstream       240 k
python-rpm-macros                       noarch            3-45.el8                               ol8_appstream        16 k
python-srpm-macros                      noarch            3-45.el8                               ol8_appstream        16 k
python3-rpm-generators                  noarch            5-8.el8                                ol8_appstream        25 k
python3-rpm-macros                      noarch            3-45.el8                               ol8_appstream        15 k
Transaction Summary
=====
Install 14 Packages

Total download size: 15 M
Installed size: 48 M
Downloading Packages:
(1/14): git-core-doc-2.43.5-1.el8_10.noarch.rpm                                19 MB/s | 3.1 MB 00:00
(2/14): git-2.43.5-1.el8_10.x86_64.rpm                                         556 kB/s | 91 kB 00:00
```

```

3.64.181.164.115 (opc)
Installing      : git-core-doc-2.43.5-1.el8_10.noarch                4/14
Installing      : python3-rpm-macros-3-45.el8.noarch                5/14
Installing      : python3-rpm-generators-5-8.el8.noarch             6/14
Installing      : platform-python-devel-3.6.8-62.0.1.el8_10.x86_64  7/14
Installing      : perl-TermReadKey-2.37-7.el8.x86_64               8/14
Installing      : perl-Error-1:0.17025-2.el8.noarch                 9/14
Installing      : perl-Git-2.43.5-1.el8_10.noarch                  10/14
Installing      : git-2.43.5-1.el8_10.x86_64                       11/14
Installing      : libpq-13.11-1.el8.x86_64                        12/14
Installing      : libpq-devel-13.11-1.el8.x86_64                  13/14
Installing      : python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 14/14
Running scriptlet: python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 14/14
Verifying       : git-2.43.5-1.el8_10.x86_64                     1/14
Verifying       : git-core-2.43.5-1.el8_10.x86_64                 2/14
Verifying       : git-core-doc-2.43.5-1.el8_10.noarch             3/14
Verifying       : libpq-13.11-1.el8.x86_64                       4/14
Verifying       : libpq-devel-13.11-1.el8.x86_64                  5/14
Verifying       : perl-Error-1:0.17025-2.el8.noarch               6/14
Verifying       : perl-Git-2.43.5-1.el8_10.noarch                 7/14
Verifying       : perl-TermReadKey-2.37-7.el8.x86_64              8/14
Verifying       : platform-python-devel-3.6.8-62.0.1.el8_10.x86_64 9/14
Verifying       : python-rpm-macros-3-45.el8.noarch               10/14
Verifying       : python-srpm-macros-3-45.el8.noarch              11/14
Verifying       : python3-rpm-generators-5-8.el8.noarch           12/14
Verifying       : python3-rpm-macros-3-45.el8.noarch              13/14
Verifying       : python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64 14/14

Installed:
git-2.43.5-1.el8_10.x86_64      git-core-2.43.5-1.el8_10.x86_64
git-core-doc-2.43.5-1.el8_10.noarch  libpq-13.11-1.el8.x86_64
libpq-devel-13.11-1.el8.x86_64  perl-Error-1:0.17025-2.el8.noarch
perl-Git-2.43.5-1.el8_10.noarch  perl-TermReadKey-2.37-7.el8.x86_64
platform-python-devel-3.6.8-62.0.1.el8_10.x86_64  python-rpm-macros-3-45.el8.noarch
python-srpm-macros-3-45.el8.noarch  python3-rpm-generators-5-8.el8.noarch
python3-rpm-macros-3-45.el8.noarch  python36-devel-3.6.8-39.module+el8.10.0+90274+07ba55de.x86_64

Complete!

```

- Atualização do Python dentro do ambiente e do pip, para não haver erros na instalação, pois o airflow depende do python para rodar:

```

Collecting pip
  Downloading https://files.pythonhosted.org/packages/d4/55/90db48d85f7689ec6f81c0db0622d704306c5284850383c090e6c7195a5c/pip-24.2-py3-none-any.whl (1.8MB)
    |#####| 1.8MB 22.2MB/s
Installing collected packages: pip
  Found existing installation: pip 19.3.1
  Uninstalling pip-19.3.1:
    Successfully uninstalled pip-19.3.1
  Successfully installed pip-24.2

```

- Instalação do Airflow, utilizando o pip install apache-airflow==2.2.5

```

Collecting apache-airflow==2.2.5
  Using cached apache-airflow-2.2.5-py3-none-any.whl.metadata (98 kB)
Collecting alembic<2.0,>=1.5.1 (from apache-airflow==2.2.5)
  Downloading alembic-1.13.2-py3-none-any.whl.metadata (7.4 kB)
Collecting argcomplete<3.0,>=1.10 (from apache-airflow==2.2.5)
  Downloading argcomplete-2.1.2-py3-none-any.whl.metadata (17 kB)
Collecting attrs<21.0,>=20.0 (from apache-airflow==2.2.5)
  Using cached attrs-20.3.0-py2.py3-none-any.whl.metadata (10 kB)
Collecting blinker (from apache-airflow==2.2.5)
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting clickclicks==1.2 (from apache-airflow==2.2.5)
  Using cached clickclick-20.10.2-py2.py3-none-any.whl.metadata (7.6 kB)
Collecting colorlog<7.0,>=4.0.2 (from apache-airflow==2.2.5)
  Downloading colorlog-6.8.2-py3-none-any.whl.metadata (10 kB)
Collecting connexion==2.10.0 (from connexion[flask,swagger-ui]>=2.10.0->apache-airflow==2.2.5)
  Downloading connexion-3.1.0-py3-none-any.whl.metadata (12 kB)
Collecting croniter>=0.3.17 (from apache-airflow==2.2.5)
  Downloading croniter-3.0.3-py2.py3-none-any.whl.metadata (28 kB)
Collecting cryptography>=0.9.3 (from apache-airflow==2.2.5)
  Downloading cryptography-43.0.1-cp37-abi3-manylinux_2_28_x86_64.whl.metadata (5.4 kB)
Collecting deprecated==1.2.13 (from apache-airflow==2.2.5)
  Downloading Deprecated-1.2.14-py2.py3-none-any.whl.metadata (5.4 kB)
Collecting dill<0.4,>=0.2.2 (from apache-airflow==2.2.5)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Collecting docutils<0.17 (from apache-airflow==2.2.5)
  Using cached docutils-0.16-py2.py3-none-any.whl.metadata (2.7 kB)
Collecting flask<2.0,>=1.1.0 (from apache-airflow==2.2.5)
  Using cached Flask-1.1.4-py2.py3-none-any.whl.metadata (4.6 kB)
Collecting flask-appbuilder==3.4.5 (from apache-airflow==2.2.5)
  Using cached Flask-AppBuilder-3.4.5-py3-none-any.whl.metadata (10 kB)
Collecting flask-caching<2.0.0,>=1.5.0 (from apache-airflow==2.2.5)
  Downloading Flask-Caching-1.11.1-py3-none-any.whl.metadata (2.2 kB)

```



```

Requirement already satisfied: apache-airflow in ./venv/lib/python3.8/site-packages (2.2.5)
Requirement already satisfied: alembic<2.0,>=1.5.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.13.2)
Requirement already satisfied: argcomplete<3.0,>=1.10 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.1.2)
Requirement already satisfied: attrs<21.0,>=20.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (20.3.0)
Requirement already satisfied: blinker in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.8.2)
Requirement already satisfied: clickclick>=1.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (20.10.2)
Requirement already satisfied: colorlog<7.0,>=4.0.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (6.8.2)
Requirement already satisfied: connexion>=2.10.0 in ./venv/lib/python3.8/site-packages (from connexion[flask,swagger-ui]>=2.10.0->apache-airflow) (2.14.2)
Requirement already satisfied: croniter>=0.3.17 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.0.3)
Requirement already satisfied: cryptography>=0.9.3 in ./venv/lib/python3.8/site-packages (from apache-airflow) (43.0.1)
Requirement already satisfied: deprecated>=1.2.13 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.2.14)
Requirement already satisfied: dill<0.4,>=0.2.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.3.8)
Requirement already satisfied: docutils<0.17 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.16)
Requirement already satisfied: flask<2.0,>=1.1.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.1.4)
Requirement already satisfied: flask-appbuilder==3.4.5 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.4.5)
Requirement already satisfied: flask-caching<2.0.0,>=1.5.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.11.1)
Requirement already satisfied: flask-login<0.5,>=0.3 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.4.1)
Requirement already satisfied: flask-session<0.4.0,>=0.3.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.4.0)
Requirement already satisfied: flask-wtf<0.15,>=0.14.3 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.14.3)
Requirement already satisfied: graphviz>=0.12 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.20.3)
Requirement already satisfied: gunicorn>=20.1.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (23.0.0)
Requirement already satisfied: httpx in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.27.2)
Requirement already satisfied: iso8601>=0.1.12 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.1.0)
Requirement already satisfied: itsdangerous<2.0,>=1.1.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.1.0)
Requirement already satisfied: jinja2<3.1,>=2.10.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.11.3)
Requirement already satisfied: jsonschema==3.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.2.0)
Requirement already satisfied: lazy-object-proxy in ./venv/lib/python3.8/site-packages (from apache-airflow) (1.10.0)
Requirement already satisfied: lockfile>=0.12.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (0.12.2)
Requirement already satisfied: markdown<4.0,>=2.5.2 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.7)
Requirement already satisfied: markupsafe<2.1.0,>=1.1.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.0.1)
Requirement already satisfied: marshmallow-oneofschema>=2.0.1 in ./venv/lib/python3.8/site-packages (from apache-airflow) (3.1.1)
Requirement already satisfied: packaging>=14.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (24.1)
Requirement already satisfied: pendulum>=2.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (2.1.2)
Requirement already satisfied: psutil<6.0.0,>=4.2.0 in ./venv/lib/python3.8/site-packages (from apache-airflow) (5.9.8)

```

```

Requirement already satisfied: dnspython>=1.15.0 in ./venv/lib/python3.8/site-packages (from email-validator<2,>=1.0.5->flask-appbuilder==3.4.5->apache-airflow) (2.6.1)
Requirement already satisfied: Babel>=2.3 in ./venv/lib/python3.8/site-packages (from Flask-Babel<3,>=1->flask-appbuilder==3.4.5->apache-airflow) (2.16.0)
Requirement already satisfied: mdurl<=0.1 in ./venv/lib/python3.8/site-packages (from markdown-it-py>=2.2.0->rich>=9.2.0->apache-airflow) (0.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in ./venv/lib/python3.8/site-packages (from requests<3,>=2.9.1->connexion>=2.10.0->connexion[flask,swagger-ui]>=2.10.0->apache-airflow) (3.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./venv/lib/python3.8/site-packages (from requests<3,>=2.9.1->connexion>=2.10.0->connexion[flask,swagger-ui]>=2.10.0->apache-airflow) (2.2.3)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in ./venv/lib/python3.8/site-packages (from aiohttp->apache-airflow-providers-http->apache-airflow) (2.4.0)
Requirement already satisfied:aiosignal>=1.1.2 in ./venv/lib/python3.8/site-packages (from aiohttp->apache-airflow-providers-http->apache-airflow) (1.3.1)
Requirement already satisfied:frozenset>=1.1.1 in ./venv/lib/python3.8/site-packages (from aiohttp->apache-airflow-providers-http->apache-airflow) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in ./venv/lib/python3.8/site-packages (from aiohttp->apache-airflow-providers-http->apache-airflow) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in ./venv/lib/python3.8/site-packages (from aiohttp->apache-airflow-providers-http->apache-airflow) (1.11.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in ./venv/lib/python3.8/site-packages (from aiohttp->apache-airflow-providers-http->apache-airflow) (4.0.3)
Requirement already satisfied: exceptiongroup>=1.0.2 in ./venv/lib/python3.8/site-packages (from anyio->httpx->apache-airflow) (1.2.2)

```

- Inicialização do banco com o airflow db init:

```

INFO [airflow.models.dag] Creating ORM DAG for example_time_delta_sensor_async
INFO [airflow.models.dag] Setting next_dagrun for example_bash_operator to 2024-09-19 00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_branch_datetime_operator_2 to 2024-09-19 00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_branch_dop_operator_v3 to 2024-09-20T22:48:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_branch_labels to 2024-09-19 00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_branch_operator to 2024-09-19 00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_complex to None
INFO [airflow.models.dag] Setting next_dagrun for example_dag_decorator to None
INFO [airflow.models.dag] Setting next_dagrun for example_external_task_marker_child to None
INFO [airflow.models.dag] Setting next_dagrun for example_external_task_marker_parent to None
INFO [airflow.models.dag] Setting next_dagrun for example_nested_branch_dag to 2024-09-19 00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_passing_params_via_test_command to 2024-09-20T22:48:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_python_operator to None
INFO [airflow.models.dag] Setting next_dagrun for example_short_circuit_operator to 2024-09-19T22:49:44.976375+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_skip_dag to 2024-09-19T22:49:44.976638+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_sla_dag to 2024-09-20T22:46:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_subdag_operator to 2024-09-18T00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-1 to None
INFO [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-2 to None
INFO [airflow.models.dag] Setting next_dagrun for example_task_group to 2024-09-19T22:49:44.980027+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_task_group_decorator to 2024-09-19T22:49:44.980350+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_time_delta_sensor_async to None
INFO [airflow.models.dag] Setting next_dagrun for example_trigger_controller_dag to 2021-01-01T00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_trigger_target_dag to None
INFO [airflow.models.dag] Setting next_dagrun for example_weekday_branch_operator to 2024-09-19 00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_xcom to 2021-01-01T00:00:00+00:00
INFO [airflow.models.dag] Setting next_dagrun for example_xcom_args to None
INFO [airflow.models.dag] Setting next_dagrun for example_xcom_args_with_operators to None
INFO [airflow.models.dag] Setting next_dagrun for latest_only to 2024-09-20T18:49:44.983035+00:00
INFO [airflow.models.dag] Setting next_dagrun for latest_only_with_trigger to 2024-09-20T18:49:44.983303+00:00
INFO [airflow.models.dag] Setting next_dagrun for tutorial to 2024-09-19T22:49:44.983513+00:00
INFO [airflow.models.dag] Setting next_dagrun for tutorial_etl_dag to None
INFO [airflow.models.dag] Setting next_dagrun for tutorial_taskflow_api_etl to None
INFO [airflow.models.dag] Sync 2 DAGs
INFO [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-1 to None
INFO [airflow.models.dag] Setting next_dagrun for example_subdag_operator.section-2 to None
Initialization done

```

- Criação do usuário admin:

```
(venv) [opc@solarwindiq-rr-airflow airflow]$ airflow users create \
> --username admin \
> --firstname fiap \
> --lastname solar \
> --role Admin \
> email fiana.santos84@gmail.com \
> --password Flap24
```

```
/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/configuration.py:276: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion(sqlite3.sqlite_version) < StrictVersion(min_sqlite_version):
[2024-09-20 22:53:32,805] {manager.py:779} WARNING - No user yet created, use flask fab command to do it.
[2024-09-20 22:53:32,934] {manager.py:496} INFO - Created Permission View: menu access on List Users
[2024-09-20 22:53:32,940] {manager.py:558} INFO - Added Permission menu access on List Users to role Admin
[2024-09-20 22:53:32,953] {manager.py:496} INFO - Created Permission View: menu access on Security
[2024-09-20 22:53:32,960] {manager.py:558} INFO - Added Permission menu access on Security to role Admin
[2024-09-20 22:53:32,989] {manager.py:496} INFO - Created Permission View: menu access on List Roles
[2024-09-20 22:53:32,996] {manager.py:558} INFO - Added Permission menu access on List Roles to role Admin
[2024-09-20 22:53:33,013] {manager.py:496} INFO - Created Permission View: can read on User Stats Chart
[2024-09-20 22:53:33,020] {manager.py:558} INFO - Added Permission can read on User Stats Chart to role Admin
[2024-09-20 22:53:33,034] {manager.py:496} INFO - Created Permission View: menu access on User's Statistics
[2024-09-20 22:53:33,040] {manager.py:558} INFO - Added Permission menu access on User's Statistics to role Admin
[2024-09-20 22:53:33,070] {manager.py:496} INFO - Created Permission View: menu access on Base Permissions
[2024-09-20 22:53:33,077] {manager.py:558} INFO - Added Permission menu access on Base Permissions to role Admin
[2024-09-20 22:53:33,104] {manager.py:496} INFO - Created Permission View: can read on View Menus
[2024-09-20 22:53:33,110] {manager.py:558} INFO - Added Permission can read on View Menus to role Admin
[2024-09-20 22:53:33,124] {manager.py:496} INFO - Created Permission View: menu access on Views/Menus
[2024-09-20 22:53:33,130] {manager.py:558} INFO - Added Permission menu access on Views/Menus to role Admin
```

- Inicialização do scheduler que cuida da monitoração das DAGs (Directed Acyclic Graphs) e executa as tarefas agendadas no fluxo de trabalho.

DAGs é o termo em inglês que divide o trabalho seja em um ou mais etapas em tarefas (tasks), todas essas tarefas juntas formam uma DAG. No nosso caso, vamos ter uma DAG que pega todas as etapas do fluxo de dados do storage até o carregamento dos dados no Power BI.

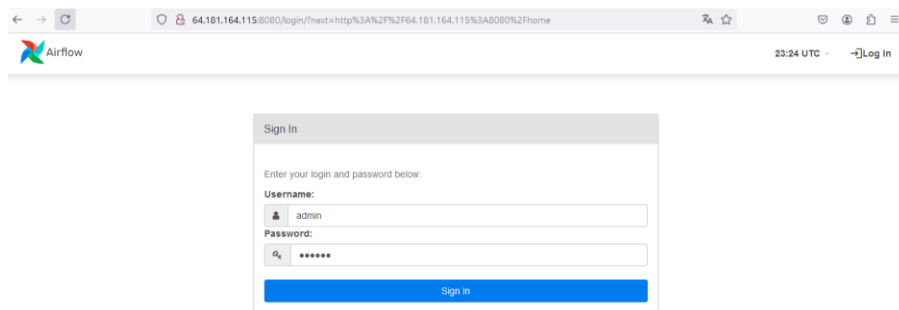
```
/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/configuration.py:276: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion(sqlite3.sqlite_version) < StrictVersion(min_sqlite_version):
[2024-09-20 22:54:24,795] {scheduler_job.py:694} INFO - Starting the scheduler
[2024-09-20 22:54:24,795] {scheduler_job.py:699} INFO - Processing each file at most -1 times
[2024-09-20 22:54:24 +0000] [107417] [INFO] Starting gunicorn 23.0.0
[2024-09-20 22:54:24 +0000] [107417] [INFO] Listening at: http://0.0.0.0:8793 (107417)
[2024-09-20 22:54:24 +0000] [107417] [INFO] Using worker: sync
```

- Também realizamos a mesma na máquina usando o **add port 8080** e depois liberamos a url para login airflow:

```
/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/configuration.py:276: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion(sqlite3.sqlite_version) < StrictVersion(min_sqlite_version):
[2024-09-20 23:20:26,578] {dagbag.py:500} INFO - Filling up the DagBag from /dev/null
[2024-09-20 23:20:26,605] {manager.py:779} WARNING - No user yet created, use flask fab command to do it.
[2024-09-20 23:20:26,848] {manager.py:512} WARNING - Refused to delete permission view, assoc with role exists DAG Runs.can_create Admin
Traceback (most recent call last):
  File "/home/opc/airflow/venv/bin/airflow", line 8, in <module>
    sys.exit(main())
  File "/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/_main_.py", line 48, in main
    args.func(args)
  File "/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/cli/cli_parser.py", line 48, in command
    return func(*args, **kwargs)
  File "/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/commands/webserver.py", line 368, in webserver
    check_if_pidfile_process_is_running(pid_file=pid_file, process_name="webserver")
  File "/home/opc/airflow/venv/lib64/python3.8/site-packages/airflow/commands/process_utils.py", line 297, in check_if_pidfile_process_is_running
    raise AirflowException(f"The {process_name} is already running under PID {pid}.")
airflow.exceptions.AirflowException: The webserver is already running under PID 108590.
```



- Realização do login:



- Edição do arquivo py, para aparecer na DAG da interface gráfica do airflow:

```
[opc@solarwindiq-rr-airflow ~]$ nano airflow/dags/solarwindiq.py
[opc@solarwindiq-rr-airflow ~]$ ll airflow/dags/
total 16
-rw-rw-r--. 1 opc airflow 4600 Nov 18 09:15 solarwindiq.py
```

- Exemplo do script python da DAG:

```
# solarwindiq.py

from airflow import DAG
from airflow.providers.microsoft.azure.operators.data_factory import AzureDataFactoryOperator
from airflow.providers.databricks.operators.databricks import DatabricksRunNowOperator
from airflow.providers.jupyter.operators.jupyter import JupyterOperator
from airflow.providers.microsoft.powerbi.operators.powerbi import PowerBIDatasetPushOperator
from datetime import datetime

# Definir as variáveis do projeto
DATA_FACTORY_SOLAR = 'ingest_data_pipeline'
DATABRICKS_SOLAR = 'process_data_job'
JUPYTER_NOTEBOOK_PATH = '/path/to/jupyter_notebook.ipynb'
DATABRICKS_SOLAR = 'final_energy_predictions'
PBI_DATASET_ID = 'your_pbi_dataset_id'
PBI_GROUP_ID = 'your_pbi_group_id'

# Configurar o DAG do Airflow
default_args = {
    'owner': 'airflow',
    'start_date': datetime(2024, 11, 20),
    'retries': 1,
}

dag = DAG(
    'solarwindiq_dag',
    default_args=default_args,
    description='Ingestão de dados para o projeto SolarWindIQ',
    schedule_interval=None,
)

# 1. Ingestão de dados do Azure Data Factory
ingest_data = AzureDataFactoryOperator(
    task_id='ingest_data_from_factory',
    pipeline_name=DATA_FACTORY_SOLAR,
    azure_data_factory_conn_id='azure_data_factory_conn',
    dag=dag,
)

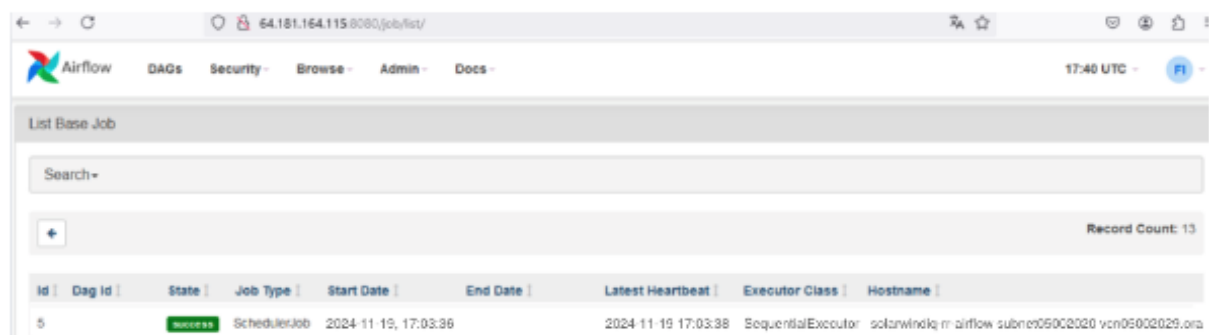
# 2. Processamento de dados no Databricks
process_data_databricks = DatabricksRunNowOperator(
    task_id='process_data_in_databricks',
    job_name=DATABRICKS_SOLAR,
    databricks_conn_id='databricks_default',
    notebook_params={'input_data': 'input_data', 'output_table': DATABRICKS_SOLAR},
    dag=dag,
)
```

```
# 4. Processamento no Jupyter e retorno para Databricks
process_data_in_jupyter = JupyterOperator(
    task_id='process_data_in_jupyter',
    notebook_path=JUPYTER_NOTEBOOK_PATH,
    jupyter_conn_id='jupyter_conn', # Certifique-se de configurar sua conexão com o Jupyter
    dag=dag,
)

# 5. Carregar dados para Power BI
load_to_pbi = PowerBIDatasetPushOperator(
    task_id='load_data_to_pbi',
    dataset_id=PBI_DATASET_ID,
    group_id=PBI_GROUP_ID,
    data_source='databricks',
    dag=dag,
)

# Definindo a ordem das execuções das tarefas
ingest_data >> process_data_databricks >> extract_data_from_databricks >> process_data_in_jupyter >> load_to_pbi
```

- Confirmação do fluxo cadastrado no airflow para orquestração:



id	Dag id	State	Job Type	Start Date	End Date	Latest Heartbeat	Executor Class	Hostname
5		success	SchedulerJob	2024-11-19, 17:03:36		2024-11-19 17:03:38	SequentialExecutor	solarwindiq-m-airflow-subnet05002020-vcn05002020-ora

Com isso, a orquestração precisa do Airflow do SolarWindIQ demonstra ser uma ótima ferramenta para a otimização dos processos e a redução de custos. Acreditamos que ao utilizar as instâncias do Azure no módulo Free Tier, conseguimos maximizar a utilização dos recursos computacionais disponíveis, minimizando gastos com infraestrutura quando falamos de arquitetura robusta nos fluxos de trabalho.

## 8. Cálculo de Custos, ROI e Benefícios da Solução

Nossa proposta utiliza ferramentas gratuitas e de baixo custo para maximizar a eficiência da arquitetura SolarWindIQ, com o foco principal em máquinas virtuais da Azure, Data Factory e Blob Storage como elementos pagos. Este modelo balanceia o uso de tecnologias open-source e soluções de mercado para alcançar alta performance com investimento otimizado.

### Custos Estimados

#### 1. Azure Data Factory

- Pipeline de ingestão de dados eficiente e escalável.
- Custo anual: R\$ 20 mil.

## 2. Azure Blob Storage

- Armazenamento seguro e flexível para dados brutos, tratados e modelos.
- Custo anual: R\$ 35 mil.

## 3. Máquinas Virtuais Azure

- Servidores sob demanda para processar e treinar modelos, evitando GPUs para reduzir custos.
- Custo anual: R\$ 20 mil.

**Total Anual Estimado:** R\$ 75 mil, podendo variar dependendo caso o cliente queira personalizar o modelo

## Ferramentas Open Source

- Databricks Community Edition: Processamento de dados e treinamento de modelos sem custo adicional.
- Airflow: Orquestração dos pipelines de dados.
- Oracle Data Modeler: Modelagem de dados para documentação e otimização de estruturas.
- Power BI Free: Visualização de dados para dashboards simples e gerenciais.
- Jupyter Notebooks: Treinamento e análise de modelos de machine learning.

## Benefícios Financeiros e Operacionais

A solução reduz custos energéticos e promove sustentabilidade:

Economia Anual Estimada: R\$ 120 mil, considerando redução de 12% em contas de energia de empresas com gasto de R\$ 1 milhão/ano.

- ROI:

$$\text{ROI} = 75.000(120.000 - 75.000) \times 100 = 60\%$$

- Payback:

$$\text{Payback} = 120.000 / 75.000 = 7,5 \text{ meses}$$

A solução SolarWindIQ, que se baseia principalmente em ferramentas open-source como Databricks Community, Airflow e Power BI Free, enquanto utiliza Azure apenas para armazenamento e ingestão de dados (*com ferramentas como Blob Storage e Data Factory*), estimamos uma redução de custos de até **60% a 80%** quando comparado às soluções proprietárias mais custosas.

Com base em algumas pesquisas em relação a concorrência:

- **Custo Anual de Soluções (IBM/Grid Edge):** R\$250.000 a R\$400.000, pois dependendo tanto a IBM quando a Grid, ambas são em dólar.
- **Custo Anual da Solução SolarWindIQ:** R\$75.000, considerando Azure e serviços necessários para os modelos, sendo em moeda local brasileira.

Essa diferença de custo pode resultar em uma economia anual de até **R\$175.000 a R\$325.000**, o que, em termos de ROI, representaria uma recuperação do investimento inicial em um período mais curto.

Por isso, a arquitetura SolarWindIQ oferece um modelo sustentável e econômico ao utilizar tecnologias gratuitas em conjunto com ferramentas da Azure. O investimento inicial é recuperado em menos de oito meses, consolidando uma solução robusta, escalável e alinhada às demandas do setor de energia renovável.

## 9. Links para Solução/ Protótipo

Link Vídeo Pitch: [https://youtu.be/9KHHqsRdQ\\_4?si=MbhJ3JwFos\\_KgDBO](https://youtu.be/9KHHqsRdQ_4?si=MbhJ3JwFos_KgDBO)

Link Vídeo Técnico:

[https://www.canva.com/design/DAGXPvcGCbc/cTcQkHKBrIZmue5qarIWlw/edit?utm\\_content=DAGXPvcGCbc&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGXPvcGCbc/cTcQkHKBrIZmue5qarIWlw/edit?utm_content=DAGXPvcGCbc&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

Link Apresentação:

[https://www.canva.com/design/DAGXL1-tkIY/hRrOiR683fS\\_HtIMWtZQcA/view?utm\\_content=DAGXL1-tkIY&utm\\_campaign=designshare&utm\\_medium=link&utm\\_source=editor](https://www.canva.com/design/DAGXL1-tkIY/hRrOiR683fS_HtIMWtZQcA/view?utm_content=DAGXL1-tkIY&utm_campaign=designshare&utm_medium=link&utm_source=editor)

Link Power BI:

<https://app.powerbi.com/view?r=eyJrIjoiazDBmMzM4MzctMzkwZS00Ym11LTliNiMtZDYyZDM1NDY5OGZhliwidCI6IjExZGJiZmUyLTg5YjgtNDU0OS1iZTEwLWNiYzM2NGU1OTU1MSIsImMiOiR9>

Link Website/Protótipo:

<https://seahub.my.canva.site/solarwindiq>

## 10. Conclusão

O projeto SolarWindIQ apresenta uma abordagem inovadora para enfrentar os desafios da transição energética sustentável no Brasil. A combinação de tecnologias avançadas como Machine Learning, Deep Learning e visualizações interativas proporciona uma solução robusta para prever e otimizar a geração de energia renovável. A arquitetura proposta, fundamentada em práticas modernas de tratamento e armazenamento de dados, oferece precisão e eficiência ao mesmo tempo que reduz custos operacionais.

Ao integrar dados meteorológicos e históricos de produção com modelos preditivos, o SolarWindIQ não só fomenta o desenvolvimento econômico local, mas também promove um impacto ambiental positivo, reduzindo a dependência de fontes poluentes. Através de uma interface intuitiva e acessível, a solução facilita a tomada de decisão estratégica para empresas, governos e outros stakeholders, possibilitando uma matriz energética mais limpa e eficiente.

Além de sua eficiência tecnológica e viabilidade econômica, o SolarWindIQ destaca-se pelo impacto positivo na questão ambiental e sustentável. Ao fomentar a utilização de energias limpas, como solar e eólica, a plataforma contribui diretamente para a redução das emissões de gases de efeito estufa, minimizando os impactos das mudanças climáticas e promovendo a transição para uma matriz energética mais sustentável. Sua abordagem favorece a substituição de fontes poluentes, como termelétricas a carvão, por alternativas renováveis, alinhando-se às metas globais de preservação ambiental e sustentabilidade.

Com custos operacionais reduzidos e alto retorno sobre o investimento, o SolarWindIQ reafirma seu compromisso com a inovação tecnológica e a responsabilidade ambiental, consolidando-se como uma ferramenta essencial para a promoção de energia renovável no Brasil. Ao conectar avanços tecnológicos com práticas sustentáveis, o projeto impulsiona o desenvolvimento econômico local, fortalece a segurança energética e contribui de maneira significativa para um futuro mais equilibrado, resiliente e alinhado aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU.

## Referências

<https://exame.com/esferabrasil/brasil-tera-demanda-crescente-de-energia-ate-2026-diz-relatorio-da-iea/>

<https://g1.globo.com/pr/parana/especial-publicitario/cogecom/energia-sustentavel-e-mais-barata-para-todos/noticia/2024/09/30/quais-sao-os-paises-que-mais-investem-em-energia-limpa.ghtml>

<https://www.itatiaia.com.br/politica/2024/11/16/na-cop-fiemg-defende-a-substituicao-de-100-das-termeletricas-de-carvao-ate-2035>

<https://www.nationalgeographicbrasil.com/meio-ambiente/2024/04/os-6-fatos-sobre-as-energias-renovaveis-elas-sao-fundamentais-contras-mudancas-climaticas>

<https://g1.globo.com/economia/noticia/2024/02/02/energia-limpa-brasil-bate-recorde-com-93percent-da-energia-gerada-em-2023-vindo-de-fontes-renovaveis-diz-ccee.ghtml>

<https://g1.globo.com/rn/rio-grande-do-norte/noticia/2024/08/26/rn-atinge-marca-de-10-gw-de-capacidade-de-geracao-de-energia-eolica.ghtml>

<https://www.alemdaenergia.engie.com.br/rio-grande-do-norte-avanca-na-producao-de-energia-eolica/>

<https://www.rn.gov.br/materia/governo-do-rn-aposta-em-parcerias-com-empresas-eolicas/>

<https://petronoticias.com.br/o-grupo-ccr-faz-acordo-com-a-neoenergia-e-assume-sociedade-em-tres-complexos-eolicos-no-piaui/>

<https://petronoticias.com.br/statkraft-inicia-operacao-comercial-de-parque-eolico-na-bahia/>

<https://canalsolar.com.br/empresas-de-energia-solar-no-brasil-devem-estar-preparadas-para-2024/>

<https://origoenergia.com.br/blog/energia/energia-solar-no-brasil>

<https://www.fiern.org.br/rn-ganha-13-novos-parques-eolicos-nos-primeiros-dois-meses-de-2024-mostra-mais-rn/>

<https://www.ccee.org.br/precos/painel-precos>