2TSCF

Cloud Solutions
Global Solutions

Solar Grid Connect

Fabio Pereira Lima	RM98803
Giullia Bianca Rocha Souza	RM552108
Renan M Carrara Coimbra	RM552187

Criação do Grupo de Recursos no Azure CLI

az group create --name SolarGridResourceGroup --location brazilSouth

```
Solicitando um Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
azureuser [ ~ ]$ az group create --name SolarGridResourceGroup --location brazilSouth
{
  "id": "/subscriptions/de8235b7-d2dd-4614-82ee-9b0b2f6a122a/resourceGroups/SolarGridResourceGroup",
  "location": "brazilsouth",
  "managedBy": null,
  "name": "SolarGridResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
azureuser [ ~ ]$
```

O comando cria um grupo de recursos chamado SolarGridResourceGroup na região brazilSouth

Conta de Armazenamento

az storage account create --name solargridstore --resource-group SolarGridResourceGroup --location brazilSouth --sku Standard_LRS

```
azureuser [ ~ ]$ az storage account create --name solargridstore --resource-group SolarGridResourceGroup --location brazilSouth --sku Standard_LRS
{
  "accessTier": "Hot",
  "accountMigrationInProgress": null,
  "allowBlobPublicAccess": false,
  "allowCrossTenantReplication": false,
  "allowSharedKeyAccess": null,
  "allowedCopyScope": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2024-11-19T15:43:53.544734+00:00",
  "customDomain": null,
  "defaultToOAuthAuthentication": null,
  "dnsEndpointType": null,
  "enableExtendedGroups": null,
  "enableHttpsTrafficOnly": true,
  "enableNfsV3": null,
  "encryption": {
    "encryptionIdentity": null,
    "keySource": "Microsoft.Storage",
    "keyVaultProperties": null,
    "requireInfrastructureEncryption": null,
    "services": {
      "blob": {
```

Uma conta de armazenamento foi criada dentro do grupo de recursos. O sku Standard_LRS

Inserindo container na conta de armazenamento

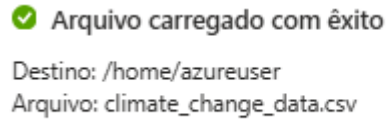
az storage container create --account-name solargridstore --name datalake

```
azureuser [ ~ ]$ az storage container create --account-name solargridstore --name datalake

There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is assigned required RBAC roles.
For more information about RBAC roles in storage, visit https://docs.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more information about environment variable usage.
```

Um container foi criado dentro da conta de armazenamento

* alerta que não foi fornecida uma credencial explícita para autenticação. Embora o comando tenha funcionado

Upload do Dataset no Container



az storage blob upload --account-name solargridstore --container-name datalake --file /home/azureuser/climate_change_data.csv --name climate_change_data.csv

```
azureuser [ ~ ]$ az storage blob upload --account-name solargridstore --container-name datalake --file /home/azureuser/climate_change_data.csv --name climate_change_data.csv

There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is assigned required RBAC roles.
For more information about RBAC roles in storage, visit https://docs.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more information about environment variable usage.
Finished[#############################################################]  100.0000%
{
  "client_request_id": "c1d6c930-a68f-11ef-b3f6-00155d40c69c",
  "content_md5": "NAKqtRS5sotlhhFPh799AA==",
  "date": "2024-11-19T16:03:03+00:00",
  "encryption_key_sha256": null,
  "encryption_scope": null,
  "etag": "\"0x8DD08B3A6CB69F1\"",
  "lastModified": "2024-11-19T16:03:04+00:00",
  "request_id": "a59dcf3a-801e-005c-7a9c-3abbd6000000",
  "request_server_encrypted": true,
  "version": "2022-11-02",
  "version_id": null
}
```

Name	Blob Type	Blob Tier	Length	Content Type	Last Modified	Snapshot
climate_change_data.csv	BlockBlob	Hot	1669507	text/csv	2024-11-19T16:03:04+00:00	

Fazendo o upload do csv para dentro do container.

Criando CosmoDB

az cosmosdb create --name solargridcosmosdb --resource-group SolarGridResourceGroup --locations regionName=brazilSouth failoverPriority=0 isZoneRedundant=false

```
zureuser [ ~ ]$ az cosmosdb create --name solargridcosmosdb --resource-group SolarGridResourceGroup --locations regionName=brazilSouth failoverPriority=0 isZoneRedundant=false
{
  "analyticalStorageConfiguration": {
    "schemaType": "WellDefined"
  },
  "apiProperties": null,
  "backupPolicy": {
    "migrationState": null,
    "periodicModeProperties": {
      "backupIntervalInMinutes": 240,
      "backupRetentionIntervalInHours": 8,
      "backupStorageRedundancy": "Geo"
    },
    "type": "Periodic"
  },
  "capabilities": [],
  "capacity": null,
  "connectorOffer": null,
  "consistencyPolicy": {
    "defaultConsistencyLevel": "Session",
    "maxIntervalInSeconds": 5,
    "maxStalenessPrefix": 100
  },
  "cors": [],
  "createMode": null,
  "customerManagedKeyStatus": null,
  "databaseAccountOfferType": "Standard",
  "defaultIdentity": "FirstPartyIdentity",
  "disableKeyBasedMetadataWriteAccess": false,
```

Esse comando cria o CosmoDB, que será usado para armazenar os dados processados.

Criando banco de dados

az cosmosdb sql database create --account-name solargridcosmosdb --resource-group SolarGridResourceGroup --name solargriddb

```
azureuser [ ~ ]$ az cosmosdb sql database create --account-name solargridcosmosdb --resource-group SolarGridResourceGroup --name solargriddb
{
  "id": "/subscriptions/de8235b7-d2dd-4614-82ee-9b0b2f6a122a/resourceGroups/SolarGridResourceGroup/providers/Microsoft.DocumentDB/databaseAccounts/solargridcosmosdb/sqlDatabases/solargriddb",
  "location": null,
  "name": "solargriddb",
  "options": null,
  "resource": {
    "_self": "dbs/iRQkAA==/",
    "colls": "colls/",
    "createMode": null,
    "etag": "\"00001e04-0000-0b00-0000-673cbccc0000\"",
    "id": "solargriddb",
    "restoreParameters": null,
    "rid": "iRQkAA==",
    "ts": 1732033740.0,
    "users": "users/"
  },
  "resourceGroup": "SolarGridResourceGroup",
  "tags": null,
  "type": "Microsoft.DocumentDB/databaseAccounts/sqlDatabases"
}
```

Esse comando cria o banco de dados sql no CosmoDB

Container no CosmoDB

az cosmosdb sql container create --account-name solargridcosmosdb --resource-group SolarGridResourceGroup --database-name solargriddb --name ProcessedClimateData --partition-key-path "/id"

```
azureuser [ ~ ]$ az cosmosdb sql container create --account-name solargridcosmosdb --resource-group SolarGridResourceGroup --database-name solargriddb --name ProcessedClimateData --partition-
ey-path "/id"
{
  "id": "/subscriptions/de8235b7-d2dd-4614-82ee-9b0b2f6a122a/resourceGroups/SolarGridResourceGroup/providers/Microsoft.DocumentDB/databaseAccounts/solargridcosmosdb/sqlDatabases/solargriddb/c
ntainers/ProcessedClimateData",
  "location": null,
  "name": "ProcessedClimateData",
  "options": null,
  "resource": {
    "_conflicts": "conflicts/",
    "_docs": "docs/",
    "_self": "dbs/iRQkAA==/colls/iRQkAIxKLX0=/",
    "_sprocs": "sprocs/",
    "_triggers": "triggers/",
    "_udfs": "udfs/",
    "analyticalStorageTtl": null,
    "clientEncryptionPolicy": null,
    "computedProperties": [],
    "conflictResolutionPolicy": {
      "conflictResolutionPath": "/_ts",
      "conflictResolutionProcedure": "",
      "mode": "LastWriterWins"
    },
    "createMode": null,
    "defaultTtl": null,
    "etag": "\"00002004-0000-0b00-0000-673cbd8e0000\"",
    "geospatialConfig": {
      "type": "Geography"
    },
    "id": "ProcessedClimateData",
```

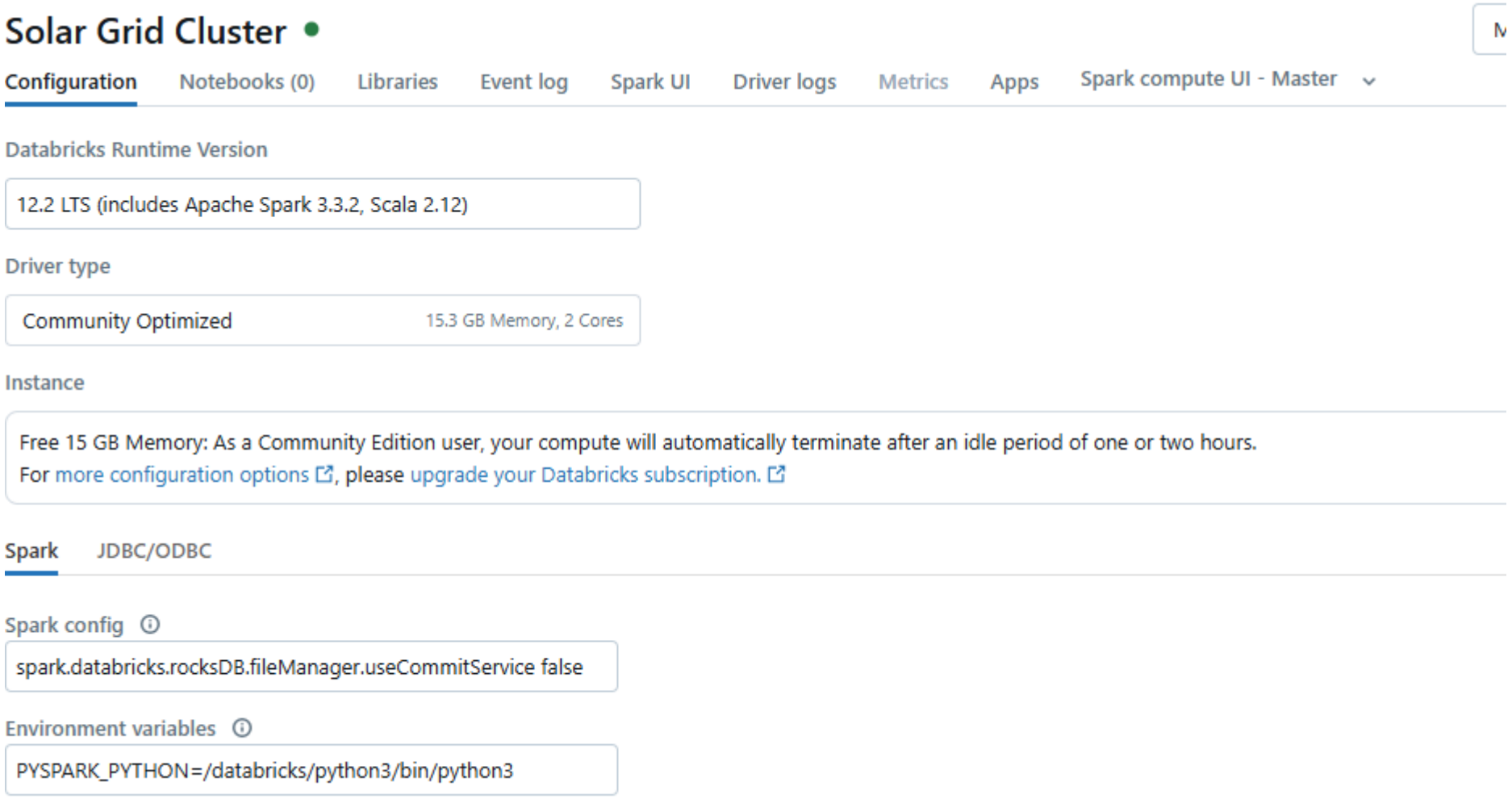
Esse comando cria o container dentro do CosmoDB

az cosmosdb sql container list --account-name solargridcosmosdb --resource-group SolarGridResourceGroup --database-name solargriddb --output table

```
}
azureuser [ ~ ]$ az cosmosdb sql container list --account-name solargridcosmosdb --resource-group SolarGridResourceGroup --database-name solargriddb --output table
Name                ResourceGroup
------------------  --------------------
ProcessedClimateData  SolarGridResourceGroup
```

Esse comando lista os container

Criando Cluster no databricks



Explorando o Dataset pelo Notebook no Databricks

+ Code + Text

02:24 PM (15s) 1

```
# Configuração do Blob Storage
storage_account_name = "solargridstore"
storage_account_key = "HCnpPyD+AOEzgvov3b09CQ4qg8JrnJS47SrmchsfjlWS3lScal5so7QOtffGBTv5AEbgyfyqFsoO+AStHzjKoQ=="
container_name = "datalake"
mount_point = "/mnt/solargridblob"

# Montar o Blob Storage
dbutils.fs.mount(
    source = f"wasbs://{container_name}@{storage_account_name}.blob.core.windows.net",
    mount_point = mount_point,
    extra_configs = {f"fs.azure.account.key.{storage_account_name}.blob.core.windows.net": storage_account_key}
)
```

```
Out[1]: True
```

Just now (12s) 2 Python

```
# Listar os arquivos no contêiner
display(dbutils.fs.ls("/mnt/solargridblob"))
```

(3) Spark Jobs

Table +

	path	name	size	modificationTime
1	dbfs:/mnt/solargridblob/climate_change_data.c...	climate_change_data.c...	1669507	1732032184000

File Edit View Run Help Last edit was 28 minutes ago

02:39 PM (16s) 2 Python

```python
# Carregar os dados
df = spark.read.csv("/mnt/solargridblob/climate_change_data.csv", header=True,
inferSchema=True)

# Mostrar as primeiras linhas
df.show(5)

# Exibir o esquema
df.printSchema()

# Estatísticas descritivas
df.describe().show()
```

(5) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [Date: timestamp, Location: string ... 7 more fields]

```
 |-- Wind Speed: double (nullable = true)

+-------+-----------+-----------+------------------+-----------------+----------------
----+--------------------+------------------+--------------------+
|summary|   Location|    Country|       Temperature|    CO2 Emissions|      Sea Level
Rise|       Precipitation|          Humidity|          Wind Speed|
+-------+-----------+-----------+------------------+-----------------+----------------
----+--------------------+------------------+--------------------+
|  count|      10000|      10000|             10000|            10000|               1
0000|               10000|             10000|               10000|
|   mean|       null|       null| 14.93603377863015|400.2204688976376|-0.0031516854837
4...|   49.88120758239161|49.771301928591484|  25.082065534372408|
| stddev|       null|       null| 5.030615849200168| 49.6969327066168|  0.991348668374
0705|  28.862417088869915| 28.92932002726051|    14.4666480385455|
|    min|  Aaronberg|Afghanistan|-3.803588598564197|182.1312203487651|  -4.09215497048
3335|0.010142670585777669|0.0189976288834659|0.001732327457104...|
|    max|Zunigaburgh|   Zimbabwe| 33.97695597775728|582.8997013258188|   4.11655898365
9821|   99.99190025064706| 99.95966453034534|  49.997663994105636|
+-------+-----------+-----------+------------------+-----------------+----------------
----+--------------------+------------------+--------------------+
```

File Edit View Run Help Last edit was 30 minutes ago

02:16 PM (<1s) 3

```
df.printSchema()
```

```
root
 |-- Date: timestamp (nullable = true)
 |-- Location: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- Temperature: double (nullable = true)
 |-- CO2 Emissions: double (nullable = true)
 |-- Sea Level Rise: double (nullable = true)
 |-- Precipitation: double (nullable = true)
 |-- Humidity: double (nullable = true)
 |-- Wind Speed: double (nullable = true)
```

02:16 PM (7s) 4

```
df.describe().show()
```

(2) Spark Jobs

```
+-------+-----------+-----------+------------------+-----------------+--------------------+-----------------
---+------------------+--------------------+
|summary|   Location|    Country|       Temperature|    CO2 Emissions|      Sea Level Rise|       Precipitat
ion|          Humidity|          Wind Speed|
+-------+-----------+-----------+------------------+-----------------+--------------------+-----------------
---+------------------+--------------------+
|  count|      10000|      10000|             10000|            10000|               10000|               10
000|             10000|               10000|
|   mean|       null|       null| 14.93603377863015|400.2204688976376|-0.00315168548374...|   49.88120758239
161|49.771301928591484|  25.082065534372408|
| stddev|       null|       null| 5.030615849200168| 49.6969327066168|  0.9913486683740705|  28.862417088869
915| 28.92932002726051|   14.4666480385455|
|    min|  Aaronberg|Afghanistan|-3.803588598564197|182.1312203487651|  -4.092154970483335|0.010142670585777
669|0.0189976288834659|0.001732327457104...|
|    max|Zunigaburgh|   Zimbabwe| 33.97695597775728|582.8997013258188|   4.116558983659821|   99.99190025064
706| 99.95966453034534|  49.997663994105636|
+-------+-----------+-----------+------------------+-----------------+--------------------+-----------------
---+------------------+--------------------+
```

Cálculo de Correlações

GS_solar_grid Python

File Edit View Run Help Last edit was 31 minutes ago

Interrupt Gs

02:40 PM (5s) 5

```
# Converter para Pandas para calcular correlações
df_pandas = df.toPandas()
correlation_matrix = df_pandas.corr()

print(correlation_matrix)
```

(1) Spark Jobs

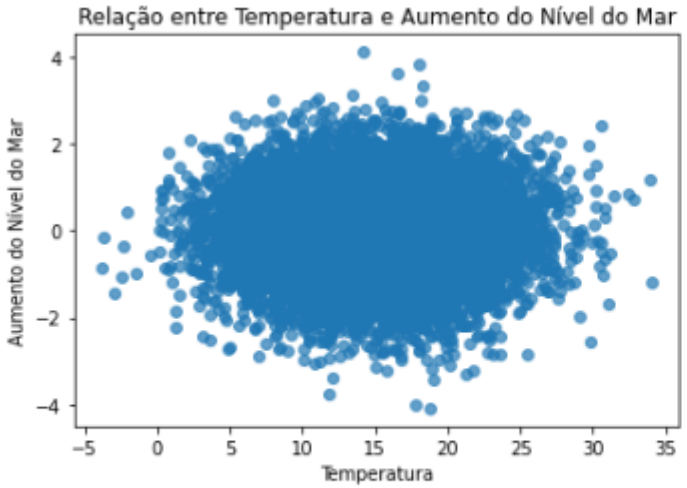
```
                Temperature  CO2 Emissions  Sea Level Rise  Precipitation  \
Temperature        1.000000      -0.002775        0.011663       0.004916
CO2 Emissions     -0.002775       1.000000       -0.004751      -0.007443
Sea Level Rise     0.011663      -0.004751        1.000000      -0.000249
Precipitation      0.004916      -0.007443       -0.000249       1.000000
Humidity          -0.015737      -0.003661       -0.007610       0.003732
Wind Speed         0.021779      -0.003990        0.011789      -0.018798

                Humidity  Wind Speed
Temperature    -0.015737    0.021779
CO2 Emissions  -0.003661   -0.003990
Sea Level Rise -0.007610    0.011789
Precipitation   0.003732   -0.018798
Humidity        1.000000    0.028591
Wind Speed      0.028591    1.000000
```

Gráfico de dispersão

Relação entre Temperatura e Aumento do Nível do Mar

02:41 PM (<1s) 6

```
import matplotlib.pyplot as plt

# Gráfico: Temperatura x Aumento do nível do mar
plt.scatter(df_pandas["Temperature"], df_pandas["Sea Level Rise"], alpha=0.7)
plt.title("Relação entre Temperatura e Aumento do Nível do Mar")
plt.xlabel("Temperatura")
plt.ylabel("Aumento do Nível do Mar")
plt.show()
```

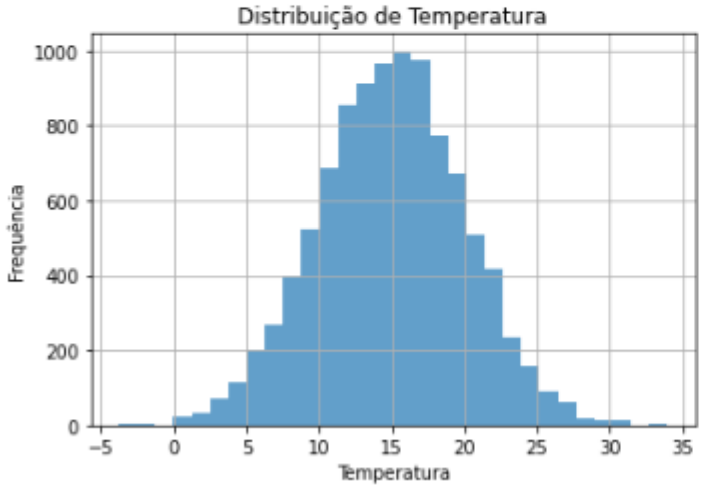


O gráfico representa a relação entre temperatura e aumento do nível do mar, com uma dispersão significativa dos dados. Não há uma correlação visual evidente, sugerindo que fatores adicionais ou ruídos podem estar influenciando os resultados. A amplitude dos valores é ampla (-5 a 35 graus para temperatura e -4 a 4 unidades para o nível do mar).

Histograma mostrando distribuição de Frequências das Temperaturas

02:48 PM (<1s) 7

```
df_pandas["Temperature"].hist(bins=30, alpha=0.7)
plt.title("Distribuição de Temperatura")
plt.xlabel("Temperatura")
plt.ylabel("Frequência")
plt.show()
```



Esse histograma apresenta a distribuição das temperaturas no conjunto de dados. A forma do gráfico indica uma distribuição aproximadamente normal, com maior concentração de temperaturas entre 10 e 20 graus. Isso sugere que essas temperaturas são mais frequentes nos dados analisados. No entanto, extremos fora dessa faixa (abaixo de 5 e acima de 30 graus) também são observados, mas ocorrem com menor frequência.

Exibindo a média do aumento do nível do mar agrupada por ano usando PySpark.

02:43 PM (6s) 8

```
from pyspark.sql.functions import year

# Adicionar uma coluna de ano
df = df.withColumn("Year", year("Date"))

# Agrupar por ano e calcular a média
df.groupBy("Year").avg("Sea Level Rise").show()
```

(2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [Date: timestamp, Location: string ... 8 more fields]

```
|2018| 0.05091619098201069|
|2015|  0.0172197749632445|
|2006| -0.0421065910516791|
|2022|-0.00201331002750...|
|2013|-0.00493550172806...|
|2014| 0.05126234314088981|
|2019| -0.0329682443711795|
|2004| 0.09811426462619739|
|2020|-0.05568564539748...|
|2012|-0.07325312749586939|
|2009|-0.07191149160687238|
|2016|-0.03944828492901061|
|2001|0.021330949511707095|
|2005|-0.02854530109231...|
|2000| 0.06677902371759098|
|2010| 0.04973530879173803|
|2011|-0.00179115121625562|
|2008|-0.13764484625473306|
+----+--------------------+
only showing top 20 rows
```

Regressão Linear com pyspark

02:43 PM (8s) 10

```
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import LinearRegression

# Selecionar features
assembler = VectorAssembler(
    inputCols=["Temperature", "CO2 Emissions", "Precipitation", "Humidity", "Wind Speed"],
    outputCol="features"
)
df_features = assembler.transform(df).select("features", "Sea Level Rise")

# Treinar modelo de regressão
lr = LinearRegression(featuresCol="features", labelCol="Sea Level Rise")
lr_model = lr.fit(df_features)

# Exibir métricas
print(f"Coeficiente: {lr_model.coefficients}")
print(f"Intercepto: {lr_model.intercept}")
print(f"R2: {lr_model.summary.r2}")
```

(2) Spark Jobs

df_features: pyspark.sql.dataframe.DataFrame

```
Coeficiente: [0.0022213218540691783,-9.379371017418392e-05,-3.0720775452711987e-06,-0.000266789487316944 1,0.
0008048798906575228]
Intercepto: -0.005547612166171338
R2: 0.0003515424484459295
```

GS_solar_grid_cloud Python

File Edit View Run Help Last edit was 1 hour ago

Run all Gs Clus

06:21 PM (34m) 13

```python
from azure.cosmos import CosmosClient
import json
import uuid  # Para gerar IDs únicos

# Configuração do Cosmos DB
ENDPOINT = "https://solargridcosmosdb.documents.azure.com:443/"
PRIMARY_KEY = "[redacted]"
DB_NAME = "solargriddb"
COLLECTION_NAME = "ProcessedClimateData"

# Conectar ao Cosmos DB
client = CosmosClient(ENDPOINT, PRIMARY_KEY)
database = client.get_database_client(DB_NAME)
container = database.get_container_client(COLLECTION_NAME)

# Carregar o dataset do Blob Storage
df = spark.read.csv("/mnt/solargridblob/climate_change_data.csv", header=True, inferSchema=True)
df.show(5)

# Converter o DataFrame Spark para Pandas
df_pandas = df.select("Temperature", "CO2 Emissions", "Sea Level Rise", "Precipitation", "Humidity",
"Wind Speed").toPandas()

# Inserir os documentos no Cosmos DB
for _, row in df_pandas.iterrows():
    document = row.to_dict()  # Converte cada linha em um dicionário
    document["id"] = str(uuid.uuid4())  # Adiciona um campo 'id' único
    try:
        container.create_item(body=document)
        print(f"Document inserted: {document}")
    except Exception as e:
        print(f"Error inserting document: {e}")
```

(4) Spark Jobs

Inserido no cosmoDB

06:21 PM (34m) 13 Python

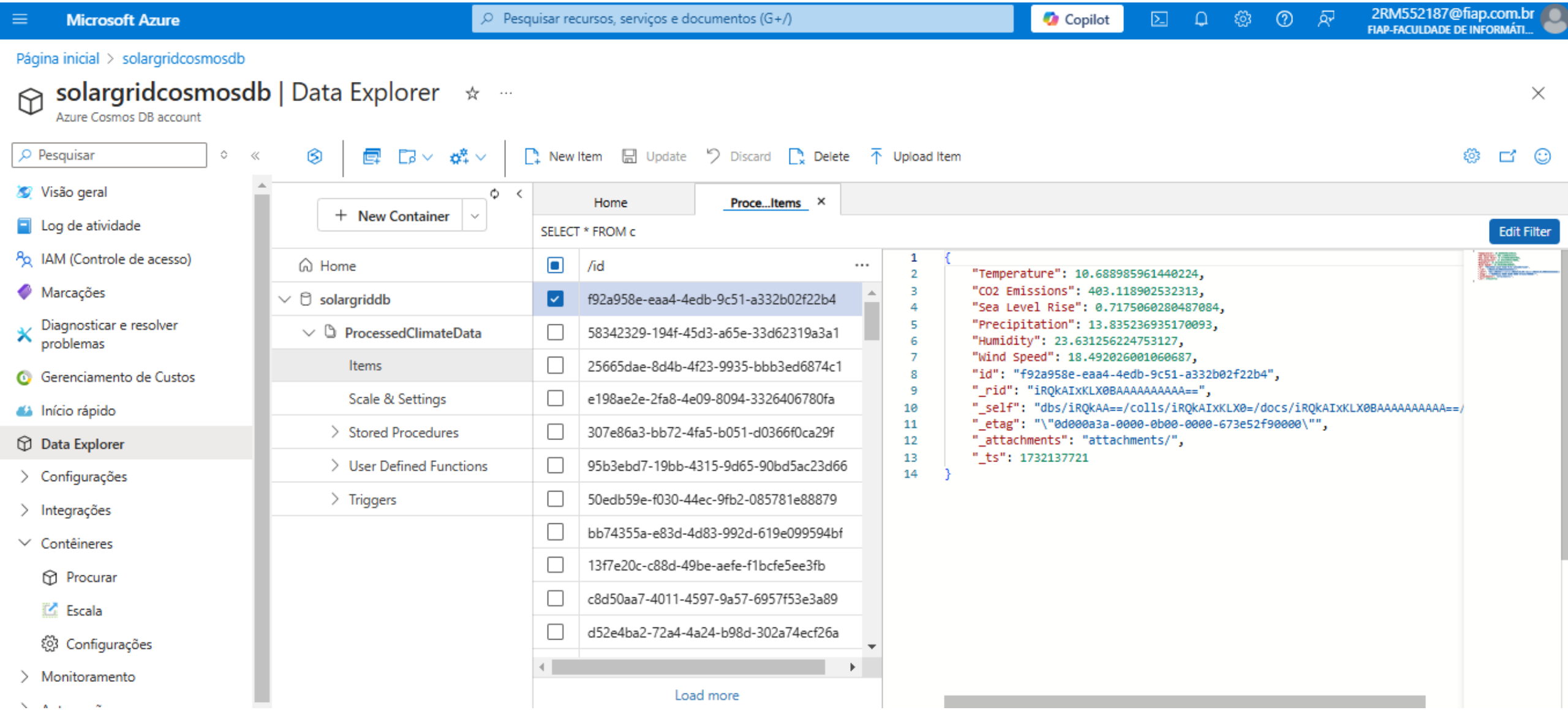
```python
df = spark.read.csv("/mnt/solargridblob/climate_change_data.csv", header=True, inferSchema=True)
df.show(5)

# Converter o DataFrame Spark para Pandas
df_pandas = df.select("Temperature", "CO2 Emissions", "Sea Level Rise", "Precipitation", "Humidity",
"Wind Speed").toPandas()

# Inserir os documentos no Cosmos DB
for _, row in df_pandas.iterrows():
    document = row.to_dict()  # Converte cada linha em um dicionário
    document["id"] = str(uuid.uuid4())  # Adiciona um campo 'id' único
    try:
        container.create_item(body=document)
        print(f"Document inserted: {document}")
    except Exception as e:
        print(f"Error inserting document: {e}")
```

(4) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [Date: timestamp, Location: string ... 7 more fields]

```
Document inserted: {'Temperature': 17.667266824242354, 'CO2 Emissions': 423.68688133089 45, 'Sea Level Ris
e': 1.033963219371248, 'Precipitation': 66.0631857980186, 'Humidity': 35.660745341554936, 'Wind Speed': 1
2.749541759539085, 'id': 'dc20db79-1c79-455a-8359-1ad96e3bbb86'}
Document inserted: {'Temperature': 17.1479361762514, 'CO2 Emissions': 414.4752588945823, 'Sea Level Ris
e': -0.046996247587941714, 'Precipitation': 20.81550365057425, 'Humidity': 39.215361826498274, 'Wind Spee
d': 15.988625895526182, 'id': '701ba28b-c327-48aa-9e24-addd3552b2b3'}
Document inserted: {'Temperature': 15.020522569953128, 'CO2 Emissions': 391.3795370789171, 'Sea Level Ris
e': -1.4522429118024336, 'Precipitation': 93.41710895142636, 'Humidity': 25.29381440842362, 'Wind Speed':
6.531865887798527, 'id': '755854e7-181b-44c2-b9bc-f0bed19078b1'}
Document inserted: {'Temperature': 16.772450667974947, 'CO2 Emissions': 346.9211903903783, 'Sea Level Ris
e': 0.5436163741336263, 'Precipitation': 49.88294706738166, 'Humidity': 96.78740241919833, 'Wind Speed':
42.249013785429206, 'id': 'b8fc032d-3551-4f84-97f7-0de715658691'}
Document inserted: {'Temperature': 22.370024600115745, 'CO2 Emissions': 466.04213566492433, 'Sea Level Ri
se': 1.0267041595219062, 'Precipitation': 30.659841366358766, 'Humidity': 15.211824847884525, 'Wind Spee
d': 18.293707683950718, 'id': '4fd91b0b-8b46-484b-8d3d-e642020542c3'}
Document inserted: {'Temperature': 19.430853325169924, 'CO2 Emissions': 337.89977578684676, 'Sea Level Ri
se': -0.8953287894374228, 'Precipitation': 18.93227471726909, 'Humidity': 82.77452008188295, 'Wind Spee
d': 42.424255047722376, 'id': '45e8b689-a6f8-4e82-91e3-42cfe5756fb8'}
Document inserted: {'Temperature': 12.6619278429786, 'CO2 Emissions': 381.1727464400333, 'Sea Level Ris
e': 2.2607880342329145, 'Precipitation': 78.33965815369248, 'Humidity': 99.24392268965309, 'Wind Speed':
41.85653890509427, 'id': '461a4e53-efac-4872-92d2-d485d0a710bb'}
```

Verificando no portal azure

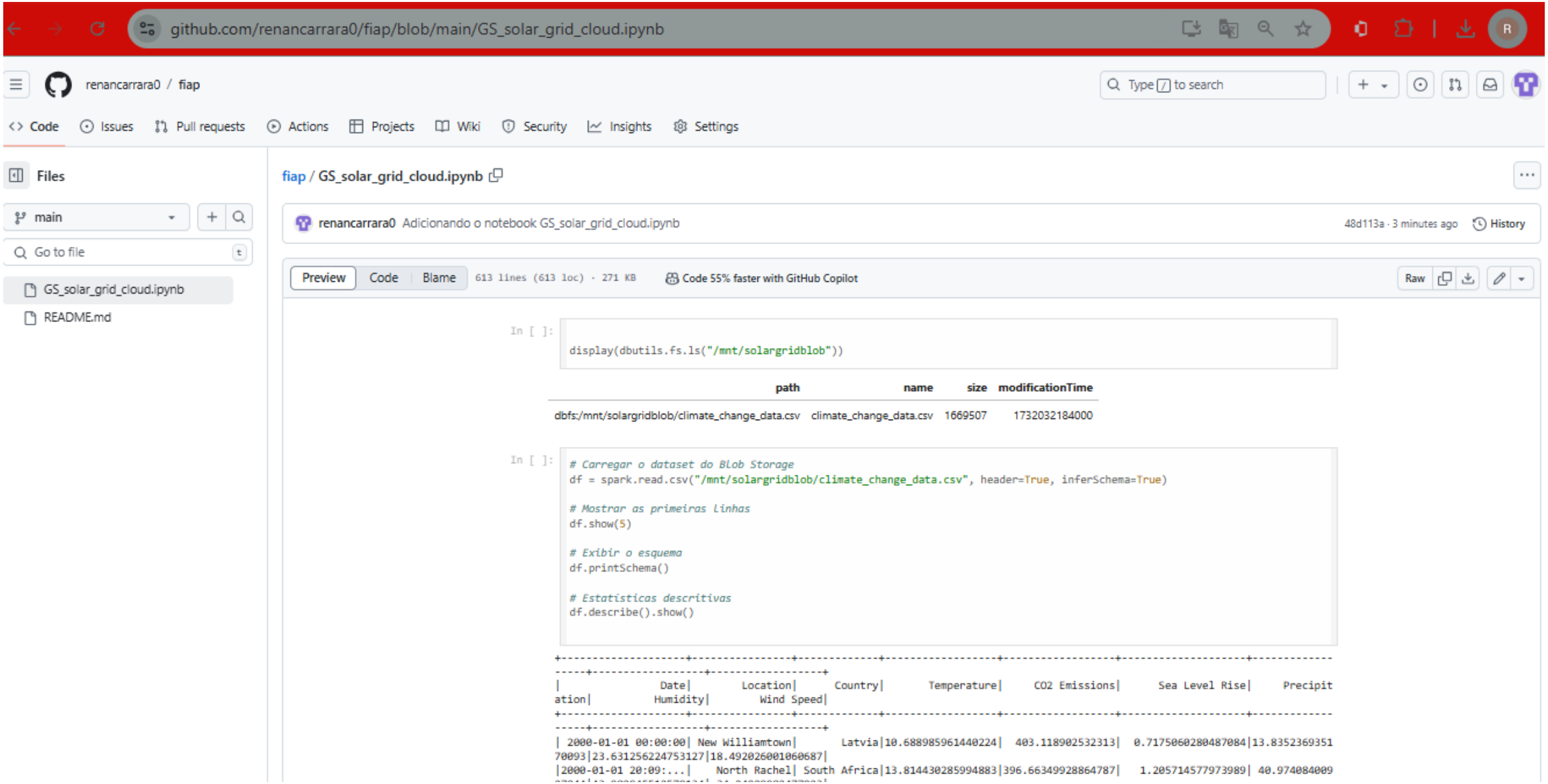


Publicando o notebook no github via powershell

```
PS C:\Windows\System32\fiap\fiap-clean> git add GS_solar_grid_cloud.ipynb
>> git commit -m "Adicionando o notebook GS_solar_grid_cloud.ipynb"
>> git push origin main
>>
warning: in the working copy of 'GS_solar_grid_cloud.ipynb', LF will be replaced by CRLF the next time Git touches it
[main 48d113a] Adicionando o notebook GS_solar_grid_cloud.ipynb
 1 file changed, 613 insertions(+)
 create mode 100644 GS_solar_grid_cloud.ipynb

Exceção Sem Tratamento: System.ComponentModel.Win32Exception: O nome do diretório é inválido
   em System.Diagnostics.Process.StartWithCreateProcess(ProcessStartInfo startInfo)
   em System.Diagnostics.Process.Start()
   em GitCredentialManager.ChildProcess.Start(Trace2ProcessClass processClass)
   em GitCredentialManager.GitProcessConfiguration.Enumerate(GitConfigurationLevel level, GitConfigurationEnumerationCallback cb)
   em GitCredentialManager.GitConfigurationExtensions.Enumerate(IGitConfiguration config, GitConfigurationEnumerationCallback cb)
   em GitCredentialManager.Settings.<GetSettingValues>d__6.MoveNext()
   em System.Linq.Enumerable.FirstOrDefault[TSource](IEnumerable`1 source)
   em GitCredentialManager.Settings.TryGetSetting(String envarName, String section, String property, String& value)
   em GitCredentialManager.Settings.GetTrace2Settings()
   em GitCredentialManager.Trace2.Initialize(DateTimeOffset startTime)
   em GitCredentialManager.Program.AppMain(Object o)
   em System.Threading.ThreadHelper.ThreadStart_Context(Object state)
   em System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext, ContextCallback callback, Object state, Boolean preserveSyncCtx)
   em System.Threading.ExecutionContext.Run(ExecutionContext executionContext, ContextCallback callback, Object state, Boolean preserveSyncCtx)
   em System.Threading.ExecutionContext.Run(ExecutionContext executionContext, ContextCallback callback, Object state)
   em System.Threading.ThreadHelper.ThreadStart(Object obj)
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 164.13 KiB | 6.08 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/renancarrara0/fiap.git
   7a59b27..48d113a  main -> main
PS C:\Windows\System32\fiap\fiap-clean>
```

Conferindo a publicação via browser

https://github.com/renancarrara0/fiap/



Conclusão

O uso de tecnologias em nuvem, como Azure Databricks e CosmosDB, para o processamento eficiente de dados climáticos. Apesar de os resultados obtidos, como a análise da relação entre temperatura e aumento do nível do mar, não indicarem padrões claros, a atividade demonstrou a capacidade de estruturar e processar dados complexos. A aplicação de modelos de regressão linear reforça a necessidade de explorar variáveis adicionais e realizar ajustes no modelo para obter previsões mais precisas. Este trabalho destaca o potencial de ferramentas em nuvem para análises avançadas e o impacto positivo na compreensão de fenômenos climáticos