

Relatório Técnico Interno: Análise do Projeto de Extração de Dados do ERA5

06 de setembro de 2025

1 Resumo do Projeto

O projeto consiste em um pipeline para a automação do download de dados de reanálise climática do **ERA5**, disponibilizados pelo ECMWF através da API do Climate Data Store (CDS). O fluxo de trabalho implementado abrange três etapas principais:

1. **Estudo e Interação com a API do CDS:** O projeto demonstra uma compreensão clara das regras, formatos e limitações da API, como os limites de *fields* por requisição e o tamanho máximo de download, informações cruciais documentadas no `README.md`.
2. **Extração de Dados via Script Python:** Foi desenvolvido o script `cdsAPI.py`, que automatiza a autenticação, definição de parâmetros (variáveis, área, período) e o download dos dados. O formato de saída padrão é o **NetCDF** (`.nc`), ideal para dados geoespaciais multidimensionais.
3. **Pós-processamento com CDO (Climate Data Operators):** A documentação orienta o uso de ferramentas de linha de comando como o CDO para manipulações subsequentes, como o cálculo de médias espaciais (`fldmean`), médias temporais (`daymean`) e conversão de unidades (ex: Kelvin para Celsius com `subc`). Isso demonstra uma abordagem eficiente, separando a extração da transformação de dados.

O objetivo final é fornecer um método robusto e replicável para obter subconjuntos de dados do ERA5, preparando-os para análises.

2 Construção do Código (`cdsAPI.py`)

O script `cdsAPI.py` foi estruturado de forma modular e clara, utilizando funções específicas para cada etapa do processo, o que facilita a manutenção e personalização.

2.1 Inicialização e Autenticação

O script utiliza a biblioteca `python-dotenv` para carregar credenciais (`CDSAPI_URL` e `CDSAPI_KEY`) de um arquivo `.env`. Trazendo boas práticas de segurança, pois evita a exposição de chaves de API no código-fonte. O cliente da API é instanciado com `cdsapi.Client()`, que gerencia a comunicação com o servidor do CDS.

2.2 Estrutura Funcional

1. `get_request_params()` -> `Dict[str, Any]`: Centraliza todos os parâmetros da requisição. Isolar os parâmetros nesta função torna a personalização da consulta extremamente simples.

```
1 def get_request_params() -> Dict[str, Any]:
2
3     return {
4         "product_type": ["reanalysis"],
5         "year": ["2002"],
6         "month": ["03"],
7         "day": ["11", "12", "13", "14", "15"],
8         "time": ["00:00", "01:00", "02:00",
9                 "03:00", "04:00", "05:00",
10                "06:00", "07:00", "08:00",
11                "09:00", "10:00", "11:00",
12                "12:00", "13:00", "14:00",
13                "15:00", "16:00", "17:00",
14                "18:00", "19:00", "20:00",
15                "21:00", "22:00", "23:00"],
16         "data_format": "netcdf",
17         "download_format": "unarchived",
18         "variable": [
19             "soil_temperature_level_1",
20             "soil_temperature_level_2",
21             "lake_bottom_temperature",
22         ],
23         "area": [-19, -48, -23, -43],
24     }
```

2. `generate_output_filename(prefix: str) -> str`: Cria um nome de arquivo de saída único com data e hora (ex: `extract_20250906_100000.nc`), evitando a sobrescrita de arquivos.

```
1 def generate_output_filename(prefix: str = "extract") -> str:
2     timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
3     return f"{prefix}_{timestamp}.nc"
```

3. `extract_data(...)`: Executa a chamada principal à API (`client.retrieve`). Inclui um tratamento de exceções e logging para registrar o sucesso ou falha da operação.

```
1 def extract_data(dataset: str, request: Dict[str, Any], output_file
2 : str) -> None:
3     logging.info(f"Iniciando extra o para {output_file} ...")
4     try:
5         client.retrieve(dataset, request, output_file)
6         logging.info(f"Extra o conclu da com sucesso: {
7 output_file}")
8     except Exception as e:
9         logging.error(f"Erro durante a extra o: {e}")
10        raise
```

4. `preview_dataset(nc_file: str) -> None`: Após o download, utiliza a biblioteca `xarray` para abrir o arquivo `.nc` e imprimir um resumo de suas dimensões, coordenadas e variáveis, validando a integridade dos dados.

```

1 def preview_dataset(nc_file: str) -> None:
2     try:
3         ds = xr.open_dataset(nc_file, engine="netcdf4")
4         logging.info(f"Pr -visualiza o do dataset: {ds}")
5     except Exception as e:
6         logging.error(f"Erro ao abrir o arquivo NetCDF: {e}")
7         raise

```

2.3 Bibliotecas Utilizadas

- **cdsapi**: Cliente oficial para a API do CDS.
- **xarray & netCDF4**: Ferramentas padrão para manipulação de arrays multidimensionais e arquivos NetCDF.
- **logging**: Implementado para fornecer feedback claro sobre o progresso e possíveis erros.

A organização do código é lógica e segue as melhores práticas para scripts de automação e extração de dados.

3 Diferença entre Formatos de Arquivos: NetCDF vs. CSV

A escolha do formato **NetCDF** (.nc) como padrão no script é tecnicamente justificada e superior ao formato **CSV** para este tipo de aplicação.

Tabela 1: Comparativo entre os formatos NetCDF e CSV.

Característica	NetCDF (.nc)	CSV (.csv)
Estrutura	Multidimensional (lat, lon, tempo, etc.)	Bidimensional (linhas e colunas)
Compactação	Altamente eficiente (compressão binária)	Baseado em texto (arquivos maiores)
Performance	Otimizado para I/O de subconjuntos	Leitura sequencial e lenta

Quando usar cada formato

- **NetCDF**: É o formato preferencial para **armazenamento e manipulação primária** de dados climáticos. A estratégia correta é manter os dados em seu formato .nc original, que é compacto e rico em informações.
- **CSV**: Deve ser usado apenas como um formato de **saída final** para aplicações específicas (ex: análise de uma série temporal para um único ponto ou importação em planilhas) que não suportam NetCDF.

Recomendação: A transformação de NetCDF para CSV deve ser a **última etapa** do pipeline de análise e aplicada apenas ao subconjunto de dados estritamente necessário.

4 Instruções de Uso e Preparação do Ambiente

1. Criação de Conta e Obtenção de Chave de API:

- Registrar-se no portal do [CDS \(Climate Data Store\)](#).
- Copiar a chave de acesso (composta por UID e Key) do perfil de usuário.

2. Configuração do Ambiente:

- Crie um arquivo chamado `.env` na raiz do projeto.
- Adicione as seguintes linhas ao arquivo, substituindo os valores:

```
1 CDSAPI_URL=https://cds.climate.copernicus.eu/api/v2
2 CDSAPI_KEY=SUA_UID:SUA_API_KEY
3
```

3. Instalação de Dependências:

- A instalação das bibliotecas podem ser feitas por meio do uso de um arquivo `requirements.txt` ou a via `pip` diretamente:

```
1 pip install requiriments.txt
2 pip install cdsapi xarray python-dotenv netCDF4
3
```

4. Execução do Script:

- Personalize a função `get_request_params()` em `cdsAPI.py` com quaisquer parâmetros.
- Execute o script diretamente do terminal:

```
1 python cdsAPI.py
2
```

5 Pós-Processamento com CDO (Climate Data Operators)

Após o download, os arquivos `.nc` podem ser manipulados de forma eficiente com ferramentas de linha de comando como o **CDO (Climate Data Operators)**, que permitem realizar transformações rápidas sem a necessidade de carregar os dados completos em memória.

5.1 Média Espacial (`fldmean`)

```
1 cdo fldmean output.nc media_espacial_horaria.nc
```

- **O que faz:** calcula a **média espacial** de todas as grades da variável ao longo da área definida no arquivo.
- **Exemplo:** se `output.nc` contém temperatura a 2 m (`t2m`) em toda a América do Sul, `fldmean` gera uma série temporal da temperatura média da região.

```

(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo fldmean extract_20250905_085651.nc media_especial_horaria.nc
cdi warning (set_coordinates_varids): Coordinates variable number can't be assigned!
cdi warning (set_coordinates_varids): Coordinates variable expver can't be assigned!
cdo fldmean: Processed 128520 values from 3 variables over 120 timesteps [0.06s 49MB]
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo infon media_especial_horaria.nc
-1 :      Date      Time      Level Gridsize      Miss :      Minimum      Mean      Maximum : Parameter name
 1 : 2002-03-11 00:00:00      0      1      0 :      297.76      297.76      297.76 : stl1
 2 : 2002-03-11 00:00:00      0      1      0 :      298.07      298.07      298.07 : stl2
 3 : 2002-03-11 00:00:00      0      1      0 :      294.52      294.52      294.52 : lbt1
 4 : 2002-03-11 01:00:00      0      1      0 :      297.35      297.35      297.35 : stl1
 5 : 2002-03-11 01:00:00      0      1      0 :      297.92      297.92      297.92 : stl2
 6 : 2002-03-11 01:00:00      0      1      0 :      294.50      294.50      294.50 : lbt1
 7 : 2002-03-11 02:00:00      0      1      0 :      296.90      296.90      296.90 : stl1
 8 : 2002-03-11 02:00:00      0      1      0 :      297.74      297.74      297.74 : stl2
 9 : 2002-03-11 02:00:00      0      1      0 :      294.47      294.47      294.47 : lbt1
10 : 2002-03-11 03:00:00      0      1      0 :      295.65      295.65      295.65 : stl1
11 : 2002-03-11 03:00:00      0      1      0 :      297.55      297.55      297.55 : stl2
12 : 2002-03-11 03:00:00      0      1      0 :      294.44      294.44      294.44 : lbt1
13 : 2002-03-11 04:00:00      0      1      0 :      295.39      295.39      295.39 : stl1
14 : 2002-03-11 04:00:00      0      1      0 :      297.35      297.35      297.35 : stl2
15 : 2002-03-11 04:00:00      0      1      0 :      294.41      294.41      294.41 : lbt1
16 : 2002-03-11 05:00:00      0      1      0 :      295.04      295.04      295.04 : stl1
17 : 2002-03-11 05:00:00      0      1      0 :      297.14      297.14      297.14 : stl2
18 : 2002-03-11 05:00:00      0      1      0 :      294.38      294.38      294.38 : lbt1
19 : 2002-03-11 06:00:00      0      1      0 :      294.65      294.65      294.65 : stl1
20 : 2002-03-11 06:00:00      0      1      0 :      296.93      296.93      296.93 : stl2
21 : 2002-03-11 06:00:00      0      1      0 :      294.35      294.35      294.35 : lbt1
22 : 2002-03-11 07:00:00      0      1      0 :      294.47      294.47      294.47 : stl1
23 : 2002-03-11 07:00:00      0      1      0 :      296.72      296.72      296.72 : stl2
24 : 2002-03-11 07:00:00      0      1      0 :      294.32      294.32      294.32 : lbt1
25 : 2002-03-11 08:00:00      0      1      0 :      294.15      294.15      294.15 : stl1
26 : 2002-03-11 08:00:00      0      1      0 :      296.51      296.51      296.51 : stl2
27 : 2002-03-11 08:00:00      0      1      0 :      294.29      294.29      294.29 : lbt1

```

Figura 1: Resultado do comando fldmean aplicado ao arquivo output.nc.

- **Resultado:** o arquivo media_especial_horaria.nc tem **1 valor por timestep**, reduzindo drasticamente o tamanho do arquivo e facilitando análises temporais.

O comando a seguir exibe informações sobre o novo arquivo:

```
1 cdo infon media_especial_horaria.nc
```

5.2 Média Diária (daymean)

```
1 cdo daymean -mergetime output.nc teste_medial.nc
```

```

(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo daymean -mergetime extract_20250905_085651.nc teste_medial.nc
cdi(1) mergetime: Process started
cdi warning (set_coordinates_varids): Coordinates variable number can't be assigned!
cdi warning (set_coordinates_varids): Coordinates variable expver can't be assigned!
cdo daymean: Processed 128520 values from 3 variables over 120 timesteps [0.06s 50MB]
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo infon teste_medial.nc
-1 :      Date      Time      Level Gridsize      Miss :      Minimum      Mean      Maximum : Parameter name
 1 : 2002-03-11 11:30:00      0      357      0 :      293.93      297.51      302.65 : stl1
 2 : 2002-03-11 11:30:00      0      357      0 :      293.52      297.05      301.73 : stl2
 3 : 2002-03-11 11:30:00      0      357      0 :      289.16      294.40      301.07 : lbt1
 4 : 2002-03-12 11:30:00      0      357      0 :      293.68      297.61      302.99 : stl1
 5 : 2002-03-12 11:30:00      0      357      0 :      293.36      296.89      301.88 : stl2
 6 : 2002-03-12 11:30:00      0      357      0 :      289.11      294.37      300.84 : lbt1
 7 : 2002-03-13 11:30:00      0      357      0 :      294.00      298.00      303.95 : stl1
 8 : 2002-03-13 11:30:00      0      357      0 :      293.49      297.16      302.39 : stl2
 9 : 2002-03-13 11:30:00      0      357      0 :      289.11      294.43      301.01 : lbt1
10 : 2002-03-14 11:30:00      0      357      0 :      294.10      298.17      304.20 : stl1
11 : 2002-03-14 11:30:00      0      357      0 :      293.67      297.52      302.92 : stl2
12 : 2002-03-14 11:30:00      0      357      0 :      289.12      294.54      301.12 : lbt1
13 : 2002-03-15 11:30:00      0      357      0 :      293.48      297.93      303.36 : stl1
14 : 2002-03-15 11:30:00      0      357      0 :      293.60      297.64      303.08 : stl2
15 : 2002-03-15 11:30:00      0      357      0 :      289.13      294.67      301.14 : lbt1
cdo infon: Processed 5355 values from 3 variables over 5 timesteps [0.01s 47MB]
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$

```

Figura 2: Resultado do comando daymean após combinar timesteps horários.

- **O que faz:** calcula a **média diária** a partir de dados horários.
- **-mergetime:** combina vários arquivos ou timesteps em uma sequência contínua antes de calcular a média diária.
- **Exemplo:** dados horários de temperatura ao longo de um mês se tornam uma série de médias diárias.
- **Resultado:** o arquivo teste_medial.nc contém **1 valor por dia** para cada variável.

5.3 Conversão de Unidades (subc)

```
1 cdo subc,273.15 teste_medial.nc teste_medial1.nc
```

```
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo subc,273.15 teste_medial.nc teste_medial1.nc
cdo subc: Processed 5355 values from 3 variables over 5 timesteps [0.02s 49MB]
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo infon teste_medial1.nc
-1 :      Date      Time      Level Gridsize      Miss :      Minimum      Mean      Maximum : Parameter name
 1 : 2002-03-11 11:30:00      0      357      0 :      20.935      24.510      29.655 : stl1
 2 : 2002-03-11 11:30:00      0      357      0 :      20.517      24.054      28.731 : stl2
 3 : 2002-03-11 11:30:00      0      357      0 :      16.101      21.398      28.075 : lbt1
 4 : 2002-03-12 11:30:00      0      357      0 :      20.681      24.612      29.989 : stl1
 5 : 2002-03-12 11:30:00      0      357      0 :      20.365      23.891      28.885 : stl2
 6 : 2002-03-12 11:30:00      0      357      0 :      16.106      21.369      27.939 : lbt1
 7 : 2002-03-13 11:30:00      0      357      0 :      20.997      24.996      30.946 : stl1
 8 : 2002-03-13 11:30:00      0      357      0 :      20.498      24.160      29.388 : stl2
 9 : 2002-03-13 11:30:00      0      357      0 :      16.113      21.428      28.007 : lbt1
10 : 2002-03-14 11:30:00      0      357      0 :      21.180      25.174      31.199 : stl1
11 : 2002-03-14 11:30:00      0      357      0 :      20.666      24.522      29.919 : stl2
12 : 2002-03-14 11:30:00      0      357      0 :      16.122      21.538      28.115 : lbt1
13 : 2002-03-15 11:30:00      0      357      0 :      20.476      24.534      30.356 : stl1
14 : 2002-03-15 11:30:00      0      357      0 :      20.602      24.638      30.078 : stl2
15 : 2002-03-15 11:30:00      0      357      0 :      16.126      21.572      28.139 : lbt1
cdo infon: Processed 5355 values from 3 variables over 5 timesteps [0.01s 47MB]
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$
```

Figura 3: Arquivo teste_medial1.nc convertido de Kelvin para Celsius.

- **O que faz:** subtrai 273.15 de todos os valores, convertendo temperatura de Kelvin para Celsius.
- **Exemplo:** t2m passa de 297 K → 24 °C.
- **Resultado:** teste_medial1.nc está pronto para análises e visualizações em unidades mais intuitivas.

5.4 Observações Importantes

- Sempre **crie um novo arquivo** ao aplicar operações para evitar erros de permissão ou sobrescrita.
- É possível combinar várias operações em um **pipeline CDO**, reduzindo a quantidade de arquivos intermediários:

```
youserz@m028-emater:~/Documentos/programas/ClimateDataStore
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo subc,273.15 -daymean -mergetime extract_20250905_005651.nc temperatura_diaria_celsius.nc
cdo(1) daymean: Process started
cdo(2) mergetime: Process started
cdi warning (set_coordinates_varids): Coordinates variable number can't be assigned!
cdi warning (set_coordinates_varids): Coordinates variable expver can't be assigned!
cdo subc: Processed 5355 values from 3 variables over 5 timesteps [0.08s 50MB]
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$ cdo infon temperatura_diaria_celsius.nc
-1 :      Date      Time      Level Gridsize      Miss :      Minimum      Mean      Maximum : Parameter name
 1 : 2002-03-11 11:30:00      0      357      0 :      20.785      24.360      29.585 : stl1
 2 : 2002-03-11 11:30:00      0      357      0 :      20.367      23.904      28.581 : stl2
 3 : 2002-03-11 11:30:00      0      357      0 :      15.951      21.248      27.925 : lbt1
 4 : 2002-03-12 11:30:00      0      357      0 :      20.531      24.462      29.839 : stl1
 5 : 2002-03-12 11:30:00      0      357      0 :      20.215      23.741      28.735 : stl2
 6 : 2002-03-12 11:30:00      0      357      0 :      15.956      21.219      27.769 : lbt1
 7 : 2002-03-13 11:30:00      0      357      0 :      20.847      24.846      30.796 : stl1
 8 : 2002-03-13 11:30:00      0      357      0 :      20.340      24.010      29.238 : stl2
 9 : 2002-03-13 11:30:00      0      357      0 :      15.963      21.278      27.857 : lbt1
10 : 2002-03-14 11:30:00      0      357      0 :      20.950      25.024      31.049 : stl1
11 : 2002-03-14 11:30:00      0      357      0 :      20.516      24.372      29.769 : stl2
12 : 2002-03-14 11:30:00      0      357      0 :      15.972      21.388      27.965 : lbt1
13 : 2002-03-15 11:30:00      0      357      0 :      20.326      24.784      30.206 : stl1
14 : 2002-03-15 11:30:00      0      357      0 :      20.452      24.408      29.828 : stl2
15 : 2002-03-15 11:30:00      0      357      0 :      15.976      21.422      27.963 : lbt1
cdo infon: Processed 5355 values from 3 variables over 5 timesteps [0.02s 47MB]
(cdo_env) youserz@m028-emater:~/Documentos/programas/ClimateDataStore$
```

Figura 4: Combinação das operações.

```
1 cdo subc,273.15 -daymean -mergetime output.nc
  temperatura_diaria_celsius.nc
```

Neste exemplo, o CDO executa as seguintes operações em sequência:

1. Junta os timesteps (`mergetime`).
2. Calcula a média diária (`daymean`).
3. Converte de Kelvin para Celsius (`subc`).
4. Salva o resultado em `temperatura_diaria_celsius.nc`.