

# Manual de Uso

## Gerador de Figuras — Modelo Eta 3km / BESM

SisMOM / CPTEC-INPE

*Versao 2.3 — Junho de 2026*

*Grade gaussiana | Campos fixos | Variaveis 3D | Novos args ctl\_to\_yaml*



## 1. Introducao

Este pacote de scripts Python gera figuras e arquivos GeoTIFF Cloud Optimized (COG) a partir das saidas binarias do modelo Eta 3km acoplado ao BESM, operado pelo SisMOM/CPTEC-INPE. A configuracao e feita por arquivos YAML, sem necessidade de editar codigo Python.

### 1.1 Funcionalidades

- \* Leitura de binarios GrADS (big/little-endian float32, STREAM e SEQUENTIAL)
- \* Grade regular e grade irregular/gaussiana (YDEF LEVELS + OPTIONS YREV)
- \* Figuras PNG para variaveis 2D e 3D (por nivel de pressao)
- \* Acumulados de precipitacao (24h sinoptico ou periodo configuravel)
- \* Exportacao COG GeoTIFF por nivel de pressao para variaveis 3D
- \* Campos fixos (\_FF): topografia, mascara terra/mar — gerados uma unica vez
- \* Processamento paralelo com multiprocessing
- \* Configuracao completa via YAML: grade, modelo, caminhos, colormaps
- \* Geracao automatica de config.yaml e variables.yaml via ctl\_to\_yaml.py

### 1.2 Estrutura de arquivos

```
Figuras Eta/
config.yaml           # configuracao do modelo
variables.yaml        # definicao das variaveis
config_BESM.yaml      # exemplo: BESM global (grade gaussiana)
variables_BESM.yaml   # 61 vars (48 2D + 13 3D x18 niveis)
run.sh                # script de execucao
scripts/
  main.py              # orquestrador principal (CLI)
  config.py            # carregador YAML e constantes globais
  reader.py            # leitura dos binarios GrADS
  accumulate.py        # calculo e plot dos acumulados
  export_cog.py        # exportacao COG GeoTIFF
  plot_variables.py    # geracao das figuras PNG
  plot_utils.py        # colormaps e utilitarios de plot
  ctl_to_yaml.py       # gera config/variables a partir de CTL ou NetCDF
  download_data.py     # download dos dados do SisMOM
requirements.txt       # dependencias Python
```

## 2. Instalacao

### 2.1 Requisitos

Python 3.8 ou superior. Dependencias em scripts/requirements.txt:

| Pacote     | Versao minima | Finalidade                                      |
|------------|---------------|-------------------------------------------------|
| numpy      | >= 1.24       | Leitura e manipulacao dos arrays binarios       |
| rasterio   | >= 1.3        | Exportacao COG GeoTIFF                          |
| pyyaml     | >= 6.0        | Leitura dos arquivos YAML                       |
| matplotlib | >= 3.7        | Geracao das figuras PNG                         |
| scipy      | opcional      | Interpolacao grade gaussiana e fallback NetCDF3 |
| netCDF4    | opcional      | Leitura de NetCDF em ctl_to_yaml.py             |

### 2.2 Instalacao

```
pip install -r scripts/requirements.txt
```

```
# Para grade gaussiana (recomendado):  
pip install scipy
```

```
# Para NetCDF:  
pip install netCDF4
```

```
# Linhas de costa nas figuras (opcional):  
conda install -c conda-forge cartopy
```

### 3. Configuracao

Toda a configuracao e feita em dois arquivos YAML na raiz do projeto. O script `ctl_to_yaml.py` (secao 8) pode gerar esses arquivos automaticamente.

#### 3.1 config.yaml

```
run:
  ntimes: 121          # numero de passos de tempo (inclui analise = t0)
  dt_hours: 1          # intervalo de saida em horas

grid:
  nx: 2399  ny: 1671
  lon0: -90.99252  lat0: -38.12884
  dlon: 0.03  dlat: 0.03
  # Grade gaussiana: dlat: null + lat_levels: [...]

model:
  undef: 1.0e+20
  dtype: ">f4"          # big-endian float32
  file_prefix: "Eta03_BESM_{run_tag}+"
  file_suffix: "_2D.bin"
  file_timestamp: "{yyyy}{mm}{dd}{hh}"
  sequential: false
  yrev: false

paths:
  data_dir: "/dados/sismom/{data}/regional/eta/2D"
  output_dir: "figuras/campos"
  log_dir: "logs"
```

#### 3.2 Campo file\_timestamp

Define a parte variavel do nome do arquivo binario (timestamp de validade):

| Valor                | Formato               | Uso                |
|----------------------|-----------------------|--------------------|
| "{yyyy}{mm}{dd}{hh}" | 2026060600 (10 chars) | Padrao — Eta 2D    |
| "{yyyy}{dd}{hh}"     | 20260600 (8 chars)    | Eta 3D sem mes     |
| "" (vazio)           | sem timestamp         | Arquivo fixo (_FF) |

#### 3.3 Grade irregular/gaussiana

Quando a grade Y nao e regular (ex.: grade gaussiana do BESM/AGCM), use `dlat: null` e forneça `lat_levels`:

```
grid:
  nx: 192  ny: 96
  lon0: 0.0  lat0: -88.57217
  dlon: 1.875
  dlat: ~          # null = grade irregular
  lat_levels: [    # 96 latitudes gaussianas S -> N
    -88.57217, -86.72253, -84.86197, ..., 88.57217]

model:
  yrev: true          # OPTIONS YREV: dados invertidos em Y
```

O sistema interpola automaticamente para grade regular ao gerar COG GeoTIFF (scipy.interpolate.interp1d ou fallback np.interp).

### 3.4 Campos fixos (\_FF)

Campos estaticos como topografia e mascara terra/mar sao gerados uma unica vez por rodada:

```
model:
  file_prefix: "Eta03_BESM_{run_tag}"
  file_suffix: "_FF.bin"
  file_timestamp: "" # vazio = arquivo fixo
  fixed: true       # gerado uma unica vez por rodada
```

### 3.5 Variaveis 3D

Variaveis 3D sao configuradas em variables.yaml com campos adicionais:

```
- name: UVMT
  description: "Vento Zonal (U)"
  cmap: bwr
  vmin: -30  vmax: 30
  ndim: 3
  nlev: 18
  levels: [1000, 925, 850, 775, 700, 500, 400, 300, 250, 200, 150, 100, 70, 50, 30,
20, 10, 3]
  plot_levels: [1000, 850, 500, 200] # niveis a gerar figuras/COG
```

COGs 3D sao nomeados com o nivel: UVMT\_850hPa\_2026060600.tif

### 3.6 Template data\_dir

O campo paths.data\_dir aceita variaveis de substituicao:

| Variavel            | Substituicao                        |
|---------------------|-------------------------------------|
| {data} ou {run_tag} | YYYYMMDDHH (tag completa da rodada) |
| {yyyy}              | Ano com 4 digitos                   |
| {mm}                | Mes com 2 digitos                   |
| {dd}                | Dia com 2 digitos                   |
| {hh}                | Hora com 2 digitos                  |

## 4. Tag da Rodada (--run)

A tag identifica o horario de inicio da previsao no formato YYYYMMDDHH. Passada sempre via CLI, nunca armazenada no config.yaml.

| Formato              | Exemplo            | Resultado                    |
|----------------------|--------------------|------------------------------|
| YYYYMMDDHH           | --run 2026060600   | Rodada de 06/06/2026 00Z     |
| HH (so a hora)       | --run 00           | 00Z da data atual do sistema |
| H (1 digito)         | --run 0            | 00Z da data atual            |
| Variavel de ambiente | export RUN_TAG=... | Equivalente ao --run         |

```
./run.sh --run 2026060600    # rodada completa
./run.sh --run 00            # 00Z de hoje
./run.sh --run 12            # 12Z de hoje
export RUN_TAG=2026060600 && ./run.sh --cog-only
```

## 5. Referencia de Linha de Comando

### 5.1 Execucao via run.sh

```
./run.sh --run 00 # PNG + acumulados 24h
./run.sh --run 00 --cog-only # somente COG GeoTIFF
./run.sh --run 00 --cog --workers 8 # PNG + COG em paralelo
./run.sh --run 00 --accum-hours 6 # acumulado de 6h
./run.sh --run 00 --only-accum # somente acumulados
```

### 5.2 Todos os parametros

| Parametro        | Padrao         | Descricao                                    |
|------------------|----------------|----------------------------------------------|
| --run TAG        | —              | Tag da rodada: YYYYMMDDHH ou HH. Obrigatorio |
| --config YAML    | config.yaml    | Arquivo de configuracao alternativo          |
| --vars-file YAML | variables.yaml | Arquivo de variaveis alternativo             |
| --data_dir DIR   | —              | Caminho completo para os .bin                |
| --vars V1 V2 ... | enabled=true   | Lista de variaveis a processar               |
| --accum_hours N  | 24             | Periodo do acumulado em horas                |
| --workers N      | ncpus          | Processos paralelos                          |
| --only_accum     | false          | Gera somente acumulados                      |
| --only_fields    | false          | Gera somente campos horarios                 |
| --sequential     | false          | Leitura com marcadores Fortran               |
| --cog            | false          | Exportar COG alem dos PNGs                   |
| --cog_only       | false          | Exportar somente COG                         |
| --skip_existing  | false          | Pula arquivos ja gerados                     |
| --quiet          | false          | Suprime saida de progresso                   |



## 6. Campos Acumulados

O sistema calcula acumulados subtraindo o valor do horario inicial do horario final.

### 6.1 Nomenclatura

```
PREC_2026060600_acum24h.png    # acumulado 24h  
PREC_2026060600_acum06h.png    # acumulado 6h
```

### 6.2 Período

```
./run.sh --run 00 --accum-hours 24    # padrao  
./run.sh --run 00 --accum-hours 6     # janelas de 6h  
./run.sh --run 00 --only-accum        # somente acumulados
```

## 7. Exportacao COG GeoTIFF

Alem das figuras PNG, o sistema exporta campos como Cloud-Optimized GeoTIFF (COG), adequado para ingestao em sistemas GIS.

### 7.1 Ativando

```
./run.sh --run 00 --cog          # PNG + COG  
./run.sh --run 00 --cog-only    # somente COG
```

### 7.2 Nomenclatura

```
# Variaveis 2D:  
PREC_2026060600.tif  
  
# Variaveis 3D (por nivel):  
UVMT_850hPa_2026060600.tif  
GHMT_500hPa_2026060606.tif  
  
# Campos fixos:  
TOPO_2026060600.tif    # gerado uma unica vez
```

### 7.3 Grade gaussiana

Para grades irregulares, o COG e gerado apos interpolacao automatica para grade regular de mesmo espacamento medio (scipy ou np.interp). O transform rasterio reflete a grade regular resultante.

## 8. Geracao de Configuracao (ctl\_to\_yaml.py)

Gera automaticamente config.yaml e variables.yaml a partir de um descritor GrADS CTL ou NetCDF.

### 8.1 Uso basico

```
# A partir de CTL:
python scripts/ctl_to_yaml.py Eta03_BESM.ctl

# Com data_dir e ntimes explicitos:
python scripts/ctl_to_yaml.py template.ctl \
  --data-dir "/dados/sismom/{data}/global/atmos" \
  --ntimes 30

# A partir de NetCDF:
python scripts/ctl_to_yaml.py saida_modelo.nc

# Salvar junto ao input:
python scripts/ctl_to_yaml.py Eta03.ctl --next-to-input
```

### 8.2 Todas as opcoes

| Opcao            | Padrao         | Descricao                                               |
|------------------|----------------|---------------------------------------------------------|
| input            | (posicional)   | Caminho para CTL ou NetCDF. Obrigatorio                 |
| --format         | auto           | Forca formato: auto   ctl   netcdf                      |
| --config ARQUIVO | config.yaml    | Nome/caminho do config.yaml gerado                      |
| --vars ARQUIVO   | variables.yaml | Nome/caminho do variables.yaml gerado                   |
| --data-dir PATH  | auto-detectado | Substitui data_dir. Aceita {data} {yyyy} {mm} {dd} {hh} |
| --ntimes N       | do CTL         | Substitui ntimes (util em CTL de timestep unico)        |
| --dt-hours H     | do CTL         | Substitui dt_hours                                      |
| --next-to-input  | false          | Salva YAMLS no mesmo diretorio do input                 |
| --stdout         | false          | Imprime YAMLS no terminal                               |
| --dry-run        | false          | Simula sem escrever arquivos                            |
| --list-vars      | false          | Lista variaveis detectadas e encerra                    |
| --force          | false          | Sobrescreve arquivos existentes                         |

### 8.3 Deteccao automatica por tipo de CTL

| Padrao de CTL                         | Detectado como            | Config gerado                        |
|---------------------------------------|---------------------------|--------------------------------------|
| DSET ^...<br>%y4%m2%d2%h2..._2D.bin   | CTL com template<br>GrADS | file_timestamp: "{yyyy}{mm}{dd}{hh}" |
| DSET ^...{run_tag}+{valid}<br>_3D.bin | CTL timestep unico 3D     | file_timestamp: "{yyyy}{dd}{hh}"     |
| DSET ^..._FF.bin (TDEF 1)             | Campos fixos              | file_timestamp: "", fixed: true      |
| YDEF N LEVELS lat1 lat2 ...           | Grade gaussiana           | dlat: null + lat_levels: [...]       |
| OPTIONS YREV                          | Eixo Y invertido          | yrev: true                           |
| OPTIONS SEQUENTIAL                    | Formato Fortran           | sequential: true                     |

## 8.4 Campos fixos (\_FF)

Arquivo com run\_tag mas sem timestamp de validade (TDEF 1). O script detecta automaticamente e emite fixed: true:

```
python scripts/ctl_to_yaml.py Eta03_BESM_2026060600_FF.ctl \
  --config config_FF.yaml --vars variables_FF.yaml

# config_FF.yaml:
model:
  file_prefix: "Eta03_BESM_{run_tag}"
  file_suffix: "_FF.bin"
  file_timestamp: ""
  fixed: true

# variables_FF.yaml:
- name: TOPO
  fixed: true # gerado uma unica vez por rodada
- name: LSMK
  fixed: true
```

## 8.5 Grade gaussiana (YDEF LEVELS)

O script coleta todas as latitudes multi-linha e emite lat\_levels no config:

```
# CTL:
ydef 96 levels
-88.57217 -86.72253 ...

# config.yaml gerado:
grid:
  dlat: ~
  lat_levels: [-88.57217, -86.72253, ...]
model:
  yrev: true
```

## 8.6 Variaveis 3D

Variaveis com NLEV > 0 recebem ndim: 3, nlev, levels e plot\_levels (primeiros 6 niveis):

```
# CTL: UVMT 18 99 TIME MEAN ZONAL WIND (U)

# variables.yaml gerado:
- name: UVMT
  ndim: 3
  nlev: 18
  levels: [1000, 925, 850, ..., 3]
  plot_levels: [1000, 925, 850, 500, 250, 200]
```

## 8.7 Exemplo BESM global

```
python scripts/ctl_to_yaml.py \
  /dados/sismom/SisMOM/sismom_forecast/2026060600/global/atmos/template.ctl \
  --data-dir "/dados/sismom/SisMOM/sismom_forecast/{data}/global/atmos" \
  --config config_BESM.yaml \
```

```
--vars    variables_BESM.yaml

# Resultado: grade gaussiana 192x96, YREV, SEQUENTIAL
#           61 variaveis: 48 2D + 13 3D x18 niveis
```

## 8.8 Inferencia de colormap

| Criterio                    | Exemplos               | Colormap inferido           |
|-----------------------------|------------------------|-----------------------------|
| Nome exato (tabela interna) | PREC, PRCV, NEVE       | precip (colormap proprio)   |
| Nome exato                  | TP2M, TSMT, ATMT       | RdBu_r, hot_r               |
| Nome exato                  | UVMT, VVMT, U10M, V10M | bwr                         |
| Fallback por unidade        | K, Kelvin              | RdBu_r                      |
| Fallback por unidade        | m/s, kt                | bwr                         |
| Fallback por unidade        | mm/h, mm/d             | precip                      |
| Padrao final                | qualquer outro         | viridis / escala automatica |

## 9. Solucao de Problemas

| Sintoma                        | Causa provavel                   | Solucao                                   |
|--------------------------------|----------------------------------|-------------------------------------------|
| FileNotFoundError ao ler .bin  | data_dir incorreto               | Verifique paths.data_dir ou --data_dir    |
| Imagens em branco              | undef nao configurado            | Verifique undef no config.yaml            |
| ValueError: shape mismatch     | nx/ny diverge do arquivo         | Re-execute ctl_to_yaml.py                 |
| Nenhum arquivo encontrado (3D) | file_timestamp usa {dd} sem {mm} | file_timestamp: "{yyyy}{dd}{hh}" para 3D  |
| BrokenProcessPool / OOM        | vars 3D carregam todos os niveis | Normal; sistema usa read_fields_selective |
| COG com coord erradas          | grade gaussiana sem lat_levels   | Regenere config com ctl_to_yaml.py        |
| ModuleNotFoundError: scipy     | biblioteca nao instalada         | pip install scipy                         |
| ModuleNotFoundError: netCDF4   | biblioteca nao instalada         | pip install netCDF4                       |

## 10. Historico de Versoes

| Versa<br>o | Data    | Alteracoes                                                                                                   |
|------------|---------|--------------------------------------------------------------------------------------------------------------|
| 1.0        | 2025-01 | Versao inicial: campos 2D, PNGs, acumulados 24h                                                              |
| 1.1        | 2025-03 | Suporte COG GeoTIFF, --cog-only, piramides opcionais                                                         |
| 1.2        | 2025-06 | Tag de rodada via CLI (--run), formato HH e YYYYMMDDHH                                                       |
| 1.3        | 2025-09 | Paralelismo (--workers), --skip-existing, --quiet                                                            |
| 2.0        | 2026-05 | ctl_to_yaml.py: geracao automatica config.yaml/variables.yaml a partir de CTL                                |
| 2.1        | 2026-06 | Suporte NetCDF/CF: deteccao por nome, axis e standard_name                                                   |
| 2.2        | 2026-06 | Variaveis 3D: VAR_INDEX binario, NLEV/LEVELS/PLOT_LEVELS, COG por nivel                                      |
| 2.3        | 2026-06 | Campos fixos (_FF). Grade gaussiana (YDEF LEVELS, YREV, regrid auto). Args: --data-dir, --ntimes, --dt-hours |