In [1]:
```
from src import *
from verde_source import regular, interp_at

from tqdm import tqdm
from shapely.ops import transform
from pylab import *
from shapely.geometry import Point, Polygon

import verde as vd
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import geopandas as gpd
import pyproj
import os
```

```
/home/ggrl/.config/ambiente_geologico/lib/python3.10/site-packages/geopandas/_compat.py:112: UserWarning: The Shapely GEOS version (3.10.2-CAPI-1.16.0) is incompatible with the GEOS version PyGEOS was compiled with (3.10.4-CAPI-1.16.2). Conversions between both will be slow.
  warnings.warn(
```

In [2]:
```
import warnings
warnings.filterwarnings("ignore")
%load_ext autoreload
%autoreload 2


%matplotlib inline
```

In [3]:
```
list_geof = os.listdir('/home/ggrl/database/geof/')
list_geof
```

Out[3]:
```
['mag_line_1105',
 'mag_1039',
 'gama_line_1105',
 'mag_line_1089',
 'gama_1039',
 'mag_3022',
 'mag_1105',
 'mag_tie_1105',
 'gama_1039_mdt.csv',
 'gama_line_1089',
 'gama_3022',
 'gama_line_1092',
 'geof_1039']
```

Construindo Quadrícula

In [4]:
```
quadricula = Build_mc(escala='100k',ID=['SF23_YA','SF23_YB','SF23_VC','SF23_VD'],verbose=True)
```

```
100%|██████████| 4/4 [00:00<00:00, 246.57it/s]
24it [00:00, 10088.52it/s]
```

```
- Folha "SF23_YA_I" adicionada.

- Folha "SF23_YA_IV" adicionada.

- Folha "SF23_YA_II" adicionada.

- Folha "SF23_YA_V" adicionada.

- Folha "SF23_YA_III" adicionada.

- Folha "SF23_YA_VI" adicionada.

- Folha "SF23_YB_I" adicionada.

- Folha "SF23_YB_IV" adicionada.

- Folha "SF23_YB_II" adicionada.

- Folha "SF23_YB_V" adicionada.

- Folha "SF23_YB_III" adicionada.

- Folha "SF23_YB_VI" adicionada.

- Folha "SF23_VC_I" adicionada.

- Folha "SF23_VC_IV" adicionada.

- Folha "SF23_VC_II" adicionada.

- Folha "SF23_VC_V" adicionada.

- Folha "SF23_VC_III" adicionada.

- Folha "SF23_VC_VI" adicionada.

- Folha "SF23_VD_I" adicionada.

- Folha "SF23_VD_IV" adicionada.

- Folha "SF23_VD_II" adicionada.

- Folha "SF23_VD_V" adicionada.

- Folha "SF23_VD_III" adicionada.

- Folha "SF23_VD_VI" adicionada.

 24 folhas adicionadas.
```

Adicionando dados brutos à Quadrícula

In [5]:

```
gama_3022,mag_3022=Upload_geof(quadricula,'gama_3022','mag_3022',1100)
gama_1105,mag_1105=Upload_geof(quadricula,'gama_line_1105','mag_line_1105',1100)
gama_1039,mag_1039=Upload_geof(quadricula,'gama_1039','mag_1039',1100)
```

```
67%|██████▋   | 16/24 [00:00<00:00, 18.73it/s]
- gama_3022 atualizado na folha: SF23_VC_II com 3977 pontos
- mag_3022 atualizado na folha: SF23_VC_II com 39600 pontos
- gama_3022 atualizado na folha: SF23_VC_III com 42550 pontos
- mag_3022 atualizado na folha: SF23_VC_III com 424403 pontos
- gama_3022 atualizado na folha: SF23_VC_VI com 65135 pontos
83%|████████▎ | 20/24 [00:01<00:00, 14.73it/s]
- mag_3022 atualizado na folha: SF23_VC_VI com 649894 pontos
- gama_3022 atualizado na folha: SF23_VD_I com 116131 pontos
- mag_3022 atualizado na folha: SF23_VD_I com 1159366 pontos
- gama_3022 atualizado na folha: SF23_VD_IV com 208327 pontos
- mag_3022 atualizado na folha: SF23_VD_IV com 2080714 pontos
- gama_3022 atualizado na folha: SF23_VD_II com 258250 pontos
92%|█████████▏| 22/24 [00:01<00:00, 13.07it/s]
- mag_3022 atualizado na folha: SF23_VD_II com 2579468 pontos
- gama_3022 atualizado na folha: SF23_VD_V com 351373 pontos
- mag_3022 atualizado na folha: SF23_VD_V com 3510117 pontos
- gama_3022 atualizado na folha: SF23_VD_III com 396326 pontos
100%|██████████| 24/24 [00:01<00:00, 15.02it/s]
- mag_3022 atualizado na folha: SF23_VD_III com 3959064 pontos
- gama_3022 atualizado na folha: SF23_VD_VI com 479609 pontos
- mag_3022 atualizado na folha: SF23_VD_VI com 4791195 pontos
25%|██▌       | 6/24 [00:00<00:01, 11.95it/s]
```

```
- gama_line_1105 atualizado na folha: SF23_YA_III com 40936 pontos
- mag_line_1105 atualizado na folha: SF23_YA_III com 426368 pontos
- gama_line_1105 atualizado na folha: SF23_YA_VI com 44459 pontos
- mag_line_1105 atualizado na folha: SF23_YA_VI com 462963 pontos
- gama_line_1105 atualizado na folha: SF23_YB_I com 121895 pontos
```

```
33%|███▎      | 8/24 [00:00<00:01, 10.15it/s]
```

```
- mag_line_1105 atualizado na folha: SF23_YB_I com 1275320 pontos
- gama_line_1105 atualizado na folha: SF23_YB_IV com 197696 pontos
- mag_line_1105 atualizado na folha: SF23_YB_IV com 2076440 pontos
- gama_line_1105 atualizado na folha: SF23_YB_II com 278542 pontos
```

```
42%|████▏     | 10/24 [00:01<00:01,  8.94it/s]
```

```
- mag_line_1105 atualizado na folha: SF23_YB_II com 2927434 pontos
- gama_line_1105 atualizado na folha: SF23_YB_V com 359624 pontos
- mag_line_1105 atualizado na folha: SF23_YB_V com 3765382 pontos
- gama_line_1105 atualizado na folha: SF23_YB_III com 437570 pontos
```

```
50%|█████     | 12/24 [00:01<00:01,  7.72it/s]
```

```
- mag_line_1105 atualizado na folha: SF23_YB_III com 4561010 pontos
- gama_line_1105 atualizado na folha: SF23_YB_VI com 516831 pontos
- mag_line_1105 atualizado na folha: SF23_YB_VI com 5290643 pontos
```

```
62%|██████▎   | 15/24 [00:01<00:01,  8.79it/s]
```

```
- mag_line_1105 atualizado na folha: SF23_VC_II com 5295619 pontos
- gama_line_1105 atualizado na folha: SF23_VC_III com 539857 pontos
```

```
75%|███████▌  | 18/24 [00:02<00:00,  7.84it/s]
```

```
- mag_line_1105 atualizado na folha: SF23_VC_III com 5527344 pontos
- gama_line_1105 atualizado na folha: SF23_VC_VI com 579605 pontos
- mag_line_1105 atualizado na folha: SF23_VC_VI com 5927381 pontos
```

```
83%|████████▎ | 20/24 [00:02<00:00,  7.16it/s]
```

```
- mag_line_1105 atualizado na folha: SF23_VD_I com 5934046 pontos
- gama_line_1105 atualizado na folha: SF23_VD_IV com 582867 pontos
- mag_line_1105 atualizado na folha: SF23_VD_IV com 5969405 pontos
```

```
92%|█████████▏| 22/24 [00:02<00:00,  7.70it/s]
```

```
- gama_line_1105 atualizado na folha: SF23_VD_V com 585325 pontos
- mag_line_1105 atualizado na folha: SF23_VD_V com 5996612 pontos
```

```
100%|██████████| 24/24 [00:02<00:00,  8.58it/s]
```

```
- gama_line_1105 atualizado na folha: SF23_VD_VI com 588216 pontos
- mag_line_1105 atualizado na folha: SF23_VD_VI com 6023142 pontos
```

```
38%|███▊      | 9/24 [00:00<00:00, 87.67it/s]
```

```
- gama_1039 atualizado na folha: SF23_YA_II com 34745 pontos
- mag_1039 atualizado na folha: SF23_YA_II com 34745 pontos
- gama_1039 atualizado na folha: SF23_YA_V com 69142 pontos
- mag_1039 atualizado na folha: SF23_YA_V com 69142 pontos
- gama_1039 atualizado na folha: SF23_YA_III com 96501 pontos
- mag_1039 atualizado na folha: SF23_YA_III com 96501 pontos
- gama_1039 atualizado na folha: SF23_YA_VI com 130239 pontos
- mag_1039 atualizado na folha: SF23_YA_VI com 130239 pontos
- gama_1039 atualizado na folha: SF23_VC_V com 132926 pontos
- mag_1039 atualizado na folha: SF23_VC_V com 132926 pontos
```

```
75%|███████▌  | 18/24 [00:00<00:00, 88.53it/s]
```

```
- gama_1039 atualizado na folha: SF23_VC_VI com 134733 pontos
- mag_1039 atualizado na folha: SF23_VC_VI com 134733 pontos
```

```
100%|██████████| 24/24 [00:00<00:00, 89.19it/s]
```

geof_list_ids = list(quadricula.keys())

print(len(geof_list_ids)) for id in geof_list_ids: print(f' - Folha: {id}') carta=quadricula[id] for data in list(carta.keys())[2:]: print(f' - {data}')

In [6]:
```
quadricula=pop_nodata(quadricula)
len(quadricula.keys())
```

```
100%|██████| 24/24 [00:00<00:00, 466033.78it/s]
```

Out[6]:
```
20
```

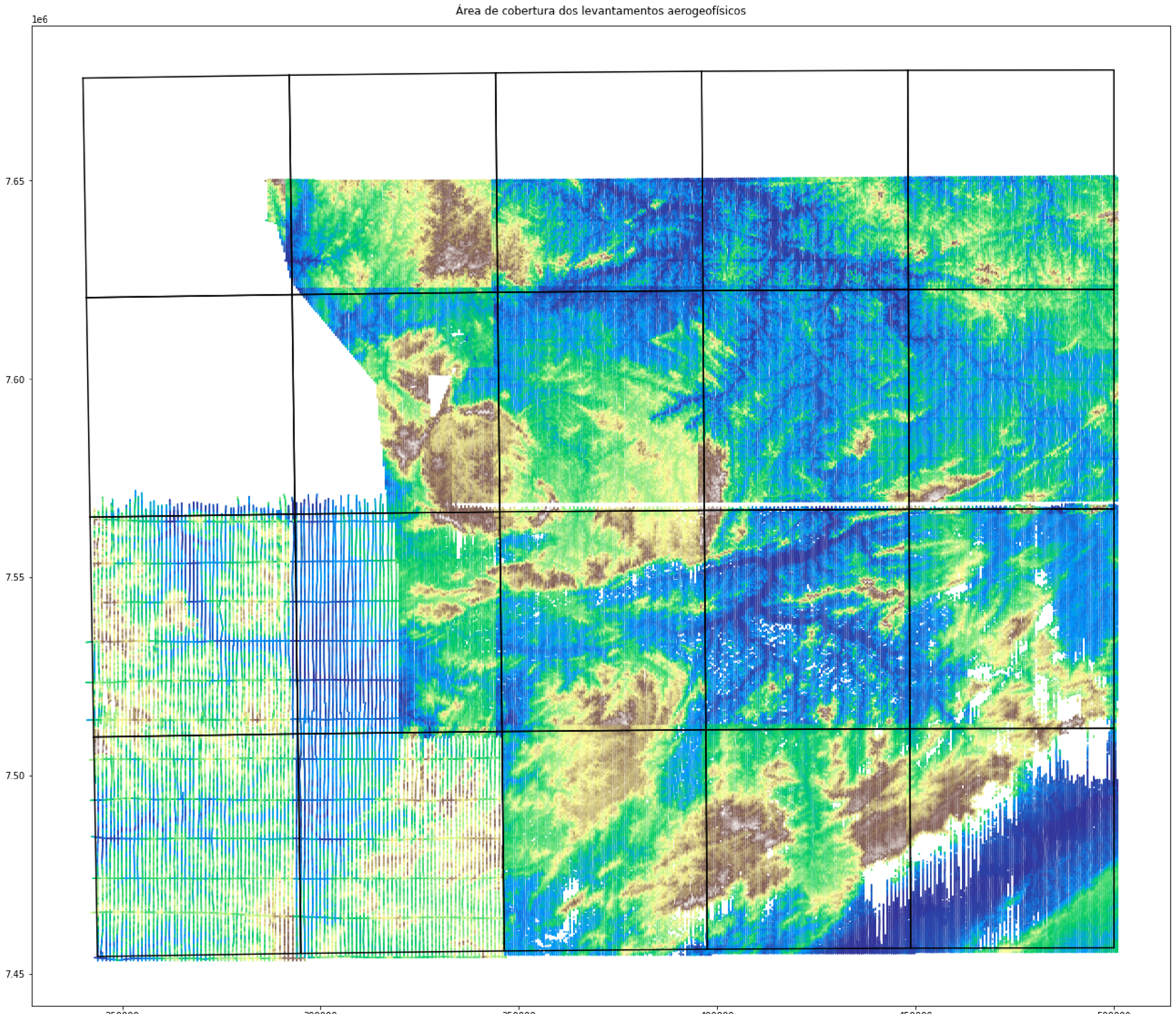
In [7]:
```
for id in list(quadricula.keys()):


    print(f' - Folha:  {id}')
    carta = quadricula[id]
    print(f'    - {list(carta.keys())}')
```

```
- Folha:  SF23_YA_II
   - ['folha', 'escala', 'gama_1039', 'mag_1039']
- Folha:  SF23_YA_V
   - ['folha', 'escala', 'gama_1039', 'mag_1039']
- Folha:  SF23_YA_III
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105', 'gama_1039', 'mag_1039']
- Folha:  SF23_YA_VI
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105', 'gama_1039', 'mag_1039']
- Folha:  SF23_YB_I
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_YB_IV
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_YB_II
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_YB_V
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_YB_III
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_YB_VI
   - ['folha', 'escala', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_VC_II
   - ['folha', 'escala', 'gama_3022', 'mag_3022', 'mag_line_1105']
- Folha:  SF23_VC_V
   - ['folha', 'escala', 'gama_1039', 'mag_1039']
- Folha:  SF23_VC_III
   - ['folha', 'escala', 'gama_3022', 'mag_3022', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_VC_VI
   - ['folha', 'escala', 'gama_3022', 'mag_3022', 'gama_line_1105', 'mag_line_1105', 'gama_1039', 'mag_1039']
- Folha:  SF23_VD_I
   - ['folha', 'escala', 'gama_3022', 'mag_3022', 'mag_line_1105']
- Folha:  SF23_VD_IV
   - ['folha', 'escala', 'gama_3022', 'mag_3022', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_VD_II
   - ['folha', 'escala', 'gama_3022', 'mag_3022']
- Folha:  SF23_VD_V
   - ['folha', 'escala', 'gama_3022', 'mag_3022', 'gama_line_1105', 'mag_line_1105']
- Folha:  SF23_VD_III
   - ['folha', 'escala', 'gama_3022', 'mag_3022']
- Folha:  SF23_VD_VI
   - ['folha', 'escala', 'gama_3022', 'mag_3022', 'gama_line_1105', 'mag_line_1105']
```

Vizualisando Área de Estudo

In [8]:

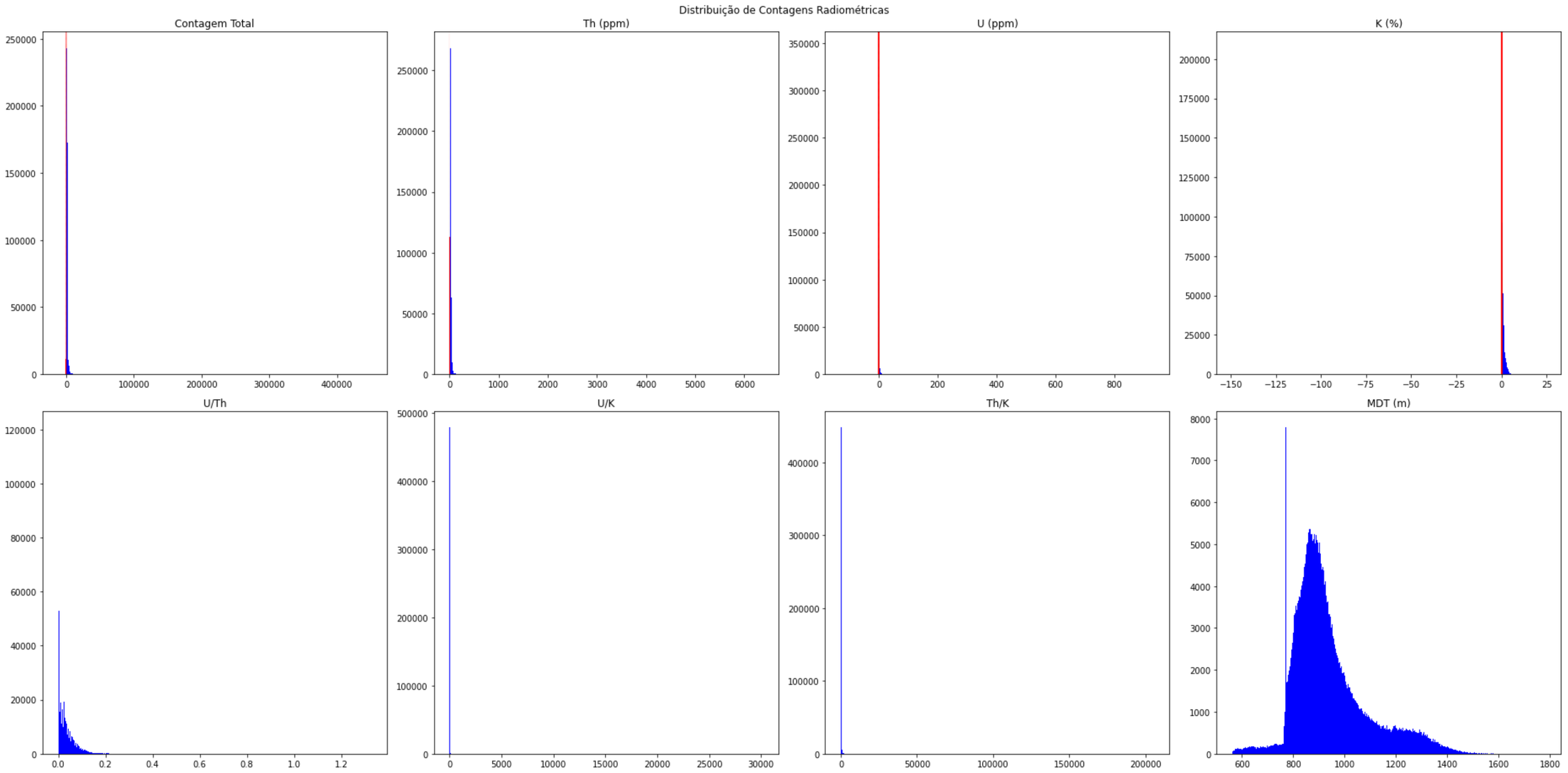
```
plt.figure(figsize=(21,16))
for id in list(quadricula.keys()):
    carta=quadricula[id]
    plt.plot(*transform_to_carta_utm(carta['folha']).exterior.xy,color='black')
    for data in list(carta.keys())[2:]:
        if 'mag' in data:
            pass
        else:
            plt.scatter(carta[data].X,carta[data].Y,c=carta[data].MDT,cmap='terrain',s=0.5,marker='H')
            plt.axis('scaled')
plt.suptitle('Área de cobertura dos levantamentos aerogeofísicos')
plt.tight_layout()
```



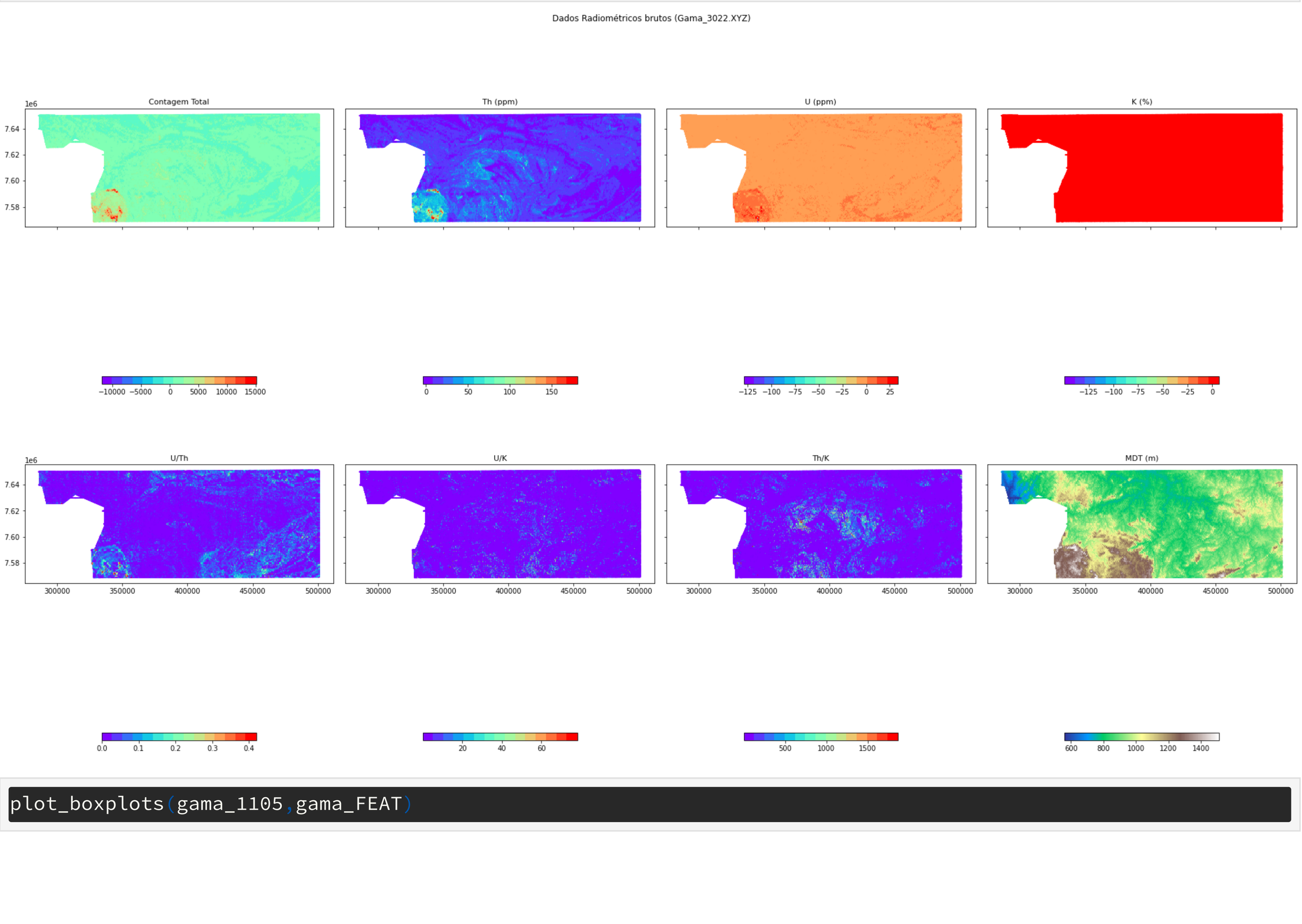
Visualizando dados Radiométricos Brutos

In [9]:
```
plot_histograms(gama_3022)
```

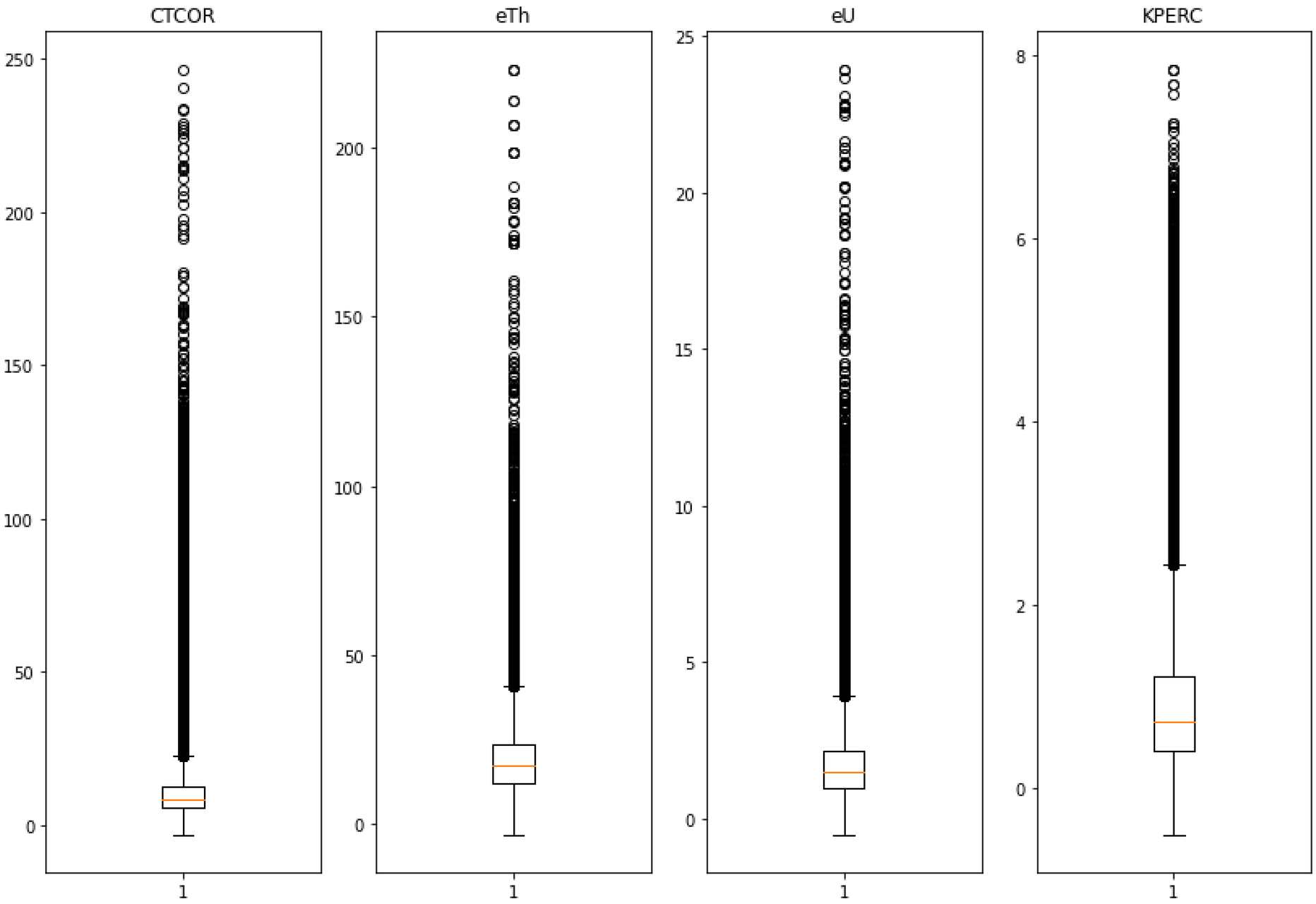
```
             count         mean          std        min       0.1%        1%  \
CTCOR     479609.0  1291.294875  1779.898133 -11834.130 -129.89784  -10.0492
eTh       479609.0    16.962850    23.742883     -3.800   -0.30000    0.7000
eU        479609.0     0.438626     3.492612   -129.300   -2.60000   -1.0000
KPERC     479609.0     0.675418     0.936380   -149.323   -0.18900   -0.0920
MDT       479609.0   940.063014   146.910103    562.040  577.03000  650.7500
THKRAZAO  479609.0   108.895627   619.252370      0.155    0.76900    2.3010
UTHRAZAO  479609.0     0.029148     0.038417      0.000    0.00000    0.0010
UKRAZAO   479609.0     2.947827    88.325810      0.002    0.00400    0.0060

                5%      25%       50%       75%         99.95%         max
CTCOR     345.8280  729.320  1079.270  1511.140  15080.726160  450895.020
eTh         4.4000    9.200    13.800    19.900    180.100000    6371.400
eU         -0.6000   -0.100     0.200     0.600     33.900000     934.800
KPERC      -0.0160    0.133     0.354     0.881      7.021392      24.286
MDT       772.2000  845.950   903.290   996.270   1510.670000    1782.280
THKRAZAO    5.3184   15.517    36.123    99.831   1870.690336  205529.094
UTHRAZAO    0.0010    0.002     0.016     0.042      0.419588       1.328
UKRAZAO     0.0130    0.094     0.593     2.155     78.128912   30156.090
```



In [10]:
```
plot_raw_gama_data(gama_3022,suptitle='Dados Radiométricos brutos (Gama_3022.XYZ)')
```



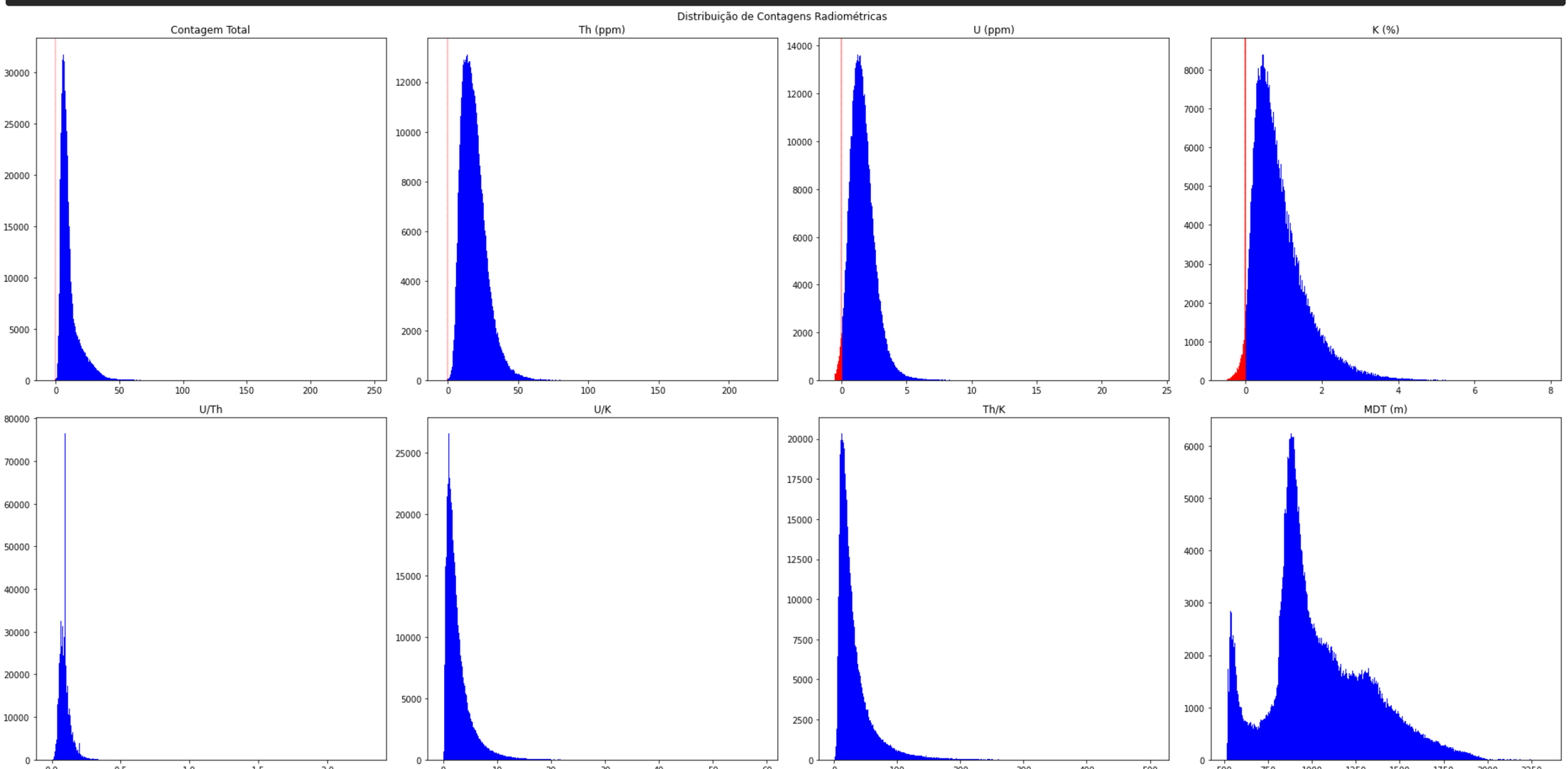
In [11]:
```
plot_boxplots(gama_1105,gama_FEAT)
```



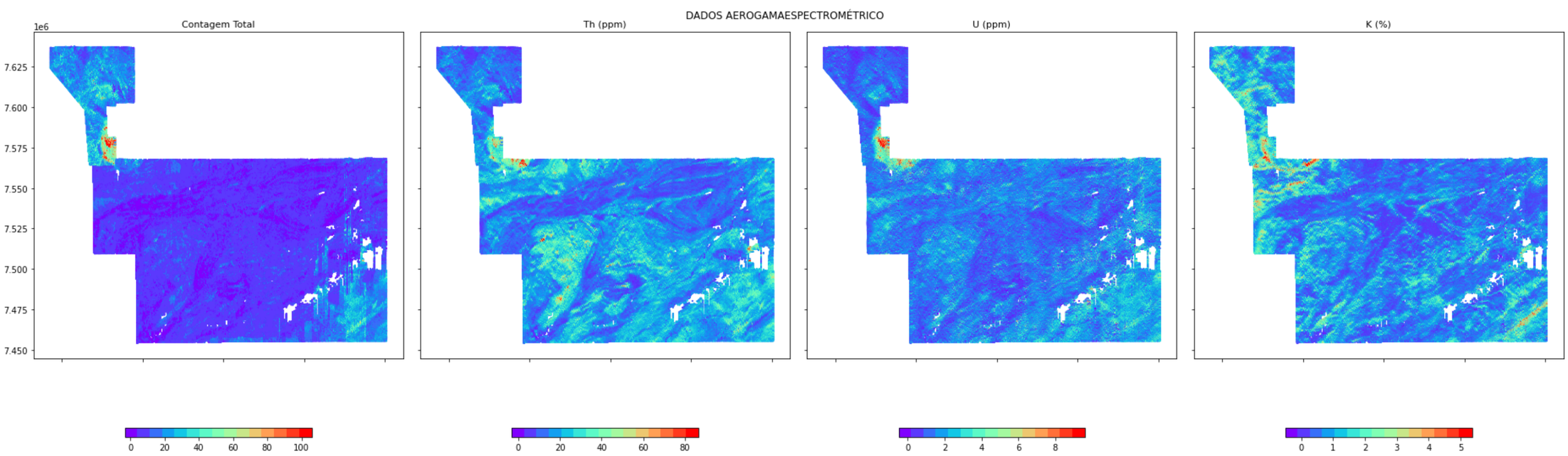
In [12]:
```
plot_histograms(gama_1105)
```

	count	mean	std	min	0.1%	1%
CTCOR	588216.0	10.961601	9.083312	-2.947	1.008860	2.43300
eTh	588216.0	18.883276	9.484438	-3.080	2.190430	4.75915
eU	588216.0	1.646568	1.019306	-0.500	-0.400785	-0.09000
KPERC	588216.0	0.911165	0.717427	-0.499	-0.330000	-0.07600
MDT	588216.0	1040.922448	287.145718	513.280	521.060000	532.31000
THKRAZAO	588216.0	31.855617	29.509486	1.079	2.615000	4.94500
UTHRAZAO	588216.0	0.094981	0.043069	0.000	0.000000	0.02100
UKRAZAO	588216.0	2.798359	2.571587	0.046	0.159000	0.30000

	5%	25%	50%	75%	99.95%	max
CTCOR	3.53500	5.788	8.221	12.500	106.078500	246.600
eTh	7.20775	12.133	17.330	23.679	86.530710	223.136
eU	0.31700	0.972	1.514	2.153	9.640000	23.934
KPERC	0.11500	0.412	0.735	1.227	5.360000	7.840
MDT	568.60750	864.500	973.390	1226.570	2004.160975	2326.220
THKRAZAO	7.61000	13.825	22.353	38.557	259.671625	505.144
UTHRAZAO	0.04300	0.067	0.091	0.109	0.412000	2.311
UKRAZAO	0.53800	1.186	2.018	3.500	24.198140	59.007

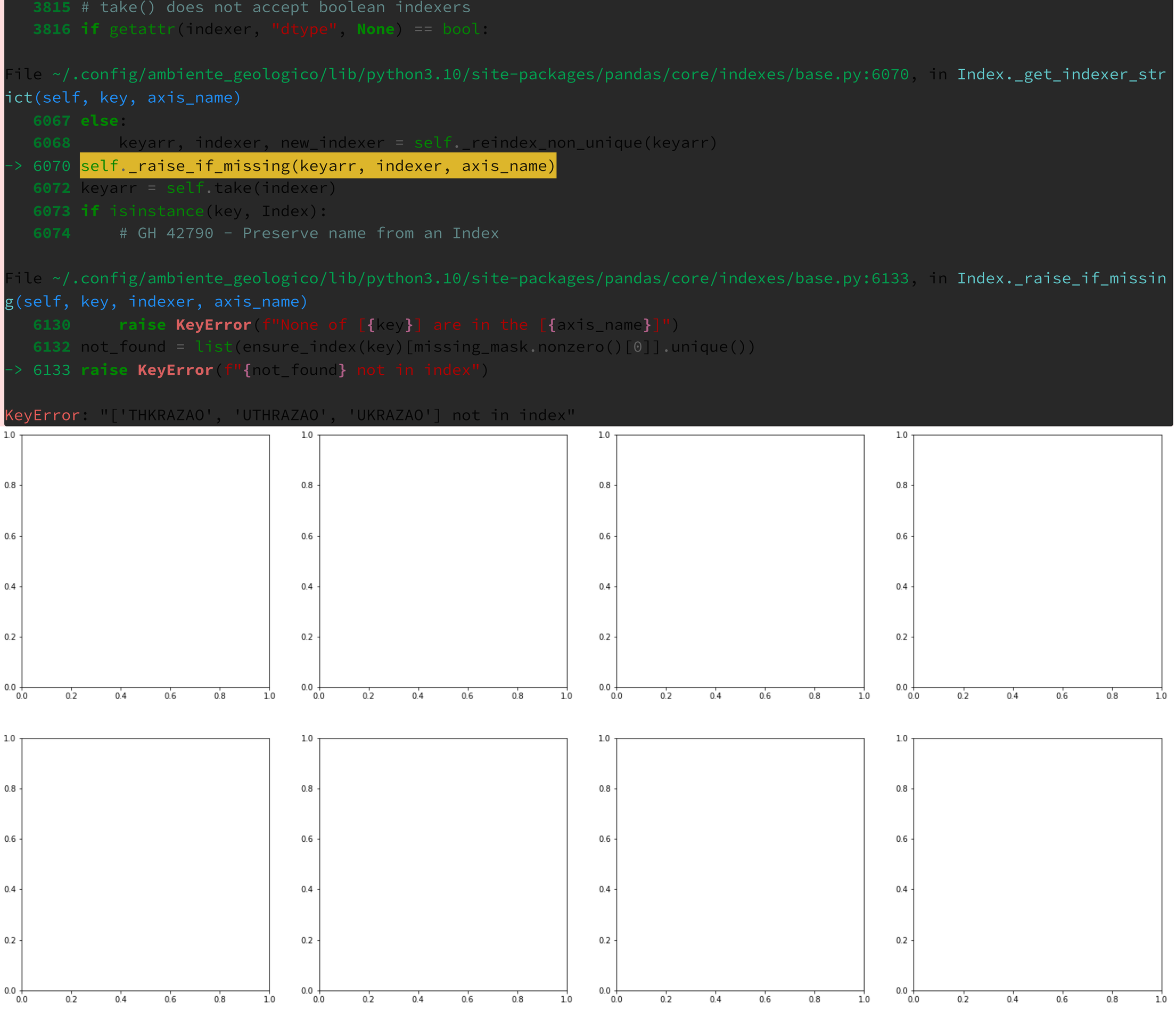


In [13]:
```
plot_raw_gama_data(gama_1105,suptitle='DADOS AEROGAMAESPECTROMÉTRICO')
```



In [14]:
```
gama_1039.rename(columns={'CTC':'CTCOR','KC':'KPERC','UC':'eU','THC':'eTh'},inplace=True)
plot_histograms(gama_1039)
plot_raw_gama_data(gama_1039,suptitle='Dados Radiométricos brutos (XYZ)',orientation='vertical')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Input In [14], in <cell line: 2>()
      1 gama_1039.rename(columns={'CTC':'CTCOR','KC':'KPERC','UC':'eU','THC':'eTh'},inplace=True)
----> 2 plot_histograms(gama_1039)
      3 plot_raw_gama_data(gama_1039,suptitle='Dados Radiométricos brutos (XYZ)',orientation='vertical')

File ~/projetos/geologist/src.py:922, in plot_histograms(dataframe, bins, suptitle)
    920 def plot_histograms(dataframe=None,bins=500,suptitle='Distribuição de Contagens Radiométricas'):
    921     fig, axs = plt.subplots(nrows = 2, ncols = 4, figsize = (26, 13))
--> 922     print(dataframe[['CTCOR','eTh','eU','KPERC','MDT','THKRAZAO','UTHRAZAO','UKRAZAO']].describe(percentiles).T)
    923     for ax, f in zip(axs.flat, gama_FEAT):
    924         _,_,bars = ax.hist(dataframe[f],color='blue',bins=bins)

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/frame.py:3813, in DataFrame.__getitem__(self, key)
   3811     if is_iterator(key):
   3812         key = list(key)
-> 3813     indexer = self.columns._get_indexer_strict(key, "columns")[1]
   3815 # take() does not accept boolean indexers
   3816 if getattr(indexer, "dtype", None) == bool:

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/indexes/base.py:6070, in Index._get_indexer_strict(self, key, axis_name)
   6067 else:
   6068     keyarr, indexer, new_indexer = self._reindex_non_unique(keyarr)
-> 6070 self._raise_if_missing(keyarr, indexer, axis_name)
   6072 keyarr = self.take(indexer)
   6073 if isinstance(key, Index):
   6074     # GH 42790 - Preserve name from an Index

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/indexes/base.py:6133, in Index._raise_if_missing(self, key, indexer, axis_name)
   6130     raise KeyError(f"None of [{key}] are in the [{axis_name}]")
   6132 not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 6133 raise KeyError(f"{not_found} not in index")

KeyError: "['THKRAZAO', 'UTHRAZAO', 'UKRAZAO'] not in index"
```

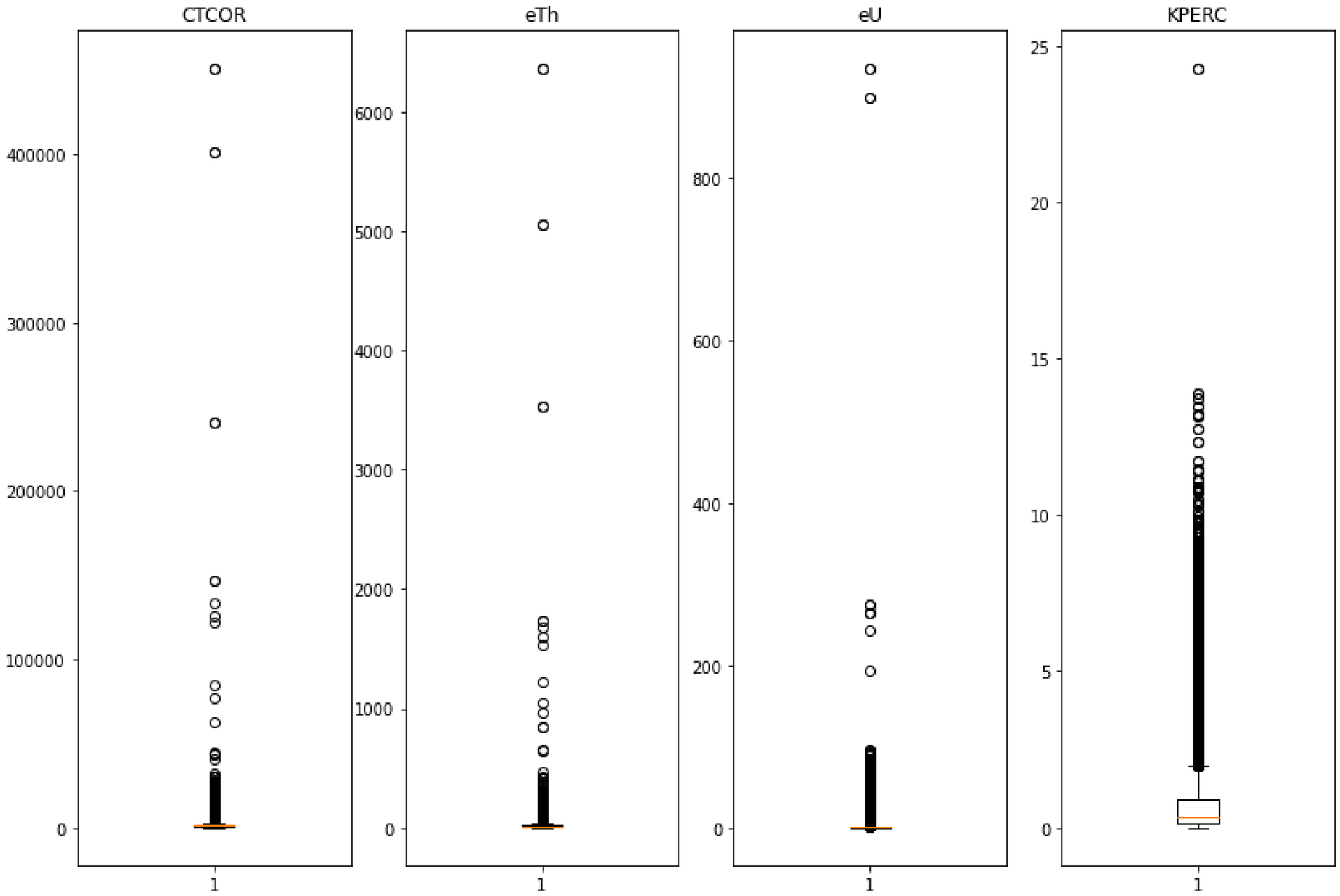


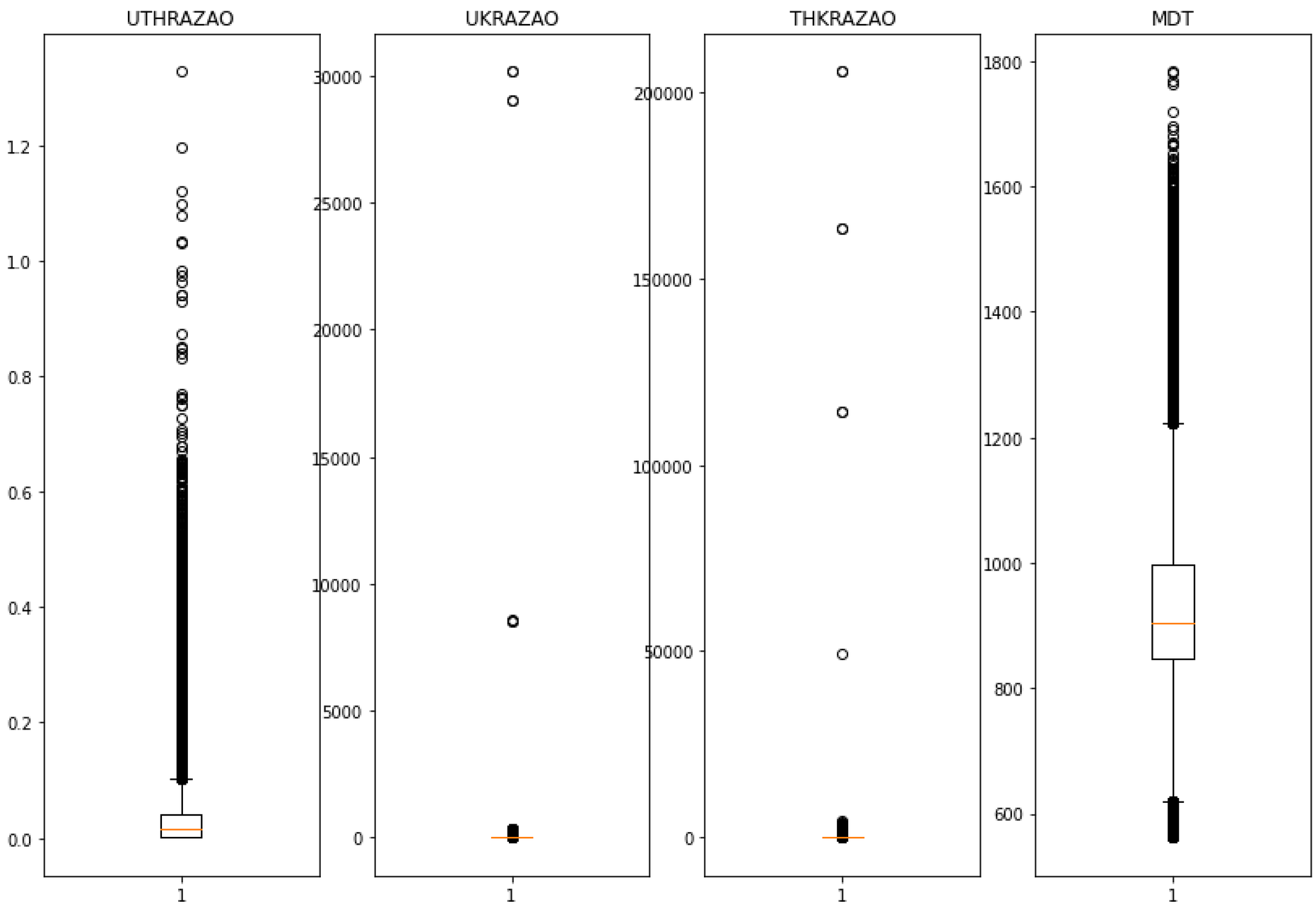
Removendo valores negativos das contagens radiométricas

In [15]:
```
gama_3022_positive = remove_negative_values(gama_3022)
```

```
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
```

In [16]:
```
plot_boxplots(gama_3022_positive,gama_FEAT)
```

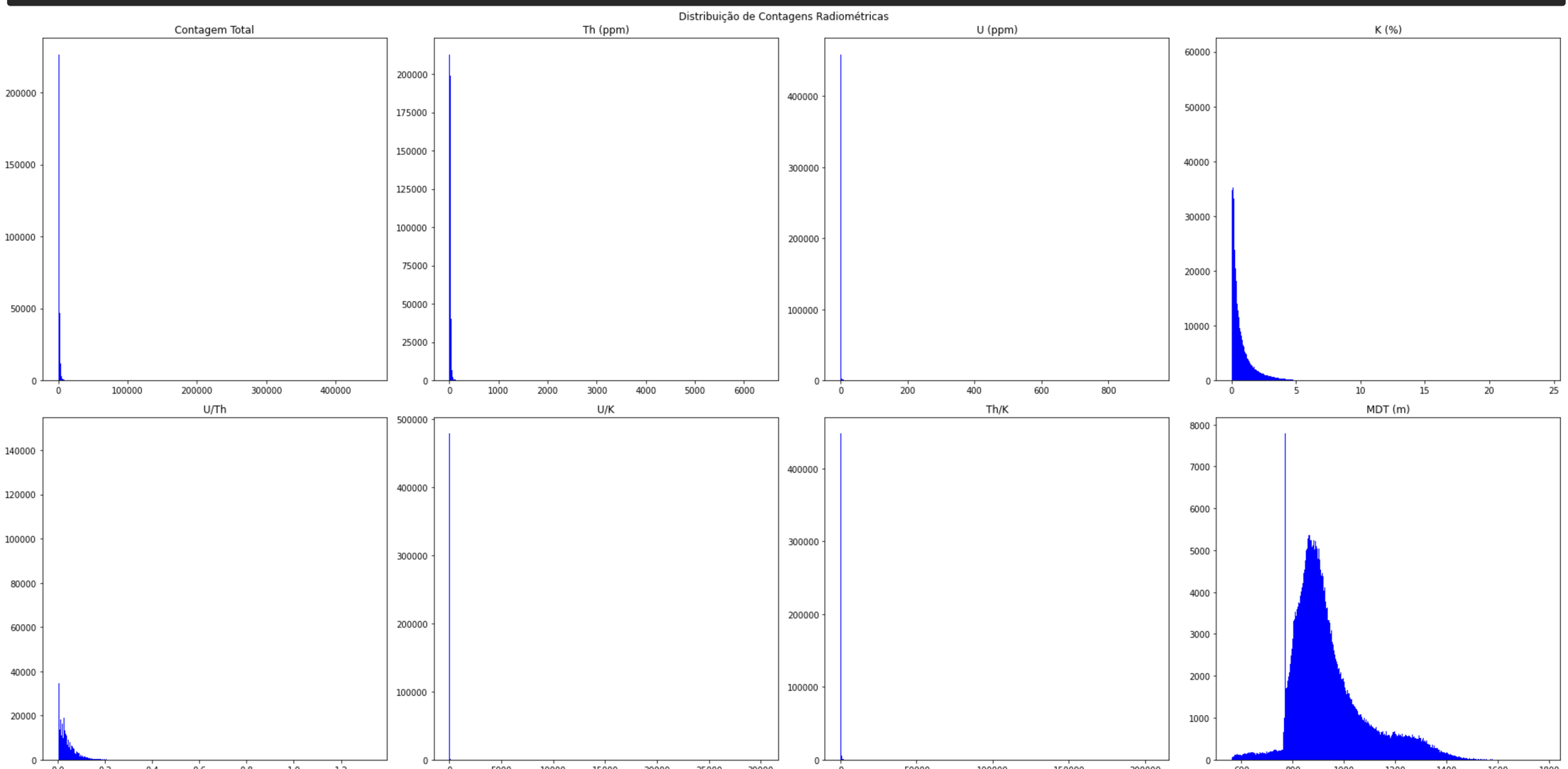




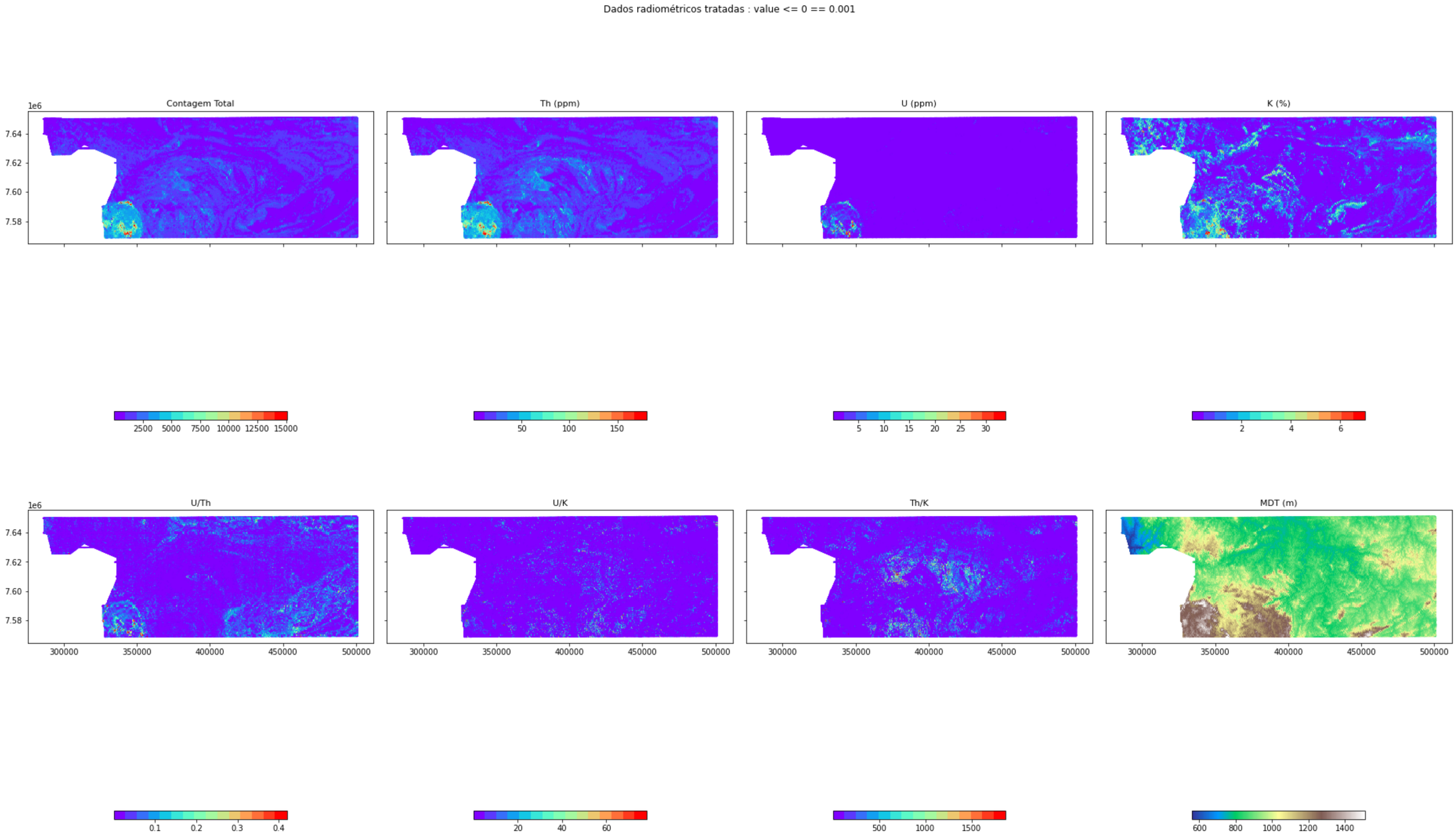
In [17]:
```
plot_histograms(gama_3022_positive)
```

	count	mean	std	min	0.1%	1%	\
CTCOR	479609.0	1292.510550	1777.613302	0.001	0.001	0.001	
eTh	479609.0	16.963804	23.742188	0.001	0.001	0.700	
eU	479609.0	0.559569	3.335543	0.001	0.001	0.001	
KPERC	479609.0	0.679807	0.867536	0.001	0.001	0.001	
MDT	479609.0	940.063014	146.910103	562.040	577.030	650.750	
THKRAZAO	479609.0	108.895627	619.252370	0.155	0.769	2.301	
UTHRAZAO	479609.0	0.029154	0.038412	0.001	0.001	0.001	
UKRAZAO	479609.0	2.947827	88.325810	0.002	0.004	0.006	

	5%	25%	50%	75%	99.95%	max
CTCOR	345.8280	729.320	1079.270	1511.140	15080.726160	450895.020
eTh	4.4000	9.200	13.800	19.900	180.100000	6371.400
eU	0.0010	0.001	0.200	0.600	33.900000	934.800
KPERC	0.0010	0.133	0.354	0.881	7.021392	24.286
MDT	772.2000	845.950	903.290	996.270	1510.670000	1782.280
THKRAZAO	5.3184	15.517	36.123	99.831	1870.690336	205529.094
UTHRAZAO	0.0010	0.002	0.016	0.042	0.419588	1.328
UKRAZAO	0.0130	0.094	0.593	2.155	78.128912	30156.090



In [18]:
```
plot_raw_gama_data(gama_3022_positive,'Dados radiométricos tratadas : value <= 0 == 0.001')
```

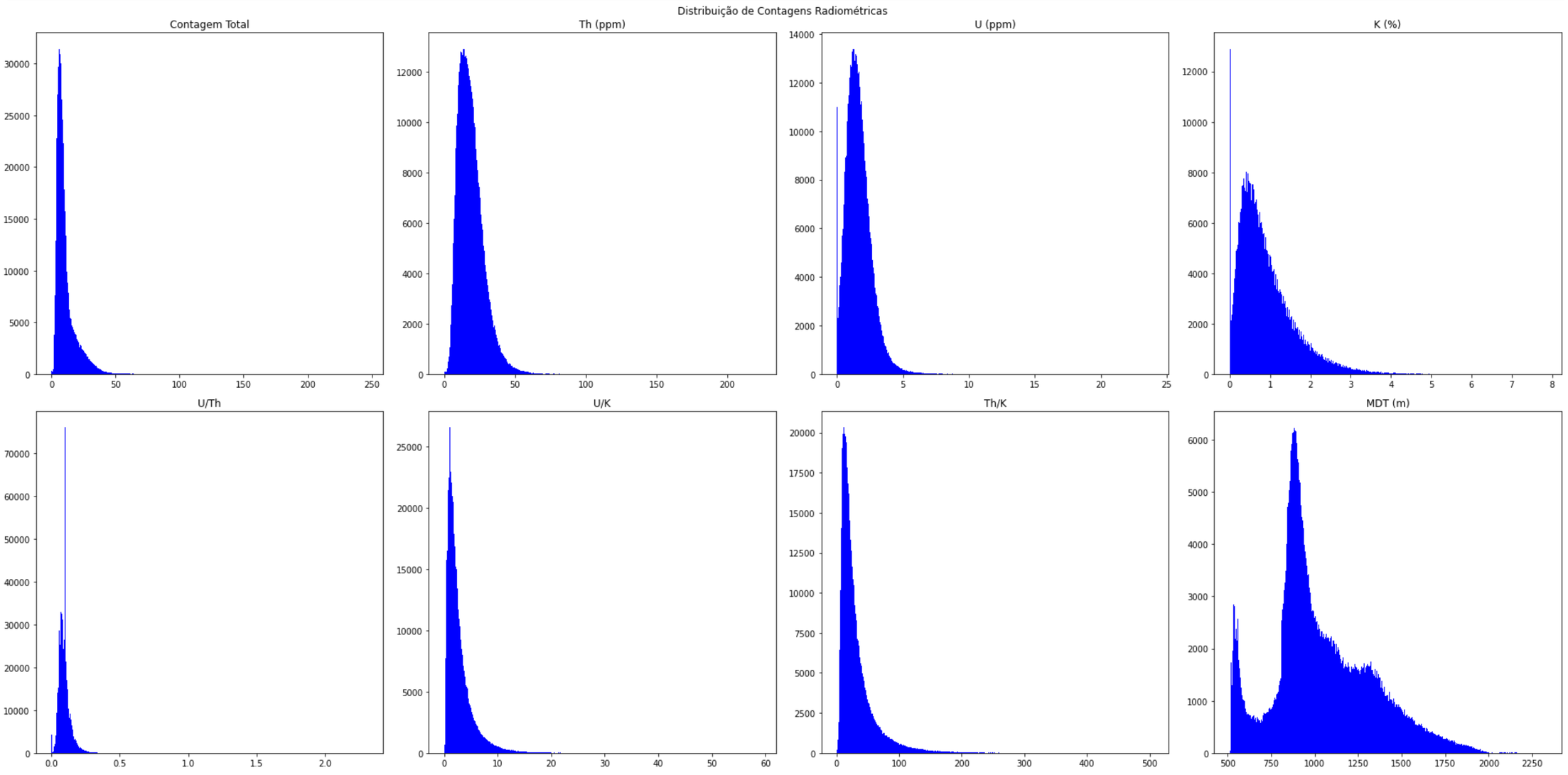


In [19]:
```
gama_1105_positive = remove_negative_values(gama_1105)
```

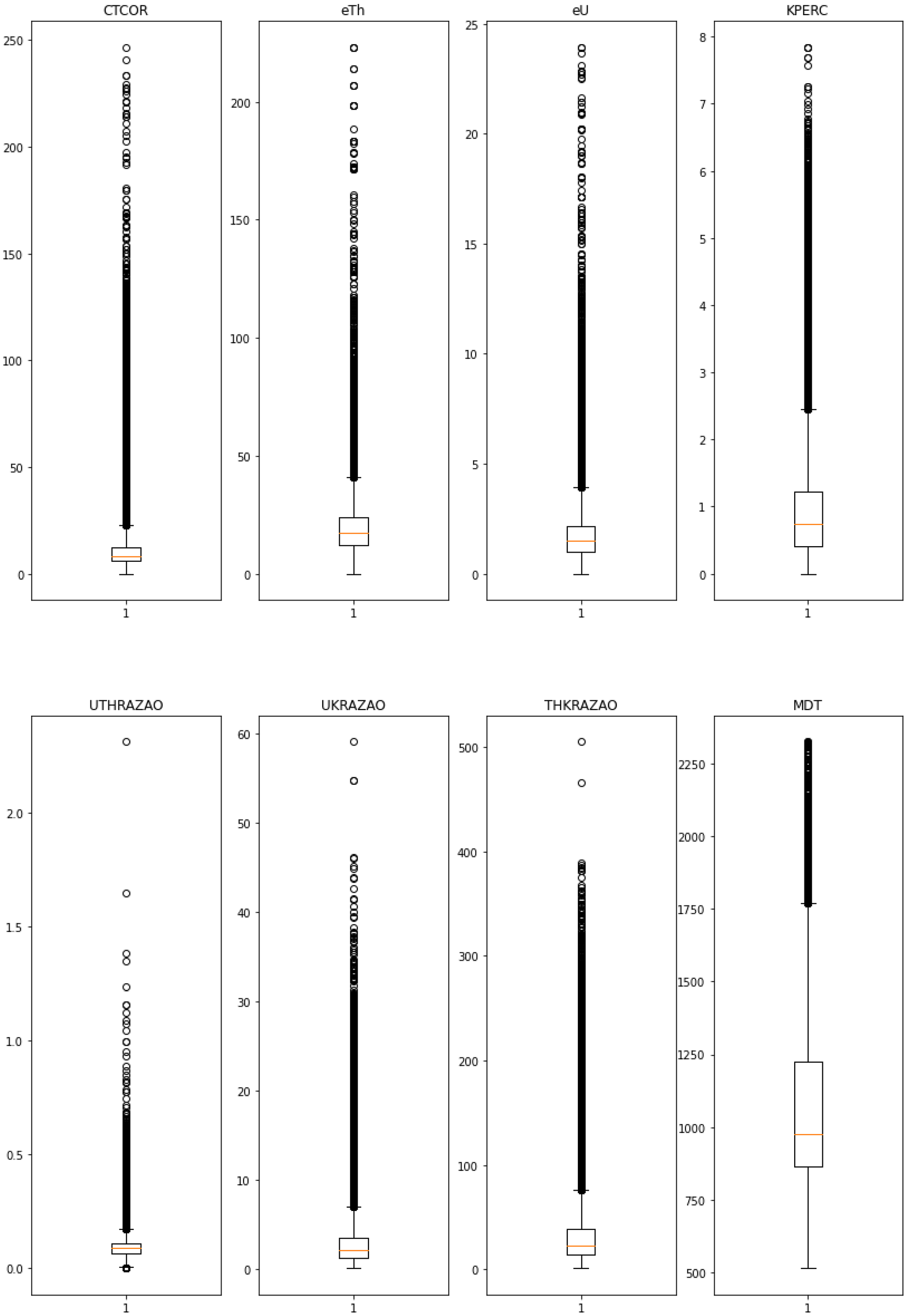
```
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
```

In [20]:
```
plot_histograms(gama_1105_positive)
```

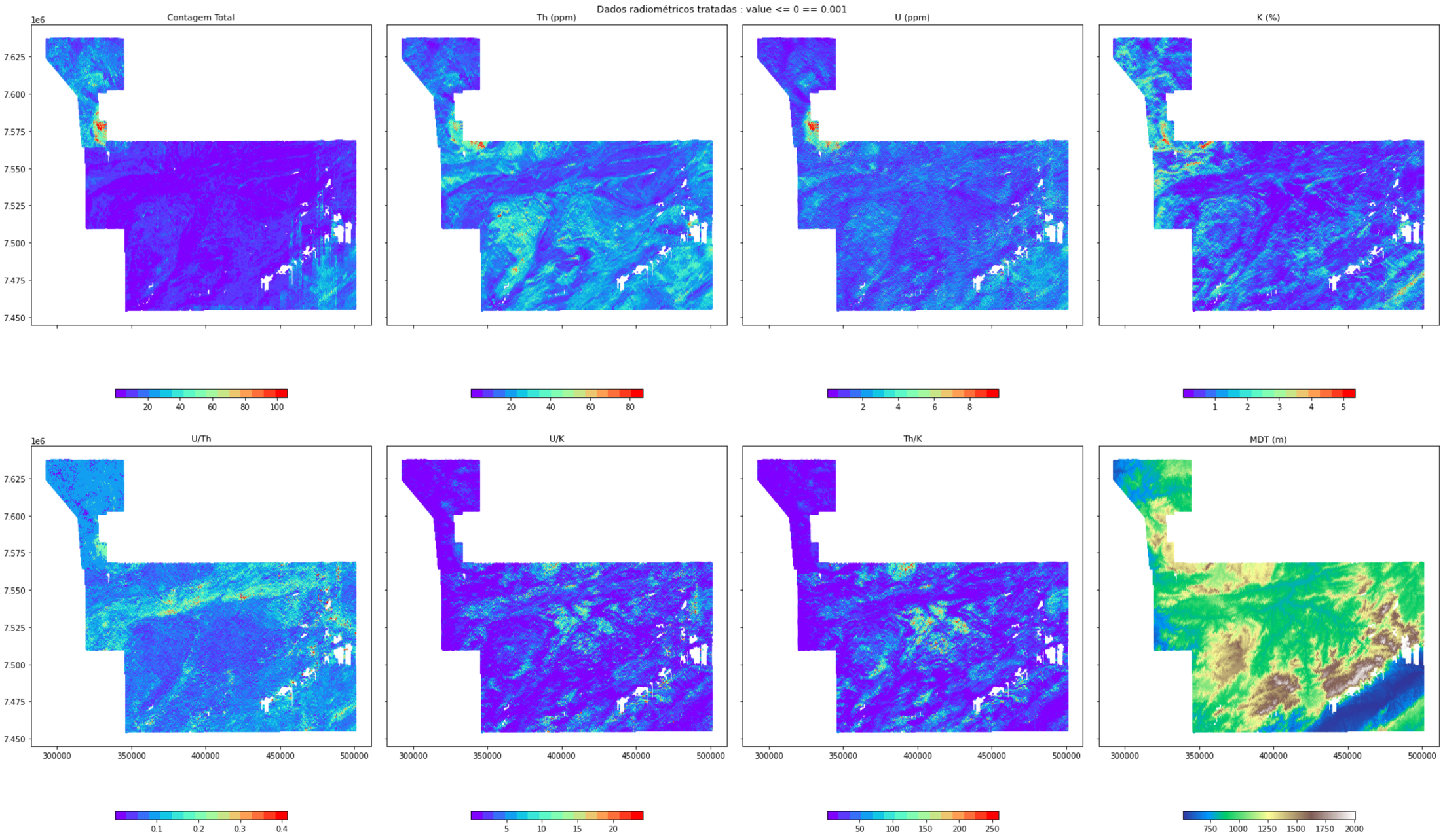
```
             count         mean          std      min       0.1%         1%  \
CTCOR     588216.0    10.961682     9.083210    0.001    1.00886    2.43300
eTh       588216.0    18.883351     9.484284    0.001    2.19043    4.75915
eU        588216.0     1.649159     1.014771    0.001    0.00100    0.00100
KPERC     588216.0     0.913368     0.714298    0.001    0.00100    0.00100
MDT       588216.0  1040.922448   287.145718  513.280  521.06000  532.31000
THKRAZAO  588216.0    31.855617    29.509486    1.079    2.61500    4.94500
UTHRAZAO  588216.0     0.094988     0.043054    0.001    0.00100    0.02100
UKRAZAO   588216.0     2.798359     2.571587    0.046    0.15900    0.30000

                 5%      25%      50%       75%       99.95%       max
CTCOR       3.53500    5.788    8.221    12.500   106.078500   246.600
eTh         7.20775   12.133   17.330    23.679    86.530710   223.136
eU          0.31700    0.972    1.514     2.153     9.640000    23.934
KPERC       0.11500    0.412    0.735     1.227     5.360000     7.840
MDT       568.60750  864.500  973.390  1226.570  2004.160975  2326.220
THKRAZAO    7.61000   13.825   22.353    38.557   259.671625   505.144
UTHRAZAO    0.04300    0.067    0.091     0.109     0.412000     2.311
UKRAZAO     0.53800    1.186    2.018     3.500    24.198140    59.007
```



In [21]:
```
plot_boxplots(gama_1105_positive,gama_FEAT)
```

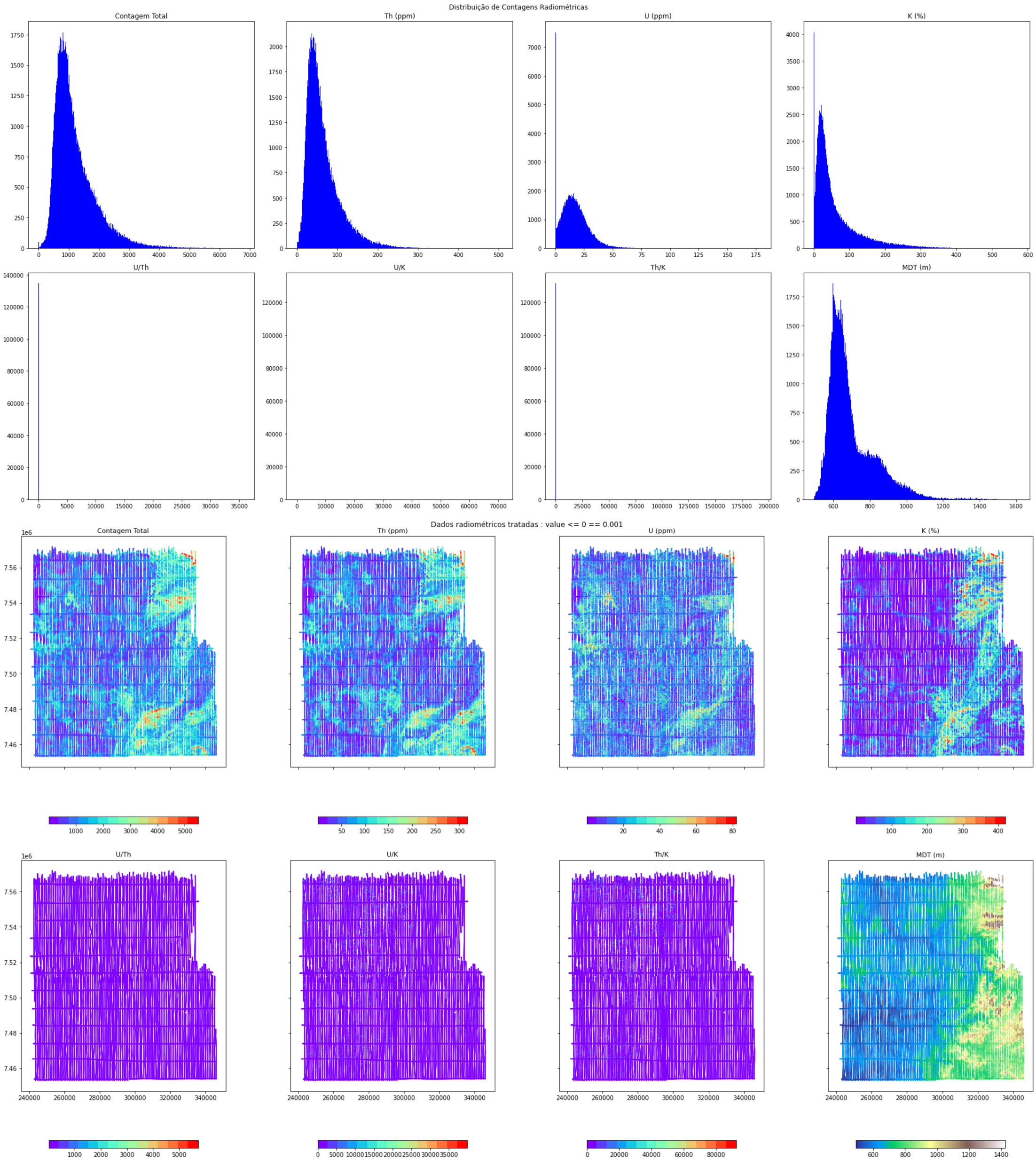


In [22]:
```
plot_raw_gama_data(gama_1105_positive,'Dados radiométricos tratadas : value <= 0 == 0.001')
```



In [23]:
```
gama_1039_positive=remove_negative_values(gama_1039,lista=['X','Y','LATITUDE','LONGITUDE','geometry'])
gama_1039_positive['UTHRAZAO']=gama_1039_positive['eU']/gama_1039_positive['eTh']
gama_1039_positive['UKRAZAO']=gama_1039_positive['eU']/gama_1039_positive['KPERC']
gama_1039_positive['THKRAZAO']=gama_1039_positive['eTh']/gama_1039_positive['KPERC']

plot_histograms(gama_1039_positive)
plot_raw_gama_data(gama_1039_positive,'Dados radiométricos tratadas : value <= 0 == 0.001')
```

```
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - CTCOR
Atributo - MDT
             count         mean          std         min        0.1%  \
CTCOR     134733.0  1216.764120   691.317857    0.001000   97.452720
eTh       134733.0    65.037710    42.215629    0.001000    0.001000
eU        134733.0    16.963088    11.558419    0.001000    0.001000
KPERC     134733.0    60.768322    64.563670    0.001000    0.001000
MDT       134733.0   698.099053   127.998577  494.500000  503.938115
THKRAZAO  134733.0   973.557966  6951.973783    0.000005    0.028435
UTHRAZAO  134733.0     8.189802   331.917130    0.000002    0.000005
UKRAZAO   134733.0   420.775703  3089.246304    0.000002    0.000004

                 1%          5%         25%          50%          75%  \
CTCOR     306.930000  472.510000  741.600000  1025.910000  1514.700000
eTh         9.913200   19.680000   36.100000    53.680000    82.230000
eU          0.001000    0.001000    8.800000    15.600000    23.310000
KPERC       0.001000    4.090000   18.980000    36.790000    79.260000
MDT       531.500000  561.902905  609.575562   657.420715   757.302795
THKRAZAO    0.169906    0.297134    0.708279     1.485831     2.853717
UTHRAZAO    0.000009    0.000063    0.147159     0.279611     0.460572
UKRAZAO     0.000009    0.000153    0.146893     0.368850     0.851773

                99.95%           max
CTCOR      5488.819380    6797.270000
eTh         316.627140     509.050000
eU           82.101440     179.140000
KPERC       420.236940     576.170000
MDT        1425.166084    1616.607056
THKRAZAO  92782.160000  192660.000000
UTHRAZAO   5725.560000   35840.000000
UKRAZAO   39922.680000   71460.000000
```



CONSTRUINDO UM GRID SINTÉTICO

In [24]:

```
# REMOVING PART OF THE SINTETIC GRID
'''
df_xu_yu = pd.DataFrame(np.array([xu,yu]))
df_xu_yu=df_xu_yu.T
df_xu_yu.rename(columns={0:'xu',1:'yu'},inplace=True)

df_xu_yu[(df_xu_yu.xu < 540937) & (df_xu_yu.yu > 8866937)]

df_xu_yu.drop(df_xu_yu[(df_xu_yu.xu < 540937) & (df_xu_yu.yu > 8866937)].index,inplace=True)
plt.figure(figsize=(18,12))

plt.scatter(df_xu_yu.xu,df_xu_yu.yu,s=0.1,marker='.')
plt.axis('scaled')
'''
```

Out[24]:
```
"\ndf_xu_yu = pd.DataFrame(np.array([xu,yu]))\ndf_xu_yu=df_xu_yu.T\ndf_xu_yu.rename(columns={0:'xu',1:'yu'},inplace=True)\n\ndf_xu_yu[(df_xu_yu.xu < 540937) & (df_xu_yu.yu > 8866937)]\n\ndf_xu_yu.drop(df_xu_yu[(df_xu_yu.xu < 540937) & (df_xu_yu.yu > 8866937)].index,inplace=True)\nplt.figure(figsize=(18,12))\n\nplt.scatter(df_xu_yu.xu,df_xu_yu.yu,s=0.1,marker='.')\nplt.axis('scaled')\n"
```

Interpolação dos dados Brutos

Método Cúbico

In [ ]:
```
# Test de área


# area=(344093.45426573796, 396417.36691108724, 7621768.799495494, 7677527.304557458)
# int((area[3]-area[2])/100),int((area[1]-area[0])/100)
```

In [ ]:
```
#traditional_interpolation(quadricula,'mag_3022','gama_3022','cubic','geof_3022')
```

In [ ]:
```
#list(quadricula['SF23_VC'].keys())
```

In [ ]:
```
#df = quadricula['SF23_VC']['geof_3022_cubic']
#plt.figure(figsize=(12,12))
#plt.scatter(x=df.X,y=df.Y,c=df.GMT,cmap='rainbow')
#plt.axis('scaled')
```

In [ ]:
```
# Print the output. a=


#descript_cubic = df.describe(percentiles)
#descript_cubic[['eU','eTh','KPERC','CTCOR','UTHRAZAO','THKRAZAO','UKRAZAO']].T
```

In [ ]:
```
#plot_histograms(geof_1089_cubic,suptitle='Distribuição dos dados radiométricos interpolados (cúbico, pixel 100m)')
#plot_raw_data(geof_1089_cubic,suptitle='Dados radiométricos interpolados (cúbico, pixel 100m)')
```

Método Nearest

In [ ]:
```
#plot_histograms(geof_1089_nearest,suptitle='Distribuição dos dados radiométricos interpolados (nearest, pixel 100m)')
```

In [ ]:
```
#plot_raw_data(geof_1089_nearest,suptitle='Dados radiométricos interpolados (nearest, pixel 100m)')
```

Método Linear

In [25]:
```
traditional_interpolation(quadricula,'mag_3022','gama_3022','linear','geof_3022')
```

```
 - Folha: SF23_VC_II
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VC_III
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VC_VI
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VD_I
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VD_IV
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VD_II
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VD_V
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
```

```
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VD_III
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
 - Folha: SF23_VD_VI
Atributo - ALTURA
Atributo - MAGIGRF
Atributo - MDT
Atributo - CTCOR
Atributo - eTh
Atributo - eU
Atributo - KPERC
Atributo - MDT
Atributo - THKRAZAO
Atributo - UKRAZAO
Atributo - UTHRAZAO
Index(['ALTURA', 'LATITUDE', 'LONGITUDE', 'MAGIGRF', 'MDT', 'X', 'Y'], dtype='object')
```

In [26]:

```
traditional_interpolation(quadricula,'mag_line_1105','gama_line_1105','linear','geof_1105')
```

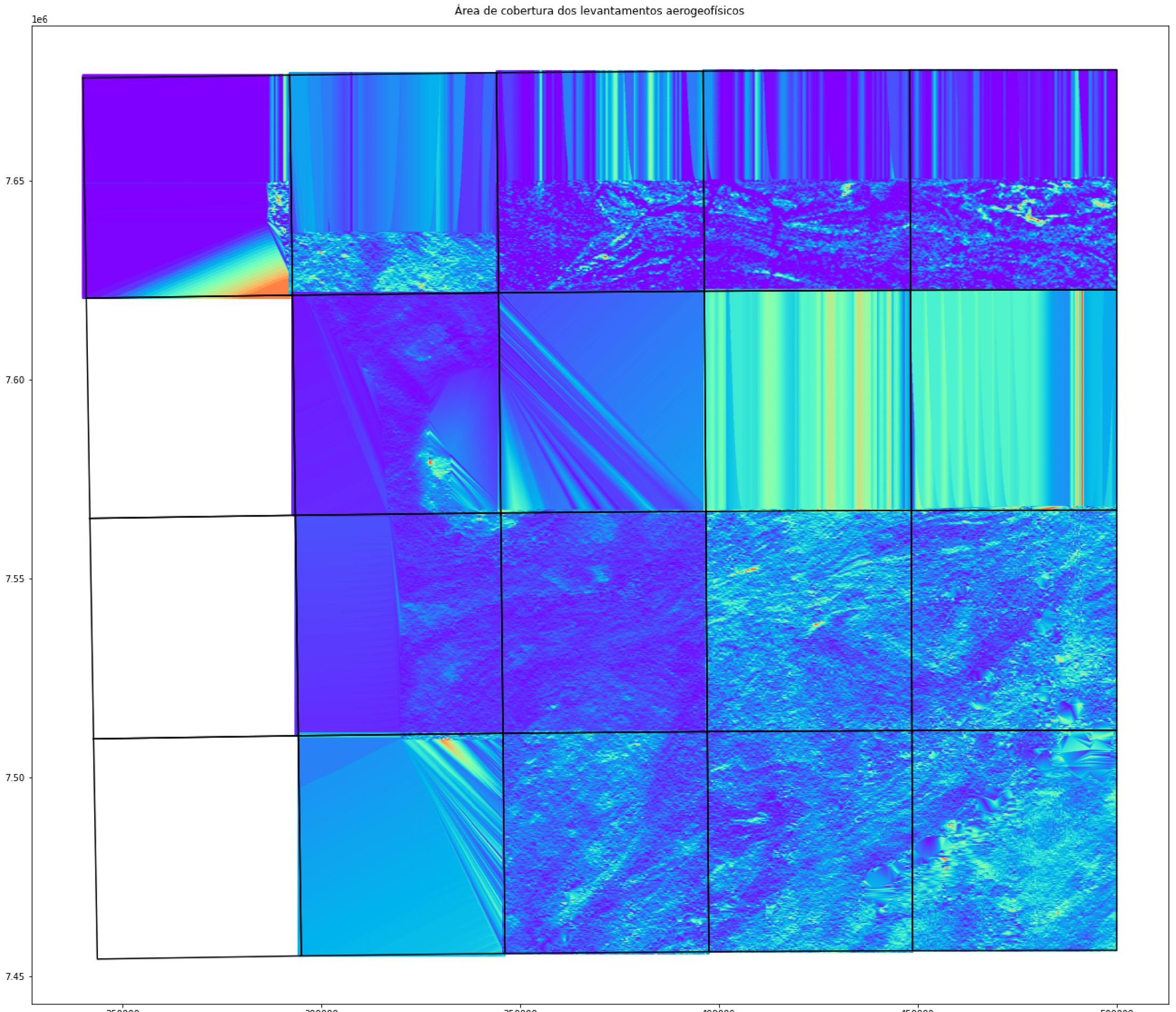
```
 - Folha: SF23_YA_III
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_YA_VI
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_YB_I
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_YB_IV
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_YB_II
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_YB_V
Atributo - ALTURA
Atributo - MDT
```

```
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_YB_III
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_YB_VI
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_VC_III
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_VC_VI
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_VD_IV
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
```

```
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_VD_V
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
 - Folha: SF23_VD_VI
Atributo - ALTURA
Atributo - MDT
Atributo - ALTURA_1
Atributo - MAGIGRF
Atributo - KPERC
Atributo - eU
Atributo - eTh
Atributo - UTHRAZAO
Atributo - UKRAZAO
Atributo - MDT
Atributo - THKRAZAO
Atributo - CTCOR
Index(['ALTURA', 'X', 'Y', 'MDT', 'LATITUDE', 'LONGITUDE', 'ALTURA_1',
       'MAGIGRF'],
      dtype='object')
```

In [27]:

```
plt.figure(figsize=(24,16))

# PLOTANDO A MALHA CARTOGRÁFICA
for id in list(quadricula.keys()):
    carta=quadricula[id]
    plt.plot(*transform_to_carta_utm(carta['folha']).exterior.xy,color='black')

    # PLOTANDO OS DADOS INTERPOLADOS
    for data in list(carta.keys())[2:]:
        if 'geof' in data:
            plt.scatter(carta[data].X,carta[data].Y,c=carta[data].eU,cmap='rainbow',s=0.5,marker='H')
            plt.axis('scaled')
        # SE NÃO TIVER DADOS NÃO PLOTA NADA
        else:
            pass

plt.suptitle('Área de cobertura dos levantamentos aerogeofísicos')
plt.tight_layout()
```



In [ ]:
```
#traditional_interpolation(quadricula,'mag_1039','gama_1039','linear','geof_1039')
```

In [ ]:
```
'''
plt.figure(figsize=(24,16))

for id in list(quadricula.keys()):
    carta=quadricula[id]
    plt.plot(*transform_to_carta_utm(carta['folha']).exterior.xy,color='black')

    for data in list(carta.keys())[2:]:
        if 'geof' in data:
            plt.scatter(carta[data].X,carta[data].Y,c=carta[data].MDT,cmap='terrain',s=0.5,marker='H')
            plt.axis('scaled')
        else:
            pass

plt.suptitle('Área de cobertura dos levantamentos aerogeofísicos')
plt.tight_layout()
'''
```

Classificações Não-Supervisionadas

Self-organizing maps (SOM)

In [28]:
```
import matplotlib
import seaborn as sns
```

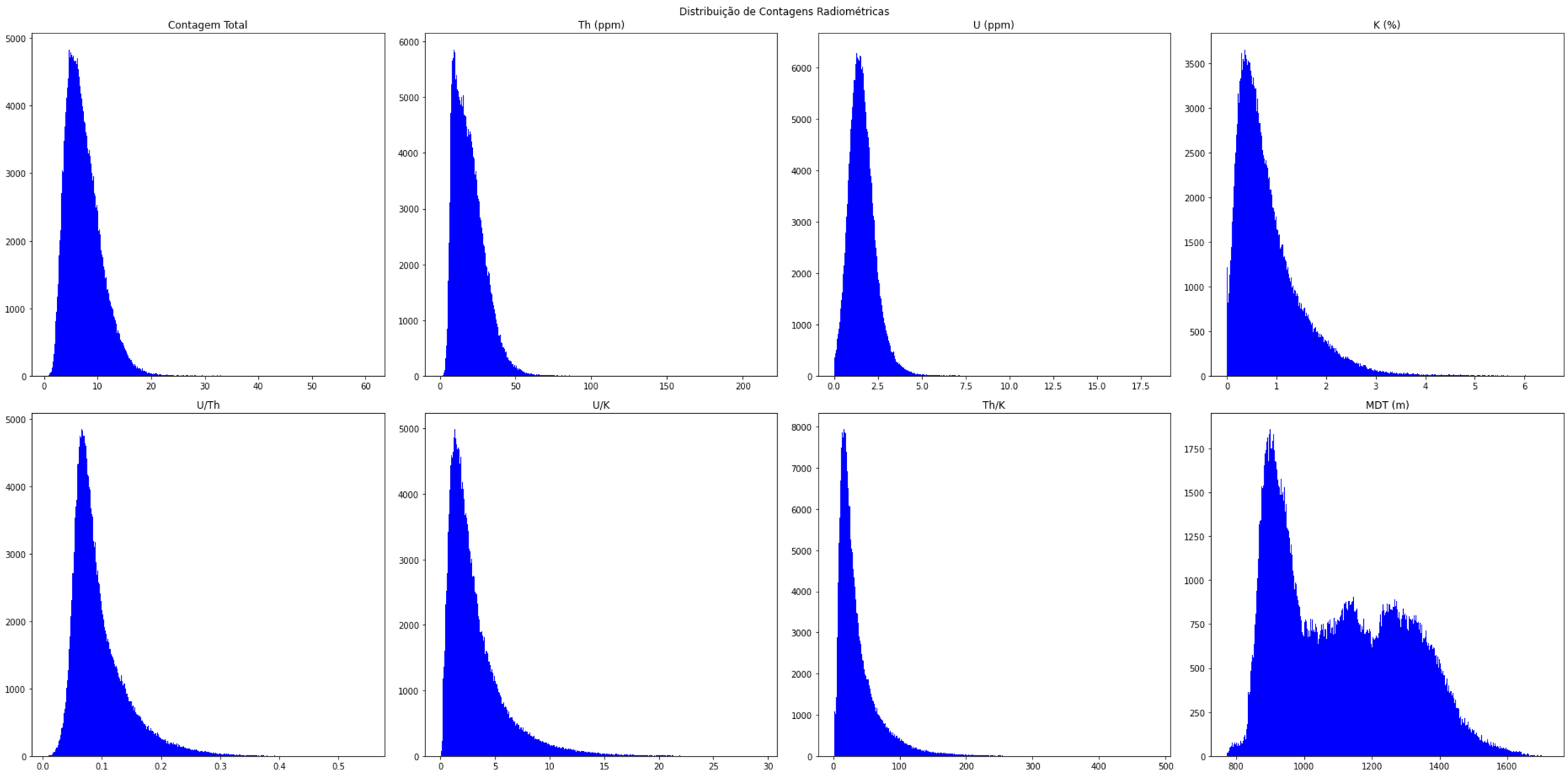
```
from sklearn.preprocessing import StandardScaler
from sklearn_som.som import SOM

def plot_corr(dataframe, size=10):
    plt.figure(figsize = (size+size*0.2, size), facecolor='w')
    corrMatrix = dataframe.corr()
    sns.heatmap(np.round(corrMatrix,2), annot=True)
    plt.xticks(rotation=90, ha='right')
    plt.yticks(rotation=0, ha='right')
```

teste

In [42]:
```
df = quadricula['SF23_YB_I']['geof_1105_linear']

plot_histograms(df)

plot_boxplots(df,FEAT)
```

```
           count         mean          std         min        0.1%  \
CTCOR     289083.0     7.456336     3.469340    0.740521    1.550968
eTh       289083.0    19.792178    10.868278    0.783635    3.535928
eU        289083.0     1.623539     0.794915    0.001000    0.032192
KPERC     289083.0     0.845806     0.680086    0.001000    0.001000
MDT       289083.0  1109.056562   188.220433  772.957847  787.519403
THKRAZAO  289083.0    36.763569    32.065770    1.422133    1.995555
UTHRAZAO  289083.0     0.095841     0.046902    0.009123    0.021459
UKRAZAO   289083.0     3.201816     2.688475    0.051656    0.165101

                1%         5%         25%         50%         75%  \
CTCOR     2.283863    3.156535    4.968312    6.844628    9.303219
eTh       5.304612    7.118408   11.573633   17.892590   25.607301
eU        0.215587    0.535568    1.096739    1.533234    2.032280
KPERC     0.038815    0.137963    0.377563    0.661917    1.114827
MDT     836.778920  867.950563  934.551848  1089.805541  1264.853947
THKRAZAO  4.374887    8.394507   16.365710   26.412424   45.739390
UTHRAZAO  0.033860    0.046442    0.064512    0.082148    0.114318
UKRAZAO   0.318782    0.633961    1.413830    2.402183    4.081731

              99.95%          max
CTCOR      35.624912    60.484803
eTh       113.246429   212.262973
eU          7.577335    18.262299
KPERC       5.501595     6.467313
MDT      1648.790448  1718.466326
THKRAZAO  259.641402   482.983565
UTHRAZAO    0.360536     0.549974
UKRAZAO    20.549815    29.385300
```

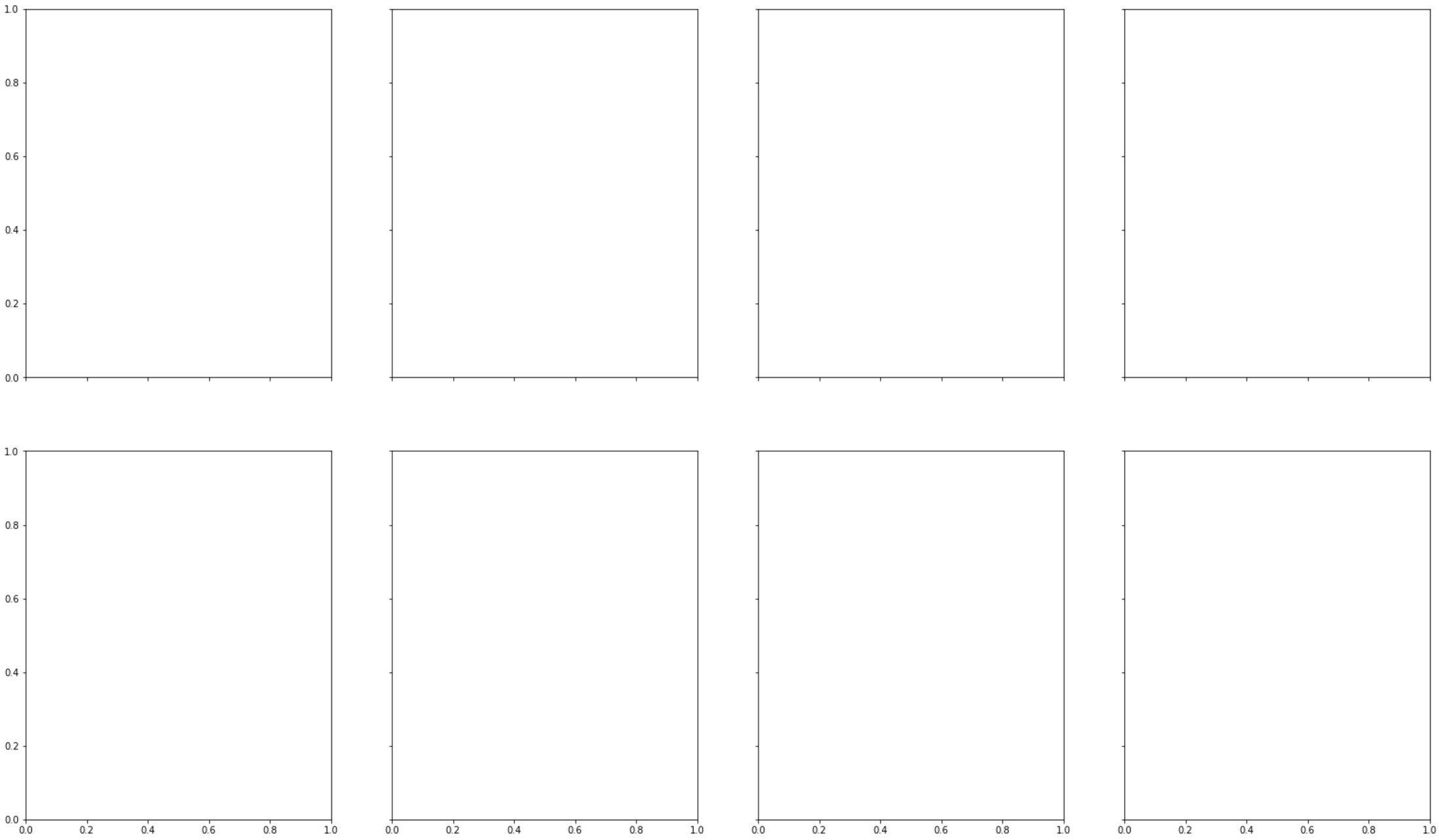
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [42], in <cell line: 5>()
      1 df = quadricula['SF23_YB_I']['geof_1105_linear']
      3 plot_histograms(df)
----> 5 plot_boxplots(df,FEAT)

NameError: name 'FEAT' is not defined
```



In [ ]:
```
df
```

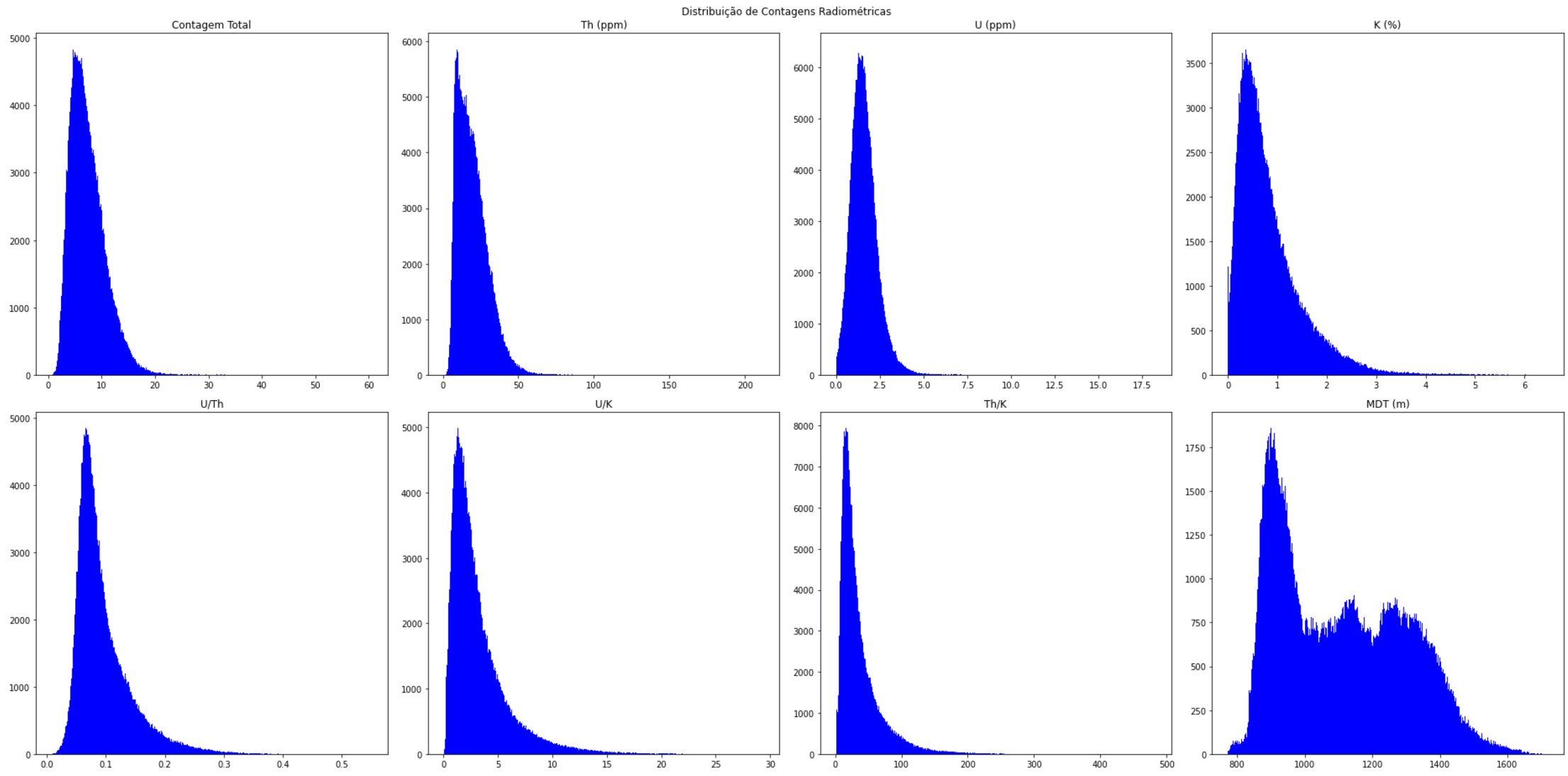
In [45]:
```
plot_raw_gama_data(df,suptitle='Dados Radiométricos interpolados (Algoritmo: Linear)',figsize=(27,16))
plot_raw_mag_data(df,suptitle='Dados Magnetométricos interpolados (Algoritmo: Linear)')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [45], in <cell line: 1>()
----> 1 plot_raw_gama_data(df,suptitle='Dados Radiométricos interpolados (Algoritmo: Linear)',figsize=(27,16))
      2 plot_raw_mag_data(df,suptitle='Dados Magnetométricos interpolados (Algoritmo: Linear)')

File ~/projetos/geologist/src.py:903, in plot_raw_gama_data(raw_data, suptitle, minimo, maximo, orientation, figsize)
    901 fig, axs = plt.subplots(nrows = 2, ncols = 4, figsize =figsize,sharex='all',sharey='all')
    902 raw_data_describe = raw_data.describe(percentiles=percentiles)
--> 903 X, Y = raw_data.X, raw_data.Y
    904 for ax, f in zip(axs.flat, gama_dic_titles):
    905     vmin = raw_data_describe[f][minimo]

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/generic.py:5902, in NDFrame.__getattr__(self, name)
   5895 if (
   5896     name not in self._internal_names_set
   5897     and name not in self._metadata
   5898     and name not in self._accessors
   5899     and self._info_axis._can_hold_identifiers_and_holds_name(name)
   5900 ):
   5901     return self[name]
-> 5902 return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'X'
```



In [47]:
```
df = quadricula['SF23_YB_I']['geof_1105_linear']
plot_histograms(df)
```

```
              count         mean          std         min         0.1%  \
CTCOR      289083.0     7.456336     3.469340    0.740521     1.550968
eTh        289083.0    19.792178    10.868278    0.783635     3.535928
eU         289083.0     1.623539     0.794915    0.001000     0.032192
KPERC      289083.0     0.845806     0.680086    0.001000     0.001000
MDT        289083.0  1109.056562   188.220433  772.957847   787.519403
THKRAZAO   289083.0    36.763569    32.065770    1.422133     1.995555
UTHRAZAO   289083.0     0.095841     0.046902    0.009123     0.021459
UKRAZAO    289083.0     3.201816     2.688475    0.051656     0.165101

                1%          5%         25%         50%          75%  \
CTCOR     2.283863    3.156535    4.968312    6.844628     9.303219
eTh       5.304612    7.118408   11.573633   17.892590    25.607301
eU        0.215587    0.535568    1.096739    1.533234     2.032280
KPERC     0.038815    0.137963    0.377563    0.661917     1.114827
MDT     836.778920  867.950563  934.551848  1089.805541  1264.853947
THKRAZAO  4.374887    8.394507   16.365710   26.412424    45.739390
UTHRAZAO  0.033860    0.046442    0.064512    0.082148     0.114318
UKRAZAO   0.318782    0.633961    1.413830    2.402183     4.081731

              99.95%         max
CTCOR      35.624912   60.484803
eTh       113.246429  212.262973
eU          7.577335   18.262299
KPERC       5.501595    6.467313
MDT      1648.790448  1718.466326
THKRAZAO  259.641402  482.983565
UTHRAZAO    0.360536    0.549974
UKRAZAO    20.549815   29.385300
```

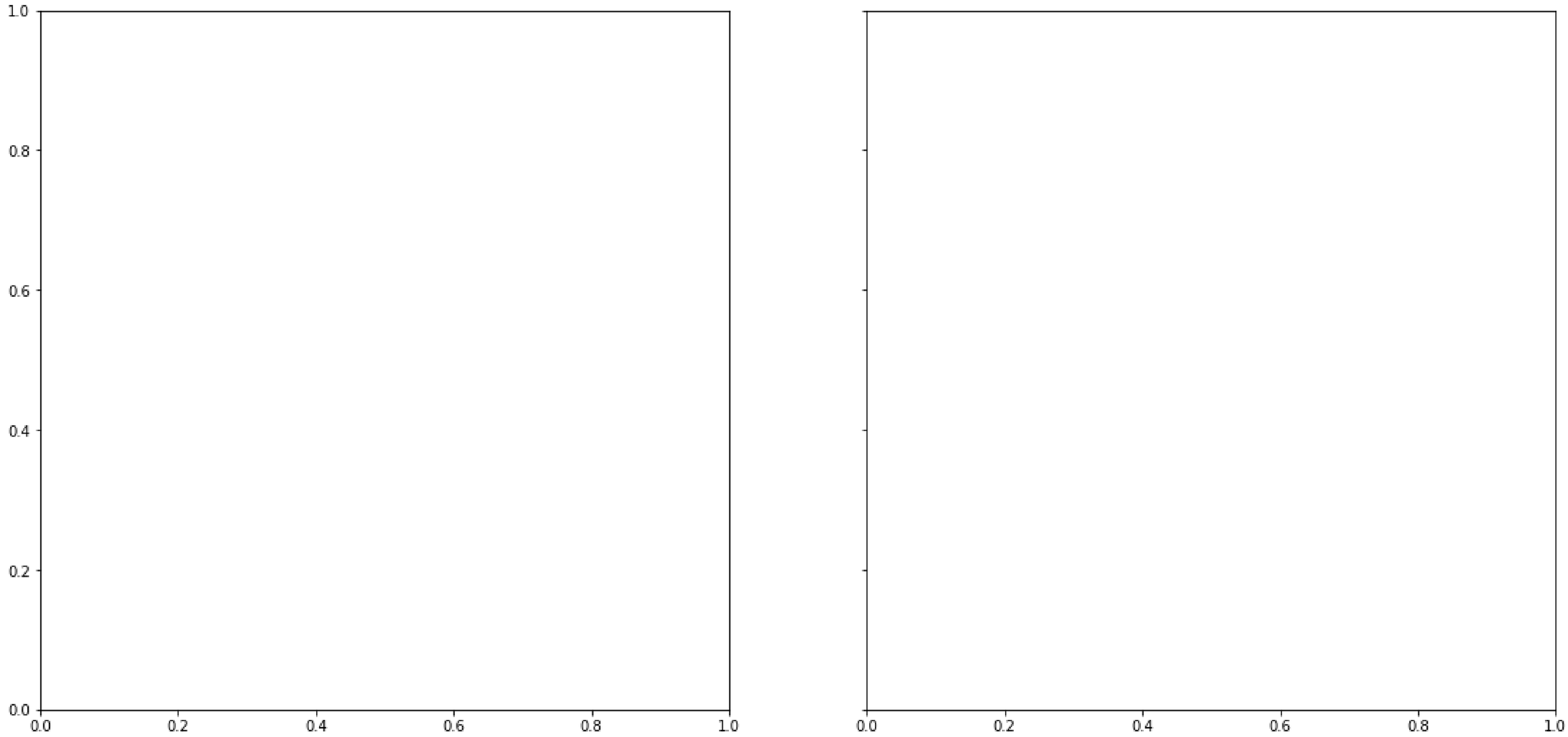


plot_raw_gama_data(df,suptitle='Dados Radiométricos interpolados (Algoritmo: Linear)',figsize=(27,16))

In [ ]:
```
df
```

In [52]:
```
df.rename(columns={'GMT':'MAGIGRF'},inplace=True)
plot_raw_mag_data(df,suptitle='Dados Magnetométricos interpolados (Algoritmo: Linear)')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [52], in <cell line: 2>()
      1 df.rename(columns={'GMT':'MAGIGRF'},inplace=True)
----> 2 plot_raw_mag_data(df,suptitle='Dados Magnetométricos interpolados (Algoritmo: Linear)')

File ~/projetos/geologist/src.py:881, in plot_raw_mag_data(raw_data, suptitle, minimo, maximo)
    879 fig, axs = plt.subplots(nrows = 1, ncols = 2, figsize = (19,9),sharex='all',sharey='all')
    880 raw_data_describe = raw_data.describe(percentiles=percentiles)
--> 881 X, Y = raw_data.X, raw_data.Y
    882 for ax, f in zip(axs.flat, mag_dic_titles):
    883     vmin = raw_data_describe[f][minimo]

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/generic.py:5902, in NDFrame.__getattr__(self, name)
   5895 if (
   5896     name not in self._internal_names_set
   5897     and name not in self._metadata
   5898     and name not in self._accessors
   5899     and self._info_axis._can_hold_identifiers_and_holds_name(name)
   5900 ):
   5901     return self[name]
-> 5902 return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'X'
```

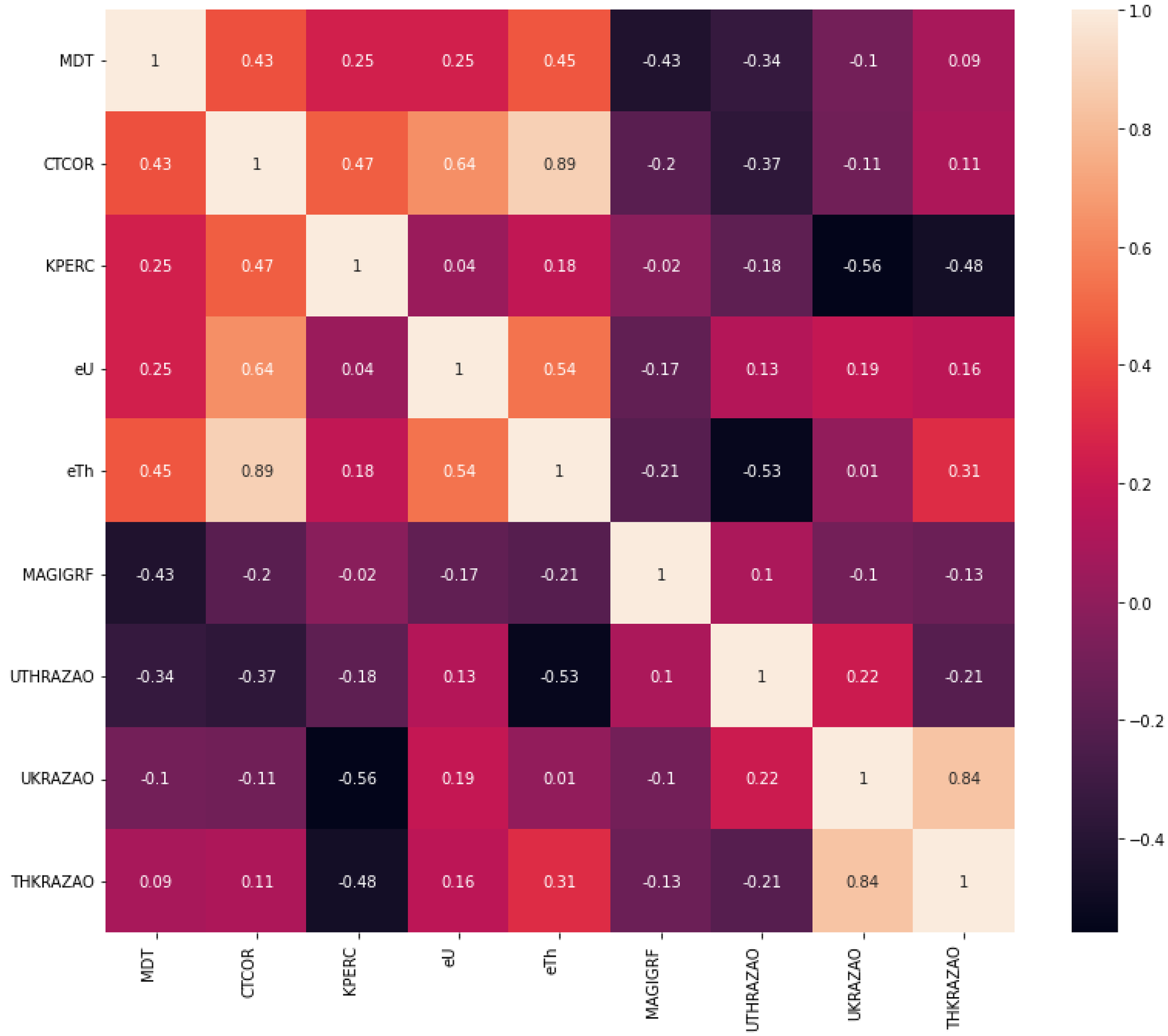


Pixel size

In [49]:
```
df_rs = df
df_rs.rename(columns={'X':'E_utm','Y':'N_utm'},inplace=True)
#df_rs.fillna(0,inplace=True)

xpixel_size = (df_rs.E_utm.max()-df_rs.E_utm.min())/df_rs.E_utm.unique().size
ypixel_size = (df_rs.N_utm.max()-df_rs.N_utm.min())/df_rs.N_utm.unique().size
print('x:', xpixel_size, 'y:', ypixel_size)

nx=df_rs.E_utm.unique().size
ny=df_rs.N_utm.unique().size
ratio=ny/nx
xs = df_rs.E_utm.values.reshape(ny, nx)
ys = df_rs.N_utm.values.reshape(ny, nx)
```

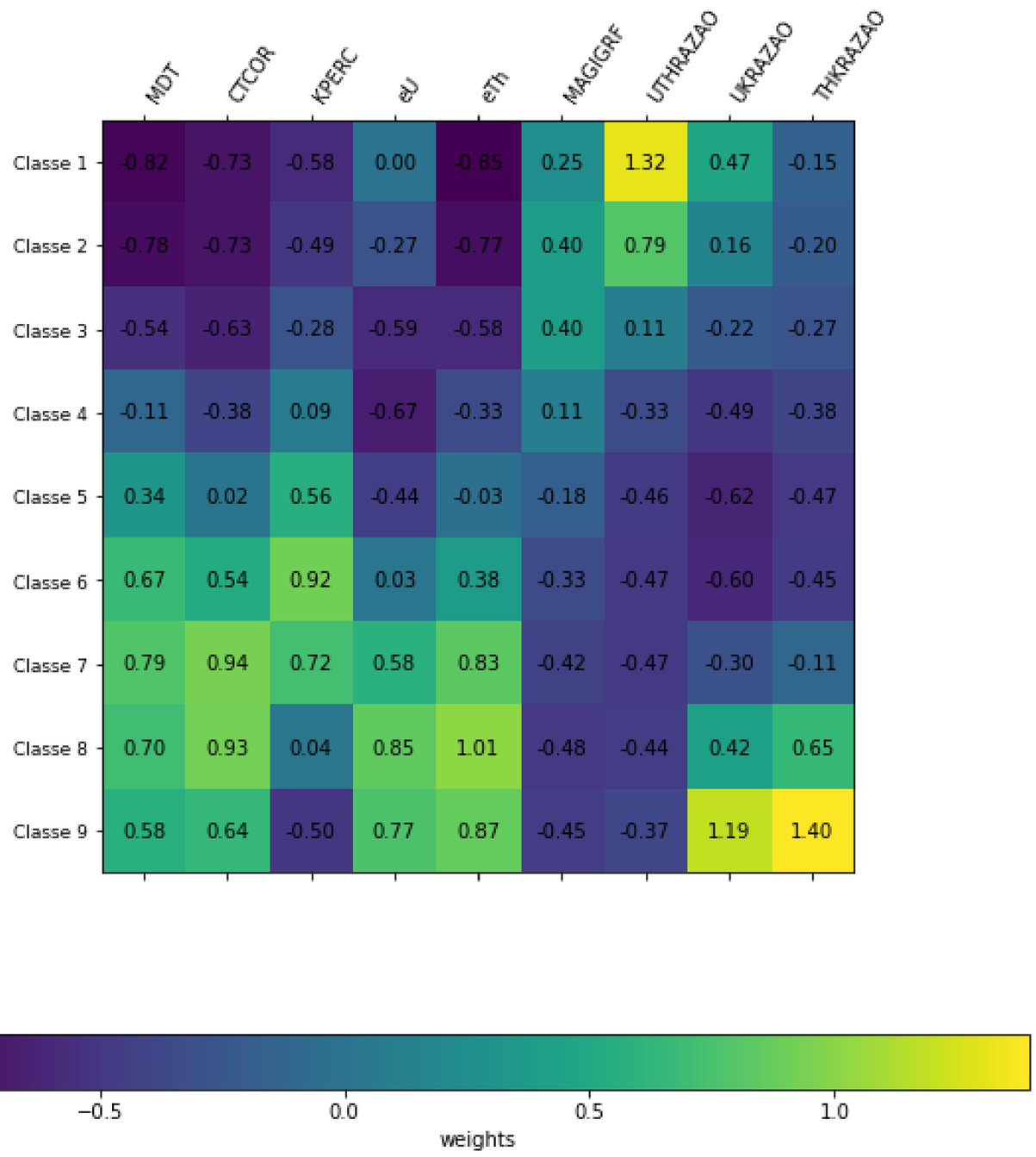
```
x: 100.16429510488899 y: 100.1440749294668
```

In [50]:
```
features = list(df_rs.columns[2:])
plot_corr(df_rs[features], size=11)
#print(features)
#plt.savefig('figs/correlation_matrix.png', dpi=400, bbox_inches='tight')
```



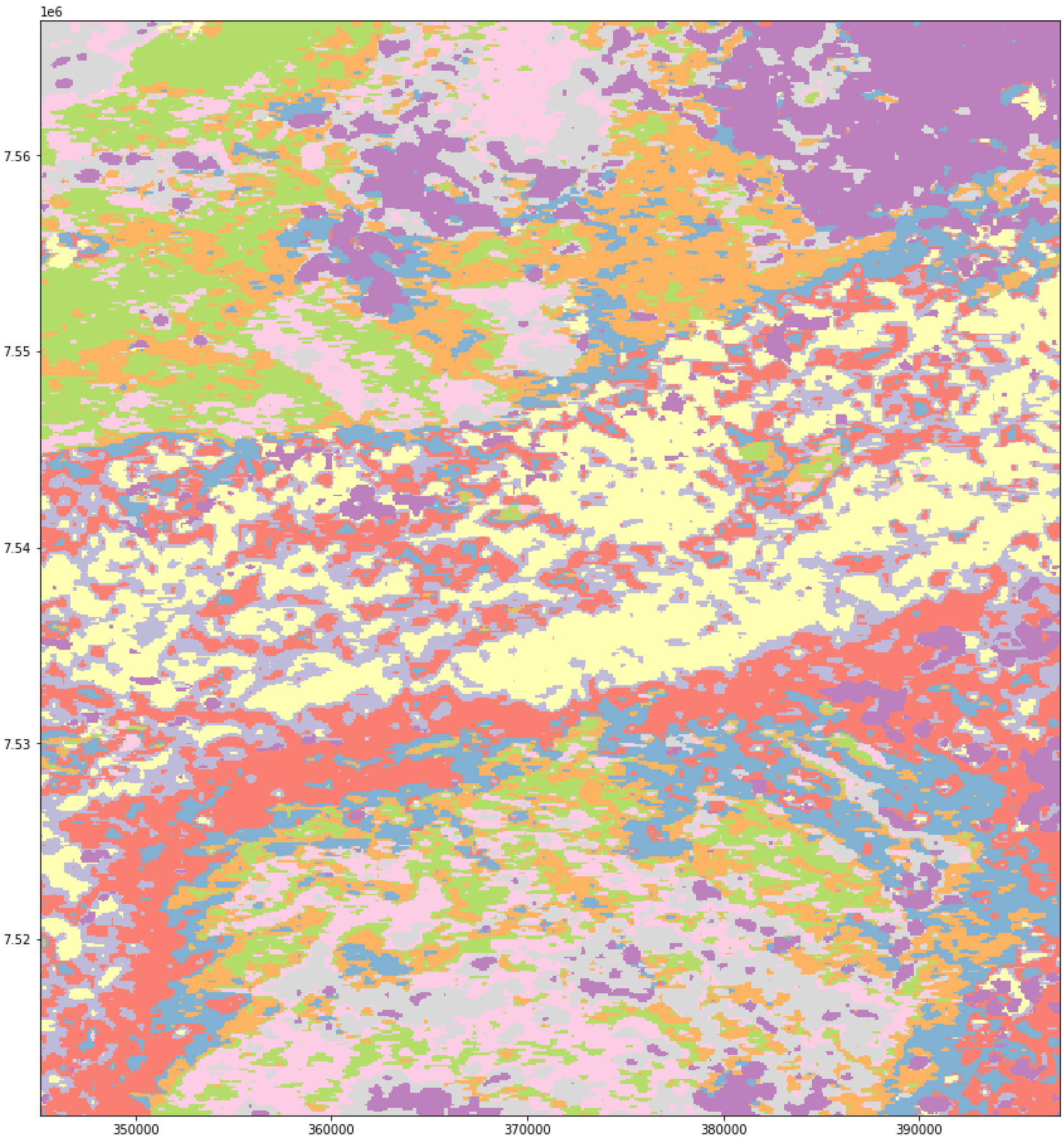
In [51]:

```
data = StandardScaler().fit_transform(df[features].values)

# data = df_rs[features].values

# NÚMERO DE CLASSES
n_clusters=9
lito_SOM = SOM(m=n_clusters,
               n=1,
           sigma=1.5,
             dim=len(features),
         max_iter=10000)

lito_SOM.fit(data)

# predição de classes
predictions = lito_SOM.predict(data)

# create labels
cluster_labels=[]
for i in range(n_clusters):
    cluster_labels+=[f'Classe {i+1}']

# classes weights
fig, ax = plt.subplots(figsize=(19,19))
im=ax.matshow(lito_SOM.weights)
for (i, j), z in np.ndenumerate(lito_SOM.weights):
    ax.text(j, i, '{:0.2f}'.format(z), ha='center', va='center')

plt.yticks(range(n_clusters), cluster_labels, fontsize=9)
plt.xticks(range(len(features)), features, rotation=55, fontsize=10, ha='left')
fig.colorbar(im, label='weights', orientation='horizontal')
plt.gca().set_aspect('equal')
plt.gcf().set_size_inches(10, 10)
plt.show()
```



In [53]:

```
id_ = [1,2,3,4,5,6,7,8,9]

relcolor =  matplotlib.cm.Set3
colors = np.array(relcolor.colors)[id_]
relcolor = matplotlib.colors.ListedColormap(colors)

idxs=np.arange(0, n_clusters, 1)
half=(idxs[1]-idxs[0])/2
ticks=np.linspace(idxs[0]+half, idxs[-1]-half, n_clusters)

datafig, ax=plt.subplots(figsize=(16, 16), facecolor='w')
im=plt.pcolormesh(xs, ys, predictions.reshape(ny, nx), cmap=relcolor, shading='auto')
plt.xlim(xs.min(), xs.max())
plt.ylim(ys.min(), ys.max())
cbar_ax = fig.add_axes([0.93, 0.3, 0.05, 0.4])
cbar = fig.colorbar(im, cax=cbar_ax, label = u'Classes', orientation='vertical',cmap='viridis',ticks=ticks)

cbar.ax.set_yticklabels(cluster_labels, fontsize=8)
plt.suptitle('SOM - Aerogeophysical Data (SF23_YA_III4)')
plt.axis('scaled')
plt.show()
```



In [ ]:

```
n_clusters=18
lito_SOM = SOM(m=n_clusters, n=1, sigma=1.5, dim=len(features), max_iter=10000)
lito_SOM.fit(data)

# predição de classes
predictions = lito_SOM.predict(data)

# create labels
cluster_labels=[]
for i in range(n_clusters):
    cluster_labels+=[f'Classe {i+1}']

# classes weights
fig, ax = plt.subplots(figsize=(27,27))
im=ax.matshow(lito_SOM.weights)

for (i, j), z in np.ndenumerate(lito_SOM.weights):
    ax.text(j, i, '{:0.2f}'.format(z), ha='center', va='center')

plt.yticks(range(n_clusters), cluster_labels, fontsize=8)
```

```
plt.xticks(range(len(features)), features, rotation=55, fontsize=9, ha='left')
fig.colorbar(im, label='weights', orientation='horizontal',cmap='Reds')
plt.gca().set_aspect('equal')
plt.gcf().set_size_inches(15, 10)
plt.show()
```

In [ ]:
```
df
```

In [ ]:
```
id_ = [1,2,3,4,5,6,7,8,9,10,11]

relcolor =  matplotlib.cm.Set3
colors = np.array(relcolor.colors)[id_]
relcolor = matplotlib.colors.ListedColormap(colors)

idxs=np.arange(0, n_clusters, 1)
half=(idxs[1]-idxs[0])/2
ticks=np.linspace(idxs[0]+half, idxs[-1]-half, n_clusters)

datafig, ax=plt.subplots(figsize=(16, 16), facecolor='w')
im=plt.pcolormesh(xs, ys, predictions.reshape(ny, nx), cmap=relcolor, shading='auto')
plt.xlim(xs.min(), xs.max())
plt.ylim(ys.min(), ys.max())
cbar_ax = fig.add_axes([0.93, 0.3, 0.05, 0.4])
cbar = fig.colorbar(im, cax=cbar_ax, label = u'Classes', orientation='vertical',
                    ticks=ticks)
cbar.ax.set_yticklabels(cluster_labels, fontsize=8)
plt.suptitle('SOM - Aerogeophysical Data ()')
plt.axis('scaled')
plt.show()
```

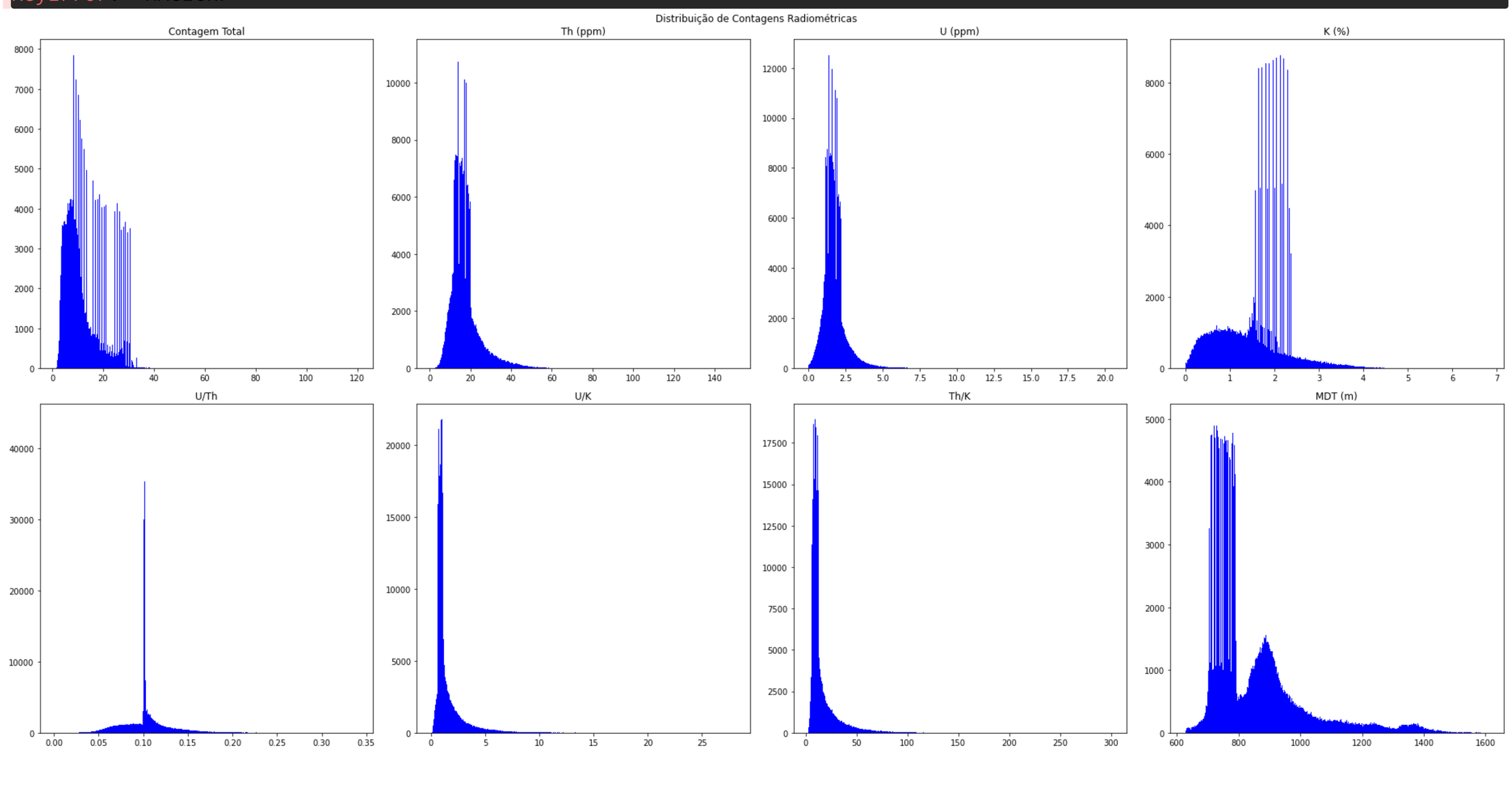
Testes

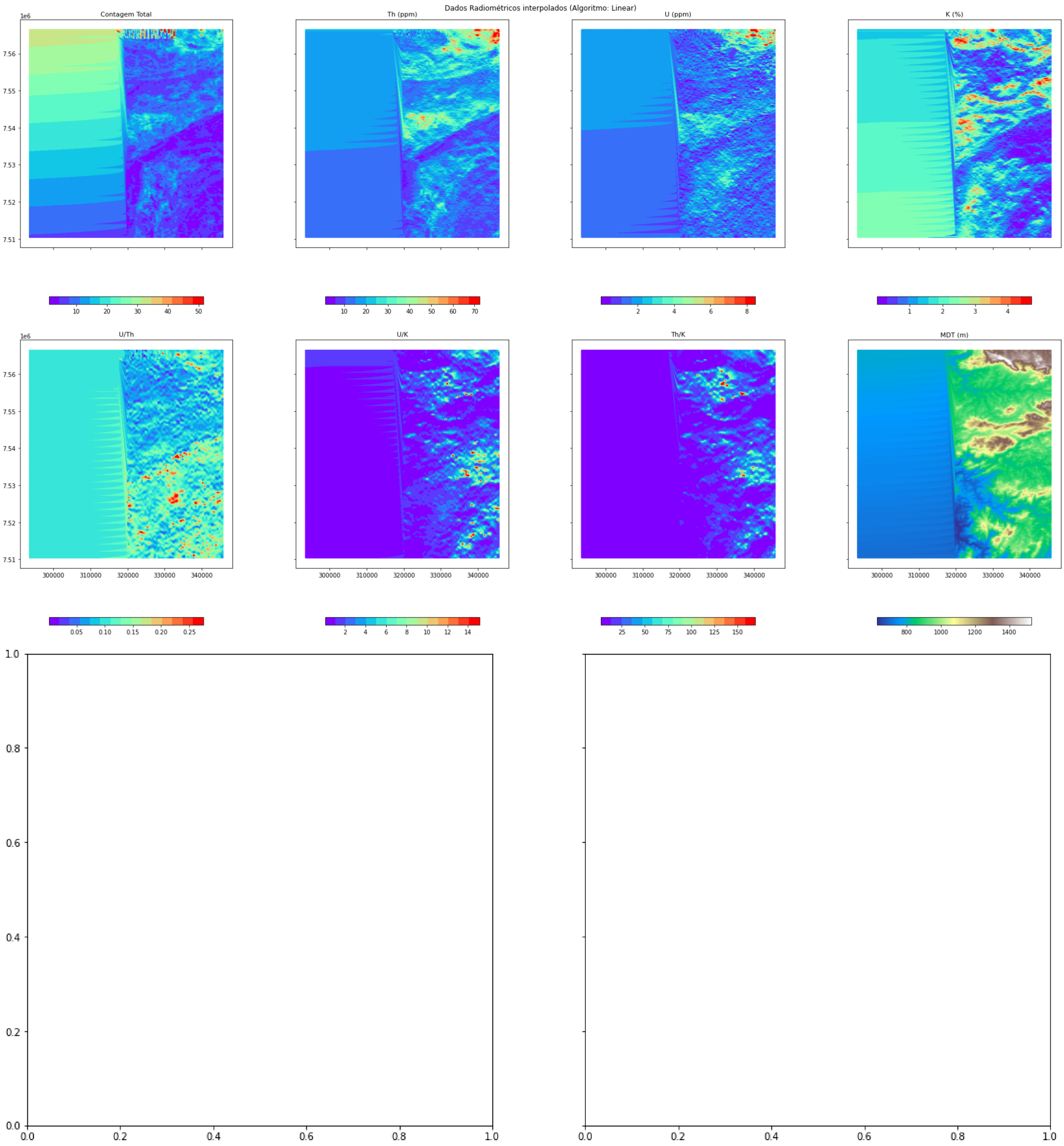
In [38]:
```
df = quadricula['SF23_YA_III']['geof_1105_linear']

plot_histograms(df)
plot_raw_gama_data(df,suptitle='Dados Radiométricos interpolados (Algoritmo: Linear)',figsize=(27,16))
plot_raw_mag_data(df,suptitle='Dados Magnetométricos i4nterpolados (Algoritmo: Linear)')



df_rs = df
df_rs.rename(columns={'X':'E_utm','Y':'N_utm'},inplace=True)
#df_rs.fillna(0,inplace=True)



xpixel_size = (df_rs.E_utm.max()-df_rs.E_utm.min())/df_rs.E_utm.unique().size
ypixel_size = (df_rs.N_utm.max()-df_rs.N_utm.min())/df_rs.N_utm.unique().size
print('x:', xpixel_size, 'y:', ypixel_size)



nx=df_rs.E_utm.unique().size
ny=df_rs.N_utm.unique().size
ratio=ny/nx
xs = df_rs.E_utm.values.reshape(ny, nx)
ys = df_rs.N_utm.values.reshape(ny, nx)

features = list(df_rs.columns[2:])
print(features)



plot_corr(df_rs[features], size=12)
#plt.savefig('figs/correlation_matrix.png', dpi=400, bbox_inches='tight')
```

```python
scaler = StandardScaler()
data = scaler.fit_transform(df[features].values)
# data = df_rs[features].values


n_clusters=9
lito_SOM = SOM(m=n_clusters, n=1, sigma=1.5, dim=len(features), max_iter=10000)
lito_SOM.fit(data)

# predição de classes
predictions = lito_SOM.predict(data)
```

```
           count        mean         std         min        0.1%  \
CTCOR     291239.0   14.057950    8.314372    1.056178    2.196800
eTh       291239.0   17.375129    7.196011    1.086301    4.289636
eU        291239.0    1.717553    0.759648    0.001000    0.082884
KPERC     291239.0    1.624323    0.718870    0.001000    0.028415
MDT       291239.0  843.954755  151.029453  628.050028  637.173393
THKRAZAO  291239.0   15.498761   14.703311    2.880782    3.458544
UTHRAZAO  291239.0    0.103280    0.026796    0.001000    0.026685
UKRAZAO   291239.0    1.510356    1.503171    0.067118    0.201504

                1%         5%         25%         50%         75%  \
CTCOR     3.021996    4.022677    7.559324   11.503030   20.045111
eTh       6.621703    9.173999   13.144339   16.138562   19.152415
eU        0.377678    0.800280    1.286038    1.625169    1.990670
KPERC     0.145315    0.360722    1.118930    1.719864    2.086522
MDT     680.539810  708.617543  740.993163  781.355940  901.930870
THKRAZAO    4.627477    6.248332    8.334885   10.854753   15.273700
UTHRAZAO    0.044972    0.061427    0.096816    0.101418    0.108092
UKRAZAO     0.337799    0.614170    0.793110    1.009742    1.520187

              99.95%         max
CTCOR      51.557656   120.246254
eTh        72.264622   149.824499
eU          8.471868    20.442278
KPERC       4.729640     6.807711
MDT      1529.348634  1608.777462
THKRAZAO   169.119402   300.580091
UTHRAZAO     0.274968     0.340606
UKRAZAO     15.106215    28.016459
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/indexes/base.py:3802, in Index.get_loc(self, ke
y, method, tolerance)
   3801 try:
-> 3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/_libs/index.pyx:138, in pandas._libs.index.IndexEngi
ne.get_loc()

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/_libs/index.pyx:165, in pandas._libs.index.IndexEngi
ne.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:5745, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:5753, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'MAGIGRF'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [38], in <cell line: 5>()
      3 plot_histograms(df)
      4 plot_raw_gama_data(df,suptitle='Dados Radiométricos interpolados (Algoritmo: Linear)',figsize=(27,16))
----> 5 plot_raw_mag_data(df,suptitle='Dados Magnetométricos i4nterpolados (Algoritmo: Linear)')
      8 df_rs = df
      9 df_rs.rename(columns={'X':'E_utm','Y':'N_utm'},inplace=True)

File ~/projetos/geologist/src.py:883, in plot_raw_mag_data(raw_data, suptitle, minimo, maximo)
    881 X, Y = raw_data.X, raw_data.Y
    882 for ax, f in zip(axs.flat, mag_dic_titles):
--> 883     vmin = raw_data_describe[f][minimo]
    884     vmax = raw_data_describe[f][maximo]
    885     if f == 'MDT':

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/frame.py:3807, in DataFrame.__getitem__(self, k
ey)
   3805 if self.columns.nlevels > 1:
   3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
   3808 if is_integer(indexer):
   3809     indexer = [indexer]

File ~/.config/ambiente_geologico/lib/python3.10/site-packages/pandas/core/indexes/base.py:3804, in Index.get_loc(self, ke
y, method, tolerance)
   3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:
-> 3804     raise KeyError(key) from err
   3805 except TypeError:
   3806     # If we have a listlike key, _check_indexing_error will raise
   3807     #  InvalidIndexError. Otherwise we fall through and re-raise
   3808     #  the TypeError.
   3809     self._check_indexing_error(key)

KeyError: 'MAGIGRF'
```





In [ ]:
```
df
```

In [ ]:
```
# create labels
cluster_labels=[]
for i in range(n_clusters):
    cluster_labels+=[f'Classe {i+1}']

# classes weights
fig, ax = plt.subplots(figsize=(19,26))
im=ax.matshow(lito_SOM.weights)
for (i, j), z in np.ndenumerate(lito_SOM.weights):
    ax.text(j, i, '{:0.2f}'.format(z), ha='center', va='center')

plt.yticks(range(n_clusters), cluster_labels, fontsize=9)
plt.xticks(range(len(features)), features, rotation=55, fontsize=10, ha='left')
fig.colorbar(im, label='weights', orientation='horizontal')
plt.gca().set_aspect('equal')
plt.gcf().set_size_inches(19, 26)
plt.show()
```

In [ ]:
```
id_ = [1,2,3,4,5,6,7,8,9]
relcolor =  matplotlib.cm.Set3
colors = np.array(relcolor.colors)[id_]
relcolor = matplotlib.colors.ListedColormap(colors)



idxs=np.arange(0, n_clusters, 1)
half=(idxs[1]-idxs[0])/2
ticks=np.linspace(idxs[0]+half, idxs[-1]-half, n_clusters)

datafig, ax=plt.subplots(figsize=(16, 16), facecolor='w')
im=plt.pcolormesh(xs, ys, predictions.reshape(ny, nx), cmap=relcolor, shading='auto')
plt.xlim(xs.min(), xs.max())
plt.ylim(ys.min(), ys.max())
cbar_ax = fig.add_axes([0.93, 0.3, 0.05, 0.4])
cbar = fig.colorbar(im, cax=cbar_ax, label = u'Classes', orientation='vertical',
                    ticks=ticks)
cbar.ax.set_yticklabels(cluster_labels, fontsize=8)
plt.suptitle('SOM - Aerogeophysical Data (SF23_YA_III4)')
plt.axis('scaled')
plt.show()
```

Classificações Supervisionadas

Rotulando amostras com classes litológicas

In [ ]:
```
import shapely.speedups
from shapely import geometry
shapely.speedups.enable()

geof_1089_linear['geometry'] = [geometry.Point(x,y) for x, y in zip(geof_1089_linear['X'],
geof_1089_linear['Y'])]
gdf_1089_linear = geof_1089_linear.set_geometry('geometry')

gdf_1089_linear.set_crs('EPSG:32723',inplace=True)
gdf_1089_linear.geometry
```

In [ ]:
```
Upload_litologia(quadricula,'litologia_100k')
```

In [ ]:
```
litologia=quadricula['SB24_ZB_II']['litologia_100k']
litologia.to_crs('EPSG:32724',inplace=True)
print(litologia.crs)
litologia.reset_index(drop=True,inplace=True)

dic_litologico = describe_geologico(litologia)
print(litologia.columns)
```

In [ ]:
```
print(dic_litologico['SIGLA']['len'])
print(dic_litologico['SIGLA']['lista'])
gdf_1089_linear
```

In [ ]:
```
litologia.plot('SIGLA',figsize=(16,16),legend=True)
```

In [ ]:
```
geof_1089_linear['closest_unit'] = geof_1089_linear['geometry'].apply(lambda x:
litologia['SIGLA'].iloc[litologia.distance(x).idxmin()]) # .idxmin() Retorna o indice do menor valor
```

In [ ]:
```
geof_1089_linear.to_csv('/home/ggrl/database/csv/SB24_ZB_II_gama_linear_100m.csv',index=False)
```