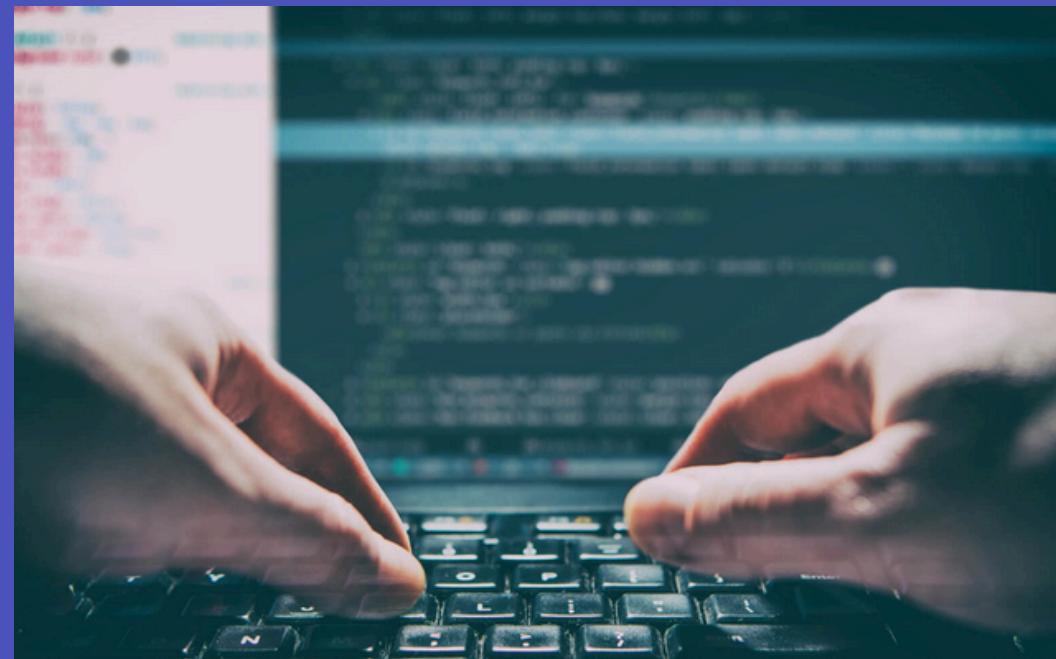


AUTOMATIZAÇÃO DE MOSAICOS RGB E NDVI (SENTINEL-2)



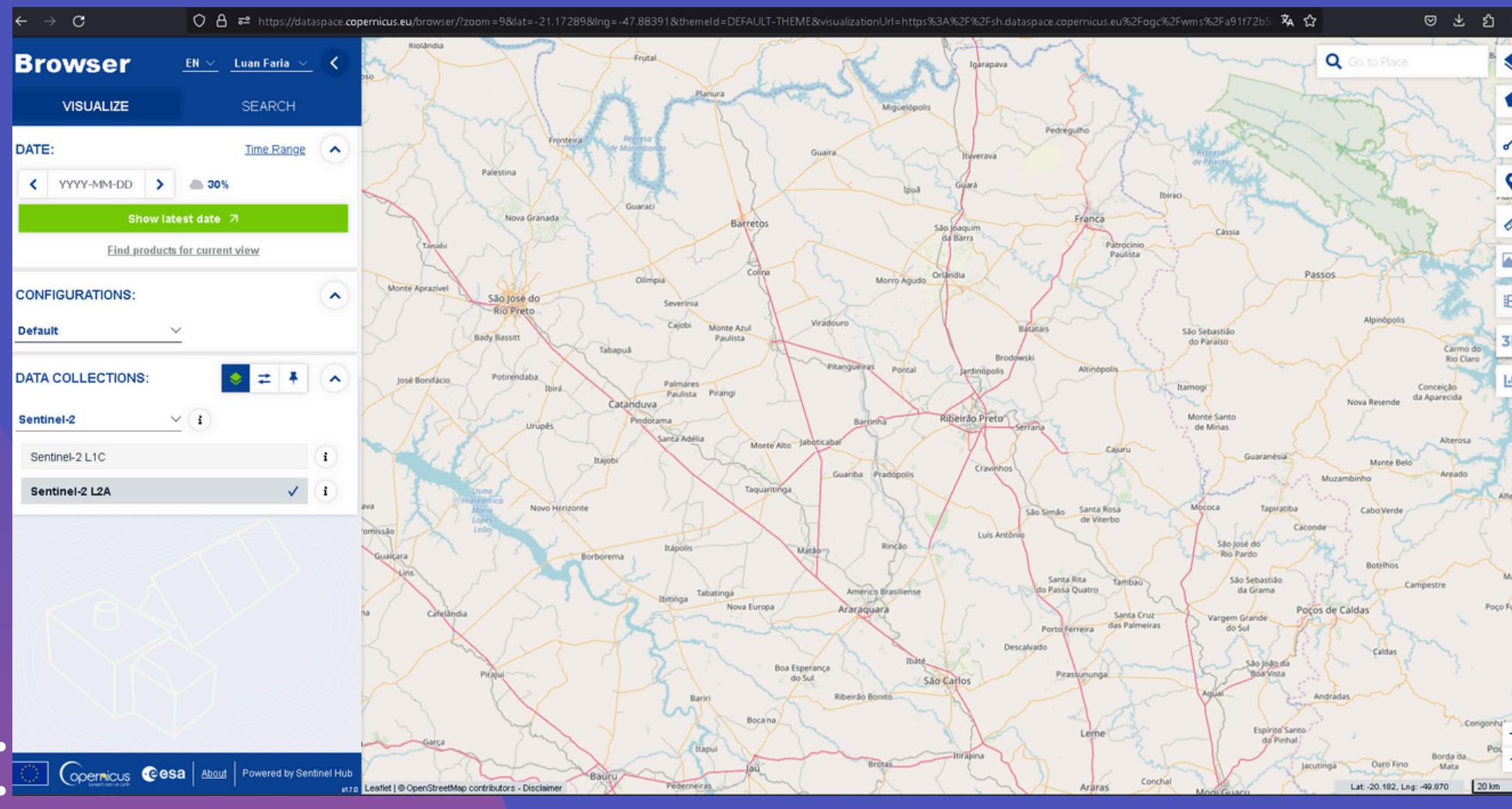
O objetivo do código é facilitar e deixar o processo de mosaicar imagens mais rápido e simples.





BAIXANDO IMAGENS

- As imagens utilizadas foram do Sentinel-2, baixadas do site da copernicus;



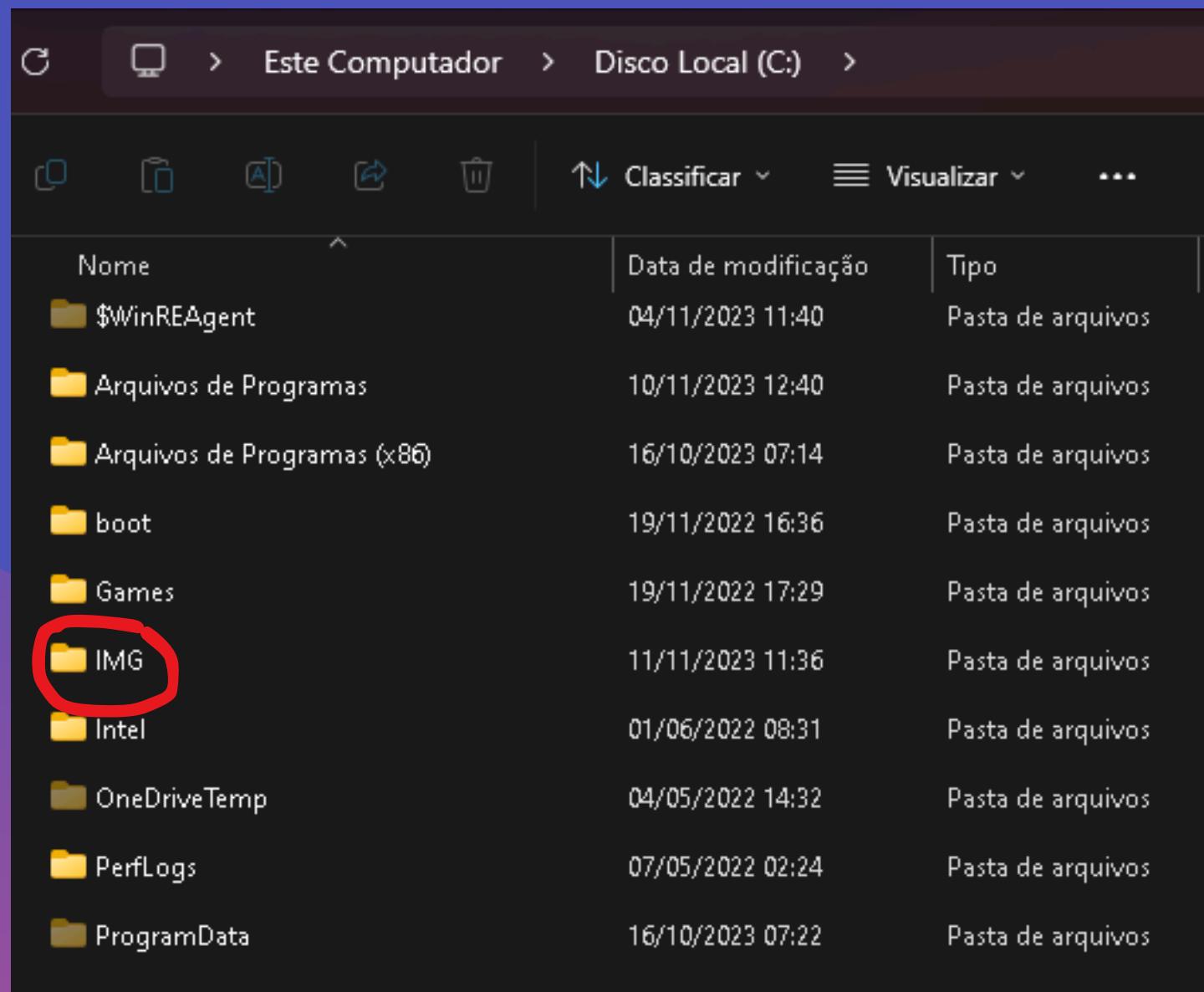
(https://dataspace.copernicus.eu/browser/?zoom=3&lat=26&lng=0&themeld=DEFAULT-THEME&visualizationUrl=https%3A%2F%2Fsh.dataspace.copernicus.eu%2Fogc%2Fwms%2Fa91f72b5-f393-4320-bc0f-990129bd9e63&datasetId=S2_L2A_CDAS&demSource3D=%22MAPZEN%22&cloudCoverage=30)

As bandas selecionadas foram 2,3,4 Referentes ao RGB, e a Banda 8 que para o Sentinel representa a Banda do Infra-Vermelho (utilizada para a composição do NDVI)

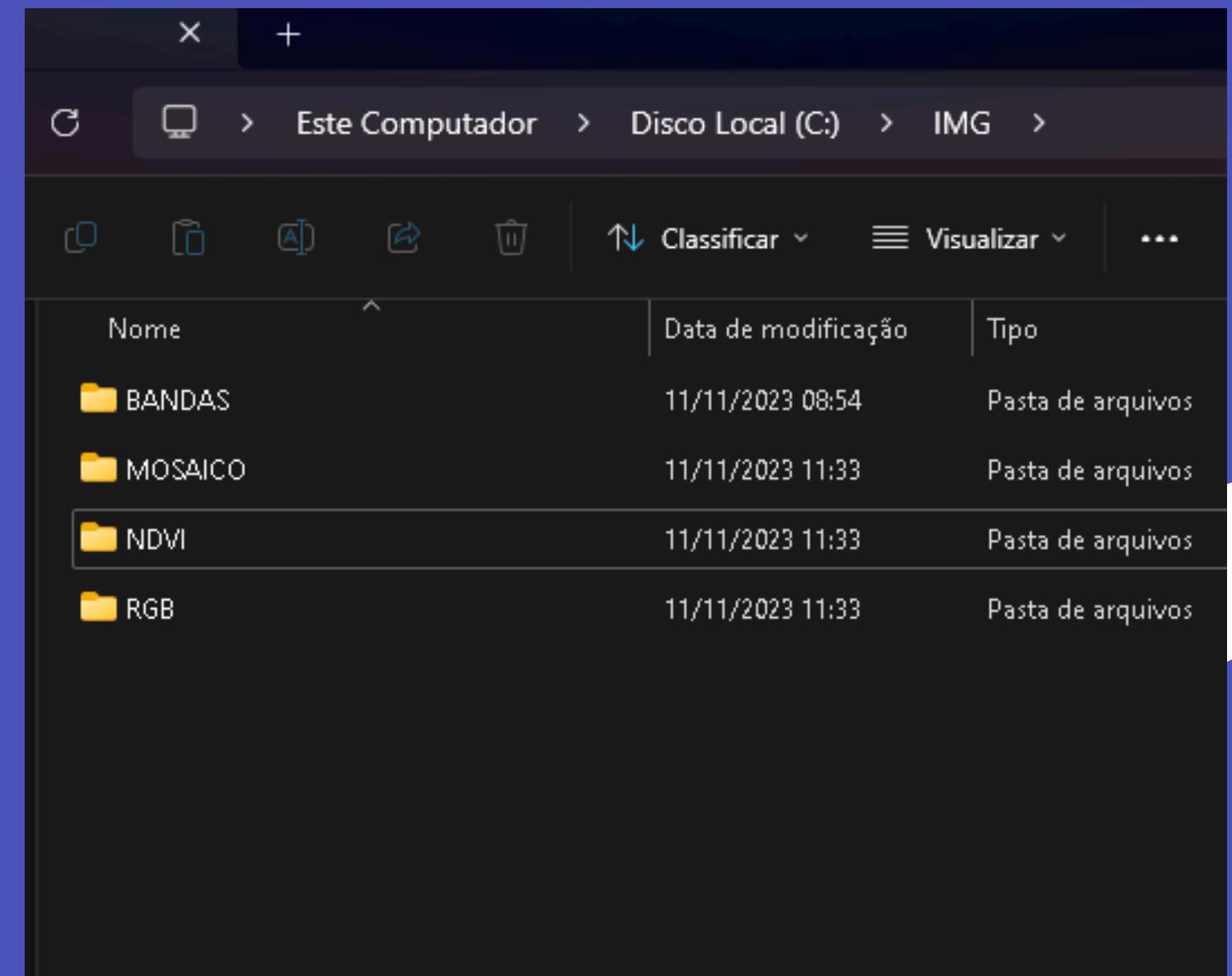
X TRABALHANDO O AMBIENTE

Após o Dowloand das bandas das imagens, será necessário criar uma estrutura de pastas no Disco local C;

É necessário a criação de uma pasta chamada IMG contendo as pastas BANDAS, MOSAICO, NDVI e RGB.



Este Computador > Disco Local (C:) >		
Nome	Data de modificação	Tipo
\$WinREAgent	04/11/2023 11:40	Pasta de arquivos
Arquivos de Programas	10/11/2023 12:40	Pasta de arquivos
Arquivos de Programas (x86)	16/10/2023 07:14	Pasta de arquivos
boot	19/11/2022 16:36	Pasta de arquivos
Games	19/11/2022 17:29	Pasta de arquivos
IMG	11/11/2023 11:36	Pasta de arquivos
Intel	01/06/2022 08:31	Pasta de arquivos
OneDriveTemp	04/05/2022 14:32	Pasta de arquivos
PerfLogs	07/05/2022 02:24	Pasta de arquivos
ProgramData	16/10/2023 07:22	Pasta de arquivos

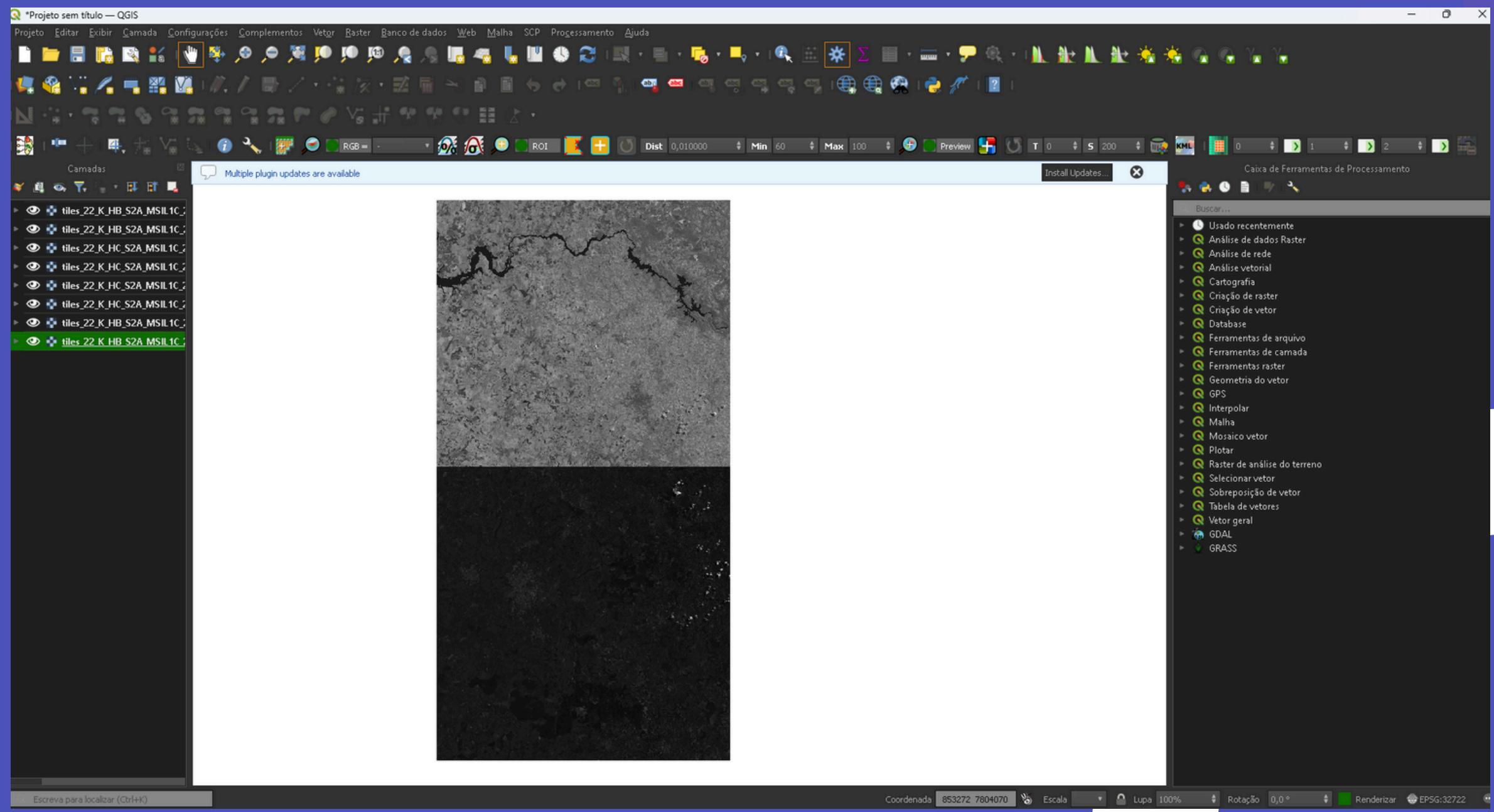


Este Computador > Disco Local (C:) > IMG >		
Nome	Data de modificação	Tipo
BANDAS	11/11/2023 08:54	Pasta de arquivos
MOSAICO	11/11/2023 11:33	Pasta de arquivos
NDVI	11/11/2023 11:33	Pasta de arquivos
RGB	11/11/2023 11:33	Pasta de arquivos

X ANTES DO PROCESSAMENTO

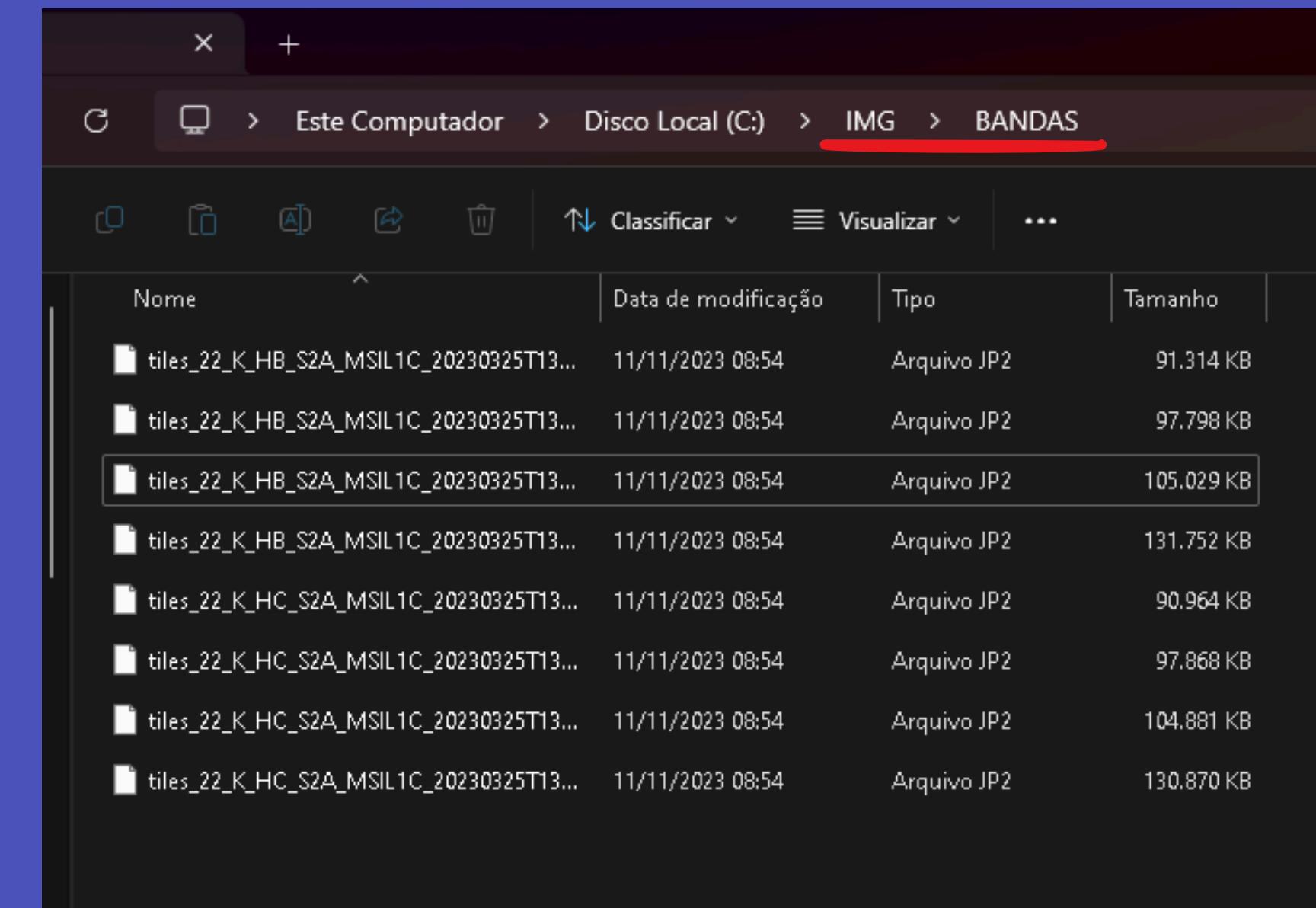
Como exemplo foram baixadas duas imagens (bandas 4328);

Pode-se observar os 8 arquivos necessários antes de rodar o código.



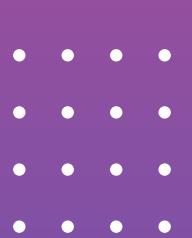
X ANTES DO PROCESSAMENTO

Insira os arquivos na pasta **BANDAS**, e se atente para que as outras pastas estejam **VAZIAS**.





CÓDIGO



Foram utilizadas algumas bibliotecas python.

Com essas bibliotecas é possível caminhar entre as pastas do meu computador, alem de poder realizar todo o processo GIS com total independencia, sem a necessidade de rodar dentro do ArcGis ou do Qgis, o que permite uma mobilidade e a generalização do processo.



```
from rasterio.transform import from_origin  
import glob  
import numpy as np  
import os  
import rasterio  
from rasterio.merge import merge
```



Comandos para instalação:

- pip install geopandas;
- pip install rasterio.



```
• trabalho_pos_puc.py 9+ ×
• trabalho_pos_puc.py > ...
9
10 pasta_geral='C:/IMG/'
11
12 pasta_BANDAS_brutas = pasta_geral + 'BANDAS/'
13 pasta_mosaico = pasta_geral + 'MOSAICO/'
14 pasta_rgb = pasta_geral + 'RGB/'
15 pasta_ndvi = pasta_geral + 'NDVI/'

16
17
18 lista_unicos = []
19 for file in os.listdir(pasta_BANDAS_brutas):
20     lista_unicos.append(file[139:158])
21
22 print(lista_unicos)
23
24 lista_unicos = (set(lista_unicos))
25 for unique in lista_unicos:
26     search_criteria = "*" + unique + ".jp2"
27
28 selecao = os.path.join(pasta_BANDAS_brutas, search_criteria)
29
30 print(selecao)
31
32 bandsJp2 = glob.glob(selecao)
33
34 print(bandsJp2)

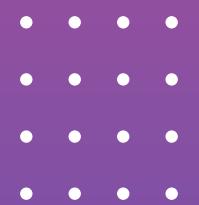
35 MUDAR O ULTIMO VALOR CONFORME O
36 src_files_to_mosaic = [] FUSO, NO EX FOI UTILIZADO IMAGENS
37
38 for band in bandsJp2:
39     src = rasterio.open(band)
40     src_files_to_mosaic.append(src)
41
42 print(src_files_to_mosaic)
43
44 mosaic, out_trans = merge(src_files_to_mosaic)
45
46 out_meta = src.meta.copy()
47 out_meta.update({"driver": "JP2OpenJPEG",
48                  "height": mosaic.shape[1],
49                  "width": mosaic.shape[2],
50                  "transform": out_trans,
51                  "crs": "EPSG:32722" #herdar das imagens "EPSG:32722"
52                  })
53
54
```

```
55     with rasterio.open(pasta_mosaico + band[34:38] + unique + ".jp2", "w", **out_meta) as dest:
56         dest.write(mosaic)
57
58     print('gerando RGB')
59     lista_bandas = []
60     for file in os.listdir(pasta_mosaico):
61         lista_bandas.append(file[:19])
62
63     lista_bandas = (set(lista_bandas))
64     for unique in lista_bandas:
65         #gerar RGB
66
67         b4 = rasterio.open(pasta_mosaico + unique + '_B04.jp2')
68         b3 = rasterio.open(pasta_mosaico + unique + '_B03.jp2')
69         b2 = rasterio.open(pasta_mosaico + unique + '_B02.jp2')
70         b8 = rasterio.open(pasta_mosaico + unique + '_B08.jp2')
71
72
73     #Create an RGB image
74     with rasterio.open(pasta_rgb + 'RGB_' + unique + '_4328.tif', 'w', driver='GTiff', width=b4.width, height=b4.height,
75                         count=4, crs=b4.crs, transform=b4.transform, dtype=b4.dtypes[0]) as rgb:
76         rgb.write(b4.read(1), 1)
77         rgb.write(b3.read(1), 2)
78         rgb.write(b2.read(1), 3)
79         rgb.write(b8.read(1), 4)
80         #rgb.close()

82
83     #gerar NDVI
84     print('gerando NDVI')
85
86     with rasterio.open(pasta_mosaico + unique + '_B08.jp2') as srcA:
87         bandNir = srcA.read()
88         profile = srcA.profile
89
90     with rasterio.open(pasta_mosaico + unique + '_B04.jp2') as srcB:
91         bandRed = srcB.read()
92
93     np.seterr(invalid='ignore') #ignora aviso: invalid value encountered in true_divide
94     ndvi = (bandNir.astype(float) - bandRed.astype(float)) / (bandNir + bandRed)
95
96     profile = srcA.meta
97     profile.update(driver='GTiff')
98     profile.update(dtype=rasterio.float32)
99
100    result = pasta_ndvi + 'NDVI_RGB_' + unique + '_4328.tif'
101    with rasterio.open(result, 'w', **profile) as dst:
102        dst.write(ndvi.astype(rasterio.float32))
103        crs = srcA.crs
```



CÓDIGO EM PROCESSO



```
PS C:\Users\luand\OneDrive\Área de Trabalho\trabalho\MODELOS_SCRIPT> & C:/Users/luand/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/luand/OneDrive/Área de Trabalho/trabalho/MODELOS_SCRIPT/trabalho_pos_puc.py"
['20230325T132231_B02', '20230325T132231_B03', '20230325T132231_B04', '20230325T132231_B08', '20230325T132231_B02', '20230325T132231_B03', '20230325T132231_B04', '20230325T132231_B08']
C:/IMG/BANDAS/*20230325T132231_B02.jp2
['C:/IMG/BANDAS\tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B02.jp2', 'C:/IMG/BANDAS\tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B02.jp2']
[<open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B02.jp2' mode='r'>, <open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B02.jp2' mode='r'>]
C:/IMG/BANDAS/*20230325T132231_B04.jp2
['C:/IMG/BANDAS\tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B04.jp2', 'C:/IMG/BANDAS\tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B04.jp2']
[<open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B04.jp2' mode='r'>, <open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B04.jp2' mode='r'>]
C:/IMG/BANDAS/*20230325T132231_B03.jp2
['C:/IMG/BANDAS\tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B03.jp2', 'C:/IMG/BANDAS\tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B03.jp2']
[<open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B03.jp2' mode='r'>, <open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B03.jp2' mode='r'>]
C:/IMG/BANDAS/*20230325T132231_B08.jp2
['C:/IMG/BANDAS\tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B08.jp2', 'C:/IMG/BANDAS\tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B08.jp2']
[<open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HB_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHB_20230325T181803.SAFE_GRANULE_L1C_T22KHB_A040504_20230325T132231_IMG_DATA_T22KHB_20230325T132231_B08.jp2' mode='r'>, <open DatasetReader name='C:/IMG/BANDAS/tiles_22_K_HC_S2A_MSIL1C_20230325T132231_N0509_R038_T22KHC_20230325T181803.SAFE_GRANULE_L1C_T22KHC_A040504_20230325T132231_IMG_DATA_T22KHC_20230325T132231_B08.jp2' mode='r'>]
gerando RGB
gerando NDVI
PS C:\Users\luand\OneDrive\Área de Trabalho\trabalho\MODELOS_SCRIPT> & C:/Users/luand/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/luand/OneDrive/Área de Trabalho/trabalho/MODELOS_SCRIPT/trabalho_pos_puc.py"
```

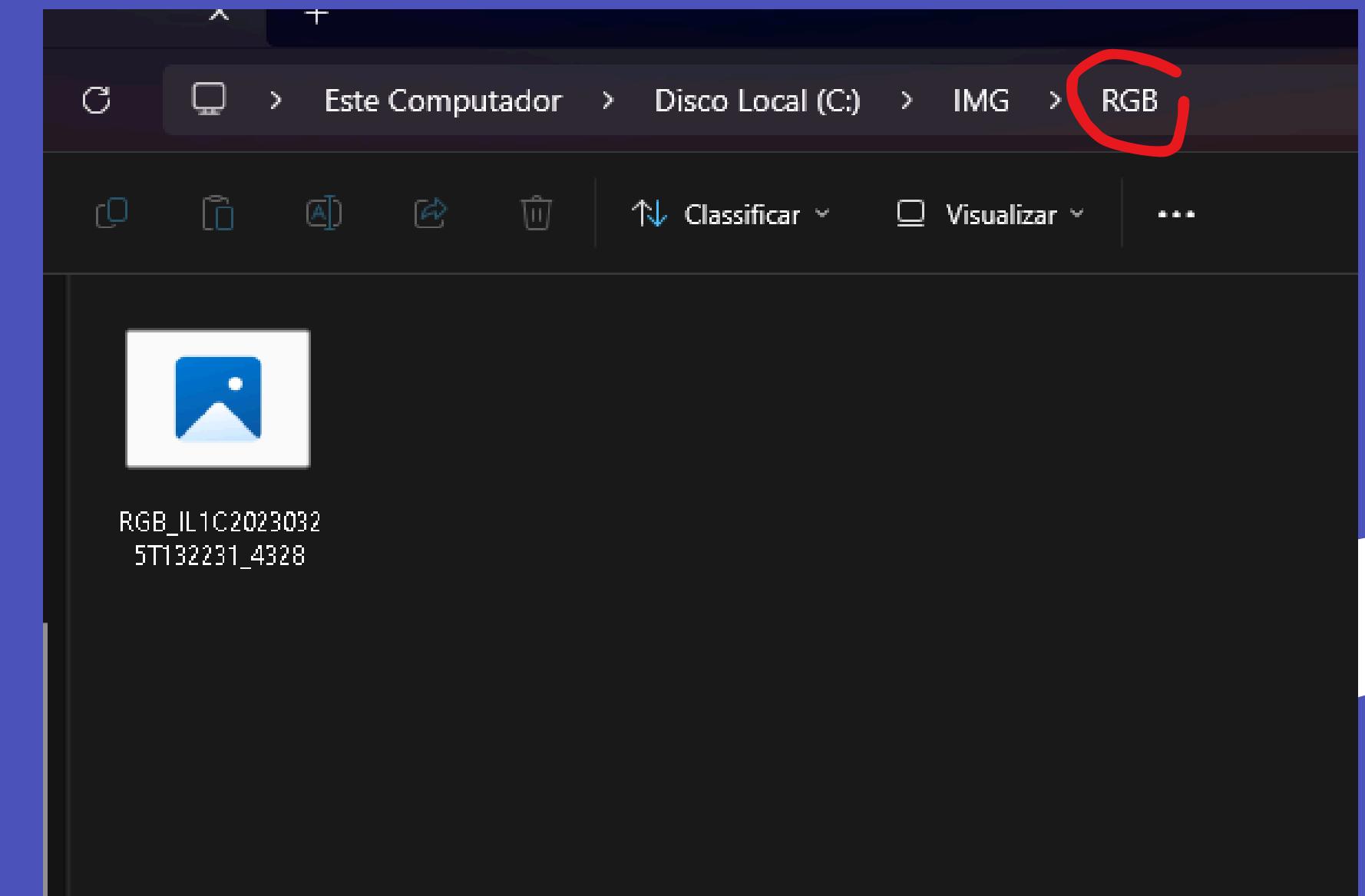
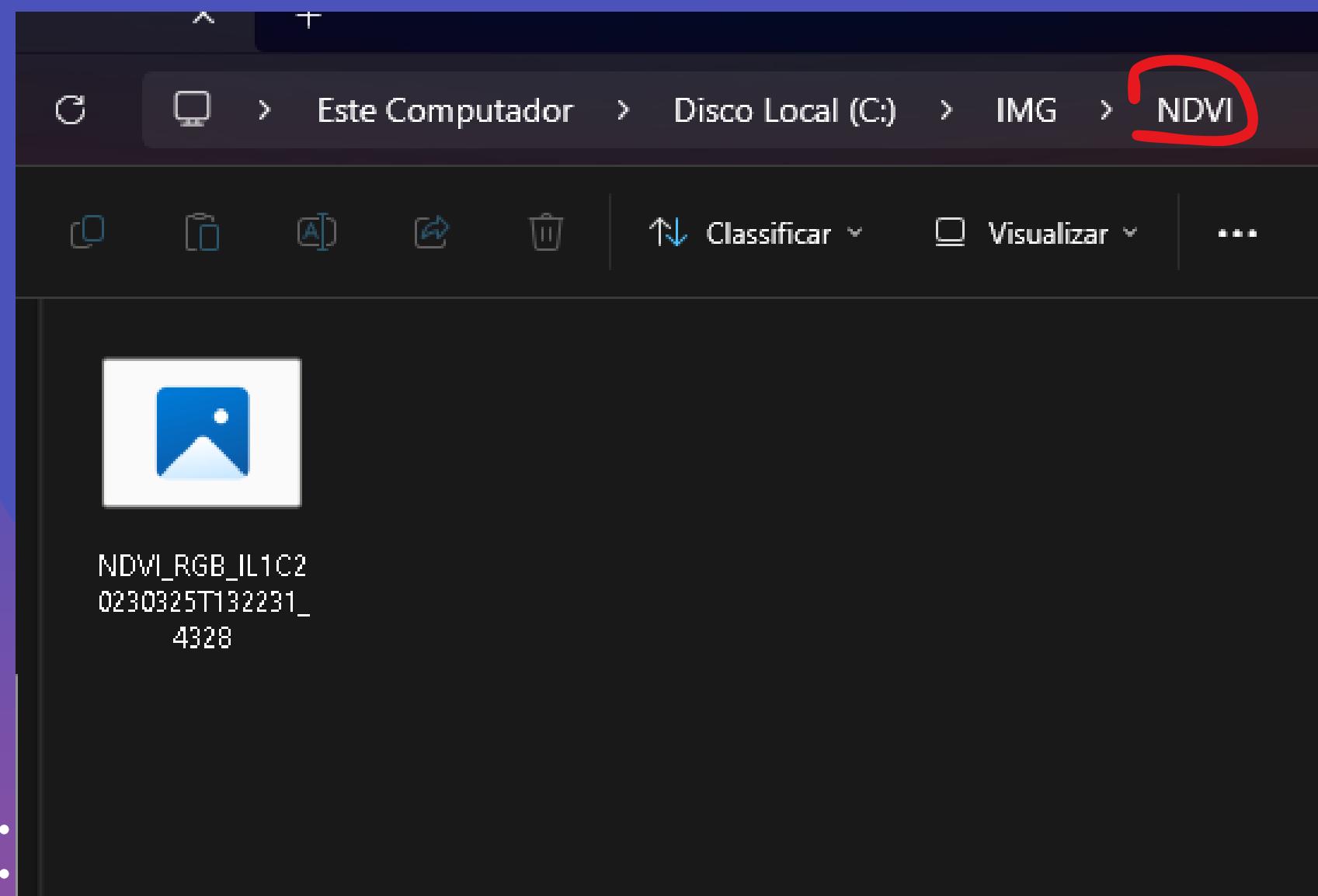


X

RESULTADOS

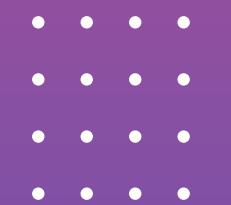


Como resultado temos a imagem RGB na Pasta RGB e a imagen NDVI na pasta NDVI;

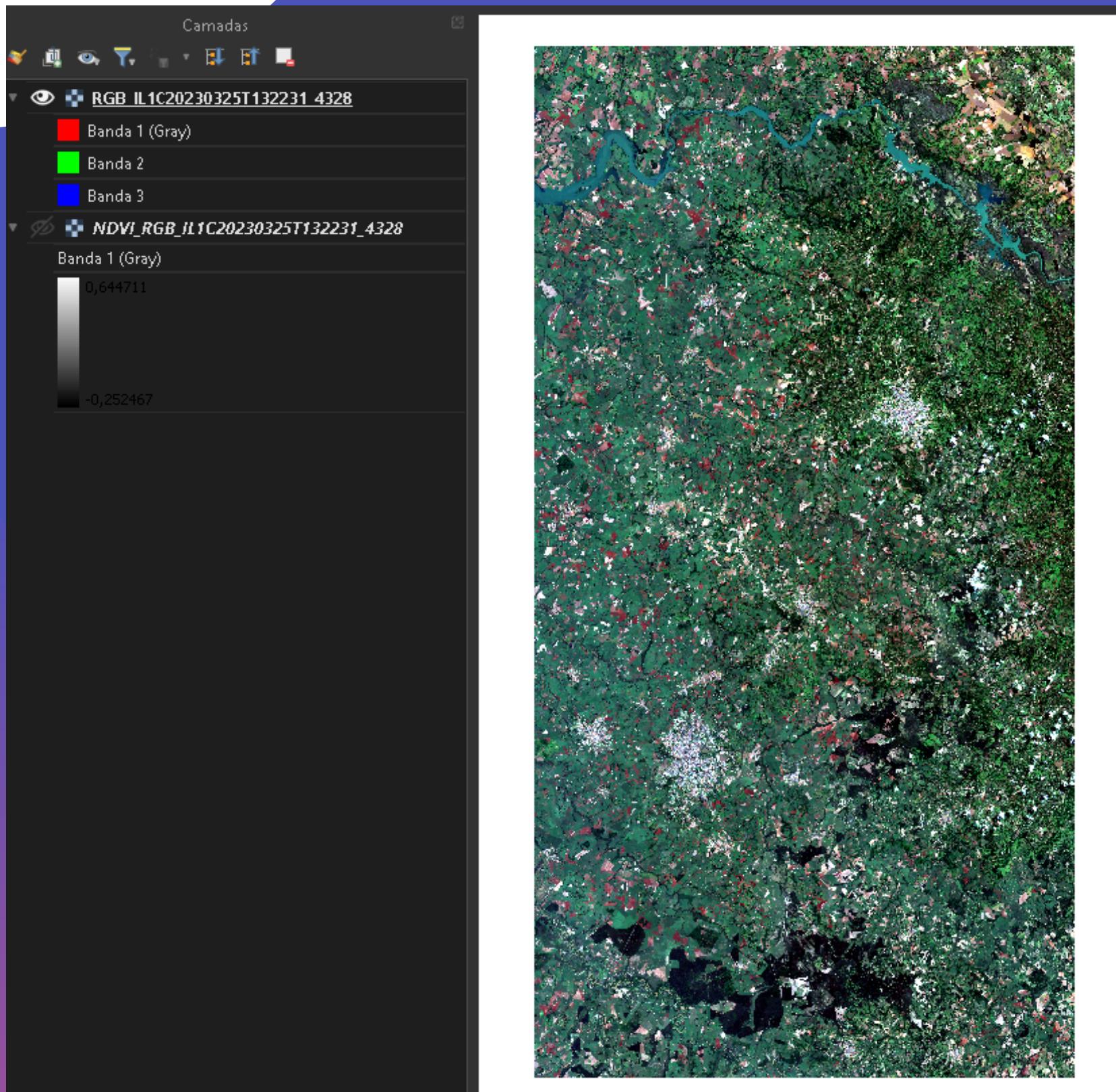


X

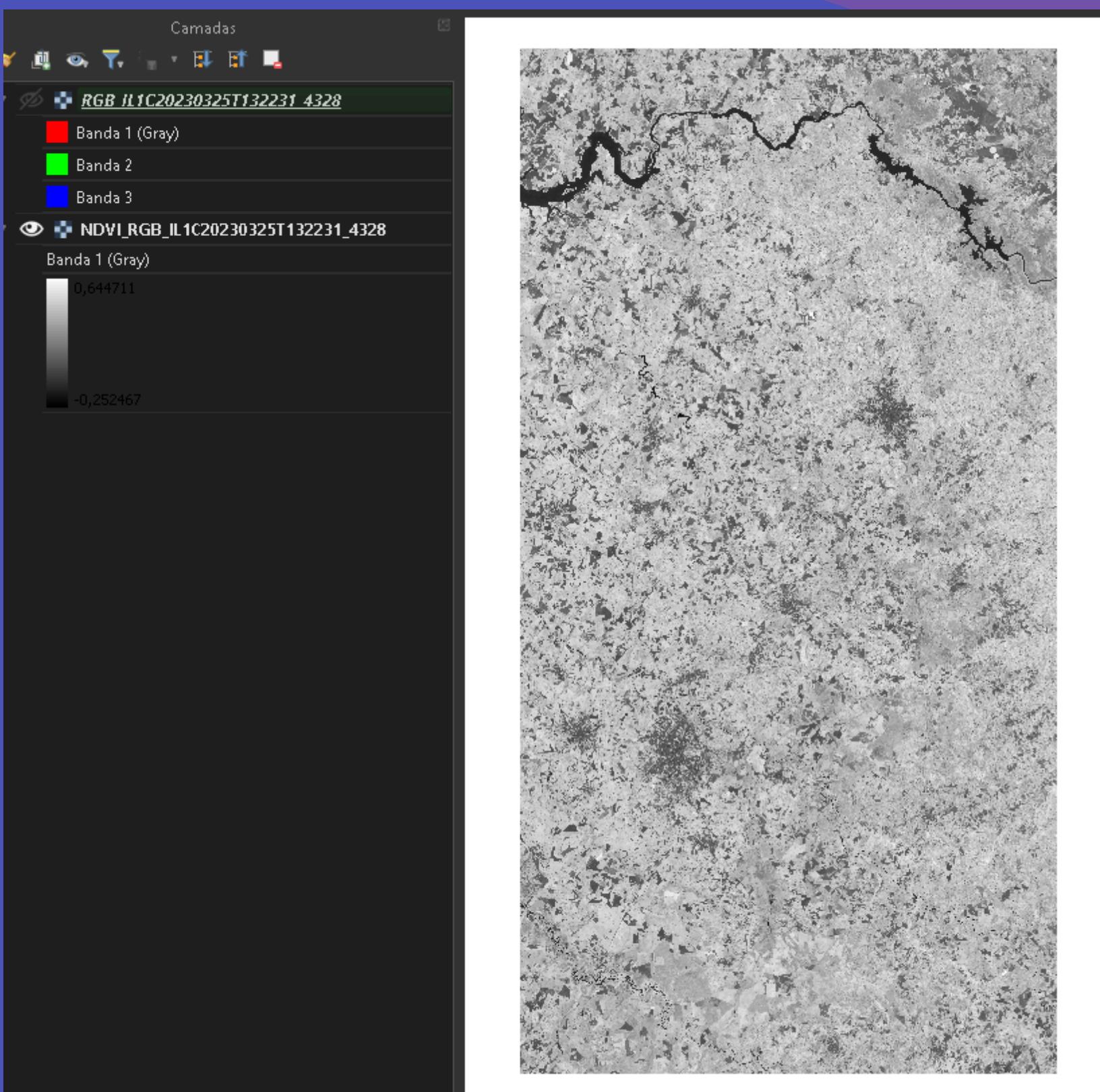
RESULTADOS



RGB



NDV



X

RESULTADOS

RGB



NDV

