

# Análise das Bandas RGB do Landsat

Nesta seção, realizamos uma análise das bandas RGB de uma imagem Landsat. Seguimos os seguintes passos:

1. **Carregamento das Bandas:** Carregamos as bandas 2 (azul), 3 (verde) e 4 (vermelha) da imagem Landsat.
2. **Visualização das Bandas:** Visualizamos as três bandas em subplots separados, utilizando a transformação geoespacial para coordenadas e diferentes mapas de cores para destacar diferentes características.

```
In [87]: # Importa a biblioteca rasterio para trabalhar com dados raster
import rasterio
```

```
In [88]: # Define o caminho para a banda 2 (azul), banda 3 (verde) e banda 4 (vermelha) da imagem Landsat
path_b2 = 'imagens/LC09_L1TP_167078_20240216_20240216_02_T1_B2.TIF'
path_b3 = 'imagens/LC09_L1TP_167078_20240216_20240216_02_T1_B3.TIF'
path_b4 = 'imagens/LC09_L1TP_167078_20240216_20240216_02_T1_B4.TIF'
```

```
In [89]: # Abre os arquivos raster correspondentes às bandas 2, 3 e 4 da imagem Landsat para leitura

with rasterio.open(path_b2, 'r') as dst_b2, \
    rasterio.open(path_b3, 'r') as dst_b3, \
    rasterio.open(path_b4, 'r') as dst_b4:
    # Lê as bandas 2, 3 e 4 dos arquivos raster e atribui os dados às variáveis banda2, banda3 e banda4, respectivamente
    banda2 = dst_b2.read(1)
    banda3 = dst_b3.read(1)
    banda4 = dst_b4.read(1)
```

```
In [91]: # Obtém o perfil (metadados) do arquivo raster da banda 3
dst_b3.profile
```

```
Out[91]: {'driver': 'GTiff', 'dtype': 'uint16', 'nodata': 0.0, 'width': 7691, 'height': 7771, 'count': 1, 'crs': CRS.from_wkt('PROJCS["WGS 84 / UTM zone 36N",GEOGCS["WGS 84",DATUM["World Geodetic System 1984",SPHEROID["WGS 84",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["degree",0.0174532925199433,AUTHORITY["EPSG","9122"]]],PROJECTION["Transverse_Mercator"],PARAMETER["latitude_of_origin",0],PARAMETER["central_meridian",33],PARAMETER["scale_factor",0.9996],PARAMETER["false_easting",500000],PARAMETER["false_northing",0],UNIT["metre",1,AUTHORITY["EPSG","9001"]],AXIS["Easting",EAST],AXIS["Northing",NORTH]]'), 'transform': Affine(30.0, 0.0, 396285.0, 0.0, -30.0, -2758485.0), 'tiled': False}
```

```
In [93]: # Obtém a transformação geoespacial do arquivo raster da banda 3  
dst_b3.profile['transform']
```

```
Out[93]: Affine(30.0, 0.0, 396285.0,  
               0.0, -30.0, -2758485.0)
```

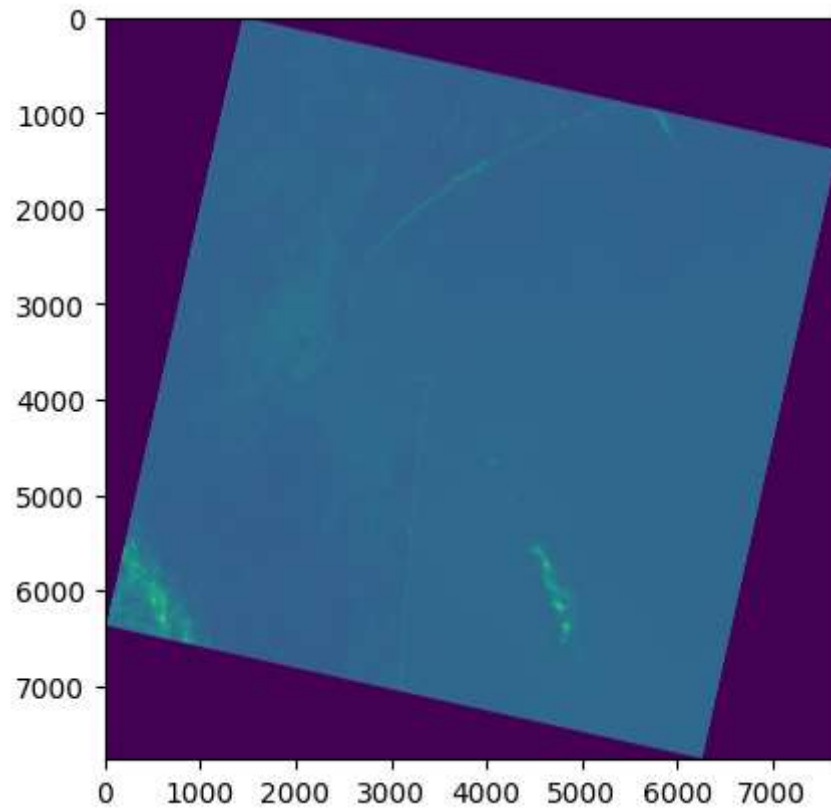
```
In [94]: # Atribui a transformação geoespacial do arquivo raster da banda 3 à variável trans_coords  
trans_coords = dst_b3.profile['transform']
```

```
In [95]: # Variável que armazena a transformação geoespacial do arquivo raster da banda 3  
trans_coords
```

```
Out[95]: Affine(30.0, 0.0, 396285.0,  
               0.0, -30.0, -2758485.0)
```

```
In [96]: # Importa as bibliotecas necessárias para visualização de dados  
import matplotlib.pyplot as plt  
from rasterio.plot import show
```

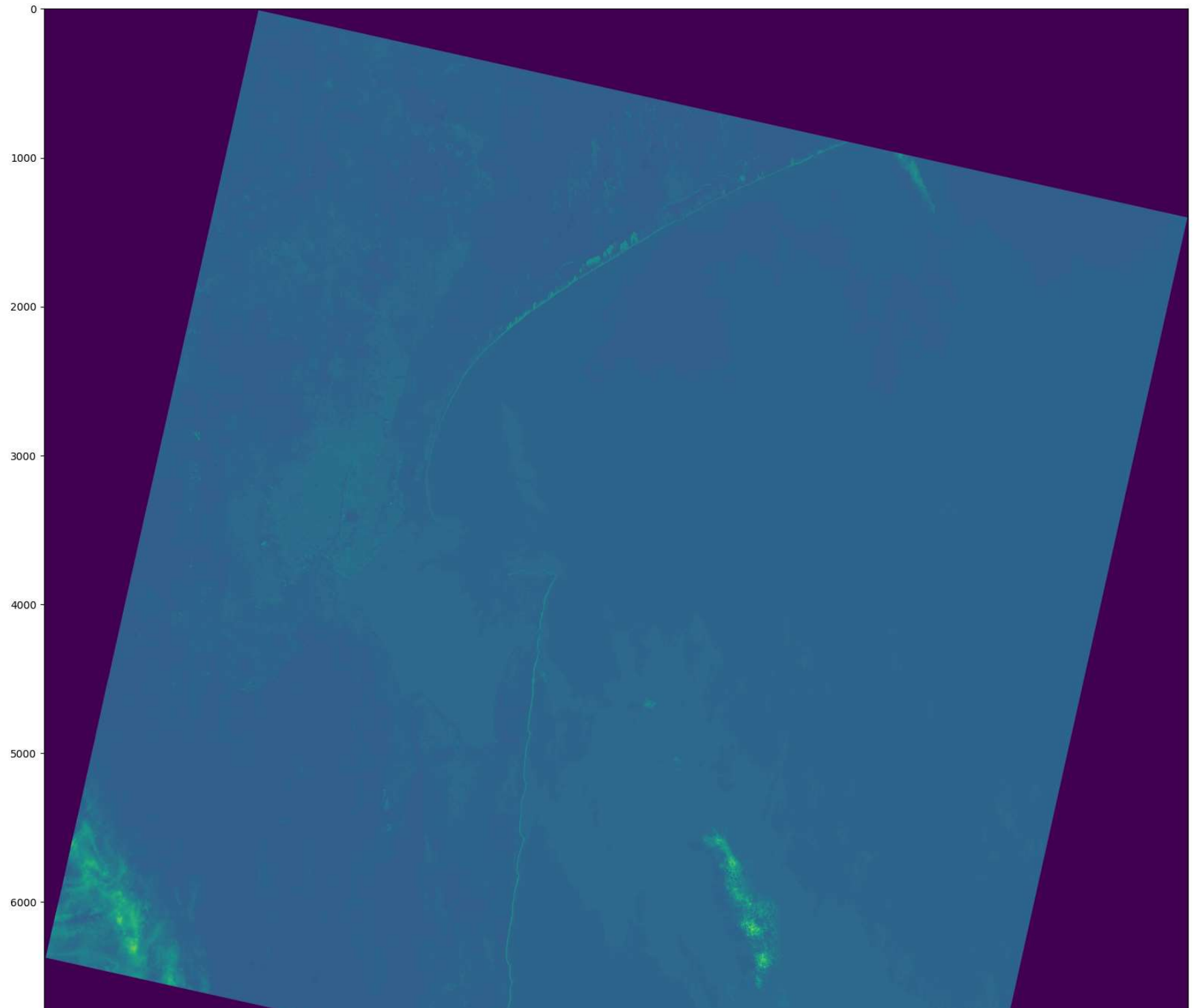
```
In [97]: # Mostra a imagem da banda 2  
show(banda2)
```

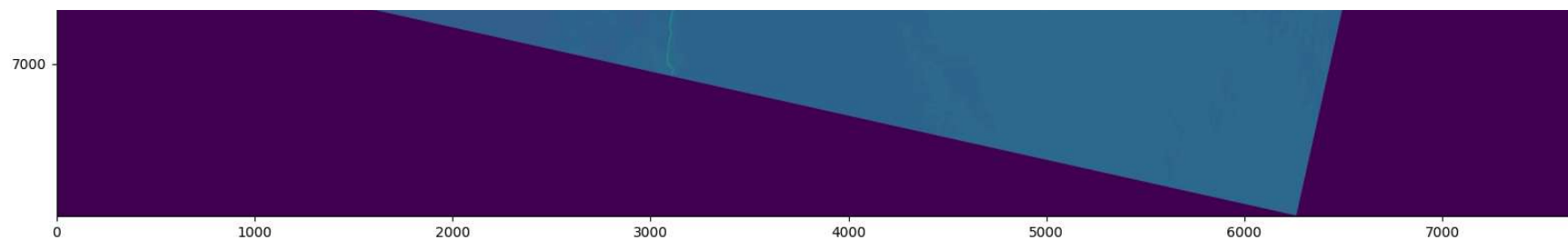


Out[97]: <Axes: >

In [70]:

```
# Cria uma figura com tamanho de 20x20 polegadas e mostra a imagem da banda 2
plt.figure(figsize=(20,20))
show(banda2)
```

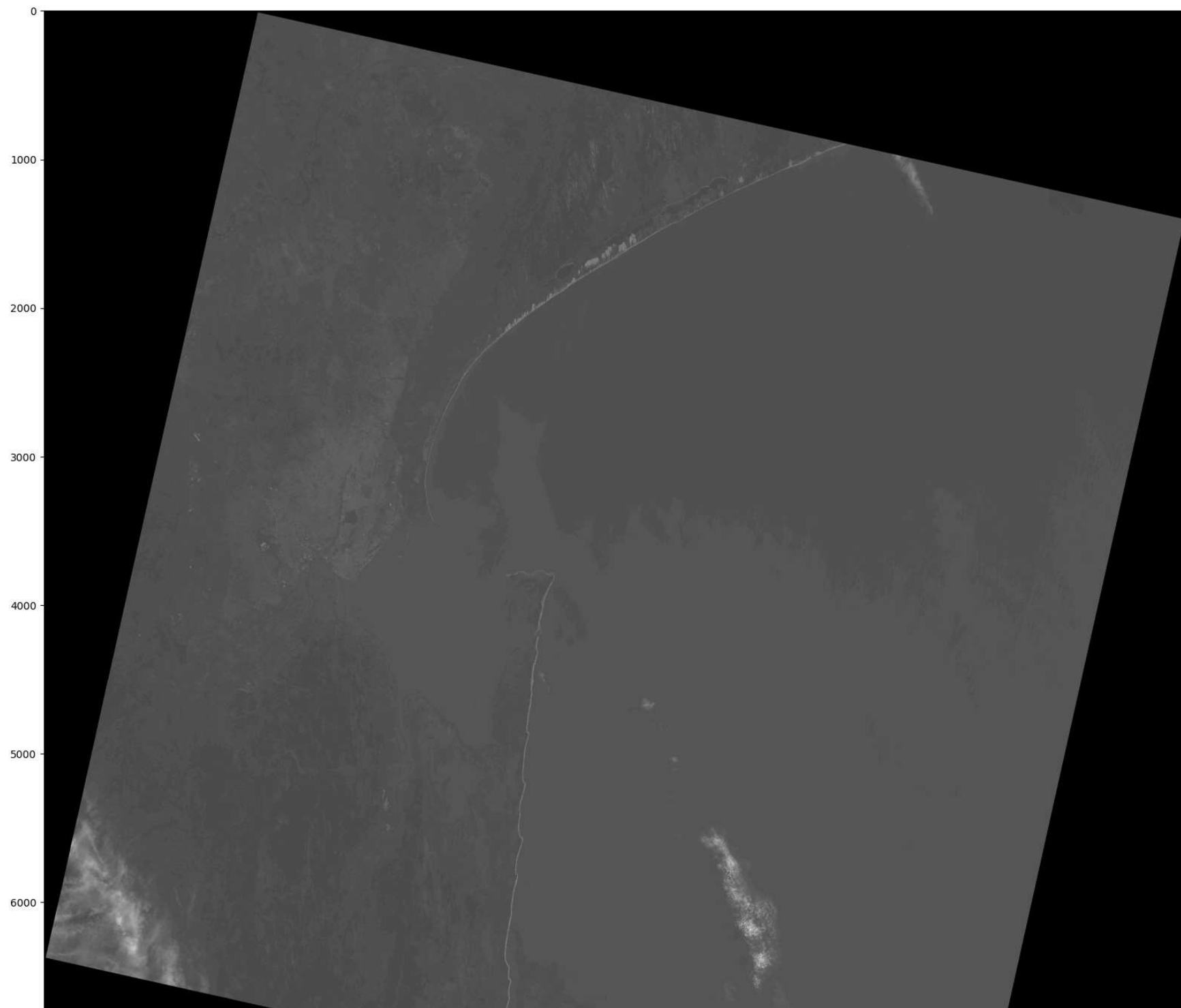


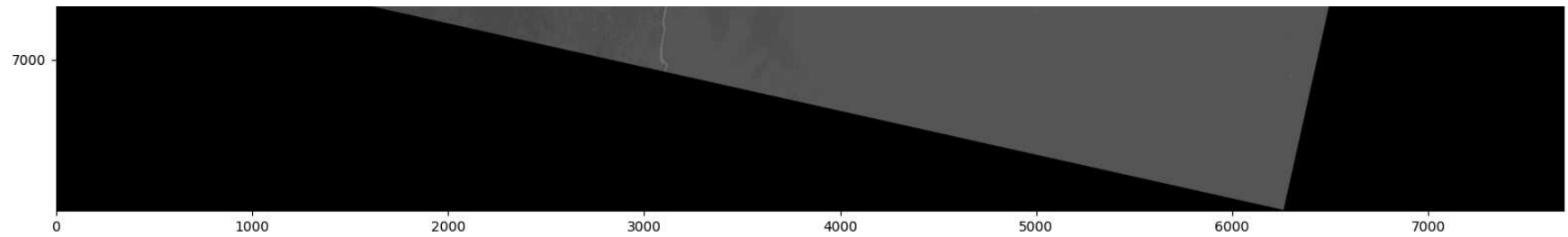


Out[70]: <Axes: >

In [116...

```
# Cria uma figura com tamanho de 20x20 polegadas e mostra a imagem da banda 2 em escala de cinza  
plt.figure(figsize=(20,20))  
show(banda2, cmap='gray')
```

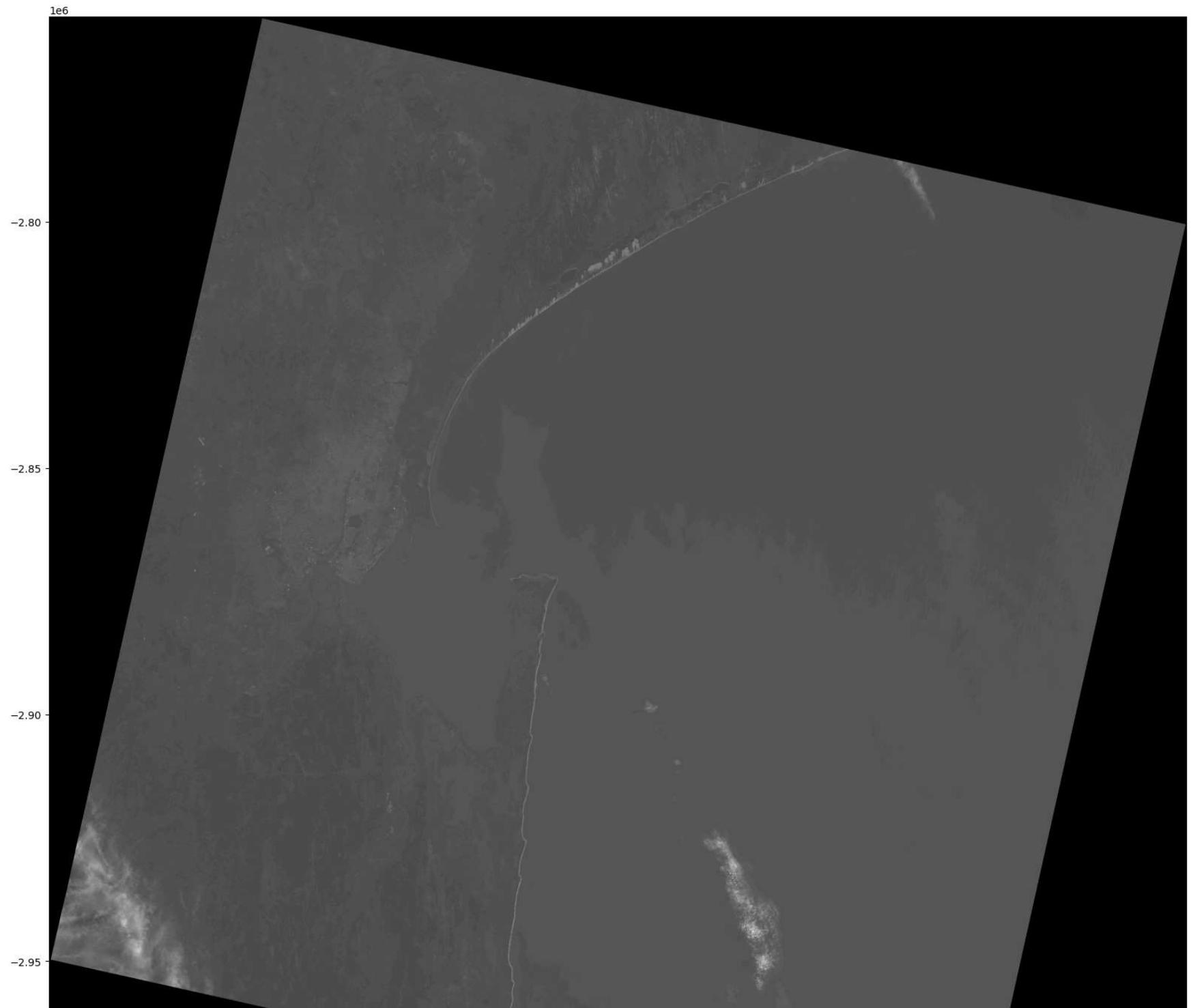




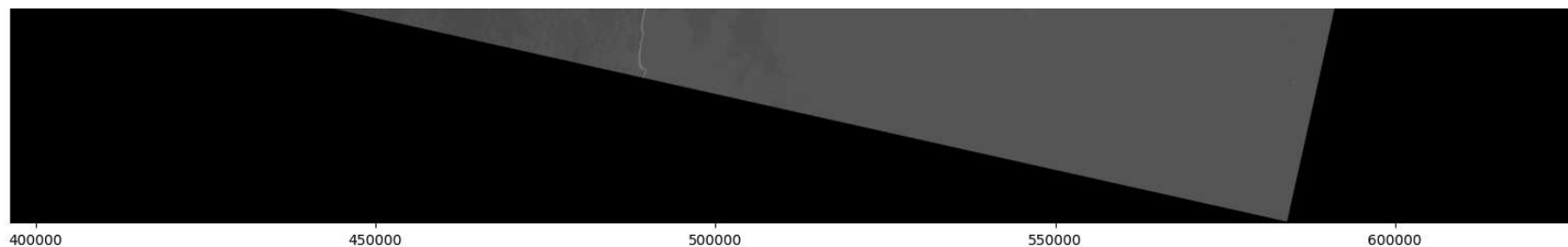
Out[116... <Axes: >

In [78]:

```
# Cria uma figura com tamanho de 20x20 polegadas e mostra a imagem da banda 2 em escala de cinza,  
# utilizando a transformação geoespacial para coordenadas  
# A transformação geoespacial mapeia as coordenadas de pixels da imagem para coordenadas geográficas no mundo real,  
# permitindo a correta representação dos dados raster em relação à superfície da Terra.  
plt.figure(figsize=(20,20))  
show(banda2, cmap='gray', transform= trans_coords)
```

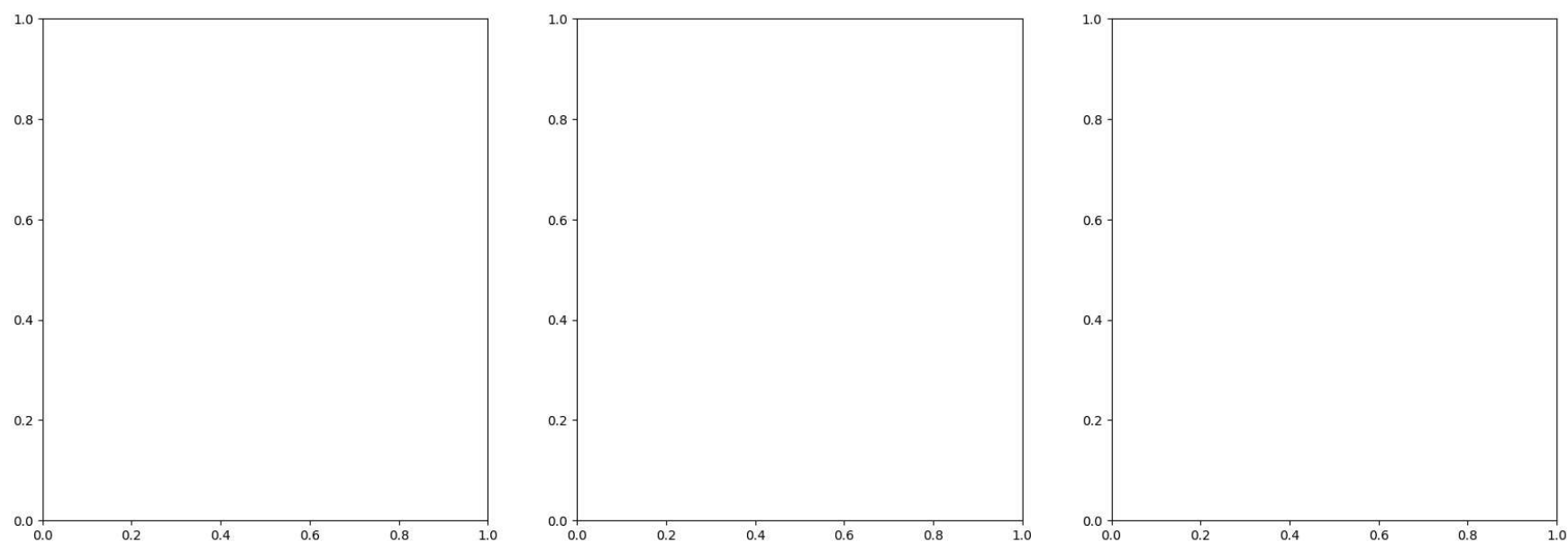






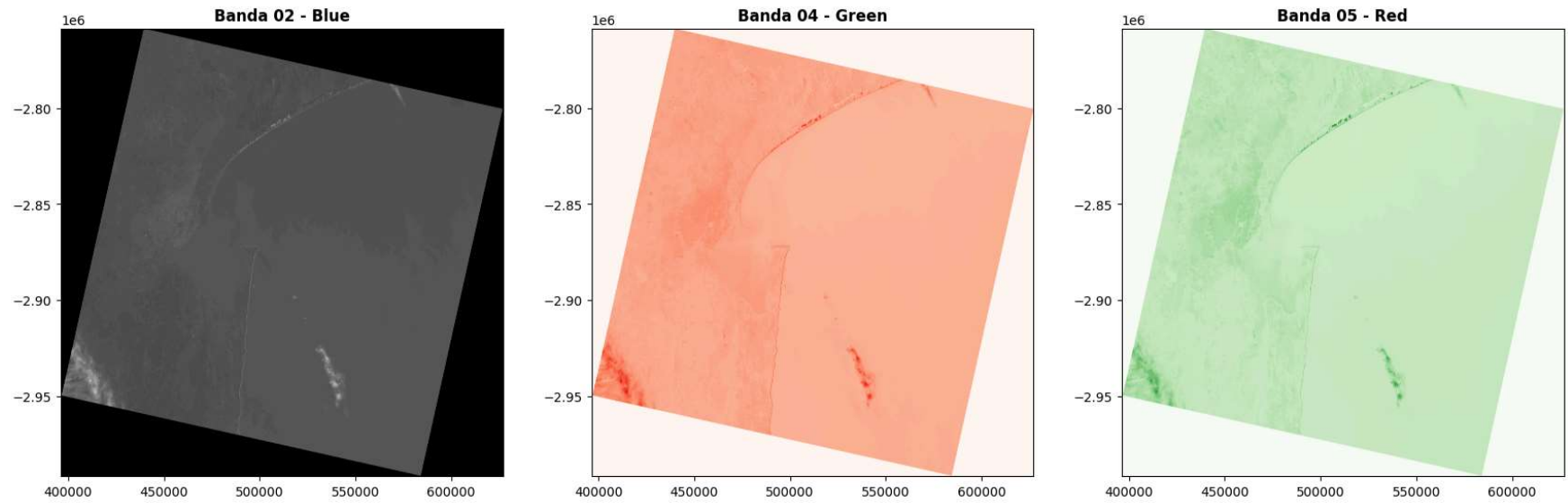
Out[78]: <Axes: >

```
In [79]: # Cria uma figura com três subplots em uma única linha, cada um representado por um eixo (ax1, ax2, ax3),
# com tamanho total de 21x7 polegadas
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(21,7))
```



```
In [100... # Mostra as imagens das bandas 2, 3 e 4 em três subplots diferentes, cada uma com um título específico,
# utilizando a transformação geoespacial para coordenadas e diferentes mapas de cores (cmap)
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(21,7))
show(banda2, ax=ax1, cmap='gray', transform= trans_coords, title='Banda 02 - Blue')
show(banda3, ax=ax2, cmap='Reds', transform= trans_coords, title='Banda 04 - Green')
show(banda4, ax=ax3, cmap='Greens', transform= trans_coords, title='Banda 05 - Red')
```

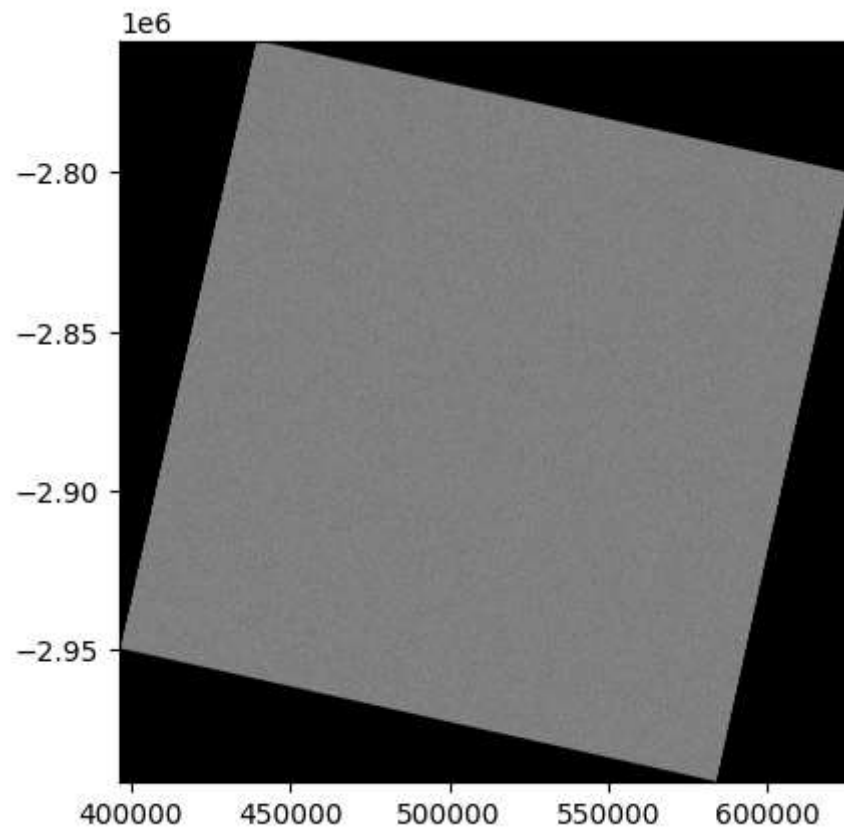
Out[100... <Axes: title={'center': 'Banda 05 - Red'}>



```
In [101...  
# Multiplica os valores de pixel das bandas 2 e 3 elemento por elemento  
resultado = banda2 * banda3
```

```
In [103...  
# Mostra a imagem resultante da multiplicação das bandas 2 e 3, utilizando a transformação geoespacial para coordenadas  
show(resultado, cmap='gray', transform= trans_coords)
```





Out[103... <Axes: >

## Composição RGB

A composição RGB é uma técnica comumente usada em imagens raster para representar cores através da combinação de três canais de cor: vermelho (R), verde (G) e azul (B). Cada canal representa a intensidade da luz em sua respectiva faixa de comprimento de onda. Ao combinar esses três canais em diferentes proporções, podemos criar uma ampla gama de cores, resultando em uma imagem colorida.

Na composição RGB, cada pixel da imagem é representado por três valores numéricos, indicando a intensidade das componentes vermelha, verde e azul. Por exemplo, um pixel com valores (0, 0, 0) representa preto, enquanto um pixel com valores (255, 255, 255) representa branco.

A composição RGB é amplamente utilizada em diversas áreas, incluindo Teledetecção (Sensoriamento Remoto), Fotografia Digital, Design Gráfico e mais. No contexto da Teledetecção, a composição RGB é frequentemente aplicada para criar imagens coloridas a partir de dados obtidos por sensores, como satélites Landsat e Sentinel, permitindo a visualização e análise de características da superfície terrestre de forma mais intuitiva.

Nesta seção, exploraremos o processo de composição RGB de imagens raster utilizando dados obtidos por sensores remotos, como Landsat. Veremos como normalizar os dados das diferentes bandas, combiná-los em uma imagem RGB e visualizar o resultado final. Vamos lá!

```
In [104...  
# Importa a biblioteca NumPy para operações numéricas  
import numpy as np
```

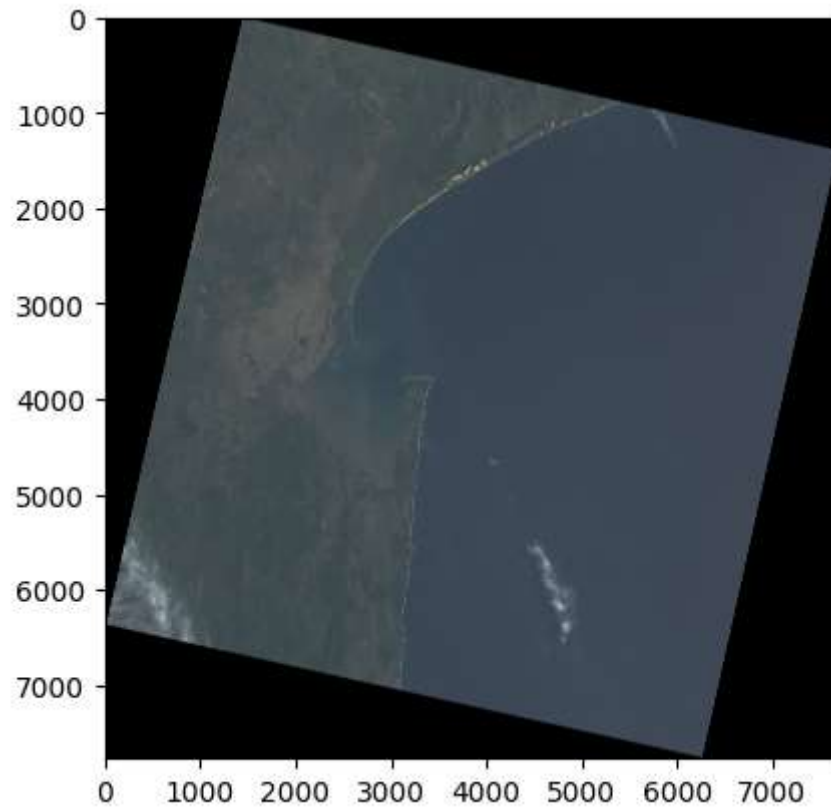
```
In [105...  
...  
Essa função pode ser encontrada na internet  
O que ela faz? Pega todo array e transforma em zeros e uns (0 e 1) para facilitar os calculos de indices  
site: https://automating-gis-processes.github.io/CSC/notebooks/L5/plotting-raster.html  
...  
  
# Function to normalize the grid values  
def normalize(array):  
    """Normalizes numpy arrays into scale 0.0 - 1.0"""  
    array_min, array_max = array.min(), array.max()  
    return ((array - array_min)/(array_max - array_min))
```

```
In [106...  
# Normaliza as bandas vermelha, verde e azul para o intervalo de 0 a 1  
red = normalize(banda4)  
green = normalize(banda3)  
blue = normalize(banda2)
```

```
In [114...  
# Empilha as bandas normalizadas vermelha, verde e azul para formar uma imagem RGB  
rgb = np.dstack((red, green, blue))
```

```
In [115...  
# Mostra a imagem RGB resultante  
plt.imshow(rgb)
```

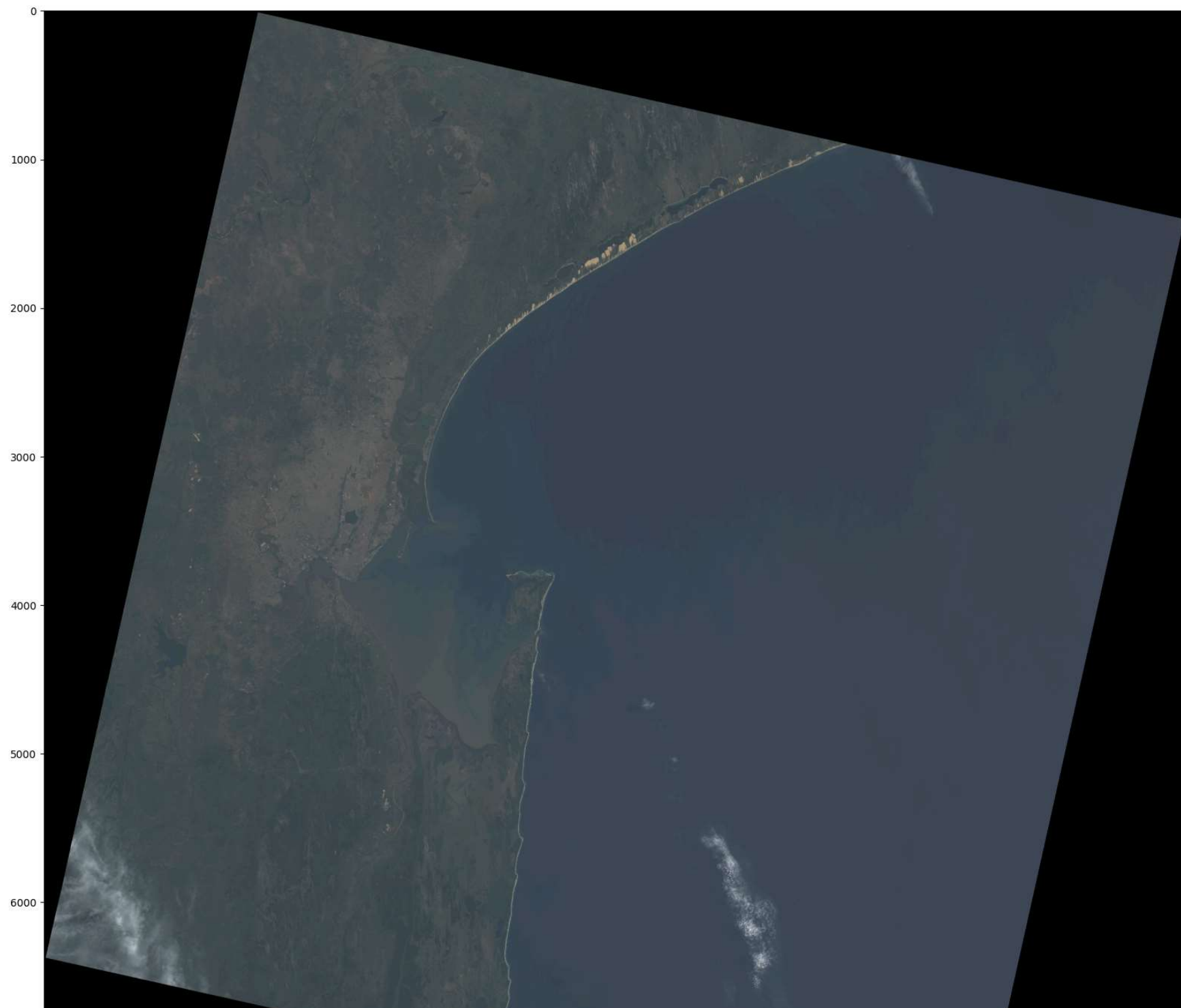
Out[115... <matplotlib.image.AxesImage at 0x215c554a620>

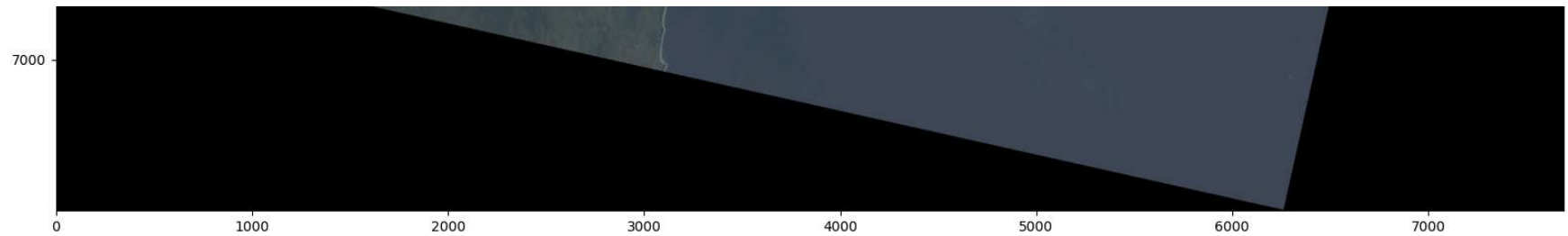


In [110...

```
# Cria uma figura com tamanho de 20x20 polegadas e mostra a imagem RGB resultante  
plt.figure(figsize=(20,20))  
plt.imshow(rgb)
```

Out[110... <matplotlib.image.AxesImage at 0x21586edd0c0>





## Recomendação: Explorando Outras Composições

Além da composição RGB, existem várias outras técnicas de composição que podem ser exploradas para visualização e análise de imagens raster. Algumas delas incluem:

1. **Composição de Cor Falsa (False Color):** Utiliza bandas em diferentes faixas espectrais para criar uma imagem colorida, onde diferentes cores representam diferentes características da superfície terrestre. Por exemplo, vegetação pode ser representada em vermelho, enquanto água pode ser representada em azul.
2. **Composição de Índices de Vegetação (NDVI):** Calcula o Índice de Vegetação por Diferença Normalizada (NDVI) a partir de bandas de infravermelho próximo e vermelho, resultando em uma imagem que destaca a saúde e densidade da vegetação.
3. **Composição de Falso Infravermelho (False Infrared):** Utiliza bandas de infravermelho próximo e vermelho para criar uma imagem colorida, onde a vegetação é representada em tons de vermelho brilhante.
4. **Composição de Componentes Principais (Principal Component Analysis - PCA):** Aplica análise de componentes principais às bandas da imagem para criar uma nova representação colorida que destaca as principais características da cena.

Recomenda-se explorar diferentes composições para melhorar a interpretação e análise de imagens raster, adaptando-se às necessidades específicas de cada aplicação. Cada composição oferece insights únicos sobre as características da superfície terrestre, permitindo uma compreensão mais profunda do ambiente.