Clase 11: Procesamiento de datos espaciales

- Rasterio: https://rasterio.readthedocs.io/en/latest/quickstart.html
- Geopandas
  - https://pygis.io/docs/a_intro.html
  - Datasets https://www.ign.gob.ar/NuestrasActividades/InformacionGeoespacial/CapasSIG

Raster

Con un DEM

```python
import numpy as np
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px


import rasterio as rio
from rasterio.plot import show
```

```python
dem = rio.open("datos/geo/ASTGTM2_S32W061_dem.tif")
```

```python
dem.name
```

```
'datos/geo/ASTGTM2_S32W061_dem.tif'
```

```python
dem
```

```
<open DatasetReader name='datos/geo/ASTGTM2_S32W061_dem.tif' mode='r'>
```

```python
dem.count
```

```
1
```

```python
dem.shape
```

```
(3601, 3601)
```

```
dem.width
```

```
3601
```

```
dem.dtypes
```

```
('int16',)
```

```
dem.bounds
```

```
BoundingBox(left=-61.00013888888889, bottom=-32.00013888888889, right=-59.999861111111116, to
```

```
dem.transform
```

```
Affine(0.0002777777777777778, 0.0, -61.00013888888889,
       0.0, -0.0002777777777777778, -30.999861111111112)
```

```
#dlon = dlat = 0.0002777777777777778
lon = np.linspace(-61.00013888888889,-59.999861111111116,3601)
lat = np.linspace(-32.00013888888889,-30.999861111111112,3601)
len(lat)
```

```
3601
```

```
dem.transform * (0,0)
```

```
(-61.00013888888889, -30.999861111111112)
```

```
dem.transform*(3601,3601)
```

```
(-59.999861111111116, -32.00013888888889)
```

```
dem.crs
```

```
CRS.from_epsg(4326)
```

```
dem.indexes # nos devuelve el índice de cada banda: ojo, arrancan de 1
```

```
(1,)
```

```
band1 = dem.read(1) # leemos la banda 1 de la imagen (es la única banda)
```

```
type(band1)
```

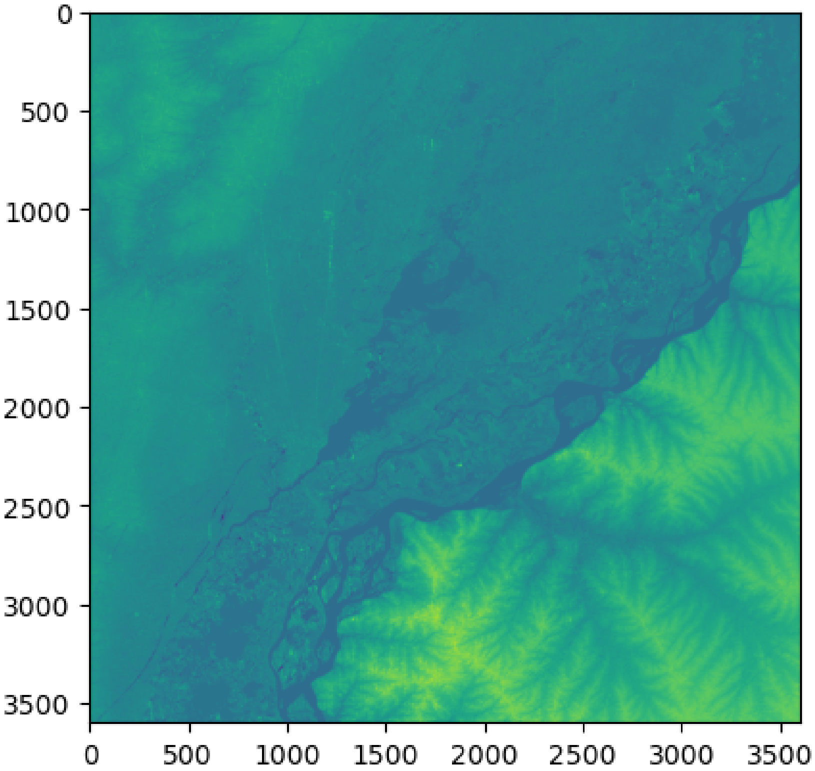
```
numpy.ndarray
```

```
band1
```

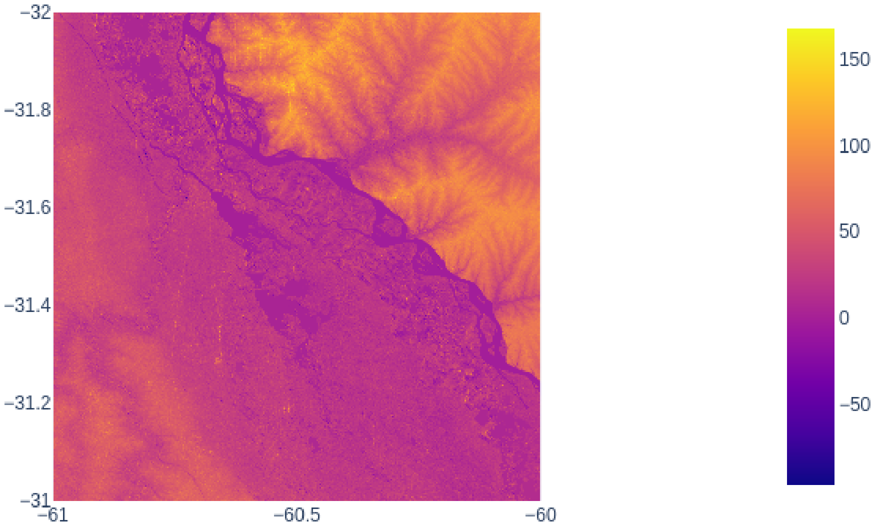
```
array([[ 51,  50,  50, ...,  11,  11,  11],
       [ 54,  53,  53, ...,  11,  11,  11],
       [ 54,  51,  50, ...,  11,  11,  11],
       ...,
       [ 45,  46,  43, ..., 105, 100, 100],
       [ 39,  39,  38, ..., 105, 103, 102],
       [ 33,  32,  32, ..., 100,  99,  98]], dtype=int16)
```

```
plt.imshow(band1, cmap='viridis')
```

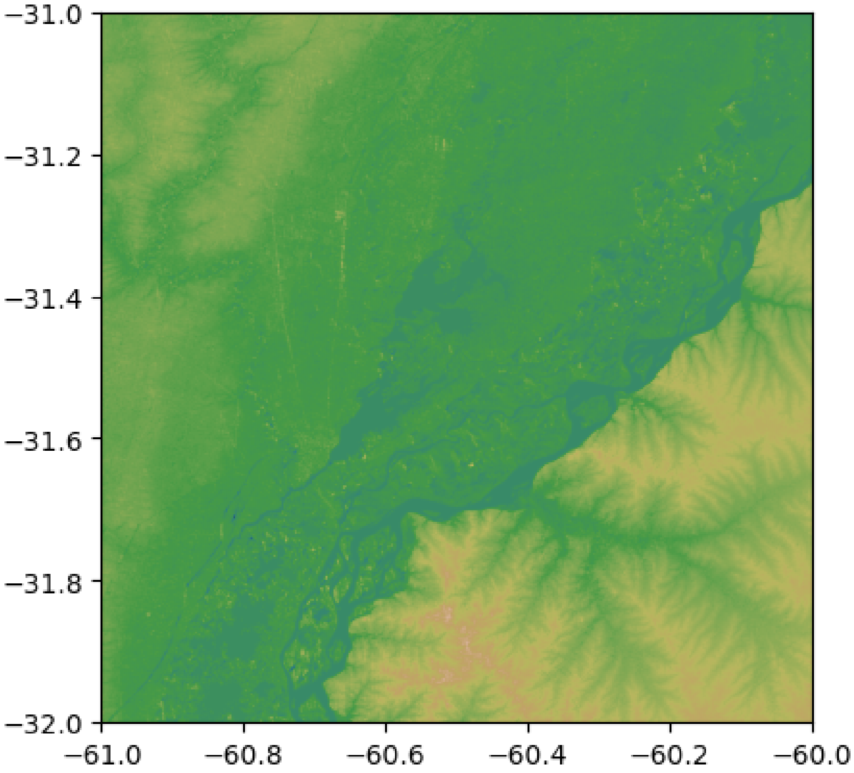
```
<matplotlib.image.AxesImage at 0x7f412fd5db20>
```



```
px.imshow(band1, x=lon, y=lat[::-1])
```



```
show(band1, cmap='gist_earth', transform=dem.transform);
```



Con una imagen Landsat

```
imglandsat_b1 = rio.open("datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_
```

```
imglandsat_b1.indexes
```

```
(1,)
```

```
datos_b1 = imglandsat_b1.read(1)
datos_b1
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=uint16)
```

```
imglandsat_b1.bounds
```

```
BoundingBox(left=606885.0, bottom=-3473715.0, right=841815.0, top=-3236985.0)
```

```
imglandsat_b1.crs
# https://epsg.org/crs_32620/WGS-84-UTM-zone-20N.html
```

```
CRS.from_epsg(32620)
```

```
imglandsat_b1.transform
```

```
Affine(30.0, 0.0, 606885.0,
       0.0, -30.0, -3236985.0)
```

```
imglandsat_b1.width
```

```
7831
```

```
imglandsat_b1.height
```

```
7891
```

```
imglandsat_b1.transform * (0, 0)
```

```
(606885.0, -3236985.0)
```

- Vamos a leer todas las bandas de la imagen: veamos el archivo de metadatos

```
bandas = 'B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11'.split()

imagenes = []

for banda in bandas:
    nom_archi = f'datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L1TP_227
    img = rio.open(nom_archi)
    imagenes.append(img)
```

```
imagenes
```

```
[<open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
 <open DatasetReader name='datos/geo/landsat/LC08_L1TP_227081_20191012_20191018_01_T1/LC08_L
```

Calculamos el NDVI

$$\text{NDVI} = \frac{NIR - R}{NIR + R}$$

In Landsat 4-7,

$$\text{NDVI} = \frac{Band4 - Band3}{Band4 + Band3}$$

.

In Landsat 8-9,

$$\text{NDVI} = \frac{Band5 - Band4}{Band5 + Band4}$$

.

```
banda4 = imagenes[3].read(1).astype('float64')
banda5 = imagenes[4].read(1).astype('float64')
banda5[5000,5000]
```

```
16662.0
```

```
nir = banda5
red = banda4
```

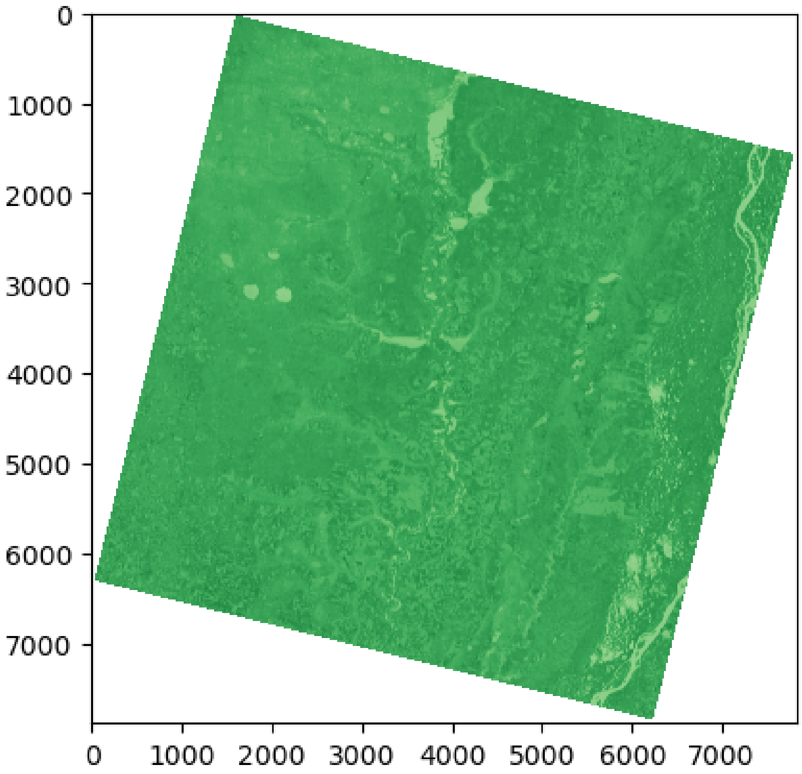
```
nume = nir - red
deno = red + nir

# >>>> ignoramos division por cero o valores inválidos <<<<
np.seterr(divide = "ignore", invalid="ignore")

ndvi = nume/deno
```

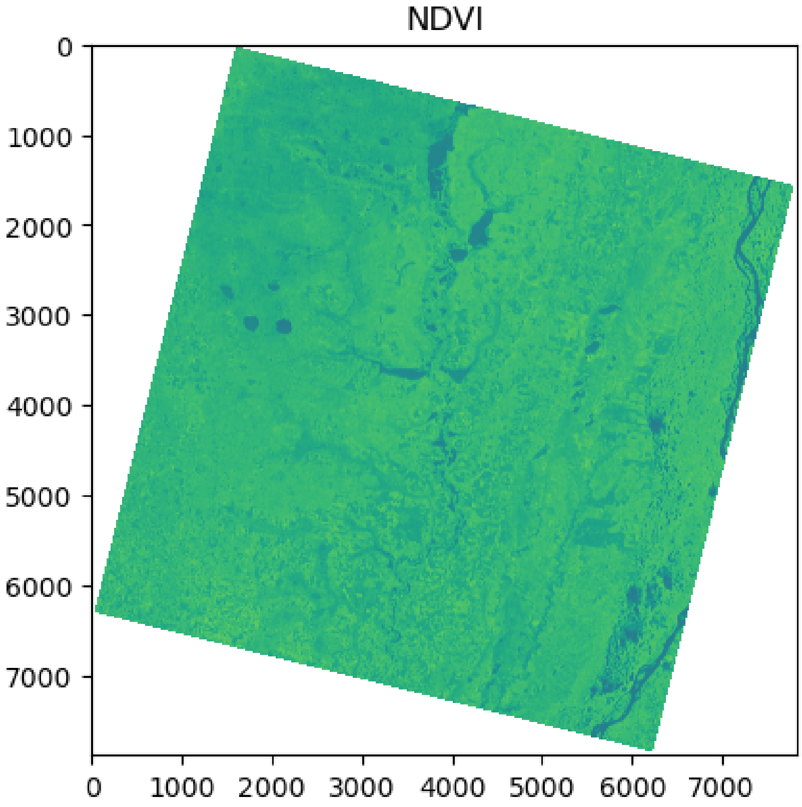
```
ndvi[ndvi > 1] = np.nan
ndvi[ndvi < -1] = np.nan
```

```
show(ndvi, cmap='Greens')
```



```
<AxesSubplot: >
```

```
# Plot raster
plt.imshow(ndvi)
plt.title("NDVI")
plt.show()
```



```
#export ndvi image
ndviImage = rio.open('ndviImage.tiff', 'w', driver='Gtiff',
                          width=imagenes[3].width, height=imagenes[3].height,
                          count=1,
                          crs=imagenes[3].crs,
                          transform=imagenes[3].transform,
                          dtype='float64'
                         )
ndviImage.write(ndvi,1) #ndvi
ndviImage.close()
```

```
np.geterr()
```

```
{'divide': 'ignore', 'over': 'warn', 'under': 'ignore', 'invalid': 'ignore'}
```