Análise de dados com Python/Jupyter

Carlos A.F Schettini 22/Dez/2021

Georeferenciamento de imagens

In [1]:
```
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
```

In [2]:
```
img_cg = plt.imread('imagem_com_grid.png')
img_sg = plt.imread('imagem_sem_grid.png')
```

In [3]:
```
plt.imshow(img_cg)
plt.show()
plt.imshow(img_sg)
plt.show()
```

In [4]:
```
print(type(img_cg))
img_cg.shape
```

```
<class 'numpy.ndarray'>
```

Out[4]:
```
(971, 1568, 3)
```

In [5]:
```
b = img_cg[:,:,0].squeeze()
print(b[:3,:3])
```

```
[[0.30588236 0.30588236 0.30588236]
 [0.3137255  0.3137255  0.3137255 ]
 [0.3019608  0.3019608  0.3019608 ]]
```

In [6]:
```
plt.imshow(b)
```

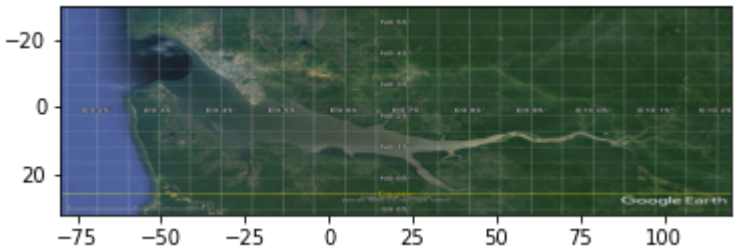
Out[6]:
```
<matplotlib.image.AxesImage at 0x1a143b3bb20>
```

In [7]:
```
linhas, colunas, _ =img_cg.shape
print(linhas, colunas)
```

```
971 1568
```

In [8]:
```
%matplotlib inline
```

```
fig, ax = plt.subplots()
ax.imshow(img_cg, extent=[-80,120,32,-30])
plt.show()
```



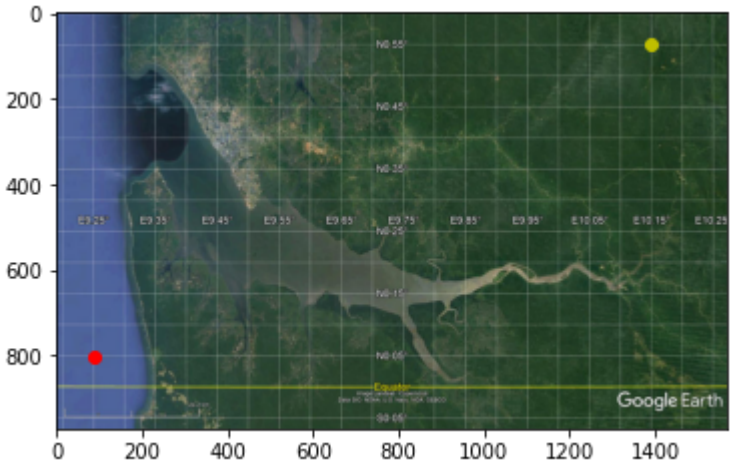
In [11]:
```
%matplotlib qt
plt.imshow(img_cg)
pontos = plt.ginput(2, timeout=-1)
plt.show()
```

In [12]:
```
print(pontos)
```

```
[(86.19660596026495, 801.7003311258279), (1390.375, 74.25413907284769)]
```

In [13]:
```
%matplotlib inline
fig, ax = plt.subplots()
ax.imshow(img_cg)
plt.plot(pontos[0][0], pontos[0][1], 'or')
plt.plot(pontos[1][0], pontos[1][1], 'oy')
```

Out[13]:
```
[<matplotlib.lines.Line2D at 0x1a14b8748b0>]
```



In [14]:
```
# pontos coordenadas pixels a partir do ginput
pt1_p = pontos[0]
pt2_p = pontos[1]

# pontos coordenadas geográficas capturados manualmnete
pt1_g = [9.25, 0.05]
pt2_g = [10.15, 0.55]
```

In [15]:
```
# equação da reta
```

```
# Y = a + b X
# b = delta Y / delta X
# a = Y - b X
```

In [16]:

```
dx_g = pt2_g[0] - pt1_g[0]
dx_p = pt2_p[0] - pt1_p[0]

b_x = dx_g / dx_p
a_x = pt1_g[0] - b_x * pt1_p[0]
```

In [17]:

```
dy_g = pt2_g[1] - pt1_g[1]
dy_p = pt2_p[1] - pt1_p[1]

b_y = dy_g / dy_p
a_y = pt1_g[1] - b_y * pt1_p[1]
```

In [21]:

```
linhas, colunas, _ = img_cg.shape

x_1 = a_x + 1*b_x
x_2 = a_x + colunas*b_x

y_1 = a_y + 1*b_y
y_2 = a_y + linhas*b_y
```

In [24]:

```
%matplotlib qt
fig, ax = plt.subplots()
ax.imshow(img_sg, extent=[x_1, x_2, y_2, y_1])
plt.show()
```