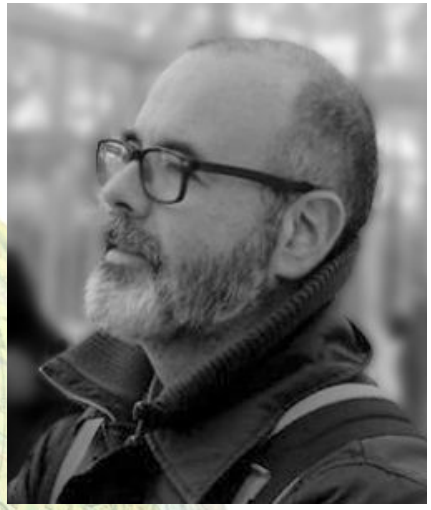
GEOPYTHON

Patricio Soriano Castro
@sigdeletras

Grupo Python Córdoba ES
21 Octubre 2019
La Zona Coworking

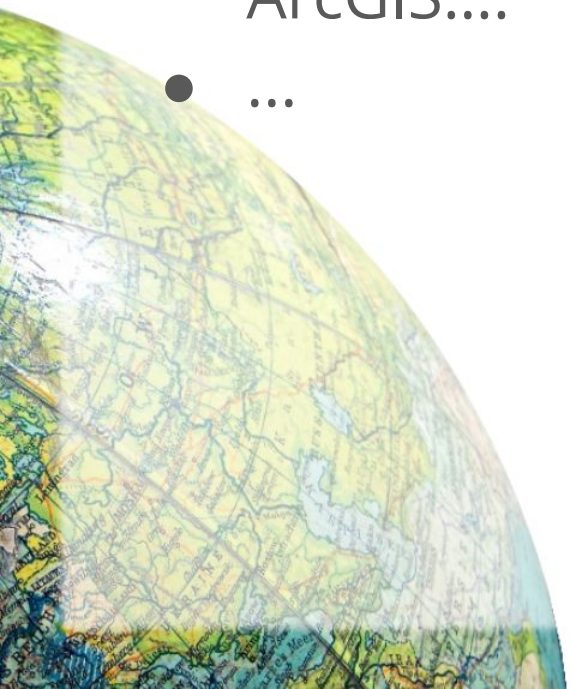
Presentación



- ✓ Patricio Soriano (@sigdeletras)
- ✓ Desarrollador NetBSS (Odoo)
- ✓ Geógrafo
- ✓ Geoinquieto “cordobé”
- ✓ sigdeletras.com

GEO para usuarios

- Sistemas de Información Geográfica: **QGIS**, ArcGIS....
- ...



GEO para usuarios

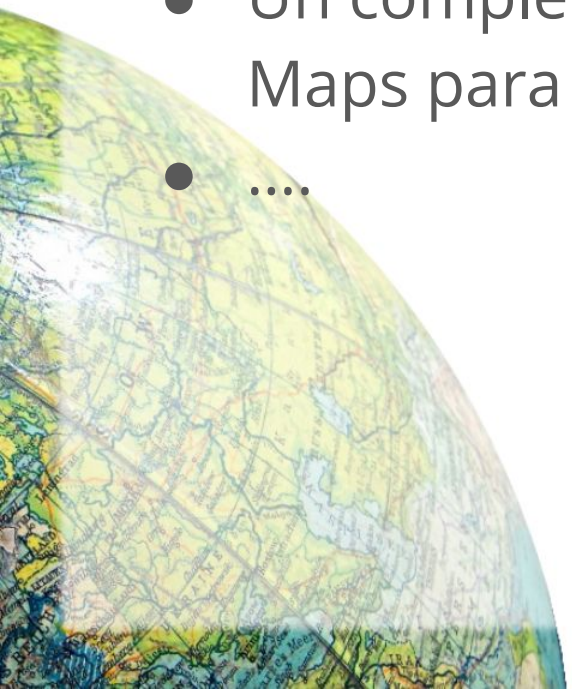
- Sistemas de Información Geográfica: **QGIS**, ArcGIS….
- Bases de datos geográficas: PostgreSQL/**PostGIS**
- Servicios web: OGC, IDEs, Geoportales, Catálogos…
- SIG en la Web. ej. **GeoWE**
- Teledetección. LiDAR. Drones
- Ciencia de datos. Machine/Deep Learning. Location Intelligence

GEO para desarrolladores

- Google Maps
- ....

GEO para desarrolladores

- Google Maps
- Un complemento de Google Maps para WordPress
- ....



GEO para desarrolladores

- Google Maps
- Un complemento de Google Maps para WordPress
- **¡¡¡¡El cliente no quiere pagar por la API de Google Maps!!!!**

  ....

GEO para desarrolladores

- Desarrollos sobre clientes GIS. Librerías geoprocesamiento, routing, topología..
- Diseño y gestión de BD geográficas. ETL
- Infraestructuras de servidores web de mapas, teselas, metadatos...
- Webmapping. Librerías JS (Leaflet, OL)
- App móviles
- ....

¿Porqué Python?

Usuarios

- Fácil de aprender (alto nivel, interpretado).
- Lenguaje de scripting en SIG (ej. PyQGIS).
- Amplia oferta de formación.
- Deep learning y data science.

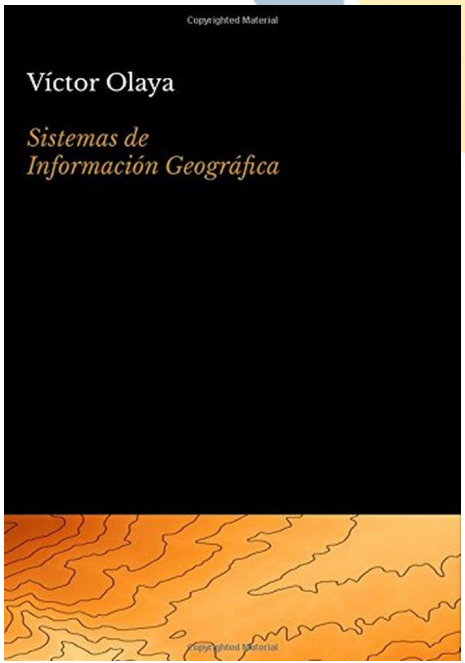
Desarrolladores

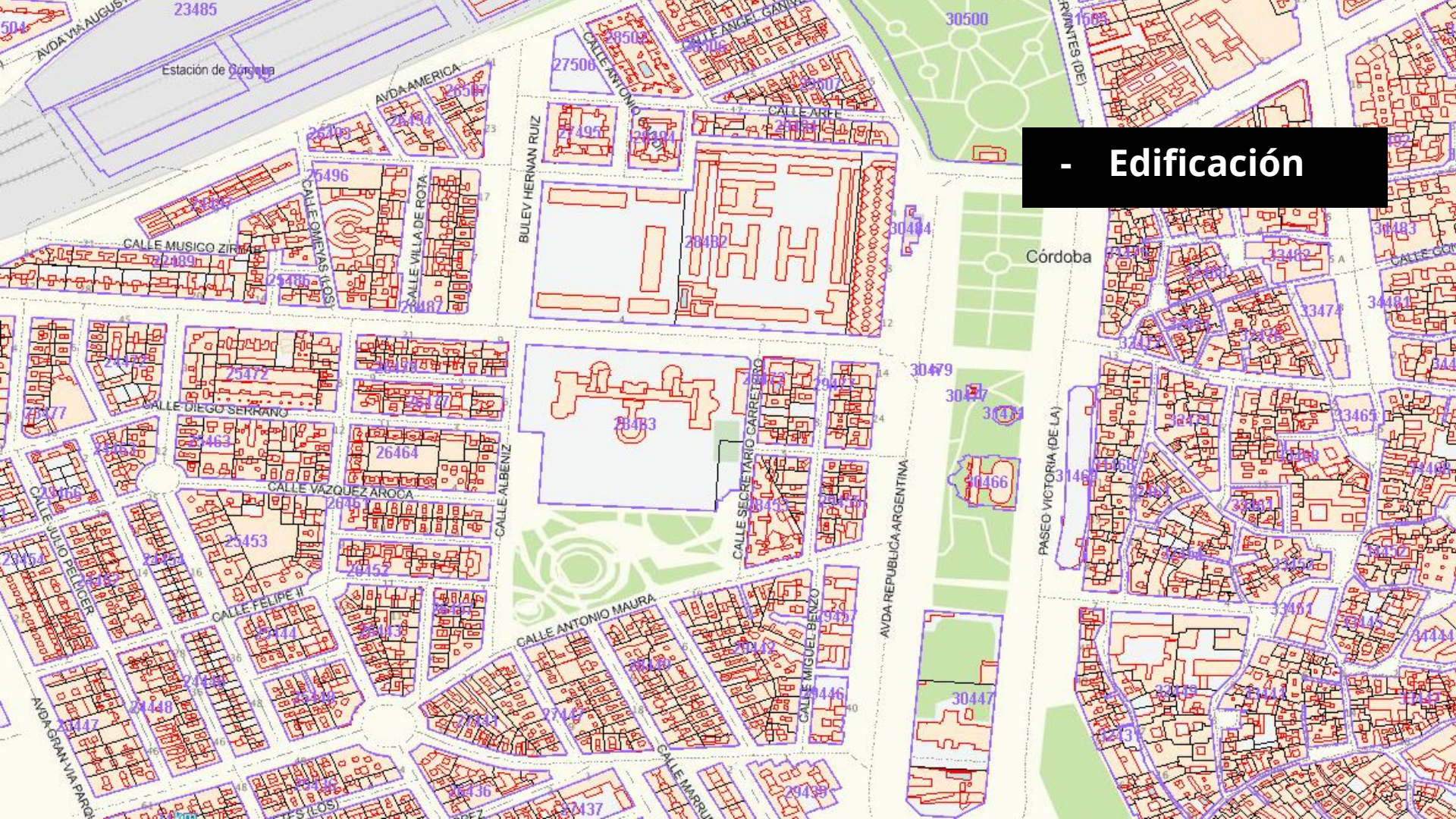
- Lenguaje maduro. Código abierto. Amplia comunidad y recursos.
- Demanda de empresas.
- Scripting, automatización de procesos.
- Framework web como Django y Flask.
- Creación de APIRestfull.

Funciones básicas

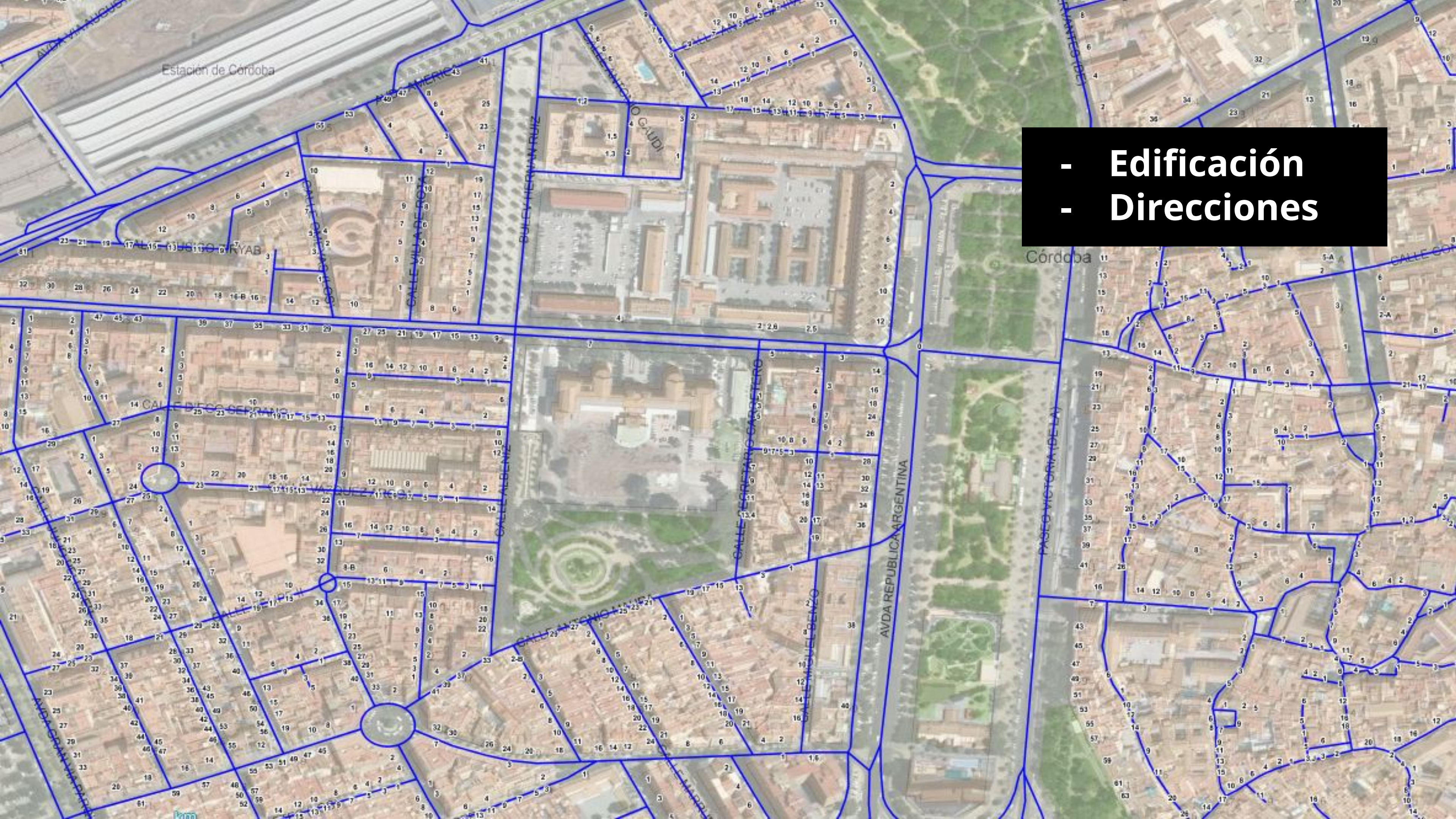
Según Víctor Olaya en su Libro Libre SIG (OLAYA 2012)

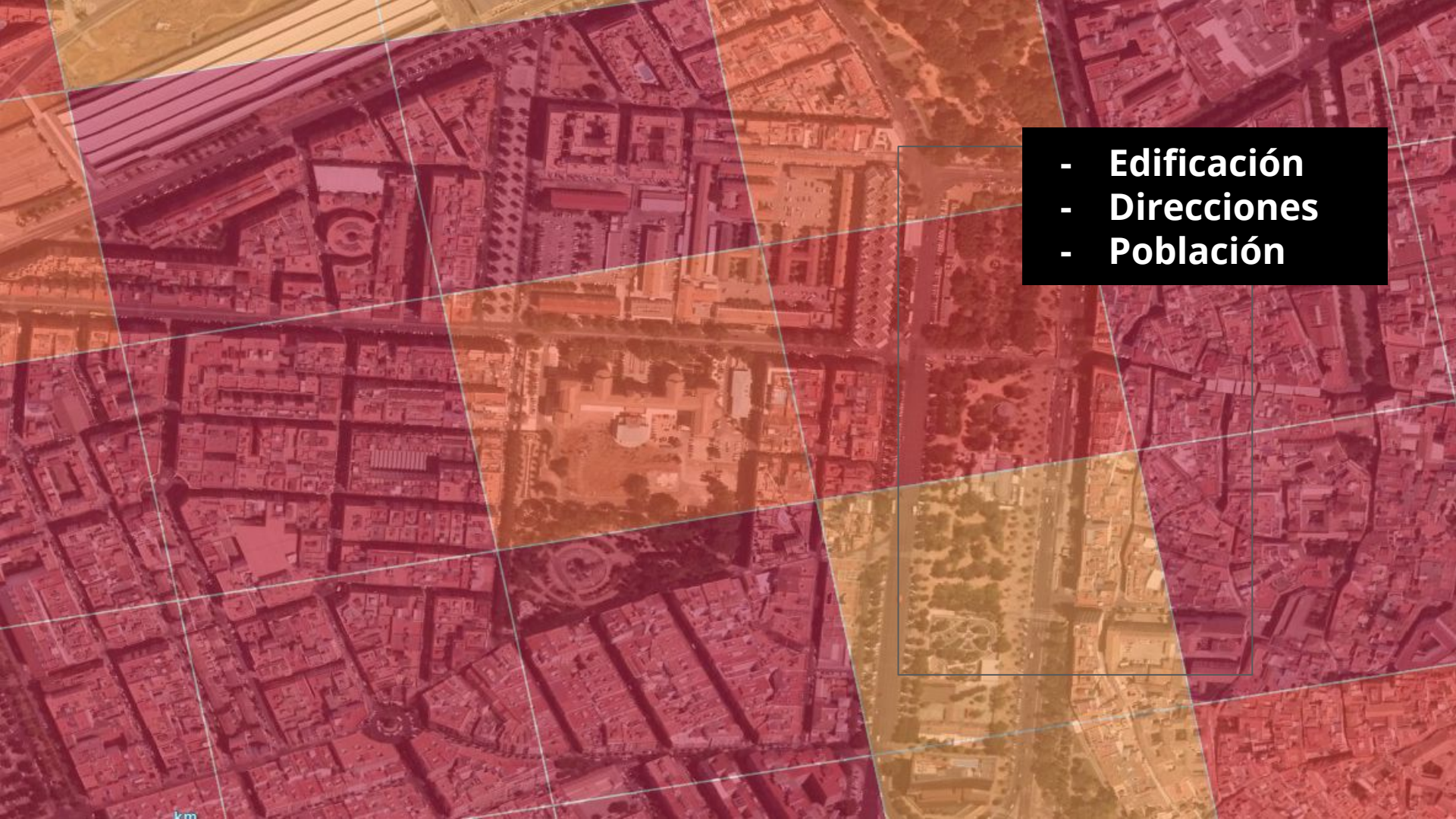
- ENTRADA/SALIDA DATOS.
- CREACIÓN/EDICIÓN.
- ANÁLISIS.
- VISUALIZACIÓN.




- Edificación
Córdoba
AVDA VIA AUGUSTA
Estación de Córdoba
AVDA AMERICA
CALLE ANTONIO CANO
CALLE ARFE
BULEV HERNAN RUIZ
CALLE MUSICO ZIRYAB
CALLE OMEYAS (LOS)
CALLE VILLA DE ROTA
CALLE DIEGO SERRANO
CALLE VAZQUEZ AROCA
CALLE ALBENIZ
CALLE SECRETARIO CARRETERO
CALLE JULIO PELUCER
CALLE FELIPE II
CALLE ANTONIO MAURA
CALLE MIGUEL BENZO
AVDA REPUBLICA ARGENTINA
PASEO VICTORIA (DE LA)
AVDA GRAN VIA PARQ
23485
30500
30484
30479
30466
30447

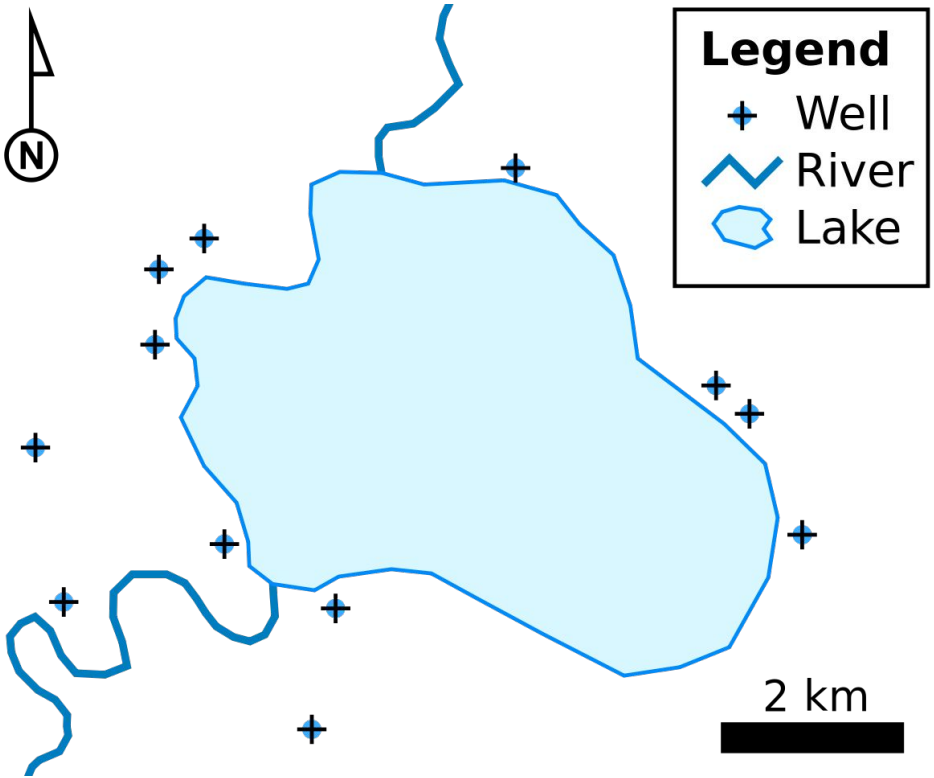
- Edificación
- Direcciones



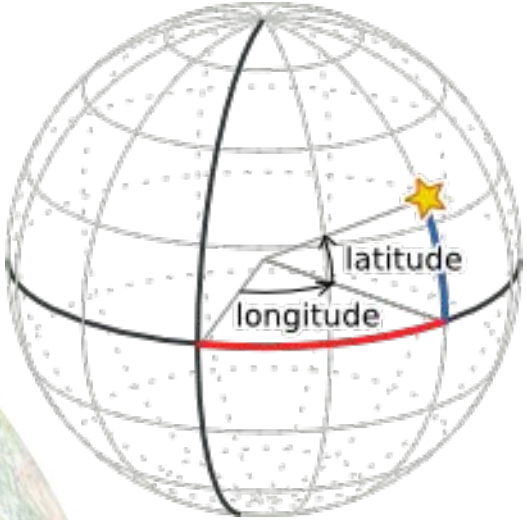

- Edificación
- Direcciones
- Población

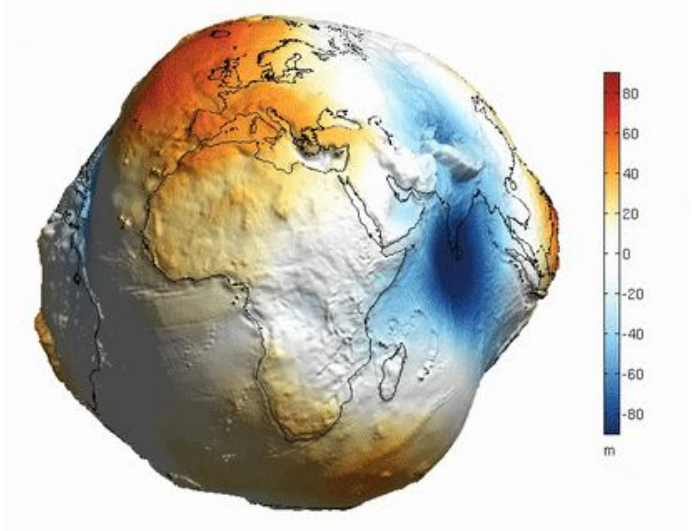

- Edificación
- Direcciones
- Población
- Usos
- Comercio
- ..

Modelo vectorial



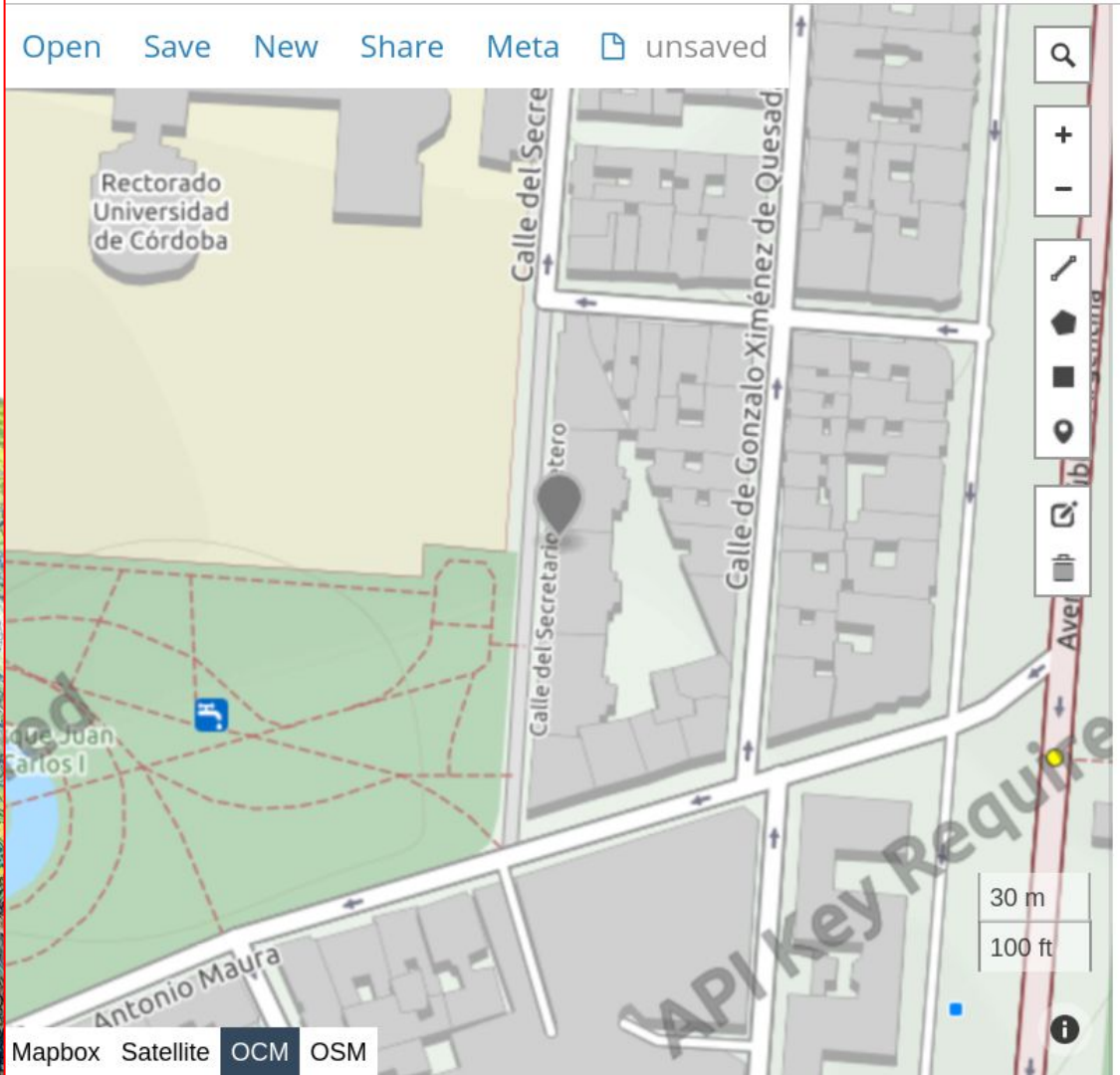
Modelo vectorial





Coordenadas: 37° 53' 0" N, 4° 46' 0" W

En decimal 37.883333°, -4.766667



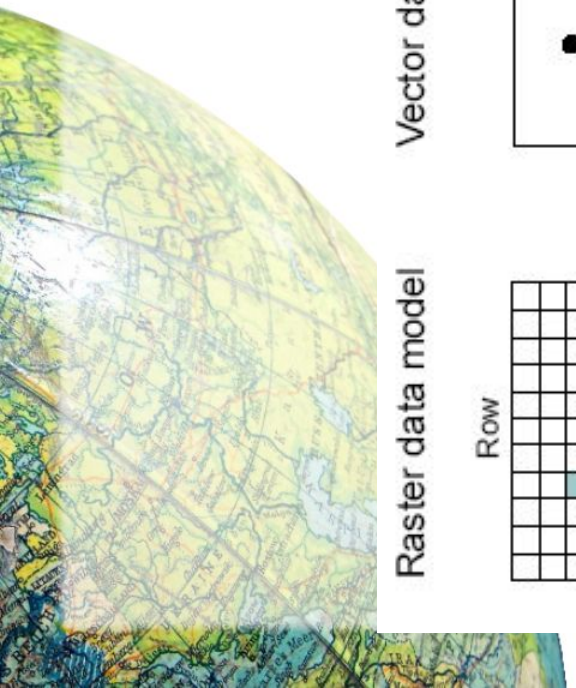
JSON | Table | Help

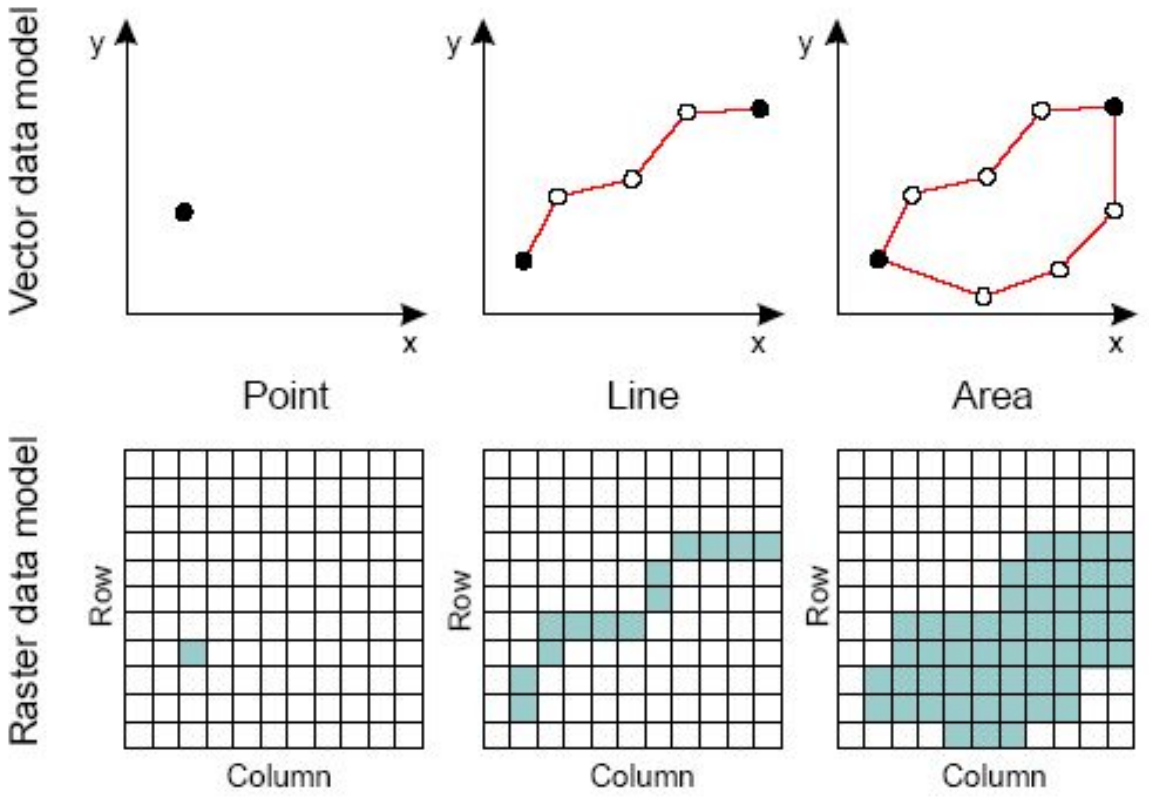
```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "name": "La Zona Coworking"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          -4.787496328353882,
          37.88363929673743
        ]
      }
    }
  ]
}
```

JSON | Table | Help

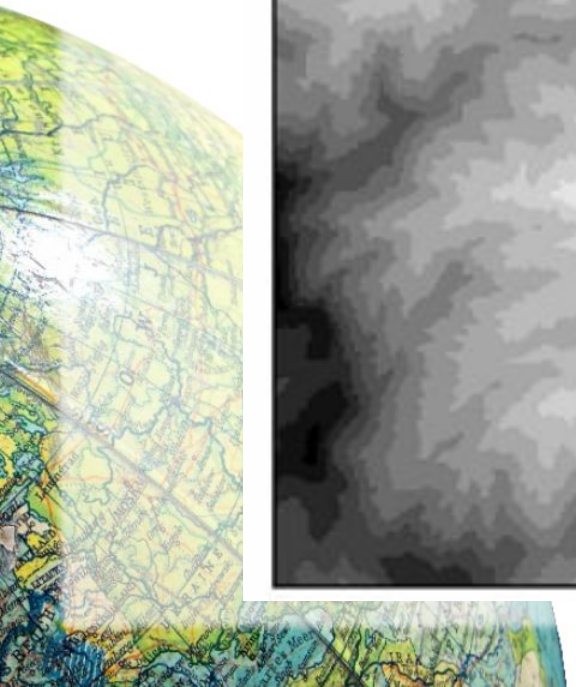
```
      "fill-opacity": 0.5
    },
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          [
            -4.786941111087801,
            37.88320108495368
          ],
          [
            -4.786525368690491,
            37.88328999770037
          ],
          [
            -4.7864744067192095,
            37.884022465290734
          ],
          [
            -4.786887466907503,
            37.88403516697659
          ],
          [
            -4.786941111087801,
            37.88320108495368
          ]
        ]
      ]
    }
  }
```

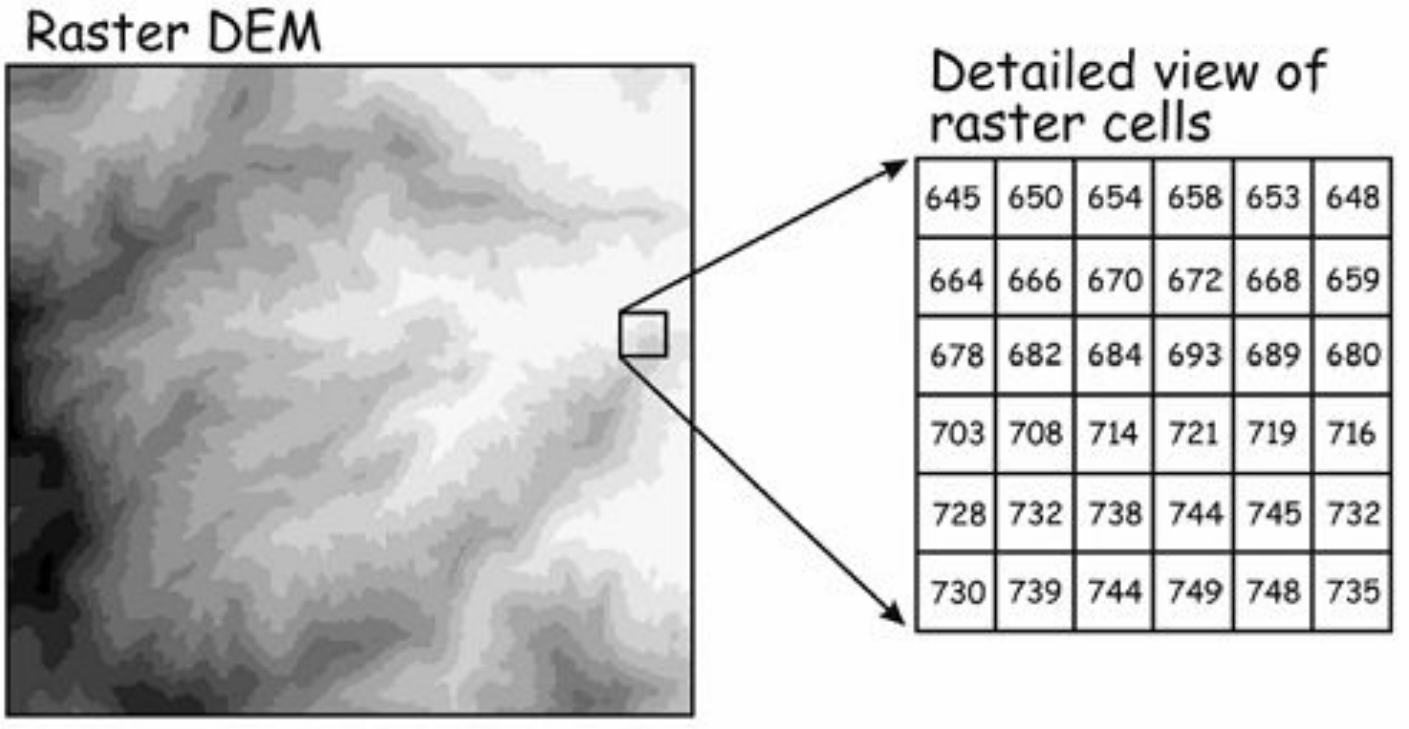
Modelo raster

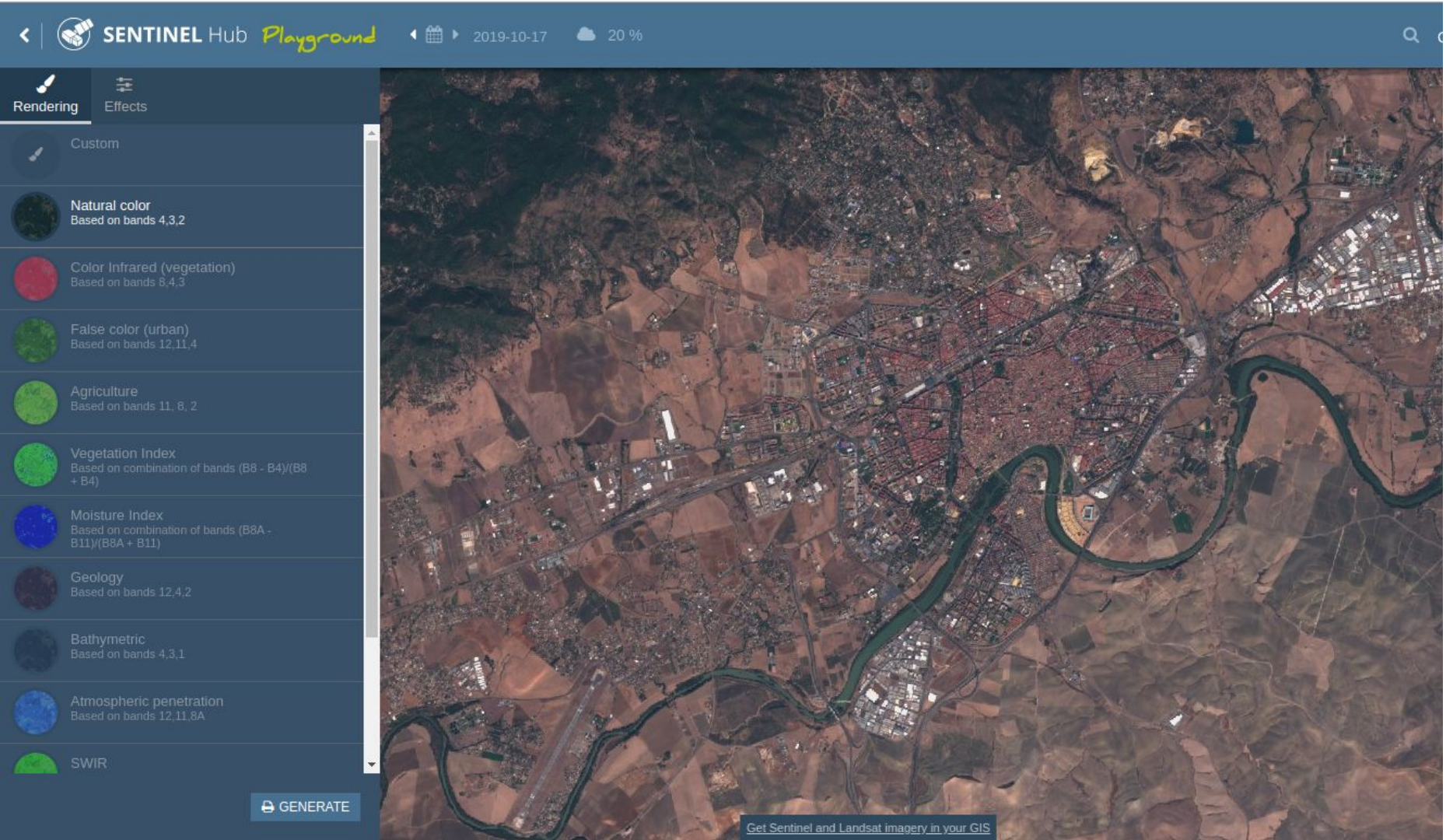




Modelo raster






SENTINEL Hub Playground
2019-10-17
20 %
Rendering
Effects
Custom
Natural color
Based on bands 4,3,2
Color Infrared (vegetation)
Based on bands 8,4,3
False color (urban)
Based on bands 12,11,4
Agriculture
Based on bands 11, 8, 2
Vegetation Index
Based on combination of bands (B8 - B4)/(B8 + B4)
Moisture Index
Based on combination of bands (B8A - B11)/(B8A + B11)
Geology
Based on bands 12,4,2
Bathymetric
Based on bands 4,3,1
Atmospheric penetration
Based on bands 12,11,8A
SWIR
GENERATE
Get Sentinel and Landsat imagery in your GIS

Rendering Effects

- Custom
- Natural color
  Based on bands 4,3,2
- Color Infrared (vegetation)
  Based on bands 8,4,3
- False color (urban)
  Based on bands 12,11,4
- Agriculture
  Based on bands 11, 8, 2
- Vegetation Index
  Based on combination of bands (B8 - B4)/(B8 + B4)
- Moisture Index
  Based on combination of bands (B8A - B11)/(B8A + B11)
- Geology
  Based on bands 12,4,2
- Bathymetric
  Based on bands 4,3,1
- Atmospheric penetration
  Based on bands 12,11,8A
- SWIR

GENERATE

Entrada/Salida de datos. Módulos

- **sys**.Funcionalidades directamente relacionadas con el intérprete.
- **os** Funcionalidades del SO
- **os.path**, **shutil** Funcionalidades relacionadas con los nombres de las rutas de archivos y directorios, mover, copiar, cortar
- **urllib.request** Peticiones HTTP
- **csv** Manejo de archivos CSV
- **zipfile** Trabajo con archivos comprimidos
- **numpy** Mejora para el trabajo con matrices y matrices multidemensionales.
- **Panda**. Estructura de datos (dataframes)
- **psycopg2**. Conexión a PostgreSQL

https://docs.python.org/3/library/index.html

Entrada/Salida de datos. Módulos geo

- **GDAL** Librería para lectura y escritura de formatos de datos geoespaciales vectoriales (OGR) y raster (GDAL)
- **Shapely** Fiona Manipulación y el análisis de datos vectoriales
- **Rasterio** Leer, manipular y escribir archivos de tipo ráster.
- **GeoPandas** Permitir el uso de archivos y operaciones espaciales.
- **pyproj** Interfaz de la librería PROJ4 de OSGeo para proyección y conversión de geometrías entre sistemas de referencia de coordenadas.
- ...

Entrada/Salida

GeoJSON

```
from urllib.request import urlopen
import json

url = 'https://raw.githubusercontent.com/sigdeletras/geopython/master/pythoncordoba2019/grid250cordoba4326.geojson'

with urlopen(url) as response:
    grid250 = json.load(response)

grid250
```


```
{'crs': {'properties': {'name': 'urn:ogc:def:crs:OGC:1.3:CRS84'},
  'type': 'name'},
 'features': [{'geometry': {'coordinates': [[[[-4.952227536148122,
        37.862209800997846],
       [-4.952720703347326, 37.8644301936189],
       [-4.949936693012141, 37.864871546080046],
       [-4.949443611313415, 37.86265113672964],
       [-4.952227536148122, 37.862209800997846]]]],
   'type': 'MultiPolygon'},
  'properties': {'gidmp': 2554,
   'municipio': 'Almodóvar del Río',
   'pob_h': -1,
   'pob_m': -1,
   'pob_tot': 8},
```

gepython_read.ipynb ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ Código + Texto

```
[11] !pip install geopandas
```

Carga de archivo shapefile con GeoPandas

```
import geopandas as gpd
munipoints = gpd.read_file("muni_pun.shp")
munipoints
```

	municipio	codine	lat	lon	altitud	geometry
0	Abla	4001	37.141333	-2.779552	843.479980	POINT (-2.77955 37.14133)
1	Abrucena	4002	37.133339	-2.796600	972.429993	POINT (-2.79660 37.13334)
2	Adra	4003	36.749285	-3.014555	4.140000	POINT (-3.01456 36.74929)
3	Albánchez	4004	37.287141	-2.181433	469.220001	POINT (-2.18143 37.28714)
4	Alboloduy	4005	37.032782	-2.621786	383.239990	POINT (-2.62179 37.03278)

```
[ ]  !pip install rasterio
```

Abrir raster

```
[ ]  import rasterio
     fp = "cordobapnoa4326.tif"
     raster = rasterio.open(fp)
```

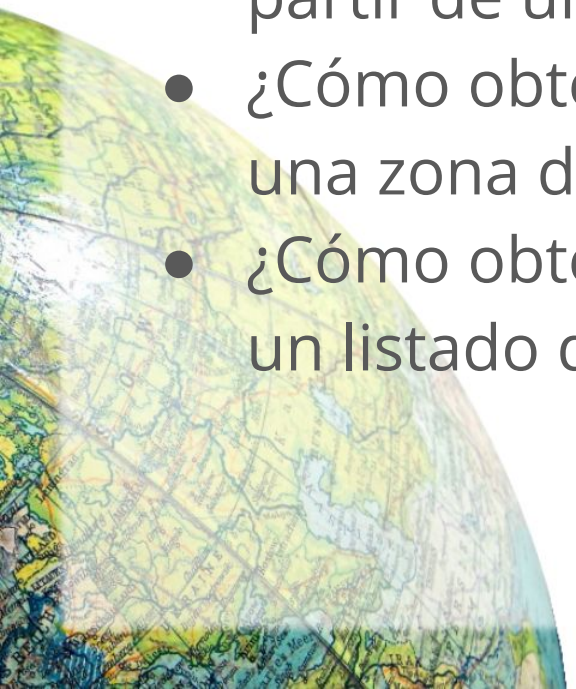
Lectura de metadatos

```
[ ]  # All Metadata for the whole raster dataset
     raster.meta
```

```
{'count': 4,
 'crs': CRS.from_epsg(4326),
 'driver': 'GTiff',
 'dtype': 'uint8',
 'height': 571,
 'nodata': None,
 'transform': Affine(8.227016910469686e-06, 0.0, -4.791206249490818,
       0.0, -8.227016910469686e-06, 37.88564203153964),
 'width': 1283}
```

Creación/Edición. Scripting

- ¿Cómo recortar (clip) varias archivos geográficos a partir de una determinada área?
- ¿Cómo obtener una serie temporal de imágenes de una zona de servicios de mapas web?
- ¿Cómo obtener las coordenadas (geocodifiación) de un listado de direcciones postales?



clipShapesZip.py

```
36
37
38  def clipShapesZip(zipfile, clipshape, dirclip):
39
40      path = dirclip
41
42      if not os.path.exists(path):
43          os.makedirs(path)
44      else:
45          print('Ya existe una carpeta \'%s\' en el directorio.' % (dirclip))
46          print('Bórrela o indique otro nombre.')
47          sys.exit()
48
49      with ZipFile(zipfile) as myzip:  # Descomprime zip
50          myzip.extractall()
51      zipfilefolder = zipfile[0:-4:]  # Carpeta zip
52      print('Descompresión de archivos en carpeta \'%s\' \n' % (zipfilefolder))
53      for shapefile in os.listdir(zipfilefolder):
54          if shapefile[-4:] == ".shp":
55              shapefilename = shapefile[0:-4:]
56              clipshapefilename = ''
57              clipshapefilename = shapefilename+'_clip.shp'
58              ogrclip = 'ogr2ogr -clipsrc %s %s %s' % (clipshape, path+'/'+clipshapefilename, zipfilefolder+"/"+shapefile)
59              os.system(ogrclip)  # Clip
60              print("Capa %s recortada en la carpeta %s" % (
61                  clipshapefilename, dirclip))
62      shutil.rmtree(zipfilefolder)
63      print('\nElimimnación de carpeta\'%s\'' % (zipfilefolder))
64
65  if __name__ == '__main__':
66      zipfile = sys.argv[1]     # Archivo ZIP
67      clipshape = sys.argv[2]  # Archivo DXF de origen
68      dirclip = sys.argv[3]    # Archivo GML de salida
69      clipShapesZip(zipfile, clipshape, dirclip)
70
```

https://github.com/sigdeletras/clipShapesZip

wms2image.py

```
60  def wms2image(csvfile, code, bbox, imgwidth, imgformat):
61      if code not in EPSG_ZONES:
62          return (
63              False,
64              u'Error: El código EPSG "%s" es incorrecto' % code
65          )
66          sys.exit(1)
67
68      if imgformat not in IMAGE_FORMAT:
69          return (
70              False,
71              u'Error: El formato "%s" es incorrecto' % imgformat
72          )
73          sys.exit(1)
74
75      wmslist = csv.reader(open(csvfile, "rt"), delimiter=',')
76
77      bboxcoord = ast.literal_eval(bbox)
78      imgheight = ((bboxcoord[3] - bboxcoord[1]) * int(imgwidth)) / (bboxcoord[2] - bboxcoord[0])
79      epgscode = str(code)
80      for row in wmslist:
81          wms = row[0]
82          layer = row[1]
83          file = row[2]
84          url = wms + '?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&LAYERS=' + layer + '&STYLES=&SRS=EPSG:' + epgscode + '&BBOX=' \
85              + bbox + '&WIDTH=' + str(imgwidth) + '&HEIGHT=' + str(imgheight) + '&FORMAT=image/' + imgformat
86
87          r = requests.get(url)
88          with open(file + "." + imgformat, "wb") as code:
89              code.write(r.content)
90
91          print("Downloading..." + file + "." + imgformat)
92          print(url)
93
```

https://github.com/sigdeletras/wms2image

https://github.com/sigdeletras/wms2image

geopyton_ej._geocoder.ipynb ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Guardado por última vez: 15:05

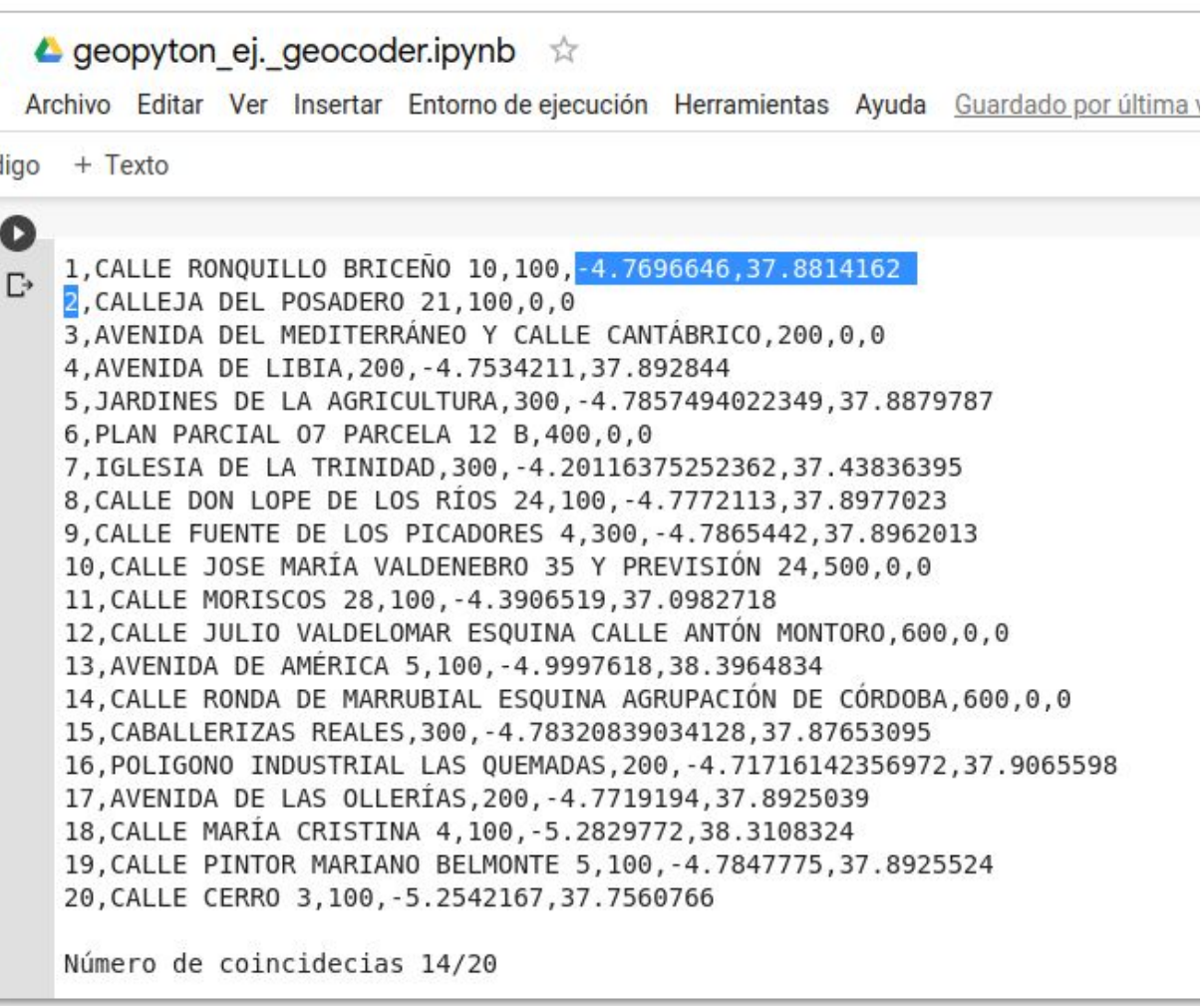
+ Código + Texto

```
[ ]  import csv
     import json

     import geocoder

     geojson = {
         'type': 'FeatureCollection',
         'features': []
     }
     report_noloc = ''
     results = 0
     total = 0

     with open('csv_in.csv', encoding="utf8") as csv_file:
         csv_data = csv.reader(csv_file, delimiter=',', quotechar='\'')
         next(csv_data) # skip header

         for row in csv_data:
             geodir = geocoder.osm('{}, Córdoba, Andalucía, España'.format(row[1]))
             total += 1
             if geodir:
                 print('{},{},{},{},{}'.format(
                     int(row[0]), row[1], row[2], geodir.latlng[1], geodir.latlng[0]))
                 geojson['features'].append({
                     'type': 'Feature',
                     'geometry': {
                         'type': 'Point',
                         'coordinates': [geodir.latlng[1], geodir.latlng[0]],
                     },
                     "properties": {
                         "id": row[0],
                         "direccion": row[1],
                         "tipo": row[2],
                     }
                 })
                 results += 1
             else:
                 print('{},{},{},0,0'.format(int(row[0]), row[1], row[2]))
                 report_noloc += '{},{},{}\n'.format(int(row[0]), row[1], row[2])

         print('\nNúmero de coincidecias {}/{}'.format(results, total))
```

Geocodificación

Geocoder

- OpenStreetMap
- Google
- HERE
- Yahoo
- ...



Geocodificación

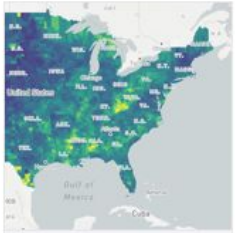
- csv
- json
- geocoder

[Notebook Google Colab]

Visualización

Fundamental el uso de intérpretes interactivos con **Jupyter Notebook, Anaconda, Google Colaboratory.**

- **Matplotlib** Visualizaciones 2D. Gráficos de barras, ternarios, de líneas, temporales, diagramas de dispersión... ¡y también mapas!
- **Ploty** Librería de de gráficos para crear gráficos interactivos



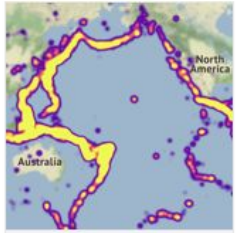
Mapbox Choropleth Maps

Lines on Mapbox

Filled Area on Maps

Scatter Plots on Maps

Bubble Maps



Mapbox Density Heatmap

Lines on Maps

Choropleth Maps

Mapbox Map Layers

Scatter Plots on Mapbox

USA County Choropleth Maps

geopython_raster.ipynb

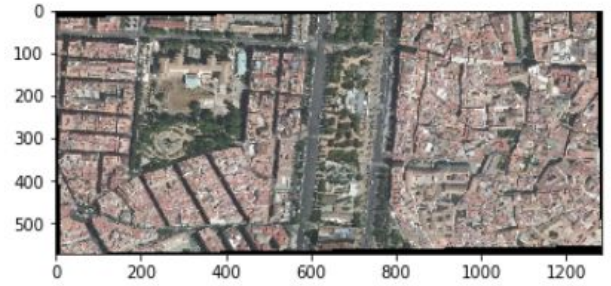
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuc

+ Código + Texto

[11]
```
# Lectura de las bandas
red = raster.read(1)
green = raster.read(2)
blue = raster.read(3)

# Función para la normalización de los valores
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))

# Variables de normalización
redn = normalize(red)
greenn = normalize(green)
bluen = normalize(blue)
```

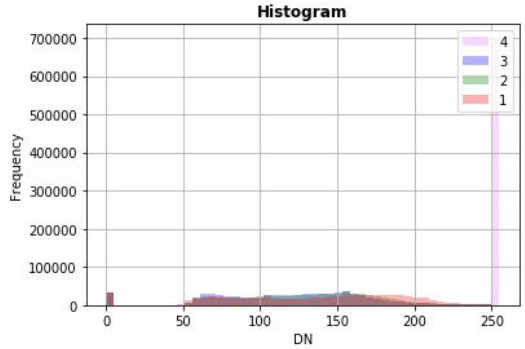
[12]
```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

# Creación de composición RGB color natural
rgb = np.dstack((redn, greenn, bluen))

# Resultado
plt.imshow(rgb)
```

```
<matplotlib.image.AxesImage at 0x7fce170d3128>
```



Historiograma

```
from rasterio.plot import show_hist

show_hist(raster, bins=50, lw=0.0, stacked=False, alpha=0.3,
      histtype='stepfilled', title="Histogram")
```



geopython_ej_maptolib.ipynb ☆

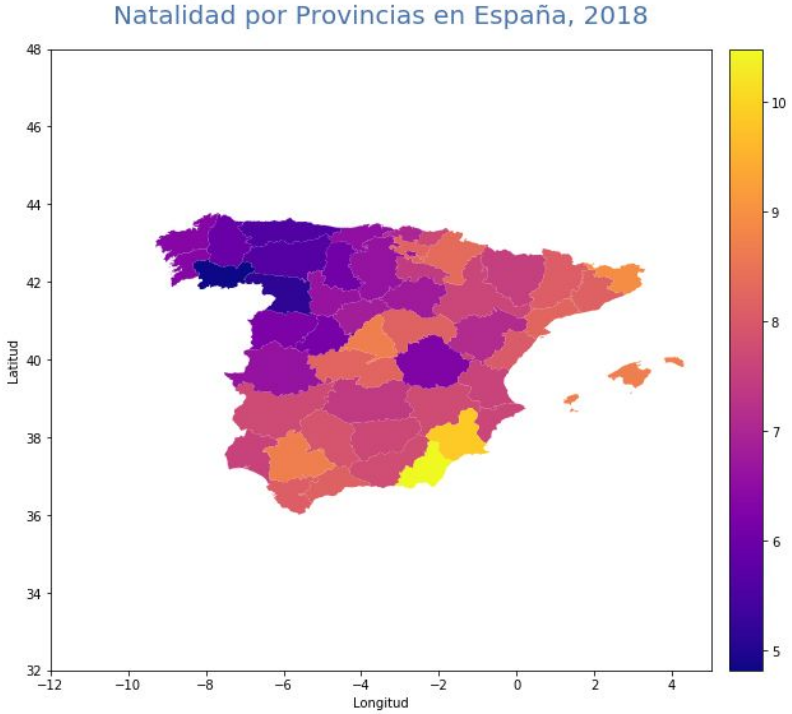
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ Código + Texto

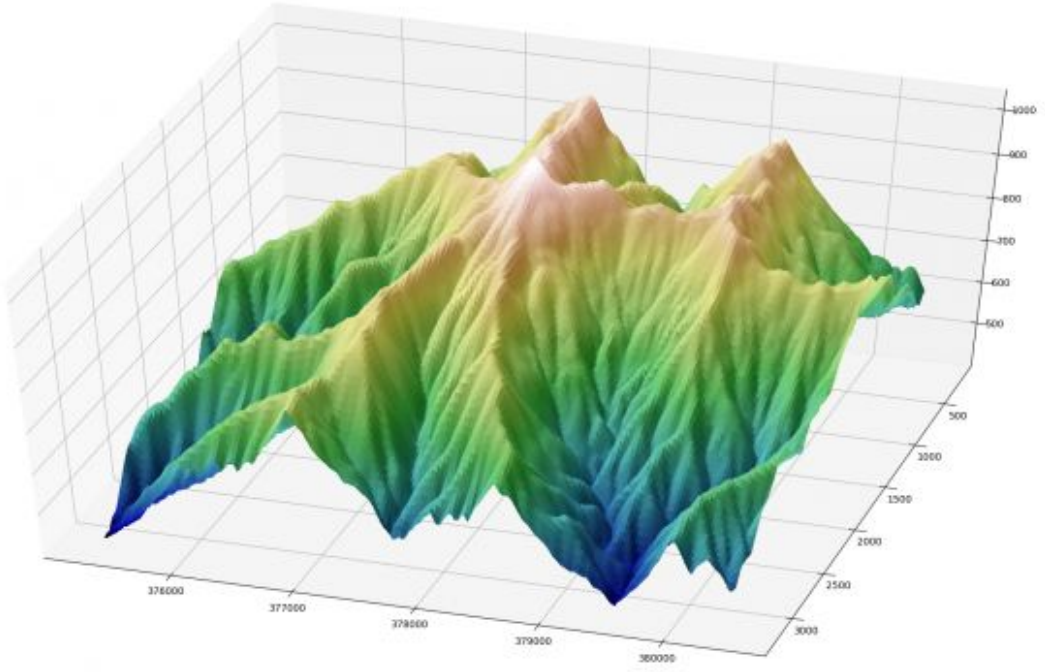
Añadimos info

```
[ ]  # Control del tamaño de la figura del mapa
     fig, ax = plt.subplots(1, 1, figsize=(10, 10))

     # Control del encuadre (área geográfica) del mapa
     ax.axis([-12, 5, 32, 48])

     # Control del título y los ejes
     ax.set_title('Natalidad por Provincias en España, 2018',
                  pad = 20,
                  fontdict={'fontsize':20, 'color': '#4873ab'})
     ax.set_xlabel('Longitud')
     ax.set_ylabel('Latitud')

     # Añadir la leyenda separada del mapa
     from mpl_toolkits.axes_grid1 import make_axes_locatable
     divider = make_axes_locatable(ax)
     cax = divider.append_axes("right", size="5%", pad=0.2)

     # Generar y cargar el mapa
     map_data.plot(column='NAT2018', cmap='plasma', ax=ax,
                   legend=True, cax=cax, zorder=5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7198fabcc0>
```



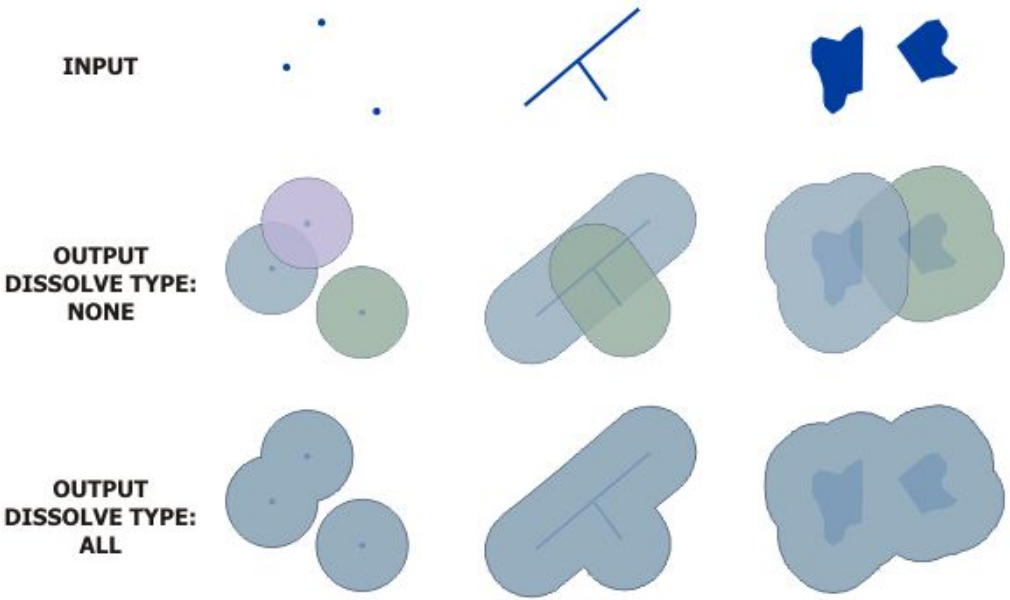
Visualización



Fuente: https://www.cursosgis.com/

Análisis

Geoprocesos: extracción, superposición, proximidad... **GDAL, Shapely y por comodidad GeoPandas**

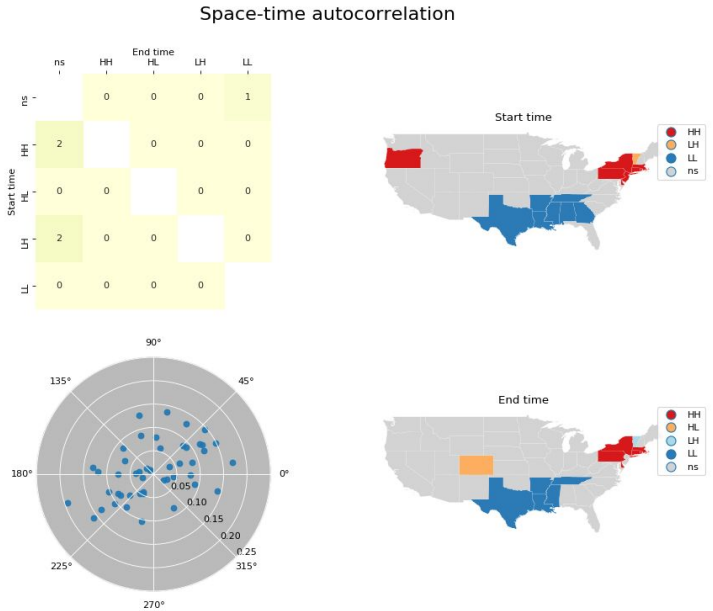


Fuente: https://pro.arcgis.com/

Análisis

PySAL

- Análisis de correlación espacial. Detección de agrupaciones (cluster), hot-spots, valores atípicos
- Construcción de gráficas a partir de datos espaciales.
- Spatial regression and statistical modeling on geographically embedded networks
- Análisis espacio-temporales



Análisis

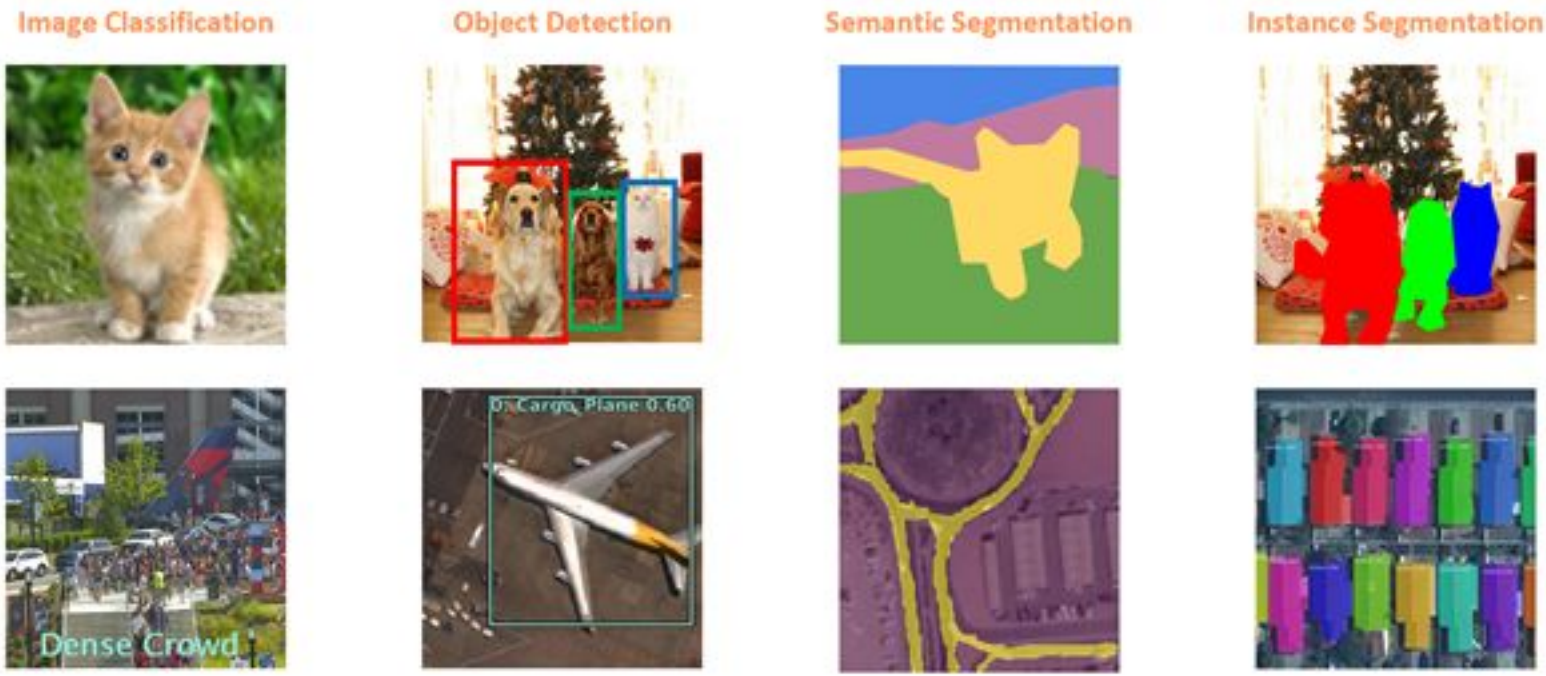
Deep learning. Ejemplo de aplicación en el tratamiento de grande volúmenes de imágenes (aéreas, teledetección...)

- **Clasificación:** ¿qué tipo de cobertura de superficie se observa en una imagen de satélite?
- **Detección:** Detectar árboles en imágenes de drones
- **Segmentación:** ¿qué píxeles pertenecen a un edificio y cuáles no?

Librerías Keras, TensorFlow, Pytorch

Fuente: https://www.cursosgis.com/

Análisis



Fuente: https://medium.com/geoai/integrating-deep-learning-with-gis-70e7c5aa9dfe

Web. Folium

Folium Visualización y generación de mapas usando Leaflet

- Mapas base
- Marcadores (Icons)
- GeoJSON
- Controles de Leaflet
- Mapas temáticos

Creación de mapa con capa base

```
[9]  import folium

     map = folium.Map(
         location=[ 37.88366470023913,-4.787523150444031],
         zoom_start=16,
         tiles='Stamen Terrain'
     )

     map
```

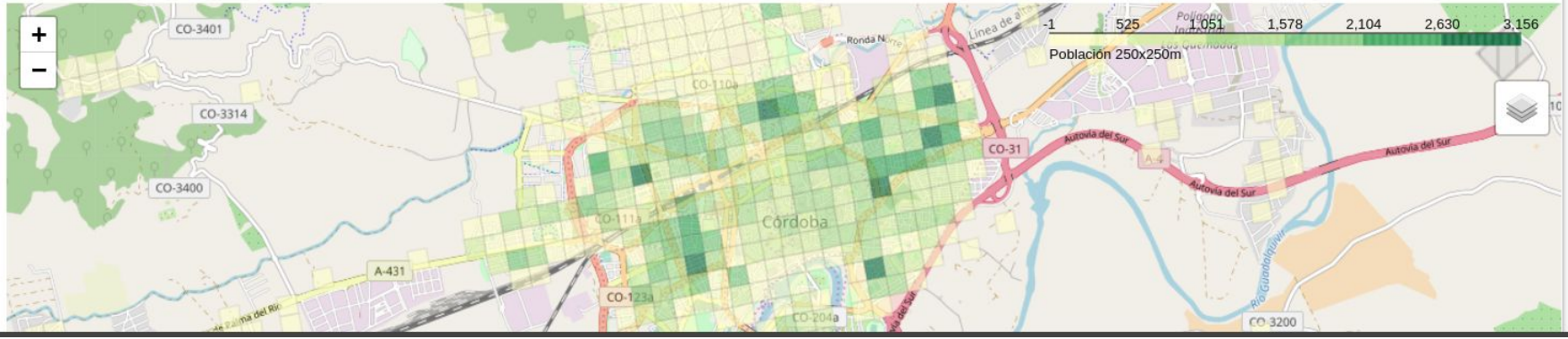
Notebook

+ Código + Texto

RAM Disco

Editar

```
import pandas as pd

url ='https://raw.githubusercontent.com/sigdeletras/geopython/master/pythoncordoba2019'

geo = f'{url}/grid250cordoba4326.geojson'
pop = f'{url}/pob.csv'

pop_data = pd.read_csv(pop)

m = folium.Map(location=[37.88366470023913,-4.787523150444031], zoom_start=12)

folium.Choropleth(
    geo_data=geo,
    name='Población',
    data=pop_data,
    columns=['gidmp','pob_tot'],
    key_on='feature.properties.gidmp',
    fill_color='YlGn',
    fill_opacity=0.6,
    line_opacity=0.1,
    legend_name='Población 250x250m'
).add_to(m)

folium.LayerControl().add_to(m)

m
```



Notebook

Geodjango

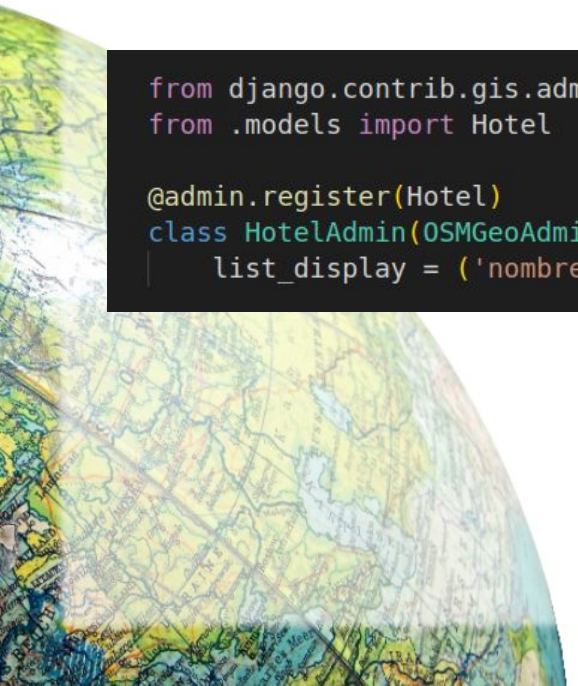
Geodjango una expansión **Django** que permiten almacenar y manipular datos geográficos. Herramienta enfocada a la creación de aplicaciones web geográficas de forma rápida y sencilla.

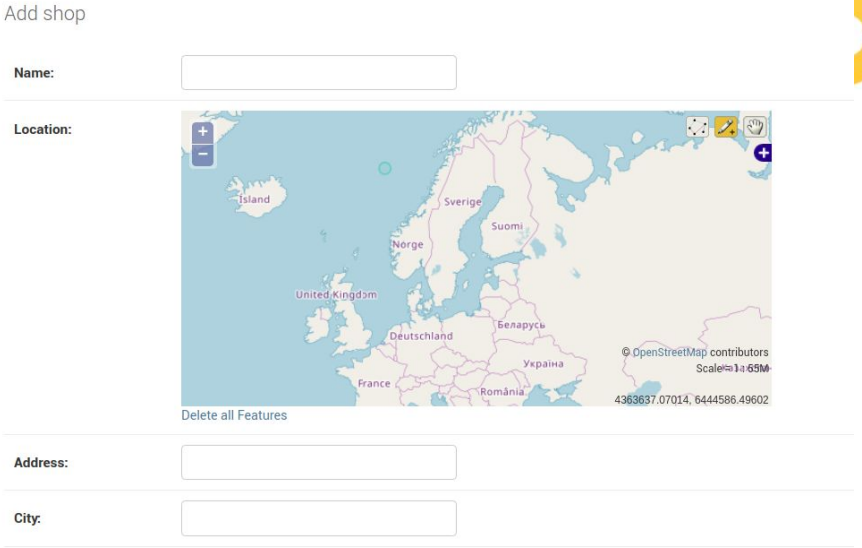
- BBDD Geo (PostGIS, MySQL, Oracle, SpatiaLite)
- Administrador
- Migraciones
- Extensión del ORM con funciones Geo

```python
class Hotel(models.Model):
    nombre = models.CharField(max_length=100)
    localizacion = models.PointField()
    direccion = models.CharField(max_length=100)
    ciudad = models.CharField(max_length=50)
```

Geodjango

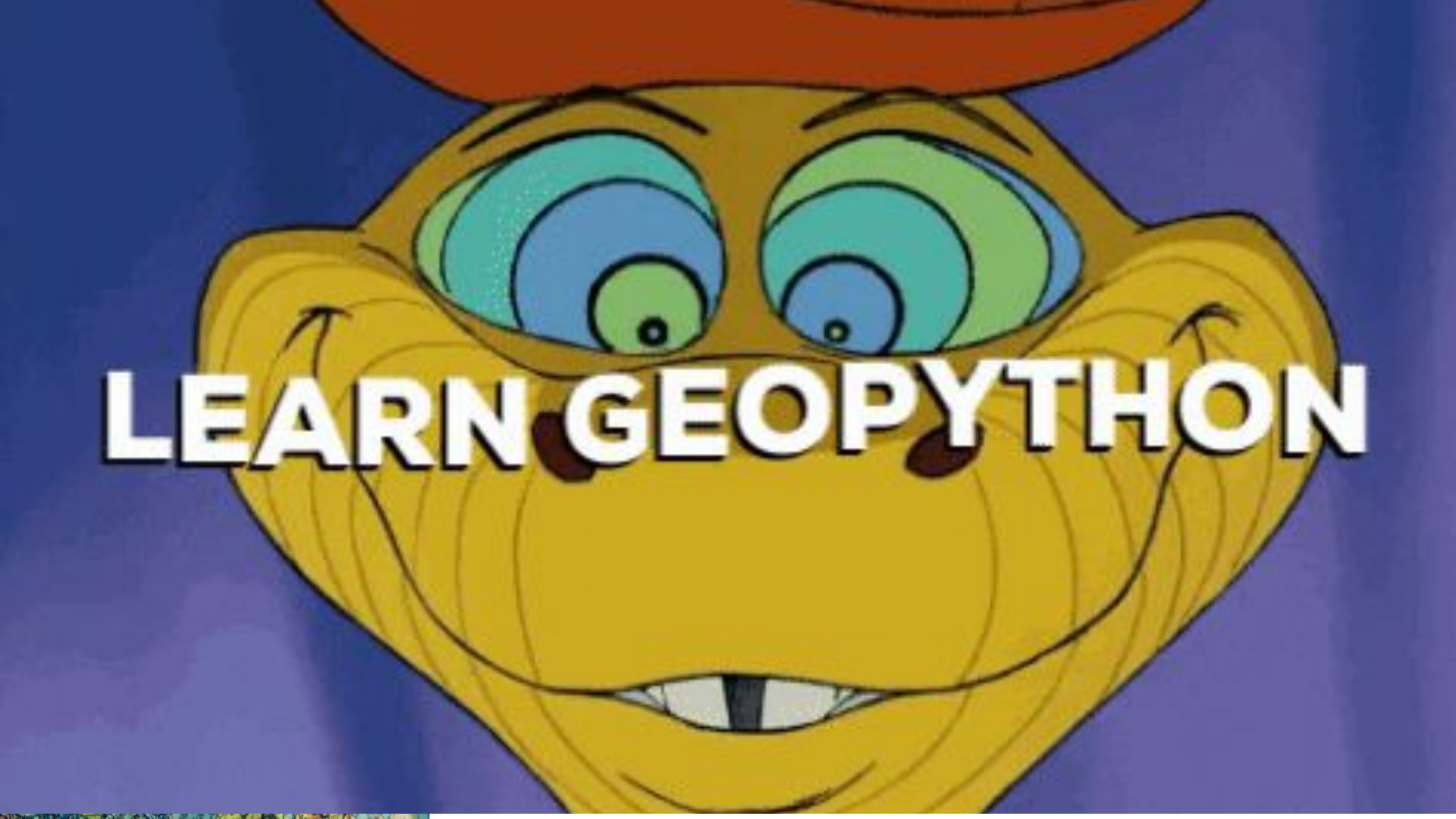
```python
from django.contrib.gis.admin import OSMGeoAdmin
from .models import Hotel

@admin.register(Hotel)
class HotelAdmin(OSMGeoAdmin):
    list_display = ('nombre', 'localizacion')
```





Algunas conclusiones...

- Entrada y salida de geo(datos) mejor SIG.
- Python como navaja suiza para scripting, sin olvidar desarrollos de plugins.
- Visualización de mapas (SIG). Gráficas y estadísticas (Complementos SIG y Python).
- Muy potente (más R) para geoestadística y Deep Learnig.
- Ojo a GeoDjando para aplicaciones Web.


LEARN GEOPYTHON

GEOPYTHON

Patricio Soriano Castro
@sigdeletras

Grupo Python Córdoba ES
21 Octubre 2019
La Zona Coworking