

1- Configurando ambiente

☰ Tags

Configurando ambiente Python para análise geoespacial

Introdução

Análise de dados geográficos é um campo de rápido crescimento que envolve o uso de Sistemas de informação Geográfica (SIG) para analisar e extrair valor de dados geoespaciais. O Python é uma linguagem de programação poderosa amplamente utilizada em análises geo devido à sua flexibilidade, à quantidade crescente de bibliotecas e comunidade muito ativa.

Neste blog, nós discutiremos como configurar um ambiente Python para análise de dados geoespaciais, desde a criação de um ambiente virtual, a instalação das bibliotecas Geo e a configuração do IDE.

Uma dica interessante para quem está iniciando os estudos é não começar com as bibliotecas de baixo nível como o GDAL, mas sim utilizar bibliotecas de alto nível, i.e. aquelas que abstraem detalhes para facilitar a utilização das funcionalidades mais utilizadas.

Instalar o Miniconda e criar o Ambiente Virtual

O primeiro passo para configurar o ambiente Python para análises geoespaciais é a instalação do Python em si. Porém, como o sistema operacional (OS) do seu computador utiliza o python para diversas funções, é importante criar um ambiente virtual para não corromper o Python usado pelo OS. Uma boa pratica é criar um ambiente virtual para cada projeto.

Um ambiente virtual é um ambiente Python isolado que lhe permite instalar pacotes e dependências específicas para cada projeto sem afetar o Python instalado com o sistema. Para criar um ambiente virtual precisamos de uma ferramenta de gerenciamento de bibliotecas e versões do Python.

O Anaconda é um serviço de gestão de bibliotecas muito utilizado pelos cientistas de dados, mas, por vir com muitas bibliotecas pré instaladas fica muito pesado e consome muito espaço em disco, por isso recomendo utilizar a versão “lite”, o **miniconda**.

Para instalar o miniconda basta seguir os passos no site oficial (<https://docs.conda.io/en/latest/miniconda.html>).

Após a instalação do miniconda o comando conda será habilitado no seu terminal de comando (prompt). Para criar um ambiente virtual, abra o prompt e siga as etapas:

1. Criar um ambiente virtual

```
conda create -n geos
```

2. Ativar o ambiente virtual

```
conda activate geos
```

3. Verifique as bibliotecas já instaladas

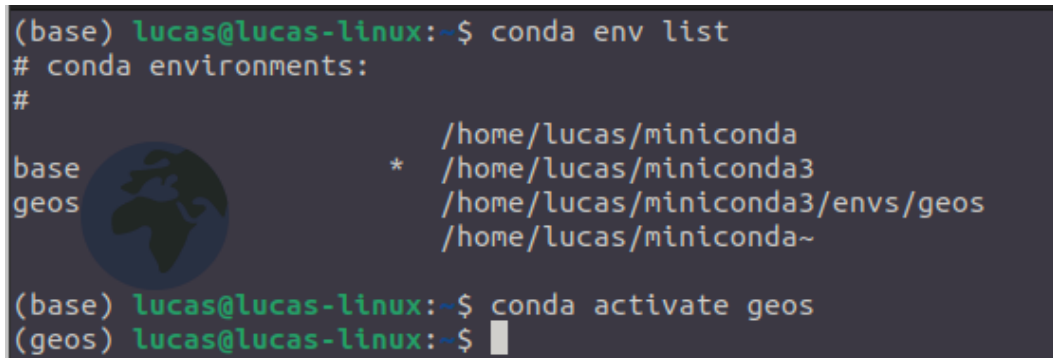
```
conda list
```

Se tiver instalado o conda pelo miniconda o ambiente criado deverá estar vazio, isto é sem bibliotecas.

Uma das vantagens em se utilizar o conda ao invés do pip é que o conda lida melhor com dependências entre bibliotecas, como por exemplo o GDAL que é base para a maioria das aplicações geo de código aberto ou livre, e que costuma causar muita dor de cabeça para ser instalado corretamente.

Bibliotecas Geo

Uma vez configurado o ambiente virtual, certifique-se de estar neste ambiente (o nome do ambiente vai aparecer no prompt entre parenteses).



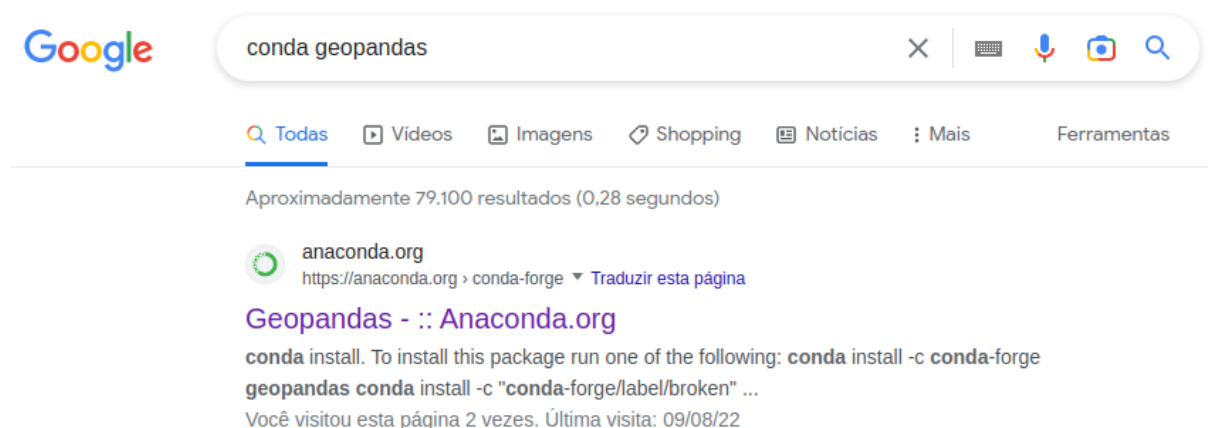
```
(base) lucas@lucas-linux:~$ conda env list
# conda environments:
#
base          *  /home/lucas/miniconda
geos          *  /home/lucas/miniconda3
              /home/lucas/miniconda3/envs/geos
              /home/lucas/miniconda~

(base) lucas@lucas-linux:~$ conda activate geos
(geos) lucas@lucas-linux:~$
```

Dentro do ambiente virtual recém criado você pode instalar as bibliotecas necessárias utilizando o comando `conda install`. Algumas das bibliotecas Geo mais populares são:

1. GeoPandas: Para trabalhar com dados vetoriais, como pontos e polígonos.
2. Shapely: Biblioteca que lida com objetos geométricos em Python.
3. Fiona: Biblioteca usada para ler e salvar dados geográficos nos formatos mais utilizados como shapefile, geopackage e GeoJSON.
4. Rasterio: biblioteca para trabalhar com dados matriciais (raster).

Para instalar essas bibliotecas eu recomendo que sempre faça uma busca online, isto porque muitos dessas bibliotecas de SIG possuem condições específicas para seu funcionamento adequado. Por exemplo para instalar o geopandas eu procuro por “conda geopandas” no google e a primeiro link é do proprio site do anaconda,



Seguindo as instruções temos:

```
conda install -c conda-forge geopandas
```

Veja que passamos o parâmetro **-c conda-forge**, isso quer dizer que o geopandas está em um repositório chamado conda-forge e não no repositório padrão do anaconda.

Após aceitar e prosseguir com o processo, serão instaladas diversas bibliotecas além do geopandas, podemos verificar com os comandos:

```
conda list
# ou
conda list <part do nome da biblioteca>
```

Veja que o GDAL, fiona e shapely estão todos ali.

Agora só precisamos de uma biblioteca para lidar com dados matriciais, fazendo o mesmo processo de busca online chegamos em:

```
conda install -c conda-forge rasterio
```

Visual Studio Code

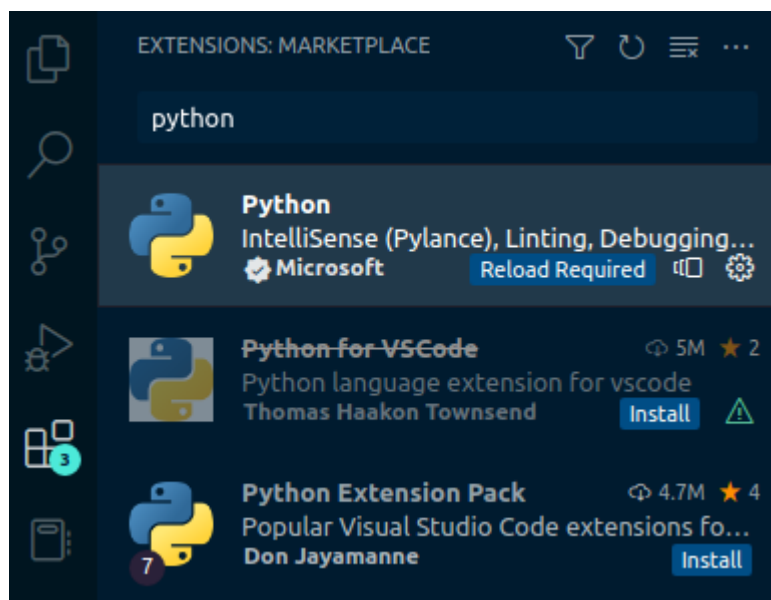
Precisamos agora de uma interface gráfica para criar os scripts e notebooks, as chamadas IDE's. O VSCode tem uma integração muito boa com o python, permitindo escolher diferentes ambientes do anaconda para executar os scripts e até mesmo trabalhar diretamente com Jupyter notebooks.

A instalação é bem simples, basta seguir a documentação:

<https://code.visualstudio.com/download>. Porém, para trabalharmos com o python precisamos de mais algumas configurações:

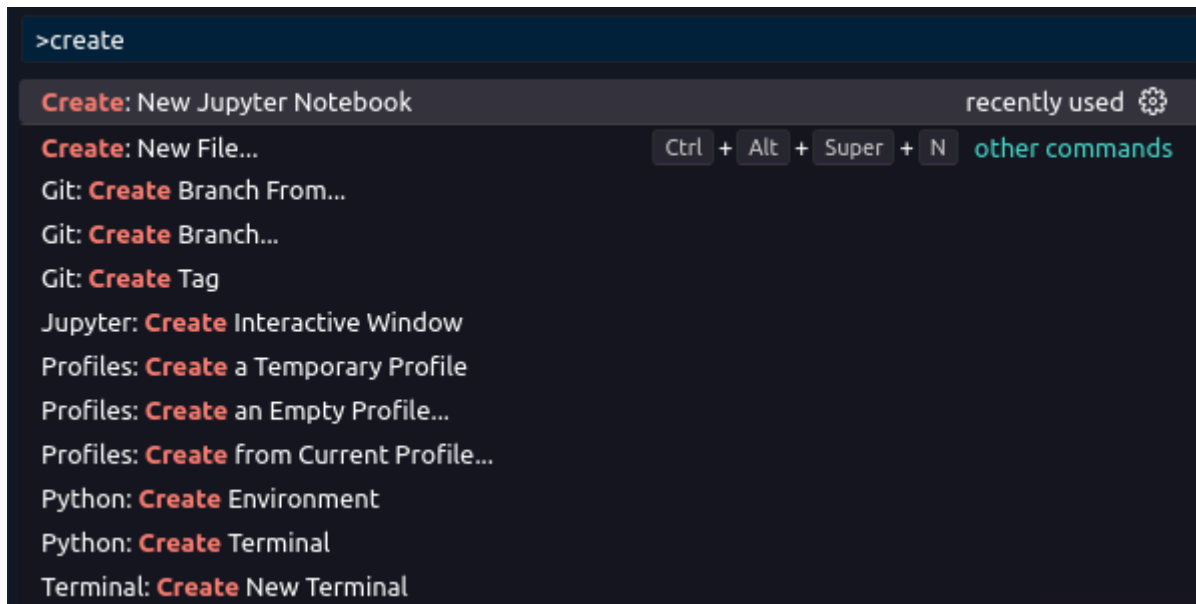
1. Instalar a extensão python

Dentro do VSCode abra a caixa de extensões no canto esquerdo e procure por “python”. Depois é só clicar em install e pronto.



2. Criar um jupyter notebook

Utilizando o comando Ctrl + Shift + P o VSCode exibirá uma janela de opções, procure por Create: New Jupyter Notebook



Se tentarmos executar algum comando receberemos um erro dizendo que é necessário instalar dois packages, o VSCode dá a opção de ele mesmo instalar.

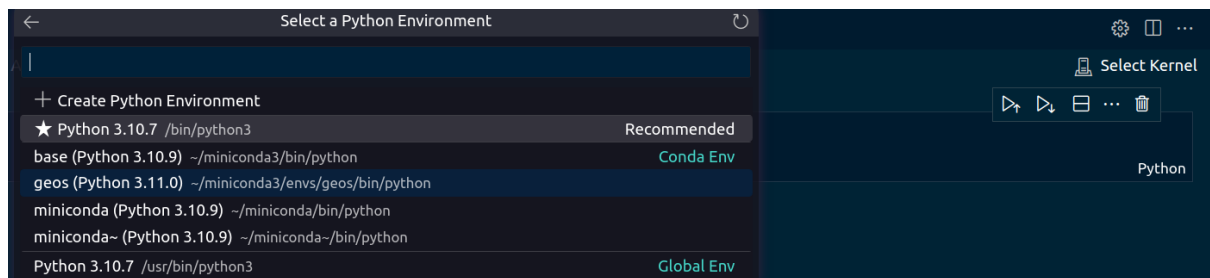
2. Instalar o Jupyter notebooks e ipykernel

Caso, de algum erro na instalação via VSCode, podemos voltar ao prompt e instalar com o conda

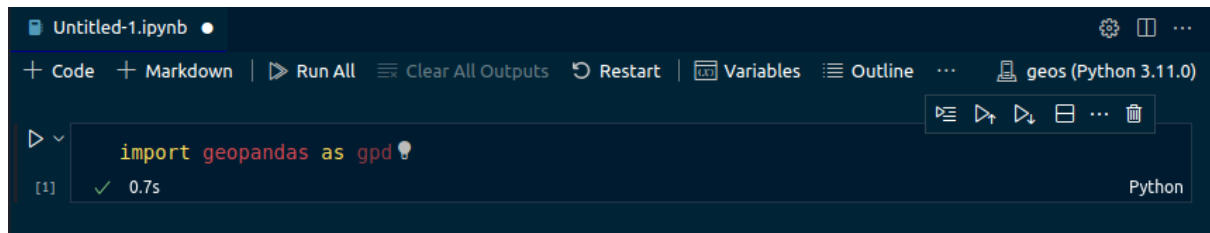
```
conda install -c anaconda ipykernel jupyter
```

3. Selecionar o kernel

Uma vantagem de trabalhar no VSCode ao invés de codar diretamente em um Jupyter notebook é que podemos escolher diferentes ambientes virtuais (kernel). Basta clicar em “kernel” no canto superior direito e selecionar o ambiente geos.



Para testar se está tudo funcionando corretamente execute o comando para importar uma das bibliotecas instaladas

A screenshot of a Jupyter Notebook interface. The title bar shows 'Untitled-1.ipynb'. The top toolbar includes buttons for '+ Code', '+ Markdown', 'Run All', 'Clear All Outputs', 'Restart', 'Variables', 'Outline', and a dropdown menu showing 'geos (Python 3.11.0)'. The code cell contains the line 'import geopandas as gpd' with a lightbulb icon. Below the code, the output shows '[1] ✓ 0.7s' and the word 'Python'.

Conclusão

Configurar um ambiente de desenvolvimento Python é uma etapa fundamental para realizar análises de dados geoespaciais. Seguindo os passos mostrados neste blog você poderá criar um ambiente virtual isolado, instalar as bibliotecas SIG necessárias para o seu projeto e configurar o VSCode como interface de desenvolvimento Python. Com esse setup você terá toda a liberdade para utilizar bibliotecas de software livre para realizar suas análises e criar suas próprias aplicações.

Happy coding!