

Instituto Nacional de Pesquisas Espaciais

Programa institucional de bolsas INPE/CNPq

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Previsão e monitoramento de eventos climáticos extremos sobre o Brasil usando Inteligência Artificial**

****

06 de Novembro de 2020

****

**3 Manipulação e visualização de dados climáticos**

Assim como na seção 2, será feita a manipulação e a visualização dos dados climáticos baixados porém dessa vez será utilizada a linguagem de programação python. Python é uma linguagem de programação de alto nível com uma sintaxe clara e muitos recursos poderosos como o acesso ao pacote do software NCL (NCAR Command Language) que foi feito pelo National Center for Atmospheric Research para pesquisa, que será utilizado juntamente ao python.

**Instalação do Python**

Para utilizar o Python no ubuntu foi necessário instalar o Miniconda. O Miniconda é um instalador mínimo, uma versão menor do Anaconda que apenas inclui o conda, Python e alguns pacotes necessários para a utilização e funcionamento correto.

Como instruído, foram utilizados os seguintes comandos no terminal para a instalação do Miniconda:

cd /tmp

curl <https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh>

Porém, após esta última linha de comando foi exibida a seguinte tela:



Com a seguinte mensagem na última linha:



Em seguida foi inserido o comando “bash Miniconda3-latest-Linux-x86\_64.sh” e a seguinte mensagem foi exibida:



Logo percebi que o arquivo não havia sido corretamente baixado na pasta tmp do ubuntu e assim impossibilitando a instalação.

Então, realizei o download manualmente do “Miniconda3-latest-Linux-x86\_64.sh” utilizando o link <https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh> o qual foi colocado em um diretório especifico.

Depois o arquivo baixado foi movido para a pasta tmp utilizando o comando “mv Miniconda3-latest-Linux-x86\_64.sh /tmp”

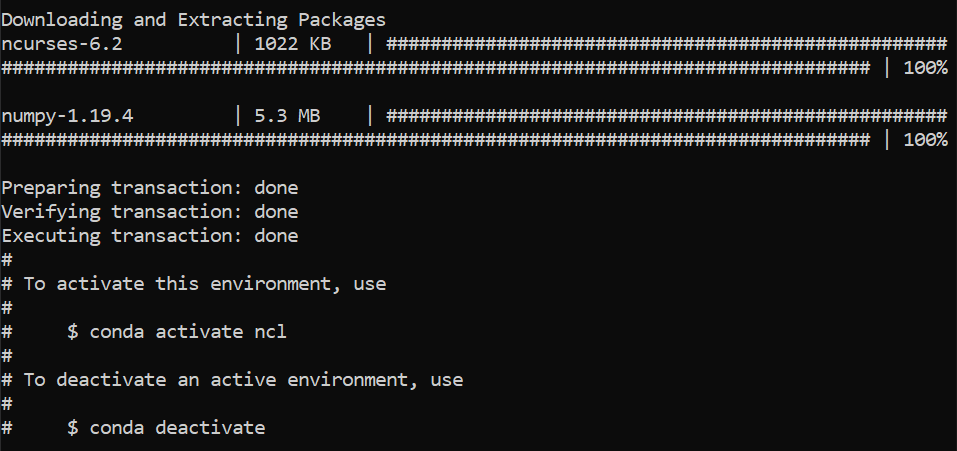


Após isso, nesse diretório, foi utilizado o comando “bash Miniconda3-latest-Linux-x86\_64.sh” no ubuntu para instalar o miniconda.





Depois foi utilizado o comando “conda create -n ncl -c conda-forge xarray netcdf4 scipy pyngl pynio ncl” para criar um ambiente de trabalho e carregar os pacotes e após algum tempo de espera e vários downloads e instalações foi exibida a seguinte mensagem:

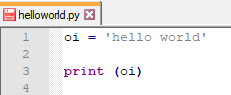


Com isso foi concluída a instalação do miniconda3 e criação de ambiente de trabalho ncl.

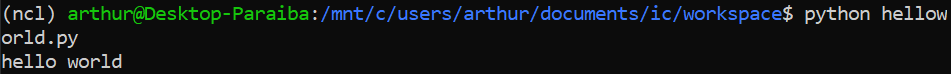
**Utilizando o Python**

Para começar a utilização do python, inicialmente foram baixados dois scripts disponibilizados pelo NCAR ([aqui](http://ncl.ucar.edu/Applications/NCL_to_Python/)) que possuem algumas funções aplicadas ao python e ao Ncl com a finalidade de comparar as duas linguagens e mostrar como ler arquivos netCDF. Com os dois arquivos prontos para utilização, foi ativado o ambiente de trabalho Ncl e os scripts seriam colocados em execução.

Porém, para me certificar que os scripts rodariam de maneira correta, foi criado um arquivo “helloworld.py” com uma unica função de imprimir a mensagem “hello world” quando executado.



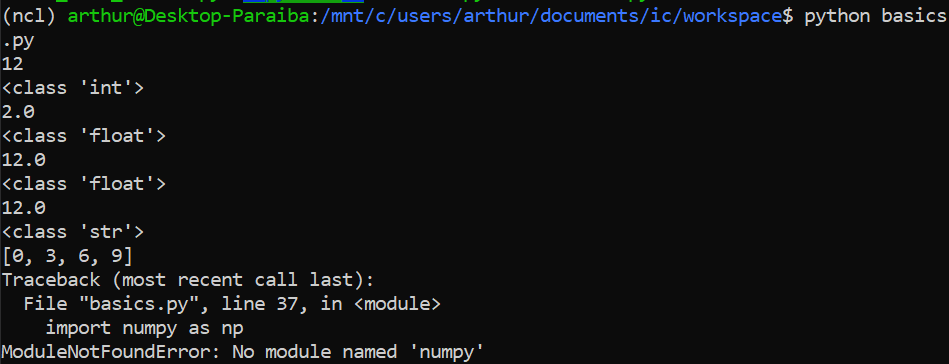
Ao rodar esse arquivo com o comando “python helloworld.py’ a seguinte mensagem foi exibida no terminal:



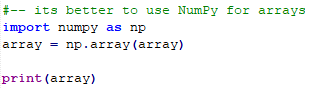
Logo, tudo ocorreu corretamente.

**basics.py:**

O mesmo processo utilizado para rodar o helloworld.py foi utilizado para rodar o basics.py e o resultado foi o seguinte:



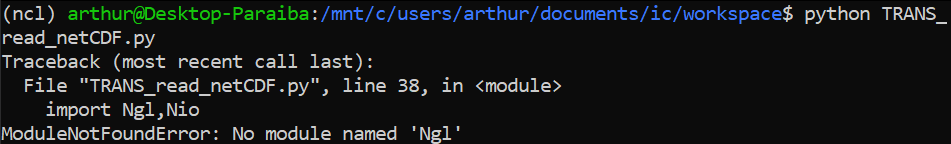
Comparando com os dados escritos no script, tudo ocorreu corretamente até a seguinte função:



onde o python não reconheceu ‘numpy’ como um módulo e parou o script.

**TRANS\_read\_netCDF.py:**

Assim como feito para o basics.py, também foi realizado o mesmo processo para a execução do TRANS\_read\_netCDF.py e o resultado foi o seguinte:

****

Nesse caso na linha 38 havia uma função:



O script não conseguiu dar continuidade após essa linha pois não encontrou um módulo chamado de “Ngl”.