

INSTALAÇÃO DE SOFTWARES CIENTÍFICOS

João Marcello Pereira

jmarcellopereira@ufpi.edu.br jmarcellopereira@gmail.com

PACOTES INICIAIS

```
$sudo apt-get update && sudo apt-get upgrade
$sudo apt-get install build-essential
```

OBS: É IMPORTANTE INSTALAR PRIMEIRO O ANACONDA ANTES DE INSTALAR OS OUTROS INSTALAÇÃO DO ANACONDA VERSÃO 2.X OU 3.X

download do conda para a pasta download

<https://www.continuum.io/downloads>

Abra um terminal e digitar:

```
$sudo chmod +x Anaconda-X.X.X-Linux-x86_64.sh
$./Anaconda-X.X.X-Linux-x86_64.sh
```

feche o terminal, abra novamente e digite:

```
$conda update conda
```

para remover, no diretório no qual se encontra o anaconda digite:

```
$rm -rf ~/anaconda
```

NÃO É NECESSARIO ROOT! Verifique se o pip está instalado:

```
$pip -help
```

Se não tiver o pip instalado digite:

```
$sudo apt-get install python-pip
```

Alternativa ao anaconda

```
$sudo apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook pyt.
```

INSTALAR O JUPYTER

Verifique se o python3 ou superior está instalado

```
$python3
```

Se não tiver o python3 ou superior instalado digite:

```
$sudo apt-get install python3
```

Verifique se o ipython está instalado

```
$ipython
```

Se não tiver o ipython instalado digite:

```
$sudo apt-get install ipython
```

instalar o jupyter:

```
$conda install jupyter
```

INSTALAÇÃO DO JULIA

Via synaptic

digite julia e veja se mostra a versão 0.4.3 ou superior. Se não, instale via PPA

Via terminal

```
$sudo add-apt-repository ppa:staticfloat/juliareleases  
$sudo add-apt-repository ppa:staticfloat/julia-deps  
$sudo apt-get update  
$sudo apt-get install julia
```

Adicionando pacotes via terminal

```
$julia  
$julia> Pkg.add("IJulia")
```

Se ocorrer:

Failed to execute "git ls-remote --tags --heads git://github.com/twbs/bootstrap-sass.git", exit

Isso acontece em funcao do firewall. Use:

```
$ git config --global url."https://".insteadOf git://
```

cuidado que é Of ("o" mesmo) maiusculo.

se ocorrer o erro:

WARNING: ZMQ had build errors.

- packages with build errors remain installed in /home/jmp/.julia/v0.4
- build the package(s) and all dependencies with `Pkg.build("ZMQ")`
- build a single package by running its `deps/build.jl` script

significa que ha dependencia de algum recurso de compilação. Instale o CodeLite que resolve. Ou

```
$sudo apt-get update && apt-get upgrade
$sudo apt-get install build-essential
```

Pacotes importantes

```
Julia>Pkg.add("Gadfly")      :Graficos
Julia>Pkg.add("PyPlot")     :Facilitar graficos
Julia>Pkg.add("Roots")      : raizes e outras funções
Julia>Pkg.add("Winston")    : graficos
Julia>Pkg.add("ODE")        : equações diferenciais
Julia>Pkg.add("Sundials")
Julia>Pkg.add("SymPy")      : para trabalhar com matematica simbolica no julia
```

Instalar pacote nao oficial

```
Julia>Pkg.clone("git://github.com/hwborchers/NumericalMath.jl.git")
Julia>Pkg.init("NumericalMath")
```

INSTALAR O OCTAVE 4

Via synaptic

Digite octave e veja se mostra a versão 4 ou superior ### Via console:

```
$ sudo apt-add-repository ppa:octave/stable
$ sudo apt-get update
$ sudo apt-get install octave
```

ou

```
$sudo apt-get install g++ gcc fort77
$sudo apt-get build-dep octave
$wget ftp://ftp.gnu.org/gnu/octave/octave-4.x.x.tar.xz
$tar Jxvf octave-4.x.x.tar.xz
$cd octave-4.x.x
$./configure
$make
$sudo make install
```

Pacotes importantes:

via synaptic (VERSÕES IGUAIS OU SUPERIORES):

control | 2.8.3 symbolic | 2.2.2 linear-algebra | 2.2.2 optiminterp | 0.3.4 dataframe
| 1.1.0 zenity | 0.5.7 miscellaneous | 1.2.1 specfun | 1.1.0 general | 2.0.0 statistics
| 1.2.3 odepkg | 0.8.5 splines | 1.2.8 io | 2.2.4 struct | 1.0.11 optim | 1.4.1 strings
| 1.2.0 strings | 1.2.0

via terminal para instalar o symbolic instalar o liboctave-dev e sympy download
o pacote symbolic no <http://octave.sourceforge.net/symbolic/index.html> e usar
o comando dentro do octave

```
octave: pkg install symbolic-xxx.tar.gz
```

usar um pacote (dentro do código)

```
pkg load pacote symbolic
```

para listar todos os pacotes instalados e ou carregados

```
pkg list
```

para a interface grafica do octave nao abrir quando for digitado octave no kon-
sole, vá em /usr/bin/ e mude o nome do binario octave para octave-gui e du-
plique o arquivo octave-cli para octave, depois tem que alterar no menu do kde.
Instalação do kernel do octave para o jupyter

```
$ pip install octave_kernel
$ python -m octave_kernel.install
```

Agora é só abrir o jupyter que vai aparecer no botão new o notebook do Octave

INSTALANDO O SCILAB

Via synaptic

Digite scilab e veja se mostra a versão 5.5 ou superior ### Via terminal \$sudo apt-get install scilab

ou baixe o arquivo fonte e:

```
$ apt-get build-dep scilab
$ ./configure
```

```
# Note that the target 'all' is not building the localization except if you enable the option "--
$ make all
$ make install
```

Instalação do kernel do scilab para o jupyter

antes, instalar o scilab2py

```
$ pip install scilab2py
```

depois

```
$ pip install scilab_kernel
$ python -m scilab_kernel.install
```

Agora é só abrir o jupyter que vai aparecer no botão new a opção para criar um notebook Scilab

INSTALANDO O SAGE

SAGE NÃO FUNCIONA COM O PYTHON 3.X . PORTANTO, INSTALE O ANACONDA CORRESPONDENTE AO PYTHON 2.X POIS O SCIPY E O NUMPY SO FUNCIONAM COM PYTHON 2.X

No Ubuntu, digite no terminal:

```
$sudo apt-add-repository -y ppa:aims/sagemath
$ apt-get update
$sudo apt-get install sagemath-upstream-binary
```

Para outras distribuições linux, siga as instruções do site (link). Obs: importante atualizar a biblioteca libgmp. Faça isso via synaptic Para iniciar, basta digitar no seu terminal:

```
$ ./sage -notebook
```

Instalar a partir de do arquivo .tar.gz (melhor opção):

baixar o arquivo sage-7.x-distro_xx.xx-x86_64.tar.bz2 do site <http://linorg.usp.br/sage/linux/64bit/index.htm> no diretório /home/usuario. Depois extrair e atribuir controle total:

```
$ sudo chmod 777 -R SageMath
```

Acesse a pasta e execute:

```
$/sage - notebook
```

ou

```
$/sage  
sage: notebook
```

Se ocorrer erro, use

```
$ sudo chmod 777 -R SageMath
```

para atribuir controle total sobre todos os arquivos e pastas do diretório SageMath e execute novamente (\$./sage). Se ocorrer novamente erro, o sage deverá executado com sudo, use:**

```
$ sudo ./sage
```

uma pasta .sage será criada no diretório /home, apague-a e execute novamente :

```
$/sage  
sage:notebook()
```

para rodar como servidor para outros poderem acessar via rede (ATENÇÃO PARA OS DETALHES: SÃO DUAS ASPAS SIMPLES SEPARADAS E O NOME INTERFACE NO SINGULAR):

```
$sage  
sage: notebook(interface=' ')
```

Para acessar o Sage de um celular, tablet ou outro computador, basta conectar à mesma rede do servidor (Com ou sem fio) e usar no navegador:

http://ip_servidor:8080

Para usar o sage no jupyter

```
sage: !sage -n jupyter
```

ou no terminal

```
$ sage -n jupyter "se o sage foi instalado via apt-get ou synaptics"
```

```
$ ./sage -n jupyter "se o sage foi extraído para uma pasta"
```

Instalar pacote

Se estiver rodando a partir de uma pasta: Entre na pasta do sage e execute:

```
$/sage -sh  
SageMath$ pip install nome_pacote
```

Se estiver rodando a partir de uma instalação pelo synaptics ou apper ou... de qualquer lugar no terminal execute:

```
$ sage -sh  
SageMath$ pip install nome_pacote
```

Atualizar o Sage local

```
$/sage -sh  
SageMath$ sage --upgrade
```

INSTALANDO O MAXIMA E WXMAXIMA

Via synaptic

Digite maxima e veja se mostra a versão 5.3 ou superior e 15.04 para o wxmaxima
Via terminal

```
$sudo apt-get install maxima  
$sudo apt-get install wxmaxima
```

Instalação do kernel do maxima para o jupyter