

Hay varias cosas para hacer. Están itemizadas, con una explicación después de cada tarea:

- Instalar git.

```
#Para tener las versiones de paquetes más recientes
sudo apt-get update
#GIT
sudo apt-get install git
```

Dicho mal y pronto, git es un programa usado para trabajar en equipo sobre un proyecto (como overleaf/sharelatex, dropbox, etc), pero con ciertas diferencias. La más importante es que si uno hace cambios en los archivos, éstos **no se actualizan instantáneamente** en la computadora de la otra persona.

Además, podemos trabajar sobre el mismo archivo (en lugares distintos), y luego combinar los cambios de manera muy fácil.

- Crease una cuenta en <https://github.com/>

El proyecto que hice para el final está alojado (hosteado) en un sitio web - el que más se usa - llamado **github**. Es gratis. Tendrías que hacerte una cuenta ahí así te puedo agregar al proyecto.

(¿Estás de acuerdo con que sea público?)

- Compartirme tu usuario de github por email, así te agrego al proyecto y eso te habilita a subir cambios. Esto es independiente del resto de los pasos del instructivo.

- Elegir carpeta donde trabajar.

Esta carpeta va a ser la que tendrá tus proyectos de github. Yo uso **/home/billy/Projects/git/github**, pero no hace falta que sea ni remotamente parecida.

- Bajarse el proyecto de github.

El comando de abajo de baja el proyecto a una carpeta llamada “MESSI”, así que correlo parada en la carpeta que va a tener todos tus proyectos de github:

```
git clone https://github.com/billy-mosse/MESSI.git
```

Deberías ver una carpeta “MESSI” con subcarpetas “code, garbage, instructions”, etc.

- Instalar la última versión de **python3**:

```
#Creo que viene por default ya, igual.
sudo apt-get install python3
#Si ya lo tenías instalado, actualizalo
sudo apt-get upgrade python3
```

Podés chequear que está instalado con:

```
python3 --version
#Output: Python 3.5.2
```

- Instalar **pip**, para bajarse librerías de python, y otros yuyos:

```
sudo apt-get -y install python3-pip
pip3 install --upgrade pip

#Yo tuve un problema y tuve que hacer esto
pip3 install --upgrade setuptools pip
```

- (Opcional, pero recomendado) Instalar **virtualenv**:

```
#Instalar virtualenv por medio de pip.
python3 -m pip install --user virtualenv
```

Virtualenv sirve para cuando tenés más de un proyecto en python. Tal vez para uno necesitás la versión 3.7 de la librería matplotlib, pero otro es medio antiguo y sólo anda con la versión 2.2. Virtualenv hace que puedas no instalar paquetes **globalmente**.

La idea es que cuando trabajemos en este proyecto, “prendamos” el entorno virtual de python correspondiente, que va a tener instaladas las versiones específicas de las librerías necesarias. (Es un comando muy simple.)

No hace realmente falta que lo instales pero es muy útil para no tener problemas de versiones en el futuro.

- (Opcional, cont) Crear el virtualenv correspondiente al proyecto, en la misma carpeta del proyecto, o en una carpeta virtualenvs/ en tu home:

```
#Crea un entorno virtual llamado env_messi
#y deja los archivos de configuracion
#en una carpeta env_messi
#en el directorio donde estás parada
python3 -m virtualenv env_messi
```

¡Avisame si en vez de env_messi usás otro nombre para el virtualenv, porque en ese caso tengo que hacer algo!

- Prender el entorno virtual:

```
source env_messi/bin/activate
```

- Instalar las librerías que necesita el proyecto:

```
#El doble AND es para correr comandos en serie (uno detras de otro)
#siempre y cuando no devuelvan error
pip3 install numpy &&\
pip3 install matplotlib &&\
pip3 install pandas &&\
pip3 install networkx
```

Si esto no te anda, cambiá pip3 por pip. Luego, chequeá que las librerías estén instaladas corriendo:

```
#Te abre una consola de python  
python3
```

```
#Dentro de la consola  
import numpy
```

- (Totalmente opcional) Intentar instalar sage para python:

Sage funciona bien con python 2.6, y está “en fase experimental” para python 3. Si querés podés intentar instalarlo para python 3. Yo usé python 2.6 para el final de Alicia pero quiero pasar todo a python 3, y me parece que sage es totalmente reemplazable por la librería `scipy.linalg`.

Bajarse sage de <http://linorg.usp.br/sage/linux/index.html>

Descomprimirlo, y correr:

```
make configure &&\  
./configure --with-python=3 &&\  
make build
```

y rezar. Si te tira error “make: execvp: ./bootstrap: Permission denied” podés intentar arreglarlo con:

```
chmod +x bootstrap
```

pero no gastes demasiado tiempo.