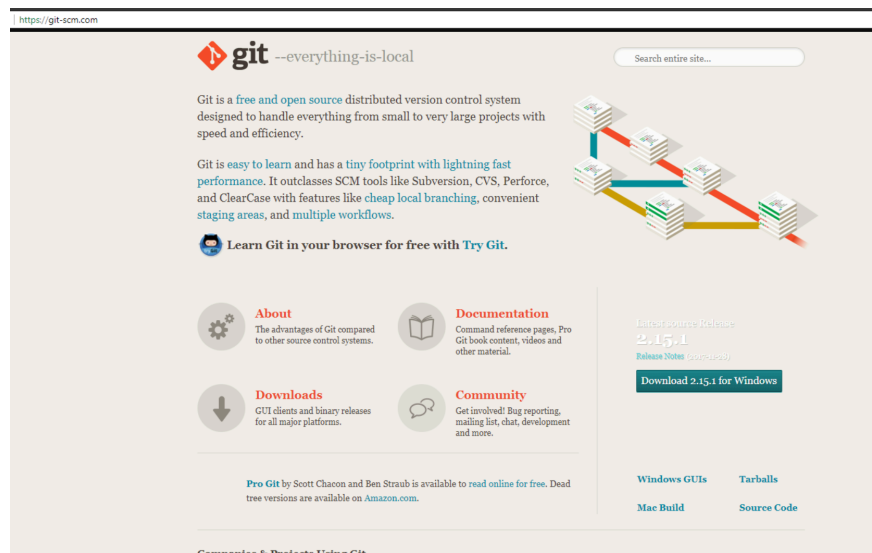


Instalación y configuración de Git

El primer paso es descargar GIT en nuestra computadora, para esto vamos al sitio oficial de Git: <https://git-scm.com/>



Una vez descargado instalamos Git como cualquier aplicación de nuestra computadora, al finalizar la instalación abrimos la línea de comandos de Git para personalizar el entorno, este paso se realiza solo una vez, Git mantendrá estas configuraciones y te permite cambiarlas en cualquier momento.

Vamos a configurar la identidad en Git, para esto hay establecer tu nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información. Para configurar la identidad utilizamos la herramienta **git config** de la siguiente manera:

```
$ git config --global user.name "Erick Diaz"
$ git config --global user.email erickdiaz@ufm.edu"
```

```
MINGW64:/c/Users/Erick

Erick@Erick-PC MINGW64 ~
$ git config --global user.name "Erick Diaz"

Erick@Erick-PC MINGW64 ~
$ git config --global user.email "erickdiaz@ufm.edu"

Erick@Erick-PC MINGW64 ~
$
```

Luego de ingresar estos comandos podemos comprobar la configuración, para esto podemos utilizar el comando **git config --list** para listar todas las propiedades de git

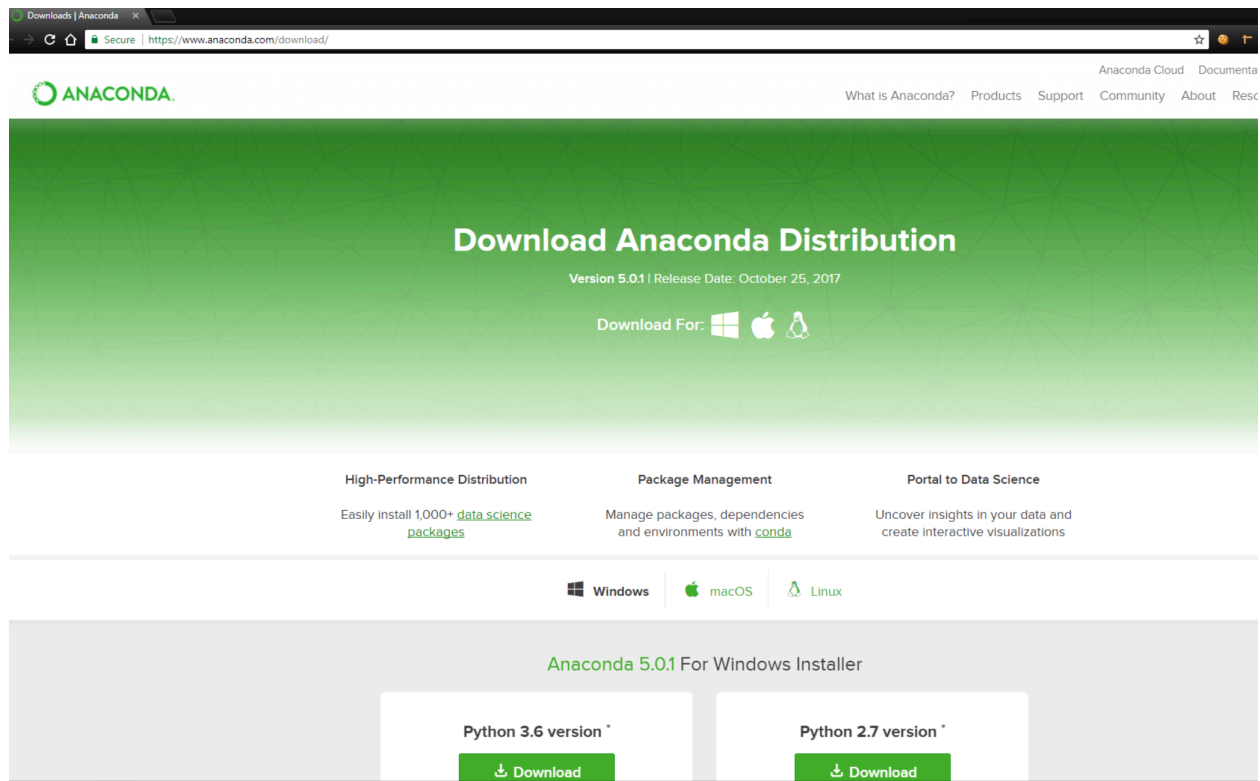
```
$ git config --list
```

```
Erick@Erick-PC MINGW64 ~
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
credential.helper=manager
user.name=Erick Diaz
user.email=erickdiaz@ufm.edu
```

Instalación de Anaconda

La instalación de Anaconda es sencilla debemos descargar el programa e instalarlo como cualquier otro, el proceso es sencillo y no necesita configuración extra.]

<https://www.anaconda.com/download/>



Creación de ambientes en Anaconda

Para crear un ambiente en Anaconda tenemos 2 opciones, desde la línea de comandos o desde la aplicación **Anaconda Navigator**, como ejemplo crearemos desde la línea de comandos un ambiente llamado **ml_ufm** con la versión de Python 3.6 y los paquetes numpy y también instalaremos Jupyter.

1) Para crear un ambiente, el comando es el siguiente:

```
conda create --name myenv
```

NOTA: Reemplazar **myenv** con el nombre que le quiero dar al ambiente

2) Cuando conda te pregunta si quieres preguntar ingresas y:

```
proceed ([y]/n)?
```

Esto crea el ambiente myenv en /envs/.

Para Crear un ambiente con una versión específica de Python : T

```
conda create -n myenv python=3.4
```

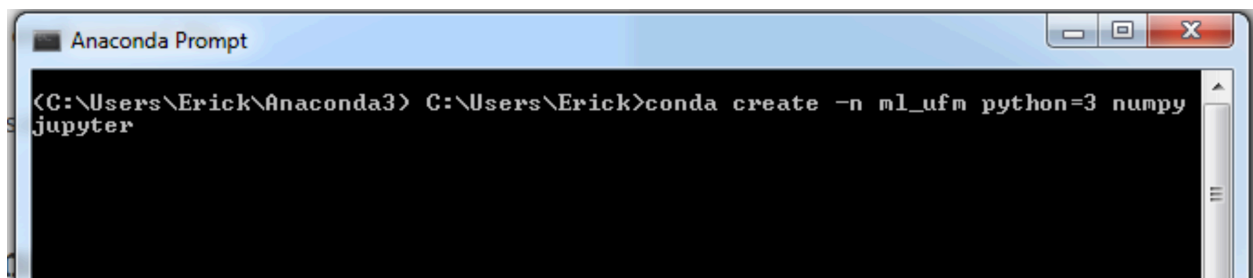
Para crear un ambiente con un paquete específico:

```
conda create -n myenv scipy
```

Ejemplos:

Creamos el ambiente ml_uvm con Python 3.6 e instalamos numpy y Jupyter:

```
conda create -n ml_uvm python=3 numpy jupyter
```



```
conda create -n ml_ufm python=3 numpy jupyter

ipykernel: 4.7.0-py36h2f9c1c0_0
ipython: 6.2.1-py36h9cf0123_1
ipython_genutils: 0.2.0-py36h3c5d0ee_0
ipywidgets: 7.1.0-py36_0
jedi: 0.11.1-py36_0
jinja2: 2.10-py36h292fed1_0
jpeg: 9b-hb83a4c4_2
jsonschema: 2.6.0-py36h7636477_0
jupyter: 1.0.0-py36h422fd7e_2
jupyter_client: 5.2.1-py36_0
jupyter_console: 5.2.0-py36h6d89b47_1
jupyter_core: 4.4.0-py36h56e9d50_0
libpng: 1.6.32-h140d38e_4
markupsafe: 1.0-py36h0e26971_1
mistune: 0.8.3-py36_0
mkl: 2018.0.1-h2108138_4
nbconvert: 5.3.1-py36h8dc0fde_0
nbformat: 4.4.0-py36h3a5bc1b_0
notebook: 5.2.2-py36hc48260a_0
numpy: 1.14.0-py36h4a99626_0
openssl: 1.0.2n-h74b6da3_0
pandoc: 1.19.2.1-hb2460c7_1
pandocfilters: 1.4.2-py36h3ef6317_1
parso: 0.1.1-py36hae3edee_0
pickleshare: 0.7.4-py36h9de030f_0
pip: 9.0.1-py36h226ae91_4
prompt_toolkit: 1.0.15-py36h60b8f86_0
pygments: 2.2.0-py36hb010967_0
pyqt: 5.6.0-py36hb5ed885_5
python: 3.6.3-h3b118a2_4
python-dateutil: 2.6.1-py36h509ddcb_1
pyzmq: 16.0.3-py36he714bf5_0
qt: 5.6.2-vc14h6f8c307_12
qtconsole: 4.3.1-py36h99a29a9_0
setuptools: 38.4.0-py36_0
simplegeneric: 0.8.1-py36heab741f_0
sip: 4.10.1-py36h9c25514_2
six: 1.11.0-py36h4db2310_1
sqlite: 3.20.1-h9eeafa9_2
testpath: 0.3.1-py36h2698cfe_0
tornado: 4.5.3-py36_0
traitlets: 4.3.2-py36h096827d_0
vc: 14-h2379b0c_2
vs2015_runtime: 14.0.25123-hd4c4e62_2
wcwidth: 0.1.7-py36h3d5aa90_0
webencodings: 0.5.1-py36h67c50ae_1
wheel: 0.30.0-py36h6c3ec14_1
widgetsnbextension: 3.1.0-py36_0
wincertstore: 0.2-py36h7fe50ca_0
zlib: 1.2.11-h8395fce_2

Proceed ([y]/n)? y_
```

Una vez instalado el ambiente, si lo queremos utilizar debemos activarlo para esto ingresamos el siguiente comando:

```
activate ml_ufm
```

```
#
# To activate this environment, use:
# > activate ml_ufm
#
# To deactivate an active environment, use:
# > deactivate
#
# * for power-users using bash, you must source
#

C:\Users\Erick\Anaconda3> C:\Users\Erick>activate ml_ufm
```

Sabemos que el ambiente esta activo porque aparece su nombre entre paréntesis al inicio de la línea. Con el ambiente activo ya podemos ejecutar Jupyter y crear nuestro primer notebook:

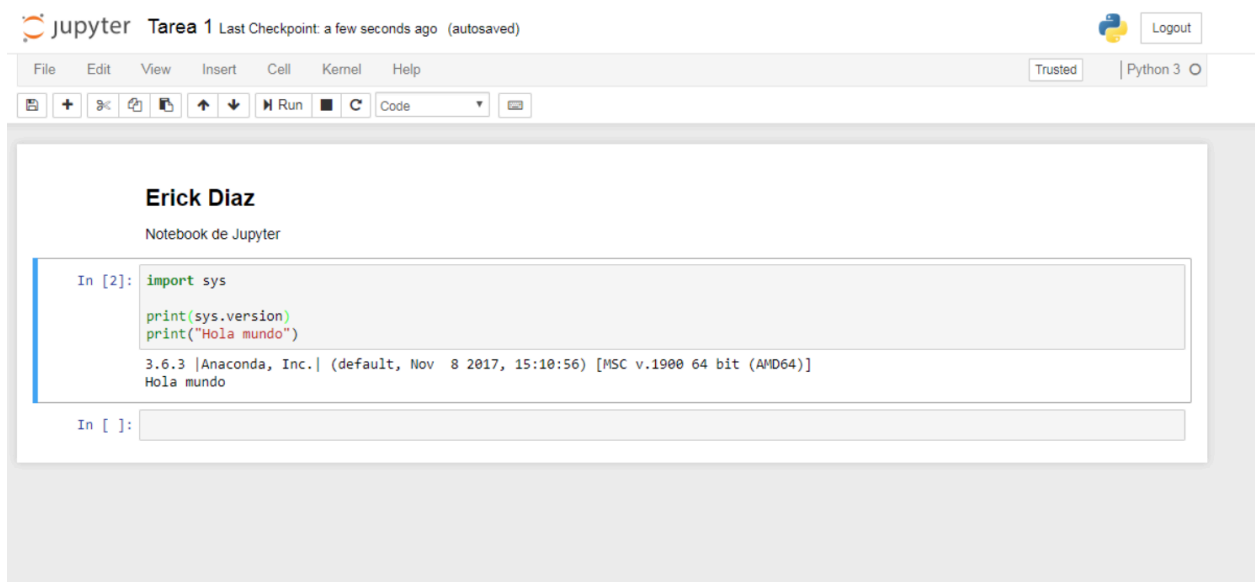
```
jupyter notebook
```

```
<ml_ufm> C:\Users\Erick\Documents>jupyter notebook
```

Al presionar la tecla enter lanzara Jupyter en el navegador web, una vez en Jupyter si queremos crear un nuevo notebook que utilice Python presionamos en botón new y seleccionamos Python 3, que es la versión de Python que instalamos en el ambiente **ml_ufm** :

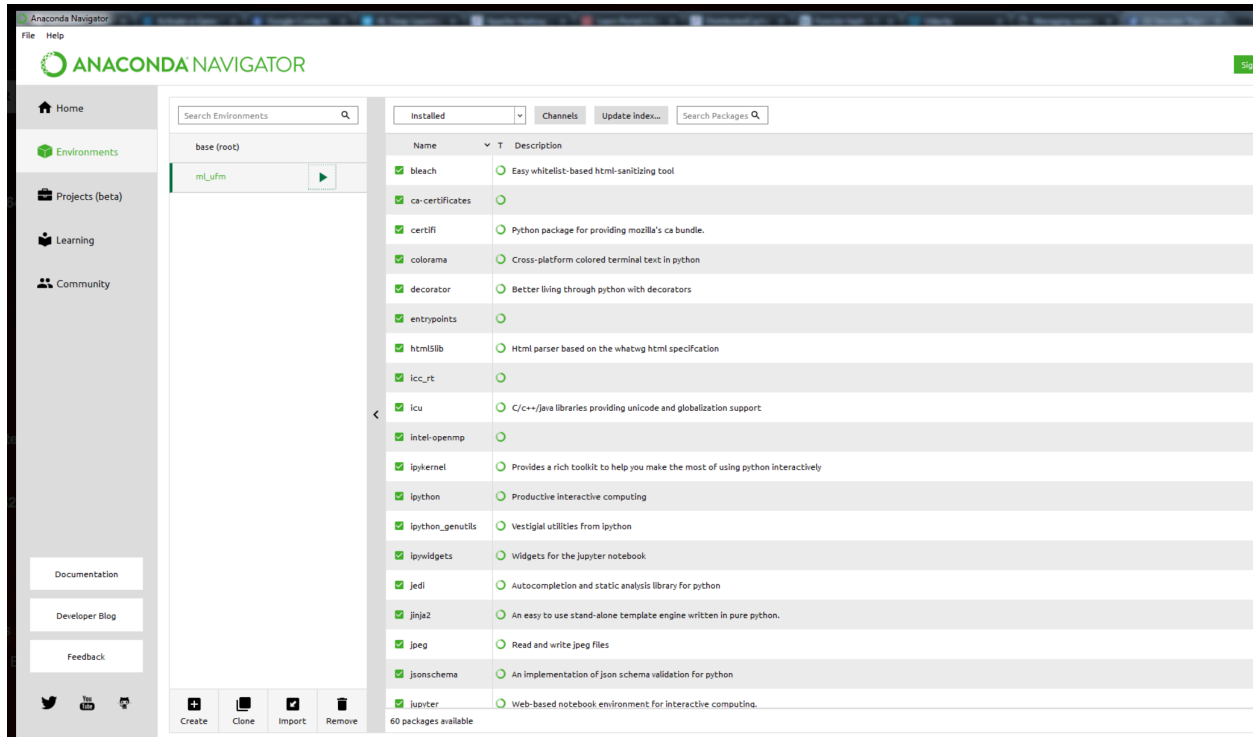


De esta forma de debe de ver la tarea:



Si no queremos utilizar la línea de comandos podemos crear el ambiente y ejecutar Jupyter desde la aplicación Anaconda Navigator

En la pestaña de ambientes nos lista los ambientes que tenemos disponibles y lista que paquetes tiene cada ambiente, en la parte inferior de la ventana este el botón para crear un nuevo ambiente.



En la pestaña “Home” podemos seleccionar el ambiente y ejecutar Jupyter

