



# conda introduction

virtual environments manager



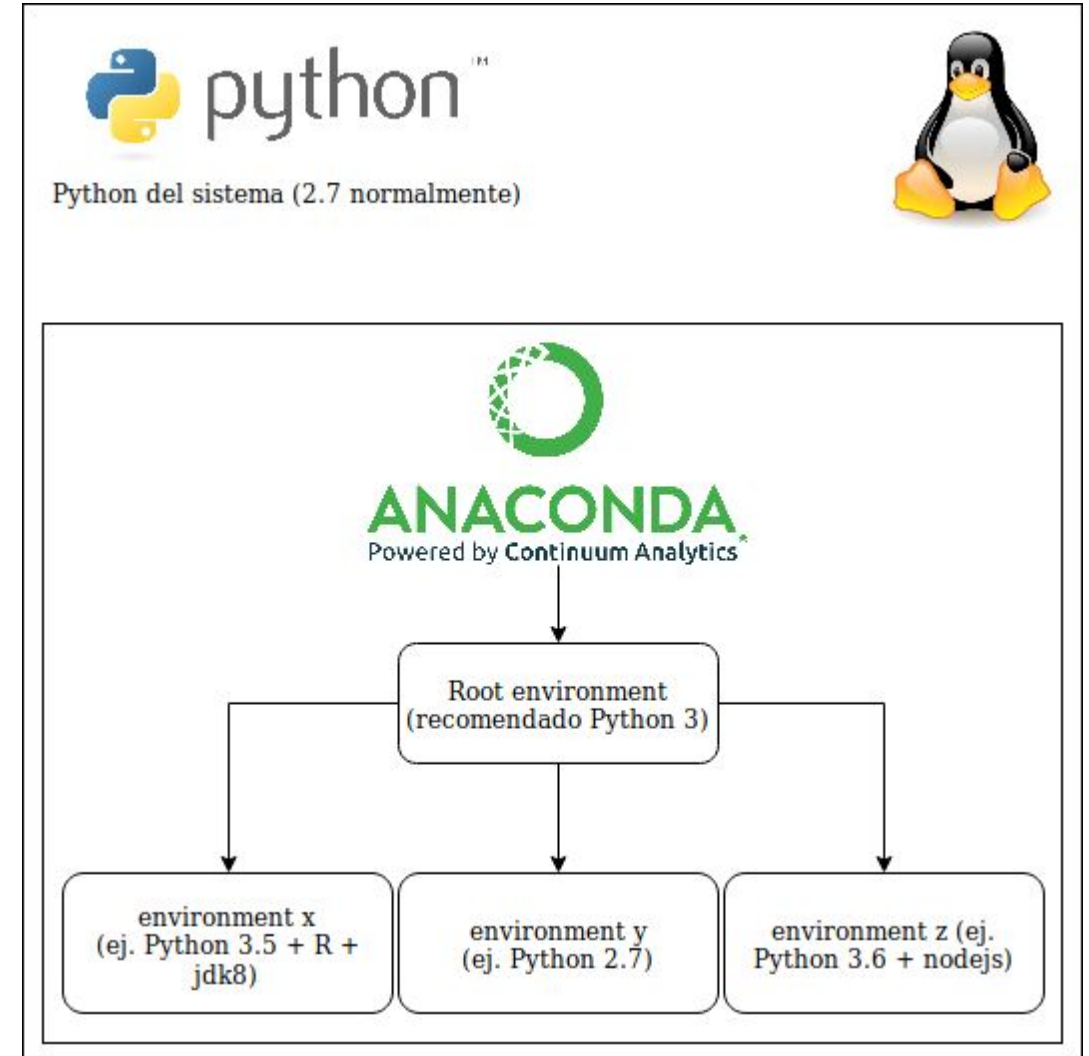
1. Entornos virtuales: concepto
2. Anaconda: ventajas
3. Anaconda flavors: Python 2 vs Python 3
4. Anaconda flavors: Anaconda vs Miniconda
5. conda: quickstart



# 1. Entornos virtuales: concepto

## Concepto:

- Distribución de Python de Anaconda Inc.
- Se instala independientemente al Python del sistema
- Incluye el conocido gestor de entornos virtuales y paquetes: conda
- Anaconda Python Distribution  $\neq$  conda

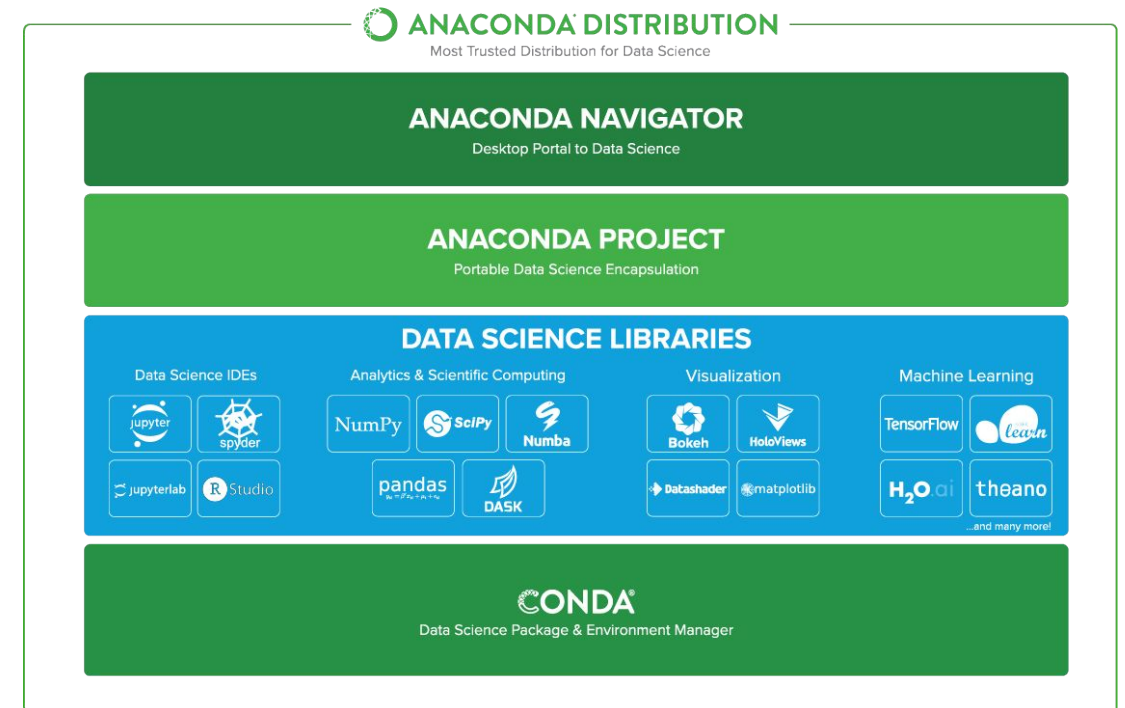




## 2. Anaconda: ventajas

### Ventajas del uso de Anaconda:

- Creación de entornos virtuales e independientes del Python del sistema y del resto de entornos
- Mayor seguridad con respecto a otros gestores de paquetes (pip)
- Resolución automática de entornos
- Mejor rendimiento de los paquetes instalados con respecto a otras alternativas (no siempre)
- Más simplicidad a la hora de instalar librerías complejas (Tensorflow, Jupyter, etc.)
- Existen otras alternativas como virtualenv

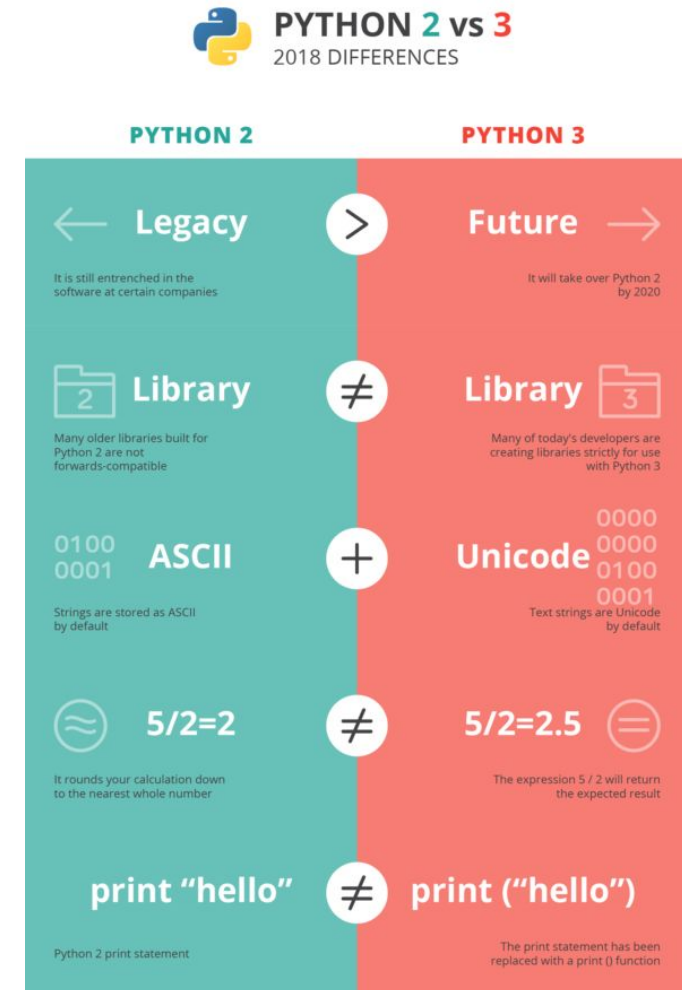




### 3. Anaconda flavors: Python 2 vs Python 3

#### Python 2 vs Python 3:

- Siempre Python 3 (especialmente ahora)
- No se recomienda iniciar nuevos desarrollos en Python 2. Este pierde cualquier tipo de soporte en 2020
- Es sencillo actualizar la mayor parte de códigos 'legacy' a Python 3
- Anaconda ofrece instaladores de conda tanto en flavor Python 3 como Python 2

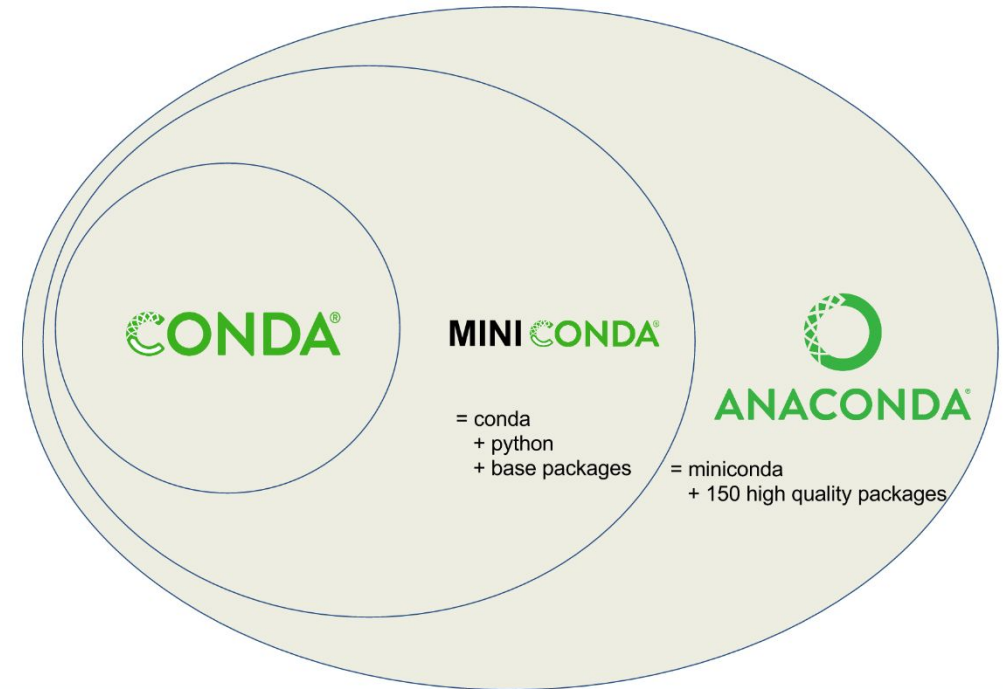




## 4. Anaconda flavors: Anaconda vs Miniconda

### Anaconda vs Miniconda:

- Anaconda incluye un gran número de librerías
- Anaconda ocupa aprox. 500Mb
- Ambas versiones incluyen conda
- Ambas versiones tienen acceso a todas las librerías posibles via conda (y pip)
- La opción recomendada en infraestructura cloud y equipos con almacenamiento limitado es Miniconda





### Instalación:

- Versión Linux (instalador = Bash script)
- Versión Python 3 (todo el código Python del proyecto está en Python 3)
- Aceptar términos y condiciones y añadir al Path cuando lo pregunte



```
~$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
~$ bash Miniconda3-latest-Linux-x86_64.sh
```



## 5. conda: quickstart

### Información de la instalación:

- Muy útil para corregir instalaciones corruptas
- Especial atención a la versión de conda, Python, y rutas de creación de los entornos virtuales

```
pi@raspberrypi:~ $ conda info

active environment : None
  shell level      : 0
  user config file  : /home/pi/.condarc
populated config files : /home/pi/berryconda3/.condarc
  conda version     : 4.5.11
conda-build version : not installed
  python version    : 3.6.1.final.0
base environment    : /home/pi/berryconda3 (writable)
  channel URLs      : https://conda.anaconda.org/rpi/linux-armv7l
                    https://conda.anaconda.org/rpi/noarch
  package cache     : /home/pi/berryconda3/pkg
                    /home/pi/.conda/pkg
  envs directories  : /home/pi/berryconda3/envs
                    /home/pi/.conda/envs
  platform          : linux-armv7l
  user-agent        : conda/4.5.11 requests/2.18.1 CPython/3.6.1 Linux/4.14.94-v7+ raspbian/9
glibc/2.24
  UID:GID           : 1000:1000
  netrc file        : None
  offline mode      : False
```





### Creación de nuevo entorno:

- El nombre del entorno debe ser único, ya que se utiliza activamente en la gestión del mismo (activar, desactivar, etc.)

```
~$ conda create -n <nombre del entorno>
```



### Obtención de entornos instalados:

- Útil cuando se trabaja con múltiples entornos y no se recuerda exactamente el nombre de aquel con el que se desea interactuar
- Se indica además el entorno actualmente activo y la localización de cada uno de los entornos

```
(analytics) pi@raspberrypi:~ $ conda info --envs
# conda environments:
#
base                /home/pi/berryconda3
analytics            *  /home/pi/berryconda3/envs/analytics
py                   /home/pi/berryconda3/envs/py
```



### Activación de entornos:

- La activación de los entornos virtuales se realiza desde la aplicación CLI de conda
- Hay que ser extremadamente cuidadoso al activar y desactivar los entornos, ya que pueden irse acumulando (activar un entorno dentro de otro)



```
~$ conda (o source) activate <nombre del entorno>
```



### Instalación de Python en entorno:

- El entorno viene por defecto vacío, sin Python (aunque probablemente vea el Python del entorno base)
- Se recomienda instalar Python nada más crear el entorno
- Se puede especificar una versión específica en caso de ser necesario. En caso contrario instalará la última



```
~$ conda install python[=<version>]
```



### Instalación de librerías en entorno (I):

- Las librerías se instalan desde unos repositorios llamados 'canales'
- Si no se especifica lo contrario, se buscará en el canal por defecto 'conda'
- Otro canal muy popular con las últimas librerías es 'conda-forge'
- Cualquier usuario puede tener su propio canal y subir librerías

```
~$ conda install [-c <nombre_canal>] <nombre_libreria>[=<version>]
```



## 5. conda: quickstart

### Instalación de librerías en entorno (II, plan de instalación):

- Antes de realizar la instalación se recomienda revisar el plan de instalación, donde se especifica los cambios que se producirán en el entorno durante la instalación:
  - Descargas
  - Instalación de nuevas librerías
  - Actualizaciones de librerías
  - Retrocesos de versiones
- Se ha de prestar especial atención al retroceso de versiones que se puede producir al instalar nuevas librerías

```
(analytics) pi@raspberrypi:~ $ conda install pandas
Solving environment: done

## Package Plan ##

environment location: /home/pi/berryconda3/envs/analytics

added / updated specs:
- pandas

The following packages will be downloaded:

package | build | size
-----|-----|-----
numpy-1.14.0 | py36h73aad69_0 | 7.4 MB
pytz-2018.5 | py_0 | 193 KB
six-1.11.0 | py36_1 | 20 KB
libgfortran-3.0.0 | 0 | 206 KB
openblas-0.2.19 | 0 | 2.6 MB
pandas-0.23.4 | py36h6b76cdf_0 | 25.8 MB
python-dateutil-2.6.1 | py_0 | 190 KB
-----|-----|-----
Total: | | 36.4 MB

The following NEW packages will be INSTALLED:

libgfortran: 3.0.0-0
numpy: 1.14.0-py36h73aad69_0
openblas: 0.2.19-0
pandas: 0.23.4-py36h6b76cdf_0
python-dateutil: 2.6.1-py_0
pytz: 2018.5-py_0
six: 1.11.0-py36_1

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



### Obtención de librerías instaladas en entorno:

- Con el entorno activado, se pueden obtener aquellas librerías actualmente instaladas en el entorno, así como:
  - Versión.
  - Origen de la librería (canal, conda/pip)

```
(analytics) pi@raspberrypi:~ $ conda list
# packages in environment at /home/pi/berryconda3/envs/analytics:
#
# Name                          Version                      Build      Channel
ca-certificates                 2018.8.24                    0
certifi                         2018.8.24                    py36_1
libgfortran                     3.0.0                        0
ncurses                         6.1                          h4f752ac_1
numpy                           1.14.0                       py36h73aad69_0
openblas                       0.2.19                       0
openssl                         1.0.2q                       hdf2a78_0
pandas                          0.23.4                       py36h6b76cdf_0
pip                             18.0                         py36_1
python                         3.6.6                       hd0568c0_1
python-dateutil                2.6.1                        py_0
pytz                           2018.5                       py_0
readline                       7.0                          hcb560eb_1
setuptools                     40.2.0                       py36_0
six                             1.11.0                       py36_1
sqlite                         3.24.0                       hfcb1bcf_1
tk                              8.6.8                       h849d6a0_0
wheel                          0.31.1                       py36_1
xz                             5.2.4                       hdf2a78_1
zlib                           1.2.11                       hdf2a78_1003
```



### Obtención de información de la librería (previa a instalación):

- Si se desea obtener el listado de opciones de instalación de una librería, se puede realizar una búsqueda de la misma
- Se puede especificar el canal de la librería para filtrar únicamente aquellas librerías disponibles en ciertos canales

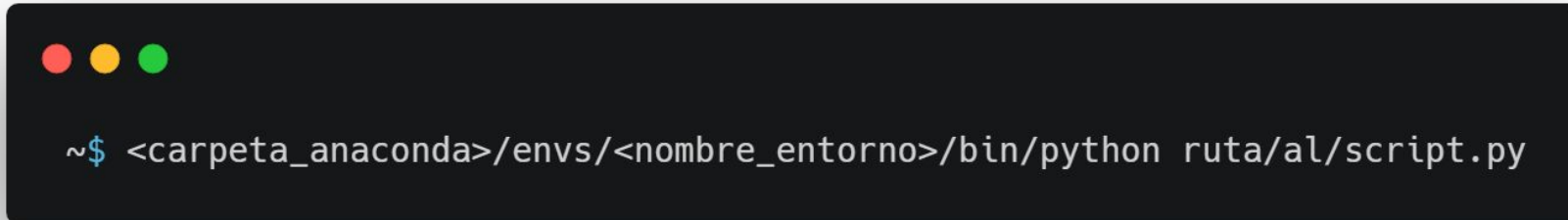
```
~$ conda search <nombre_librería> [-c <nombre_canal>]
```





### Ejecutable de Python del entorno:

- Para utilizar el ejecutable de Python de los entornos virtuales sin necesidad de activarlos previamente (por ejemplo desde procesos automáticos o programados) este se encuentra en la carpeta 'bin' del entorno virtual, por ejemplo, para ejecutar un script



```
~$ <carpeta_anaconda>/envs/<nombre_entorno>/bin/python ruta/al/script.py
```



### Eliminación de un entorno y sus librerías:

- Una vez un entorno no va a utilizarse más, puede eliminarse de forma sencilla sin dejar rastro
- Una vez se elimina un entorno no hay forma de recuperarlo, únicamente realizando la creación de uno del mismo nombre desde cero



```
~$ conda remove -n <nombre_entorno> --all
```



### Distribución de entornos (I, yml):

- Una forma de distribuir un entorno entre equipos que tienen conda instalado es definirlo en un archivo .yml de configuración

```
name: airflow
channels:
  - conda
  - conda-forge
dependencies:
  - opencv
  - airflow-with-postgres
  - pyhive
# - thrift_sasl=0.2.1
# - psycpg2
  - tensorflow
  - h5py
  - pip:
    - hmsclient
```

```
~$ conda env create -f ruta/al/archivo.yml
```



## 5. conda: quickstart

### Distribución de entornos (II, conda-pack):

- En el caso de que conda (o incluso Python) no se encuentren instalados en el sistema, la mejor opción es utilizar conda-pack
- conda-pack es una utilidad que empaqueta un entorno en un archivo .tar.gz y que permite distribuirlo en cualquier máquina Linux tenga o no instalado Python
- Para distribuir el entorno empaquetado con conda-pack, se ha de copiar el archivo generado a la máquina correspondiente, descomprimirlo en una carpeta, y ejecutar el script activate que hay en la carpeta 'bin' del mismo:

```
~$ conda install -c conda-forge conda-pack
```

```
(analytics) pi@raspberrypi:~ $ conda pack
Collecting packages...
Packing environment at '/home/pi/berryconda3/envs/analytics' to 'analytics.tar.gz'
[#####] | 100% Completed | 2min 10.1s
```

```
~$ mkdir carpeta_entorno
~$ tar xzvf entorno.tar.gz -C carpeta_entorno/
~$ source carpeta_entorno/bin/activate
~$ conda-unpack
```



### Alternativas a conda: virtualenv

- Otra solución para crear entornos virtuales (con menos funcionalidades que conda) es virtualenv
- Para crear un entorno virtual, se debe previamente crear una carpeta del proyecto
- El entorno virtual se aloja en una carpeta en el interior

```
$ mkdir project_folder  
$ cd project_folder  
$ virtualenv environment_name  
$ source environment_name/bin/activateQ
```