


Instalación de Miniconda3

Conda es un administrador de paquetes de Python (y de otros lenguajes como R, Ruby, Java, C++, etc.) y ambientes virtuales. En este curso lo usaremos para crear ambientes virtuales de una manera sencilla. Un ambiente virtual es un “espacio” donde instalamos todas las dependencias que queremos usar para cierto proyecto o conjunto de proyectos. También lo podemos usar para probar nuestros programas en diferentes versiones de Python o con diferentes librerías (o versiones de librerías) sin generar problemas de dependencias entre paqueterías. Conda viene como parte de la plataforma Anaconda. Sin embargo, debido a que únicamente requeriremos Conda (y no otros elementos de Anaconda como Jupyter), lo obtendremos mediante la instalación de Miniconda, un paquete más ligero únicamente contiene lo que necesitamos. En este documento se mostrará cómo instalar y usar Miniconda3 en Windows.

Instalación

Podemos instalar miniconda desde <https://docs.conda.io/en/latest/miniconda.html>. Ahí seleccionaremos el instalador que queremos usar (para este ejemplo será Python 3.7 para Windows de 64-bit).

Windows installers

Windows			
Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3 Windows 64-bit	51.5 MiB	 b50ae75e4d602b7ce35197c8e31e81288d069b758594f1bb46ab45
	Miniconda3 Windows 32-bit	54.0 MiB	7c30778941d2bba03531ba269a78a108b01fa366530290376e7c3b467f3c66ba

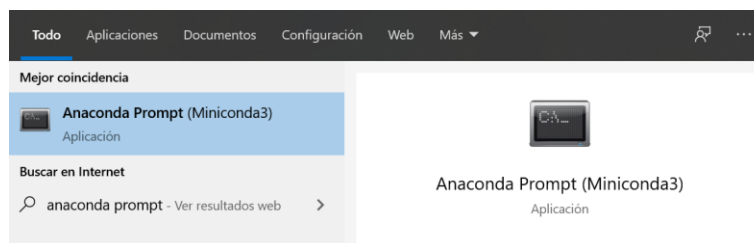
Al descargarlo, ejecutamos el instalador. Presionamos Next y aceptamos los términos de la licencia. En las opciones de tipo de instalación, seleccionaremos

- Just Me (recommended)
- La carpeta de instalación¹ (default: C:\Users\TU_USUARIO\Miniconda3)
- Seleccionar únicamente “Register Anaconda as my default Python 3.7”

Después seleccionamos “Instalar” y esperamos a que termine.

Comprobar instalación

Para comprobar la instalación, buscamos en Inicio “Anaconda Prompt” y lo ejecutamos



¹ Nota: el nombre de usuario **no** debe de contener espacios.

Al ejecutarlo, deberíamos de ver la consola con la palabra **(base)** antes del prompt

```
Anaconda Prompt (Miniconda3)

(base) C:\Users\Piton>
```

Podemos comprobar la instalación escribiendo el comando `conda info`, lo que nos dirá la versión de conda.

Crear ambientes virtuales

Para crear un ambiente virtual usamos

```
> conda create --name NOMBRE_DEL_AMBIENTE python=VERSION_DE_PYTHON
```

Por ejemplo, si queremos crear un ambiente con python 3.6 ejecutamos

```
> conda create --name mi-primer-ambiente python=3.6
```

Este comando nos descargará las librerías iniciales para poder usar la versión de Python que escribimos

Activar ambientes

Activar un ambiente significa usar el interpretador de python de dicho ambiente. Para activarlo simplemente usamos

```
> conda activate NOMBRE_DEL_AMBIENTE
```

Para saber cuál es el ambiente actualmente activo, nos fijamos en el nombre que está a la izquierda entre paréntesis del prompt

```
Anaconda Prompt (Miniconda3)

(base) C:\Users\Piton>conda activate mi-primer-ambiente
(mi-primer-ambiente) C:\Users\Piton>
```

Listar los ambientes

Para mostrar una lista con todos los ambientes usamos el comando

```
> conda env list
```

Clonar ambientes

Si queremos crear un nuevo ambiente clonando uno existente usamos el comando

```
> conda create --clone NOMBRE_DEL_AMBIENTE_A_CLONAR --name NUEVO_AMBIENTE
```

Borrar ambientes

Para eliminar un ambiente y todo su contenido, usamos el comando

```
> conda env remove --name NOMBRE_DEL_AMBIENTE
```

Todo el contenido de la carpeta de dicho ambiente (como librerías) será borrado. Las librerías con el mismo nombre instaladas en otro ambiente no se verán afectadas.

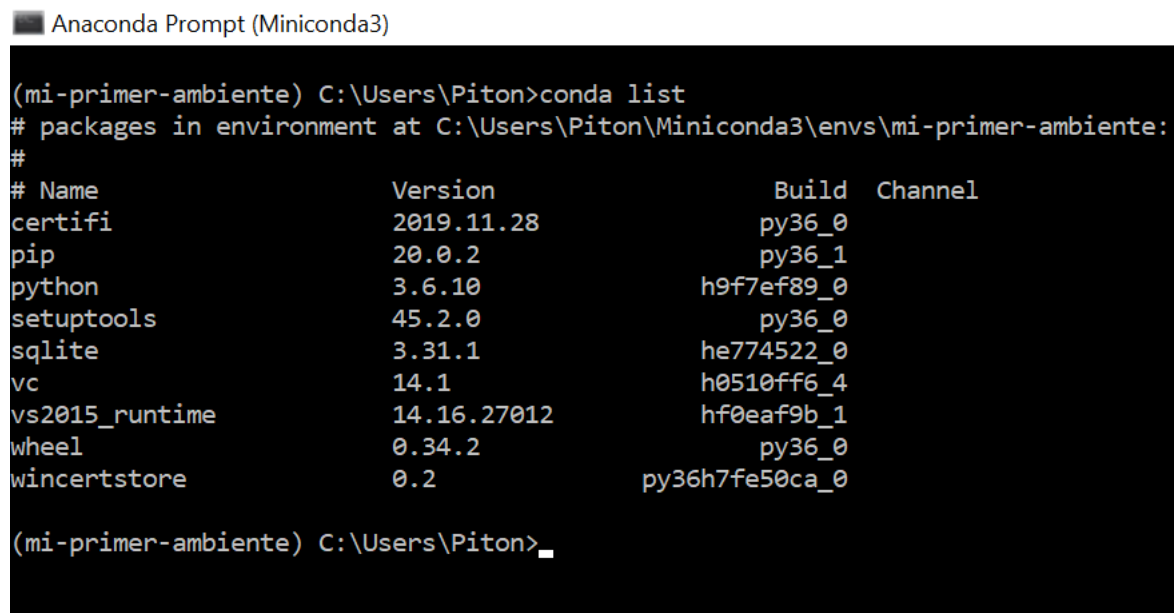
Enlistar las librerías de un ambiente

Para mostrar las librerías instaladas en un ambiente usamos el comando

```
> conda list --name NOMBRE_DEL_AMBIENTE
```

Para listar las librerías del ambiente activo únicamente usamos

```
> conda list
```



```
Anaconda Prompt (Miniconda3)

(mi-primer-ambiente) C:\Users\Piton>conda list
# packages in environment at C:\Users\Piton\Miniconda3\envs\mi-primer-ambiente:
#
# Name                      Version                Build    Channel
certifi                     2019.11.28             py36_0
pip                         20.0.2                 py36_1
python                      3.6.10                 h9f7ef89_0
setuptools                  45.2.0                 py36_0
sqlite                      3.31.1                 he774522_0
vc                          14.1                   h0510ff6_4
vs2015_runtime              14.16.27012            hf0eaf9b_1
wheel                       0.34.2                 py36_0
wincertstore                0.2                    py36h7fe50ca_0

(mi-primer-ambiente) C:\Users\Piton>_
```

Buscar librerías

Para buscar si una librería está en conda, usamos el siguiente comando

```
> conda search NOMBRE_DE_LA_LIBRERÍA
```

Por ejemplo, si queremos ver el canal de distribución de pymysql (una librería que sirve como interfaz de bases de datos en MySQL)

```
Anaconda Prompt (Miniconda3)

(mi-primer-ambiente) C:\Users\Piton>conda search pymysql
Loading channels: done
# Name                               Version          Build           Channel
pymysql                             0.7.11          py27hbd53ec_0   pkgs/main
pymysql                             0.7.11          py35h7f9ff51_0   pkgs/main
pymysql                             0.7.11          py36hf59f3ba_0   pkgs/main
pymysql                             0.8.0           py27_0          pkgs/main
pymysql                             0.8.0           py35_0          pkgs/main
pymysql                             0.8.0           py36_0          pkgs/main
pymysql                             0.8.1           py27_0          pkgs/main
pymysql                             0.8.1           py35_0          pkgs/main
pymysql                             0.8.1           py36_0          pkgs/main
pymysql                             0.9.1           py27_0          pkgs/main
pymysql                             0.9.1           py35_0          pkgs/main
pymysql                             0.9.1           py36_0          pkgs/main
pymysql                             0.9.1           py37_0          pkgs/main
pymysql                             0.9.2           py27_0          pkgs/main
pymysql                             0.9.2           py35_0          pkgs/main
pymysql                             0.9.2           py36_0          pkgs/main
pymysql                             0.9.2           py37_0          pkgs/main
pymysql                             0.9.2           py38_0          pkgs/main
pymysql                             0.9.3           py27_0          pkgs/main
pymysql                             0.9.3           py36_0          pkgs/main
pymysql                             0.9.3           py37_0          pkgs/main
```

Instalar librerías

Para instalar una librería en el ambiente virtual activo, ejecutamos

```
> conda install NOMBRE_DE_LA_LIBRERIA
```

Algunas librerías necesitan de otras librerías que no están instaladas en el ambiente, por lo que se instalarán de igual forma. Estas son las librerías en el ambiente de ejemplo después de instalar pymysql

```
(mi-primer-ambiente) C:\Users\Piton>conda list
# packages in environment at C:\Users\Piton\Miniconda3\envs\mi-primer-ambiente:
#
# Name                               Version          Build           Channel
asn1crypto                          1.3.0            py36_0
ca-certificates                     2020.1.1         0
certifi                             2019.11.28       py36_0
cffi                                1.14.0           py36h7a1dbc1_0
cryptography                        2.8              py36h7a1dbc1_0
idna                                2.8              py36_0
openssl                             1.1.1d           he774522_4
pip                                 20.0.2           py36_1
pycparser                           2.19             py36_0
pymysql                             0.9.3            py36_0
python                              3.6.10           hf9f7ef89_0
setuptools                          45.2.0           py36_0
six                                 1.14.0           py36_0
sqlite                              3.31.1           he774522_0
vc                                  14.1             h0510ff6_4
vs2015_runtime                     14.16.27012      hf0eaf9b_1
wheel                               0.34.2           py36_0
wincertstore                        0.2              py36h7fe50ca_0
```

Como se puede ver, se instalaron otras librerías como “openssl” y “six”.