

Guía de instalación de Conda en Windows, Linux y MacOX

ffelix@inf.uc3m.es

31 de enero de 2021

- El objetivo de esta guía es ayudar en la instalación de un gestor de entornos para poder programar cómodamente en Python controlando las dependencias.
- **NO** es obligatorio instalarlo. Es completamente opcional. Es, sin embargo, recomendable y una práctica muy común entre los desarrolladores en Python.
- Este documento es una guía de instalación básica y muestra cómo crear un primer entorno para realizar las tareas de la asignatura. No es un tutorial sobre Conda.

1. Introducción

Hay varios problemas prácticos a la hora de trabajar con Python en cualquier sistema operativo. Por regla general, tanto Linux como macOS vienen con una instalación predeterminada de Python (ya sea versión 2.X o versión 3.x). A veces, el software con el que trabajemos puede requerir usar una u otra versión o depender de paquetes que están en una versión pero no en otra. El hecho cierto es que intentar mantener dos o más instalaciones independientes de Python en un mismo sistema suele acabar en pesadilla. No os lo aconsejo.

Conda es un software que nos ayuda muchísimo en esta tarea. Es un sistema, de código abierto, para gestión de paquetes y gestión de entornos que corre en Windows, Linux y macOS. Conda instala, ejecuta y actualiza paquetes y sus dependencias. Conda, además, permite crear, salvar, cargar y cambiar entre entornos diferentes dentro de tu propio PC realmente rápido.

Hay dos distribuciones donde podemos encontrar Conda, Anaconda y Miniconda. La primera es una distribución muy grande, con muchas herramientas para trabajo científico. No nos interesa ahora mismo. Miniconda solo trae lo imprescindible para poder gestionar los entornos y los paquetes. En el resto del documento describiremos como instalar Miniconda y crear nuestro primer entorno.

2. Requerimientos del sistema

Para instalar Miniconda, los requerimientos de nuestro sistema son muy modestos:

- Computador con arquitectura de 32 o 64 bits
- 400 MiB de espacio en disco
- Sistema operativo: Windows 10, Linux, macOS

Si empezamos a crear multiples entornos y a instalar muchos paquetes en cada uno de ellos, las necesidades de disco crecerán proporcionalmente. En cualquier caso, en lo que nos atañe, esto no debería ser un problema.

3. Windows

Para instalar Miniconda en Windows procederemos siguiendo los siguientes pasos:

1. Sigue este enlace (<https://conda.io/miniconda.html>) donde aparecen la lista de instaladores disponibles.
2. A no ser que tengas un equipo anormalmente antiguo, escogeremos la versión de Python 3.X y 64-bits y la descargamos (https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe).
3. Ejecuta el instalador recién descargado. Si no estas seguro sobre algunas de las opciones que se proponen, escoge el valor propuesto por defecto. La configuración se puede cambiar posteriormente.
4. Una vez finalizada la instalación, borra el instalador y, a continuación, haz clic en el botón de Inicio de Windows. Deberían aparecer dos nuevos programas en lo alto de la lista. Abre el que se denomina ".Anaconda Powershell Prompt (miniconda3)"
5. Finalmente, para comprobar que todo funciona correctamente, en la ventana de terminal que se ha abierto, teclea: `conda list`. Una lista de paquetes instalados en el entorno **base** debería aparecerte.

4. Linux/macOS

Para instalar Miniconda en Linux o macOS procederemos siguiendo los siguientes pasos:

1. Sigue este enlace (<https://conda.io/miniconda.html>) donde aparecen la lista de instaladores disponibles.
2. A no ser que tengas un equipo anormalmente antiguo, escogeremos la versión de Python 3.X y 64-bits y la descargamos (https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh).
3. Abre un terminal y localiza el instalador recién descargado. Ejecútalo tecleando:

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

Si no estas seguro sobre algunas de las opciones que se proponen, escoge el valor propuesto por defecto. La configuración se puede cambiar posteriormente.

4. Una vez finalizada la instalación, borra el instalador y, cierra y reabre tu ventana del terminal. Conda ha modificado el fichero de inicialización de la *shell* y necesitamos hacer esto para que los cambios tengan efecto
5. Finalmente, para comprobar que todo funciona correctamente, en la ventana de terminal que se ha abierto, teclea: `conda list`. Una lista de paquetes instalados en el entorno **base** debería aparecerte.

5. Crear un entorno para los tutoriales con Pacman

A continuación, vamos a crear un entorno nuevo, distinto al entorno **base** creado por defecto. Dicho entorno lo denominaremos "**pacman**", ya que pretendemos usarlo para trabajar con la recreación del famoso juego que son el objeto de algunos tutoriales y prácticas de la asignatura. Antes de crearlo debemos tener en cuenta que nuestro *pacman* necesita la versión 2.7 de Python, por tanto, el entorno que creemos debería usar esa versión del lenguaje.

```
$ conda create --name pacman python=2.7
```

Y esta es la ventaja de usar Conda. Aunque la versión de Conda que descargamos era para Python 3.8 (la versión que, de hecho, tiene el entorno base), hemos creado un entorno donde la versión de Python es 2.7 y es completamente independiente de cualquier otra existente o que creemos en el futuro. Podemos tener múltiples entornos, cada uno con una versión distinta de Python y con aquellos paquetes que nos interesen en cada caso.

5.1. Cambiar de entorno

Para cambiar de entorno, basta con escribir:

```
$ conda activate pacman
```

Tengamos en cuenta que el hecho de cambiar de entorno no tiene ningún efecto sobre el sistema de ficheros sobre el que trabajamos. La única diferencia es que, cuando llamemos a Python, se usará la versión que hayamos especificado para el entorno que tengamos activo y los paquetes disponibles serán aquellos que hayamos instalado en el mismo. Dónde se crean los distintos entornos con el correspondiente intérprete y los paquetes que decidamos no nos interesa ahora mismo, los curiosos, si quieren, pueden echar un vistazo en:

- **Linux** : /home/<username>/miniconda3
- **macOS** : /Users/<username>/miniconda3
- **Windows** : C:\Users\<username>\miniconda3

5.2. Instalar un paquete en el entorno activo

Imaginemos que necesitésemos, por alguna razón, instalar un determinado paquete Python en un entorno Conda que hayamos creado ¿cómo procedemos?. En primer lugar, debemos asegurarnos que el entorno elegido está realmente activo¹:

```
$ conda activate pacman
```

Supongamos, ahora, que queremos instalar el paquete NumPy (una biblioteca de estructuras de datos archiconocida, base de muchos otros paquetes para ciencia de datos y aprendizaje automático):

```
$ conda install numpy
```

Para ver qué paquetes tenemos actualmente instalados en el entorno activo:

```
$ conda list
```

A veces, Conda no tiene el paquete que nos interesa instalar en su repositorio público (tiene cientos pero no todos). En ese caso, podemos recurrir a PIP que es otra herramienta del mundo Python para instalar paquetes, muy popular también, que se puede usar con Conda. Bastará con ejecutar:

```
$ pip install package_name
```

PIP también instalará el paquete en cuestión dentro del entorno activo. Se recomienda intentar instalar los paquetes usando Conda en primer lugar

6. Recursos

Hay múltiples y variados recursos disponibles en Internet sobre cómo trabajar con Conda. Aquí solo hemos apuntado lo **mínimo que necesitamos saber** para realizar los tutoriales. Para aprender a manejarse con Conda de manera experta, en caso de que se estuviera interesado, recomendamos acudir a la documentación oficial en primer lugar:

- **Documentación sobre Conda** <https://docs.conda.io/projects/conda/en/latest/index.html>

¹Es muy habitual olvidar esto y acabar instalando el paquete en el entorno base o cualquier otro