

Crear un entorno virtual con `conda`

Mathieu Kessler

Departamento de Matemática Aplicada y Estadística
Universidad Politécnica de Cartagena

28 de enero de 2021



Un entorno virtual permite crear un espacio de trabajo aislado para un proyecto. Se realiza para evitar conflictos de versiones de paquetes entre diferentes proyectos. Quizás uno de vuestro proyecto depende de una versión anterior de un paquete que otro proyecto más reciente. Si los dos proyectos se apoyan en el entorno ‘base’ de conda, la actualización del paquete para un proyecto podría implicar que el otro dejara de funcionar.

Un entorno virtual permite crear un espacio de trabajo aislado para un proyecto. Se realiza para evitar conflictos de versiones de paquetes entre diferentes proyectos. Quizás uno de vuestro proyecto depende de una versión anterior de un paquete que otro proyecto más reciente. Si los dos proyectos se apoyan en el entorno ‘‘base’’ de conda, la actualización del paquete para un proyecto podría implicar que el otro dejara de funcionar.

La solución es crear un entorno virtual para un proyecto, que usará su propia versión de Python y de las librerías que necesita.

Un entorno virtual permite crear un espacio de trabajo aislado para un proyecto. Se realiza para evitar conflictos de versiones de paquetes entre diferentes proyectos. Quizás uno de vuestro proyecto depende de una versión anterior de un paquete que otro proyecto más reciente. Si los dos proyectos se apoyan en el entorno ‘‘base’’ de conda, la actualización del paquete para un proyecto podría implicar que el otro dejara de funcionar.

La solución es crear un entorno virtual para un proyecto, que usará su propia versión de Python y de las librerías que necesita.

Entornos virtuales en conda

Existen varios gestores de entornos virtuales. Conda tiene el suyo propio, la documentación está disponible en el [documento Anaconda sobre gestión entornos](#).

Cinco puntos básicos

- La creación de un entorno virtual se realiza desde la consola de Anaconda.
- El comando básico para crear un entorno virtual es `conda create --name myenv` donde `myenv` es el nombre que habéis escogido para vuestro entorno virtual.
- Una vez creado, tenéis que activarlo si queréis trabajar con él desde la consola de Anaconda.
- En Visual Studio Code, puede asignar un entorno virtual a un workspace y todos los ficheros python de este proyecto se ejecutarán en este entorno.
- Podéis crear un entorno virtual “bare” `conda create --name myenv python`, activarlo y después instalar los paquetes necesarios para vuestro proyecto con `conda install`, o podéis directamente especificar los paquetes requeridos en el momento de la creación del entorno.

Cinco puntos básicos

- La creación de un entorno virtual se realiza desde la consola de Anaconda.
- El comando básico para crear un entorno virtual es `conda create --name myenv` donde `myenv` es el nombre que habéis escogido para vuestro entorno virtual.
- Una vez creado, tenéis que activarlo si queréis trabajar con él desde la consola de Anaconda.
- En Visual Studio Code, puede asignar un entorno virtual a un workspace y todos los ficheros python de este proyecto se ejecutarán en este entorno.
- Podéis crear un entorno virtual “bare” `conda create --name myenv python`, activarlo y después instalar los paquetes necesarios para vuestro proyecto con `conda install`, o podéis directamente especificar los paquetes requeridos en el momento de la creación del entorno.

Vamos a crear el entorno virtual `ids` para la asignatura, con las librerías que necesitaremos y veremos cómo asignarlo a nuestro workspace en Visual Studio Code.

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a instalar los siguientes paquetes para IDS:

- `numpy`
- `matplotlib`
- `pandas`
- `scikit-learn`
- `ipykernel`

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a instalar los siguientes paquetes para IDS:

- `numpy` Un paquete para la computación científica. <https://numpy.org/>
- `matplotlib`
- `pandas`
- `scikit-learn`
- `ipykernel`

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a instalar los siguientes paquetes para IDS:

- `numpy` Un paquete para la computación científica. <https://numpy.org/>
- `matplotlib` Librería para crear visualizaciones. <https://matplotlib.org/>
- `pandas`
- `scikit-learn`
- `ipykernel`

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a instalar los siguientes paquetes para IDS:

- `numpy` Un paquete para la computación científica. <https://numpy.org/>
- `matplotlib` Librería para crear visualizaciones. <https://matplotlib.org/>
- `pandas` Análisis y manipulación de datos <https://pandas.pydata.org/>
- `scikit-learn`
- `ipykernel`

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a instalar los siguientes paquetes para IDS:

- `numpy` Un paquete para la computación científica. <https://numpy.org/>
- `matplotlib` Librería para crear visualizaciones. <https://matplotlib.org/>
- `pandas` Análisis y manipulación de datos <https://pandas.pydata.org/>
- `scikit-learn` Algoritmos de machine learning <https://scikit-learn.org/stable/>
- `ipykernel`

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a instalar los siguientes paquetes para IDS:

- `numpy` Un paquete para la computación científica. <https://numpy.org/>
- `matplotlib` Librería para crear visualizaciones. <https://matplotlib.org/>
- `pandas` Análisis y manipulación de datos <https://pandas.pydata.org/>
- `scikit-learn` Algoritmos de machine learning <https://scikit-learn.org/stable/>
- `ipykernel` Proporciona un entorno interactivo para blocs de notas Jupyter.

Vamos a crear el entorno virtual e instalar directamente los paquetes relevantes.

Instalación de paquetes y su gestión en conda

El gestor de paquetes de conda los descarga desde [Anaconda cloud](#), donde la compañía Anaconda y otras mantienen versiones binarias para diferentes sistemas operativos.

Vamos a instalar los siguientes paquetes para IDS:

- `numpy` Un paquete para la computación científica. <https://numpy.org/>
- `matplotlib` Librería para crear visualizaciones. <https://matplotlib.org/>
- `pandas` Análisis y manipulación de datos <https://pandas.pydata.org/>
- `scikit-learn` Algoritmos de machine learning <https://scikit-learn.org/stable/>
- `ipykernel` Proporciona un entorno interactivo para blocs de notas Jupyter.

Es recomendable instalar a la vez todos los paquetes que pensamos que vamos a utilizar. La instalación posterior de un paquete puede dar lugar a conflictos de versiones.

Instalaremos también un paquete para el auto-formateo del código.

- `autopep8`

Instalaremos también un paquete para el auto-formateo del código.

- `autopep8` Un formateador que busca cumplir con el estilo de código PEP8

Instalaremos también un paquete para el auto-formateo del código.

- `autopep8` Un formateador que busca cumplir con el estilo de código PEP8

En la consola de Anaconda, teclead (en una línea)

```
conda create --name ids -c conda-forge python numpy pandas  
scikit-learn matplotlib ipykernel autopep8
```

Instalaremos también un paquete para el auto-formateo del código.

- `autopep8` Un formateador que busca cumplir con el estilo de código PEP8

En la consola de Anaconda, teclead (en una línea)

```
conda create --name ids -c conda-forge python numpy pandas  
scikit-learn matplotlib ipykernel autopep8
```

Otra opción más sencilla es descargar del Aula virtual el fichero de requerimientos `ids_environment.yml` y en la consola de Anaconda escribir

```
conda env create -f ids_environment.yml
```

Instalaremos también un paquete para el auto-formateo del código.

- `autopep8` Un formateador que busca cumplir con el estilo de código PEP8

En la consola de Anaconda, teclead (en una línea)

```
conda create --name ids -c conda-forge python numpy pandas  
scikit-learn matplotlib ipykernel autopep8
```

Otra opción más sencilla es descargar del Aula virtual el fichero de requerimientos `ids_environment.yml` y en la consola de Anaconda escribir

```
conda env create -f ids_environment.yml
```

Para comprobar que se ha creado correctamente

En la consola de Anaconda, activad el entorno con `conda activate ids` y a continuación, obtened la lista de paquetes con `conda list`, comprobando que están los requeridos.