

# Instalación del entorno de Python

Instalación y gestión de entornos virtuales con Anaconda

## Índice

[¿Por qué usar entornos virtuales?](#)

[Descargar e instalar Anaconda](#)

[Gestión de entornos virtuales](#)

[Línea de comandos](#)

[Anaconda Navigator](#)

[Preguntas frecuentes](#)

## ¿Por qué utilizar entornos virtuales?

Cuando en nuestro día a día escribimos líneas de código en Python, lo más habitual es que nos apoyemos en librerías o módulos que nos aporten herramientas adicionales para poder llevar a cabo nuestro cometido de forma más eficiente o confortable. Dichos módulos, por lo general, están vivos, es decir, hay desarrolladores detrás de ellos que no dejan de mejorarlos o evolucionarlos continuamente. Estas mejoras suelen ser transparentes para nosotros, pero en algunas ocasiones pueden conllevar cambios tan drásticos que impliquen que nuestro programa deje de funcionar, obligándonos a realizar cambios a nosotros también si queremos recuperar su buen funcionamiento.

Afortunadamente, los módulos están versionados, por lo que podemos anotar la versión de cada módulo que empleemos para poder reproducir su ejecución en el futuro. Sin embargo, es posible que queramos tener diferentes programas en nuestra máquina y que cada uno de ellos necesite diferentes versiones de un mismo módulo. En dado caso, tendríamos que estar reinstalando la versión necesaria para cada uno de ellos cada vez que quisiéramos ejecutarlos.

Por suerte, existen los llamados entornos virtuales, entornos aislados donde podremos instalar las versiones de los módulos que vayamos a requerir durante la ejecución de nuestro proyecto, guardando consigo una foto de sus dependencias. En una misma máquina podemos tener más de un entorno virtual configurado y nos bastará con activar el entorno adecuado para cada programa para que éste se ejecute correctamente.

Con el objetivo de que cada una de los Notebooks de cada una de las PECs funcione como es debido, se facilitará un entorno virtual descrito en el formato YAML (en un fichero .yaml) con los módulos necesarios y sus versiones adecuadas.

## Descargar e instalar Anaconda

Clica *Download* en el siguiente enlace: <https://www.anaconda.com/download/>

Después, elige la versión de Anaconda para el sistema operativo (Windows, MacOS o Linux) y la arquitectura (32-Bit o 64-Bit) de tu equipo e instálalo siguiendo los pasos del instalador gráfico. Si tienes dudas, probablemente tengas 64-Bit.

## Gestión de entornos virtuales

La gestión de los entornos virtuales puede realizarse de dos formas distintas: usando la línea de comandos o a través de la interfaz gráfica *Anaconda Navigator*. Puede emplearse indistintamente una u otra opción y también pueden combinarse ambas, aunque es suficiente con utilizar únicamente una de ellas. A continuación, detallaremos cómo proceder con cada una de las dos opciones.

### Línea de comandos

Crearemos el entorno descrito en el fichero ***environment\_X.yaml*** (donde X es la PEC a la que hace referencia) con el siguiente comando:

```
conda env create -f environment_X.yaml
```

Una vez creado, deberemos activar el entorno que acabamos de crear, cuyo nombre vendrá especificado en la sección “***name:***” del mismo fichero. Para ello, ejecutaremos el siguiente comando:

```
conda activate environment_name
```

Con el entorno virtual activo, ya puedes ejecutar el comando ***jupyter notebook***, con lo que arrancarás *Jupyter*.

Aquí puedes encontrar más información sobre la gestión de entornos virtuales con Anaconda: <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

# Anaconda Navigator

Abre *Anaconda Navigator*, clicas en la sección **Environments** (a la izquierda), después en **Import** (abajo), localiza el fichero **environment\_X.yml** (donde X es la PEC a la que hace referencia) que desees importar y clicas en **Import** nuevamente. Una vez finalizada la importación, el entorno aparecerá en la lista y podrá ser activado cuando sea necesario. Sabremos que un entorno está activo cuando a su derecha tenga un triángulo verde. Si queremos activar un entorno, bastará con clicar dos veces seguidas sobre el entorno que deseemos activar. En el entorno activo, si clicamos en el triángulo verde, podremos abrir (entre otras cosas) un terminal para lanzar comandos o *Jupyter Notebook*.

## Preguntas frecuentes

**¿Por qué, cuando ejecuto una celda del Notebook, me sale “*ModuleNotFoundError: No module named ...*” si el módulo en cuestión aparece en el fichero del entorno importado?**

→ Dicho entorno ha de estar activo, no sólo importado. Abra *Jupyter Notebook* desde el entorno cuando esté activo.

**Si necesito utilizar otros módulos además de los que ya vienen en el entorno, ¿cómo los instalo?**

→ *Línea de comandos*: una vez dentro del entorno virtual (tras el `conda activate <nombre del entorno>`), puedes instalar un módulo con:

```
conda install <nombre módulo>
```

→ *Anaconda Navigator*: dentro de la sección **Environments** (a la izquierda), clicas en el entorno en que quieras instalar un módulo nuevo y, en el desplegable de la derecha, selecciona *Not installed* u *All* en vez de *Installed*, por último, introduce el nombre del módulo en el recuadro *Search Packages*. Si no ves el módulo, prueba a actualizar el índice de búsqueda pulsando *Update index...* o añadiendo algún canal adicional en *Channels*.

→ *Desde el propio enunciado de la PEC (notebook de Jupyter)*: habiendo arrancado el notebook desde el entorno virtual facilitado. En una celda cualquiera escribir:

```
conda install -y <nombre módulo>
```

Con el parámetro `-y` estarás indicando “sí a todo” de manera que no espere confirmación, lo que haría la celda se quede bloqueada sin opciones de respuesta. Así solucionamos el problema e instalará la librería.