Curso Nbdev Introducción

Instalación – Curso nbdev

# Instalación

Para hacer uso de *nbdev* es necesario los siguientes softwares:

* Python (Python o [conda](https://www.anaconda.com/))
* Jupyter Notebooks
* Quarto
* Github
* nbdev

A continuación se da una breve explicación de cada uno de estos y como lo utilizamos.

## Ambientes virtuales (conda) (Tema 2.1)

Anaconda es una empresa la cual desarrollo un manejador de ambientes virtuales con un manejador de paquetes (conda), éste nos permite la creación de ambientes virtuales. Dentro de dichos ambientes se puede hacer un manejo eficiente de los diferentes softwares instalados dentro del ambiente virtual y las dependencias de los mismos. Al tener un manejo eficiente de las paqueterias y las dependencias dentro de este se hace más fácil la creación de ambientes de desarrollo.

De esta forma algo instalado dentro de un ambiente virtual no afectara a las distintas paqueterias que se encuentran fuera del ambiente virtual. Lo cual nos permite tener tener aislado un ambiente de desarrollo con respecto al sistema operativo u otro ambiente de desarrollo.

### Instalación Anaconda (conda)

Para la instalación de conda lo más sencillo es seguir el este [manual](https://docs.conda.io/projects/conda/en/latest/user-guide/install/windows.html) aquí sólo se hace un esbozo de la instalación.

1. Bajar el instalador:
   * [Anaconda](https://www.anaconda.com/download/) para windows.
2. Verificar que se ha bajano correctamente ([ver](https://docs.conda.io/projects/conda/en/latest/user-guide/install/download.html#hash-verification) ).
3. Hacer doble click en el archivo .exe (similar a Anaconda3-######-Windows-x86\_64.exe). Para la instalación en Linux o Mac se debe de bajar el archivo Anaconda3-####-Linux-###.sh o bien el Anaconda3-####-MACOSX-###.sh y ejecutar el archivo usando la terminal.
4. Seguir las instrucciones en la pantalla en caso de windows. Y seguir la instrucciones que apareceran en caso del uso de la terminal.
5. Para probar la instalación abra la terminal de windows (cmd) o el anaconda prompt y corra el comando conda list. Una lista de paquetes aparecera si se instaló de forma correcta.

### Hacer un ambiente virtual

Para generar un ambiente virtual usando anaconda se puede hacer a través de la terminal. En caso de windows se puede hacer llamando a **Anaconda Prompt**.

Dentro de la terminal se puede observar que antes del símbolo del sistema aparece la leyenda (base) esta no indica que estamos usando el ambiente virtual base.

Para crear un nuevo ambiente virtual se ejecuta la siguiente linea en la linea de comandos.

conda create -n desarrollo

Este comando crea un nuevo ambiente virtual llamano desarrollo. Se puede especificar la versión de python a instalar dentro del ambiente virtual. También es posible instalar otros lenguajes de programación dentro del ambiente virtual cono son R o Julia.

Para entrar al ambiente virtual la forma de hacerlo es ejecutar

conda activate desarrollo

Al ejecutar el comando se puede observar que la leyenda (base) fue sustituida por la leyenda (desarrollo) lo cual nos indica que hemos activado nuestro ambiente virtual.

### Canales

Dentro de nuestro ambiente virtual podemos instalar las distintas bibliotecas y paquetes que sean necesarios. Por defecto Anaconda tiene un canal para poner a la disposición de la comunidad, que utiliza conda, muchos paquetes. Anaconda es la encargada de verificar que el el desarrollo de dichos paquetes sean de acuerdo a las mismas reglas establecidas por Anaconda. De la misma forma hay distintos canales los cuales tiene paquetes que estan de acuerdo a las reglas establecidas por cada canal. Los distintos canales tienen distintos objetivos por ejemplo [bioconda](https://bioconda.github.io/) contiene las paqueterias desarrolladas relacionadas con desarrollos biomédicos.

Otro canal muy utilizado es [conda-forge](https://conda-forge.org/), el cual contiene paquetes desarrollados por una extensa comunidad. Por facilidad en este curso utilizaremos el canal conda-forge pues tiene la mayoria de las paqueterias necesarias y se encuentran muchas versiones de los mismos.

### Uso de Conda para instalación paquetes

Para instalar un paquete dentro del ambiente virtual es necesario activar el ambiente. Ya dentro de este para hacer la instalación de los distintos paquetes de python tenemos dos formas distintas **PIP** o **conda**. PIP es el manejador de paquetes usado por python por defecto si se a trabajado anteriormente con Python lo más seguro es estar familiarizado con su uso.

**conda** es el manejador de paquetes del ambiente virtual y nos permite tanto hacer instalar paquetería de python y otras. Por ejemplo podemos instalar R o Julia haciendo uso de conda.

Como ejemplo para instalar numpy usando conda y el canal conda-forge executamos

(desarrollo) $ conda install -c conda-forge numpy

## Instalar Jupyter Notebook (Tema 2.2)

Jupyter es un conjunto de softwares que nos permite hacer uso de multiples herramientas, una de sus principales herramientas es el uso de noteboooks los cuales son documentos donde podemos escribir texto plano y hacer ejecución de código. Para que los documentos finales tengan una mayor riqueza se utiliza [Markdown](https://en.wikipedia.org/wiki/Markdown) y adicionalmente se puede hacer uso de [LaTex](https://en.wikipedia.org/wiki/LaTeX) para escribir fórmulas matemáticas.

A través del uso de JupyterLab es posible hacer desarrollo de software de forma interactiva usando notebooks, código e información. Esto ayuda al desarrollador a generar un archivo que tiene todo el contexto del desarrollo y entender la forma en la cual el software fue desarrollado, por que se tomaron ciertas decisiones en la creación del código. Esto es particularmente útil en el análisis exploratorio de datos, en la ciencia de dátos y en el desarrollo de la inteligencia artificial.

El uso de los notebooks ha sido usado principalmente para hacer análisis exploratorios y esto generó la necesidad de poder generar gráficos dentro de los notebooks lo cual nos permite no sólo tener una descripción del código con texto plano sino de manera gráfica poder ver los resulrado de los dátos analizados. Adicionalmente se puede hacer uso de tiempo de animaciones para complementar las narrativas.

Los notebooks en cierta forma han ido remplazando el uso de las terminales interactivas, pues enriquecen las descripción del código y facilitan la visualización. Al tener los mismos beneficios que las terminales interactivas y poder hacer pruebas con pequeños fragmentos de código la presentación del código desarrollado resulta de fácil lectura. Adicionalmente se tienen ciertas herramientas que ayudan al desarrollo las cuales han sido pensadas para los usos más comunes. Conocer todas las funcionalidades de los notbooks de jupter queda fuera del alcance de este curso pero veremos ciertas cosas básicas para poder trabajar con ellos.

Usaremos **JupyterLab** el cual integra una plataforma web para generar notebooks, esta interfaz se hace desplegando un servicio web de forma local y puede ser accedida desde cualquier navegador web. Aunque en este curso se utiliza Python también los notebooks de Jupyter-lab puede hacer uso de distintos lenguajes de programación como son R, Julia.

### Instalación JupyterLab

Para su instalación desde un ambiente virtual de conda se ejecuta

(desarrollo) conda install -c conda-forge -y jupyterlab

O con pip

(desarrollo) pip install jupyterlab

### Uso de JupyterLab

Para iniciar una instancia de JupyterLab desde terminal dentro del ambiente virtual que se genero se hace usando el siguiente comando

(desarrollo) jupyter-lab

En windows tambien se puede hacer uso de la interfaz gráfica de conda para lanzar el JupyterLab.

|  |
| --- |
| Figure 1: Inicio de JupyterLab |

Una vez ejecutado el comando se desplegara un servicio web el cual se puede acceder desde un navegador y nos mostrara una pagina web muy similar a la que se muestra en [Figure 1](#fig-inicioJL).

Para empezar un nuevo notebook se elige la version de python que se desea que se ejecute y esta generará un nuevo notebook con nombre ‘Untitled.ipynb’ como en [Figure 2](#fig-nuevo_notebook)

|  |
| --- |
| Figure 2: Nuevo Notebook |

En los notebooks se tienen tres tipos de celda (‘Code’, ‘Markdown’ y ‘Raw’) las cuales se usan con distintos propositos.

* ‘Code’ se utiliza para la ejecución de código
* ‘Markdown’ nos permite hacer uso de el lenguaje Markdown para el enriquecimiente del texto plano.
* ‘Raw’ simplemente va a mostrar el contenido dentro de la celda.

Como ejemplo este curso hace uso de los Notebooks y Markdown, como se ve es de especial útilidad para la creación de texto relacionado con la creación de código y para la generación de manuales técnicos.

### Basicos de Markdown

Markdown es un simple conjunto de instrucciones en texto plano que nos permiten enriquecer el texto para crear documentos más completos. Las siguientes son una muestra de como se utiliza Markdown.

#### Cursiva

La letra cursiva es un recurso que se utiliza en la creación de texto para resaltar alguna palabra en particular para distinguirla del texto usual. Es común utilizarla para palabras en otro idioma como *ballet* o *software*. En el caso de las ciencias es común su uso cuando se hace uso del latín como *curriculum vitae*.

Para escribir con negritas en Markdown se utiliza el caracter \*, todo lo que se entre los asteríscos queda como *cursiva* \*cursiva\*.

#### Negritas

La letras en **negritas** nos sirve para resaltas una palabra o palabras del texto normal. Para que el texto este en **negritas** en Markdown tenemos que colocarlo entre dos asteriscos \*\* \*\*. De esta forma podemos hacer **enfasis** \*\*enfasis\*\* en algo particular.

#### Para escribrir código en el texto.

En ciertos casos es necesario escribir código dentro del texto sin que este sea ejecutado por tal motivo se utiliza la tilde invertida (`) y nos permite escribir cosas sin que estas sean ejecutadas en los distintos lenguajes soportados por Quarto 1+1 `1+1`.

#### Listas

Para hacer listas en Markdown es muy fácil pues esto se puede hacer usando un sólo \* para definirlas, y un espacio para determina la lista.

* Cosa 1
* Cosa 2
* Cossa 3
  + Cosa 3.1
  + Cosa 3.

\* Cosa 1  
\* Cosa 2  
\* Cossa 3  
 \* Cosa 3.1  
 \* Cosa 3.2

#### Secciones

Markdown nos permite tener mantener un orden dentro del documento haciendo uso de #, este funciona para indicar un encabezado e inicio de la sección. Markdown nos permite hacer subsecciones a partir del uso consecutivo de #.

# Sección 1  
  
# Sección 2  
  
# Sección 3  
  
## Sección 3.1  
  
### Sección 3.1.1  
### Sección 3.1.2  
  
## Sección 3.2   
## Sección 3.3  
  
# Seccion 4

### Otras

Markdown nos permite hacer tablas referencia insertar imagenes, etc. Se puede hacer un curso para conocer los distintos usos de forma más extensiva,pero queda fuera del alcance de este curso.

## Instalacion de git (github) (Tema 2.3)

Git es un systema de control de versiones, aunque existen distintos sistemas de control de versiones git es el más utilizado. Los sistemas de control de versiones son una herramienta fundamental en el desarrollo de software pues nos permiten tener un control sobre las versiones del código desarrollado a través de los cambios en el ćodigo. A través del seguimiento de dichos cambios es posible tener un trabajo en equipo sobre el mismo archivo de forma autónoma permitiendo un desarrollo más rápido.

Una de las herramientas más usadas en los sistemas de control de versiones es la idea básica para lo que fueron creados que es **poder regresar a una versión anterior en el desarrollo de software**, lo cual resulta fundamental si hay un error en el desarrollo (bug).

y se recomienda tener una mejor comprensión de los mismos. Por desgracia esto queda fuera del campo de este curso y sólo daremos una explicación simplificada de esto.

**Git** es un sistema de control de versiones . Los sistemas de control de versiones. . De igual forma nos facilita encontrar y corregir los posibles bugs dentro del código.

Existen distintos servicios para almacenar código de manera remota dentro de los cuales podemos destacar [github](https://github.com/) y [gitlab](http://gitlab.com/). En este curso se utiliza github por facilidad pues se usa su servicio de *github-pages* para tener en linea la documentación del paquete que se desarrolla.

Para la instalación de git dentro de un ambiente virtual de anaconda se puede hacer de la siguiente forma

(desarrollo) conda install -c conda-forge git

Lo más seguro es que git ya se encuentre instalado en su sistemas, al menos es el caso para Mac y Linux. Para saber si se tiene instalado git se puede abrir una terminal y ejecutar:

git version

Este comando mostrara la versión instalada dentro del sistema.

En caso de que no se tenga instalado se puede seguir los pasos que se encuentran en [*Github*](https://github.com/git-guides/install-git) o se puede descargar desde <https://git-scm.com/download/win>. Usando cualquiera de los dos metodos su instalación será de forma global, en caso de desear instalarlo dentro de una ambiente virtual se puede hacer con conda o usando el canal oficial de *anaconda*:

conda install -c anaconda git

o el canal de la comunidad *conda-forge*:

conda install -c conda-forge git

### Uso de git

A continuación solo enlistamos ciertos comandos de git pues su uso completo se encuentra fuera del alcance de este curso, en la [sección](../Tema3/Clusters_ejemplo.qmd#otros-comandos-git) se describen un pooco más.

* git clone sirve poara clonar repositorios.
* git pull sirve para traer los cambios desde el servidor de git. El servidor depende de donde se tiene el servicio (github, gitlab o similares).
* git add <archivo> añade el archivo que se modificó y que se desea que actualizar, o bien se añade el archivo si no se tiene dentro de nuesto conjunto de archivos vigilados.
* git commit -a “Comitea” todos los cambios en de los archivos añadidos anteriormente de forma local.
* git push Sube los cambios hechos a el servidor de git, en nuestro caso lo va a subir a github pues el es servicio que determinamos usar.
* git status nos dice los archivos que han sido añadidos cuales se estan vigilando pero sus cambios no han sido añadidos y cuales no se estan vigilando.

## Instalación **nbdev** rápida (linux o Mac) (Tema 2.4)

La instalación de nbdev es igual de fácil que los anteriores paquetes, pero en este caso se utiliza el canal de la empresa que desarrolla *nbdev*. Se instala a través de pip usando:

pip install nbdev

o instalar con conda:

conda install -c fastai nbdev

## Instalación no tan rápida (windows)

La instalación en Windows es distinta pues es necesario hacerla desde PIP y usar una version especifica, al momento de escribir este curso la instalación disponible a través de PIP no funciona de forma totalmente correcta, por tal motivo vamos a especificar la version para hacer uso de esta.

pip install nbdev=2.3.12

## Quarto (Tema 2.5)

Quarto es un sistema que nos permite utilizar las utilidades de los notebook de markdown, en particular usar los notebooks de jupyter, y todas sus utilidades. Quarto esta construido sobre [pandoc](https://pandoc.org/) el cual permite convertir entre distintos tipos de documentos, en particular distintos tipos de markdown, esto permite utilizar las herramientas desarrolladas en quarto y traducir estos los notebooks a otro tipo de documentos.

Con esta idea los desarrolladores de nbdev utilizan Quarto para generar documentos html utilizando y poder generar la documentación necesaria de una biblioteca siguiendo la idea de programacion literal.

## Instalar Quarto

Ya con la instalación de nbdev un comando del mismo nos permite instalar Quarto en el caso de Mac o Linux esto se puede hacer una vez instalado nbdev usando el comando

nbdev\_install\_quarto

### Instalación Quarto windows

Si se desea se puede instalar siguiendo los siguientes [pasos](https://quarto.org/docs/get-started/) para usar con el IDE de preferencia. Esta instalación tendra disponible quarto en todo el sistema y no sólo para el ambiente de desarrollo. Aunque es posible hacer la instalación dentro de un ambiente de desarrollo, esto queda fuera de de una explicación básica.

### Otras cosas

Es recomendable instalar ciertos plug-ins para hacernos la vida más fácil estos se hace ejecutando,

nbdev\_install\_hooks