

Proyectos Conda y Jupyter (Lab y Notebook): Introducción, fundamentos, instalación y ejemplos de uso en el contexto del análisis de datos

Ing. Cristian Samuel Zidosky*

* zidotronic@gmail.com

31 de Mayo de 2019

Índice

Glosario	II
1. Introducción	1
2. Entornos de ejecución y paquetes	1
2.1. Entorno de ejecución	1
2.2. Paquetes de software	1
3. El proyecto Conda	1
3.1. Miniconda: procedimiento de instalación	2
3.1.1. Instalando Miniconda en GNU/Linux	2
3.1.2. Instalando Miniconda en Microsoft Windows	7
3.2. Uso básico	12
3.2.1. Verificar instalación	12
3.2.2. Actualizar paquetes instalados	12
3.2.3. Administración de entornos	13
3.2.4. Administración de paquetes	14
3.2.5. Interfaz gráfica para administrar entornos y paquetes	16
4. El proyecto Jupyter	17
4.1. Arquitectura	17
4.2. Jupyter notebook	17
4.3. Jupyter lab	18
4.4. Instalación	18

Glosario

open-source Código abierto, permite a los programadores acceder a su código fuente y modificarlo; con la restricción de respetar la licencia del código original.

1. Introducción

La presente guía tiene el propósito de dar a conocer los fundamentos necesarios para utilizar las herramientas *open-source* provistas por los proyectos `Conda`¹ y `Jupyter`²; en el contexto del análisis de datos.

Aunque `Conda` originalmente no sea una herramienta destinada al análisis de datos, ésta sirve de base para la instalación y administración de las demás herramientas y entornos de ejecución, entre los cuales se encuentra `Jupyter`. En la actualidad existen algunas distribuciones específicas para su uso en ciencias de datos las cuales incluyen a `Conda`.

En cuanto a `Jupyter`, es un proyecto orientado al desarrollo de software *open-source*, basado en estándares abiertos y servicios orientados al concepto de *computación interactiva*; lo cuál es de mucha utilidad en el dominio de la ciencia de datos.

2. Entornos de ejecución y paquetes

En esta sección se hace una breve descripción sobre los conceptos de *entorno de ejecución* y *paquete de software* (librerías).

2.1. Entorno de ejecución

Para que un programa, desarrollado a partir de un lenguaje de programación de alto nivel (como por ejemplo Python), pueda ejecutarse en un ordenador y acceder a sus recursos; es necesario disponer de un **entorno de ejecución**. Dicho entorno de ejecución es específico para cada versión del lenguaje y además debe ser compatible con el sistema operativo y el hardware subyacente.

Esto significa que, por ejemplo, el entorno de ejecución para Python 2.7 no es el mismo que para Python 3.7; independientemente de que se trate del mismo sistema operativo.

2.2. Paquetes de software

Una forma de distribuir, o compartir, nuestro software con los demás es mediante la creación de un **paquete** o librería. Del mismo modo, podemos instalar en nuestro entorno de ejecución el software desarrollado por un tercero.

En ambos casos existirá una dependencia entre: el paquete de software y el entorno de ejecución utilizado. Aunque también se puede dar que un paquete de software dependa de uno o más paquetes, siendo necesario, en este caso, instalar todas las dependencias antes de instalar el paquete en cuestión.

3. El proyecto Conda

El proyecto `Conda` es un sistema *open-source* de administración de paquetes y de entornos para una gran variedad de lenguajes como ser: *Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN*. Dicho sistema es capaz de ejecutarse en sistemas operativos como: *Windows, macOS y GNU/Linux*.

Mediante `Conda` podemos fácilmente instalar y actualizar los paquetes de software que podamos necesitar junto con sus dependencias, también nos permite disponer, en nuestro ordenador, de más de un entorno de ejecución, y elegir cuál queremos utilizar.

El sistema `Conda` se encuentra incluido en las distribuciones de:

- *Anaconda*: Principal distribución de `Python` para ciencia de datos, la cuál cuenta con alrededor de más de 1000 paquetes destinados para tal propósito, de los cuales la mayoría son instalados por defecto.

¹<https://docs.conda.io/en/latest/>

²<https://jupyter.org/about>

- *Miniconda*: A diferencia de *Anaconda*, sólo incluye una serie de paquetes mínimos los cuales son instalados por defecto, los demás paquetes se deben instalar manualmente mediante el uso de Conda.
- *Anaconda Enterprise*: Es una plataforma comercial para ciencia de datos, la cuál posee una gran e interesante variedad de características.

3.1. Miniconda: procedimiento de instalación

A continuación describiremos el procedimiento de instalación de la distribución Miniconda, por demandar una menor cantidad de almacenamiento; considerándose como un buen punto de partida para aquellos que se estén iniciando en el tema.

Para comenzar debemos acceder al *sitio web oficial*³ de dicha distribución para descargar el instalador compatible con nuestro sistema operativo, teniendo en cuenta la versión de Python que vamos a preferir usar (ver figura 1).

Miniconda

	Windows	Mac OS X	Linux
Python 3.7	64-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer)
	32-bit (exe installer)	64-bit (.pkg installer)	32-bit (bash installer)
Python 2.7	64-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer)
	32-bit (exe installer)	64-bit (.pkg installer)	32-bit (bash installer)

Figura 1: Instaladores disponibles de Miniconda.

Una vez descargado el instalador podemos proceder a ejecutarlo. No es necesario tener instalada la versión de Python que vayamos a preferir, es el propio instalador el que se encarga de descargar y configurar todas las dependencias requeridas; siendo para ello necesario contar con una conexión a Internet en el momento de ejecutar dicho instalador.

En la sección 3.1.1 se describe el procedimiento para la instalación de la distribución Miniconda en un sistema operativo *GNU/Linux*, mientras que en la sección 3.1.2 se muestra el respectivo procedimiento para un sistema operativo *Microsoft Windows*.

3.1.1. Instalando Miniconda en GNU/Linux

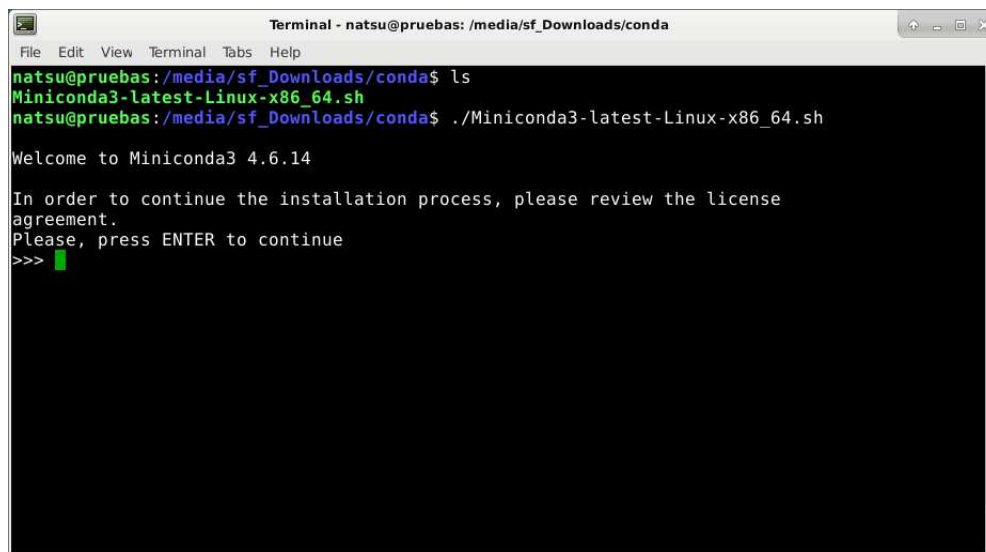
Lo primero que haremos es abrir una *terminal interactiva* del sistema y nos posicionaremos en el directorio donde hemos descargado el instalador (*bash installer*). Una vez hecho esto ejecutaremos el siguiente comando:

```
./Miniconda3-latest-Linux-x86_64.sh
```

Donde en este caso el nombre del archivo de instalación descargado es: *Miniconda3-latest-Linux-x86_64.sh*. Si ha descargado un archivo cuyo nombre es diferente, reemplace dicho nombre en el comando (ver figura 2).

Téngase en cuenta que no es necesario ejecutar el comando de instalación con privilegios de *super usuario* (*root* o *sudo*).

³<https://docs.conda.io/en/latest/miniconda.html>

A terminal window titled "Terminal - natsu@pruebas: /media/sf_Downloads/conda". The prompt is "natsu@pruebas:/media/sf_Downloads/conda\$". The user enters "ls", and the output is "Miniconda3-latest-Linux-x86_64.sh". The user then enters "./Miniconda3-latest-Linux-x86_64.sh". The terminal displays "Welcome to Miniconda3 4.6.14". Below this, it says "In order to continue the installation process, please review the license agreement." and "Please, press ENTER to continue". The prompt ">>>>" is followed by a green cursor.

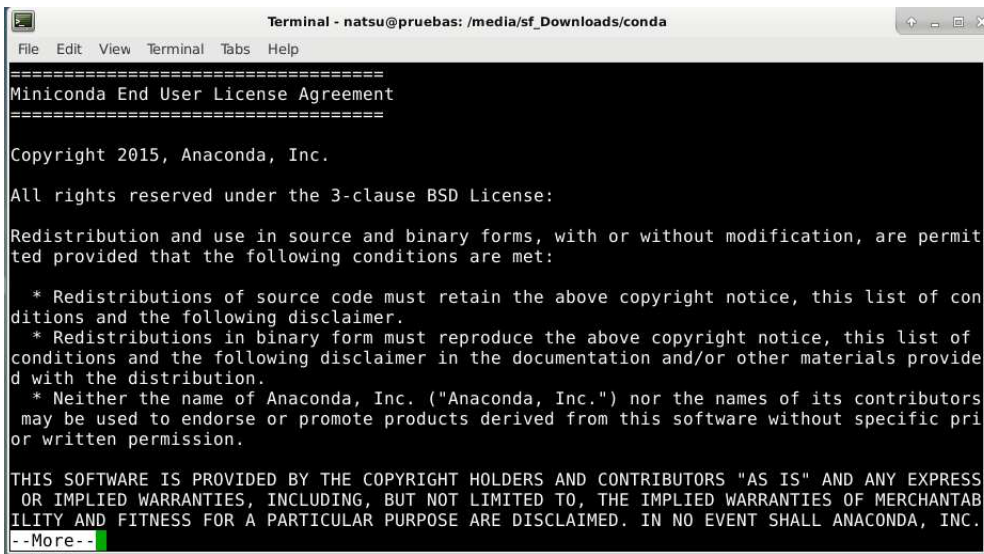
```
Terminal - natsu@pruebas: /media/sf_Downloads/conda
File Edit View Terminal Tabs Help
natsu@pruebas:/media/sf_Downloads/conda$ ls
Miniconda3-latest-Linux-x86_64.sh
natsu@pruebas:/media/sf_Downloads/conda$ ./Miniconda3-latest-Linux-x86_64.sh

Welcome to Miniconda3 4.6.14

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>> █
```

Figura 2: Ejecución del archivo de instalación.

Lo primero que veremos en la *terminal interactiva* será un aviso de que debemos leer el contrato o acuerdo de licencia. Para ello presionamos *enter* (intro) y podremos leer en la pantalla dicha licencia (ver figura 3).



```
Terminal - natsu@pruebas: /media/sf_Downloads/conda
File Edit View Terminal Tabs Help
=====
Miniconda End User License Agreement
=====

Copyright 2015, Anaconda, Inc.

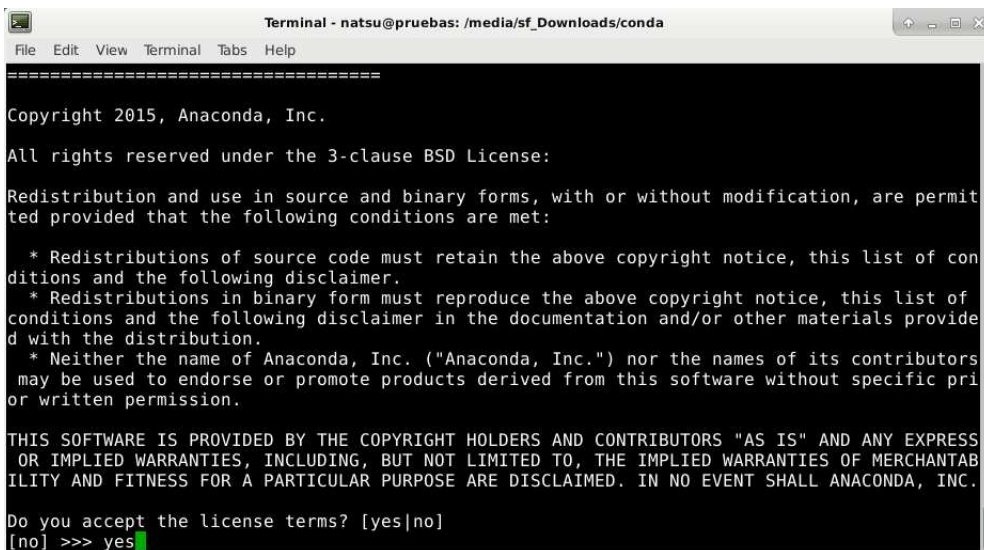
All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Anaconda, Inc. ("Anaconda, Inc.") nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANACONDA, INC.
--More--
```

(a) Contrato o acuerdo de licencia, primera parte.



```
Terminal - natsu@pruebas: /media/sf_Downloads/conda
File Edit View Terminal Tabs Help
=====

Copyright 2015, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Anaconda, Inc. ("Anaconda, Inc.") nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANACONDA, INC.

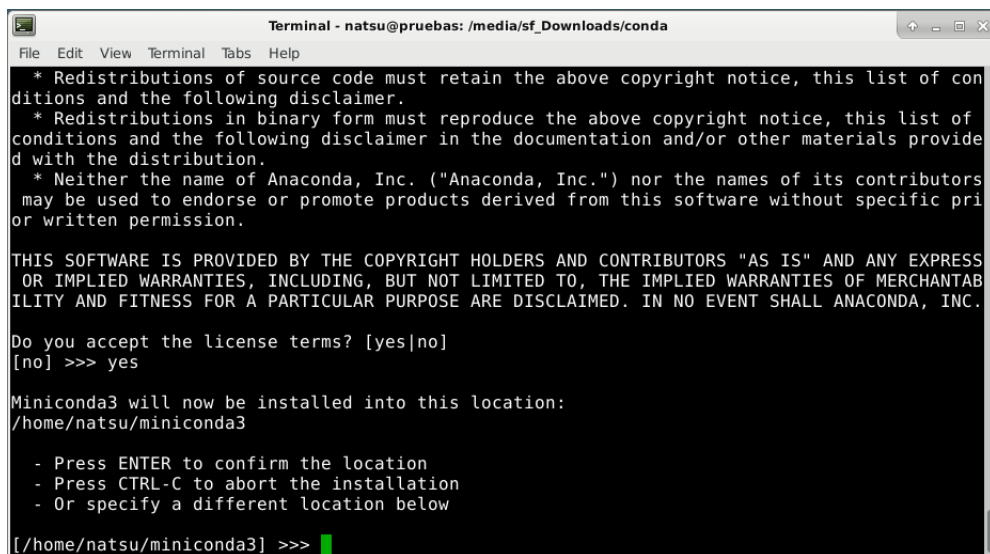
Do you accept the license terms? [yes|no]
[no] >>> yes
```

(b) Contrato o acuerdo de licencia, parte final.

Figura 3: Contrato o acuerdo de licencia de Miniconda.

Para manifestar que aceptamos la licencia debemos introducir en la *terminal interactiva* el texto *yes* y presionar *enter* (ver figura 3b).

Luego de aceptar la licencia, la *terminal interactiva* nos preguntará en qué directorio deseamos instalar Miniconda, en dicha pregunta nos indicará un directorio por defecto, el cual es `/home/nombre-usuario/miniconda3` Donde *nombre-usuario* será el nombre de nuestro usuario del sistema (ver figura 4).



```
Terminal - natsu@pruebas: /media/sf_Downloads/conda
File Edit View Terminal Tabs Help

* Redistributions of source code must retain the above copyright notice, this list of con
ditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provide
d with the distribution.
* Neither the name of Anaconda, Inc. ("Anaconda, Inc.") nor the names of its contributors
may be used to endorse or promote products derived from this software without specific pri
or written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTAB
ILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANACONDA, INC.

Do you accept the license terms? [yes|no]
[no] >>> yes

Miniconda3 will now be installed into this location:
/home/natsu/miniconda3

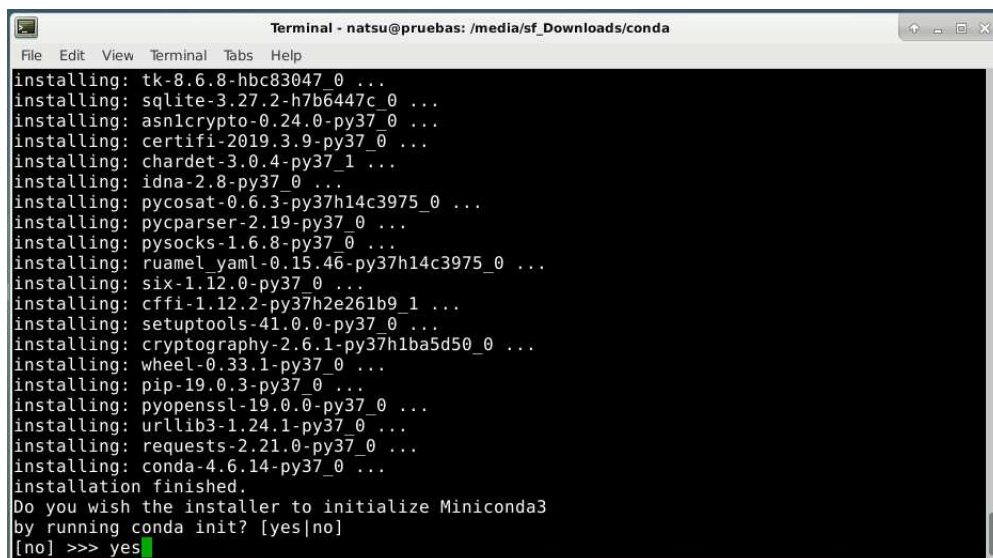
- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/natsu/miniconda3] >>>
```

Figura 4: Confirmación del directorio de instalación.

En el caso de que estemos de acuerdo, con que la instalación se realice en el directorio indicado, simplemente presionamos *enter* y continuamos con la instalación, caso contrario podemos indicar la ruta absoluta al directorio donde deseamos instalar Miniconda.

A medida que el proceso de instalación transcurre, veremos los nombres de los paquetes que se van instalando. Al finalizar dicho proceso, la *terminal interactiva* nos preguntará si queremos correr el procedimiento de inicialización de Conda, ante tal situación respondemos ingresando el texto *yes* y presionamos *enter* (ver figura 5).

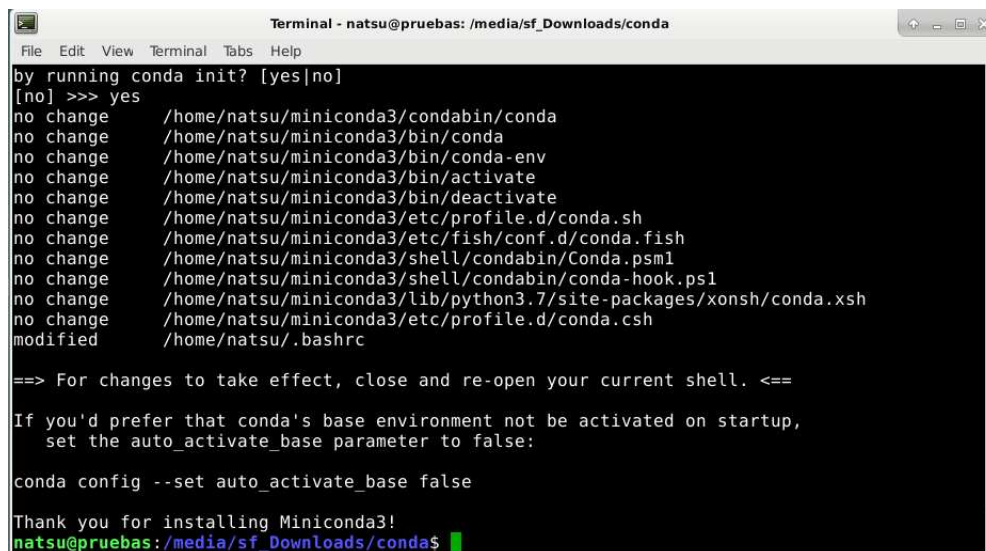


```
Terminal - natsu@pruebas: /media/sf_Downloads/conda
File Edit View Terminal Tabs Help

installing: tk-8.6.8-hbc83047_0 ...
installing: sqlite-3.27.2-h7b6447c_0 ...
installing: asn1crypto-0.24.0-py37_0 ...
installing: certifi-2019.3.9-py37_0 ...
installing: chardet-3.0.4-py37_1 ...
installing: idna-2.8-py37_0 ...
installing: pycosat-0.6.3-py37h14c3975_0 ...
installing: pycparser-2.19-py37_0 ...
installing: pysocks-1.6.8-py37_0 ...
installing: ruamel_yaml-0.15.46-py37h14c3975_0 ...
installing: six-1.12.0-py37_0 ...
installing: cffi-1.12.2-py37h2e261b9_1 ...
installing: setuptools-41.0.0-py37_0 ...
installing: cryptography-2.6.1-py37h1ba5d50_0 ...
installing: wheel-0.33.1-py37_0 ...
installing: pip-19.0.3-py37_0 ...
installing: pyopenssl-19.0.0-py37_0 ...
installing: urllib3-1.24.1-py37_0 ...
installing: requests-2.21.0-py37_0 ...
installing: conda-4.6.14-py37_0 ...
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

Figura 5: Paso final del proceso de instalación.

El procedimiento de instalación finaliza y cómo salida ofrece información básica sobre la instalación (ver figura 6).

A terminal window titled "Terminal - natsu@pruebas: /media/sf_Downloads/conda". The prompt is "by running conda init? [yes|no]". The user enters "yes". The terminal lists files that are either "no change" or "modified". The modified files include "/home/natsu/.bashrc". A message says "==> For changes to take effect, close and re-open your current shell. <==". Another message says "If you'd prefer that conda's base environment not be activated on startup, set the auto_activate_base parameter to false:". The user runs "conda config --set auto_activate_base false". The terminal says "Thank you for installing Miniconda3!". The prompt is "natsu@pruebas:/media/sf_Downloads/conda\$".

```
by running conda init? [yes|no]
[no] >>> yes
no change      /home/natsu/miniconda3/condabin/conda
no change      /home/natsu/miniconda3/bin/conda
no change      /home/natsu/miniconda3/bin/conda-env
no change      /home/natsu/miniconda3/bin/activate
no change      /home/natsu/miniconda3/bin/deactivate
no change      /home/natsu/miniconda3/etc/profile.d/conda.sh
no change      /home/natsu/miniconda3/etc/fish/conf.d/conda.fish
no change      /home/natsu/miniconda3/shell/condabin/Conda.ps1
no change      /home/natsu/miniconda3/shell/condabin/conda-hook.ps1
no change      /home/natsu/miniconda3/lib/python3.7/site-packages/xonsh/conda.xsh
no change      /home/natsu/miniconda3/etc/profile.d/conda.csh
modified       /home/natsu/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

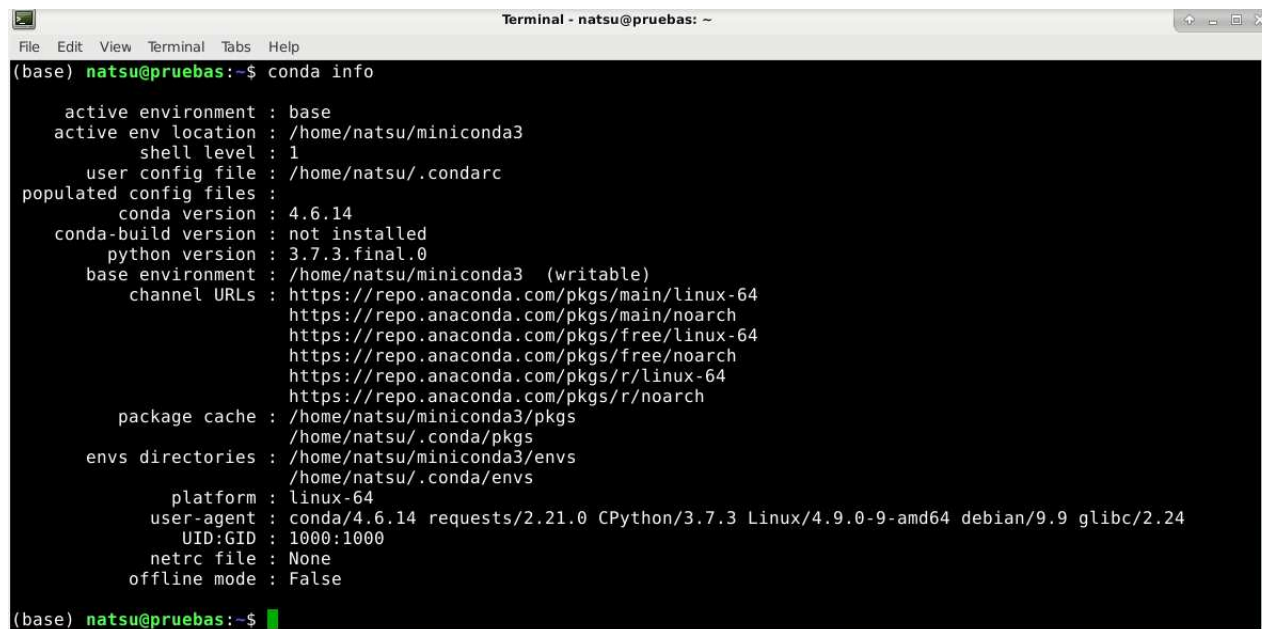
Thank you for installing Miniconda3!
natsu@pruebas:/media/sf_Downloads/conda$
```

Figura 6: Finalización de la instalación.

Para verificar que nuestra distribución de Miniconda se encuentra instalada y configurada correctamente, podemos ejecutar el siguiente comando en la *terminal interactiva*:

```
conda info
```

Donde nos mostrará toda la información referente a la instalación (ver figura 7).

A terminal window titled "Terminal - natsu@pruebas: ~". The prompt is "(base) natsu@pruebas:~\$". The user enters "conda info". The terminal displays detailed information about the conda environment, including the active environment, location, version, and various URLs.

```
(base) natsu@pruebas:~$ conda info

active environment : base
active env location : /home/natsu/miniconda3
shell level : 1
user config file : /home/natsu/.condarc
populated config files :
  conda version : 4.6.14
  conda-build version : not installed
  python version : 3.7.3.final.0
base environment : /home/natsu/miniconda3 (writable)
channel URLs : https://repo.anaconda.com/pkgs/main/linux-64
               https://repo.anaconda.com/pkgs/main/noarch
               https://repo.anaconda.com/pkgs/free/linux-64
               https://repo.anaconda.com/pkgs/free/noarch
               https://repo.anaconda.com/pkgs/r/linux-64
               https://repo.anaconda.com/pkgs/r/noarch
package cache : /home/natsu/miniconda3/pkgs
                /home/natsu/.conda/pkgs
envs directories : /home/natsu/miniconda3/envs
                  /home/natsu/.conda/envs
platform : linux-64
user-agent : conda/4.6.14 requests/2.21.0 CPython/3.7.3 Linux/4.9.0-9-amd64 debian/9.9 glibc/2.24
UID:GID : 1000:1000
netrc file : None
offline mode : False

(base) natsu@pruebas:~$
```

Figura 7: Verificación de la instalación.

Una vez alcanzado éste punto podemos dar por finalizada la instalación de Miniconda en *GNU/Linux*.

3.1.2. Instalando Miniconda en Microsoft Windows

Para instalar Miniconda en *Microsoft Windows*, simplemente hacemos *doble clic* sobre el respectivo archivo de instalación, donde luego de unos instantes aparecerá el *asistente de instalación* (ver figura 8).

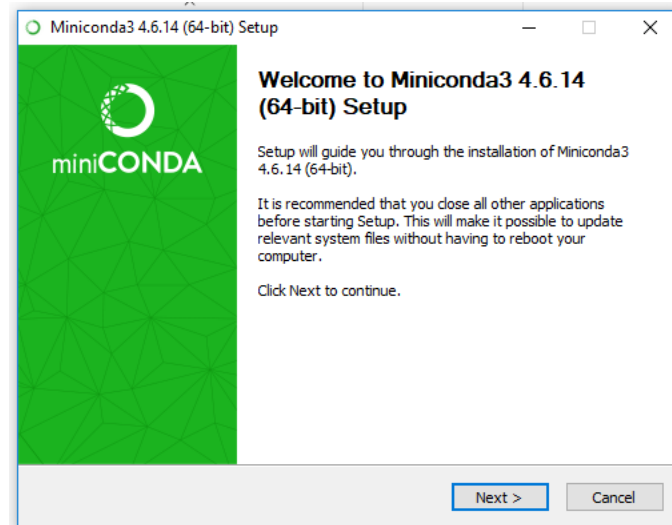


Figura 8: Asistente de instalación inicial de Miniconda para *Microsoft Windows*.

Luego de hacer *clic* sobre el botón *Next* (siguiente), nos mostrará el contrato o acuerdo de licencia (ver figura 9); el cuál aceptaremos haciendo *clic* sobre el botón *I Agree* (yo estoy de acuerdo).

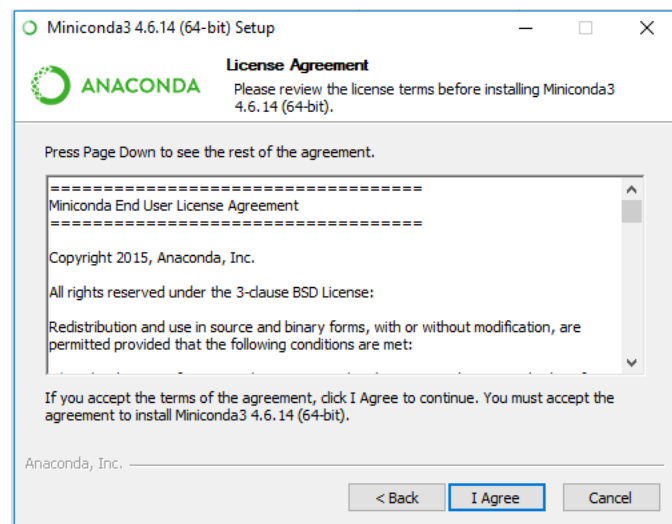


Figura 9: Contrato o acuerdo de licencia, instalación en Microsoft Windows.

Luego de ello el *asistente de instalación* nos preguntará si deseamos instalar Miniconda para todos los usuarios del sistema o sólo para el nuestro (ver figura 10). Ante esta situación, la opción más recomendable es indicar que la instalación sea exclusiva para nuestro usuario y luego de ello hacemos clic sobre el botón *Next* (siguiente).

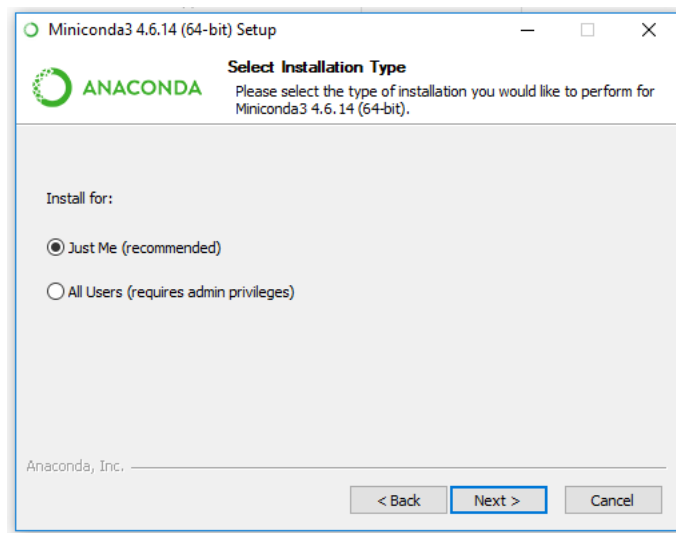


Figura 10: Selección del tipo de instalación.

El siguiente paso es indicar el directorio de instalación, donde el *asistente de instalación* nos sugiere un directorio por defecto (ver figura 11), podemos optar por utilizar dicho directorio o indicar uno de nuestra preferencia. Luego de ello, nuevamente, haremos *clik* sobre el botón *Next* (siguiente).

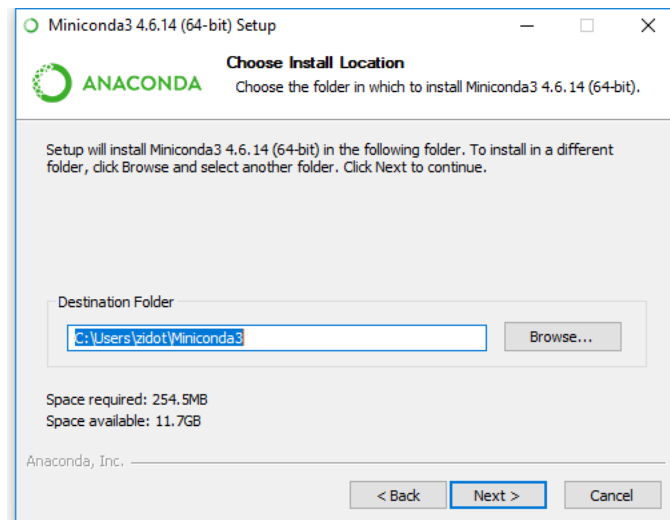


Figura 11: Selección del directorio de instalación.

Una de las últimas opciones que debemos indicar, mediante el *asistente de instalación*, es si queremos que la distribución se integre como configuración por defecto para la versión de Python del sistema (ver figura 12).

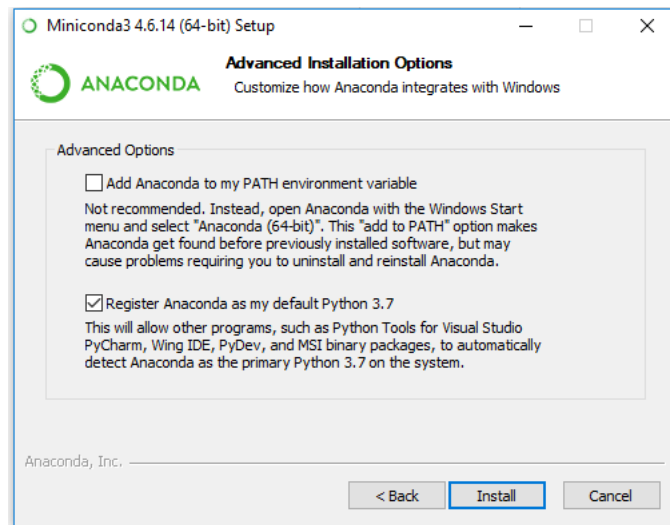


Figura 12: Opción avanzada de instalación.

Luego de indicar esta última opción, comenzará el proceso de instalación, donde podremos ir viendo su progreso (ver figura 13).

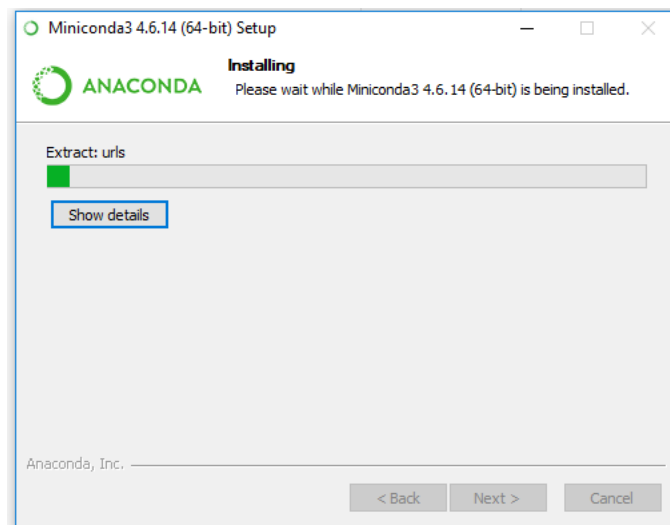


Figura 13: Indicador del progreso de la instalación.

Una vez que el proceso de instalación finaliza, el asistente de instalación nos permite pasar al paso final (ver figura 14), para ello debemos hacer *clic* sobre el botón *Next* (siguiente).

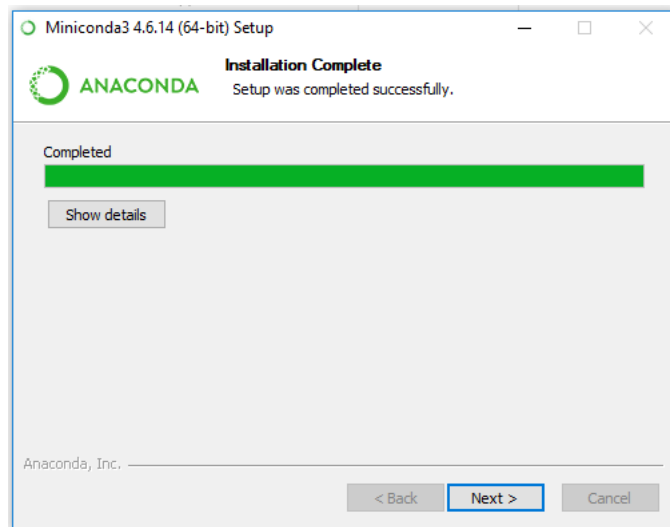


Figura 14: Confirmación de la instalación.

De ese modo se llega al paso final del proceso de instalación, donde el *asistente de instalación* nos ofrecerá la opción de acceder, mediante el navegador web, a información que nos permitirá aprender más sobre la distribución instalada (ver figura 15). Elegimos la opción que deseamos y presionamos *Finish* (finalizar).



Figura 15: Finalización de la instalación.

Una vez finalizada la instalación podremos hacer uso de nuestra distribución de Miniconda, para ello dispondremos de los enlaces correspondientes en el *menú de inicio* (ver figura 16).

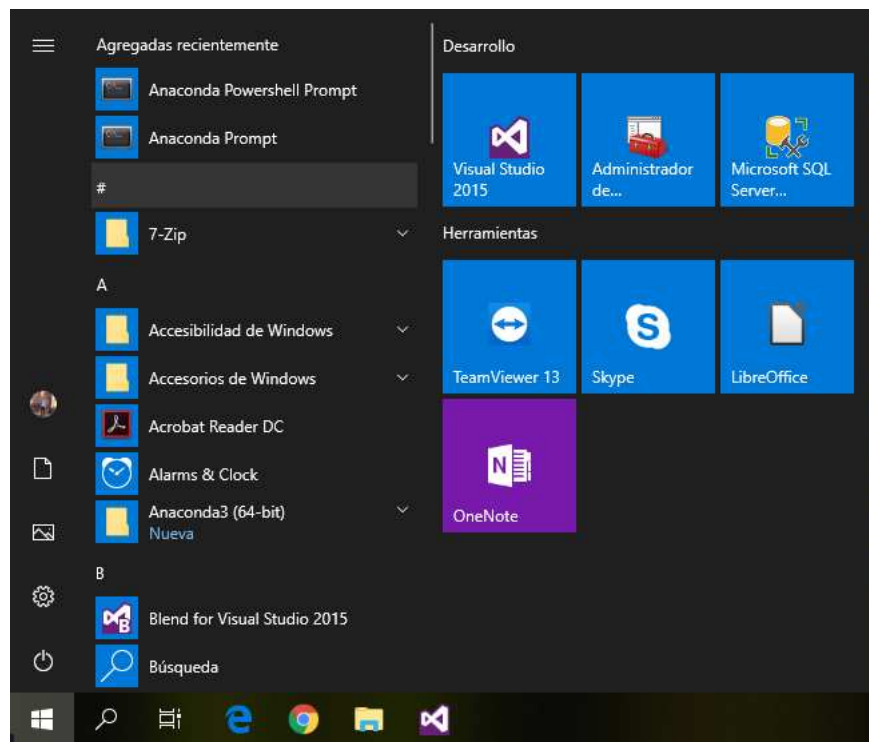
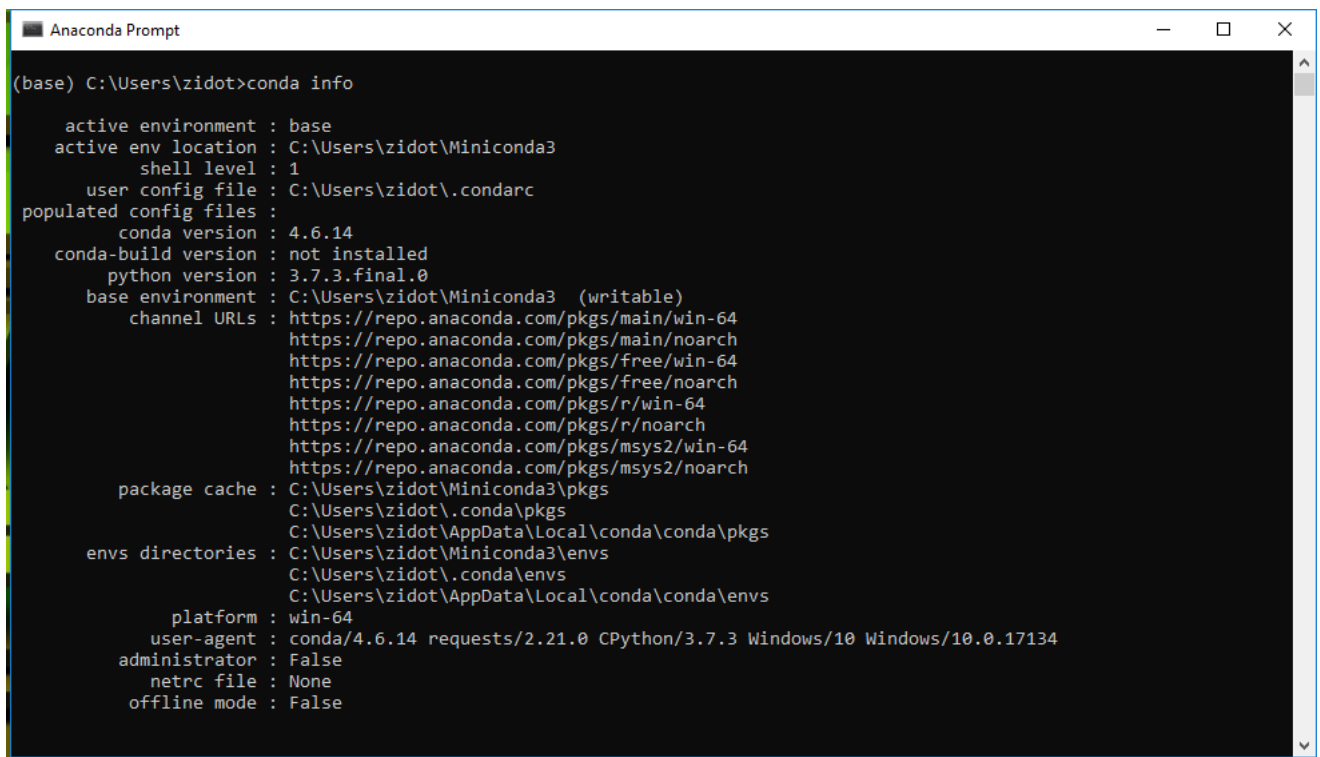


Figura 16: Enlace de las aplicaciones en el *menú de inicio*.

Para ello haremos *clic* sobre el enlace *Anaconda Prompt*, donde se abrirá una *terminal interactiva* en la cuál, entre otras cosas, podremos consultar la información de la instalación (ver figura 17).



```
(base) C:\Users\zidot>conda info

active environment : base
active env location : C:\Users\zidot\Miniconda3
shell level : 1
user config file : C:\Users\zidot\.condarc
populated config files :
  conda version : 4.6.14
  conda-build version : not installed
  python version : 3.7.3.final.0
  base environment : C:\Users\zidot\Miniconda3 (writable)
    channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                  https://repo.anaconda.com/pkgs/main/noarch
                  https://repo.anaconda.com/pkgs/free/win-64
                  https://repo.anaconda.com/pkgs/free/noarch
                  https://repo.anaconda.com/pkgs/r/win-64
                  https://repo.anaconda.com/pkgs/r/noarch
                  https://repo.anaconda.com/pkgs/msys2/win-64
                  https://repo.anaconda.com/pkgs/msys2/noarch
  package cache : C:\Users\zidot\Miniconda3\pkgs
                  C:\Users\zidot\.conda\pkgs
                  C:\Users\zidot\AppData\Local\conda\conda\pkgs
  envs directories : C:\Users\zidot\Miniconda3\envs
                  C:\Users\zidot\.conda\envs
                  C:\Users\zidot\AppData\Local\conda\conda\envs
  platform : win-64
  user-agent : conda/4.6.14 requests/2.21.0 CPython/3.7.3 Windows/10 Windows/10.0.17134
  administrator : False
  netrc file : None
  offline mode : False
```

Figura 17: Terminal interactiva de Miniconda para *Microsoft Windows*.

3.2. Uso básico

A continuación se describen algunos comandos que podemos ejecutar, independientemente del sistema operativo que usemos, desde la *terminal interactiva* mediante Conda. Dichos comandos permiten:

- Verificar y obtener información sobre la instalación (ver sección 3.2.1).
- Actualizar los paquetes (ver sección 3.2.2).
- Crear, activar, desactivar y eliminar entornos (ver sección 3.2.3).
- Buscar, instalar, y eliminar paquetes de forma independiente (ver sección 3.2.4).

Para mayor información sobre los comandos disponibles y su utilización, consultar el material que se adjunta como anexo de la presente guía (*CONDA 4.6 CHEAT SHEET*).

3.2.1. Verificar instalación

Este comando ya lo hemos utilizado anteriormente para verificar que nuestra instalación de Miniconda a sido satisfactoria (ver sección 3.1). Simplemente debemos ejecutar:

```
conda info
```

3.2.2. Actualizar paquetes instalados

Para actualizar todos los paquetes instalados debemos ejecutar:

```
conda update conda
```

Donde se verificará cada uno de los paquetes instalados en nuestro entorno y recibiremos una notificación si es necesario actualizar alguno de ellos.

Téngase en cuenta que en el material que se adjunta como anexo, el comando para tal propósito es:

```
conda update anaconda
```

Esto es válido para aquellos casos donde se tenga instalado **Anaconda** en lugar de **Miniconda**.

3.2.3. Administración de entornos

Comenzaremos creando un nuevo entorno, suponiendo que deseamos que dicho entorno sea para Python 2.7

```
conda create --name Entorno_2.7 python=2.7
```

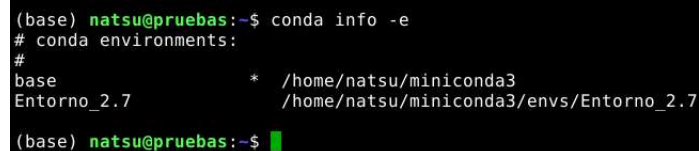
El ejecutar este comando la *terminal interactiva* nos informará de los paquetes que necesita instalar y nos pedirá que confirmemos la operación, para aceptar respondemos ingresando el texto *y* y presionamos *enter*.

Con esto creamos un entorno con el nombre *Entorno_2.7* el cual nos permite ejecutar aplicaciones en Python 2.7

Podemos verificar que se ha creado el nuevo entorno mediante el siguiente comando:

```
conda info -e
```

En la figura 18 se puede observar el resultado que obtendríamos.



```
(base) natsu@pruebas:~$ conda info -e
# conda environments:
#
base                * /home/natsu/miniconda3
Entorno_2.7         /home/natsu/miniconda3/envs/Entorno_2.7
(base) natsu@pruebas:~$
```

Figura 18: Lista de entornos disponibles.

En la dicha figura, el * indica el entorno que se encuentra actualmente en uso (activo). Para poder utilizar nuestro nuevo entorno debemos activarlo mediante el siguiente comando:

```
conda activate Entorno_2.7
```

En la figura 19 se puede observar el resultado obtenido, donde además se ejecuta el comando de información.



```
(base) natsu@pruebas:~$ conda activate Entorno_2.7
(Entorno_2.7) natsu@pruebas:~$ conda info -e
# conda environments:
#
base                /home/natsu/miniconda3
Entorno_2.7         * /home/natsu/miniconda3/envs/Entorno_2.7
(Entorno_2.7) natsu@pruebas:~$
```

Figura 19: Cambio de entorno.

Para cambiar a otro entorno debemos ejecutar el comando:

```
conda activate nombre_entorno
```


Donde `nombre_entorno` es el nombre del entorno al cuál deseamos cambiar, por ejemplo podríamos indicar el nombre `base` para volver al entorno inicial; el cual ha sido creado al momento de instalar Miniconda.

Por último nos queda ver cómo eliminar un entorno, ello se hace mediante el comando:

```
conda remove --name nombre_entorno --all
```

El ejecutar este comando, la terminal interactiva nos listará todos los paquetes que pertenecen al entorno en cuestión, los cuales serán eliminados (ver figura 20).

```
(base) natsu@pruebas:~$ conda remove --name Entorno_2.7 --all
Remove all packages in environment /home/natsu/miniconda3/envs/Entorno_2.7:

## Package Plan ##

  environment location: /home/natsu/miniconda3/envs/Entorno_2.7

The following packages will be REMOVED:

ca-certificates-2019.1.23-0
certifi-2019.3.9-py27_0
libedit-3.1.20181209-hc058e9b_0
libffi-3.2.1-hd88cf55_4
libgcc-ng-8.2.0-hdf63c60_1
libstdcxx-ng-8.2.0-hdf63c60_1
ncurses-6.1-he6710b0_1
openssl-1.1.1b-h7b6447c_1
pip-19.1.1-py27_0
python-2.7.16-h9bab390_0
readline-7.0-h7b6447c_5
setuptools-41.0.1-py27_0
sqlite-3.28.0-h7b6447c_0
tk-8.6.8-hbc83047_0
wheel-0.33.4-py27_0
zlib-1.2.11-h7b6447c_3

Proceed ([y]/n)? █
```

Figura 20: Eliminar entorno.

3.2.4. Administración de paquetes

En cuanto a la administración de paquetes, comenzaremos por el comando que nos permite listar todos los paquetes instalados en un determinado entorno:

```
conda list
```

El comando anterior lista todos los paquetes instalados en el entorno actual (ver sección 3.2.3).

Si quisiéramos listar los paquetes instalados en un determinado entorno, debemos indicar su nombre mediante `nombre_entorno`:

```
conda list --name nombre_entorno
```

Para buscar un paquete y verificar si lo podemos instalar, debemos ejecutar:

```
conda search nombre_del_paquete
```

Por ejemplo en la figura 21 se puede ver el resultado de realizar una búsqueda asociada al nombre *jupyter*.

```
(base) natsu@pruebas:~$ conda search jupyter
Loading channels: done
# Name Version Build Channel
jupyter 1.0.0 py27_0 pkgs/free
jupyter 1.0.0 py27_1 pkgs/free
jupyter 1.0.0 py27_2 pkgs/free
jupyter 1.0.0 py27_3 pkgs/free
jupyter 1.0.0 py27_4 pkgs/main
jupyter 1.0.0 py27_6 pkgs/main
jupyter 1.0.0 py27_7 pkgs/main
jupyter 1.0.0 py27h505fd4b_0 pkgs/main
```

Figura 21: Resultado de la búsqueda de un paquete.

Otra alternativa, para consultar información sobre los distintos paquetes disponibles, es accediendo al sitio web de *Anaconda Cloud*⁴ y utilizar la barra de búsqueda para indicar el paquete que nos interese (ver figura 22a) donde obtendremos mayor información y podremos acceder al comando de instalación que se debe utilizar desde Conda (ver figura 22b).

The screenshot shows the Anaconda Cloud website with a search bar containing 'jupyter'. Below the search bar, there are filters for Type, Access, and Platform. The search results are displayed in a table with columns for Favorites, Downloads, Package (owner / package), and Platforms. Two results are shown: 'conda-forge / jupyter_core 4.4.0' and 'conda-forge / jupyter_client 5.2.4'.

Favorites	Downloads	Package (owner / package)	Platforms
10	1905605	conda-forge / jupyter_core 4.4.0 Core common functionality of Jupyter projects.	linux-64 noarch osx-64 win-32 win-64
3	1557849	conda-forge / jupyter_client 5.2.4 jupyter_client contains the reference implementation of the Jupyter protocol.	linux-64 noarch osx-64 win-32 win-64

(a) Resultado de la búsqueda.

The screenshot shows the 'Installers' page for the 'jupyter_core' package. It includes an info box stating 'This package contains files in non-standard labels.' Below this, there is a section for 'conda install' with a question mark icon. It lists various platform-specific installers for version 4.3.0 and 4.4.0. At the bottom, there are three commands to install the package using conda.

Info: This package contains files in non-standard labels.

conda install ?

- linux-64 v4.3.0
- win-32 v4.3.0
- noarch v4.4.0
- osx-64 v4.3.0
- win-64 v4.3.0

To install this package with conda run one of the following:

```
conda install -c conda-forge jupyter_core
conda install -c conda-forge/label/gcc7 jupyter_core
conda install -c conda-forge/label/cf201901 jupyter_core
```

(b) Instrucciones de instalación.

Figura 22: Uso del sitio web *Anaconda Cloud* para consultar información sobre un paquete.

⁴<https://anaconda.org/>

Para instalar un paquete simplemente ejecutamos el comando:

```
conda install nombre_paquete
```

Donde `nombre_paquete` es el nombre del paquete que se desea instalar (ver el ejemplo de la figura 22b).

Para desinstalar un paquete debemos utilizar el comando:

```
conda uninstall nombre_paquete
```

Donde nuevamente `nombre_paquete` es el nombre del paquete a desinstalar.

3.2.5. Interfaz gráfica para administrar entornos y paquetes

Para finalizar con este tema, vamos a ver como instalar el paquete *anaconda-navigator*, el cuál nos permite administrar los entornos y paquetes mediante una interfaz gráfica.

Para ello ejecutamos el siguiente comando:

```
conda install anaconda-navigator
```

La *terminal interactiva* nos informará de los paquetes que se instalarán y nos pedirá la confirmación de dicha operación.

Una vez finalizada la instalación, ejecutamos el siguiente comando desde la *terminal interactiva*:

```
anaconda-navigator
```

Luego de unos instantes aparecerá una interfaz gráfica que nos permitirá realizar las mismas tareas indicadas anteriormente pero por medio de una herramienta gráfica (ver figura 23).

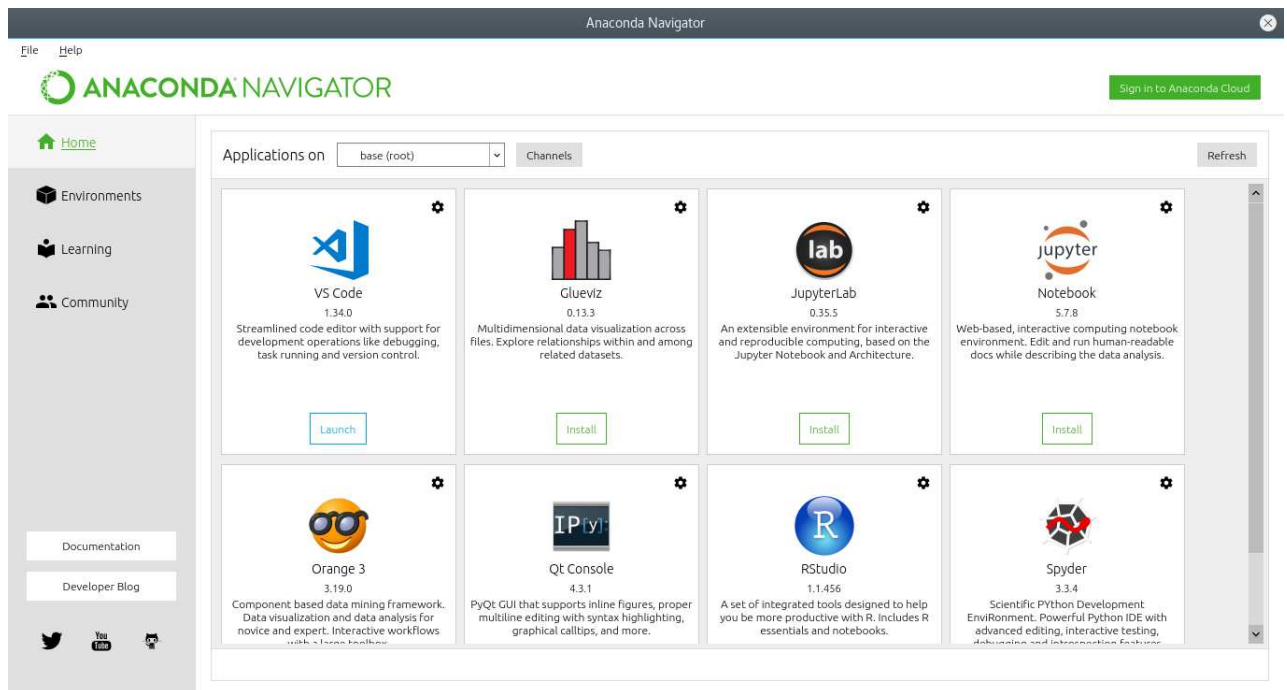


Figura 23: Interfaz gráfica de *Anaconda navigator*.

4. El proyecto Jupyter

Jupyter es un proyecto *open-source* no comercial, que ha surgido en el año 2014 a partir del proyecto IPython⁵. Con el tiempo ha ido evolucionando para dar soporte al concepto de *ciencia de datos interactiva* y *computación científica* para una gran variedad de lenguajes de programación (“todos”), y se distribuye bajo la licencia *BSD modificada*⁶.

Jupyter provee un potente conjunto de herramientas para trabajar con grandes volúmenes de datos, visualizarlos, y compartir los resultados obtenidos. La adopción de su utilización es tan grande que se está convirtiendo en un estándar para los científicos de datos.

4.1. Arquitectura

A continuación se realiza una breve descripción de cómo funciona Jupyter. Para comenzar es un sistema basado en *kernel*, donde cada *kernel* es un motor de ejecución de un determinado lenguaje de programación, o plataforma que se ejecuta en un servidor.

Mediante el sistema distribuido de mensajes ZeroMQ⁷ se envían las instrucciones que se deben ejecutar en cada *kernel*, como así también se reciben sus respectivos resultados.

Por último, mediante el protocolo de comunicación *WebSocket* se permite la interacción de un cliente (navegador web) con el servidor (ver figura 24).

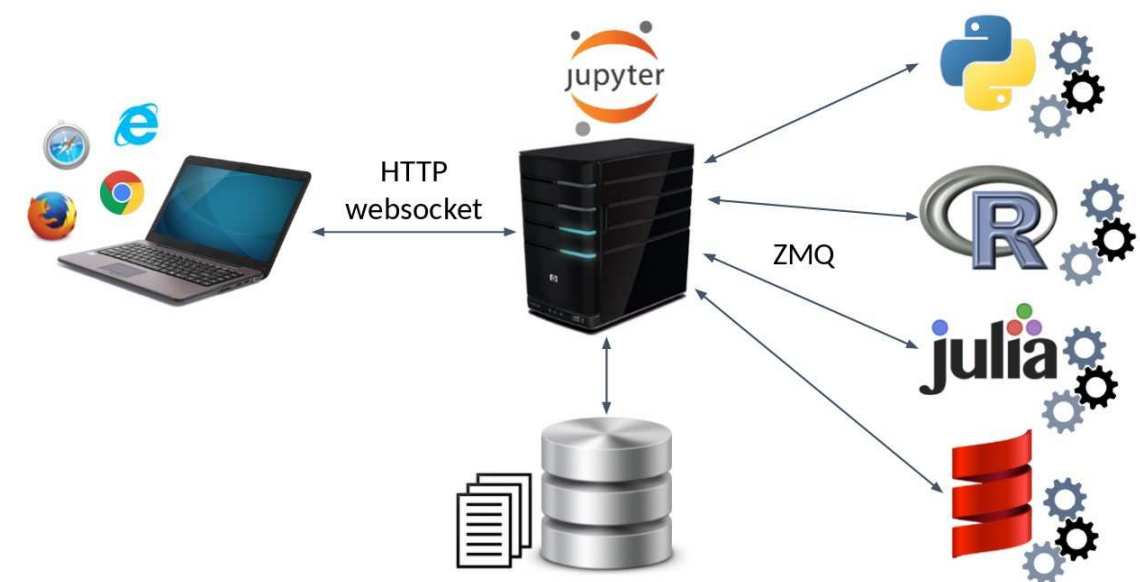


Figura 24: Arquitectura de Jupyter.

4.2. Jupyter notebook

Es una aplicación *cliente-servidor* la cuál permite compartir documentos web en formato JSON. Dichos documentos contienen una serie de celdas, de entrada y de salida, las cuales pueden almacenar: código, texto, fórmulas matemáticas y ecuaciones, etc.

Estos documentos se pueden exportar a otros formatos y compartirse con otros usuarios.

⁵<https://ipython.org/>

⁶<https://opensource.org/licenses/BSD-3-Clause>

⁷<http://zeromq.org/>

Es importante aclarar que **Jupyter notebook** no es una aplicación multiusuario, es decir que únicamente se puede interactuar con una sola instancia. Sin embargo, es posible interconectar varias instancias mediante un servidor con *proxy multiusuario*, tal es el caso del proyecto **Jupyter Hub**.

Jupyter Hub permite gestionar el acceso compartido a cada uno de los documentos.

4.3. Jupyter lab



Es el sucesor de **Jupyter notebook**, ofrece más opciones de personalización e interacción. Es más sencillo de implementar extensiones. Ofrece editores, terminales y otros componentes con mayores mejoras. Posee un nuevo diseño de interfaz de usuario e incluye accesos directos a diversos servicios en la nube.

4.4. Instalación

En esta sección veremos la instalación de **Jupyter lab** en lugar de **Jupyter notebook**; ya que esta última será reemplazada por la primera.

Obviamente que dicha instalación se realizará mediante el sistema **Conda**, siguiendo los pasos que ya hemos visto con anterioridad (ver sección 3.2.4).

Comenzaremos buscando el paquete **jupyter lab** desde **Anaconda cloud** (ver figura 25).

jupyter lab				Q
Filters Type: All ▾ Access: All ▾ Platform: All ▾				
⬆ Favorites	▾ Downloads	⬆ Package (owner / package)	Platforms	
16	1113503	 conda-forge / jupyterlab 0.35.6 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	conda	linux-64 osx-64 win-32 win-64
1	423922	 conda-forge / jupyterlab_launcher 0.13.1 A Launcher for JupyterLab based applications.	conda	linux-64 noarch osx-64 win-32 win-64

(a) Búsqueda de Jupyter lab.

Installers

Info: This package contains files in non-standard [labels](#).

conda install ?

linux-64 v0.35.6

win-32 v0.32.0

osx-64 v0.35.6

win-64 v0.35.6

To install this package with conda run one of the following:

```
conda install -c conda-forge jupyterlab
```

```
conda install -c conda-forge/label/broken jupyterlab
```

```
conda install -c conda-forge/label/cf201901 jupyterlab
```

(b) Comando de instalación.

Figura 25: Instalación de Jupyter lab.

Ahora lo único que nos hace falta es ejecutar, en la *terminal interactiva*, uno de los comandos de la figura 25b, utilizaremos el primero:

```
conda install -c conda-forge jupyterlab
```

Luego de ejecutar el comando anterior, la *terminal interactiva*, nos pedirá que confirmemos la operación, una vez confirmada dicha operación comenzará la instalación.

Cuando la instalación finaliza, podremos ejecutar nuestro Jupyter lab ingresando el siguiente comando:

```
jupyter-lab
```

En la figura 26 podemos ver a Jupyter lab ejecutándose en nuestro navegador.

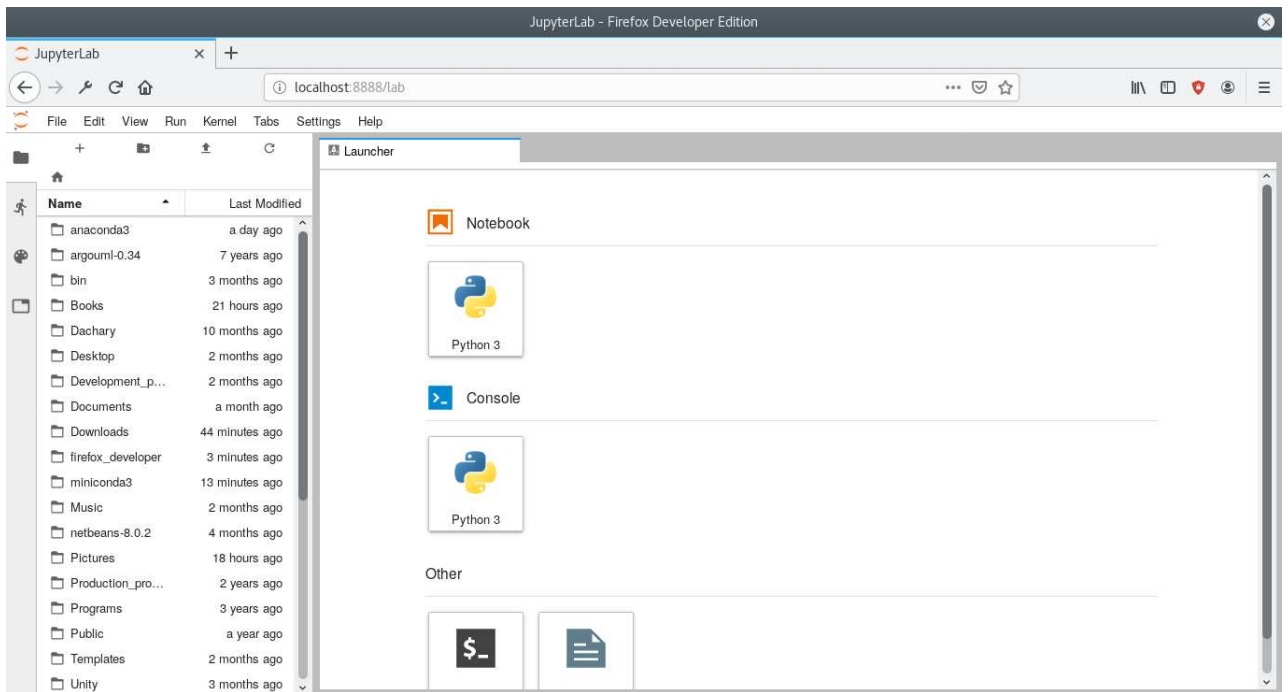


Figura 26: Jupyter lab ejecutándose en el navegador.