

Descargar Anaconda + Iniciando con Python

Iniciando con Python

- Se va a instalar Anaconda con es una interfaz que facilita el trabajo en Python.
- Python es un lenguaje de programación que se usa en múltiples aplicaciones.
- Python también puede ser instalado local (en el computador, fuera de Anaconda).






- Fácil de usar
- Rápido y eficiente
- Gran comunidad online
- Ampla cantidad de **librerías** específicas (¡pronto veremos qué son!)

Descargar exe

Anaconda | Individual Edition

anaconda.com/products/individual#Downloads

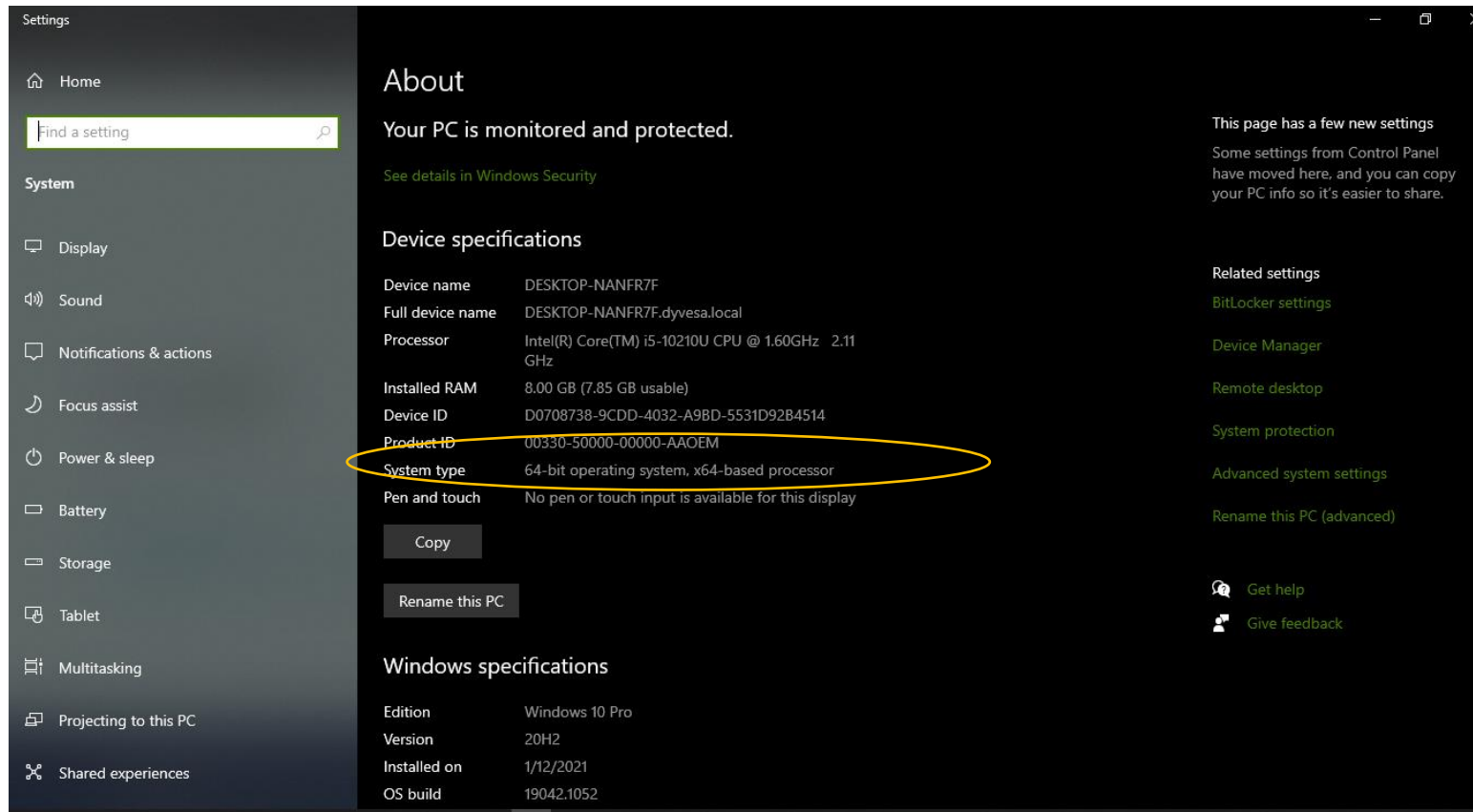
Anaconda Installers

Windows 	MacOS 	Linux 
Python 3.8	Python 3.8	Python 3.8
64-Bit Graphical Installer (477 MB)	64-Bit Graphical Installer (440 MB)	64-Bit (x86) Installer (544 MB)
32-Bit Graphical Installer (409 MB)	64-Bit Command Line Installer (433 MB)	64-Bit (Power8 and Power9) Installer (285 MB)
		64-Bit (AWS Graviton2 / ARM64) Installer (413 M)

This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#) [Accept](#)

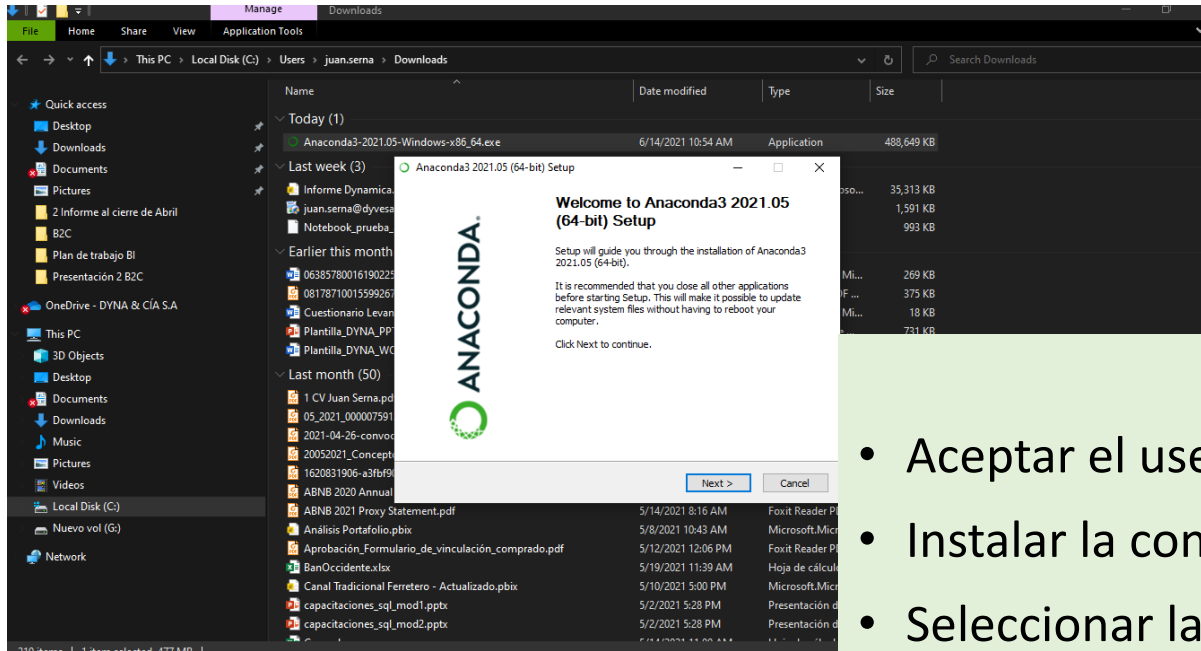
<https://www.anaconda.com/products/individual#Downloads>

Ver especificaciones del pc



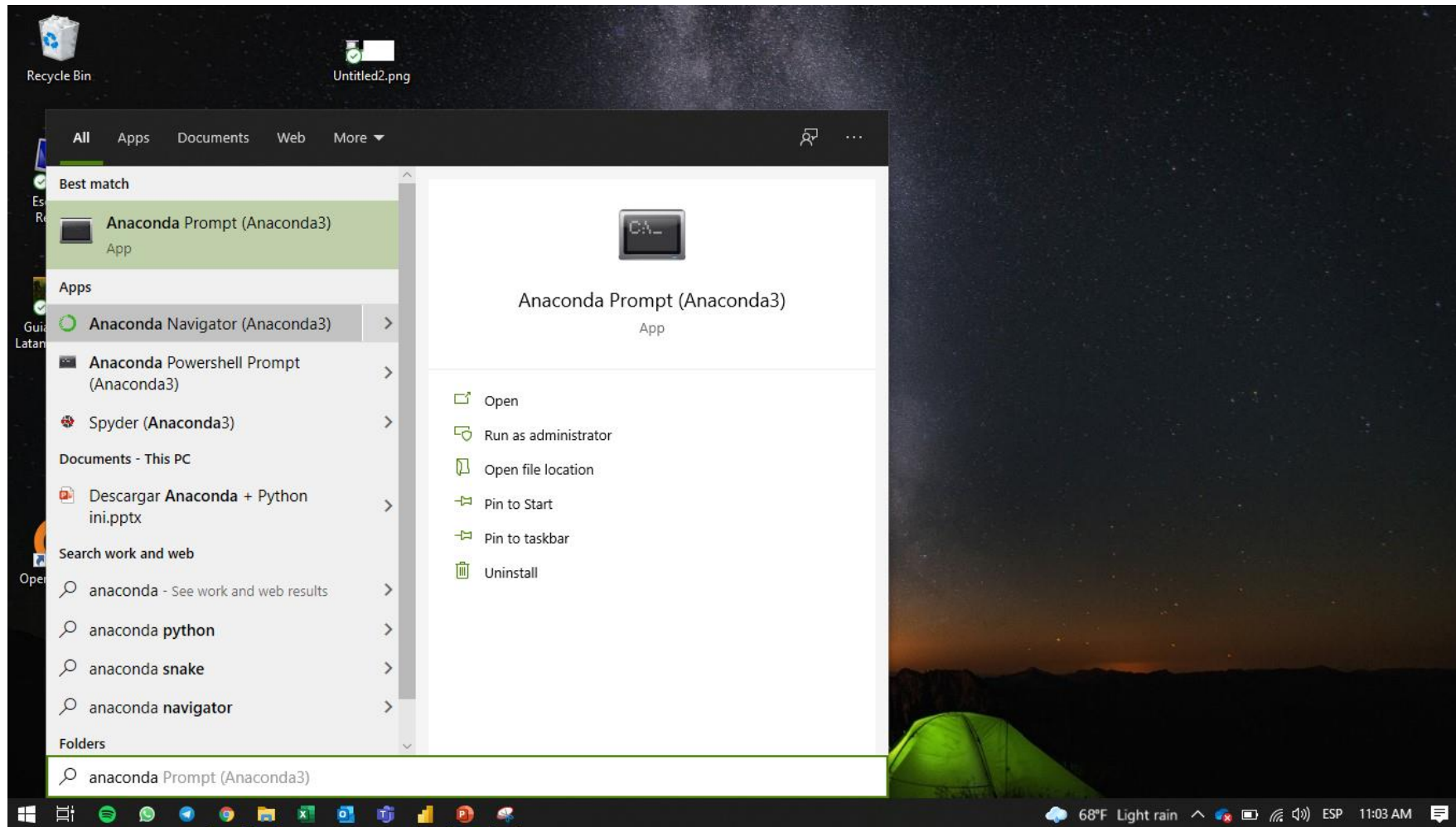
Windows → Click Derecho en botón Inicio
Seleccionar Sistema

Iniciar el .exe



- Aceptar el user agreement
- Instalar la configuración recomendada
- Seleccionar la ruta en disco (se puede aceptar la que sale por defecto)
- Registrar Anaconda como ejecutor para Python 3.8 (o la versión que sea)
- Instalar


Ejecutar Anaconda Navigator



Este es el home

Anaconda Navigator

File Help

 ANACONDA.NAVIGATOR

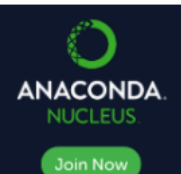
Sign in

Home

Environments

Learning




Community

 ANACONDA NUCLEUS
Join Now









Discover premium data science content

Documentation

Anaconda Blog

Applications on Channels

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated <input type="button" value="Launch"/>	 Datalore Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team. <input type="button" value="Launch"/>	 IBM Watson Studio Cloud IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling. <input type="button" value="Launch"/>	 JupyterLab 3.0.14 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. <input type="button" value="Launch"/>
 Jupyter Notebook 6.3.0 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis <input type="button" value="Launch"/>	 Powershell Prompt 0.0.1 Run a Powershell terminal with your current environment from Navigator activated <input type="button" value="Launch"/>	 Qt Console 5.0.3 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. <input type="button" value="Launch"/>	 Spyder 4.2.5 Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features <input type="button" value="Launch"/>

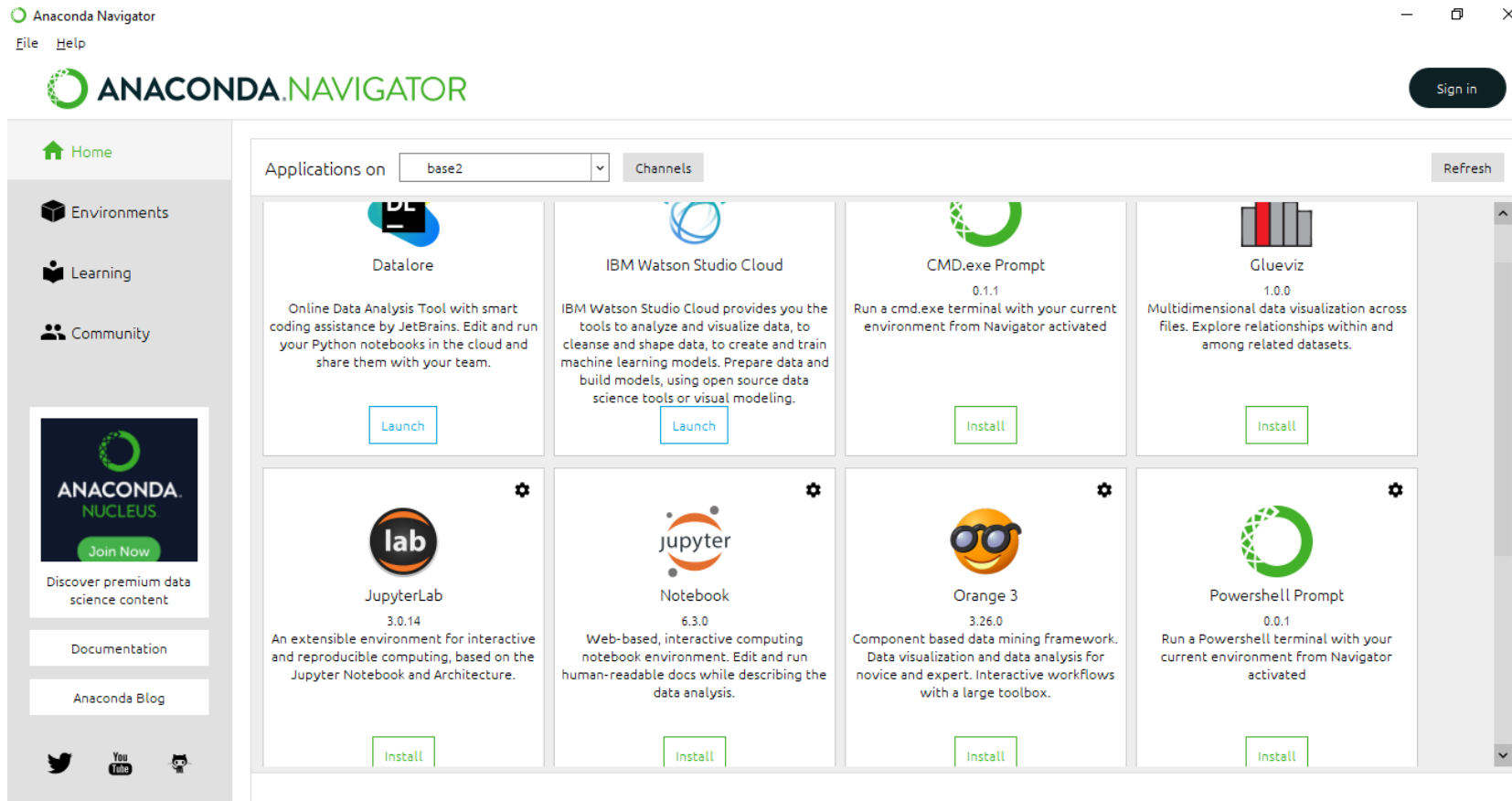
Se recomienda crear un ambiente

The screenshot shows the Anaconda Navigator application. On the left sidebar, the 'Environments' tab is selected, indicated by a yellow '1'. The main panel displays a list of environments. A 'Create new environment' dialog box is open, showing the name 'base2' (yellow '3'), location 'C:\Users\Juan.serna\Anaconda3\envs\base2', and packages 'Python 3.8' (yellow '4'). The 'Create' button is highlighted. A yellow '2' is placed near the 'Create' button in the bottom left corner of the main panel.

Name	Description	Version
base (root)		
base2		
base3		
base4		
base5		
base6		
base7		
base8		
base9		
base10		
base11		
base12		
base13		
base14		
base15		
base16		
base17		
base18		
base19		
base20		
base21		
base22		
base23		
base24		
base25		
base26		
base27		
base28		
base29		
base30		
base31		
base32		
base33		
base34		
base35		
base36		
base37		
base38		
base39		
base40		
base41		
base42		
base43		
base44		
base45		
base46		
base47		
base48		
base49		
base50		
base51		
base52		
base53		
base54		
base55		
base56		
base57		
base58		
base59		
base60		
base61		
base62		
base63		
base64		
base65		
base66		
base67		
base68		
base69		
base70		
base71		
base72		
base73		
base74		
base75		
base76		
base77		
base78		
base79		
base80		
base81		
base82		
base83		
base84		
base85		
base86		
base87		
base88		
base89		
base90		
base91		
base92		
base93		
base94		
base95		
base96		
base97		
base98		
base99		
base100		

1. Ir a Ambientes
2. Crear
3. Asignar nombre al nuevo ambiente (se puede dejar solo Python o incluir R)
4. Crear
5. Ir a Home del nuevo ambiente

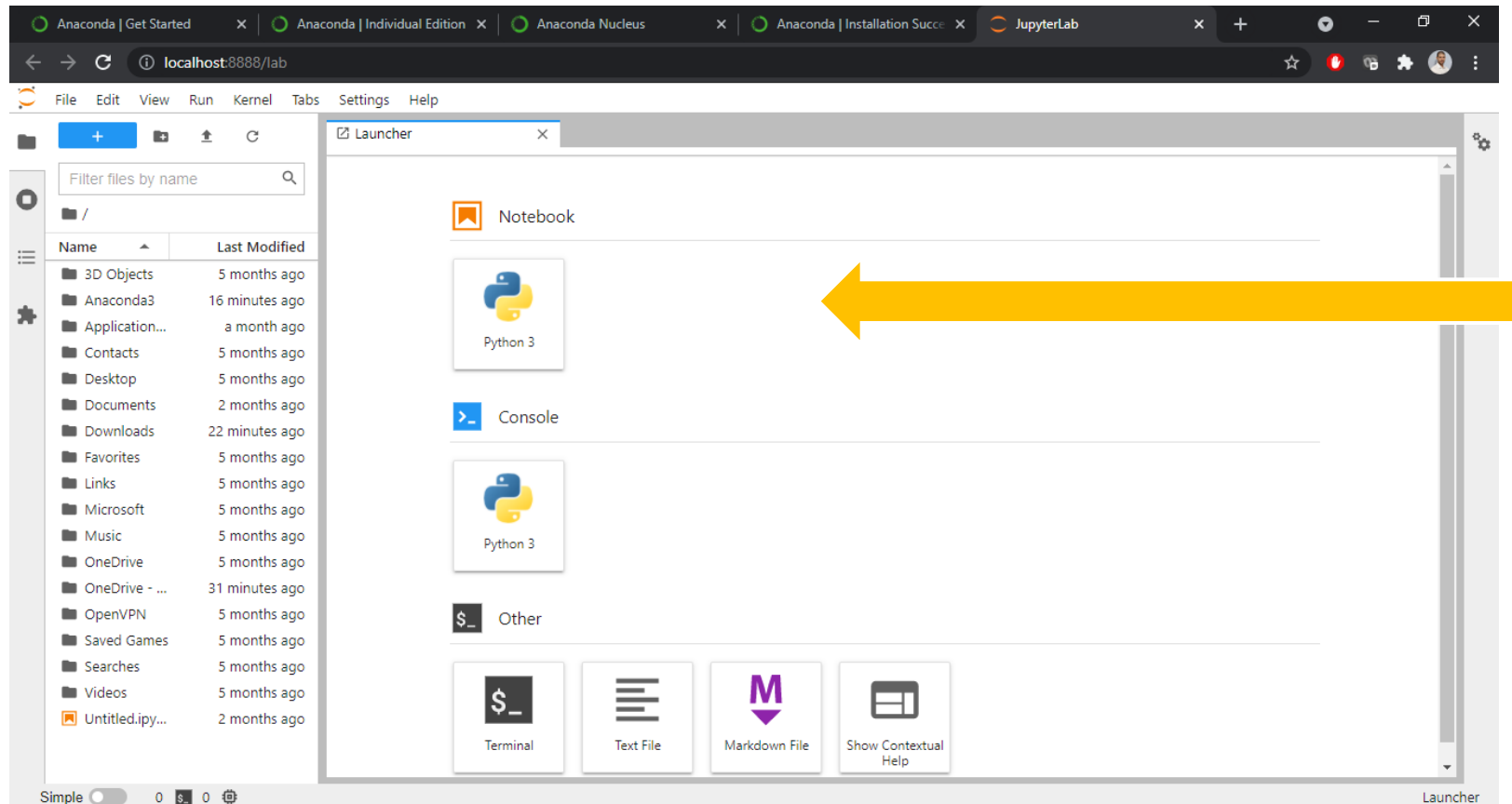
Instalar en el ambiente



1. JupyterLab (necesario)
2. CMD Prompt (opcional)

Iniciar (Launch) JupyterLab

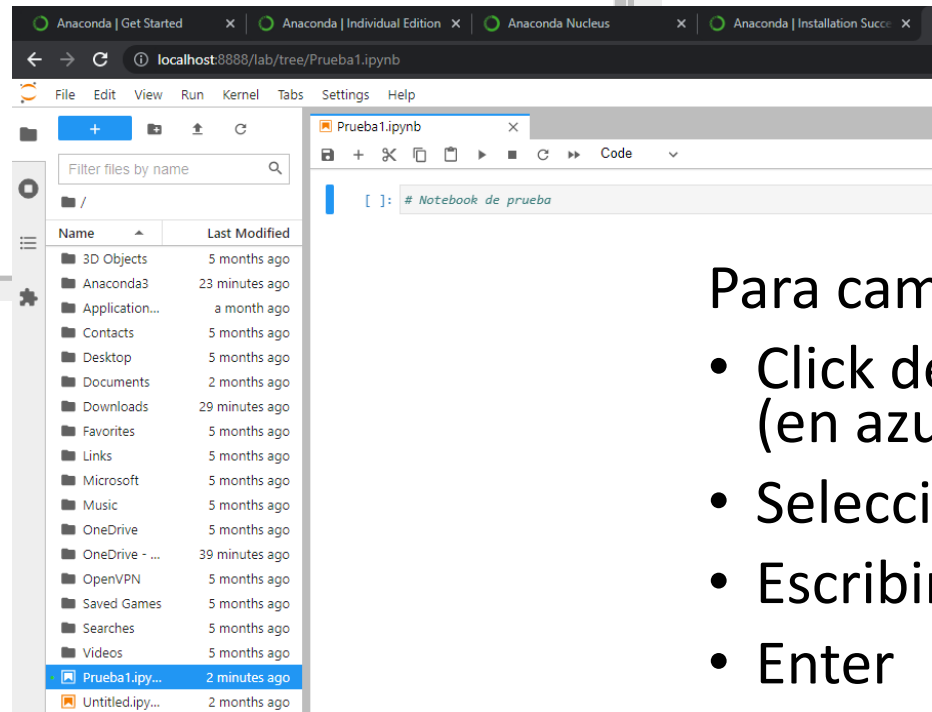
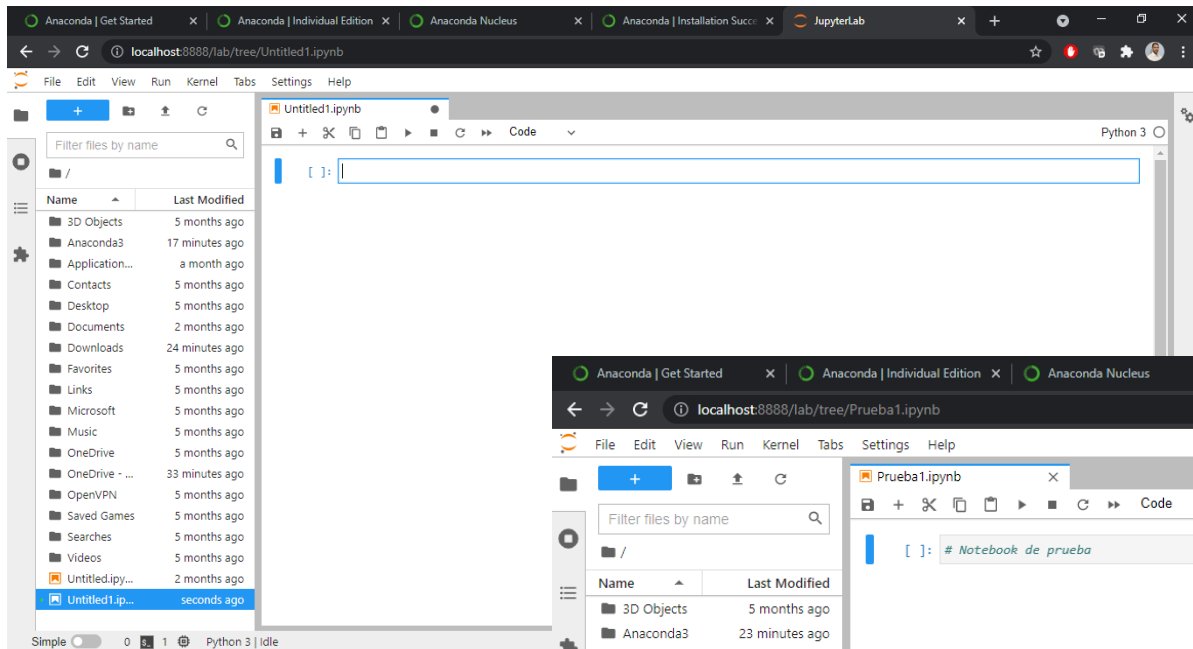
Debe aparecer esta vista en una pestaña del navegador



Hacer click en Python3
para crear un notebook

Notebook en ejecución

- Notar que ya aparece el notebook en los archivos
- Notar que .ipynb es de las extensiones de archivos Python



Para cambiar el nombre

- Click derecho sobre el nombre del Notebook (en azul)
- Seleccionar Rename
- Escribir el nuevo nombre
- Enter

Prueba en Notebook

1. Crear un Notebook vacío

2. Crear una variable *nombre* y asignarle su nombre

```
[ ]: nombre = 'Esteban'
```

¿Qué ocurre si no ponemos las comillas? ¿Y si en lugar de comillas simples usamos dobles?

3. Crear una variable *edad* y asignarle su edad

```
[ ]: edad = 31
```

¿Qué ocurre si ponemos el número entre comillas? ¿Cuál será la diferencia? ¿Y si ponemos un número "con coma"?

4. Imprimir en pantalla su *nombre* y *edad*

```
[ ]: print(nombre, edad)
```

¿Alguna forma es mejor?

```
[ ]: print('Mi nombre es', nombre, '. Mi edad es ', edad, ' años')
```

```
[ ]: print('Mi nombre es {}. Mi edad es {} años'.format(nombre, edad))
```