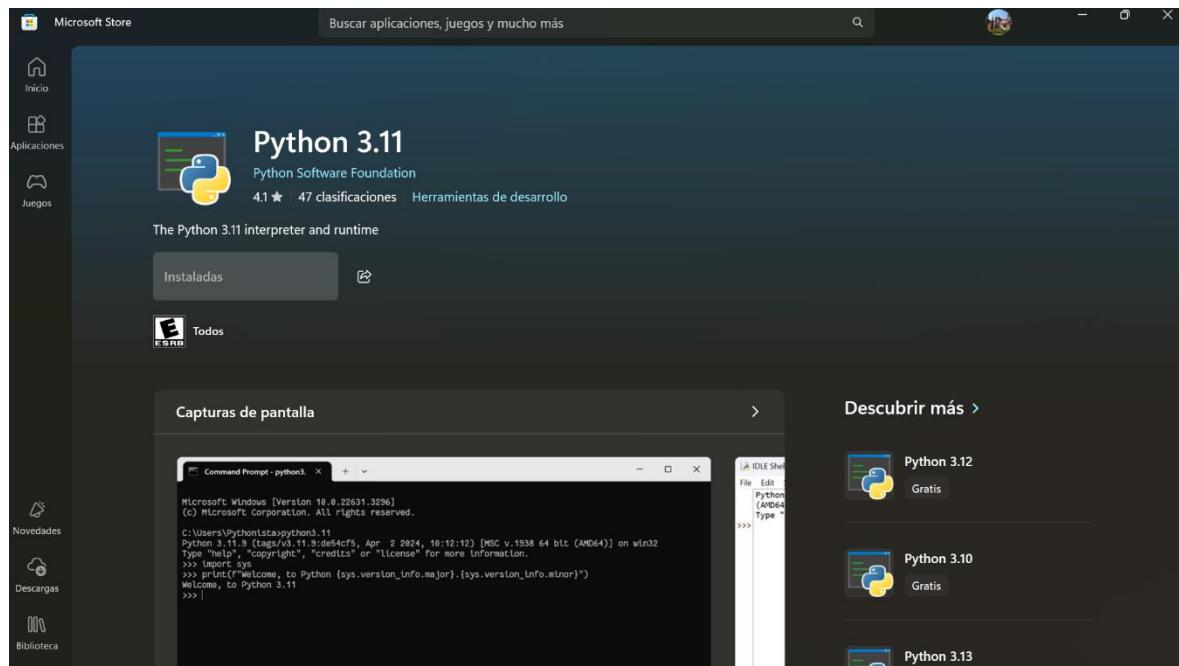


Instrucciones de Instalación de Python, Anaconda y Jupyter Lab en Windows 10/11

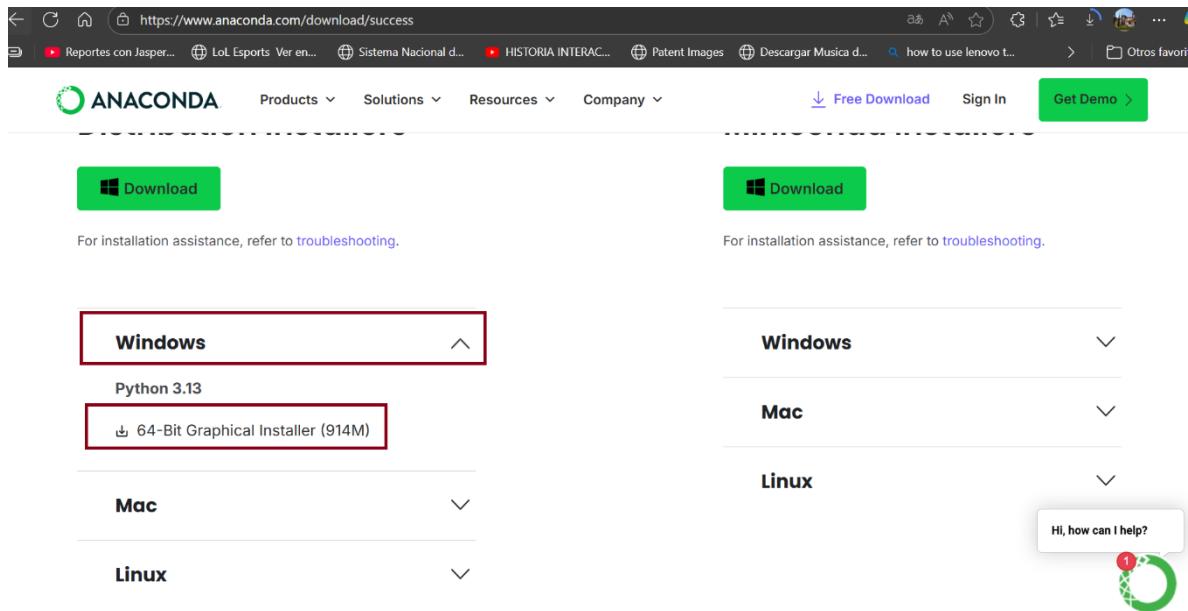
Este material guía al usuario a instalar Anaconda, un gestor de entornos y paquetes que facilita la administración de Python y sus dependencias, junto con el propio Python, el lenguaje de programación que se usará para desarrollar los ejercicios del taller. Las librerías necesarias aportan herramientas especializadas para el análisis de datos, la visualización y la manipulación de información. Además, el registro en la API del DENU-E INEGI y la obtención del token permiten autenticar las solicitudes y acceder al directorio económico de México de forma segura, tarea que se detallará a partir del paso 10.

NOTA: Si tienes macOS o Linux, se asume que eres un usuario familiarizado con la consola de sus sistemas operativos. Puedes consultar la Sección de Apéndices para guiarte en la instalación de las herramientas necesarias para el taller.

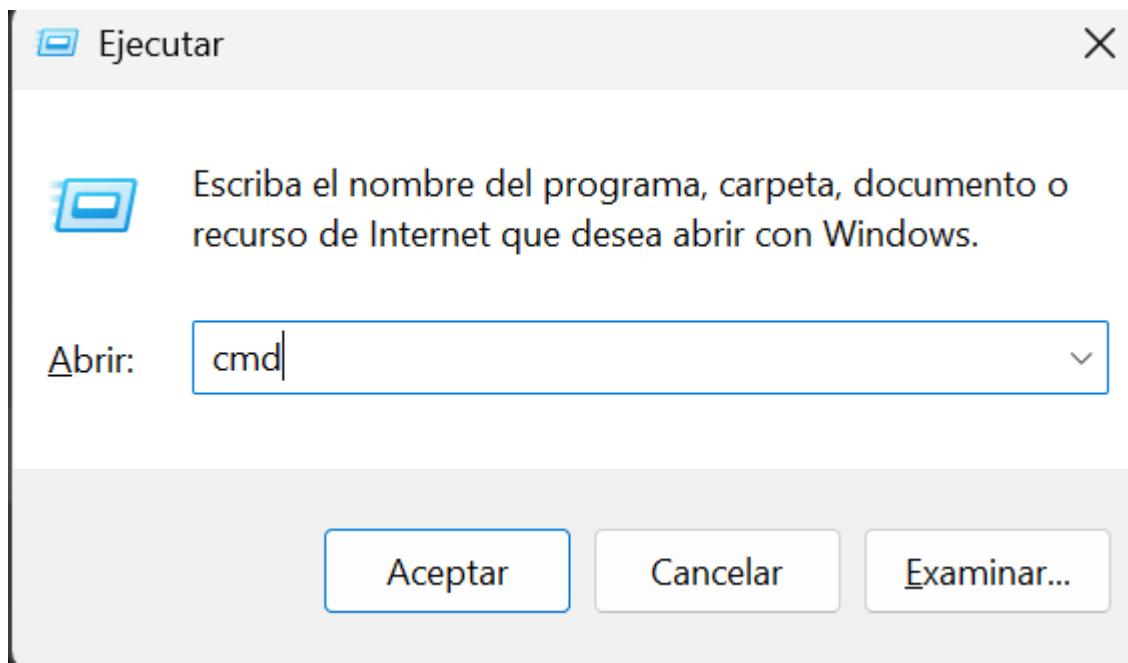
1.- Instalar Python desde la Microsoft Store.



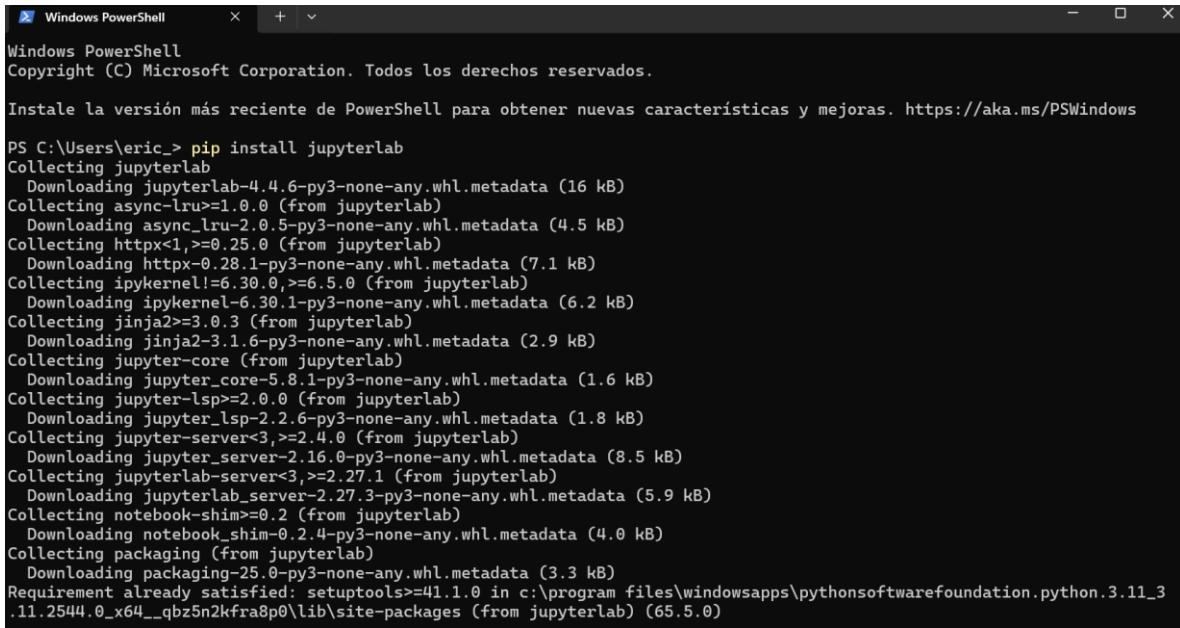
2.- Descargar anaconda desde <https://anaconda.com/download/success>.



3.- Instalar jupyterlab y librerías necesarias, en la terminal de Windows. Para abrirla es necesario presionar las Teclas Windows+R. Escribir “cmd” y presionar en Ejecutar.



4.- En la terminal ejecutamos primero el comando: “**pip install jupyterlab**” (sin comillas). Esperamos hasta que termine de instalarse, donde veremos nuevamente la consola sin ninguna instrucción o mensaje salve el directorio del usuario, en mi caso “C:\Users\eric_>”

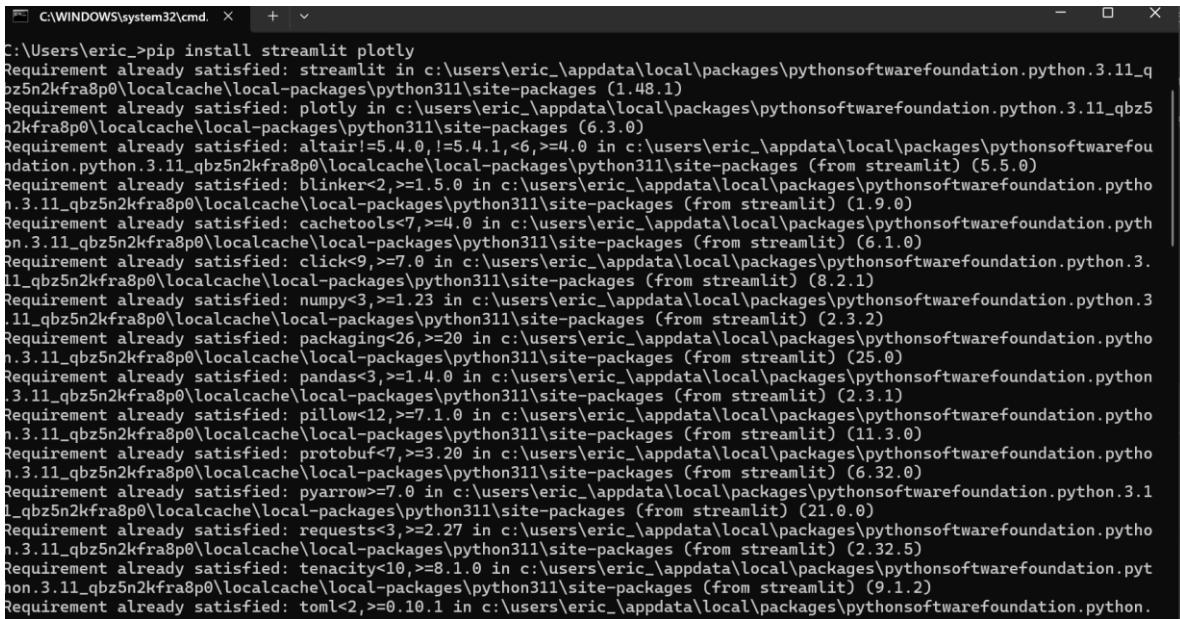


```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

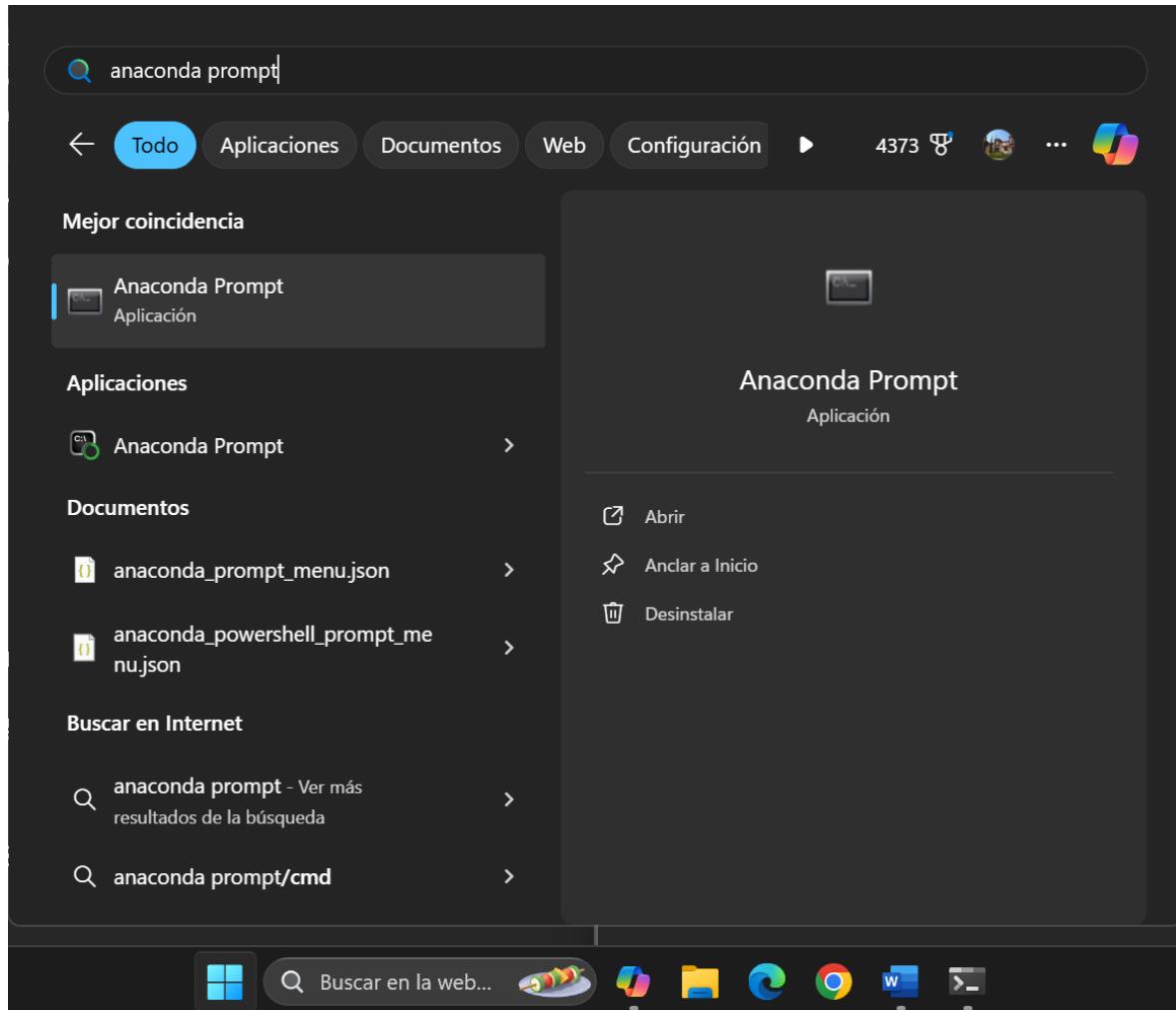
PS C:\Users\eric-> pip install jupyterlab
Collecting jupyterlab
  Downloading jupyterlab-4.4.6-py3-none-any.whl.metadata (16 kB)
Collecting async-lru>=1.0.0 (from jupyterlab)
  Downloading async_lru-2.0.5-py3-none-any.whl.metadata (4.5 kB)
Collecting httpx<1,>=0.25.0 (from jupyterlab)
  Downloading httpx-0.28.1-py3-none-any.whl.metadata (7.1 kB)
Collecting ipykernel!=6.30.0,>=6.5.0 (from jupyterlab)
  Downloading ipykernel-6.30.1-py3-none-any.whl.metadata (6.2 kB)
Collecting jinja2>=3.0.3 (from jupyterlab)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting jupyter-core (from jupyterlab)
  Downloading jupyter_core-5.8.1-py3-none-any.whl.metadata (1.6 kB)
Collecting jupyter-lsp=2.0.0 (from jupyterlab)
  Downloading jupyter_lsp-2.2.6-py3-none-any.whl.metadata (1.8 kB)
Collecting jupyter-server<3,>=2.4.0 (from jupyterlab)
  Downloading jupyter_server-2.16.0-py3-none-any.whl.metadata (8.5 kB)
Collecting jupyterlab-server<3,>=2.27.1 (from jupyterlab)
  Downloading jupyterlab_server-2.27.3-py3-none-any.whl.metadata (5.9 kB)
Collecting notebook-shim>=0.2 (from jupyterlab)
  Downloading notebook_shim-0.2.4-py3-none-any.whl.metadata (4.0 kB)
Collecting packaging (from jupyterlab)
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Requirement already satisfied: setuptools>=41.1.0 in c:\program files\windowsapps\pythonsoftwarefoundation.python.3.11_3.11.2544.0_x64_qbz5n2kfra8p0\lib\site-packages (from jupyterlab) (65.50)
```

5.- Una vez terminado de instalar jupyterlab, instalaremos las librerías necesarios con el comando:
“pip install streamlit pandas plotly matplotlib seaborn scikit-learn folium streamlit-folium numpy” (sin comillas nuevamente). Ojo, en algunas computadoras, será necesario repetir este comando en Anaconda Prompt, por lo que sugiero ejecutarlo en ambos. Seguir paso 6, y ejecutar este comando antes o después del paso 7.



```
C:\Windows\system32\cmd. >
C:\Users\eric->pip install streamlit plotly
Requirement already satisfied: streamlit in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (1.48.1)
Requirement already satisfied: plotly in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (6.3.0)
Requirement already satisfied: altair!=5.4.0,!5.4.1,<6,>4.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (5.5.0)
Requirement already satisfied: blinker<2,>1.5.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (1.9.0)
Requirement already satisfied: cachetools<7,>=4.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (6.1.0)
Requirement already satisfied: click<9,>=7.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (8.2.1)
Requirement already satisfied: numpy<3,>=1.23 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (2.3.2)
Requirement already satisfied: packaging<26,>=20 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (25.0)
Requirement already satisfied: pandas<3,>=1.4.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (2.3.1)
Requirement already satisfied: pillow<12,>=7.1.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (11.3.0)
Requirement already satisfied: protobuf<7,>=3.20 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (6.32.0)
Requirement already satisfied: pyarrow>=7.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (21.0.0)
Requirement already satisfied: requests<3,>=2.27 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (2.32.5)
Requirement already satisfied: tenacity<10,>=8.1.0 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from streamlit) (9.1.2)
Requirement already satisfied: toml<2,>=0.10.1 in c:\users\eric_\appdata\local\packages\pythonsoftwarefoundation.python.
```

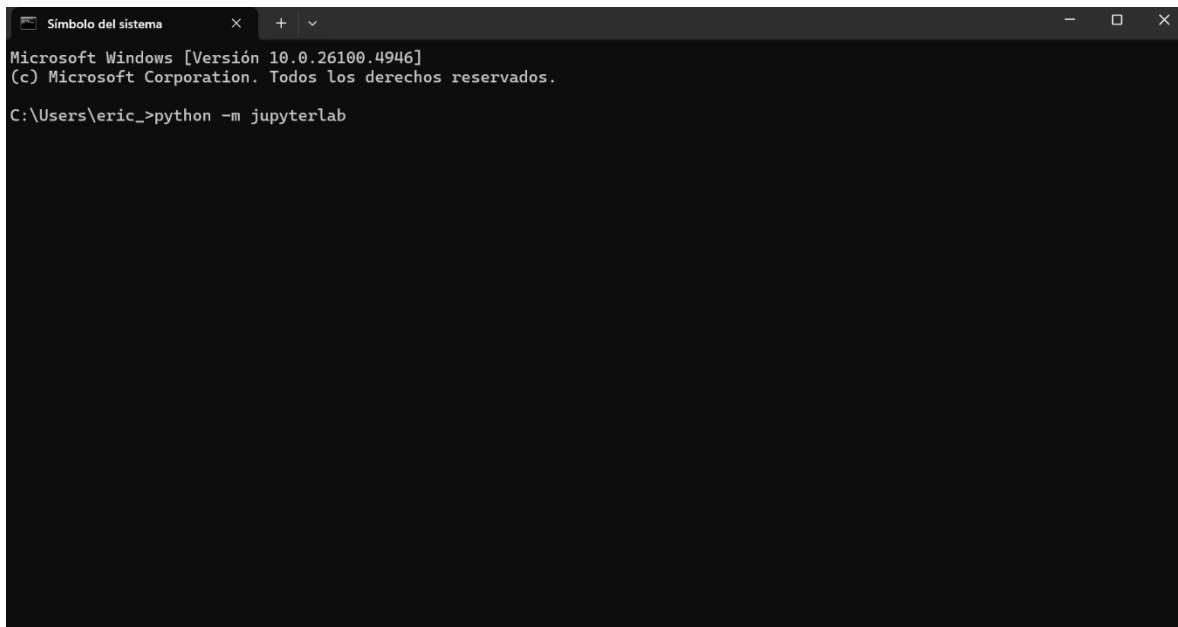
6.- Ahora, presionando la tecla Windows, buscaremos “Anaconda Prompt” y ejecutamos.



7.- Dentro de Anaconda Prompt, escribiremos el siguiente comando: “**conda install -c conda-forge folium geopandas**” (sin comillas), y ejecutamos.

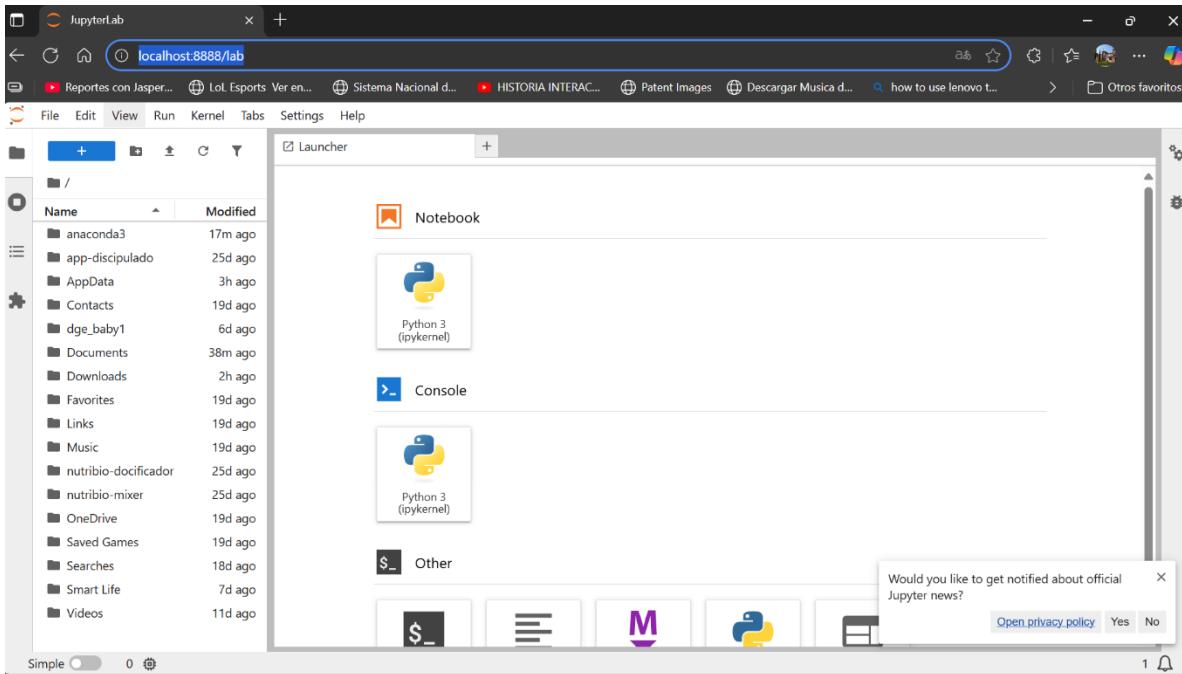
A screenshot of the Anaconda Prompt terminal window. The title bar says "Anaconda Prompt". The command "(base) C:\Users\eric_>conda install -c conda-forge folium geopandas" is visible in the terminal, indicating the user is about to run the command.

8.- Abriremos una nueva terminal del sistema (no anaconda prompt, ni la anterior que ya teníamos abierta): nuevamente presionar las Teclas Windows+R. Escribir “cmd” y presionar en Ejecutar. Solo que ahora, el comando a ejecutar será “**Python -m jupyterlab**”. NOTA: *En caso de que marque un error o mensaje con advertencia del “Path”, consultar la sección de: “Resolución de problemas comunes”, al final del documento. En caso contrario, hacer caso omiso de esta nota.*



The screenshot shows a Windows Command Prompt window titled "Símbolo del sistema". The window title bar includes standard icons for minimize, maximize, and close. The main area of the window displays the following text:
Microsoft Windows [Versión 10.0.26100.4946]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\eric_>python -m jupyterlab

9.- Lo anterior, debería abrirnos una nueva instancia de jupyter lab en nuestro navegador. De no ser así, mientras se ejecuta nuestra consola y sin cerrarla, escribimos en el navegador web de su preferencia: localhost:8888/lab y presionamos la tecla Enter, para acceder a la dirección en cuestión. En caso de pedirles un token o tener problemas aquí, consultar la sección de: “**Resolución de problemas comunes**”, al final del documento.



10.- Finalmente hay que darnos de alta en la página web del INEGI-DENUE y solicitar nuestro propio token. Para ello, vamos a la siguiente dirección:

https://www.inegi.org.mx/servicios/api_denue.html

A screenshot of the INEGI website at https://www.inegi.org.mx/servicios/api_denue.html. The page has a dark header with the INEGI logo and social media links. Below the header, there's a navigation bar with links like 'Temas', 'Programas de información', 'Sistemas de Consulta' (which is highlighted in blue), 'Desarrollo social', 'Infraestructura', and 'Acerca del INEGI'. On the left, there's a sidebar titled 'Para desarrolladores' with a tree view of APIs: 'Banco de Indicadores', 'DENUE' (which is selected and highlighted in grey), 'Google Maps', 'OpenLayers', 'Ruteo', 'Código fuente', 'MxSIG', 'Servicios Web' (with 'Catálogo Único de Claves' and 'Geostadísticas' under it), and 'Widgets'. The main content area has a title 'API del DENUE' and a description about the API. It also has tabs for 'Información disponible', 'Guía para desarrolladores' (which is selected and highlighted in blue), and 'Ejemplo'. A 'Listado de indicadores disponibles' section lists items like 'CLEE', 'Id de establecimiento', 'Nombre del establecimiento', 'Razón social', 'Clase de la actividad económica', and 'Estrato (Personal ocupado)'. On the right side, there are buttons for 'Compartir', 'Chat', and 'Sugerencias'.

11.- Allí, asegurándonos de estar en “Sistemas de Consulta”, damos click en “Guía para desarrolladores”.

Temas Programas de información **Sistemas de Consulta** Desarrollo social Infraestructura Acerca del INEGI Buscar... Buscar

Inicio / Sistemas de Consulta / APIs

Para desarrolladores

APIs

- Banco de Indicadores
- DENUE**
- Google Maps
- OpenLayers
- Ruleo

Código fuente

- MxSIG

Servicios Web

- Catálogo Único de Claves
- Geoestadísticas
- Servicio de Mapas Web

Widgets

- Banco de Indicadores

API del DENUE

La API del DENUE te permite consultar datos de identificación, ubicación, actividad económica y tamaño de más de 5 millones de establecimientos a nivel nacional, por entidad federativa y municipio. Puedes utilizar la API para crear aplicaciones que muestren la información directamente de las bases de datos del INEGI en el preciso momento en que se actualiza.

Información disponible Guía para desarrolladores Ejemplo

Guía para desarrolladores

- Introducción.
- Audiencia.
- Métodos para obtener información a través de la API
 - Buscar
 - Ficha
 - Nombre
 - BuscarEntidad

12.- Bajamos hasta encontrar la sección “Buscar”, y damos click en el vínculo marcado sobre la palabra “aqui”.

Buscar

Realiza una consulta de todos los establecimientos que cumplan las condiciones definidas.

Parámetros de entrada:

Condición Palabra(s) a buscar en el nombre del establecimiento, razón social, calle, colonia, clase de la actividad económica, entidad federativa, municipio y localidad. Para buscar más de una palabra se deberán separar con una coma.
Para buscar todos los establecimientos se deberá ingresar la palabra "todos".

Coordenadas Par de coordenadas que definen el punto en el mapa a partir del cual se hará la consulta alrededor. El formato de las coordenadas es latitud y longitud.

Distancia Cantidad de metros a partir de las coordenadas que definen el radio de búsqueda. La distancia máxima es de 5,000 metros.

Token Número único que permite hacer consultas, el cual se puede obtener al registrarse [aqui](#).

Ficha

Obtiene la información de un establecimiento en específico.

Parámetros de entrada:

Id Clave única del establecimiento.

Token Número único que permite hacer consultas, el cual se puede obtener al registrarse [aqui](#).

Nombre

Realiza una consulta de todos los establecimientos por nombre o razón social y puede ser acotado por entidad federativa.

Parámetros de entrada:

https://www.inegi.org.mx/servicios/api_denue.html#

13.- Proporcionamos nuestro correo electrónico, enviamos y revisamos la bandeja de entrada para obtener nuestro token, el cuál lo necesitaremos para realizar este taller.

The screenshot shows two main parts. The top part is a screenshot of a web browser displaying the INEGI website at https://www.inegi.org.mx/servicios/api_denue.html. A modal window titled "Cómo obtener un token" is open, asking for an email address ("Correo electrónico") which is highlighted with a red arrow. Below the input field is a blue button labeled "Obtener token". The bottom part of the screenshot shows a Gmail inbox with an incoming email from "Registro.api@inegi.org.mx" with the subject "Clave token para API DENUE". The email body contains a message to the user, a generated token "da6b1495-5952-45cb-882e-60a948768f5" (which is also highlighted with a red box), and instructions. The Gmail interface includes standard navigation and search bars.

Apéndices

El equivalente para **macOS** y **Linux (Ubuntu/Debian)**, siguiendo la misma lógica de tu guía de Windows (Python + JupyterLab + librerías, y el registro del token del **DENUE-INEGI** para el taller).

macOS

1. Instalar Python
 - Opción A (recomendada): instala **Anaconda** (incluye Python y conda).
 - Opción B: instala **Homebrew** y luego brew install python3.
2. Actualiza pip

```
python3 -m pip install --upgrade pip
```

3. Instalar JupyterLab

```
python3 -m pip install jupyterlab
```

4. Instalar librerías del taller con pip

```
python3 -m pip install streamlit plotly pandas matplotlib seaborn scikit-learn folium
```

5. (Opcional) Instalar librerías con conda (si usas Anaconda)

Usa el canal correcto conda-forge.

```
conda install -c conda-forge folium geopandas
```

6. Abrir JupyterLab

```
python3 -m jupyterlab
```

Si no abre el navegador, entra manualmente a <http://localhost:8888/lab>.

7. Solicitar el **token de la API DENUE-INEGI**

Regístrate en el portal del DENUE y solicita tu token; lo necesitarás para el taller, igual que en la guía original.

Linux (Ubuntu/Debian)

1. Instalar Python y herramientas básicas

```
sudo apt update
```

```
sudo apt install -y python3 python3-pip python3-venv
```

2. (Opcional) Instalar **Anaconda**

- Descarga el instalador de Anaconda para Linux y ejecútalo:

```
bash Anaconda3-* .sh
```

(Sigue las instrucciones del instalador).

3. Crear (opcional) un entorno virtual con venv

```
python3 -m venv venv
```

```
source venv/bin/activate
```

4. Actualiza pip

```
python3 -m pip install --upgrade pip
```

5. Instalar JupyterLab

```
python3 -m pip install jupyterlab
```

6. Instalar librerías del taller con pip

```
python3 -m pip install streamlit plotly pandas matplotlib seaborn scikit-learn folium
```

7. (Opcional) Instalar librerías con conda (si usas Anaconda)

```
conda install -c conda-forge folium geopandas
```

8. Abrir JupyterLab

```
python3 -m jupyterlab
```

Si no abre el navegador, entra a <http://localhost:8888/lab>.

9. Solicitar el **token de la API DENUE-INEGI**

Regístrate y obtén tu token como se indica en tu documento original; lo ocuparás para autenticar las consultas del taller.

Notas rápidas

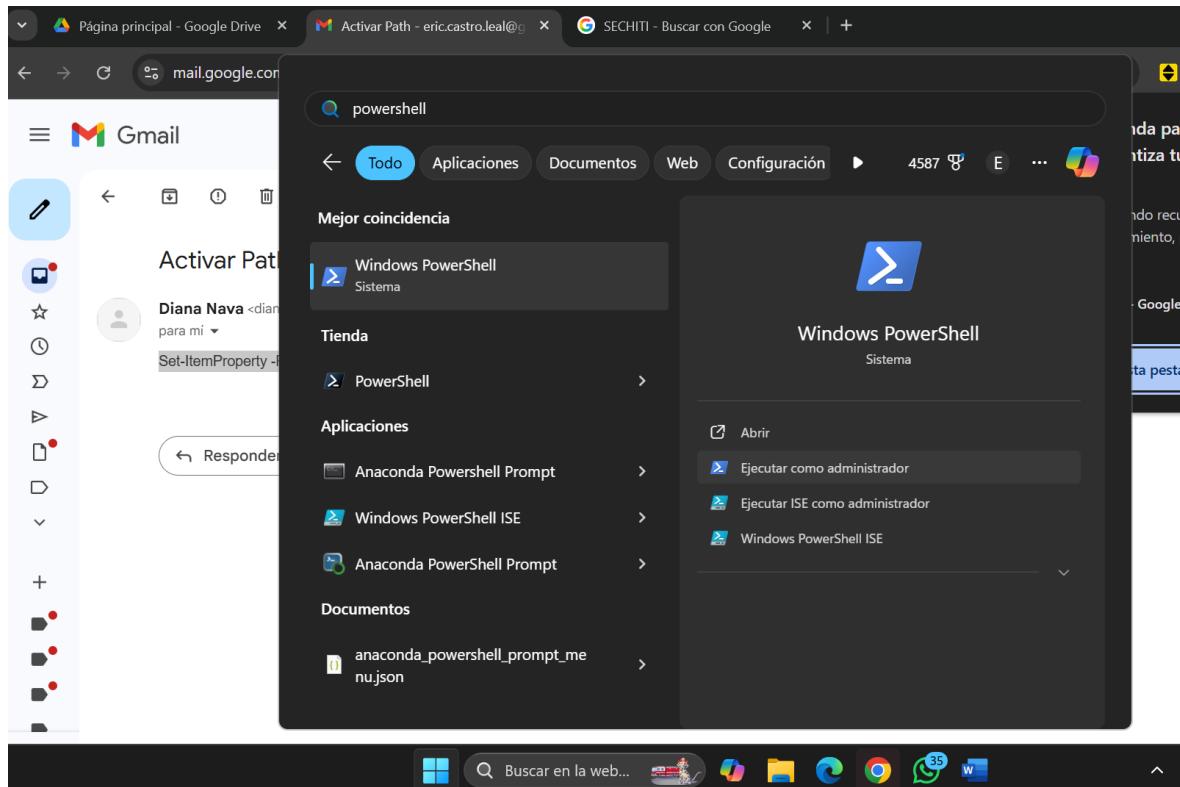
- En lugar de “CMD/Aplicación Prompt” (Windows), en macOS y Linux usa **Terminal**.
 - Los paquetes del taller son los mismos: **JupyterLab** y librerías para análisis, visualización y mapas (p. ej., **folium/geopandas**).
 - El **registro y uso del token DENUE-INEGI** es idéntico al de los pasos 10–13 del documento
-

Detalle de cada paquete

- streamlit: framework para crear dashboards interactivos
- pandas: manejo y análisis de datos
- plotly: visualizaciones interactivas (express y graph_objects vienen incluidos)
- matplotlib: trazado de gráficos básicos
- seaborn: visualizaciones estadísticas de alto nivel
- scikit-learn: algoritmos de machine learning
- folium: mapas interactivos basados en Leaflet
- streamlit-folium: integración de Folium en Streamlit (proporciona st_folium)
- numpy: operaciones numéricas y generación de datos aleatorios

Resolución de problemas comunes

Se advierte que en algunos equipos, donde los nombres de usuario o carpetas contienen espacios o son muy largos, se debe habilitar el “path” para poder instalar jupyter lab, sus librerías o bien ejecutarlo. Para ello, abriremos una instancia de “Windows PowerShell” y la ejecutaremos como administrador.



Finalmente, en esa instancia, ejecutaremos el siguiente comando: **`Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem" -Name "LongPathsEnabled" -Value 1`**. Esto nos permitirá poder ejecutar los comandos necesarios en el tutorial, en caso de tener problemas con el “path” o ruta del sistema, debido a su longitud o uso de caracteres especiales.

Otro problema común suele darse en que jupyterlab no inicia de forma automática o bien, nos pide un token para funcionar. Para solucionarlo, en la consola, después de haber ejecutado **`Python -m jupyterlab`**, revisamos los mensajes de la consola y ubicamos el que contenga la palabra “token”, manteniendo la tecla Ctrl presionada, damos click en el enlace y nos abrirá la

instancia de forma correcta y configurada.

```
npm prefix      x + v - □ ×
[I 2025-08-22 13:21:20.159 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-08-22 13:21:21.498 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-08-22 13:21:21.631 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-08-22 13:21:21.648 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-08-22 13:21:21.648 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-08-22 13:21:21.665 LabApp] JupyterLab extension loaded from C:\Users\eric_\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\jupyterlab
[I 2025-08-22 13:21:21.665 LabApp] JupyterLab application directory is C:\Users\eric_\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\share\jupyter\lab
[I 2025-08-22 13:21:21.665 LabApp] Extension Manager is 'pypi'.
[I 2025-08-22 13:21:23.097 ServerApp] jupyterlab | extension was successfully loaded.
[I 2025-08-22 13:21:23.107 ServerApp] Serving notebooks from local directory: C:\Users\eric_
[I 2025-08-22 13:21:23.107 ServerApp] Jupyter Server 2.16.0 is running at:
[I 2025-08-22 13:21:23.107 ServerApp] http://localhost:8888/lab?token=9c2fc3d81252f616d7d62fe5273caa3a0e9c0e751f7cad4
[I 2025-08-22 13:21:23.107 ServerApp] http://127.0.0.1:8888/lab?token=9c2fc3d81252f616d7d62fe5273caa3a0e9c0e751f7cad4
d4
[I 2025-08-22 13:21:23.107 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2025-08-22 13:21:23.247 ServerApp]

To access the server, open this file in a browser:
  file:///C:/Users/eric_/AppData/Roaming/jupyter/runtime/jpserver-8800-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/lab?token=9c2fc3d81252f616d7d62fe5273caa3a0e9c0e751f7cad4
  http://127.0.0.1:8888/lab?token=9c2fc3d81252f616d7d62fe5273caa3a0e9c0e751f7cad4
[I 2025-08-22 13:21:24.623 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
```