

Iniciar en el módulo 1:

# Análisis de datos en el ecosistema Python

ELÍAS NEZAMOLESLAMI

*Senior Data Scientist en Minsait,  
An Indra Company*

Curso Académico 2023-2024



Cámara de Comercio Alemana  
para España  
Deutsche Handelskammer  
für Spanien



**CES**  
Juan Pablo II  
**MADRID**

# Índice

---

- **Objetivos del curso**
- **Contenidos del módulo 1**
- **¿Por Qué Python?**
- **Entornos de Python**
- **Suite de herramientas**
  - Entornos de desarrollo integrado (IDE)
  - Jupyter Notebooke
  - Anaconda Prompt
  - Instalación de paquetes/librerías
- **Características de Python**
- **Referencias**
- **Anexos**
  - Sesiones planificadas
  - Composición de la nota final
  - Instalación de Anaconda



**CES**  
Juan Pablo II  
MADRID



Cámara de Comercio Alemana  
para España  
Deutsche Handelskammer  
für Spanien

## Objetivos del curso

- ☐ Definir el **planteamiento analítico** del problema en un **proyecto** relacionado con Análisis de Big Data.
- ☐ Identificar diferentes **fuentes de datos** y realizar un análisis **descriptivo** de sus contenidos.
- ☐ Aplicar varias técnicas de carga, ingesta, limpieza y manipulación de datos (**ETL en Python**).
- ☐ Realizar el flujo entero de **modelado estadístico/matemático** aplicando distintas técnicas de **Machine Learning** mediante **librerías open-source** en Python (scikit-learn, tensorflow, keras, skforecast...).
- ☐ **Visualizar** los resultados obtenidos en Python (matplotlib, seaborn, plotly...).
- ☐ **Documentar** los códigos desarrollados y presentar los **informes** de proyectos (Markdown).

## Contenidos del módulo

### Bloque (I)

- **Iniciación** en análisis de datos con Python

### Bloque (II)

- Modelos de **regresión** en Machine Learning con Python

### Bloque (III)

- Modelos de **clasificación** en Machine Learning con Python

### Bloque (IV)

- Modelos de **clustering** en Machine Learning con Python

### Bloque (V)

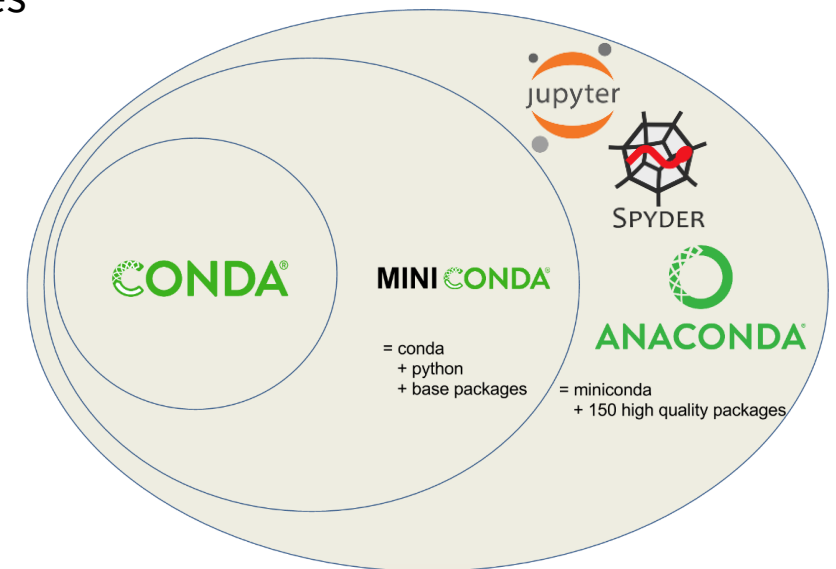
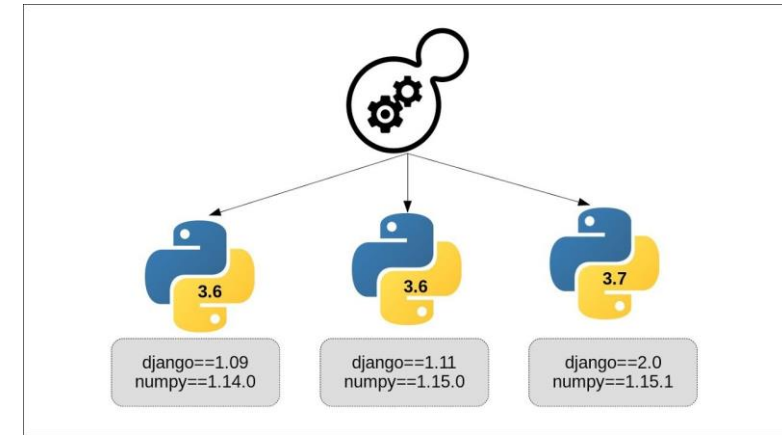
- Análisis de **series temporales** con Python

## ¿Por Qué Python?

- ☐ Python es uno de los lenguajes de programación **más populares** que existen actualmente.
- ☐ La **sintaxis** de Python te ayuda a escribir código “limpio”. Tu código será fácil de leer, de modificar y por lo tanto de mantener.
- ☐ Python es muy **eficiente**. En pocas líneas de código se puede hacer mucho más que algunos otros lenguajes.
- ☐ Se puede utilizar Python para muchas cosas: desde scripts simples hasta **desarrollos web**, sistemas **distribuidos** o tareas **científicas**.
- ☐ Python tiene una gran **comunidad** de usuarios y desarrolladores que te ayuda a resolver cualquier problema.
- ☐ Es un lenguaje accesible, portable y de código abierto (**open-source**).

## Entornos de Python

- ❑ Instalación directa de Python y las librerías necesarias:
  - Entorno único
- ❑ Entornos virtuales (multi-proyecto):
  - **venv**: entornos nativos de python
  - **virtualenv, pipenv...**: librerías como gestor de entornos virtuales
  - **conda**: gestor de entornos y dependencias
    - **Anaconda**: Suite de herramientas de Análisis de datos
    - **Miniconda**: Solamente el gestor de entornos conda



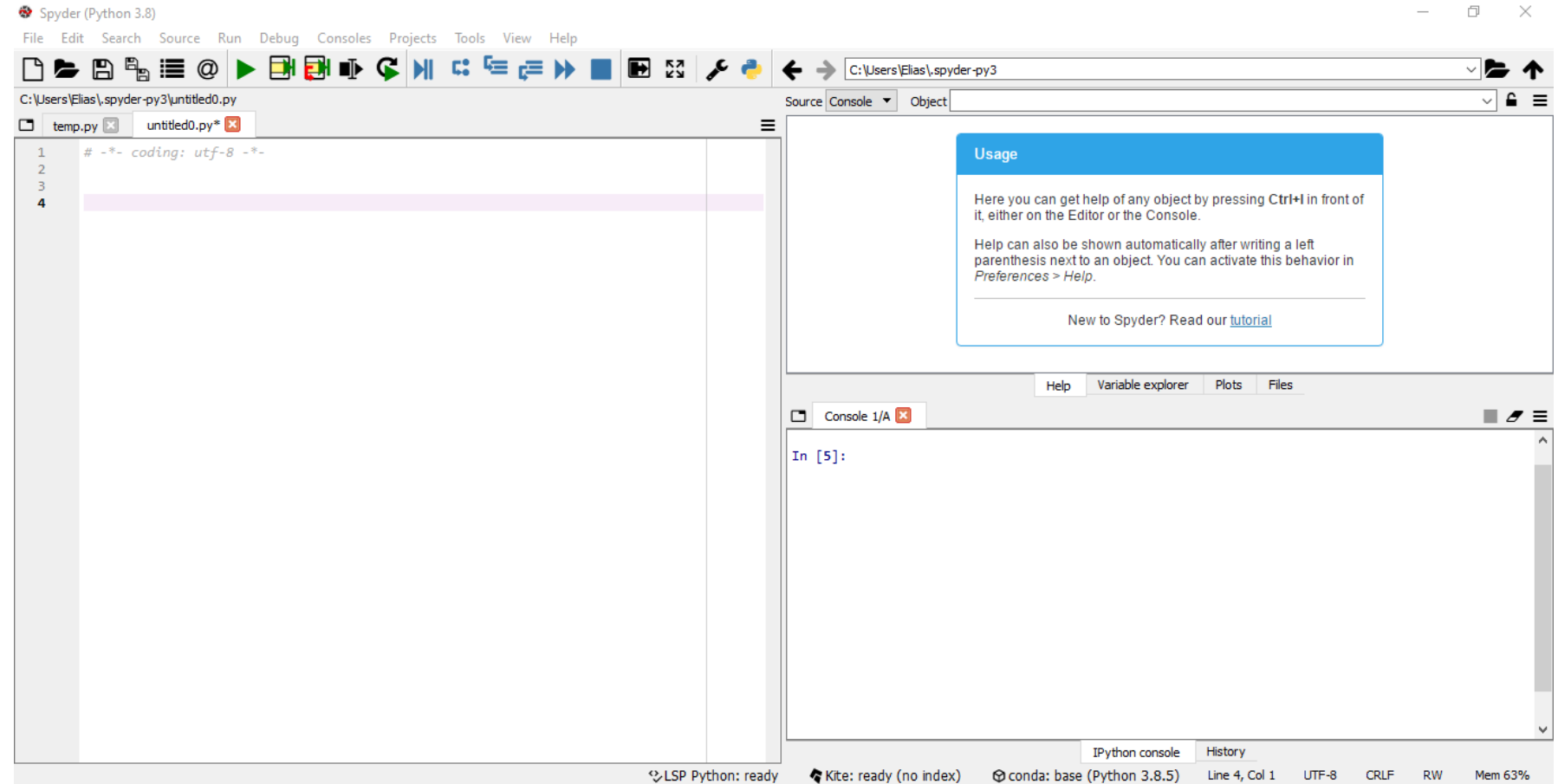
## Entornos de desarrollo integrado (IDE)

☐ Spyder

☐ Pycharm

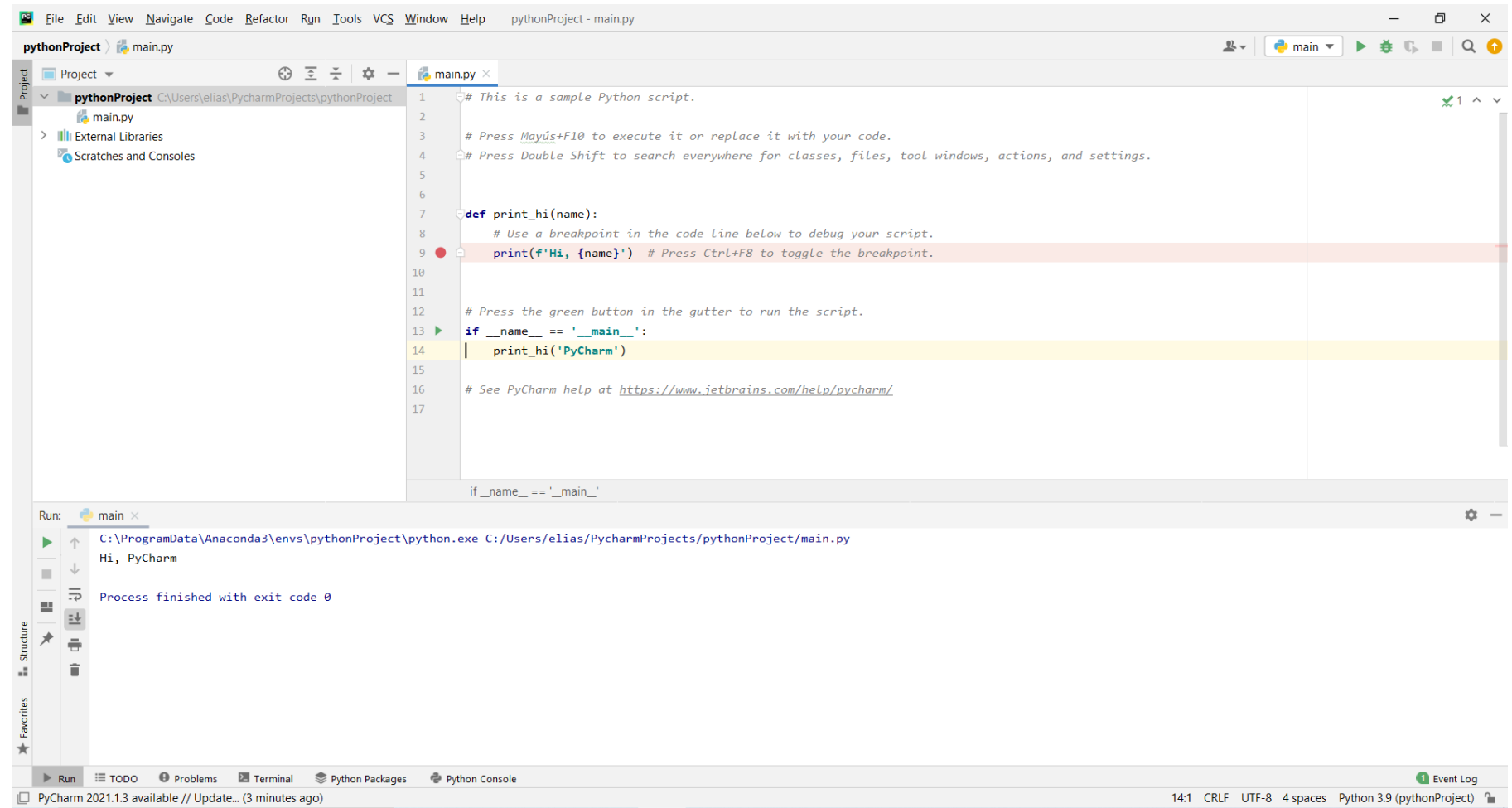
☐ Visual Studio Code

☐ Vim, Atom...



## Entornos de desarrollo integrado (IDE)

- ☐ Spyder
- ☒ **Pycharm**
- ☐ Visual Studio Code
- ☐ Vim, Atom...



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `main.py` with the following content:

```

1 # This is a sample Python script.
2
3 # Press Mayús+F10 to execute it or replace it with your code.
4 # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7 def print_hi(name):
8     # Use a breakpoint in the code line below to debug your script.
9     print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17

```

The script is executed, and the output is shown in the Run console at the bottom:

```

Run: main x
C:\ProgramData\Anaconda3\envs\pythonProject\python.exe C:/Users/elias/PycharmProjects/pythonProject/main.py
Hi, PyCharm
Process finished with exit code 0

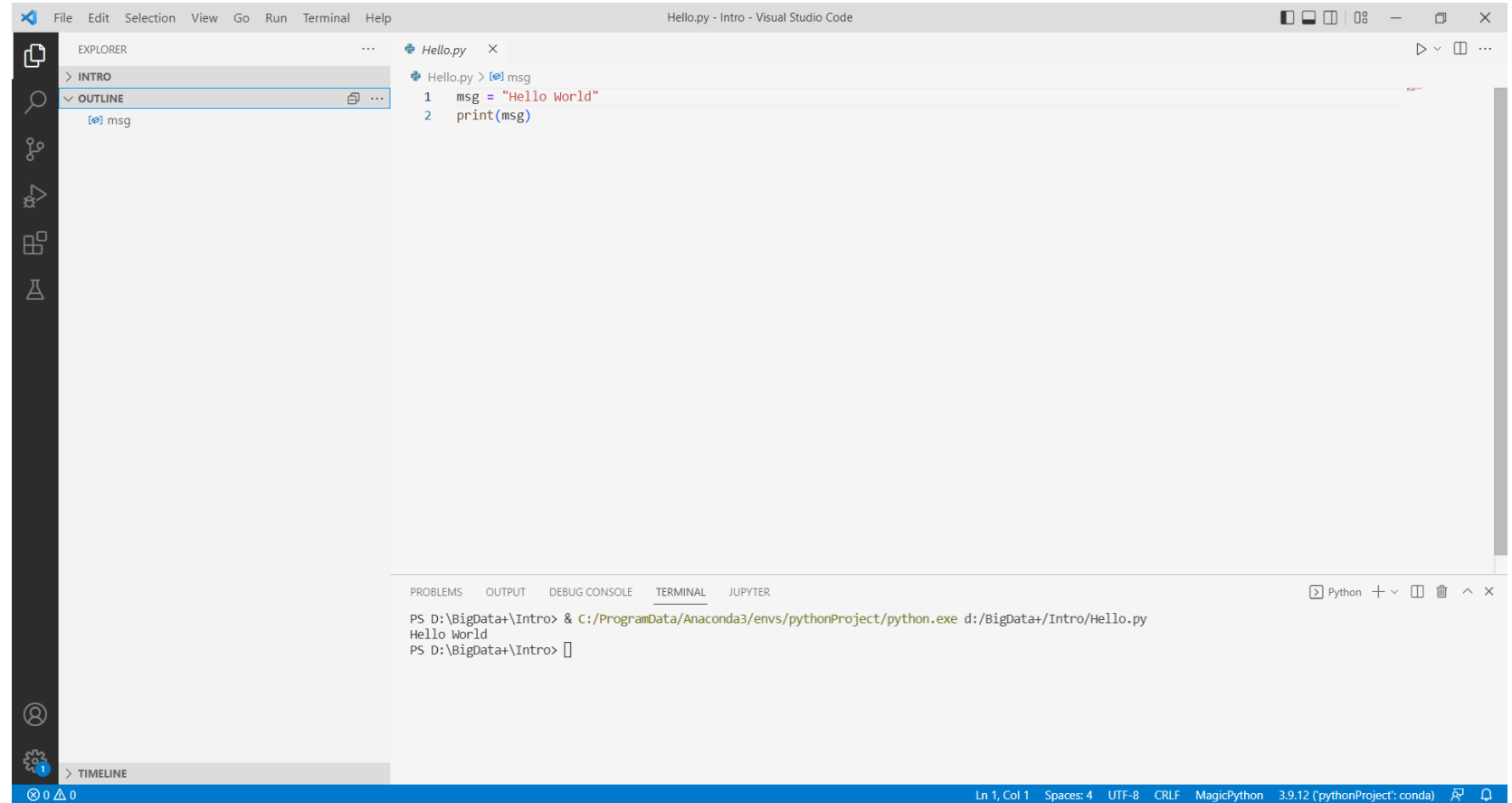
```

The status bar at the bottom indicates the current configuration: 14:1 CRLF UTF-8 4 spaces Python 3.9 (pythonProject).

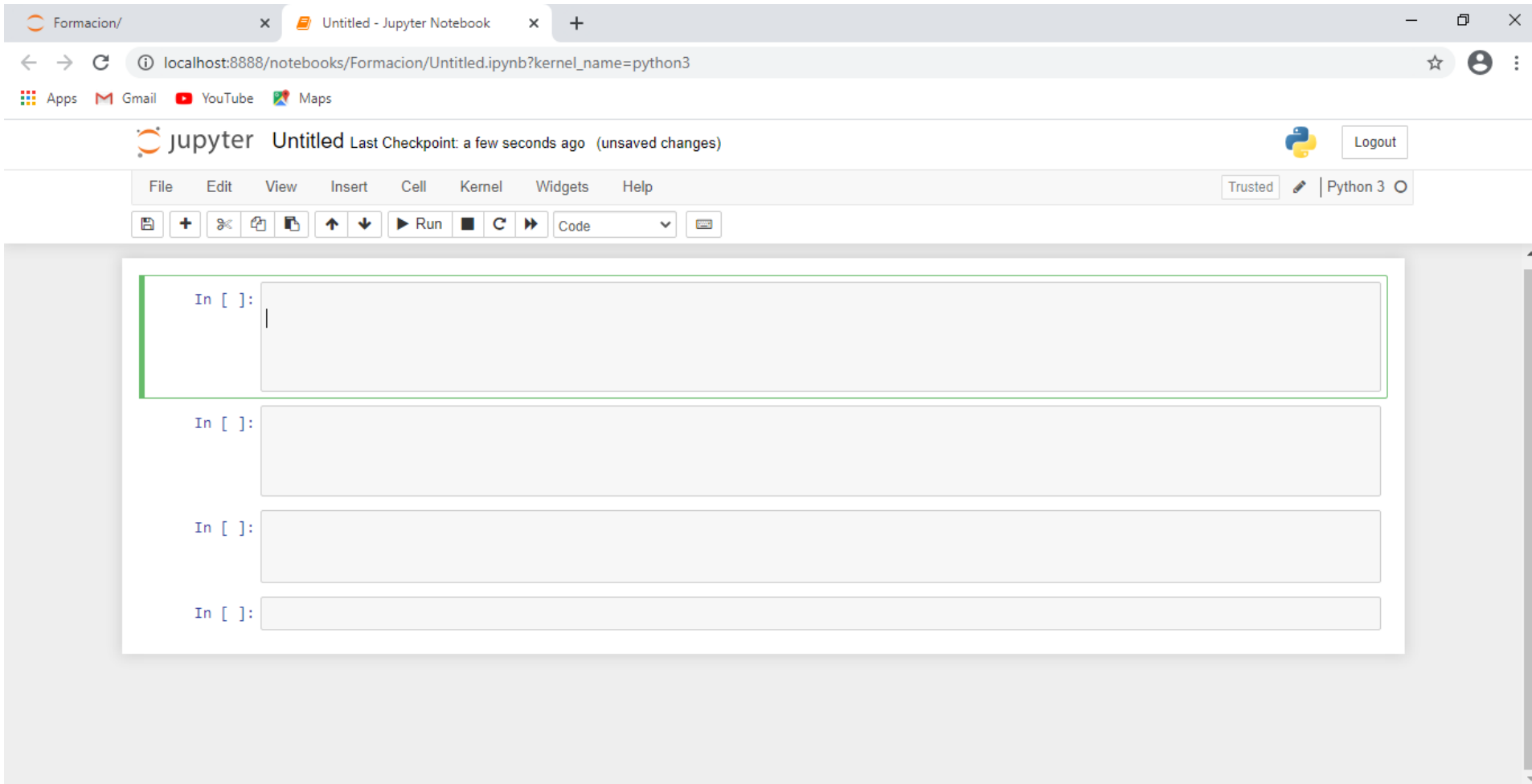


## Entornos de desarrollo integrado (IDE)

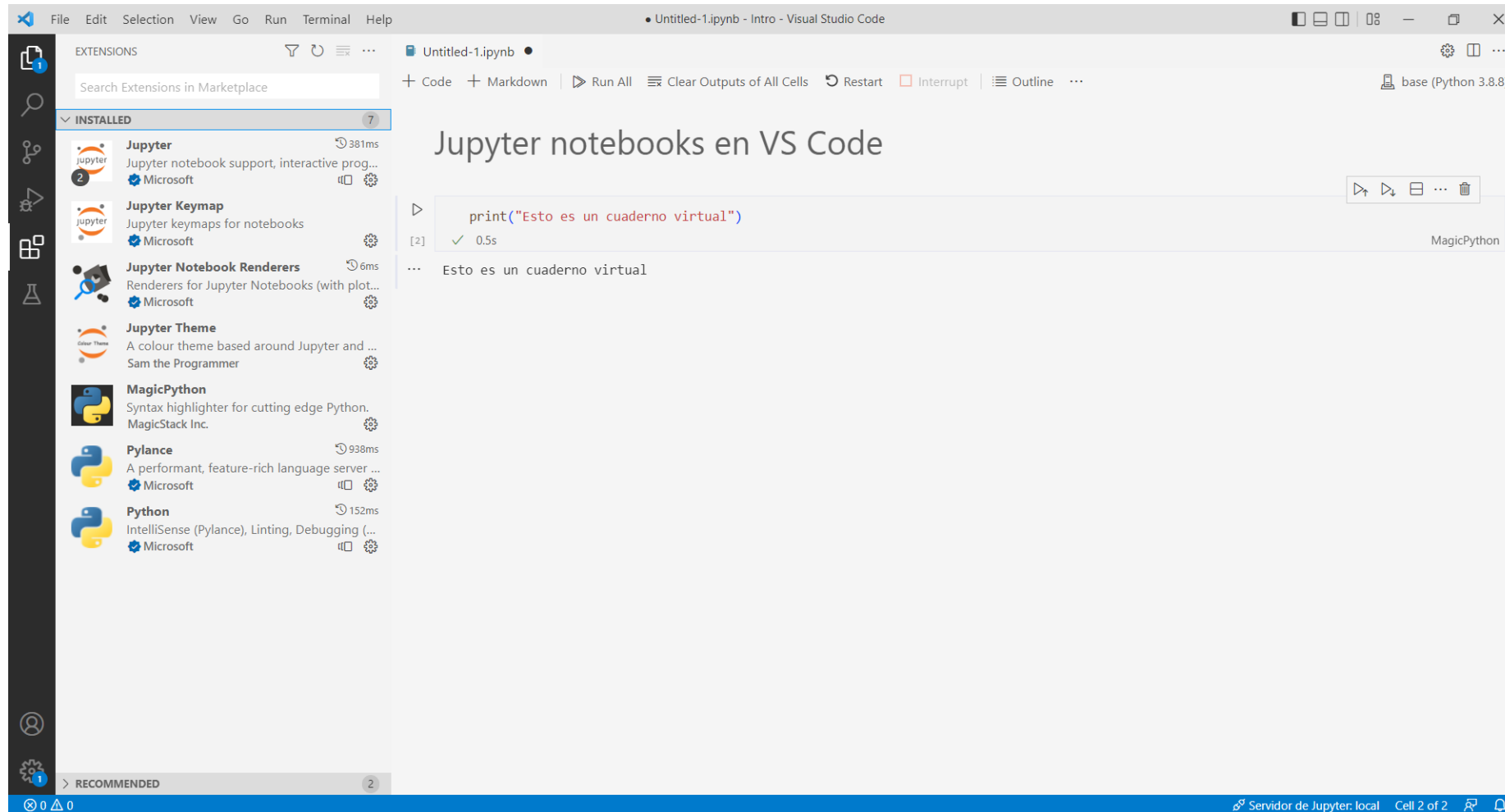
- ☐ Spyder
- ☐ Pycharm
- ☒ **Visual Studio Code**
- ☐ Vim, Atom...



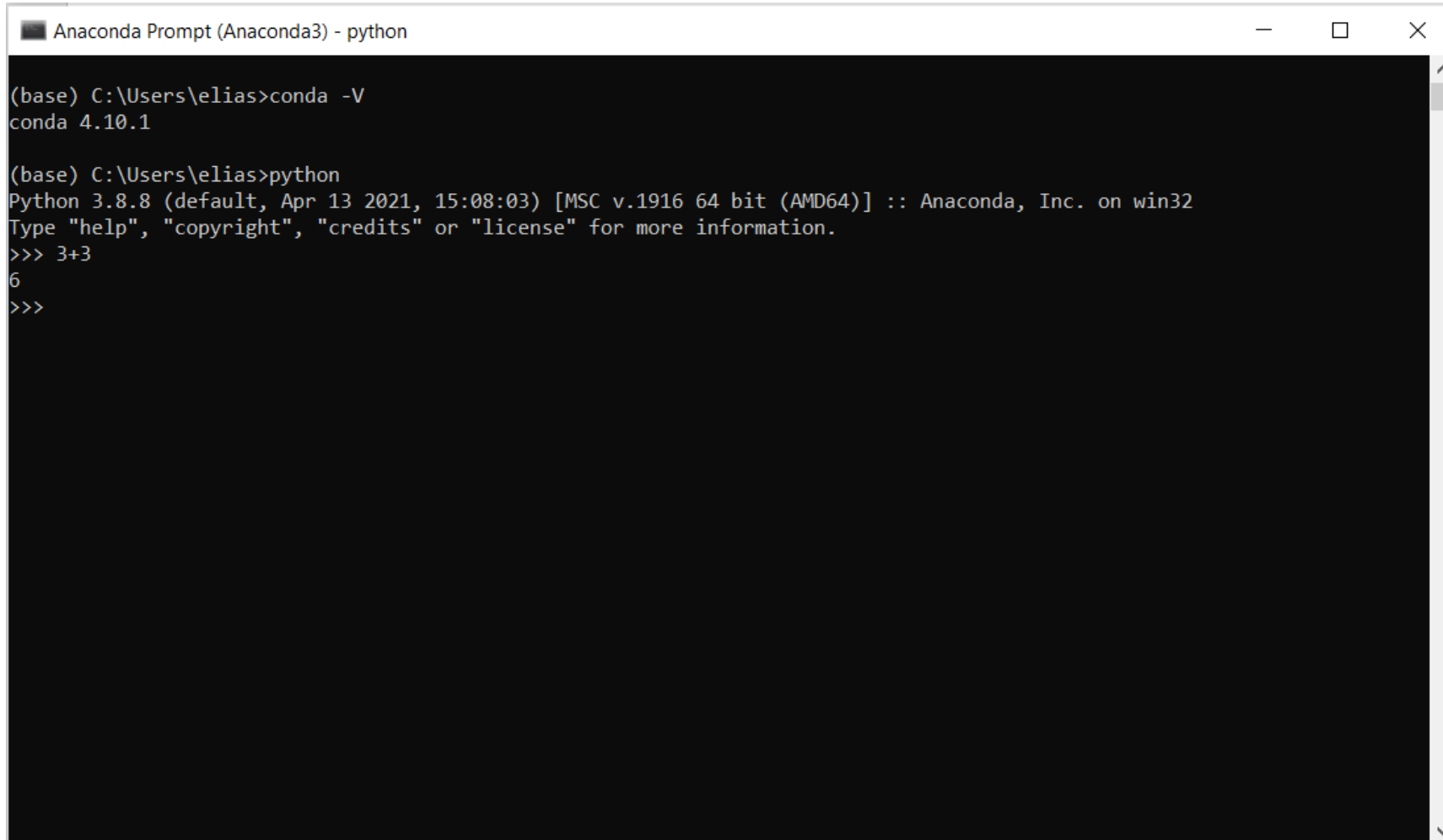
# Jupyter Notebooke – Cuaderno virtual de programación



# Jupyter Notebooks en VS Code



## Anaconda Prompt – Consola para comandos de conda



```
Anaconda Prompt (Anaconda3) - python

(base) C:\Users\elias>conda -V
conda 4.10.1

(base) C:\Users\elias>python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 3+3
6
>>>
```

## Instalación de paquetes/librerías

pip install <nombre\_del\_paquete>

conda install <nombre\_del\_paquete>

```
Command Prompt
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Elias>pip install shap
Collecting shap
  Using cached shap-0.38.1-cp38-cp38-win_amd64.whl (4.1 MB)
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (from shap) (1.1.3)
Requirement already satisfied: numba in c:\programdata\anaconda3\lib\site-packages (from shap) (0.51.2)
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\site-packages (from shap) (0.23.2)
Requirement already satisfied: tqdm>4.25.0 in c:\programdata\anaconda3\lib\site-packages (from shap) (4.50.2)
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages (from shap) (1.5.2)
Requirement already satisfied: cloudpickle in c:\programdata\anaconda3\lib\site-packages (from shap) (1.6.0)
Requirement already satisfied: slicer==0.0.7 in c:\programdata\anaconda3\lib\site-packages (from shap) (0.0.7)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (from shap) (1.19.2)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\programdata\anaconda3\lib\site-packages (from pandas->shap) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in c:\programdata\anaconda3\lib\site-packages (from pandas->shap) (2020.1)
Requirement already satisfied: llvmlite<0.35,>=0.34.0.dev0 in c:\programdata\anaconda3\lib\site-packages (from numba->shap) (0.34.0)
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\site-packages (from numba->shap) (50.0.3.post20201107)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->shap) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->shap) (0.17.0)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas->shap) (1.15.0)
Installing collected packages: shap
Successfully installed shap-0.38.1

C:\Users\Elias>
```

Comando para  
instalar un paquete

El paquete está  
instalado

## Características de Python

- ☐ Python es un lenguaje de **alto nivel** con propósito general, muy rápido de programar (comparado con C o Java) y con código muy **fácil de leer**.
- ☐ Es un lenguaje **interpretado** (como Matlab o R), con lo cual lo hace **más lento en ejecución** que C++ o por ejemplo Java pero ofrece mayor portabilidad en **diversos plataformas y sistemas operativos**.
- ☐ Además de los paquetes (**packages** o **libraries**) preinstalados con Python, hay una gran cantidad de paquetes de código abierto para hacer una **programación orientada a objetos** o incluso funcional que nos permite desarrollar **aplicaciones complejas** en Python.
- ☐ También existen muchos **conectores** con otros lenguajes, con lo que es un buen programa para **unir piezas** sueltas de otros y **gobernar procesos**.

## Referencias - Plataformas para iniciarse en Python

### ☐ udemy.com

- Amplia selección de cursos de muchas temáticas con algunos cursos gratuitos para empezar con la introducción de Python y análisis de datos.

### ☐ learnpython.org

- Perfecto para empezar con tutoriales básicos de Python.

### ☐ udacity.com

- Distintos cursos gratuitos relacionados con la programación y tratamiento de datos.

### ☐ dataquest.io

- Plataforma interactiva para aprender Python y análisis de datos. Se puede empezar de forma gratuita para hacer varios cursos de iniciación.

## Referencias - Plataformas de apoyo

### ☐ **stackoverflow.com**

- El foro de ayuda más grande de internet
- En inglés mucho mejor, más soluciones que en español
- Dicen que si la solución a tu problema no está aquí, es probable que no exista o que la pregunta esté mal planteada!! 😊

### ☐ **w3schools.com**

- Es el sitio web con más tutoriales interactivos para diferentes lenguajes de programación como: Python, Java, R, Java Script, HTML, SQL...

### ☐ **geeksforgeeks.org**

- Muchos ejemplos prácticos y muy fáciles de entender con trozos de código desarrollado

### ☐ **medium.com**

- Un foro muy popular con muchos artículos de referencia y manuales explicativos

### ☐ **¿¿ChatGPT, Copilot....??**





## Anexos:

- Sesiones planificadas
- Composición de la nota final
- Instalación de Anaconda

# Sesiones planificadas



**CES**  
Juan Pablo II  
MADRID



Cámara de Comercio Alemana  
para España  
Deutsche Handelskammer  
für Spanien

## I- Iniciación en análisis de datos con Python:

### ☐ Sesión (1)

- **04/09/2023**
- Kick-off del programa
- Introducción a Python

### ☐ Sesión (2)

- **11/09/2023**
- Estructuras de datos en Python
- Control del flujo de programación
- Clases y funciones

### ☐ Sesión (3)

- **18/09/2023**
- Tratamiento de datos (Arrays y Matrices)

### ☐ Sesión (4)

- **25/09/2023**
- Tratamiento de datos (DataFrames y Series)

### ☐ Sesión (5)

- **02/10/2023**
- Aplicar un caso de uso real (Análisis descriptivo)
- **Cierre del bloque (I)**

## II- Modelos de **regresión** en Machine Learning con Python:

- ☐ Sesión (6)
  - **09/10/2023**
  - Regresión Lineal (simple y múltiple)
- ☐ Sesión (7)
  - **16/10/2023**
  - Regularización (Ridge, Lasso, ElasticNet)
- ☐ Sesión (8)
  - **23/10/2023**
  - Árboles de decisión (Decision Tree Regression)
- ☐ Sesión (9)
  - **30/10/2023**
  - Bosques Aleatorios y Gradient Boosting (Random Forest Regression)
- ☐ Sesión (10)
  - **06/11/2023**
  - Neural Network Regression & Deep Learning
- ☐ Sesión (11)
  - **13/11/2022**
  - Aplicar un caso de uso real
  - **Cierre del bloque (II)**

### III- Modelos de **clasificación** en Machine Learning con Python:

- ☐ Sesión (12)
  - **20/11/2023**
  - Regresión Logística (Logit Regression)
- ☐ Sesión (13)
  - **27/11/2023**
  - Tratamiento y selección de variables
- ☐ Sesión (14)
  - **04/12/2022**
  - Árboles de decisión y Bosques Aleatorios
- ☐ Sesión (15)
  - **11/12/2023**
  - Naive Bayes Classifier, K-Nearest Neighbors & SVM (Support Vector Machines)
- ☐ Sesión (16)
  - **18/12/2023**
  - Neural Networks & Deep Learning (MLP y DNN)
- ☐ Sesión (17)
  - **08/01/2024**
  - Optimizar los hiper-parámetros y Ensemble Learning
- ☐ Sesión (18)
  - **15/01/2024**
  - Aplicar un caso de uso real
  - **Cierre del bloque (III)**

## IV- Modelos de **clustering** en Machine Learning con Python:

- ☐ Sesión (19)
  - **22/01/2024**
  - Aprendizaje No-Supervisado en Python
- ☐ Sesión (20)
  - **29/01/2024**
  - Ingeniería de variables (Feature engineering)
- ☐ Sesión (21)
  - **05/02/2024**
  - Centroid-based Clustering (K-Means , Mean-Shift & Mini-Batch K-Means)
- ☐ Sesión (22)
  - **19/02/2024**
  - Density-based clustering (DBSCAN, OPTICS)
  - Distribution-based clustering (GMM)
  - Hierarchical clustering (Agglomerative Clustering)
- ☐ Sesión (23)
  - **26/02/2024**
  - Aplicar un caso de uso real
  - **Cierre del bloque (IV)**

## V- Análisis de **Series Temporales** con Python :

- ☐ Sesión (24)
  - **04/03/2024**
  - Time Series Forecasting en Python
- ☐ Sesión (25)
  - **11/03/2024**
  - Modelos estadísticos (AR, ARIMA, SARIMA, Exponential Smoothing)
- ☐ Sesión (26)
  - **18/03/2024**
  - Recursive Forecasting (Random Forest, Gradient Boosting Regression)
- ☐ Sesión (27)
  - **08/04/2024**
  - Multivariate Forecasting, Ensemble modeling

- ☐ Sesión (28)
  - **15/04/2024**
  - Aplicar un caso de uso real
  - **Cierre del bloque (V)**

- ☐ Sesión (29)
  - **22/04/2024**
  - Presentación de los **informes** del **proyecto final**
- ☐ Sesión (30)
  - **29/04/2024**
  - Presentación de los **códigos** del **proyecto final (I)**
- ☐ Sesión (31)
  - **06/05/2024**
  - Presentación de los **códigos** del **proyecto final (II)**

# Composición de la nota final



CES  
Juan Pablo II  
MADRID



Cámara de Comercio Alemana  
para España  
Deutsche Handelskammer  
für Spanien



## Puntuación del módulo 1 – Composición de la nota final

Conceptos	Puntos	Porcentajes
Sesiones formativas	3 puntos	$23 \times 3 = \mathbf{69}$
Sesiones del cierre (bloques I, II, III, IV, V)	5 puntos	$5 \times 5 = \mathbf{25}$
Proyecto final – Presentación del informe	5 puntos	$5 + 10 = \mathbf{15}$
Proyecto final – Código desarrollado	10 puntos	
<b>Nota final</b>		<b><math>109 &gt;&gt;&gt; 10 + 0.9</math></b>

# Instalación de Anaconda



**CES**  
Juan Pablo II  
MADRID



Cámara de Comercio Alemana  
para España  
Deutsche Handelskammer  
für Spanien

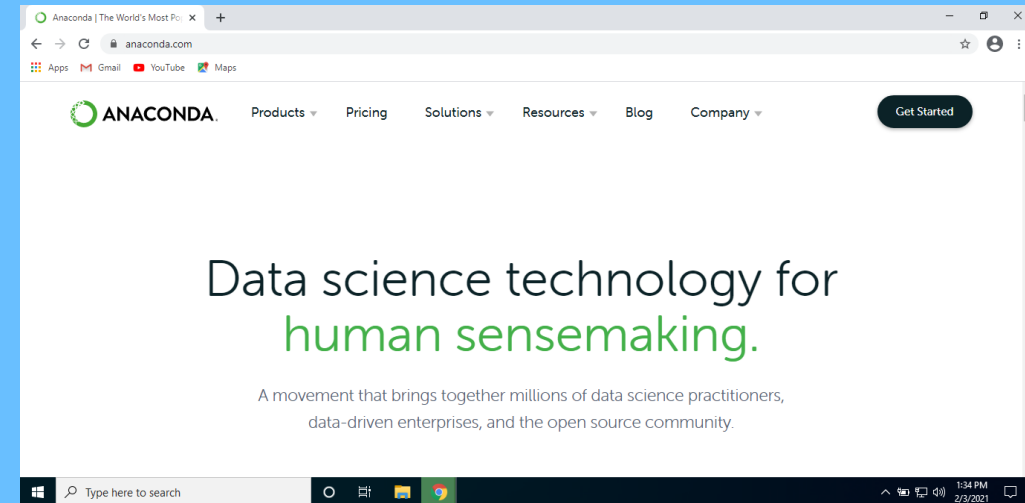
## El requisito principal para la sesión formativa de Python es Instalar la distribución de Anaconda a través de la siguiente dirección:



<https://www.anaconda.com/>

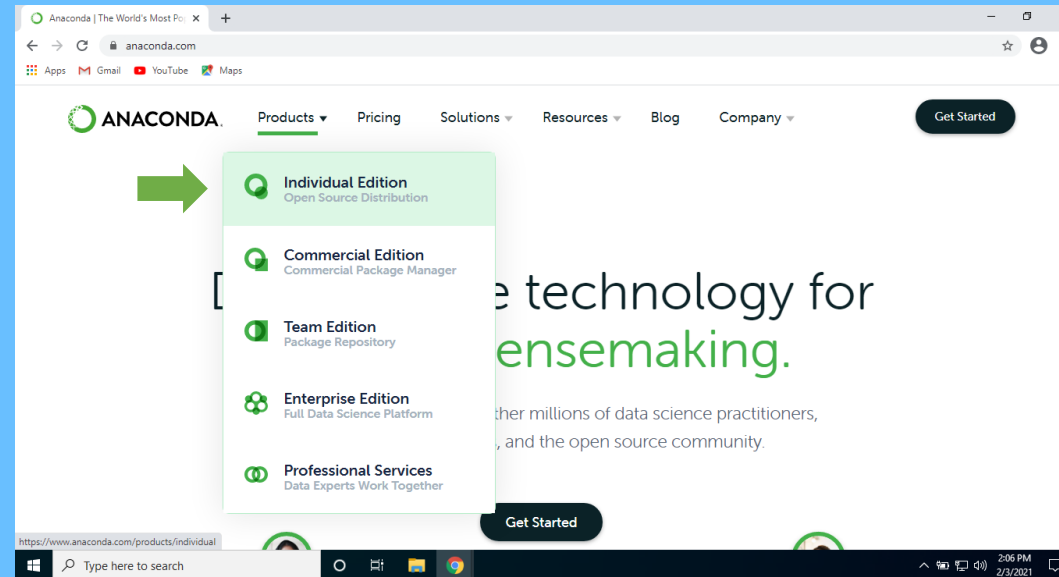


- Anaconda es una suite de código abierto o **open-source** que nos facilita la instalación y la gestión de los desarrollos con Python
- Todos los equipos del centro deberían tener **instalado este software sin problema** para las sesiones del módulo 1

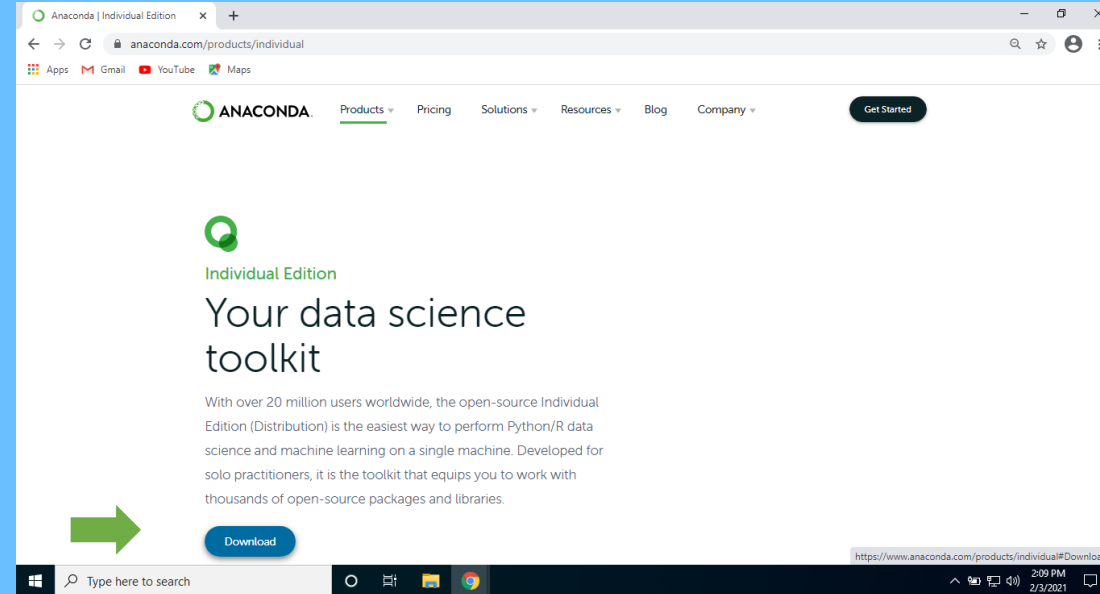


A continuación, en las siguientes slides se muestran una serie de pantallazos para explicar el paso a paso necesario a seguir...

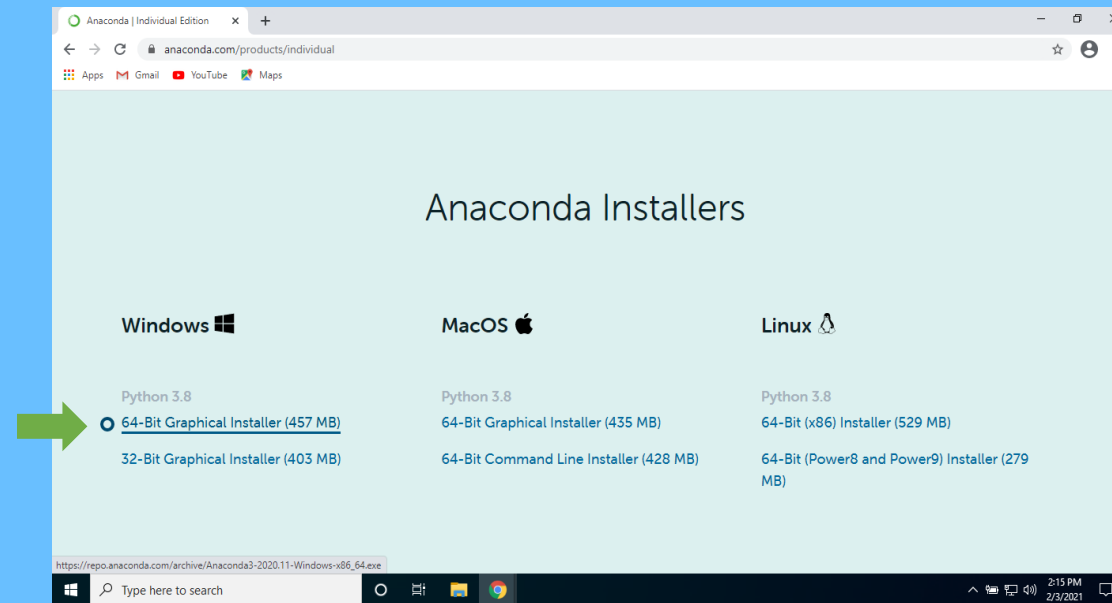
## 1. Se selecciona desde Products la versión de Individual Edition:



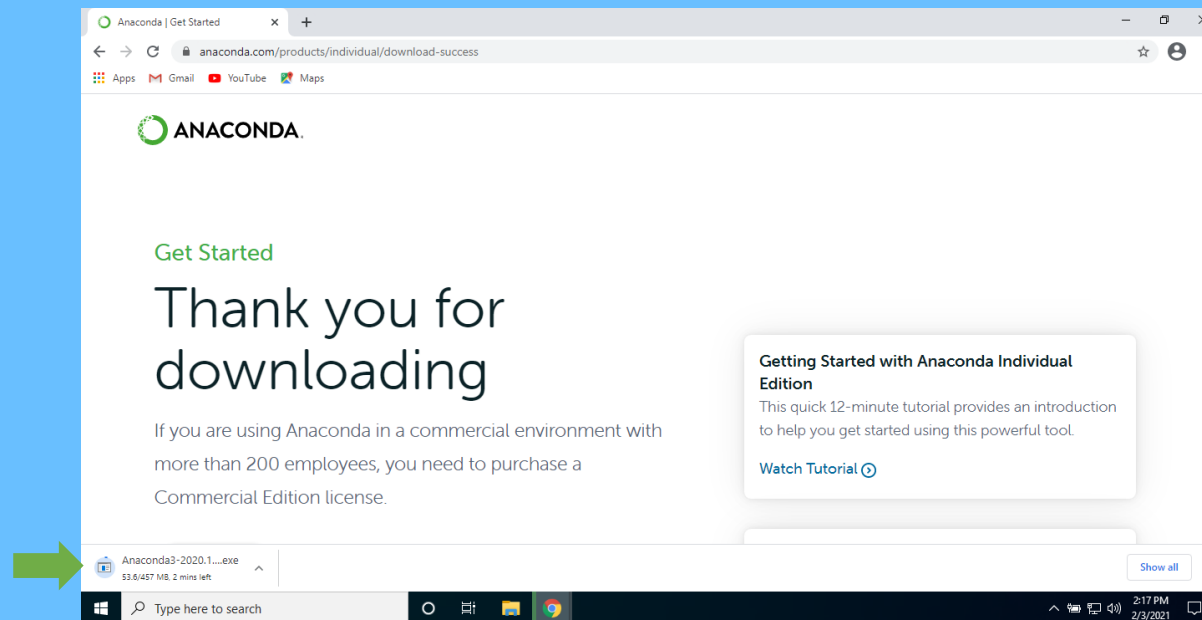
## 2. Se procede a descargar la suite individual de anaconda:



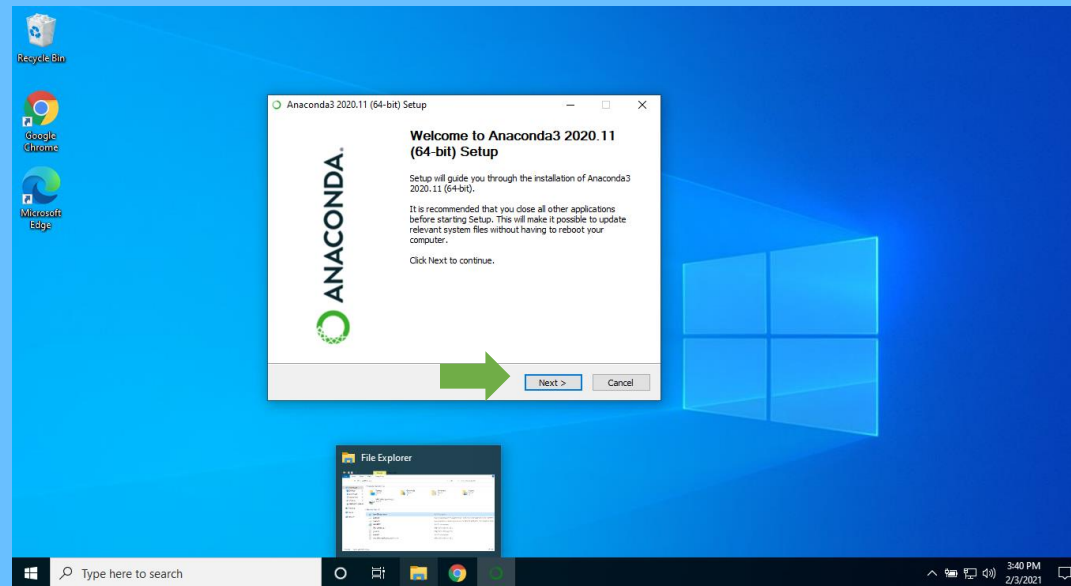
### 3. Se selecciona la opción de Windows 10 (64bits) si es el caso del equipo correspondiente:



## 4. Se completa la descarga del fichero instalable (.exe) de anaconda:

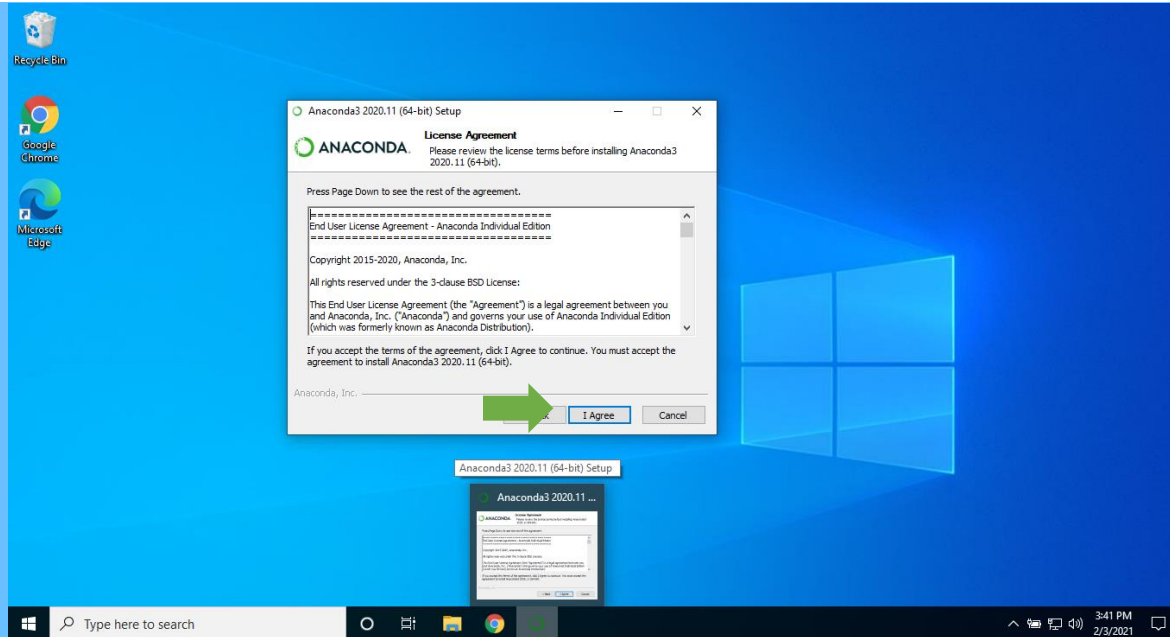


## 5. Se ejecuta el fichero (.exe) descargado anteriormente para empezar con el proceso de instalación:

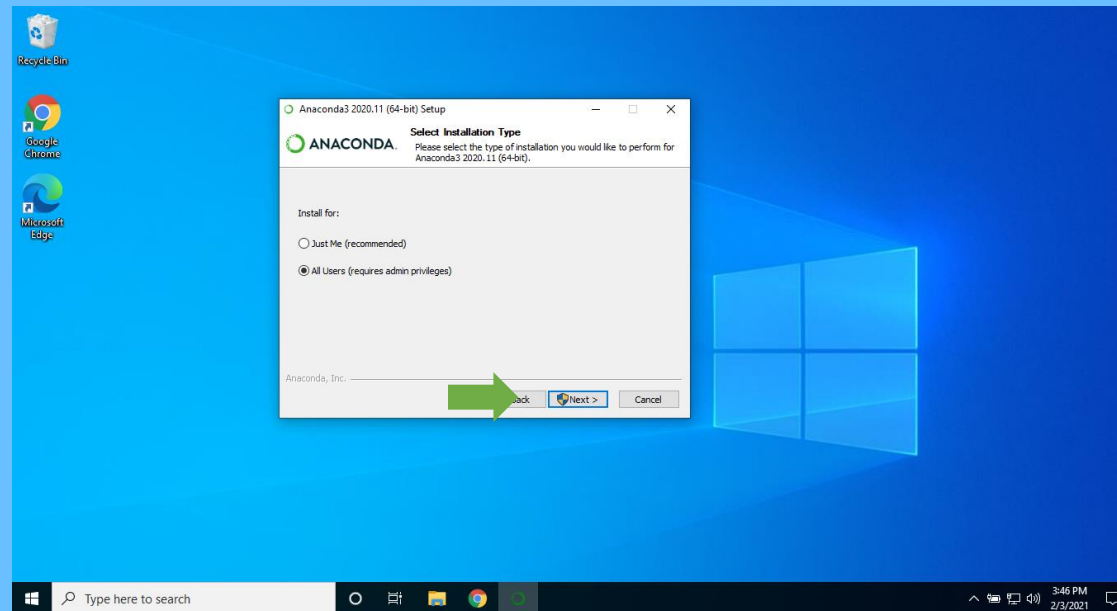




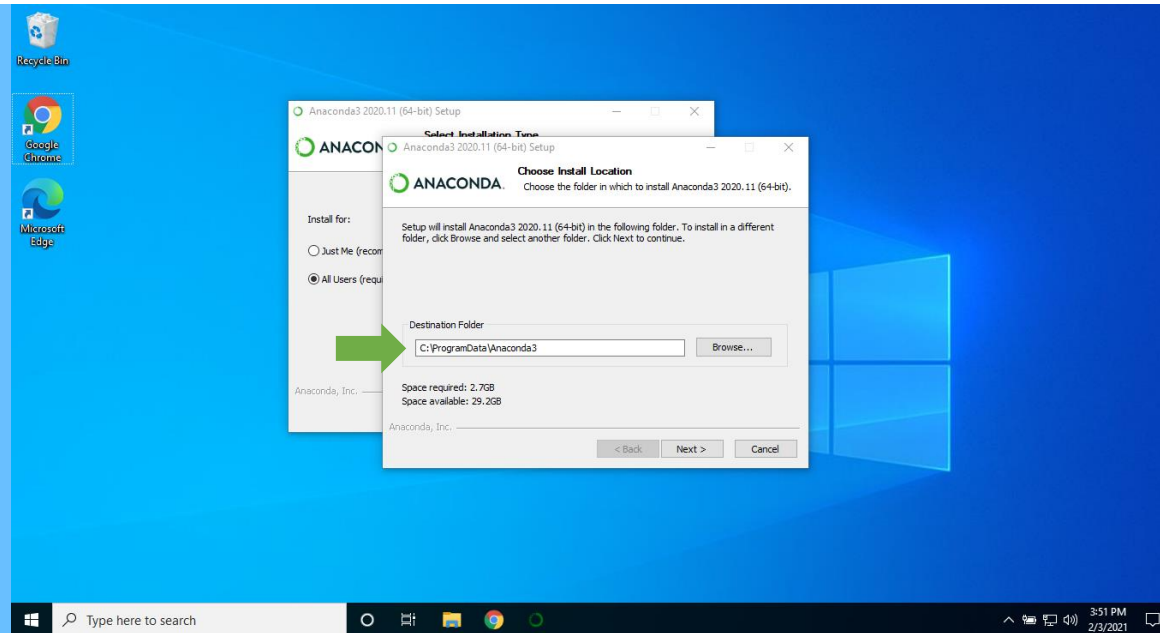
## 6. Se aceptan los términos y las condiciones a continuación:



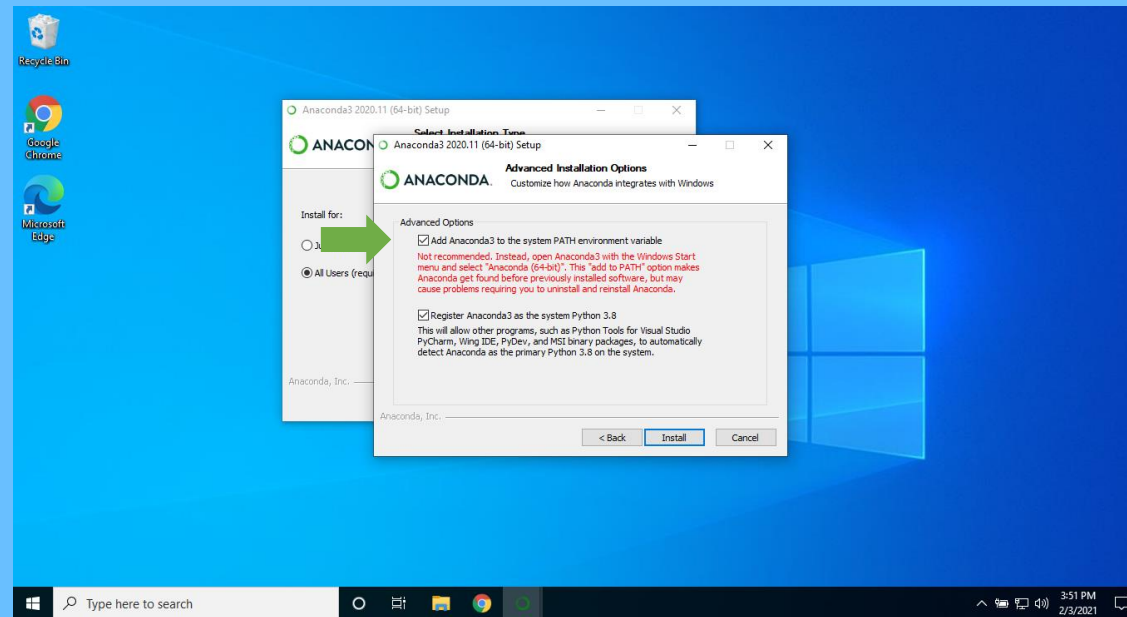
## 7. Se elige el usuario actual o todos los usuarios de la máquina para instalar el software según es el caso:



## 8. Se selecciona la carpeta de destino para instalar (recomendable seguir con la ruta por defecto) y continuar pulsando “Next>”:

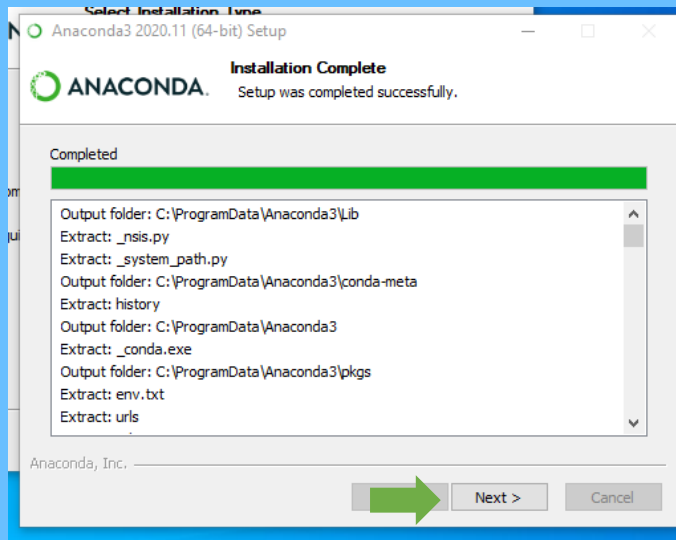


## 9. Se habilita antes de instalar, la opción de añadir Anaconda a las variables del entorno de sistema:

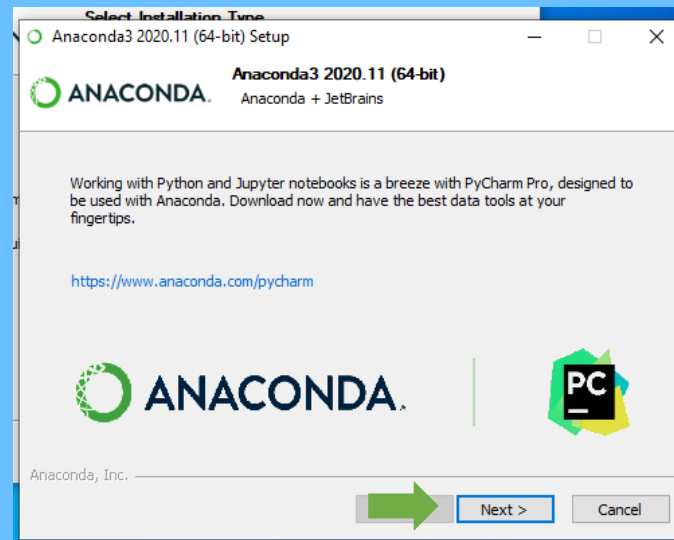


## 10. Finalmente, tras los siguientes tres pasos, se instala la suite completa de Anaconda

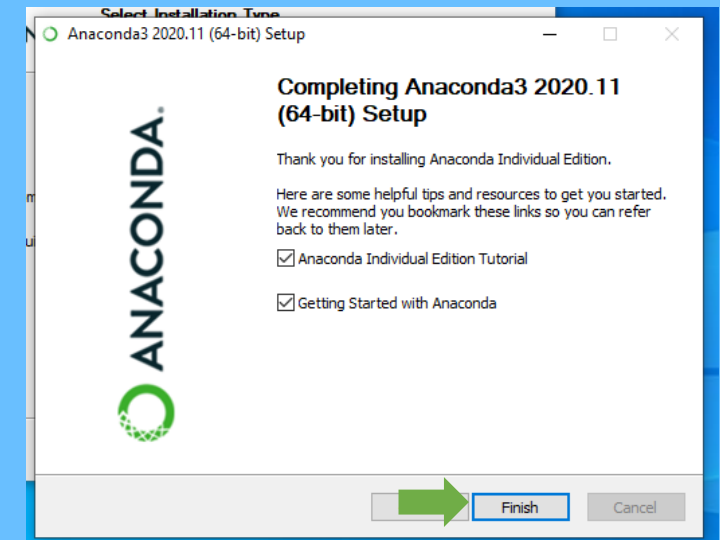
1



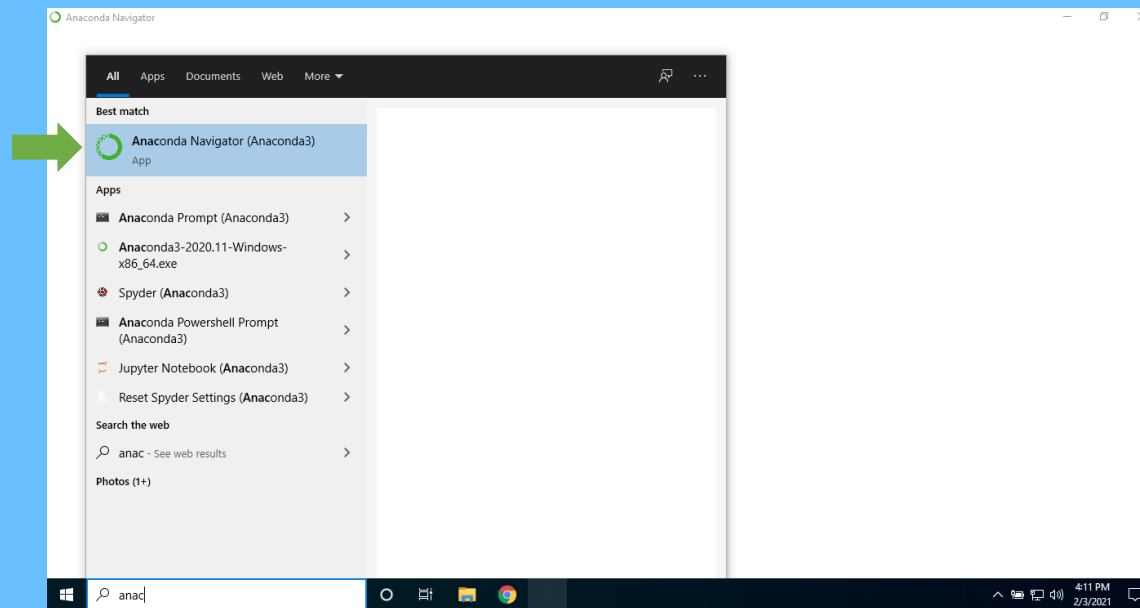
2



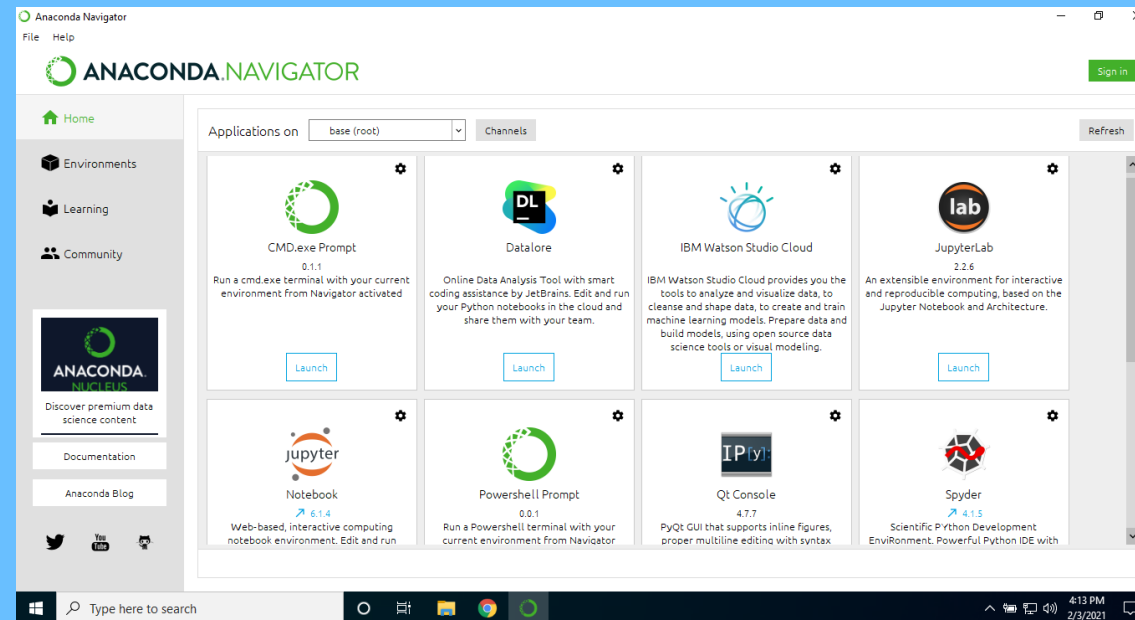
3



**Después de terminar la instalación correctamente, se puede acceder a la suite buscando Anaconda en el buscador**

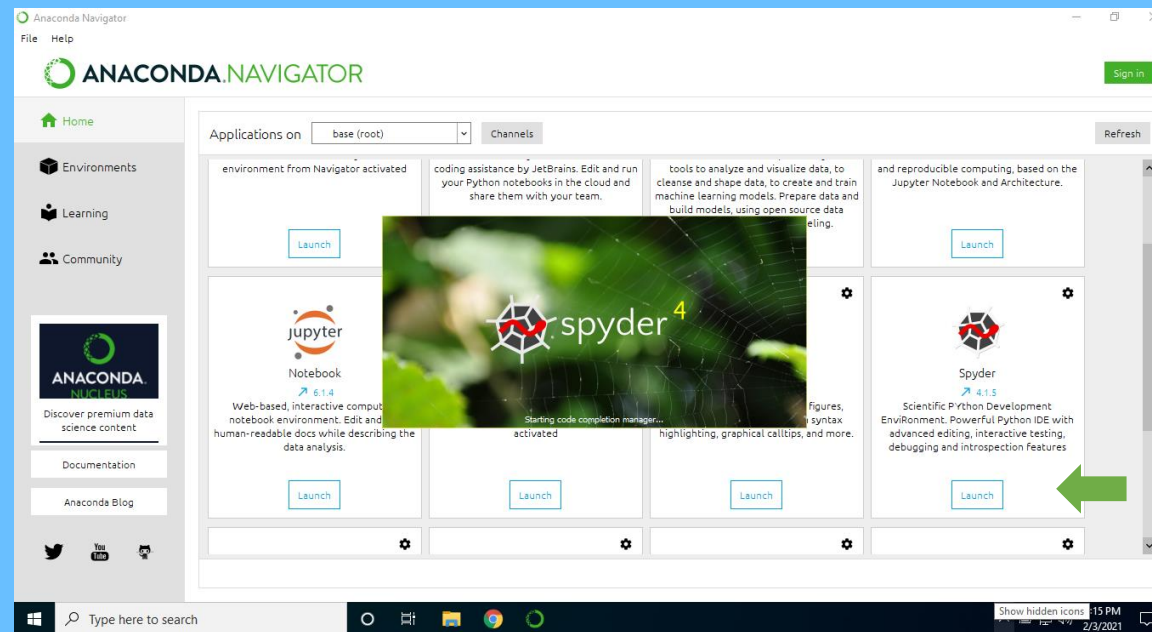


**Una vez abierto, aparece el Anaconda Navigator, es una interfaz de usuario que nos permite acceder a diferentes aplicaciones de anaconda**



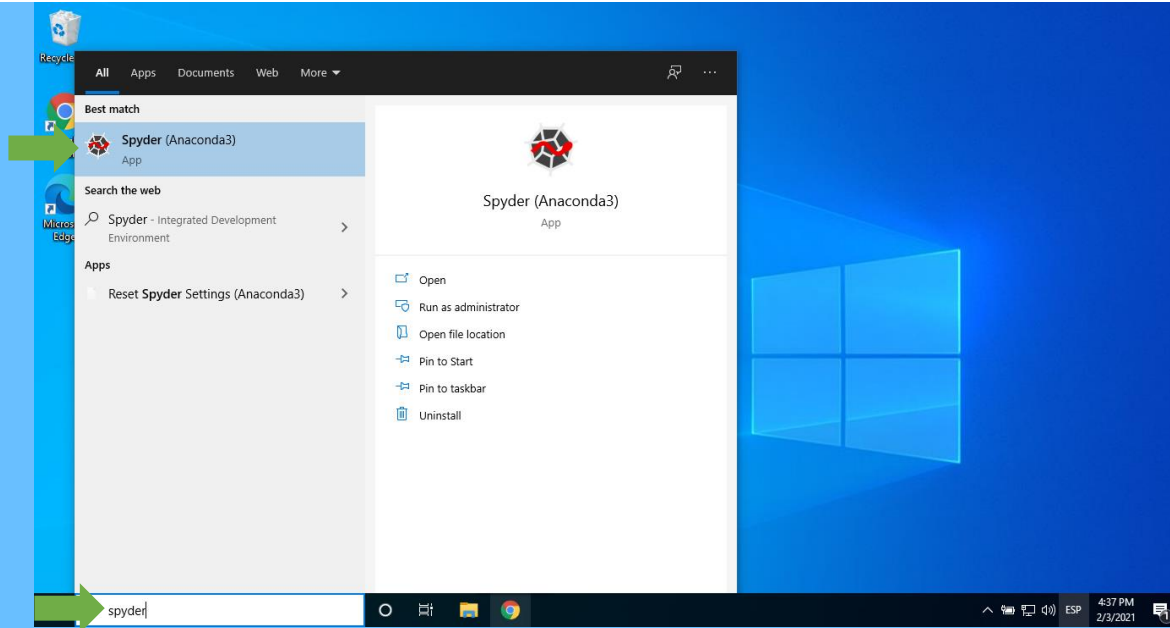
## INSTALACIÓN

**Desde Anaconda Navigator, podemos abrir uno de los entornos de desarrollo de Python llamado Spyder**





**Además, se puede acceder también de forma directa a muchas de las aplicaciones instaladas por Anaconda, como por ejemplo Spyder, del siguiente modo:**





**CES**  
Juan Pablo II  
MADRID



Cámara de Comercio Alemana  
para España  
Deutsche Handelskammer  
für Spanien



# ¡Gracias!

## Contacto:

- Email: [elias.nezamolesslami@cesjuanpablosegundo.es](mailto:elias.nezamolesslami@cesjuanpablosegundo.es)
- Linked-In: <https://www.linkedin.com/in/elias-nezamolesslami>