

# Enseñanza del aprendizaje automático (ML) aplicado a Transferencia de calor

Proyecto PAPIME PAPIME PE103422

Dr. Michel Rivero, IIM-UNAM

Dr. Heberto Ferreira Medina, IIES-UNAM

Dr. Sergio Rogelio Tinoco Martínez, ENES-UNAM

**Escuela Nacional de Estudios Superiores, Unidad Morelia. 2022**

Actividades sincrónicas Zoom:

Link de zoom: <https://cuaieed-unam.zoom.us/j/87963800329?pwd=WDhKVGFqUzdFc3NxbWY4T1lwZ2ZtUT09>

Link de classrom: <https://classroom.google.com/c/NTQ3NTYxMDU1MTA2?cjc=7armzty>

# Información del curso:

- 9 sesiones síncronas
- 1 sesión asíncrona
- 8 actividades de aprendizaje
- 15 puntos mínimo para aprobar
- Primeras cuatro sesiones lenguaje Python

# Actividades sesión 1.

- Evaluación diagnóstica.
- Plataforma del curso en línea o a distancia.
- Herramientas de software:
  - 1) Python; instalar
  - 2) Jupyter-Lab; instalar
  - 3) Matplotlib; importar
  - 4) Numpy; instalar e importar
  - 5) Pandas; instalar
  - 6) Scikit-Learn; instalar
- Equipo de cómputo y conexión a Internet.

# Lenguaje Python.

- I. Generalidades
- II. Instalación de herramientas
- III. Guía del lenguaje
- IV. Flujo de control
- V. Números, strings, listas, tuplas, conjuntos
- VI. Funciones
- VII. Listas de comprensión
- VIII. Biblioteca numpy – broadcasting (sesión 2)
- IX. Biblioteca Pandas
- X. Biblioteca Matplotlib

# I. Generalidades.



- Es un lenguaje interpretado desarrollado en los 90's por Guido Van Rossum en los laboratorios CWI de Holanda, que ahora alberga a la W3C
- Lenguaje ampliamente utilizado por su versatilidad
- IEEE lo coloca como el lenguaje de programación más utilizado (2021)

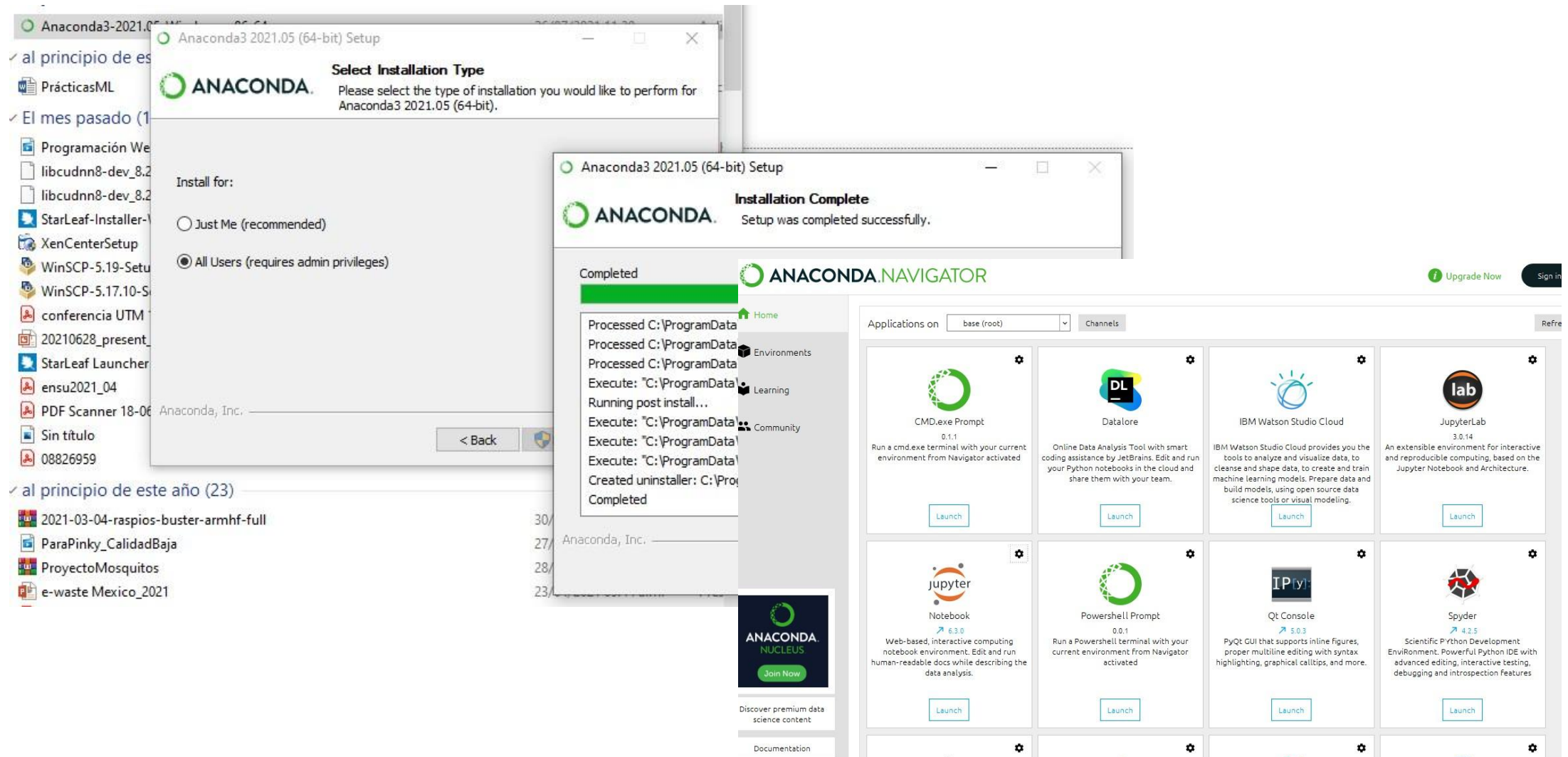


Rank	Language	Type	Score
1	Python ▼	🌐 🖥 ⚙	100.0
2	Java ▼	🌐 📱 🖥	95.3
3	C ▼	📱 🖥 ⚙	94.6
4	C++ ▼	📱 🖥 ⚙	87.0
5	JavaScript ▼	🌐	79.5
6	R ▼	🖥	78.6

## II. Instalación de herramientas.

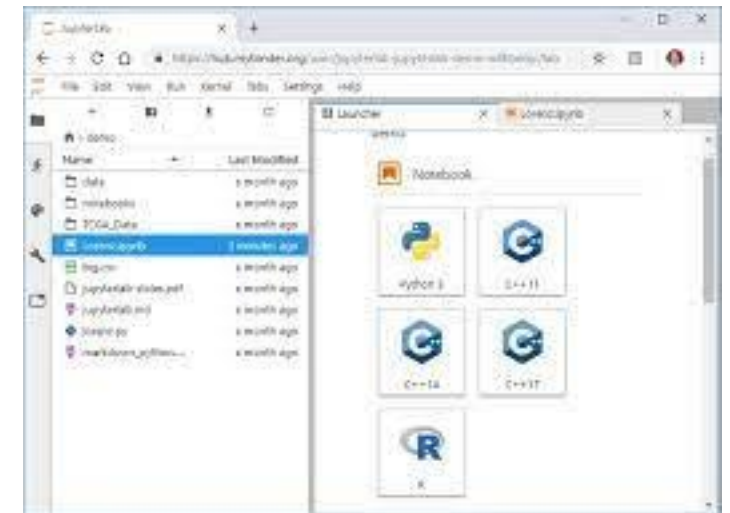
- Anaconda (entorno para integración de herramientas):
  - Windows:
    - <https://www.anaconda.com/products/individual#windows>
    - Ejecutar instalador
    - Anaconda-Navigator
  - Linux (debian):
    - `sudo apt-get install python3 python3-pip ipython3`
    - `pip3 install --upgrade pip`
- `conda install jupyter matplotlib numpy pandas scikit-learn`

# II. Anaconda Windows.



# Jupyter Lab.

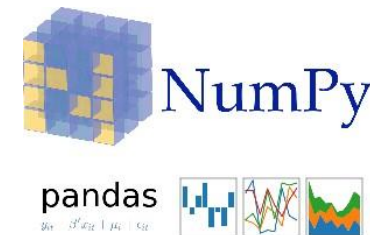
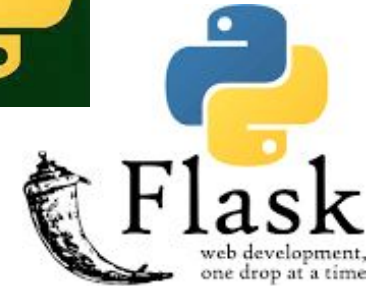
- Python requiere de varias herramientas (en teoría solo requiere de un editor de textos),
- Anaconda (plataforma para ciencia de datos) y jupyter lab, se han convertido en las mejores herramientas para desarrollo con Python; además de las bibliotecas para la IA.
- Puede utilizarse como un lenguaje estructurado u orientado a objetos (POO).
- Python.org es el sitio Web que promueve su uso y enseñanza.





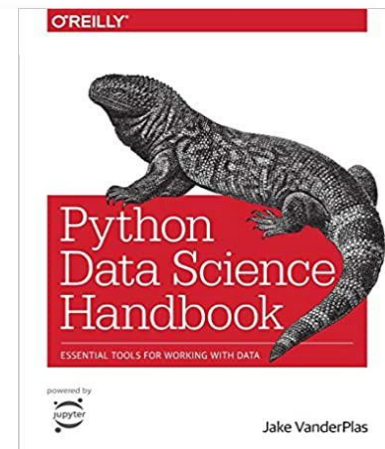
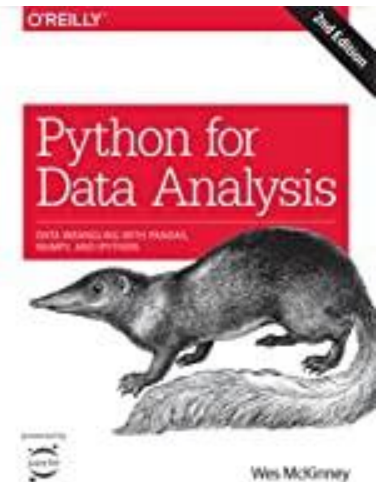
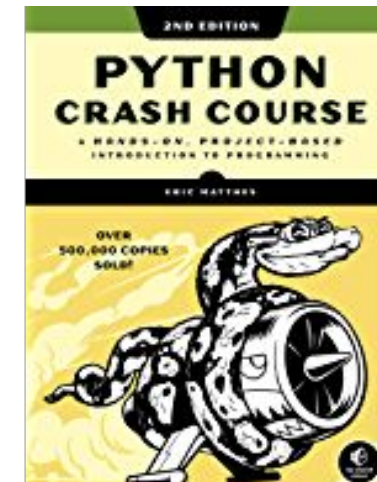
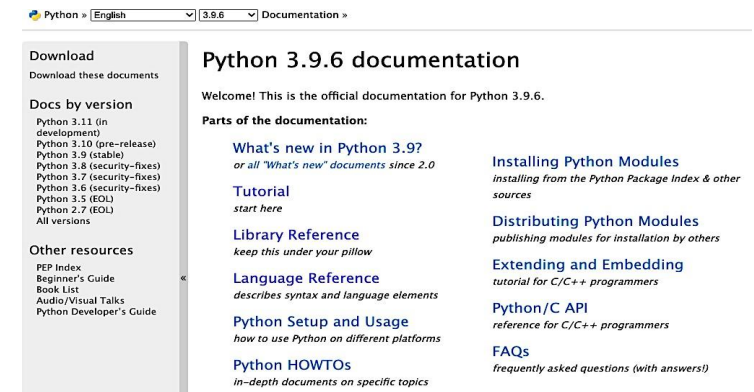
# III. Guía del lenguaje.

- Python junto con los frameworks Web; Django y Flask, son de las herramientas más utilizadas para el desarrollo Web.
- En ciencia de datos los frameworks más utilizados son; 1) TensorFlow, 2) Scikit-Learn, 2) Keras, 3) Theano, 4) SciPy, entre otras.
- Numpy, Pandas y matplotlib son bibliotecas de apoyo que se utilizan ampliamente en el ML y DL



# III. Guía del lenguaje.

- La guía completa se puede localizar en.
  - <https://docs.python.org/3/>
  - <https://docs.python.org/3/tutorial/index.html>
  - <https://docs.python.org/3/library/index.html>
- La guía de referencia del lenguaje:
  - <https://docs.python.org/3/reference/index.html>
- Los libros más recomendados son:
  - Crash Course. Eric Matthes
  - Data Analysis. Wes McKinney
  - Hands-on Machine Learning. Aurélien Géron
  - Data Science Handbook. Jake VanderPlas



## IV. Flujo de control.

- Python utiliza la indentación como forma de anidar instrucciones, debe preservarse (4 espacios o tabulador),
- El `#comentario`
- Declaración de variables: `variable = valor`
- Números: `entero = 0`    `real = 1.5e-1`    `complejo = 1.0 + 2j`
- Declaración de strings: `cadena = 'Python'`
- Declaración de funciones: `def funcion():`  
`print("Hola desde función")`
- Estructuras de datos:
  - `lista = [1, 4, 9, 16, 25]`
  - `tupla = (12345, 54321, 'hello!')`
  - `conjunto = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}`
  - `diccionario = {'jack': 4098, 'ana': 4139}`

# Más Flujo de control; if, while, for, iter

- **If ... else**

```
a = 200
b = 33
if b > a: print("b mayor que a")
elif a == b: print("a y b son iguales")
else: print("a mayor que b")
```

- **While**

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```

- **for**

```
words = ['cat', 'window', 'pc']
for w in words:
    print(w, ":", len(w))
```

- **Iter**

```
tupla = ("apple", "banana", "cherry")
myit = iter(tupla)
print(next(myit))
print(next(myit))
print(next(myit))
```

# Clases para POO.

- Python soporta la programación orientada a objetos
- Se pueden definir clases, atributos, objetos y métodos
- Se pueden encapsular datos y utilizarlos de forma segura

#ejemplo de clase

```
class Persona:
```

```
    nombre = ''
```

```
    edad = ''
```

```
    def __init__(self, nombre, edad):
```

```
        self.nombre = nombre
```

```
        self.edad = edad
```

```
class Direccion(Persona):  
    pass
```

#ejemplo de uso de la clase en un objeto

```
p1 = Persona("Juan", 36)
```

```
print(p1.nombre)
```

```
print(p1.edad)
```

# Ejercicios con jupyter Notebook.

- I. Números, strings, listas, tuplas, conjuntos
- II. Funciones
- III. Listas de comprensión

# Ejercicios primera sesión:

1. Escriba una función **Sumatoria()** que tome una **Lista** de números y devuelva la suma acumulada.
2. Escriba una función llamada **Elimina(Lista)** que tome una **Lista** y elimine el **primero** y **último** elemento de la Lista y cree una **nueva lista** con los elementos que “no fueron eliminados”.
3. Escriba una función **Ordenada(Lista)** que tome una lista como parámetro y devuelva **True** si la lista está ordenada en **orden ascendente** y devuelva **False** en caso contrario.
4. Escriba una función que calcule la aproximación a la **raíz cuadrada** de un número **n** Raiz(n). **(Tarea 1 para entrega)**