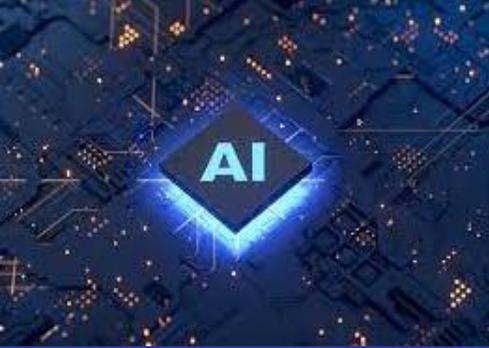


# NumPy - Tópicos avanzados

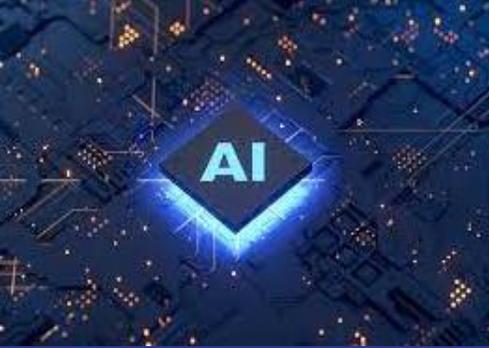




## Definición

- Es una librería de Python que ofrece soporte para **arreglos multidimensionales y matrices**, además de incluir una variedad de **funciones matemáticas para realizar cálculos de manera eficiente con estos arreglos**.
- Ha ganado gran popularidad en el ámbito de la **ciencia de datos, el aprendizaje automático y diversas otras aplicaciones de análisis numérico y científico**.

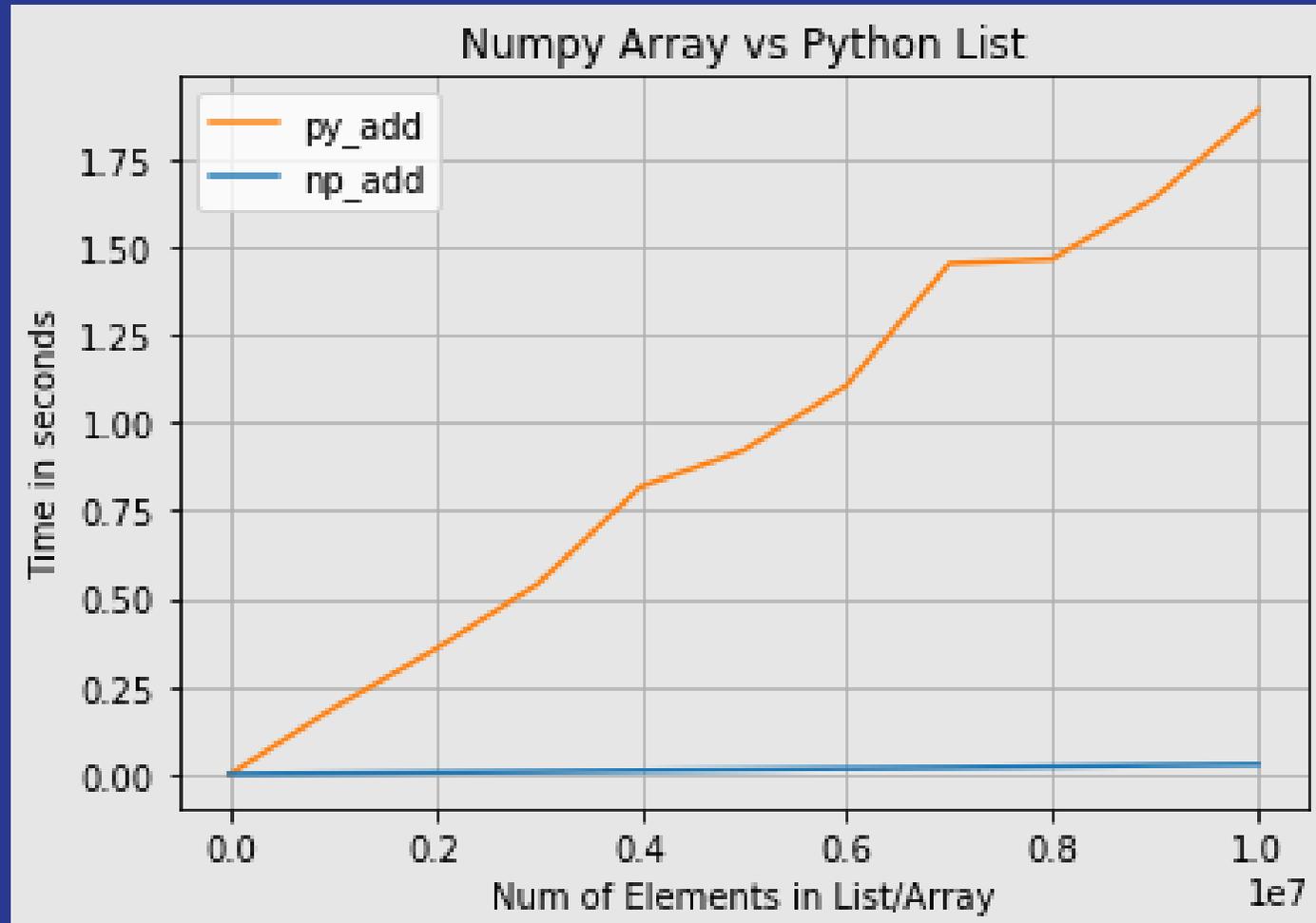
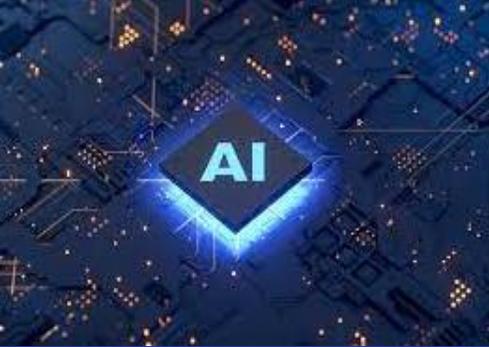


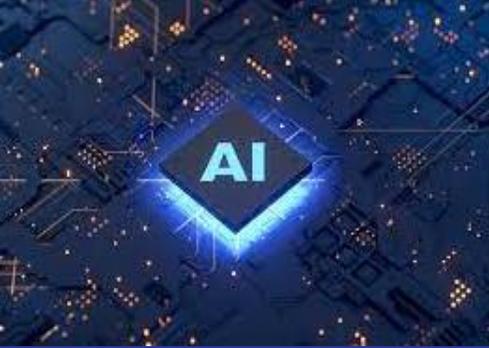


## Ventajas

- Su eficiencia en el manejo de datos se debe a las **operaciones vectorizadas**, que permiten realizar **cálculos rápidos en arrays multidimensionales**.
- La biblioteca proporciona una amplia gama de funciones **matemáticas y estadísticas avanzadas**, **facilitando el análisis de datos complejos**.
- Su estructura **ndarray** permite trabajar con datos en **múltiples dimensiones de manera efectiva**.
- Se integra bien con otras bibliotecas del ecosistema científico de Python, como **OpenCV, SciPy, Pandas y Matplotlib**.



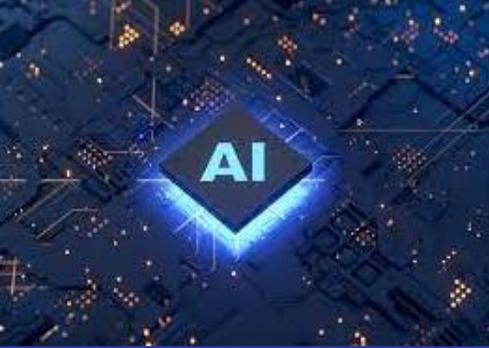




# Veamos las ventajas de usar NumPy vs Python común

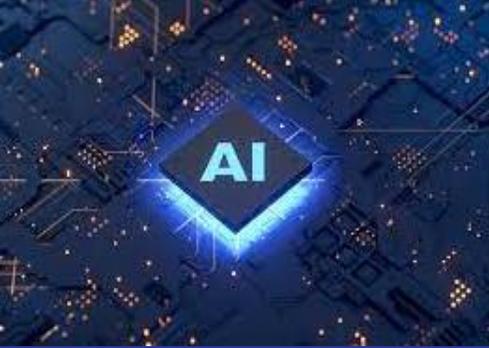
Vamos a realizar el análisis sobre un código ya creado





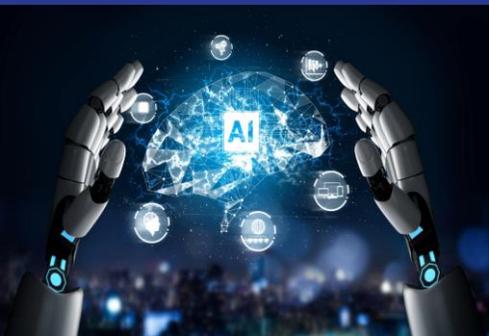
# Arreglos n- dimensionales

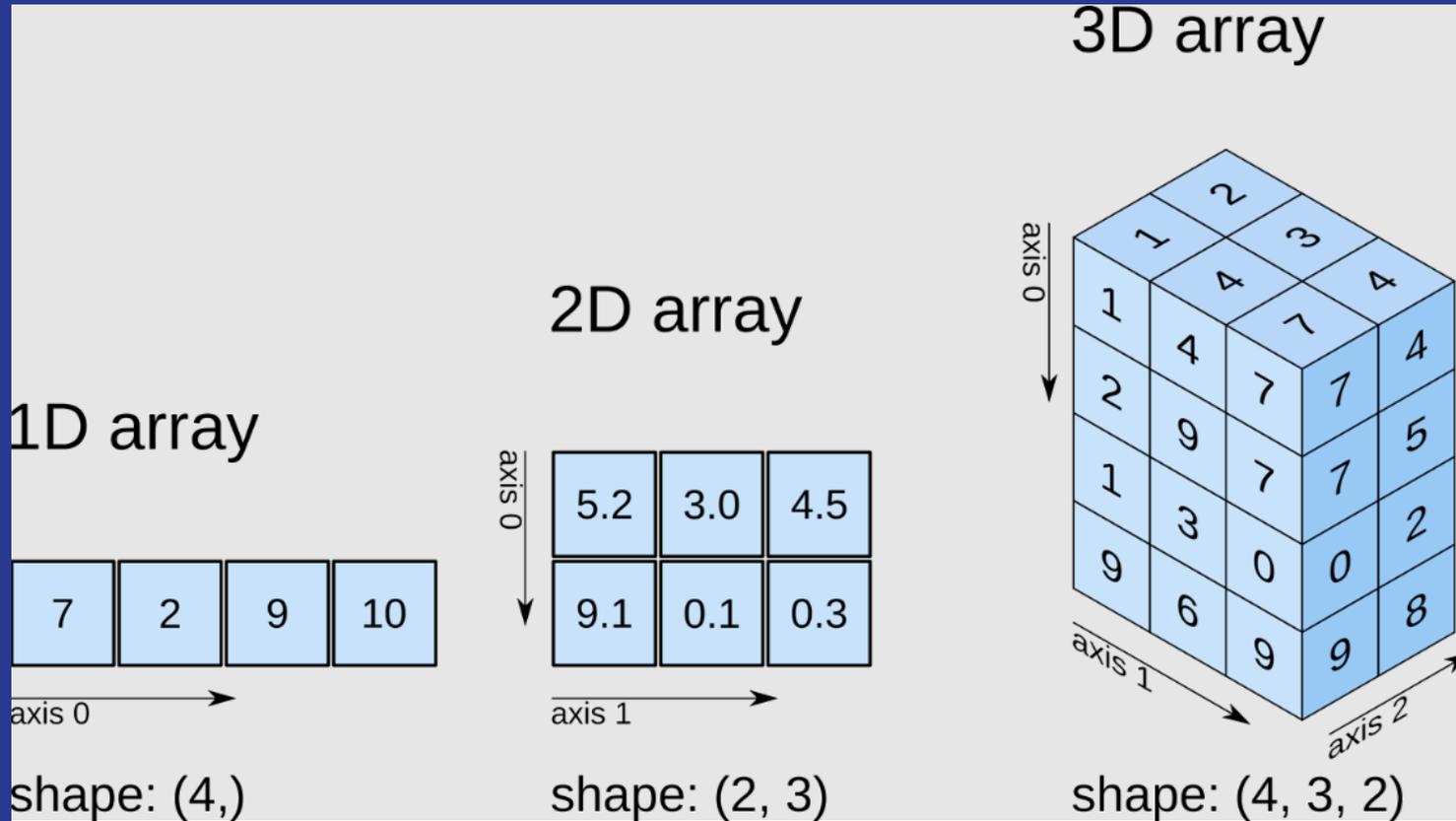
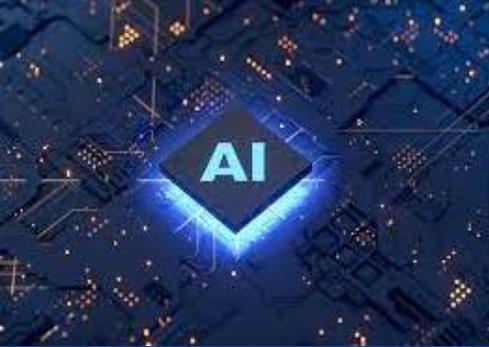


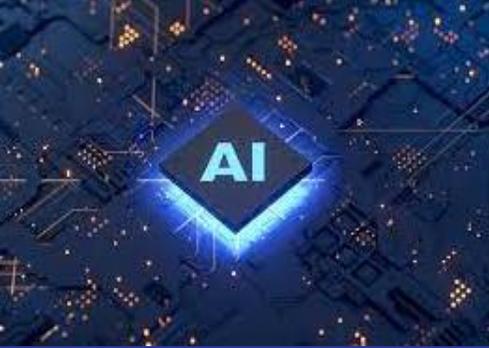


## Definición

- NumPy nos permite trabajar con arrays n-dimensionales de forma sencilla y eficiente.
- En cálculos científicos es muy común manejar operaciones con arreglos n-dimensionales, por lo cual NumPy se convierte en una opción bastante accesible.
- Casi todos los algoritmos de aprendizaje automático y profundo **utilizan internamente arrays n-dimensionales.**



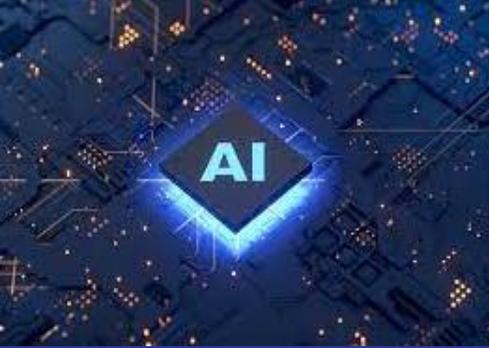




# Código práctico en Python

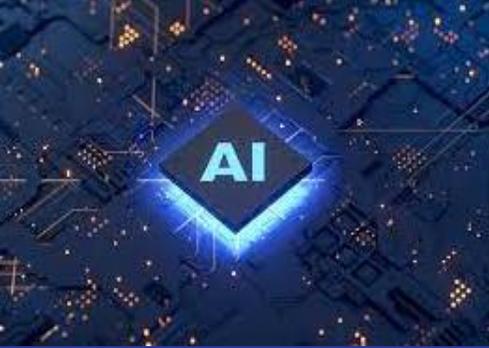
¡Vamos a probar!





# Funciones matemáticas y estadísticas

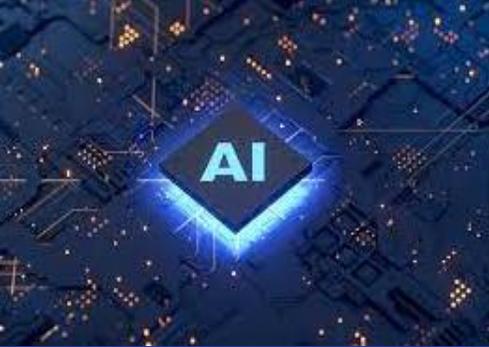




## Definición

- NumPy nos provee un montón de funciones matemáticas y estadísticas
- Tenemos desde operaciones numéricas simples hasta otras más avanzadas, incluyendo cálculo, estadística, álgebra lineal, etc.

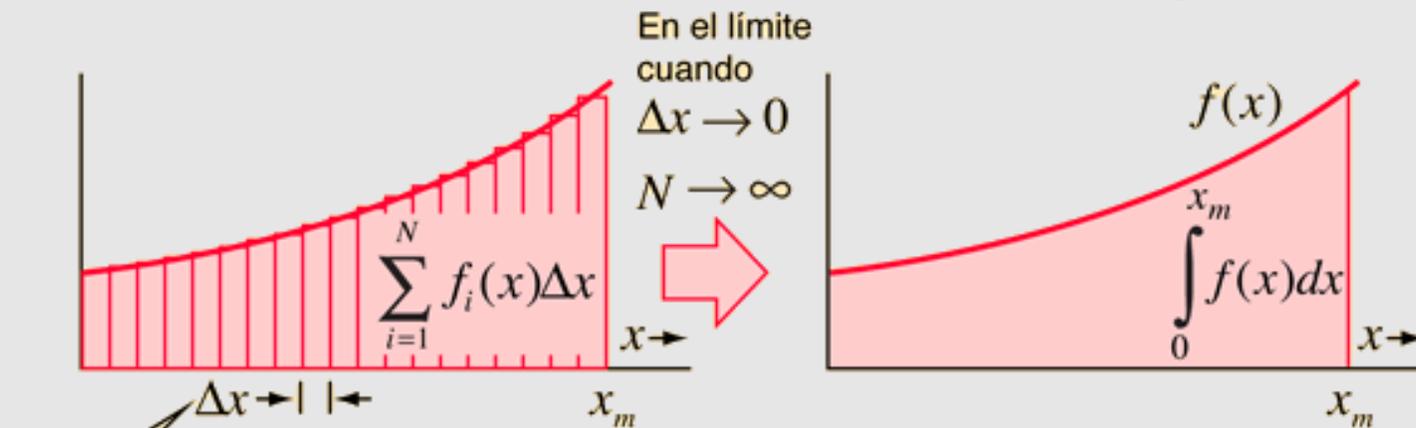




# Integral definida

NumPy nos permite realizar operaciones numéricas para obtener el área bajo la curva de una función.

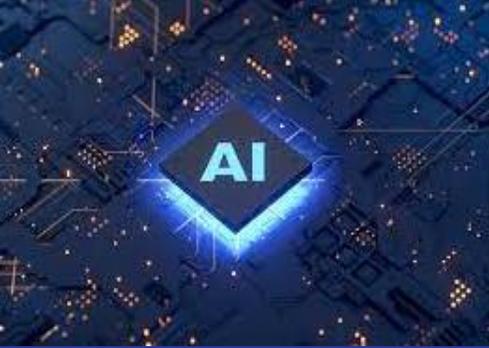
*La suma se convierte en la integral*



$$\frac{x_m}{N} = \Delta x$$

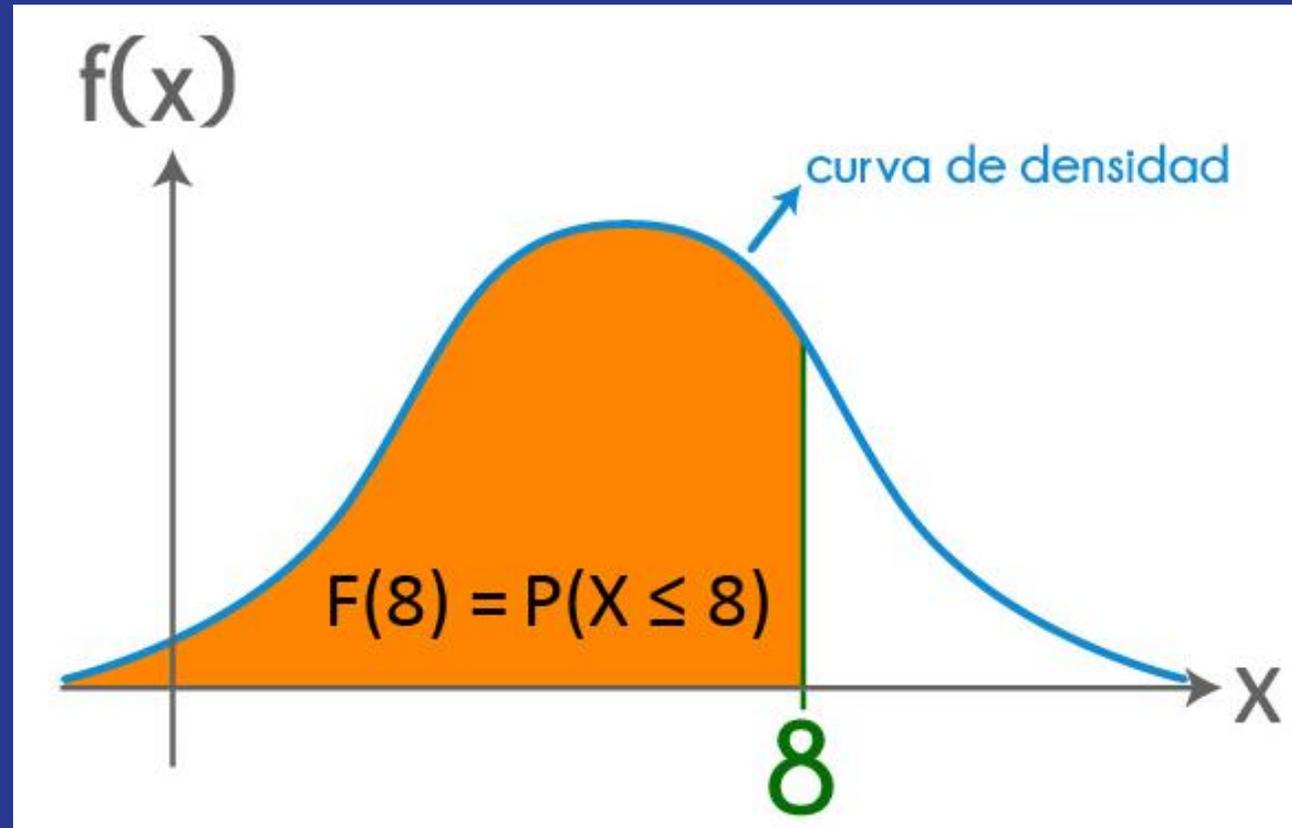
$$\text{Área} = \int_0^{x_m} f(x)dx = \lim_{\Delta x \rightarrow 0} \sum_{i=1}^N f_i(x)\Delta x$$

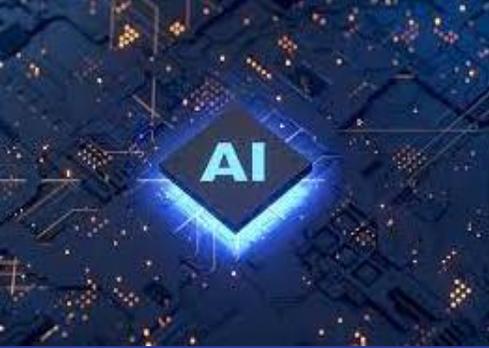




# Función de densidad

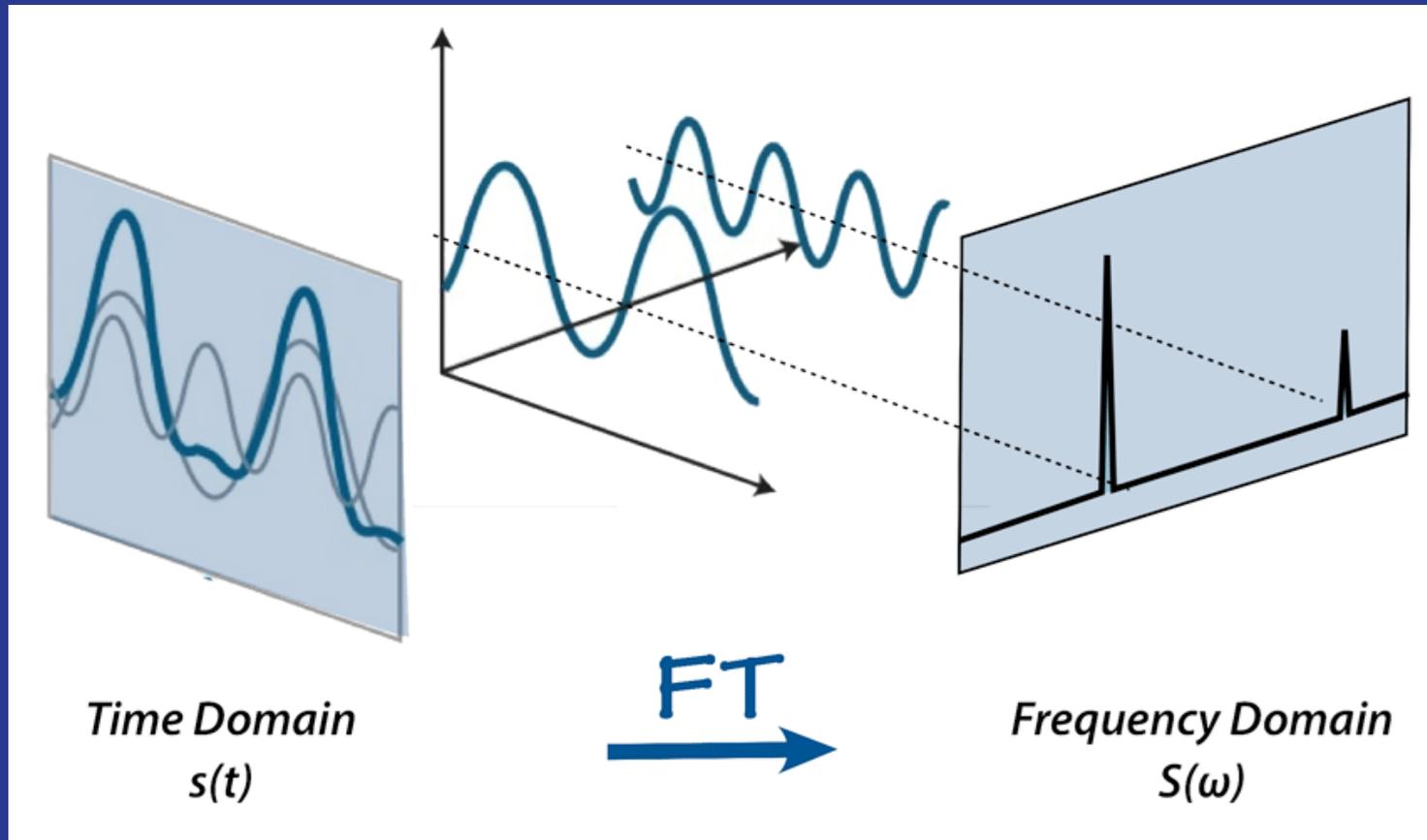
Un concepto estadístico elemental que indica que el área bajo la curva de una función de densidad continua representa la probabilidad. Muchos algoritmos de aprendizaje automático y profundo nacen desde la génesis de la estadística.

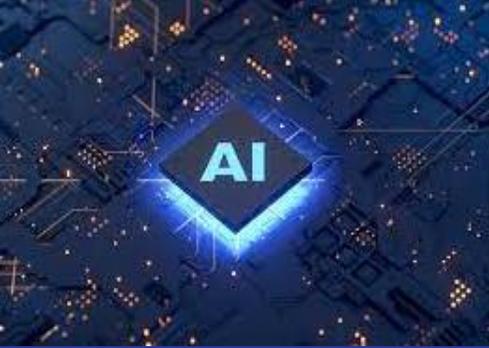




# Cambio de dominios

El cambio de dominios es algo muy común en la práctica y los algoritmos de AI no son la excepción, sobre todo en el contexto de los arrays n-dimensionales.

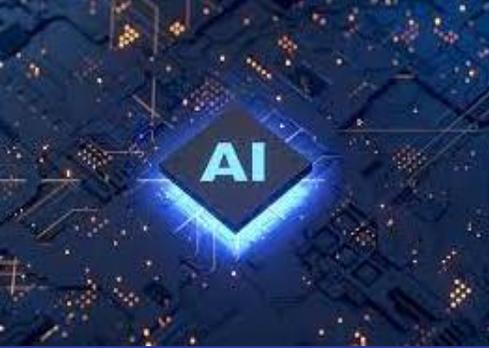




# Código práctico en Python

¡Vamos a probar!



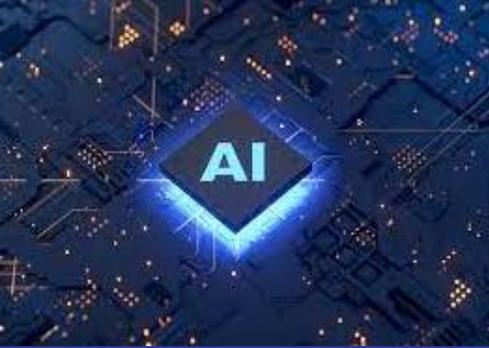


## ACTIVIDAD

- 1) Cree un programa que determine la inversa de una matriz, tenga en cuenta las restricciones correspondientes. Si no es posible calcular la inversa, muestre un mensaje justificando el por qué no se puede.

**PUEDE USAR CUALQUIER MÓDULO DE NUMPY, NO HAY RESTRICCIONES.**



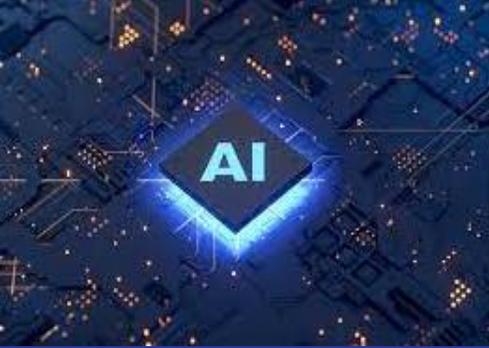


2) Cree un programa que haga lo siguiente:

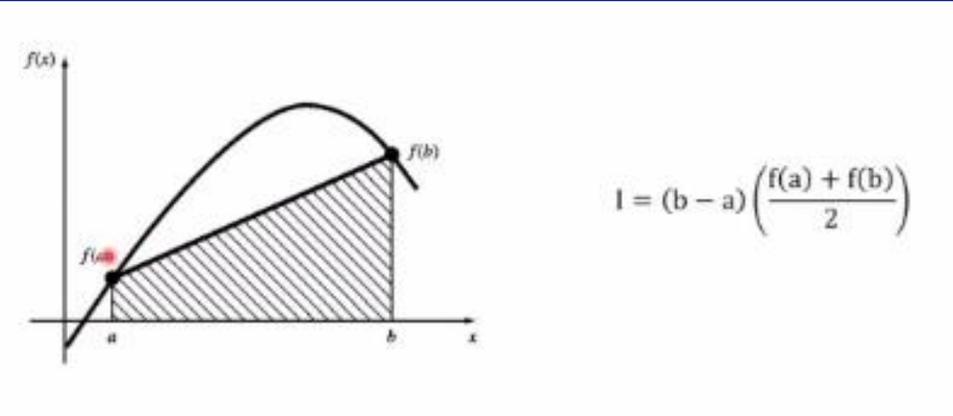
- a) Defina una función que reciba como parámetros los valores  $x$  de una función de distribución de probabilidades y devuelva esa función de distribución como el seno de esos valores  $x$ .
- b) Calcule  $P(0 \leq X \leq \pi)$
- c) Grafique la función de distribución con su área pintada del color que usted desee.

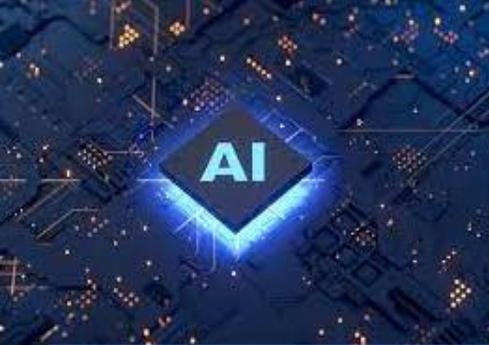
- **PARA EL LITERAL b) UTILICE LA REGLA DEL TRAPECIO (REVISAR LIBRO CÁLCULO DE UNA VARIABLE DE CLAUDIO PITA RUIZ U OTRAS FUENTES CONFIABLES)**
- **PARA EL LITERAL c) INVESTIGUE SOBRE EL USO DE LA LIBRERÍA MATPLOTLIB**





# Regla del trapecio

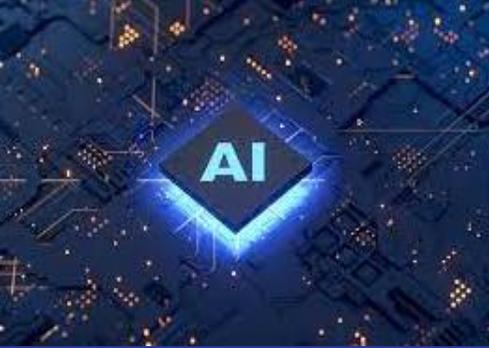




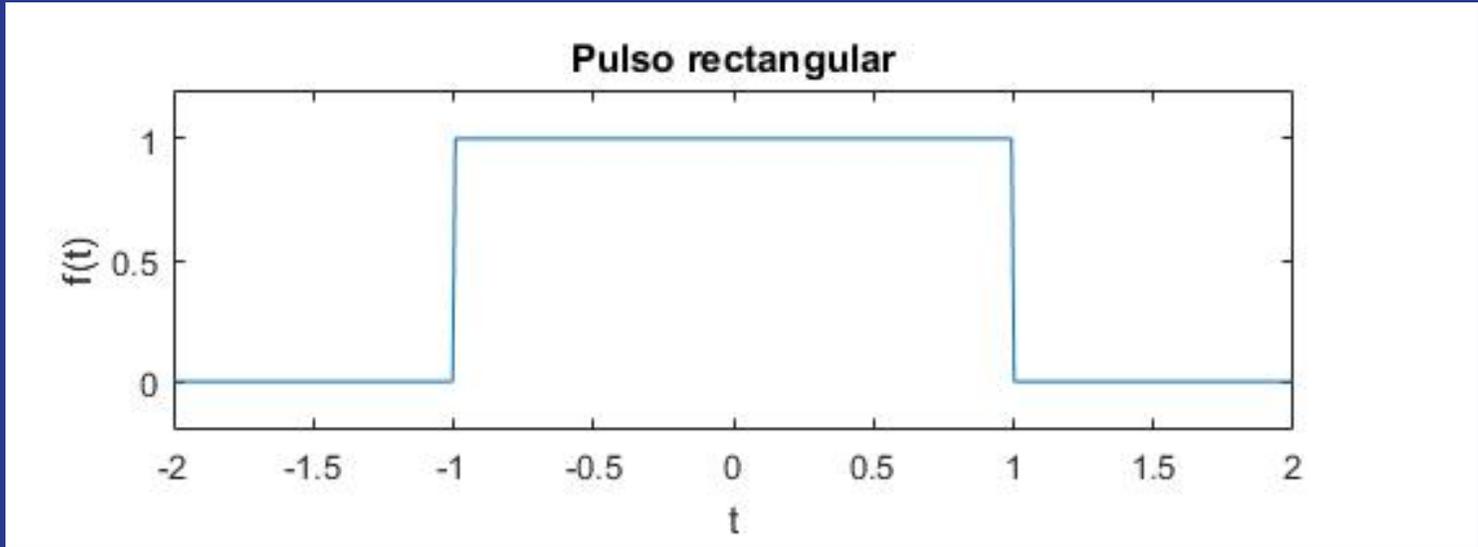
- 3) Cree un programa que haga lo siguiente:
- Defina una función que reciba como parámetros los valores  $T$ ,  $f_s$  y  $t$  donde  $T$  es la duración de una función pulso rectangular,  $f_s$  es la frecuencia de muestreo y  $t$  el intervalo de tiempo total. Queremos una función pulso rectangular con dominio de tiempo  $[-2,2]$  donde cada elemento en el mismo sea equiespaciado a través de  $1/f_s$ , del mismo modo queremos que la duración del pulso sea igual a 2 y que la frecuencia de muestreo sea igual a 1000.
  - Genere el pulso rectangular usando la función anteriormente mencionada
  - Calcule las frecuencias y magnitudes de la transformada de Fourier del pulso rectangular
  - Grafique la función pulso rectangular
  - Grafique la transformada de Fourier del pulso rectangular (si le aparece una función extraña que no respete el criterio de la recta vertical, restrinja el dominio de  $[0,2]$  y vuelva a graficar

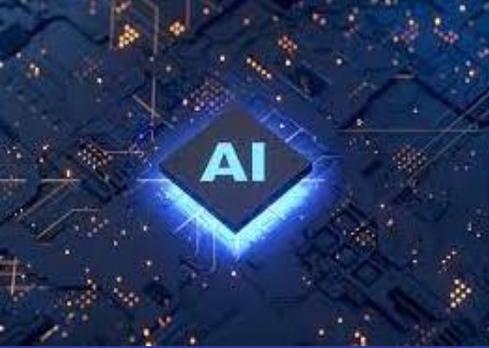
**PARA LOS LITERALES d) Y e) INVESTIGUE SOBRE EL USO DE LA LIBRERÍA MATPLOTLIB**



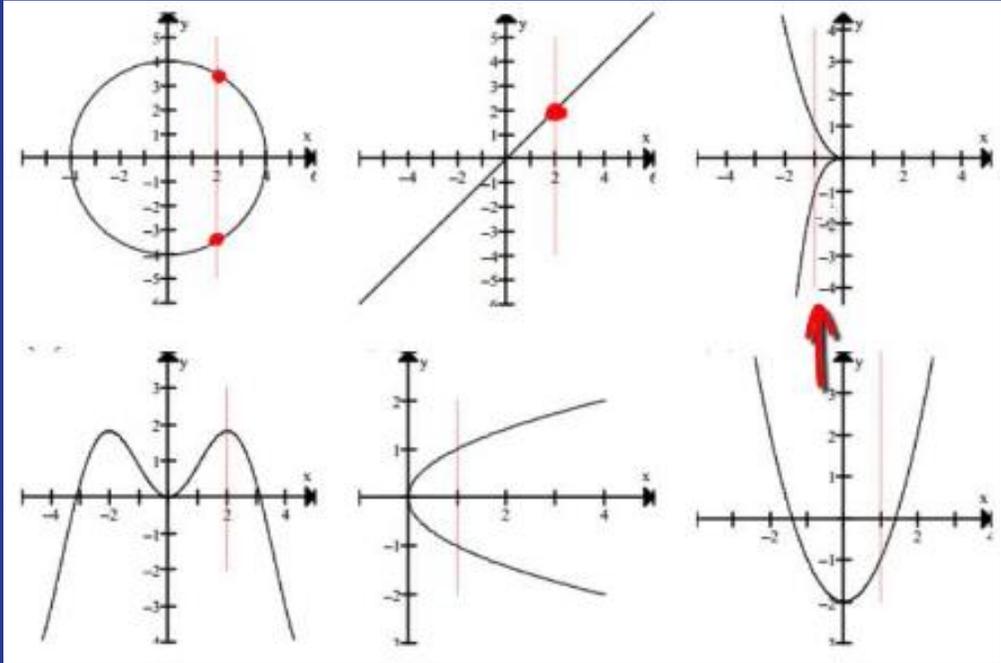


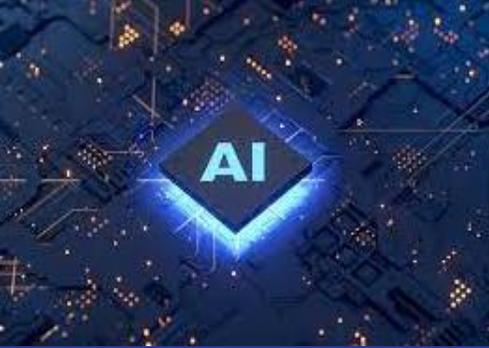
# Función pulso rectangular





# Criterio de la recta vertical





**Gracias por su atención**  
**¿Dudas?**

