

# IAGen

## PRIMER EJEMPLO DE IAGen EN PYTHON O.O

### **Introducción:**

A lo largo de nuestras guías anteriores, hemos explorado en profundidad los diversos aspectos de la Inteligencia Artificial (IA), proporcionando un enfoque completo desde los fundamentos hasta aplicaciones prácticas. Hasta ahora, hemos llegado a la guía número 12, donde hemos abordado temas clave como algoritmos, aprendizaje automático y redes neuronales. A partir de la Guía número 13, queremos dar un paso más allá y explorar no solo la IA tradicional, sino también la emergente Inteligencia Artificial Generativa (IAGen), un campo que está revolucionando la forma en que interactuamos con las máquinas y creamos contenido.

En las próximas guías, combinaremos los conceptos de IA clásica con las innovaciones de la IA generativa. Algunas de ellas estarán centradas en el uso de matemáticas y programación orientada a objetos con Python, herramientas fundamentales para quienes desean comprender y aplicar la IA en sus proyectos. Otras guías se dedicarán a explorar cómo la IA generativa está transformando áreas como la creación de texto, imágenes y música, ampliando las fronteras de lo que las máquinas pueden lograr por sí mismas.

Con este enfoque dual, nuestro objetivo es ofrecer una visión integral de la Inteligencia Artificial en sus dos vertientes: la inteligencia artificial tradicional, basada en modelos matemáticos y programación orientada a objetos, y la inteligencia artificial generativa, que introduce nuevas formas de interacción creativa. Al combinar estos dos mundos, esperamos proporcionar a nuestros lectores los conocimientos y herramientas necesarias para aprovechar al máximo las oportunidades que ofrece la IA en su totalidad.

### **Ejemplo de IAGen con Python O.O:**

**Calcular la suma de dos números enteros y diferentes, digitados por el usuario**

**Prompt usado:**

“Escribir un programa en Python orientado a objetos, para calcular la suma de dos números reales y diferentes, digitados por el usuario. El programa debe contar con las siguientes características: La clase principal se llama: "Adicion", sus parámetros son: "self, num1, num2". Esta clase "Adicion" tiene dos métodos: "\_\_init\_\_" y "sumar"(para sumar los dos números). En el ejemplo de uso le pedimos los dos números al usuario, y la clase Adicion suma estos números. Queremos que este programa tenga esta instrucción: "resultado = s.sumar()" para sumar los dos números. Al final se muestra la suma.”

**Respuesta de ChatGPT:**

Aquí tienes el código en Python orientado a objetos que cumple con lo que mencionas:

```

1  """
2  Creado marzo 12 de 2025
3  Nombre: PRIMER EJEMPLO DE IAGen EN PYTHON 0.0
4  Que hace: Calcula la suma de dos numeros reales
5  """
6  # Definicion de la clase Adicion
7  class Adicion:#Clase principal
8      def __init__(self, num1, num2): # Metodo constructor(__init__)
9          self.num1 = num1 # Primer numero
10         self.num2 = num2 # Segundo numero
11
12         def sumar(self): #Metodo para sumar los dos numeros
13             return self.num1 + self.num2
14
15 # Ejemplo de uso
16 def main():
17     # Solicitar los numeros al usuario
18     num1 = float(input("Ingrese el primer numero real: "))
19     num2 = float(input("Ingrese el segundo numero real:"))
20     # Crear una instancia de la clase Adicion
21     s = Adicion(num1, num2)
22     # Sumar los dos numeros utilizando el metodo sumar()
23     resultado = s.sumar()
24     # Mostrar el resultado
25     print(f"La suma de {num1} y {num2} es: {resultado}")
26
27 if __name__ == "__main__":
28     main()
29
30 #self es una convencion. No es una palabra reservada.
31 #self es un parametro en cada metodo
32 #self representa la instancia de la clase
33 #La sangria es la forma que usa Python para agrupar declaraciones.
34 #__name__ es una variable que evalua el nombre del modulo.

```

Se define una **clase** llamada **Adicion**. El **método** `__init__` es el constructor de la clase y recibe dos parámetros (num1 y num2).

`self.num1=num1`→Guarda el primer número en el atributo num1 del objeto.

`self.num2=num2`→Guarda el segundo número en el atributo num2 del objeto.

Se define un **método** `sumar()`, que simplemente retorna la suma de los dos números almacenados en el objeto. Se crea un objeto `s` de la clase **Adicion**, pasándole los dos números ingresados. Se llama al método `sumar()` de la clase y se almacena el resultado en la variable `resultado`.