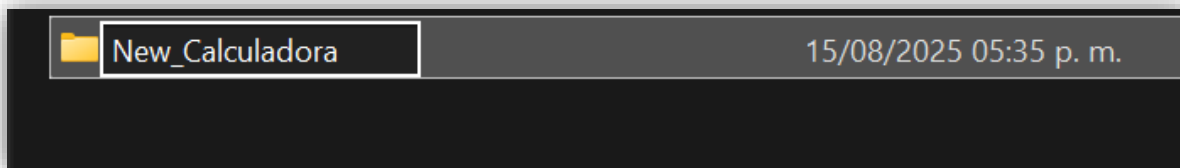


CREACIÓN DE UNA MEJOR CALCULADORA EN PYTHON

Este proyecto consiste en la creación de una calculadora en Python, la cual permite al usuario realizar operaciones básicas como suma y multiplicación de varios números, además de verificar si un número es par o si es un número entero.

1. Lo primero es crear una carpeta en el escritorio en este caso “New_Calculadora”, en la cual se ingresa un archivo readme.md, un archivo llamado test.py y un archivo main.py, los cual descargamos previamente del repositorio compartido por la instructora.



README	15/08/2025 04:36 p. m.	Archivo de origen Markdown	1 KB
main	15/08/2025 03:56 p. m.	Archivo de origen Python	3 KB
test	15/08/2025 02:09 p. m.	Archivo de origen Python	2 KB
Proyecto_Calculadora_Instrucciones	15/08/2025 01:53 p. m.	Documento de texto	2 KB

2. Después verificamos el contenido del archivo main.py

```
1     def main():
2         print("Hello learners!")
3
4     if __name__=="__main__":
5         main()
```

y archivo test.py

```
1  from main import addmultiplenumbers, multiplymultiplenumbers, isiteven, isitaninteger
2
3  # Test 1
4  def test_add_multiple_positive_integers():
5      response = addmultiplenumbers([5,7,9])
6      assert response == 21
7
8  # Test 2
9  def test_add_multiple_complicated_numbers():
10     response = addmultiplenumbers([5,-7,9.3])
11     response = round(response,1)
12     assert response == 7.3
13
14  # Test 3
15  def test_multiply_multiple_positive_integers():
16     response = multiplymultiplenumbers([4,5,6,7])
17     assert response == 840
18
19  # Test 4
20  def test_multiply_multiple_complicated_numbers():
21     response = multiplymultiplenumbers([4,-5,6.7])
22     response = round(response, 0)
23     assert response == -134
24
25  # Test 5
26  def test_is_six_even():
27     response = isiteven(6)
28     assert response == True
29
30  # Test 6
31  def test_is_minus_three_point_eight_even():
32     response = isiteven(-3.8)
33     assert response == False
34
35  # Test 7
36  def test_is_three_an_integer():
37     response = isitaninteger(3)
38     assert response == True
39
40  # Test 8
41  def test_is_seven_point_three_an_integer():
42     response = isitaninteger(7.3)
43     assert response == False
```

Que son las funciones base o de referencia para implementar las funciones dentro de la calculadora.

3. Iniciamos creando un prompt para utilizarlo como ayuda de la IA, para aplicar de forma correcta las funciones y en base al resultado se definieron de la siguiente manera:

- **addmultiplenumbers(numbers)**

```
#Funcion 1 esta suma todos los números de una lista
def addmultiplenumbers(lista):
    if not lista:
        return 0
    return sum(lista)
```

- **Función:** Suma todos los números que se ingresen en una lista.
- **Cómo funciona:** Usa la función incorporada sum() de Python para calcular la suma de todos los elementos de la lista numbers.
- Ejemplo : **addmultiplenumbers(3,5,-8,9,-2)**, lo cual devolverá como resultado: 7.

4. Se define la función para la multiplicación:

- **multiplymultiplenumbers(numbers)**

```
10 #Funcion 2 multiplica todos los números de una lista
11 def multiplymultiplenumbers(lista):
12     if not lista:
13         return 0
14     resultado = 1
15     for num in lista:
16         resultado *= num
17     return resultado
18
```

- **Función:** Multiplica todos los números que se ingresen en una lista.

- **Cómo funciona:** Inicializa una variable result en 1 y recorre la lista multiplicando cada número sucesivamente.
- **Ejemplo:** `multiplymultiplenumbers(8,9,5,6,3)`, lo cual dará como resultado= 6480

5. Definimos la función para determinar si el numero ingresado es numero par.

- **isiteven(number)**

```
19  #Funcion 3 verificara si un número es par
20  def isiteven(numero):
21      if not isitaninteger(numero):
22          return False
23      return numero % 2 == 0
24
```

- **Función:** Determinar si un número es par.
- **Cómo funciona:** Primero verificara si el número ingresado es un **entero** usando `isinstance()`.

Si es entero, calcula `number % 2` para comprobar la paridad.

Retorna True si es par y False si no lo es o si es decimal.

- **Ejemplo:** `isiteven(6)` lo marca verdadero , `isiteven(3.5)` lo marcara como falso.

6. Procedemos a definir la función para determinar si el numero ingresado es un numero entero.

- **isitaninteger(number)**

```
25  #Funcion 4 verificara si un número es entero
26  def isitaninteger(numero):
27      return isinstance(numero, int) or (isinstance(numero, float) and numero.is_integer())
```

- **Función:** Determinara si el número ingresado es un **entero**.
- **Cómo funciona:** Usa `isinstance()` para comprobar si **number** es del tipo `int`.
- **Ejemplo:** `isitaninteger(3)` lo marca como verdadero, `isitaninteger(7.3)`, lo marca como falso.

7. Definimos la función main

La función **main()** es la función principal de la calculadora. Su propósito es mostrar un menú de opciones al usuario y ejecutar la operación seleccionada de manera repetitiva hasta que el usuario decida salir.

Flujo de la función:

Bienvenida: Al iniciar, la función muestra un mensaje de bienvenida:

```
def main():  
    print("¡Bienvenido a la calculadora!")
```

Bucle principal: La función entra en un bucle infinito (**while True**) que permite al usuario realizar varias operaciones sin reiniciar la calculadora.

```
while True: # Bucle para repetir operaciones
```

Menú de opciones: Se muestran las opciones disponibles:

```
print("\nOpciones:")  
print("1) Sumar lista de números")  
print("2) Multiplicar lista de números")  
print("3) Verifica si un número es par")  
print("4) Verifica si un número es entero")  
print("5) Salir")
```

Selección del usuario: Se solicita al usuario que ingrese una opción del 1 al 5:

```
opcion = input("Elige una opción (1-5): ")
```

Salir de la calculadora: Si el usuario selecciona 5, se muestra un mensaje de despedida y se termina el bucle:

```
if opcion == "5":  
    print("Gracias por usar la calculadora. ¡Hasta luego!")  
    break
```

8. Para las operaciones con listas de números definidas como las opciones 1 (sumar) y 2 (multiplicar):
- Se solicita al usuario ingresar los números separados por comas.
 - Se convierten a una lista de float.
 - Se llama a la función correspondiente (**addmultiplenumbers** o **multiplymultiplenumbers**) y se muestra el resultado.
 - Si el usuario ingresa un valor no numérico, se muestra un mensaje de error:

```
# Operaciones con listas para la suma y multiplicacion de numeros
elif opcion in ["1", "2"]:
    numeros = input("Ingresa los números separados por coma: ")
    try:
        lista = [float(n) for n in numeros.split(",")]
        if opcion == "1":
            print("Resultado:", addmultiplenumbers(lista))
        else:
            print("Resultado:", multiplymultiplenumbers(lista))
    except ValueError:
        print("Error: Debes ingresar solo números separados por comas.")
```

9. Para las operaciones con un solo número, en este caso verificar si es par o entero definidas como las opciones 3 (verificar si es par) y 4 (verificar si es entero):
- Se solicita un número al usuario.
 - Se convierte a float.
 - Se llama a la función correspondiente (isiteven o isitaninteger) y se muestra el resultado.
 - Si el usuario ingresa un valor inválido, se muestra un mensaje de error:

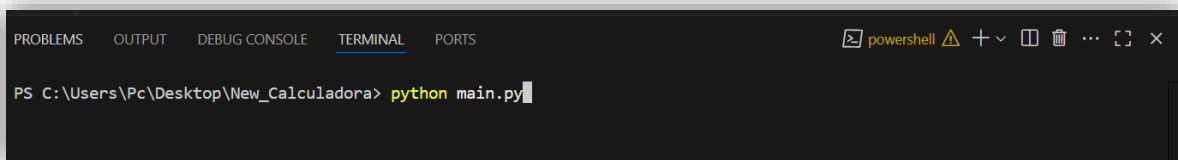
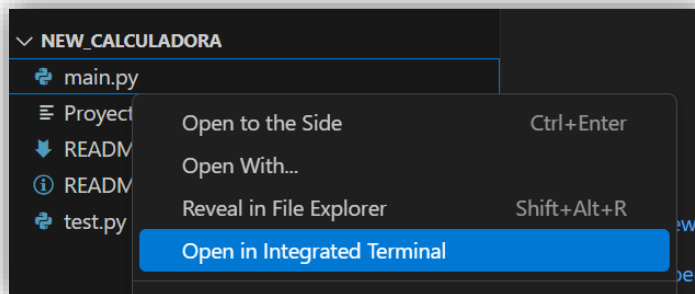
```
# Operaciones con un solo número para indicar si es par o entero
elif opcion in ["3", "4"]:
    numero_input = input("Ingresa un número: ")
    try:
        numero = float(numero_input)
        if opcion == "3":
            print("¿Es par?", isiteven(numero))
        else:
            print("¿Es entero?", isitaninteger(numero))
    except ValueError:
        print("Error: Debes ingresar un número válido.")
```

Opción que se marcará como inválida : Si el usuario ingresa cualquier valor que no esté entre 1 y 5, se muestra un mensaje de advertencia:

```
    else:
        print("Opción no válida. Por favor, elige un número entre 1 y 5.")

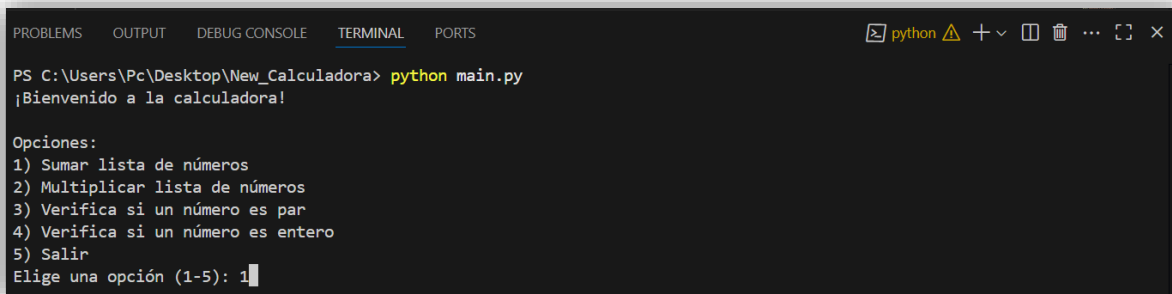
# Ejecutar la calculadora
if __name__ == "__main__":
    main()
```

10. Finalmente se ejecuta el programa desde la terminal integrada de Python, con el siguiente comando **python main.py**



Y se prueba la calculadora con las funciones determinadas:

En este caso al seleccionar la opción 1) ejecutará la opción de sumar una lista de números y mostrará el resultado, considerando números tanto negativos como positivos.



```
PS C:\Users\Pc\Desktop\New_Calculadora> python main.py
¡Bienvenido a la calculadora!

Opciones:
1) Sumar lista de números
2) Multiplicar lista de números
3) Verifica si un número es par
4) Verifica si un número es entero
5) Salir
Elige una opción (1-5): 1
Ingresa los números separados por coma: 3,5,-8,9,-2
Resultado: 7.0
```

Se prueba con la opción 2) ejecutara la opción de multiplicar una lista de números y mostrara el resultado.

```
Opciones:
1) Sumar lista de números
2) Multiplicar lista de números
3) Verifica si un número es par
4) Verifica si un número es entero
5) Salir
Elige una opción (1-5): 2
Ingresa los números separados por coma: 8,9,5,6,3
Resultado: 6480.0
```

Se prueba con la opción 3) ejecutara la opción que determina si el numero es par, si es par muestra (true) sino (false).

```
Opciones:
1) Sumar lista de números
2) Multiplicar lista de números
3) Verifica si un número es par
4) Verifica si un número es entero
5) Salir
Elige una opción (1-5): 3
Ingresa un número: 4
¿Es par? True

Opciones:
1) Sumar lista de números
2) Multiplicar lista de números
3) Verifica si un número es par
4) Verifica si un número es entero
5) Salir
Elige una opción (1-5): 3
Ingresa un número: 5
¿Es par? False
```


Se prueba con la opción 4) ejecutara la opción que determina si el número es entero, si es entero muestra (true) sino (false), al igual que con el numero par.

Se prueba con la opción 3) ejecutara la opción que determina si el número es par, si es par muestra (true) sino (false).

```
Opciones:
1) Sumar lista de números
2) Multiplicar lista de números
3) Verifica si un número es par
4) Verifica si un número es entero
5) Salir
Elige una opción (1-5): 4
Ingresa un número: 10
¿Es entero? True

Opciones:
1) Sumar lista de números
2) Multiplicar lista de números
3) Verifica si un número es par
4) Verifica si un número es entero
5) Salir
Elige una opción (1-5): 4
Ingresa un número: 5.5
¿Es entero? False
```

Al elegir la opción 5) esta finaliza y sale de la calculadora con un mensaje.

```
Opciones:
1) Sumar lista de números
2) Multiplicar lista de números
3) Verifica si un número es par
4) Verifica si un número es entero
5) Salir
Elige una opción (1-5): 5
Gracias por usar la calculadora. ¡Hasta luego!
PS C:\Users\Pc\Desktop\New_Calculadora>
```

Como opción para visualizar la calculadora con una interfaz gráfica, se pidió a la IA que tomara el código de referencia del archivo main.py e integrara la librería “Tkinter” para la interfaz gráfica de usuario (GUI). Tkinter permite construir ventanas, etiquetas,

entradas de texto y botones que hacen más intuitiva la interacción del usuario con las funciones matemáticas implementadas.

Construcción de la Ventana Principal

La función **main()** crea la ventana gráfica principal con etiquetas, entradas de texto, botones de operaciones y verificación, así como un botón para salir de la aplicación.

Ejemplo de creación de ventana:

- `ventana = tk.Tk() ventana.title("Calculadora Interactiva")
ventana.geometry("350x350") ventana.config(bg="#f0f0f0")`

Ejecución del Programa

El bloque final asegura que la calculadora se ejecute únicamente cuando el archivo es ejecutado directamente:

- `if __name__ == "__main__": main()`

Conclusión:

El uso de Tkinter permitió transformar funciones matemáticas en una aplicación interactiva, amigable y fácil de usar. Cada botón está vinculado a una función lógica mediante comandos (**command**), y los resultados se muestran al usuario con ventanas emergentes claras y dinámicas.

Nota: Para probar la calculadora con interfaz gráfica, se ejecuta el programa en la terminal con el comando : **python calculinteractiva.py**, de igual manera se adjunta su archivo Readme calculinteractiva.md

