

Bootcamp Especialista de Soporte TI.

Actividad:

Mejorar una calculadora.

Eduardo García García

Ciudad Valles, S.L.P. a 16 de agosto de 2025

Índice

Comentarios ante de iniciar la actividad.....	¡Error! Marcador no definido.
Inicio de la VM.....	¡Error! Marcador no definido.

Introducción.

La siguiente actividad consiste en mejorar una calculadora en Python haciendo que está pueda sumar y multiplicar los números de una lista, también puede identificar si un número es par o si es un número entero. El objetivo principal es aplicar el uso de funciones, estructuras de control y pruebas automáticas en Python, creando un programa modular fácil de leer. Esta calculadora se ejecuta desde la consola utilizando un menú interactivo.

Funciones del proyecto.

La calculadora contiene 4 funciones:

1. addmultiplenumbers(lista)
2. multiplymultiplenumbers(lista)
3. isiteven(numero)
4. isitaninteger(numero)

Explicación de las funciones.

I. Addmultiplenumbers(lista)

Esta función recibe una lista de números y los va sumando uno por uno. Si la lista que recibe está vacía, regresa 0. Se usa la variable “total” que empieza está inicializada en 0 y va acumulando los valores dentro de un ciclo “for”.

```
1  # Función para sumar todos los números de una lista
2  def addmultiplenumbers(list):
3      # Si la lista está vacía regresa 0 para evitar errores.
4      if not list:
5          return 0
6      total = 0
7
8      # Se hace el recorrido de cada número de la lista
9      for n in list:
10         # Se hace la suma de cada número
11         total = total + n
12     # Regresamos el valor de la suma
13     return total
```

II. Multilymultiplenumbers(lista):

Esta función hace lo mismo pero multiplicando todos los elementos de la lista. La variable “total” se inicializa en 1 (número neutro multiplicativo). Si la lista está vacía regresa 1.

```
16 # Función para multiplicar todos los números de una lista
17 def multiplymultiplenumbers(lista):
18     # Si la lista está vacía regresamos 1
19     if not lista:
20         return 1
21     total = 1
22
23     # Se hace el recorrido de cada número de la lista
24     for n in lista:
25         # Multiplicamos el total por cada número de la lista
26         total = total * n
27         # Regresamos el valor de la multiplicación
28     return total
```

III. Isitaninteger(numero):

Esta función verifica si un número es entero. Usa el operador “%1”. Si el resultado del módulo es 0 entonces el número ingresado es entero y nos devuelve “True”.

```
31 # Función para revisar si un número es entero
32 def isitaninteger(numero):
33     # Hacemos uso del módulo 1 y si el residuo es 0 el numero evaluado es entero
34     if numero % 1 == 0:
35         # Regresamos un True
36         return True
37     # Regresamos un False
38     else:
39         return False
```

IV. Isiteven(numero)

Esta función primero verifica si el número ingresado es entero. Si no lo es, regresa un “False” directamente. Si el número si es entero, entonces revisa si es divisible entre 2, lo que nos dirá si es par o no.

```
42 # Función para saber si un número es par y entero
43 def isiteven(numero):
44     # Si el número no es entero regresamos un False
45     if numero % 1 != 0:
46         return False
47
48     # Una vez que sabemos que es entero comprobamos si es par
49     if numero % 2 == 0:
50         # Si es par regresamos True
51         return True
52     # Si no es par regresamos un False
53     else:
54         return False
```

Menú de la consola.

Debajo de todas las funciones escribimos un menú dentro de la función “main”, este nos servirá para mostrar al usuario las opciones. Dependiendo de la opción que elija el usuario, se le pedirá ingresar los números a través de un “input” y llamaremos a la función que corresponda.

```
def main():  
    print("¡Bienvenido a la calculadora mejorada!")  
  
    # Bucle infinito para que el menú se repita hasta salir  
    while True:  
        # Mostramos las opciones del menú  
        print("\nElige una opción:")  
        print("1. Sumar una lista de números")  
        print("2. Multiplicar una lista de números")  
        print("3. Ver si un número es par")  
        print("4. Ver si un número es entero")  
        print("5. Salir")  
  
        # Preguntamos al usuario que desea realizar  
        opcion = input("Ingresa tu opción (1-5): ")
```

Pruebas con pytest.

Utilizamos el archivo “test.py” para hacer el test de nuestro programa ejecutándolo desde la terminal de nuestro VS Code con el comando “pytest test.py”.

```
PS C:\Users\Lalo\Documents\CH3\Introduccion\Actividades\Calculadora_Mejorada> pytest test.py  
===== test session starts =====  
platform win32 -- Python 3.13.5, pytest-8.4.1, pluggy-1.6.0  
rootdir: C:\Users\Lalo\Documents\CH3\Introduccion\Actividades\Calculadora_Mejorada  
plugins: cov-6.2.1  
collected 8 items  
  
test.py ..... [100%]  
  
===== 8 passed in 0.16s =====  
PS C:\Users\Lalo\Documents\CH3\Introduccion\Actividades\Calculadora_Mejorada>
```

Ejemplo de uso.

Ejecutamos desde la terminal de VS Code nuestro programa con el comando “py main.py”.

<pre>¡Bienvenido a la calculadora mejorada! Elige una opción: 1. Sumar una lista de números 2. Multiplicar una lista de números 3. Ver si un número es par 4. Ver si un número es entero 5. Salir Ingresa tu opción (1-5): 1 Ingresa los números separados por comas: 1,9,5,3 Resultado de la suma: 18.0</pre>	<pre>Elige una opción: 1. Sumar una lista de números 2. Multiplicar una lista de números 3. Ver si un número es par 4. Ver si un número es entero 5. Salir Ingresa tu opción (1-5): 3 Ingresa un número: 5 5.0 no es par.</pre>
---	---

Dificultades y solución.

Lo que más me costó trabajo al realizar la actividad fue comprender cómo crear funciones en Python que recibieran listas como parámetro. Al principio intenté hacerlo a mi manera, pero no entendía cuál era el propósito del archivo test.py ni porque era necesario.

Para resolver mis dudas decidí apoyarme con ChatGPT, donde fui preguntando paso a paso como hacer que una función recibiera datos en forma de lista y como hacer que hiciera las operaciones requeridas y que me explicara para que servía el archivo test.py y porque era necesario.