Proyecto Calculadora en Python

Erick David Albadan Eslava

Nicolas Herrera Cuevas

Corporación Universitaria Minuto de Dios

Facultad de Ingeniería

Ingeniería de sistemas

Programación Orientada a Objetos NRC 21621

Bogotá, septiembre de 2022

Índice

Contenido

Creación y desarrollo

Lo primero que hicimos fue crear funciones que nos impriman menús y preguntas que se le realizarán al usuario más adelante. La primera posee el menú de selección de operación mientras que la segunda otro menú para saber si el usuario desea realizar otra operación o no.

```
def menu_operaciones2():
    print("MENÚ OPERACIONES")
    print("1) Suma")
    print("2) Resta")
    print("3) Multiplicación")
    print("4) División")
    print("5) Potenciación")
    print("6) Raíz cuadrada")

def repetir_operaciones():
    print("¿Desea realizar otra operación?")
    print("1) Sí, por favor")
    print("2) No, chao")
```

Elaboramos muchas más funciones, pero para la explicación es de preferencia realizarla en otro orden.

Ahora viene el main. El programa inicia imprimiendo un saludo hacia el usuario, luego creando variables y dándoles el valor de True (esto con el fin de usar dichas variables en ciclos while).

```
# M A I N 👋

print("Bienvenido a nuestra calculadora")
print("Esperamos que te ayude en tus cálculos")

ciclo_operaciones=True
pregunta_repetir=True
```

Ahora entra a un ciclo while(ciclo_operaciones) ya que como se evidencia en la imagen anterior, esa variable es True, y no saldrá de aquel ciclo hasta que esta variable sea False. Dentro

del ciclo lo que se hace es imprimir el menú con las siete opciones que se pueden observar en la imagen.

A continuación, se crea otra variable llamada comprobar1 y se le asigna un valor True, esto se hizo con el fin de evaluar si la respuesta del usuario ante el menú de operaciones es un número entero. Supongamos que el usuario ingresó una cadena de cadena de caracteres, ya que todo está dentro de un try/except, al tomar esto como un error, imprimirá el mensaje de "Inserte un valor válido" y como comprobar1 seguirá siendo True, volverá a preguntar por una opción del menú. No se logrará salir de este ciclo hasta que el usuario inserte un número entero, ya que, de esta forma, a comprobar1 se le asignará el valor False.

```
while(ciclo_operaciones):
    print("MENÚ OPERACIONES")
    print("1) Suma")
    print("2) Resta")
    print("3) Multiplicación")
    print("4) División")
    print("5) Potenciación")
    print("6) Raíz cuadrada")
    print("7) Salir")
    comprobar1=True
    while(comprobar1):
        try:
            respuesta1=int(input("Seleccione una operación del menú: "))
            comprobar1=False
        except:
            print("Inserte un valor válido")
```

Luego de validar que el valor ingresado en respuesta1 sea un número entero, elaboramos condiciones para aquel valor. Dependiendo de la respuesta del usuario, entrará y llamará a la función que contenga la operación que el usuario desea hacer, después de realizar la operación saldrá de este ciclo y pasará a preguntar si se desea realizar otra operación, pero eso lo veremos más adelante. Si el valor ingresado no está en el menú, se le imprimirá al usuario un mensaje diciéndole que inserte un valor válido, y como ninguna variable cambia su valor a falso, repetirá el ciclo de preguntar la operación por hacer o en caso de no desear ninguna, salir.

La opción salir, como se puede observar, a diferencia de las otras opciones, cambia el valor de dos variables, mientras que en las demás opciones válidas sólo cambia el de una. Esto es

porque si se da la opción de salir, aparte de salir del ciclo en el que se pregunta por la operación a realizar, no entrará al ciclo en el que se pregunta si se desea realizar una operación.

```
    if(respuesta1==1):
        suma()
        ciclo_operaciones=False
    elif(respuesta1==2):
        resta()
        ciclo_operaciones=False
    elif(respuesta1==3):
        multiplicar()
        ciclo_operaciones=False
    elif(respuesta1==4):
        dividir()
        ciclo_operaciones=False
    elif(respuesta1==5):
        potencia()
        ciclo_operaciones=False
    elif(respuesta1==6):
        raz()
        ciclo_operaciones=False
    elif(respuesta1==7):
        print("Cuídate")
        ciclo_operaciones=False
        pregunta_repetir=False
    else:
        print("Inserte un valor válido")
```

```
def bucle_operaciones():
    ciclo_operaciones=True
    pregunta_repetir=True
    while(ciclo_operaciones):
        menu_operaciones2()
        comprobar1=True
        while(comprobar1):
            try:
                respuesta1=int(input("Seleccione una operación del menú: "))
                comprobar1=False
            except:
                print("Inserte un valor válido")
        if(respuesta1==1):
            suma()
            ciclo_operaciones=False
        elif(respuesta1==2):
            resta()
            ciclo_operaciones=False
        elif(respuesta1==3):
            multiplicar()
            ciclo_operaciones=False
        elif(respuesta1==4):
            dividir()
            ciclo_operaciones=False
        elif(respuesta1==5):
            potencia()
            ciclo_operaciones=False
        elif(respuesta1==6):
            raz()
            ciclo_operaciones=False
        else:
            print("Inserte un valor válido")
```

Este proceso cíclico lo ingresamos dentro de una función (bucle_operaciones) con el fin de disminuir líneas de código.

Como ya mencionamos, las opciones del uno al seis llaman a una función, a continuación, se dará una explicación a cada una de ellas.

Suma:

Se le pide al usuario que inserte la cantidad de dígitos que desea sumar, esto dentro de un ciclo while con try/except ya explicado con anterioridad (verifica si el valor ingresado es un número entero).

Luego se crea una lista vacía (lista_valores), la cual contendrá los valores que el usuario desea sumar, para después, ingresar a un ciclo for que se repetirá las mismas veces que la cantidad que el usuario haya ingresado de valores a sumar, esto con el fin de agregarlos a la lista.

Se crea una variable para comprobar que el número ingresado sea un número real, en caso de ser así, se agregará a la lista con lista valores.append(b), en caso opuesto, imprimirá un mensaje diciendo que se inserte un valor válido y vuelve a preguntar. Luego de haber ingresado todos los valores deseados en la lista, se suman mediante sum () la cual es una función integrada en Python que suma todos los números de una lista.

Después de sumarlos, la variable “total” tomará el valor de dicha sumatoria, para así imprimir el resultado.

```python
def suma():
    comprobar=True
    while(comprobar):
        try:
            print("¿Cuántos valores desea sumar?")
            x=int(input("Respuesta: "))
            comprobar=False
        except:
            print("Inserte un valor válido")

    lista_valores=[]

    for x in range (0,x,1):
        comprobar2=True
        while(comprobar2):
            try:
                b=float(input("Digite un valor: "))
                comprobar2=False
            except:
                print("Ingrese un valor válido")
        lista_valores.append(b)
    total=sum(lista_valores)
    print("El resultado es: ",total)
```

Resta:

En este caso, sólo resta dos números, no como en la suma que realiza la operación con todos los elementos de una lista, la justificación a esto estará en la sección final de notas.

Esta función le pide al usuario ingresar dos valores, todo en un ciclo while y try/except con el mismo fin que los anteriores casos.

Luego de insertar los valores a y b, los resta. Aquel valor obtenido de la resta de estos dos números se guarda en la variable “diferencia”, que será utilizada para imprimir el resultado.

```
def resta():
    comprobar=True
    while(comprobar):
        try:
            a=float(input("Inserte el primer valor a restar: "))
            b=float(input("Inserte el segundo valor a restar: "))
            comprobar=False
        except:
            print("Inserte un valor válido")

    diferencia=a-b
    print("La respuesta es: ", diferencia)
```

Multiplicación:

En esta función sí es posible multiplicar todos los valores que el usuario desee.

Esta ya posee una complejidad mayor. El procedimiento inicial es el mismo que en el caso de la suma: se le pide al usuario que ingrese cuántos números desea sumar (con el ciclo que evalúa si el valor ingresado es válido), se crea una lista vacía, y se repite el ciclo las veces suficientes para que el usuario ingrese los valores que desea operar, ya que el valor ingresado en x es la cantidad de veces que se repetirá el ciclo, que nos ayuda a ingresar números en la lista vacía.

Hasta ahora todo es igual que en la función de la suma. Luego de esto, se crea una variable llamada “total” y su valor inicial es 1, ya diremos el porqué. Ahora se entra en un ciclo for que se repite por cada elemento que haya en la lista lista_valores.

El valor 1 inicial sirve para este ciclo, primero se multiplicará el primer valor de la lista por uno, y total tomará el valor del resultado de dicha multiplicación, tomando así el valor del primer elemento de la lista, luego se multiplicará por el segundo y así sucesivamente hasta que no queden más elementos en la lista. Finalmente imprimirá el resultado de la operación.

```python
def multiplicar():
    comprobar=True
    while(comprobar):
        try:
            print("¿Cuántos valores desea multiplicar?")
            x=int(input("Respuesta: "))
            comprobar=False
        except:
            print("Inserte un valor válido")

    lista_valores=[]

    for x in range (0,x,1):
        comprobar2=True
        while(comprobar2):
            try:
                b=float(input("Digite un valor: "))
                lista_valores.append(b)
                comprobar2=False
            except:
                print("Ingrese un valor válido")

    total=1

    for numero in lista_valores:
        total = total * numero
    print("El resultado es: ", total)
```

División:

En este caso sólo es posible dividir con dos números, decidimos verlo como una fracción con un numerador y un denominador.

El usuario deberá ingresar los dos valores solicitados. La variable a tomará el rol de numerador y la variable b el de denominador. Es lo mismo que en la resta, solicita dos valores y los opera, para luego imprimir el resultado.

La única diferencia que posee es la del condicional, como no es posible dividir entre cero, creamos una condición que imprima un mensaje diciendo que el valor es indeterminado en caso de que el denominador (b) sea cero. En caso de que no, operará los valores normalmente e imprimirá el resultado de dicha división.

```
def dividir():
    comprobar=True
    while(comprobar):
        try:
            a=float(input("Inserte el numerador: "))
            b=float(input("Inserte el denominador "))
            comprobar=False
        except:
            print("Inserte un valor válido")

    if(b==0):
        print("No es posible dividir entre cero, su valor es indetederminado")
    else:
        div=a/b
        print("La respuesta es: ", div)
```

Potenciación:

En este caso también es posible operar únicamente entre dos números, ya que uno de estos tomará el rol de base y el otro de exponente.

En un principio funciona igual que la función de resta, sólo cambia el operador de estos valores, el cual es **, que indica que hay que elevar un número. Este mismo operador lo utilizamos en la función de sacar raíz cuadrada de un número, ya se explicará cómo.

```
def potencia():
    comprobar=True
    while(comprobar):
        try:
            a=float(input("Inserte la base: "))
            b=float(input("Inserte el exponente: "))
            comprobar=False
        except:
            print("Inserte un valor válido")

    pot=a**b
    print("El resultado es: ", pot)
```

Raíz cuadrada:

Sólo se solicitará un número, al cual se le sacará su raíz cuadrada. La raíz cuadrada de un número puede traducirse a ese número elevado a un medio (0,5), por ende, sólo basta con

solicitar un número y elevarlo a 0,5 con el operador mencionado en la función anterior (potenciación).

```
def raz():
    comprobar=True
    while(comprobar):
        try:
            a=float(input("Inserte el valor a sacar raíz cuadrada: "))
            comprobar=False
        except:
            print("Inserte un valor válido")

    raiz=a**0.5
    print("El resultado es: ", raiz)
```

Terminada la explicación de las funciones que contienen las operaciones de los números, procederemos a la explicación del ciclo en el que se pregunta por la realización de otra operación.

En un principio, se creó una variable llamada “pregunta_repetir” y se le asignó el valor True, por lo tanto, ingresará a este ciclo while (excepto cuando el usuario tomó la opción de salir, ya que en esta séptima opción aquella variable toma el valor de False).

Al entrar, llamará a la función repetir operaciones, la que nos imprime la pregunta de si se desea realizar otra operación y luego ingresar una respuesta en respuesta2 (la cual debe ser un número entero, este valor está siendo verificado con el ciclo while(comprobar) y con try/except que tanto hemos usado y explicado alrededor de documento.

Una vez ingresado un número entero, nos hallaremos con condicionales, si respuesta2 equivale a 1, entonces realizará nuevamente el proceso de realizar otra operación, ya que, al realizar otra operación, pregunta repetir seguirá siendo True, volverá a preguntar si se desea realizar otra operación y así sucesivamente.

Si respuesta2 equivale 2, entonces se imprimirá un valor de despedida al usuario y pregunta repetir tomará el valor de False, para salir de este ciclo y así finalizar el programa.

En caso de que sea un número entero, pero no sea ni uno ni dos, entonces se imprimirá un mensaje diciendo que se ingrese un valor válido, como pregunta repetir sigue siendo True, volverá a repetir si se desea hacer una operación.

```
while(pregunta_repetir):

    comprobar3=True
    while(comprobar3):
        repetir_operaciones()
        try:
            respuesta2=int(input("Respuesta: "))
            comprobar3=False
        except:
            print("Inserte un valor válido")

    if(respuesta2==1):
        bucle_operaciones()
    elif(respuesta2==2):
        print("Vale, cuídate")
        pregunta_repetir=False
    else:
        print("Inserte un valor válido")
```

Notas:

Anteriormente se mencionó algo sobre que sólo se podían operar dos números en la función de resta, en vez de todos los que el usuario quiera por un inconveniente.

Como no hallamos una función integrada en Python que nos reste los elementos de una lista (tal como sum los suma), optamos por elaborar nosotros mismos una función que lo hiciera. Decidimos tomar como base/plantilla la función de la multiplicación, ya que inicialmente posee todo lo que nos facilitaría ese proceso (el de ingresar números en una lista vacía ya creada).

```
def resta():
    comprobar=True
    while(comprobar):
        try:
            print("¿Cuántos valores desea restar?")
            x=int(input("Respuesta: "))
            comprobar=False
        except:
            print("Inserte un valor válido")

    lista_valores=[]

    for x in range (0,x,1):
        comprobar2=True
        while(comprobar2):
            try:
                b=float(input("Digite un valor: "))
                lista_valores.append(b)
                comprobar2=False
            except:
                print("Ingrese un valor válido")
```

Todo lo anterior funciona correctamente, el problema es lo siguiente:

```
    total=0

    for numero in lista_valores:
        total = total - numero
    print("El resultado es: ", total)
```

Lo que sucede es que total vale cero, para que le reste cero al primer elemento de la lista y no afectar su valor, luego tome el valor del primer valor y empiece a restarlo con el segundo, así sucesivamente. El problema con esta parte es que por ejemplo si se resta cero y uno, no toma el valor de uno, toma el de menos uno, ya que hace podría decirse que el programa lo toma como si fuese la naturaleza de ese número. El resultado es que, en vez de restarlos, lo que hace es sumarlos y convirtiendo el resultado de esa suma en un valor negativo. Aquí un ejemplo:

```
¿Cuántos valores desea restar?
Respuesta: 3
Digite un valor: 1
Digite un valor: 1
Digite un valor: 1
El resultado es:  -3.0
```

Uno menos uno, da cero, y cero menos uno, da menos uno. El programa lo que hizo fue sumar esos tres valores y darle un valor negativo.

Tratamos de eludir este problema modificando posiciones de variables, multiplicando cada resta por -1, pero ninguna de estas alternativas daba con un resultado deseado.

```
    total=0

    for numero in lista_valores:
        total = numero - total
    print("El resultado es: ", total)
```

```
total=0

for numero in lista_valores:
    total = numero - total * -1
print("El resultado es: ", total)
```

Este fue el único inconveniente que se nos presentó y que, hasta el momento, no hemos hallado una solución.

De esta forma finaliza la explicación, muchas gracias por su tiempo y esperamos que tenga un buen día.