# Calculos matemáticos básicos

(Calculadora científica)

Sarahí Montserrat Ramírez Padilla (1), Rosendo Emanuel Valencia Ojeda (2)

(1) Universidad de Colima Facultad de ingeniería Civil, Ingeniero Topógrafo Geomático 28400, sramirez29@ucol.mx.

Resumen

Se va a realizar un programa que nos ayude a realizar cálculos matemáticos básicosm empleando las TIC, fórmulas matemáticas, el Python, así como también la utilización de sus librerías. Lo que queremos lograr es obtener resultados utilizando las formulas con nuestro programa para así tener más certeza y minimizar el tiempo

**Palabras clave**: Cálculos, TIC, matemáticas.

Abstract

It is a program that helps us to perform basic mathematical calculations that use ICT, mathematical formulas, Python, as well as the use of their libraries. What we want to achieve is to obtain results using the formulas with our program in order to have more certainty and minimize the time

**Keywords:** Calculations, ICT, mathematics

## Introducción

Al realizar un problema matemático lo que se busca es la precisión de los resultados en el menor tiempo posible, y tener certeza de lo que se está calculando. No siempre es fácil realizar los cálculos mentalmente o a mano por el motivo de que llega a ser muy tardado y puede haber una confusión de los números o con la operación misma.

Para los problemas mencionados anteriormente hemos pensado una solución, la cual se trata de un programa realizado en Python que nos ayude a la obtención de los resultados de las operaciones matemáticas de una gorma mucho más rápida y eficiente para la comodidad y seguridad de la persona.

## Desarrollo

El uso de las librerías de Python nos permite añadir funciones adicionales al lenguaje sin la necesidad de utilizar otro.

Las librerías a utilizar para nuestro programa nos permiten la utilización de algunas funciones matemáticas y darle formato a nuestro programa. Las librerías a utilizar son las siguientes:

**Tkinder** es una librería orientada a diseñar la interfaz gráfica para aplicaciones de escritorio hecho a través del lenguaje de programación Python. Sirve para crear las ventanas tan útiles a niver visual.

**Math** es una librería en Python que nos permite realizar funciones hiperbólicas, trigonométricas y logarítmicas solamente para números reales.

* Para hacer uso de las librerías es importante importarlas de la siguiente manera:

****

En este caso con Tkinder crearemos una ventana de la siguiente manera:

1. Primeramente, se hará una variable para ingresar los datos. En este caso la variable se llama **ventana** se igualará y se escribirá la librería a utilizar
2. Se ingresa el título de nuestro programa con **ventana.title**
3. Con **ventana.geometry** se ingresaran los valores de dimensión que tendrá la ventana



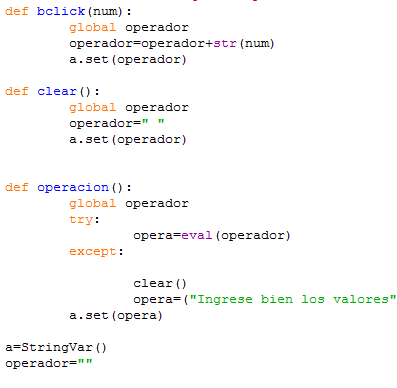
Creamos variables en la que estarán especificado el tamaño que tengan cada botón **al** define lo alto del botón y **an** el ancho



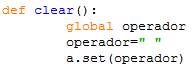
Creamos las funciones que nos permitirán realizar los cálculos al usuario

La primera función se llamará botón clic **bclick**, esta función la vamos a crear con un parámetro ya que vamos a usar una función que se llama global. El nombre del parámetro será **num**

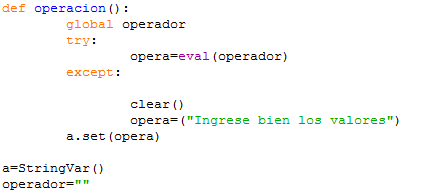
**Global** quiere decir que nos va a dar cualquier número, un número global



Definir la operación **clear** nos proporcionara la eliminación de valores que ya no se necesitan



Es importante definir operación para poder realizar las operaciones. **Try** lo que nos va a decir es si la función es evaluable, si es así la función se ejecutara y si no te pedira que ingreses bien los valores



Se creó el diseño de la calculadora la parte donde aparecerán nuestros resultados

salida=Entry(ventana, font=("Arial",25,"bold"), textvariable=a, width=22, bd=10, insertwidth=5, bg="powder blue", justify="right")

salida.pack()

Para mantener la ventana abierta se utiliza

**ventana.mainloop()**

**Codigo de los botones**

**botonarccos=** Variable del botón

**Button=** Comando para que se cree el botón

**(ventana, text="arccos" =** Nombre que tendrá la ventana, y tiene que ir entre comillas el simbolo

**width=an,** Dimensiones del botón

**height=al, Dimensiones** del botón

El comando lambda nos permite llamar a las funciones de la librería math

**command=lambda:bclick("degrees(acos"))**

Posicionamiento del botón

**botonarccos.place(x=1, y=143)**

Se crearon los botones con sus nombres correspondientes al igual que su función

**Botón para exponentes**

botonexpo=Button(ventana, text="EXP", width=an, height=al, command=lambda:bclick("\*\*"))

botonexpo.place(x=1, y=85)

**Botón para el seno**

botonsin=Button(ventana, text="Sin", width=an, height=al, command=lambda:bclick("sin(radians"))

botonsin.place(x=90, y=85)

**Botón para el coseno**

botoncos=Button(ventana, text="cos", width=an, height=al, command=lambda:bclick("cos(radians"))

botoncos.place(x=179, y=85)

**Baton para la tangente**

botontan=Button(ventana, text="tan", width=an, height=al, command=lambda:bclick("tan(radians"))

botontan.place(x=268, y=85)

**Botón para el seno inverso o arcoseno**

botonarcsin=Button(ventana, text="arcsin", width=an, height=al, command=lambda:bclick("degrees(asin"))

botonarcsin.place(x=359, y=85)

**Botón para el coseno inverso o arcocoseno**

botonarccos=Button(ventana, text="arccos", width=an, height=al, command=lambda:bclick("degrees(acos"))

botonarccos.place(x=1, y=143)

**Botón para el seno inverso o arcoseno**

botonarctan=Button(ventana, text="arctan", width=an, height=al, command=lambda:bclick("degrees(atan"))

botonarctan.place(x=90, y=143)

**Botón para la secante**

botonsec=Button(ventana, text="sec", width=an, height=al, command=lambda:bclick("1/cos(radians"))

botonsec.place(x=179, y=143)

**Botón para cosecante**

botoncsc=Button(ventana, text="csc", width=an, height=al, command=lambda:bclick("1/sin(radians"))

botoncsc.place(x=268, y=143)

**Botón para cotangente**

botoncot=Button(ventana, text="cot", width=an, height=al, command=lambda:bclick("1/tan(radians"))

botoncot.place(x=359, y=143)

**Botón raíz**

botonraiz=Button(ventana, text="√", width=an, height=al, command=lambda:bclick("sqrt"))

botonraiz.place(x=1, y=201)

**Botón logaritmo natural**

botonln=Button(ventana, text="ln", width=an, height=al, command=lambda:bclick("log"))

botonln.place(x=90, y=201)

**Botón logaritmo**

botonlog=Button(ventana, text="log", width=an, height=al, command=lambda:bclick("log10"))

botonlog.place(x=179, y=201)

**Botón parenthesis abierto**

botonpariz=Button(ventana, text="(", width=an, height=al, command=lambda:bclick("("))

botonpariz.place(x=268, y=201)

**Botón parenthesis cerrado**

botonparde=Button(ventana, text=")", width=an, height=al, command=lambda:bclick(")"))

botonparde.place(x=357, y=201)

**Boton 7**

boton7=Button(ventana, text="7", width=an, height=al, command=lambda:bclick(7))

boton7.place(x=1, y=259)

**Botón 8**

boton8=Button(ventana, text="8", width=an, height=al, command=lambda:bclick(8))

boton8.place(x=90, y=259)

**Botón 9**

boton9=Button(ventana, text="9", width=an, height=al, command=lambda:bclick(9))

boton9.place(x=179, y=259)

**Botón para eliminar todo**

botonAC=Button(ventana, text="AC", width=an, height=al, command=clear)

botonAC.place(x=268, y=259)

**Boton DEL, elimina cada elemento**

botonDEL=Button(ventana, text="DEL", width=an, height=al, command=lambda:bclick("DEL"))

botonDEL.place(x=357, y=259)

**Botón 4**

boton4=Button(ventana, text="4", width=an, height=al, command=lambda:bclick(4))

boton4.place(x=1, y=317)

**Botón 5**

boton5=Button(ventana, text="5", width=an, height=al, command=lambda:bclick(5))

boton5.place(x=90, y=317)

**Botón 6**

boton6=Button(ventana, text="6", width=an, height=al, command=lambda:bclick(6))

boton6.place(x=179, y=317)

**Botón multiplicación**

botonmultip=Button(ventana, text="x", width=an, height=al, command=lambda:bclick("\*"))

botonmultip.place(x=268, y=317)

**Botón división**

botondivi=Button(ventana, text="÷", width=an, height=al, command=lambda:bclick("/"))

botondivi.place(x=357, y=317)

**Botón 1**

boton1=Button(ventana, text="1", width=an, height=al, command=lambda:bclick(1))

boton1.place(x=1, y=375)

**Botón 2**

boton2=Button(ventana, text="2", width=an, height=al, command=lambda:bclick(2))

boton2.place(x=90, y=375)

**Botón 3**

boton3=Button(ventana, text="3", width=an, height=al, command=lambda:bclick(3))

boton3.place(x=179, y=375)

**Botón suma**

botonsuma=Button(ventana, text="+", width=an, height=al, command=lambda:bclick("+"))

botonsuma.place(x=268, y=375)

**Botón resta**

botonresta=Button(ventana, text="-", width=an, height=al, command=lambda:bclick("-"))

botonresta.place(x=357, y=375)

**Botón 0**

boton0=Button(ventana, text="0", width=an, height=al, command=lambda:bclick(0))

boton0.place(x=1, y=433)

**Botón coma**

botoncoma=Button(ventana, text=",", width=an, height=al, command=lambda:bclick("."))

botoncoma.place(x=90, y=433)

**Botón Pi**

botonpi=Button(ventana, text="π", width=an, height=al, command=lambda:bclick("pi"))

botonpi.place(x=179, y=433)

**Botón porcentaje**

botonporc=Button(ventana, text="%", width=an, height=al, command=lambda:bclick("%"))

botonporc.place(x=268, y=433)

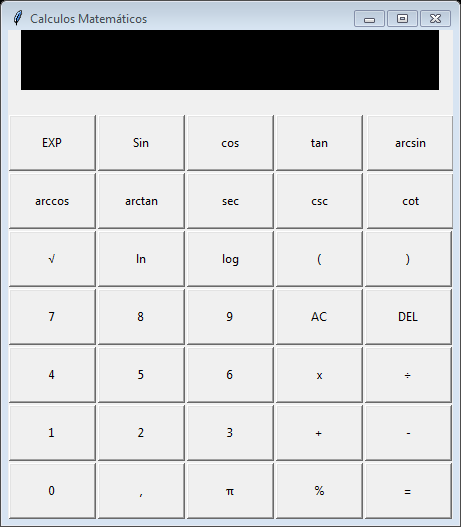
**Botón resultado**

botonresl=Button(ventana, text="=", width=an, height=al, command=operacion)

botonresl.place(x=357, y=433)

En este caso como la librería math ya nos proporciona las funciones no es necesario escribir ningúna formula

Caluladora



1. **calculadora científica**

superficie

from math import\*

a = int(input("Ingrese el valor de a:"))

b = int(input("Ingrese el valor de b:"))

c = int(input("Ingrese el valor de c:"))

p = (a+b+c)/2

print ("El perimetro es:", p)

s = (int(sqrt((p\*(p-a)\*(p-b)\*(p-c)))))

print("La superficie es:", s)

## Manejo de datos

En el proceso de elaboración del programa utilizamos la librería tkinter y math que ayudan al funcionamiento y los gráficos del programa. Este programa fue elaborado en el sistema operativo de Windows 10 y con versión de Python 3.5 y comprobado en Windows 7 con versión de Python 2.7

En la librería de Tkinter nos proporcionó la función de “bclick” esto para al momento de dar clic en el botón proporcione la función y numero deseado para el cálculo de la operación que se necesite.

Con ayuda del comando “Global” nos permite trabajar con decimales y con números enteros dentro de la calculadora para que así sea lo más completa que se pueda.

Con el comando “eval” nos ayuda a evaluar las variables sin la necesidad de guardar la operación dentro de una variable para al final imprimir la variable en la cual se está guardando la función.

El “str” en nuestro programa imprime el valor dado por el usuario, como, por ejemplo, para que el usuario vea que números se están ingresando dentro de la calculadora, también tiene el mismo funcionamiento cuando se le ingresa una operación al dar clic en ella

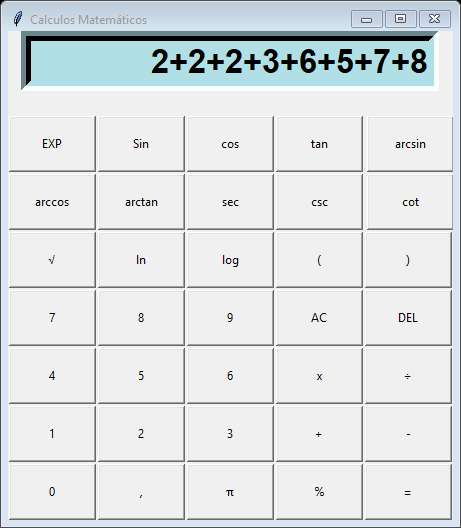
## Conclusión

El objetivo planteado para el proyecto fue realizado y logrado correctamente, todo fue elaborado en la manera en que pensábamos. La calculadora se comprobó y cada botón que utilizamos funciono a la perfección. Su elaboración fue sencilla ya que la librería math ya tenía todas las funciones que necesitamos por lo cual evitamos el uso de fórmulas.

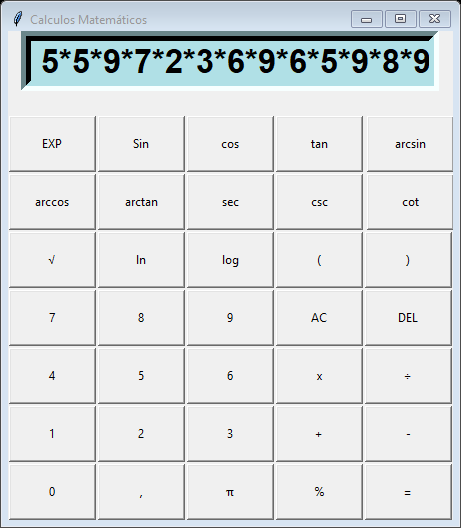
## Resultados

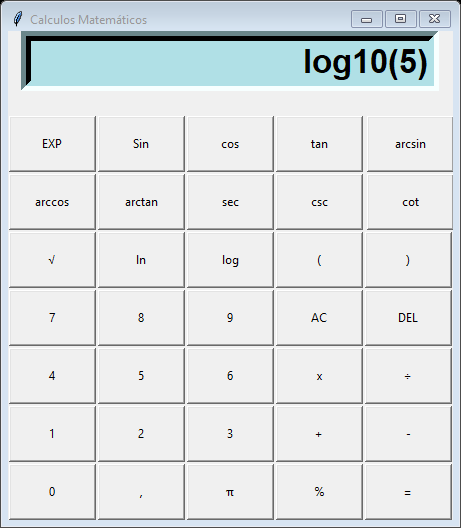
Existen algunas funciones en nuestro programa que nos permiten agregar cualquier la cantidad de números que la pantalla permite sin ningún problema, entre ellos se encuentra la suma, resta, multiplicación, así también se encuentran algunas funciones que permiten ingresar uno o dos datos entre ellas están las funciones trigonométricas, exponenciales, funciones logarítmicas, entre otras.

**Figura 3. Multiplicación. Fuente propia**



**Figura 2. Suma. Fuente propia**

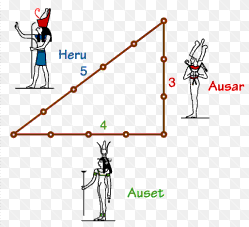




**Figura 4. Función logarítmica. Fuente propia**

Nuestro programa tiene integrado una opción para realizar el cálculo de la superficie de un polígono irregular.

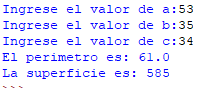
Para ello se necesita el valor de los tres lados del polígono en cuestión que se desea saber su superficie.



**Figura 5. Lados para calcular superficie. Fuente: concurso de programación**

Un problema que se puede resolver en nuestro programa es el siguiente:

Se tienen 3 lados de un polígono que son las distancias. Con este programa podemos conocer el perímetro y la superficie. En este caso las distancias a calcular son 3,4, 5 estos datos irán registrados en el programa para así tener el resultado.



**Figura 6. Resultados del cálculo de la superficie. Fuente propia**

Si el polígono es un hexágono y necesitamos la superficie de cada triangulo este programa nos ayudara a calcularlo, pero no la superficie de toda la figura sino de triángulos, la calculadora nos ayuda a sumar el resultado de la superficie de cada triángulo y así poder obtener la superficie total.

Delclaux, Isidoro; Seoane, Julio (1982). Psicología cognitiva y procesamiento de la información: teoría, investigación y aplicaciones. Madrid: Ediciones Pirámide, 1982.

Ellis, David (1992a). The physical and cognitive paradigms in Information Retrieval Re search. // Journal of Documenda d

Markey, Karen (1990). Keyword searching in an online catalog enhanced with a library classification. // Bengtson, Betty G.; Hill, Janet Swan (eds.). Classification of library materials: current and future potential for providing access. New York: Neal-Shuman Publishers, 1990. 99-125.

Sagredo Fernández, Félix; Espinosa Temiño, María Blanca (2000). Del libro, al libro electrónico-digital. // Cuadernos de Documentación Multimedia. 9 (2000).

Smith, Ph. J.; Beghtol, C.; Fidel, R.; Kwasnik, B. H. (eds.) (1993). Proceedings of the 4th ASIS SIG/CR Classification Research Workshop: Columbus, OH, Oct.24, 1993. Silver Spring, MD.: American Society for Information Science, 1993.

## Apéndic

Utilice esta página y las siguientes para situar tablas y figuras de tamaño superior al ancho de la columna, u otros materiales. Si no la utiliza, borre esta sección.