



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Nombre del alumno:

Luis Fernando Ochoa Angulo

Matricula:

372746

Semestre:

3ro- Computación

Grupo:

432

Materia:

Lenguaje de Programación Python

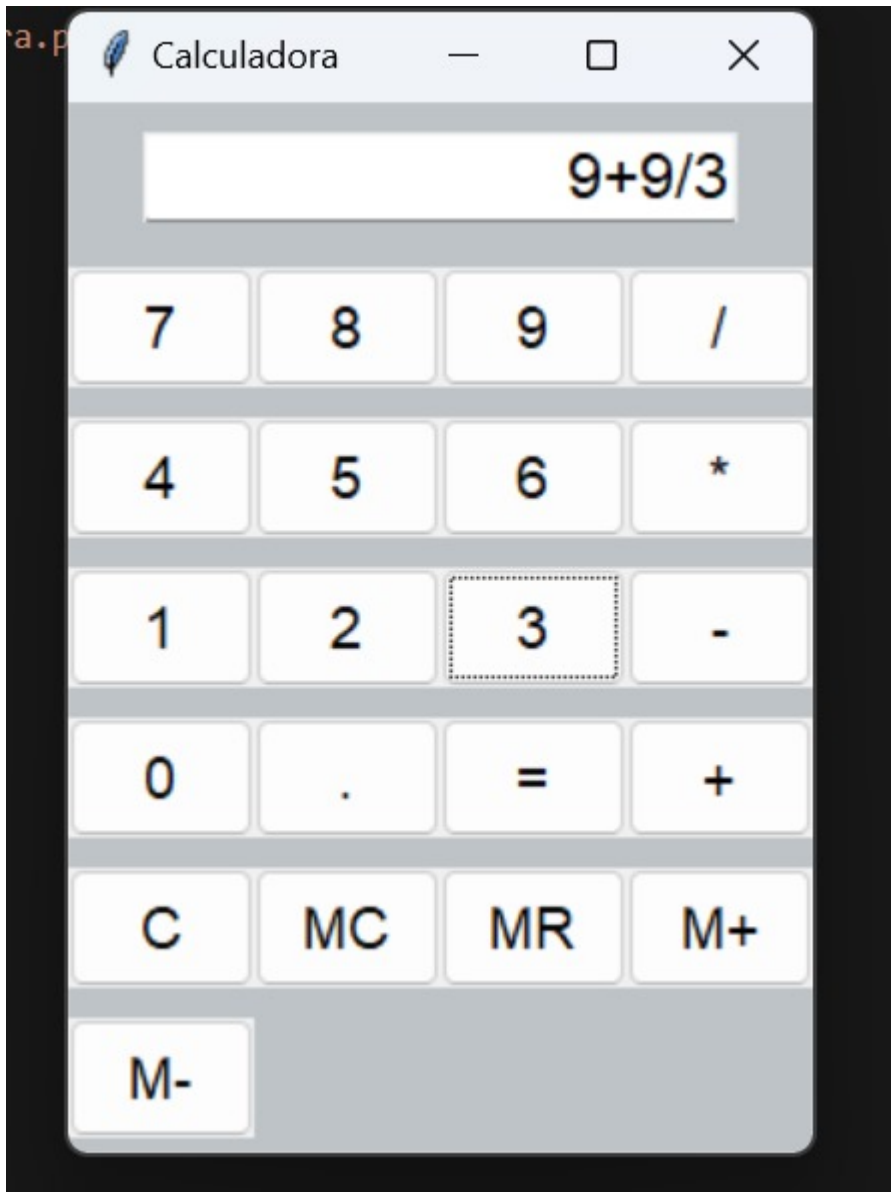
Actividad:

Actividad 12

Nombre del Profesor:

Pedro Nuñez Yepiz

Fecha:Viernes 24 de Noviembre del 2023

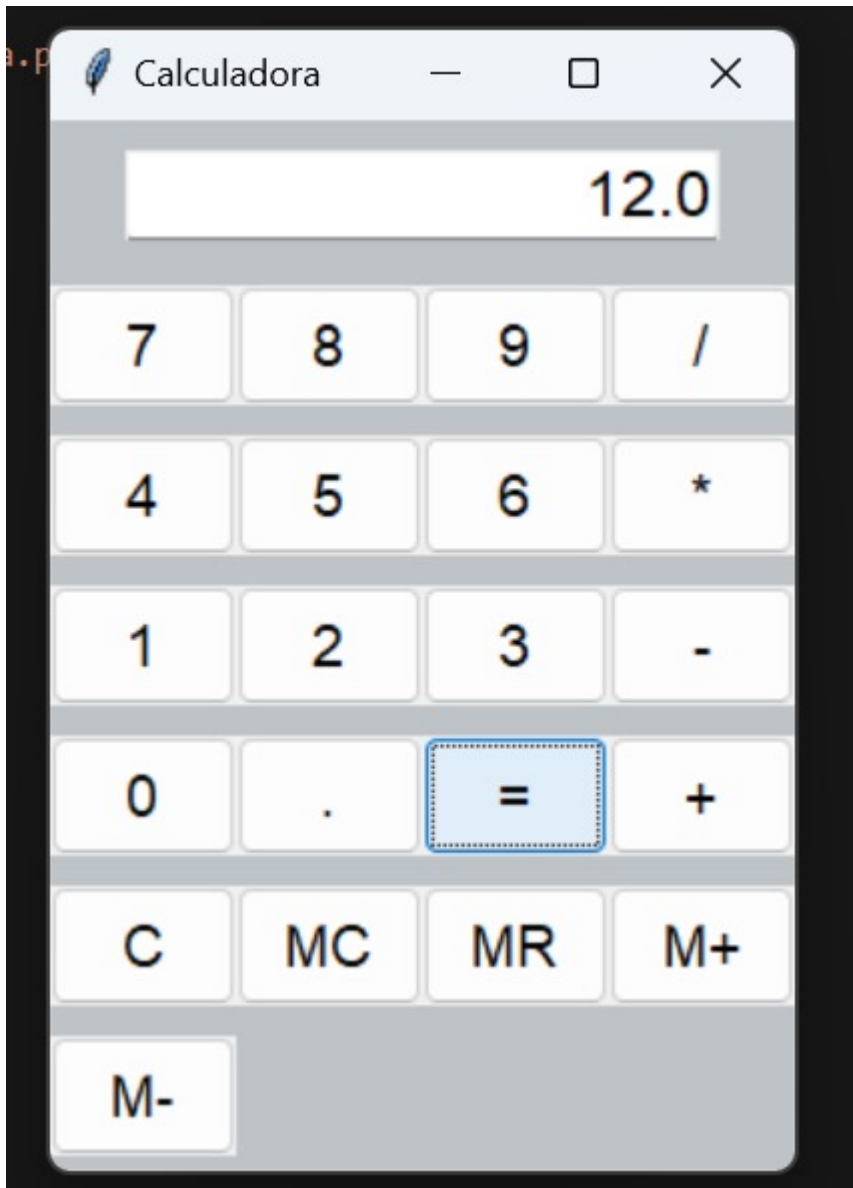




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño







Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
1 # Ensenada, Baja California a 24 de Noviembre del 2023
2 #Actividad 14
3 #REALIZA LOS EJERCICIOS EN PYTHON USANDO LIBRERIA TKINTER
4 #Nombre: Luis Fernando Ochoa Angulo
5 #Matricula: 372746
6 #Programa: calculadora usando tkinter
7 #Nombre de la actividad: LF04_AC14_01_492_PYTHON
8
9 import tkinter as tk
10 from tkinter import ttk
11
12 class Calculadora:
13     def __init__(self, master):
14         self.master = master
15         master.title("Calculadora")
16
17         # Configuración del estilo
18         self.style = ttk.Style()
19         self.style.configure('TButton', font=('Arial', 14), padding=5)
20         self.style.configure('TEntry', font=('Arial', 16))
21
22         # Color de fondo de la ventana
23         master.configure(bg='#bdc3c7')
24
25         self.entrada = ttk.Entry(master, width=16, font=('Arial', 16), justify='right', style='TEntry')
26         self.entrada.grid(row=0, column=0, columnspan=4, pady=10)
27
28         self.crear_botones()
29
30         self.valor_memoria = 0
31
32     def crear_botones(self):
33         botones = [
34             '7', '8', '9', '/',
35             '4', '5', '6', '*',
36             '1', '2', '3', '-',
37             '0', '.', '=', '+',
38             'C', 'MC', 'MR', 'M+', 'M-'
39         ]
40
41         fila = 1
42         columna = 0
43
44         for boton in botones:
45             ttk.Button(self.master, text=boton, width=4, command=lambda b=boton: self.manipular_boton(b)).grid(row=fila, column=columna, pady=5)
46             columna += 1
47             if columna > 3:
48                 columna = 0
49                 fila += 1
50
```



```
50
51 def manipular_boton(self, valor):
52     if valor == 'C':
53         self.borrar_entrada()
54     elif valor == '=':
55         self.realizar_calculo()
56     elif valor == 'M+':
57         self.memoria_sumar()
58     elif valor == 'M-':
59         self.memoria_restar()
60     elif valor == 'MR':
61         self.memoria_recordar()
62     elif valor == 'MC':
63         self.memoria_limpiar()
64     else:
65         actual = str(self.entrada.get())
66         self.entrada.delete(0, tk.END)
67         self.entrada.insert(tk.END, actual + str(valor))
68
69 def borrar_entrada(self):
70     self.entrada.delete(0, tk.END)
71
72 def realizar_calculo(self):
73     try:
74         resultado = eval(self.entrada.get())
75         self.entrada.delete(0, tk.END)
76         self.entrada.insert(tk.END, str(resultado))
77     except:
78         self.entrada.delete(0, tk.END)
79         self.entrada.insert(tk.END, "Error")
80
81 def memoria_sumar(self):
82     try:
83         self.valor_memoria += float(self.entrada.get())
84     except ValueError:
85         self.entrada.delete(0, tk.END)
86         self.entrada.insert(tk.END, "Error")
87
88 def memoria_restar(self):
89     try:
90         self.valor_memoria -= float(self.entrada.get())
91     except ValueError:
92         self.entrada.delete(0, tk.END)
93         self.entrada.insert(tk.END, "Error")
94
95 def memoria_recordar(self):
96     self.entrada.delete(0, tk.END)
97     self.entrada.insert(tk.END, str(self.valor_memoria))
98
99 def memoria_limpiar(self):
100     self.valor_memoria = 0
101     self.entrada.delete(0, tk.END)
102
103 if __name__ == "__main__":
104     raiz = tk.Tk()
105     calculadora = Calculadora(raiz)
106     raiz.mainloop()
```