

# Practica de POOE: Calculadora de Python

Baruch Mejía Martínez

Para la calculadora se definió una clase llamada calculadora con atributos que definen un entorno gráfico como se puede ver a continuación.

En el entorno se definieron dos frames y la sección principal. Un frame sirve únicamente para crear el fondo blanco de la pantalla, el otro es la pantalla en la que está definido el valor a mostrar. El valor a mostrar es definido como otro atributo el cual inicialmente se muestra como un '0'.

Además de los atributos ya mencionados, se declaran otros atributos que servirán para futuros métodos de la clase.

A continuación se muestra una captura de estos atributos:



```
self.display.config(bg='white')
self.display.config(width=350, height=50)
self.display.place(x=175,y=20)
self.res = Label(self.display, textvariable = self.equation,bg='white')
self.result = str(0)
self.equation.set(self.result)
self.ans = ''
self.res.grid(columnspan = 4)
self.numeros = []
```

Para generar el panel de números se definió otro frame con el objetivo de subdividirlo en un grid. Dentro de él se definieron botones que llaman a cierta función llamada presionar número:

```
self.panel = Frame()
self.panel.config(bg='white')
self.one = Button(self.panel, text = "1", height = 4, width = 7,command = lambda:self.presionarBoton(1), bg = "gray")
self.one.grid(row = 1, column = 0)
```

Para los botones de operación se definieron en el frame y se llamaron a sus respectivas funciones

```
self.botonSum = Button(self.panel, text="+",height = 4, width = 7, command=self.sumar)
self.botonSum.grid(row = 3, column = 3)
self.botonCancel = Button(self.panel, text="AC",height = 4, width = 7, command=self.restart, bg = "red")
self.botonCancel.grid(row = 1, column = 4)
self.botonDel = Button(self.panel, text="Del",height = 4, width = 7, command=self.delete, bg = "red")
self.botonDel.grid(row = 1, column = 3)
self.botonRetar = Button(self.panel, text="-",height = 4, width = 7, command=self.restar)
self.botonRetar.grid(row = 3, column = 4)
self.botonMulti = Button(self.panel, text="x",height = 4, width = 7, command=self.multiplicar)
self.botonMulti.grid(row = 2, column = 3)
self.botonDividir = Button(self.panel, text="÷",height = 4, width = 7, command=self.dividir)
```

Para resolver los cálculos se requirió separar la cadena en caracteres en una lista y detectar cada símbolo para saber que operación realizar:

```
def calcular(self):
    self.result += ' '
    self.result = self.result.split(" ")
    final = False
    i = 0
    while final != True:
        print(self.result)
        if self.result[i] == 'ñ':
            self.result.pop(i)
            final = True
        else:
            if self.result[i] == 'x':
                res = float(self.result[i-1])*float(self.result[i+1])
                self.result[i] = res
                self.result.pop(i-1)
                self.result.pop(i)
                i -= 1
            elif self.result[i] == '÷':
                res = float(self.result[i-1])/float(self.result[i+1])
                self.result[i] = res
                self.result.pop(i-1)
                self.result.pop(i)
                i -= 1
            elif self.result[i] == '+':
                res = float(self.result[i-1])+float(self.result[i+1])
```