CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENERIAS

Aldo Guillermo Román Del Muro

INCO

217554557

Reporte programa utilizando hilos.

Computación Tolerante a fallas.

DR. MICHEL EMANUEL LOPEZ FRANCO.

Introducción:

Utilizar la librería de threading en Python se utiliza para trabajar con hilos, estos son unidades de ejecución que más pequeñas dentro de un proceso que nos ayudan que un programa realice múltiples tareas simultáneamente o lo que nosotros conocemos como concurrencia.

Esto nos ayuda a optimizar un poco más nuestro programa utilizando un poco más de CPU y sus núcleos, nos permite también ejecutar varias tareas al mismo tiempo dentro del programa, etc.

Desarrollo:

En este caso será un caso muy sencillo, utilizaré un programa de una calculadora utilizando la librería de tkinter que nos permitirá tener una interfaz gráfica dentro del programa.

```python
def suma():
    try:
        num1 = float(entry_num1.get())
        num2 = float(entry_num2.get())
        resultado.set(num1 + num2)
    except ValueError:
        messagebox.showerror("Error", "Dato no valido.")

def resta():
    try:
        num1 = float(entry_num1.get())
        num2 = float(entry_num2.get())
        resultado.set(num1 - num2)
    except ValueError:
        messagebox.showerror("Error", "Dato no valido.")

def multiplicacion():
    try:
        num1 = float(entry_num1.get())
        num2 = float(entry_num2.get())
        resultado.set(num1 * num2)
    except ValueError:
        messagebox.showerror("Error", "Dato no valido.")
```

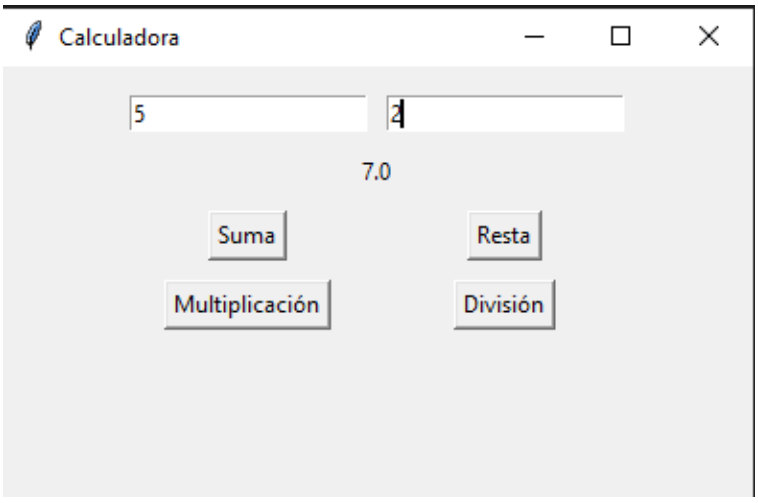
Este es un pequeño ejemplo de la estructura de cada función dentro del programa de las operaciones básicas que va a realizar nuestro programa, no son nada fuera de lo común únicamente solicitaremos dentro de la interfaz dos números al usuario y estos dos números los usaremos dentro de la función y mostraremos un resultado, en caso de que nuestro usuario no introduzca bien los números mostraremos el mensaje de que el valor introducido no es válido y que lo intente nuevamente.

```
def realizar_operacion(operacion):
    threading.Thread(target=operacion).start()
```

Ahora declaramos la función realizar_operacion y le pasaremos como parámetro nuestra operación a realizar, es decir la operación que el usuario solicite dentro del programa. Creamos un nuevo hilo y se le asignará la tarea a ejecutar el parámetro seleccionado, utilizando la función de. start() hacemos un llamado al hilo recién creado, lo que ejecutará la función de la operación en segundo plano en paralelo a la interfaz.

Ahora veamos cómo funciona.

Aquí tenemos nuestra interfaz y veamos cómo funciona.



Realmente si no es a tiempo real o viéndose en persona la ejecución no se notaría una mejora, pero dentro del funcionamiento del programa si se nota, se nota en la fluidez dentro de nuestra interfaz y de cómo funciona.

Conclusiones:

Dentro de la ejecución del programa insisto, tendría que verse en persona, pero como una breve descripción, dentro de la interfaz al momento de introducir los valores numéricos y solicitar la solución usando cualquiera de las operaciones básicas. Si hiciéramos el uso de los hilos nuestros cálculos se realizarían dentro de del botón, haciendo que la interfaz de bloquee y que el usuario no pueda interactuar con la interfaz hasta que se complete la operación, es decir mientras las operaciones se ejecutan en un segundo plano en hilos separados nuestra interfaz sigue siendo usable y manipulable por el usuario.