Trabajo Encargado

Alumna:
Esther Milagros Bautista Peralata

Semestre: *VIII - semestre*

Docente:	*Fred Torres Cruz*
Facultad:	Ingeniería Estadística e Informática
Año académico:	October 31, 2022- II

Contents

1 Ejercicio n°1

Se desea evaluar una función que calcula el área bajo la función $y = x + 1$, calcule usted la diferencia de implementación de un algoritmo lineal, y utilizando threads.

Probar para los intervalos $[1, 5]$
Probar para los intervalos $[3, 8]$

Ayuda:

Figure 1

Solucíon

```
import time
import scipy.integrate as integrate
from numpy import *

def proceso():
    time_ini = time.time()
    print("---- Inicio ----")
    time.sleep(1)

    print("Ingrese los intervalos [a, b] ")
    a = float(input("Ingrese el valor de a: "))
    b = float(input("Ingrese el valor de b: "))
    problem_one = integrate.quad(lambda x: x + 1, a, b)
    result_one = problem_one[0]
    print("El área bajo la curva es: ", result_one)

    print("------ Fin -----")
    time_end = time.time()
    total = time_end - time_ini
    print("El tiempo de ejecución es:",total)

for i in range(1):
```

```
    proceso()
```

IMPLEMENTACIÓN DE UN ALGORITMO LINEAL CON GRÁFICO

```
from matplotlib.pyplot import *
from numpy import *

x = arange(0, 6, 0.001)
y = x + 1
plot(x, y)
fill_between(x,y,where=(1<=x)&(x<=5),color='g', alpha = 0.5)
grid()

from sympy import *
x=symbols('x')
y = x + 1
res = integrate(y,(x,1,5))
print(f'El área bajo la curva es:  {res}')
show()
```

IMPLEMENTACIÓN DE UN ALGORITMO LINEAL USANDO HILOS (THREAD)

```
from threading import Thread
import time
import scipy.integrate as integrate
from numpy import *

def area(a, b):
    time_ini = time.time()

    print("---- Inicio ----")
    time.sleep(1)

    problem_one = integrate.quad(lambda x: x + 1, a, b)
    result_one = problem_one[0]
    print("El área bajo la curva es: ", result_one)

    print("------ Fin -----")
    time_end = time.time()
    total = time_end - time_ini
    print("El tiempo de ejecución es:",total)

if __name__ == '__main__':
    #distancia1()
    print("Ingrese los intervalos [a, b] ")
    a = float(input("Ingrese el valor de a: "))
    b = float(input("Ingrese el valor de b: "))
```

```
thread1 = Thread(target=area, args= (a, b))
thread1.start()
thread1.join()
```

1.1 Salida de pantalla para los intervalos $[1, 5]$

El codigo se ejecutó en `GOOGLE COLAB`.

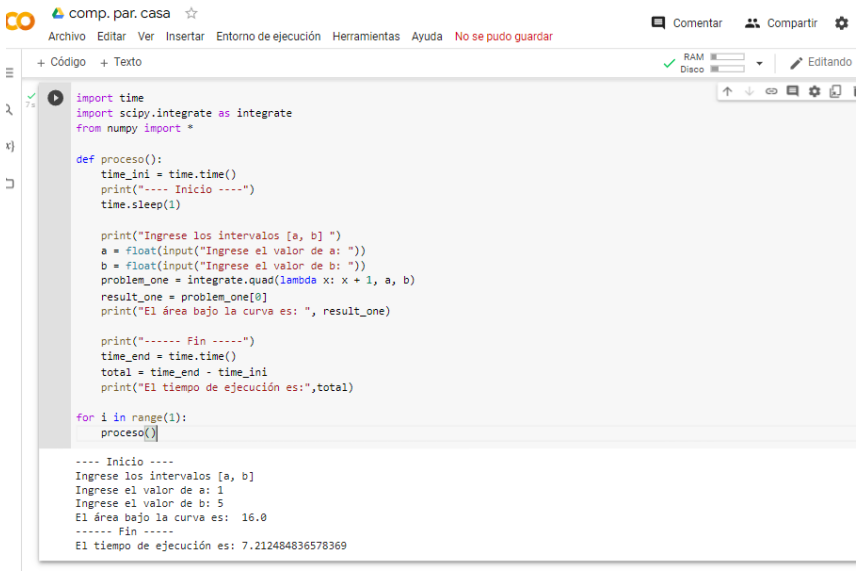

Figure 2: Ejecutando el código para los intervalos [1, 5]

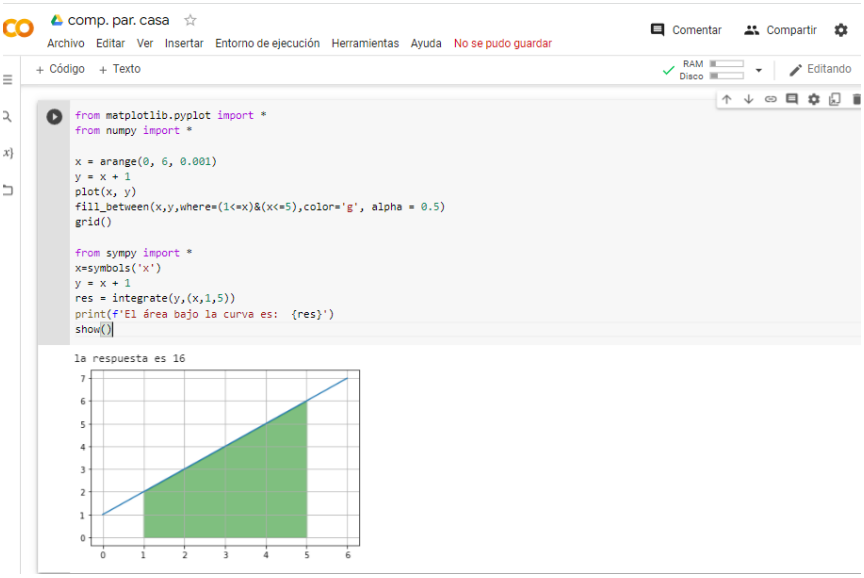

Figure 3: Ejecutando el código para los intervalos [1, 5]

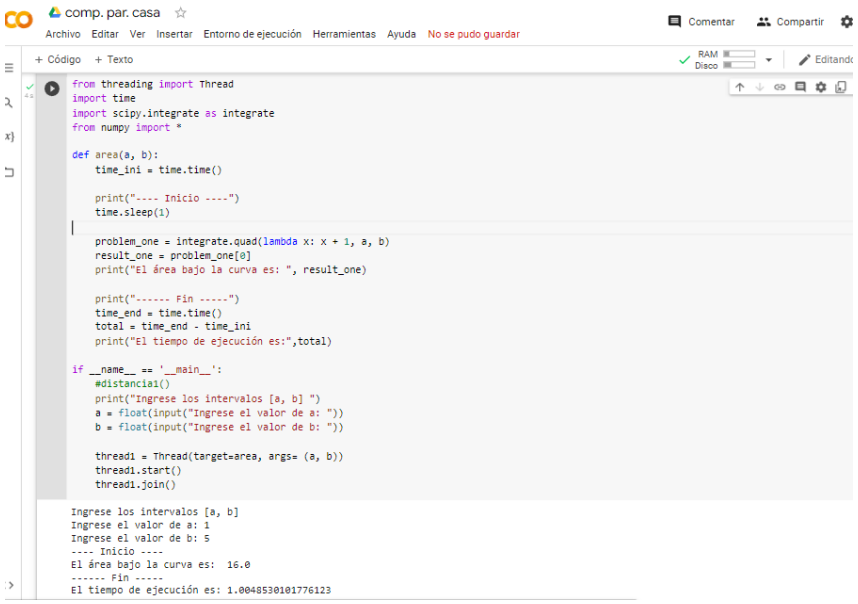

Figure 4: Ejecutando el código para los intervalos [1, 5]

1.2 Salida de pantalla para los intervalos $[3, 8]$

Figure 5: Ejecutando el código para los intervalos [3, 8]

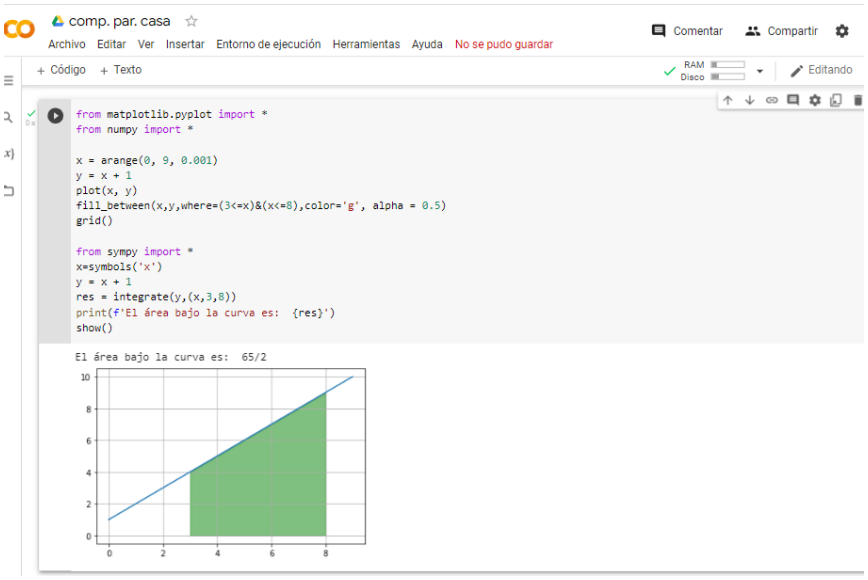

Figure 6: Ejecutando el código para los intervalos [3, 8]

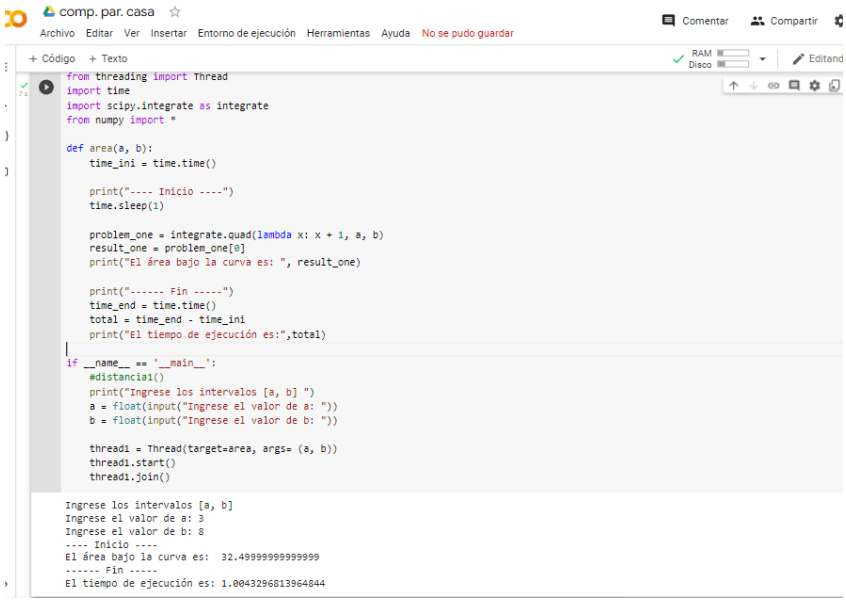

Figure 7: Ejecutando el código para los intervalos [3, 8]

2 Conclusíon y Aplicación

La diferencia más notoria es el tiempo de ejecución en los códigos, podemos visualizar las salidas de pantallas para los intérvalos $[1, 5]$ en la figura 2 que el tiempo de ejecución fué de 7.2124... segundos y usando thread (hilos) podemos visualizar en la figura 4 que el tiempo fué de 1.0048... segundos, el mismo caso se da para los intérvalos $[3, 8]$.

Aplicaciones con Hilos, hay un sin numero de ellos, por ejemplo cuando trabaja el MP3, la ejecución de Voz por IP, reproducción de video, La aplicaciones bancarias, incluso el refrigerador de la casa, la radio, la televisión etc... muchos de ello trabaja con sistemas o sub-sistemas haciendo tareas en paralelo y coordinándose para proveer el servicio al usuario.

La parte interesante del manejo de hilos, bueno cuando se tiene recursos compartidos se implementa o utiliza los esquemas de bloqueo(Semáforos) y sincronización es ahí donde esta el arte de programar con hilos ya que de no hacerlo bien se puede crear un sistema totalmente ineficiente o inútil, por ejemplo aplicaciones que tardan horas en procesar los servicios o que se bloquean y que intercambian los datos equivocadamente.
Por ello hay que revisar bien los esquemas y que es lo que se quiere lograr, diseñar con grafos puede ayudar a verificar las dependencias a recursos compartidos. En el caso de las aplicaciones WEB, generalmente el uso de procesos en paralelo esta coordinado por los contenedores y los programadores ya que no tienen que preocuparse mucho por dicha coordinación, excepto en casos muy especiales.

3 Evidencia del trabajo en LATEX

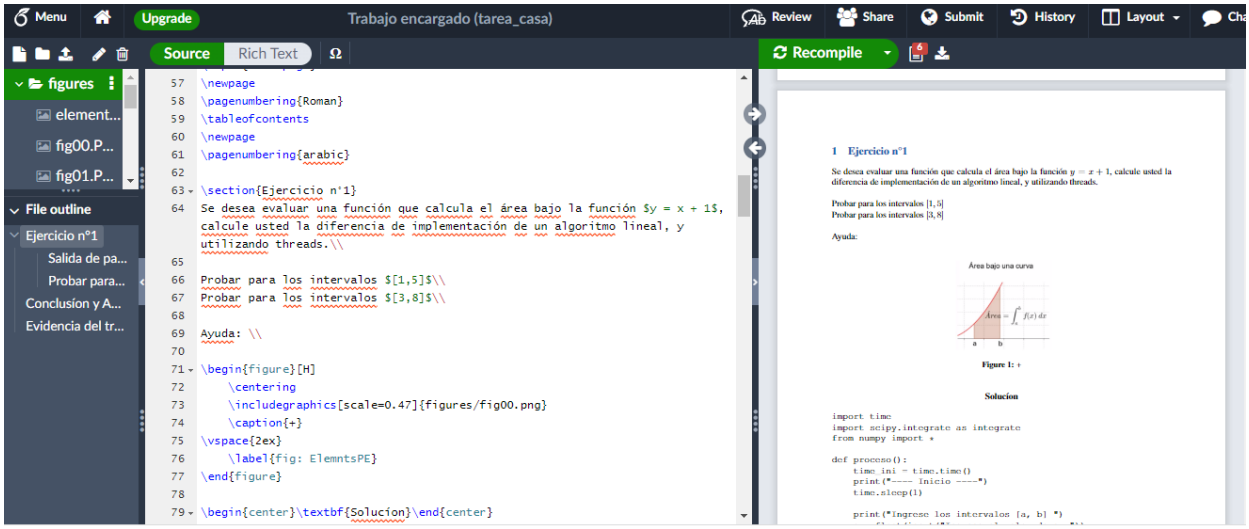

Figure 8: https://www.overleaf.com/read/dfkmgqkrhnsx