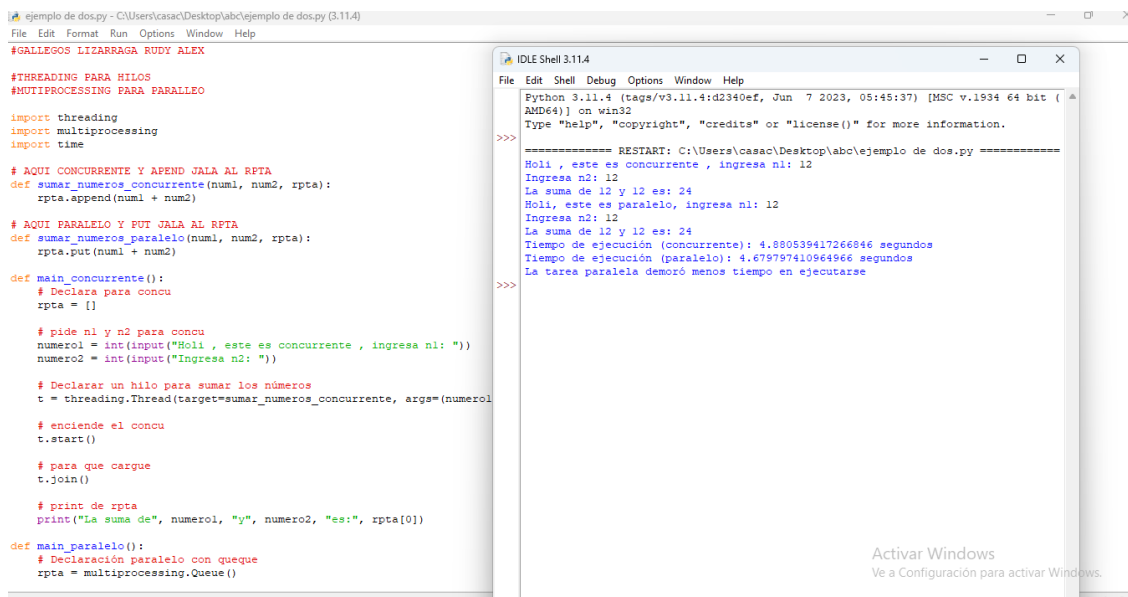


Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Docente: Fred Torres Cruz
Autor : Gallegos Lizarraga Rudy Alex

Trabajo Encargado - N° 001

COMPARACIÓN DE ALGORITMOS PARALELOS Y CONCURRENTES CON TIME

En este ejemplo muestra un código con hilos y otro con paralelismo y se muestra la diferencia mediante la función time que calcula la diferencia de tiempo entre dos puntos específicos en el código.



The image shows two windows from the IDLE Python 3.11.4 environment. The left window displays a Python script named 'ejemplo de dos.py' with the following code:

```
#GALLEGOS LIZARRAGA RUDY ALEX

#THREADING PARA HILOS
#MULTIPROCESSING PARA PARALELO

import threading
import multiprocessing
import time

# AQUI CONCURRENTENTE Y APEND JALA AL RPTA
def sumar_numeros_concurrente(num1, num2, rpta):
    rpta.append(num1 + num2)

# AQUI PARALELO Y PUT JALA AL RPTA
def sumar_numeros_paralelo(num1, num2, rpta):
    rpta.put(num1 + num2)

def main_concurrente():
    # Declara para concu
    rpta = []

    # pide n1 y n2 para concu
    numero1 = int(input("Hola , este es concurrente , ingresa n1: "))
    numero2 = int(input("Ingresa n2: "))

    # Declarar un hilo para sumar los números
    t = threading.Thread(target=sumar_numeros_concurrente, args=(numero1, numero2, rpta))

    # enciende el concu
    t.start()

    # para que cargue
    t.join()

    # print de rpta
    print("La suma de", numero1, "y", numero2, "es:", rpta[0])

def main_paralelo():
    # Declaración paralelo con queue
    rpta = multiprocessing.Queue()
```

The right window shows the output of the script after execution:

```
>>>
===== RESTART: C:\Users\casac\Desktop\abc\ejemplo de dos.py =====
Hola , este es concurrente , ingresa n1: 12
Ingresa n2: 12
La suma de 12 y 12 es: 24
Hola, este es paralelo, ingresa n1: 12
Ingresa n2: 12
La suma de 12 y 12 es: 24
Tiempo de ejecución (concurrente): 4.880539417266846 segundos
Tiempo de ejecución (paralelo): 4.679797410964966 segundos
La tarea paralela demoró menos tiempo en ejecutarse
>>>
```

Figura 1: CODIGO



Figura 2: Codigo QR