



UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO

FACULTAD DE INGENIERIA ESTADISTICA E
INFORMATICA

IMPLEMENTACIÓN DE FUNCIONES CON THREADS

PRESENTA

Danny Ronalds Quispe Ajrota

PROFESOR

ING. Fred Torres Cruz

ASIGNATURA

COMPUTACION PARALELA

17 de octubre de 2022

IMPLEMENTACIÓN DE FUNCIONES CON THREADS

Danny Ronalds Quispe Ajrota

Universidad Nacional del Altiplano

daquispea@est.unap.edu.pe

Puno-Perú

0.1. EJERCICIO

Implementar según el ejemplo de clase una función numPar que reciba un parámetro n y calcule el números pares entre 1 y n, así mismo realizar el cálculo de tiempo para la evaluación de la implementación, para hacer el llamado a la función se deberá hacer a través de uno o más threads.

Las pruebas unitarias deberán ser con los siguientes parámetros a la función numPar:

n=10

n=30

n=200

n=5030

CÓDIGO FUENTE DE LA SOLUCIÓN

```
import threading
import time

def numPar(n):
    time_ini = time.time()
    i = 0
    cont = 0
    print("Los n meros pares son: ")
    for i in range(n):
        if i % 2 == 0:
            print(i)
            cont += 1
    print("Hay",cont,"n meros pares")

    time.sleep(1)
    time_end = time.time()
    total = time_end - time_ini
    print("El tiempo de ejecuci n es:",total)

if __name__ == '__main__':
    thread = threading.Thread(target=numPar, args= (10, ))
    thread.start()
    thread.join()
```

```
D: > INGENIERIA ESTADISTICA E INFORMATICA UNA PUNO > COMPUTACION PARALELA > TRABAJO 1
1  ∨ import threading
2  import time
3
4  ∨ def numPar(n):
5      time_ini = time.time()
6      i = 0
7      cont = 0
8      print("Los números pares son: ")
9  ∨  for i in range(n):
10 ∨  |     if i % 2 == 0:
11      |         print(i)
12      |         cont += 1
13      print("Hay",cont,"números pares")
14
15      time.sleep(1)
16      time_end = time.time()
17      total = time_end - time_ini
18      print("El tiempo de ejecución es:",total)
19
20 ∨ if __name__ == '__main__':
21     thread = threading.Thread(target=numPar, args= (10, ))
22     thread.start()
23     thread.join()
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

```
Hay 5 números pares
El tiempo de ejecución es: 1.0162036418914795
PS C:\Users\danny> & C:/Users/danny/AppData/Local/Programs/Python/Python310/python
TRABAJO ENCARGADO 01/THREADS +COD.py"
Los números pares son:
0
2
4
6
8
Hay 5 números pares
El tiempo de ejecución es: 1.0134165287017822
PS C:\Users\danny>
```

Figura 1: visual studio code/Python

INTERPRETACIÓN

Este algoritmo logra que según al ejemplo mostrado en clase se implemente la función numPar calcule numeros pares entre 1 y n esto se logró implementar con un solo Thread.

Se le puede agregar un parametro "n", El cual en la terminal muestra los numero pares dentro del parametro establecido esto podemos establecerlo desde nuestro thread.

```
20 if __name__ == '__main__':
21     thread = threading.Thread(target=numPar, args= (10, ))
22     thread.start()
23     thread.join()
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

PS C:\Users\danny>
PS C:\Users\danny>
PS C:\Users\danny> & C:/Users/danny/AppData/Local/Programs/Python/Python38-32/Python.exe TRABAJO_ENCARGADO_01/THREADS +COD.py
Los números pares son:
0
2
4
6
8
Hay 5 números pares
El tiempo de ejecución es: 1.017092227935791
PS C:\Users\danny> []

Figura 2: visual studio code/Python/10 datos

```
20 if __name__ == '__main__':
21     thread = threading.Thread(target=numPar, args= (30, ))
22     thread.start()
23     thread.join()
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

10
12
14
16
18
20
22
24
26
28
Hay 15 números pares
El tiempo de ejecución es: 1.018472671508789
PS C:\Users\danny> []

Figura 3: visual studio code/Python/30 datos

```
20 if __name__ == '__main__':
21     thread = threading.Thread(target=numPar, args= (200, ))
22     thread.start()
23     thread.join()
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

180
182
184
186
188
190
192
194
196
198
Hay 100 números pares
El tiempo de ejecución es: 1.0286107063293457
PS C:\Users\danny> []

Figura 4: visual studio code/Python/200 datos

```
20 if __name__ == '__main__':
21     thread = threading.Thread(target=numPar, args= (1000, ))
22     thread.start()
23     thread.join()
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

980
982
984
986
988
990
992
994
996
998
Hay 500 números pares
El tiempo de ejecución es: 1.0888659954071045
PS C:\Users\danny> []

Figura 5: visual studio code/Python/1000 datos