|  |  |  |  |
| --- | --- | --- | --- |
| **APELLIDO PATERNO** | **APELLIDO MATERNO** | **NOMBRES** | **MATERIA** |
| **LUCANA** | **HUALLPA** | **ANTONIO JOSE** | **INF - 317** |

**13. Realice el cálculo de Fibonacci con Multiprocessing de al menos 1000 términos divididos por procesador de manera dinámica**

import multiprocessing

def calcular\_fibonacci(n):

# Calcula la secuencia de Fibonacci hasta el término n

fib = [0, 1]

for i in range(2, n):

fib.append(fib[-1] + fib[-2])

return fib

def main():

num\_procesadores = multiprocessing.cpu\_count() # Número de procesadores

terminos\_por\_procesador = 1000 // num\_procesadores # Términos por procesador

pool = multiprocessing.Pool()

resultados = pool.map(calcular\_fibonacci, [terminos\_por\_procesador] \* num\_procesadores)

pool.close()

pool.join()

# Concatenar los resultados de cada procesador

fibonacci\_completo = [numero for sublist in resultados for numero in sublist]

# Escribir la secuencia de Fibonacci en un archivo de texto

with open("fibonacci.txt", "w") as archivo:

for numero in fibonacci\_completo:

archivo.write(f"{numero}\n")

print("Secuencia de Fibonacci exportada a fibonacci.txt")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**RESULTADO**

|  |  |
| --- | --- |
|  |  |