

## TAREA ACADÉMICA N°5

### Arqtel

<u>Código</u>	<u>Alumno</u>	<u>Participación</u>
20122509	Eduardo Alonso Sanchez Marcelo	20%
20190057	Ricardo Alfonso Alvarado Ruiz	20%
20193733	Jose Luis Morillos Pinedo	20%
20203521	Julio César Aliaga Machuca	20%
20200825	Maricielo Romo Reyes	20%
Total		100%

### Análisis de los tiempos de los códigos:

#### 1. Función síncrona sin cambios:

```
Line #    Mem usage    Increment    Occurrences    Line Contents
=====
     9      27.4 MiB      27.4 MiB         1    @profile
    10                      def get_all(sites: list[str]):
    11      30.4 MiB       0.0 MiB         2        with requests.Session() as session:
    12      30.4 MiB       0.0 MiB        161            for url in sites:
    13      30.4 MiB       3.1 MiB        160                get_site(url, session)

Filename: scraper_normal.py

Line #    Mem usage    Increment    Occurrences    Line Contents
=====
    15      27.4 MiB      27.4 MiB         1    @profile
    16                      def main():
    17      30.4 MiB       3.1 MiB         1        get_all(sites)

Descarga completa síncrona en 3.329765168999984 segundos
```

#### 2. Función asíncrona (Con la librería asyncio):

```
Line #    Mem usage    Increment    Occurrences    Line Contents
=====
    11      27.2 MiB      27.2 MiB         1    @profile
    12                      async def get_all(sites: list[str]):
    13      30.3 MiB       0.3 MiB         2        with requests.Session() as session:
    14      30.3 MiB       0.0 MiB        161            for url in sites:
    15      30.3 MiB       2.8 MiB        320                await asyncio.gather(get_site(url, session))

Filename: scraper_asyncio.py

Line #    Mem usage    Increment    Occurrences    Line Contents
=====
    17      27.2 MiB      27.2 MiB         1    @profile
    18                      async def main():
    19      30.3 MiB       3.1 MiB         2        await asyncio.gather(get_all(sites))

Descarga completa asíncrona en 3.130153295999662 segundos
```

### 3. Función con hilos (Con la librería threading):

```
Line #    Mem usage    Increment    Occurrences    Line Contents
=====
11      27.3 MiB      27.3 MiB         1    @profile
12                                     def get_all(sites: list[str]):
13                                     with requests.Session() as session:
14                                     for url in sites:
15                                     thread2 = Thread(target=get_site, args=(url, session, ))
16                                     thread2.start()
17                                     thread2.join()

Filename: scraper_threads.py

Line #    Mem usage    Increment    Occurrences    Line Contents
=====
19      27.3 MiB      27.3 MiB         1    @profile
20                                     def main():
21                                     thread1 = Thread(target=get_all, args=(sites, ))
22                                     thread1.start()
23                                     thread1.join()

Descarga completa con multihilos en 1.075844123999559 segundos
```

### 4. Función con multiprocesos (Con la librería multiprocessing):

```
Line #    Mem usage    Increment    Occurrences    Line Contents
=====
11      20.8 MiB      20.8 MiB         1    @profile
12                                     def get_all(sites: list[str]):
13                                     with requests.Session() as session:
14                                     for url in sites:
15                                     process2 = Process(target=get_site, args=(url, session, ))
16                                     process2.start()
17                                     process2.join()

Filename: scraper_mprocess.py

Line #    Mem usage    Increment    Occurrences    Line Contents
=====
19      27.3 MiB      27.3 MiB         1    @profile
20                                     def main():
21                                     process1 = Process(target=get_all, args=(sites, ))
22                                     process1.start()
23                                     process1.join()

Descarga completa con multiprocesos en 13.353438699999997 segundos
```

#### **NOTA:**

Se intentó hacer uso del comando “mprof plot” tal y como señalaba la página guía de “memory profiling”, sin embargo, no se obtuvo ningún resultado a la hora de ejecutar este comando.

```
● ubuntu@ip-172-31-23-45:~$ mprof plot
Using last profile data.
○ ubuntu@ip-172-31-23-45:~$
```