

## Actividad Integradora 5.3 Resaltador de sintaxis paralela

### Problemática productor/consumidor

#### Utilizando 1 hilo :

Tiempo : 33.4655

```
(0 ) Getting 00.scn : 9 items in queue
(0 ) Getting 12.scn : 8 items in queue
(0 ) Getting 04.scn : 7 items in queue
(0 ) Getting 10.scn : 6 items in queue
(0 ) Getting 11.scn : 5 items in queue
(0 ) Getting 05.scn : 4 items in queue
(0 ) Getting 01.scn : 3 items in queue
(0 ) Getting 02.scn : 2 items in queue
(0 ) Getting 03.scn : 1 items in queue
No. hilos: 1 - Tiempo: 33.46556 seg.
```

#### Utilizando 5 hilos :

Tiempo : 2.8242

```
(producer ) === NO ITEMS LEFT TO INSERT ===
(4 ) Getting 01.html : 8 items in queue
(4 ) Getting 04.scn : 7 items in queue
(1 ) Getting 10.scn : 6 items in queue
(3 ) Getting 11.scn : 5 items in queue
(0 ) Getting 05.scn : 4 items in queue
(2 ) Getting 01.scn : 3 items in queue
(4 ) Getting 02.scn : 2 items in queue
(3 ) Getting 03.scn : 1 items in queue
No. hilos: 5 - Tiempo: 2.82454 seg.
```

### Speedup

$$Sp = \frac{T_1}{T_p} \rightarrow \frac{33.4655 s}{2.8242 s} \rightarrow 11.8495$$

### Complejidad y reflexión

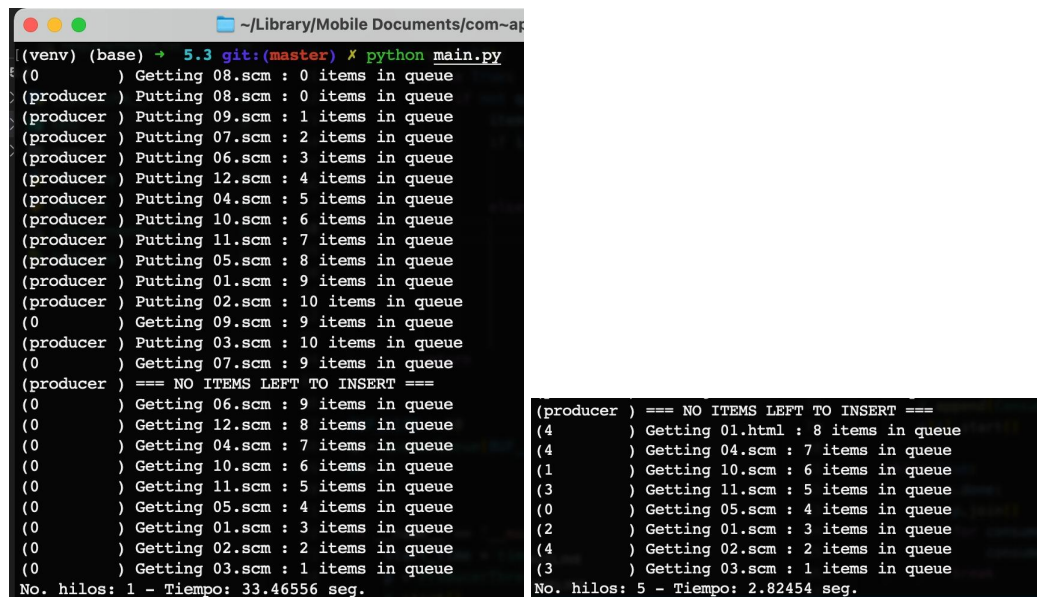
En la entrega anterior concluimos que la complejidad de nuestro programa era de  $O(n^2)$  debido a las iteraciones sobre cada línea del archivo y la función con while que analiza dichas líneas. Considerando que en esta implementación usamos la misma función, la complejidad del programa no cambia. Sin embargo, el tiempo de ejecución es mucho más rápido por la forma en que aprovechamos los recursos de la computadora. Al hacerlo secuencial el programa va leyendo y analizando cada archivo en orden. Implementando hilos, estos pueden leer varios archivos y clasificarlos al mismo tiempo. Esto lleva a que la ejecución sea mucho más rápida. Si bien directamente esto es muy beneficioso, si se tienen que leer archivos muy largos o muchos archivos igualmente podría aumentar el tiempo de ejecución de manera  $O(n^2)$ .

En el desarrollo de la tecnología siempre se implican temas éticos, cada que se presenta una tecnología nueva es posible que se utilice de manera ética o de manera no ética y que de esta manera dañe a terceros. En el caso de nuestro resaltador de sintaxis

paralela no observamos una forma directa en la que se pueda utilizar de manera no ética, sin embargo si hablamos de las herramientas tecnológicas que utilizamos dentro de nuestro proyecto podemos notar que son herramientas demasiado útiles para optimizar el tiempo de ejecución de programas, por lo tanto si se usa el paralelismo de manera no ética podría ocasionar grandes problemas.

## Liga Proyecto Github

<https://github.com/pablo-blancoc/TC2037>



```
(venv) (base) → 5.3 git:(master) X python main.py
(0 ) Getting 08.scm : 0 items in queue
(producer ) Putting 08.scm : 0 items in queue
(producer ) Putting 09.scm : 1 items in queue
(producer ) Putting 07.scm : 2 items in queue
(producer ) Putting 06.scm : 3 items in queue
(producer ) Putting 12.scm : 4 items in queue
(producer ) Putting 04.scm : 5 items in queue
(producer ) Putting 10.scm : 6 items in queue
(producer ) Putting 11.scm : 7 items in queue
(producer ) Putting 05.scm : 8 items in queue
(producer ) Putting 01.scm : 9 items in queue
(producer ) Putting 02.scm : 10 items in queue
(0 ) Getting 09.scm : 9 items in queue
(producer ) Putting 03.scm : 10 items in queue
(0 ) Getting 07.scm : 9 items in queue
(producer ) === NO ITEMS LEFT TO INSERT ===
(0 ) Getting 06.scm : 9 items in queue
(0 ) Getting 12.scm : 8 items in queue
(0 ) Getting 04.scm : 7 items in queue
(0 ) Getting 10.scm : 6 items in queue
(0 ) Getting 11.scm : 5 items in queue
(0 ) Getting 05.scm : 4 items in queue
(0 ) Getting 01.scm : 3 items in queue
(0 ) Getting 02.scm : 2 items in queue
(0 ) Getting 03.scm : 1 items in queue
No. hilos: 1 - Tiempo: 33.46556 seg.

(producer ) === NO ITEMS LEFT TO INSERT ===
(4 ) Getting 01.html : 8 items in queue
(4 ) Getting 04.scm : 7 items in queue
(1 ) Getting 10.scm : 6 items in queue
(3 ) Getting 11.scm : 5 items in queue
(0 ) Getting 05.scm : 4 items in queue
(2 ) Getting 01.scm : 3 items in queue
(4 ) Getting 02.scm : 2 items in queue
(3 ) Getting 03.scm : 1 items in queue
No. hilos: 5 - Tiempo: 2.82454 seg.
```

## Liga del video explicativo

<https://youtu.be/jHuuaGvjp1U>

### Instrucciones para su ejecución:

-Pasos para ejecutar Resaltador léxico con hilos en windows-

Instalar Python versión 3 en adelante

Asegurarse que se instaló correctamente usando el comando `python --version`

Asegurarse de tener pip con el comando: `pip -h`

Ejecutar el comando: `pip install virtualenv`

Descargar archivos de repo: `lexico.py`, `main.py`, `requirements.txt` y `tabla.csv`. Crear una carpeta y almacenarlos en dicha carpeta.

Cambiar el directorio de la consola al de la carpeta donde se encuentran los archivos descargados

Ejecutar el comando: `virtualenv venv`

Ejecutar el comando: `pip install -r requirements.txt`

Si se desea -> Modificar valores de variables dentro del archivo `main.py`:

Ejecutar el comando: `python main.py`

Verificar resultados en carpeta `test`

-Pasos para ejecutar Resaltador léxico con hilos en loS-

Abrir la terminal y ejecutar el siguiente link "`curl`

`https://raw.githubusercontent.com/pablo-blancoc/TC2037/master/Actividades/5.3/install.sh --output install.sh && bash install.sh`"

Dentro de la terminal desplegará las siguientes instrucciones a seguir