* El algoritmo fue hecho en Python y funciona procesando un archivo que almacene las urls en paralelo, para ello se usa el módulo **multiprocessing**, esto nos ayuda a trabajar con grandes cantidades de datos. El archivo es leído línea por línea y estas se almacenan en una lista, la cual se divide en partes iguales, para esto se usó workers que y luego se crea sub-listas que serán procesadas de forma independiente en paralelo. Cada sub-lista es procesada por la función procesar\_parte\_urls por cada worker. Luego la función revisa cada url de la sub-lista y verifica los requerimientos si contiene la palabra **shop** y termina en **.html** , en caso cumpla y no sea repetida, se añade a un conjunto set de urls únicas y el contador aumenta. Luego que todos los Works terminan de procesar la información, se combinan. Luego se une para eliminar duplicados y el contador total de urls válidas se calcula sumando los valores individuales. Finalmente se imprime una lista de urls válidas y su total.
* Las técnicas utilizadas en el algoritmo fueron:
* Procesamiento en paralelo usando multiprocessing para poder procesar el archivo en paralelo, dividiendo el trabajo en múltiples Workers.
* División del archivo en chunks, o mejor dicho en partes iguales basado en el número de workers, así se podrá procesar grandes cantidades de información sin sobrecargar un worker.
* El uso de set para eliminar los elementos duplicados y almacenar urls únicas de manera más eficiente, esto asegura que no haya urls repetidas sin tener que recorrer listas manualmente.
* Procesamiento de memoria, aunque sea un archivo de gran cantidad de datos, el algoritmo no intenta cargarlo completamente en memoria, en su lugar usa chunks pequeños para procesar, así se asegura un bajo uso de memoria.
* Reducción de complejidad, el algoritmo utiliza las operaciones **in**  y **.endwith()** para verificar si la url cumple con los requerimientos, si contiene la palabra **shop** y si termina en **.html**; estas operaciones comparadas a las convencionales son muy rápidas y ligeras, permite reducir el tiempo de procesamiento.

Para ejecutar el algoritmo debe tener Python instalado, contar con un editor de texto y el archivo con las urls en la misma carpeta, en mi caso tengo mi archivo urls.txt:



Luego abrir el algoritmo en el editor de texto, en mi caso usé el IDLE de Python y en la parte superior está la opción Run, al dar click seleccionar la opción Run Module:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El algoritmo devolverá una pantalla con el resultado:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente