**News Web Scraper**

Para la realización de este programa se empleó lo siguiente:

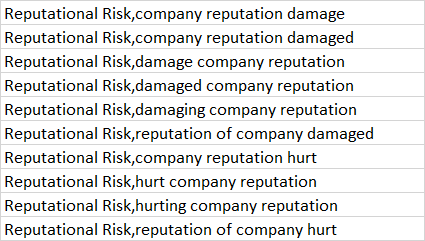
* **Lenguajes:**
  + Python 3 (3.7.3)
* **IDEs/Editores de texto:**
  + Visual Studio Code:
    - Python Extension Pack
* **Librerías/Paquetes/Dependencias/Tecnologías:**
  + Pipenv
    - pip: instala paquetes en el proyecto
    - virtualenv: genera entornos virtuales
  + pylint: ayuda en la correcta escritura de código
  + flake8: apoya en la correcta escritura de código
  + pep8: ayuda en la correcta escritura de código
  + autopep8: aplica formato al código (indentación, sangrado)
  + requests: permite enviar solicitudes HTTP
  + beautifulsoup4: extrae datos de documentos HTML o XML
  + xlsxwriter: permite crear y escribir archivos .xlsx (Excel)
  + parmap: provee una sintaxis para facilitar el uso de map (multiprocessing)
    - tqdm: muestra una barra de progreso

Para este programa se utilizó Pipenv (https://docs.pipenv.org/en/latest/) para preparar y organizar los entornos virtuales y la instalación de paquetes. Utiliza pip y virtualenv para generar un entorno virtual e instalar los paquetes que se requieran. También genera dos archivos los cuales se leen al momento de la creación del entorno para la instalación de paquetes, es un comportamiento de empaquetamiento similar al de npm, yarn, etc.

NOTA: Se requeriría instalar las dependencias si en caso no se encontraran.

Dado que los datos ingresados deben provenir de un archivo XML, se desarrolló un pequeño script para la conversión del archivo CSV. Este es llamado “csvtoxml”.

De INPUT.csv a INPUT.xml.



****

Luego se realiza la ejecución del programa principal de News Web Scraper: “webscraper”.

El proceso consiste en realizar las solicitudes HTTP hacia el RSS *Feed* de los motores de búsqueda, con base en los datos proveídos del archivo XML mencionado anteriormente.

Seguidamente, extrae los elementos que interesan, que serían los enlaces o URLs como resultado de la búsqueda de los *keywords*.

Finalmente, crea o reescribe un archivo XLSX con los enlaces obtenidos, las palabras clave y los tópicos respectivos.

Estos procesos se realizan independientemente en paralelo para cada motor de búsqueda.

Para la aplicación se ha considerado un límite de 800 solicitudes para evitar posibles vetos del dominio. Divididos entre cada uno, es decir 160 solicitudes por motor de búsqueda para fines de prueba.

También son divididos por *keywords*, en otras palabras, las cien primeras serán para el primer motor, las siguientes cien para el segundo, y así sucesivamente.

Para la elección de los motores de búsqueda de noticias, se determinó que el formato Really Simple Syndication (RSS) sería el más apropiado para enviarle las solicitudes, puesto que se encuentra en formato XML, y usando éste sería más eficiente que descargar un archivo HTML.

Los motores de búsqueda utilizados fueron:

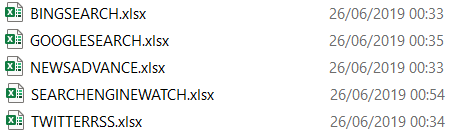
* **Google**
  + Formato: https://news.google.com/rss/search?q=company+reputation+damage&hl=en-US&gl=US&ceid=US:en
* **Bing**
  + Formato: https://www.bing.com/news/search?q=company+reputation+damage&format=rss&mkt=en-US
* **SearchEngineWatch**
  + Formato: https://searchenginewatch.com/search/company+reputation+damage/feed/rss
* **The News & Advance**
  + Formato: https://www.newsadvance.com/search/?q=company+reputation+damage&f=rss
* **Twitter RSS Feeds (No oficial)**
  + Formato: http://twitrss.me/twitter\_search\_to\_rss/?term=company+reputation+damage

Los resultados fueron prometedores, cada motor obtuvo información mayoritariamente con las palabras clave. Con excepción de Google y Bing, que sí proveyeron en su totalidad.

Cabe mencionar que SearchEngineWatch tiene cierta demora al realizar las solicitudes a su servidor, por lo que todos los procesos hacia los otros motores terminan en aproximadamente 3 minutos y 42 segundos, éste se prolonga hasta más de 19 minutos.

Estos procesos se realizan en paralelo, pero el programa principal espera a que todos terminen, por lo que este retraso de SearchEngineWatch tiene mucho impacto. Se podría omitir a este motor, pero se mantuvo para demostrar las funcionalidades del Web Scraper.

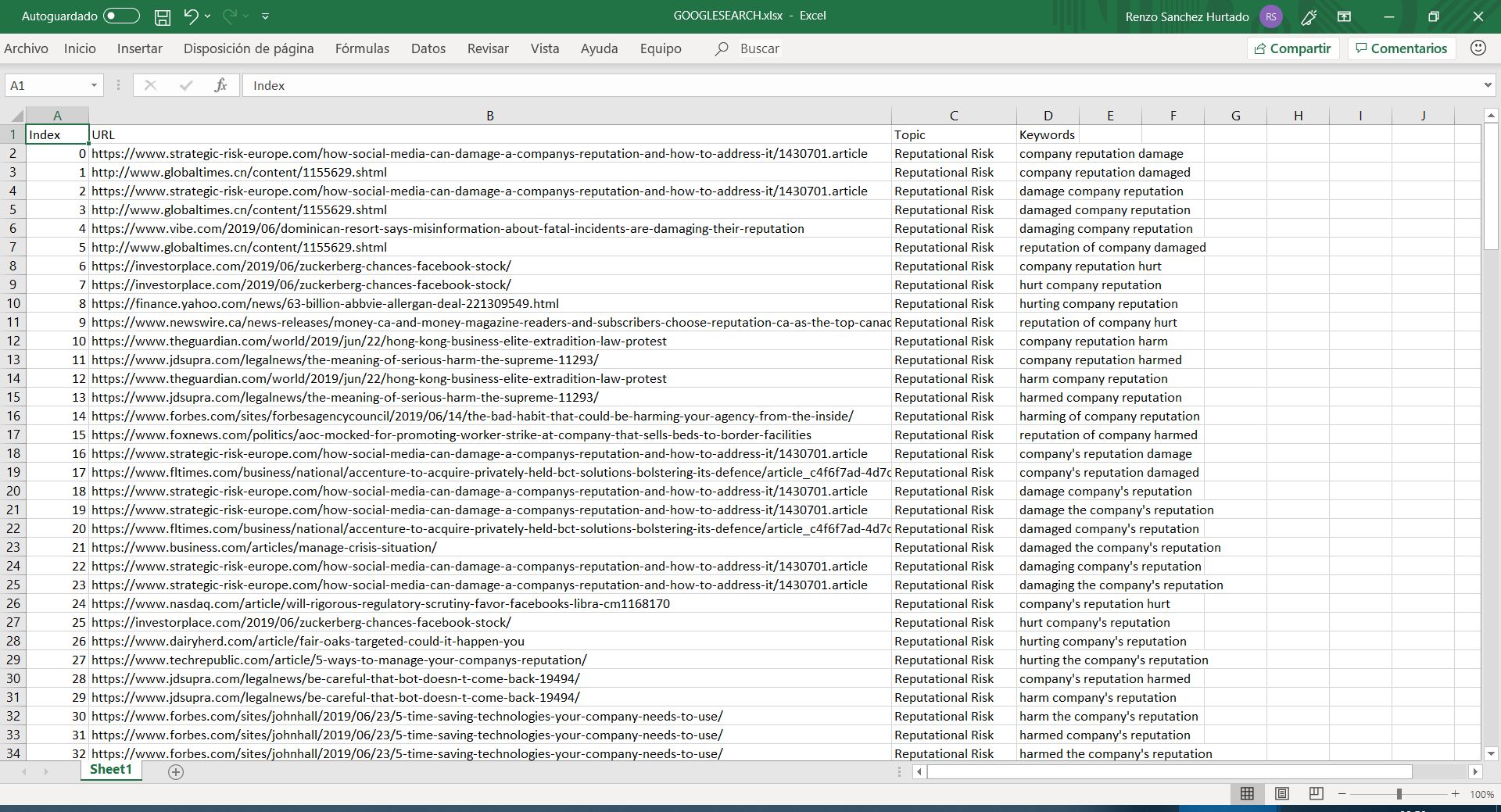




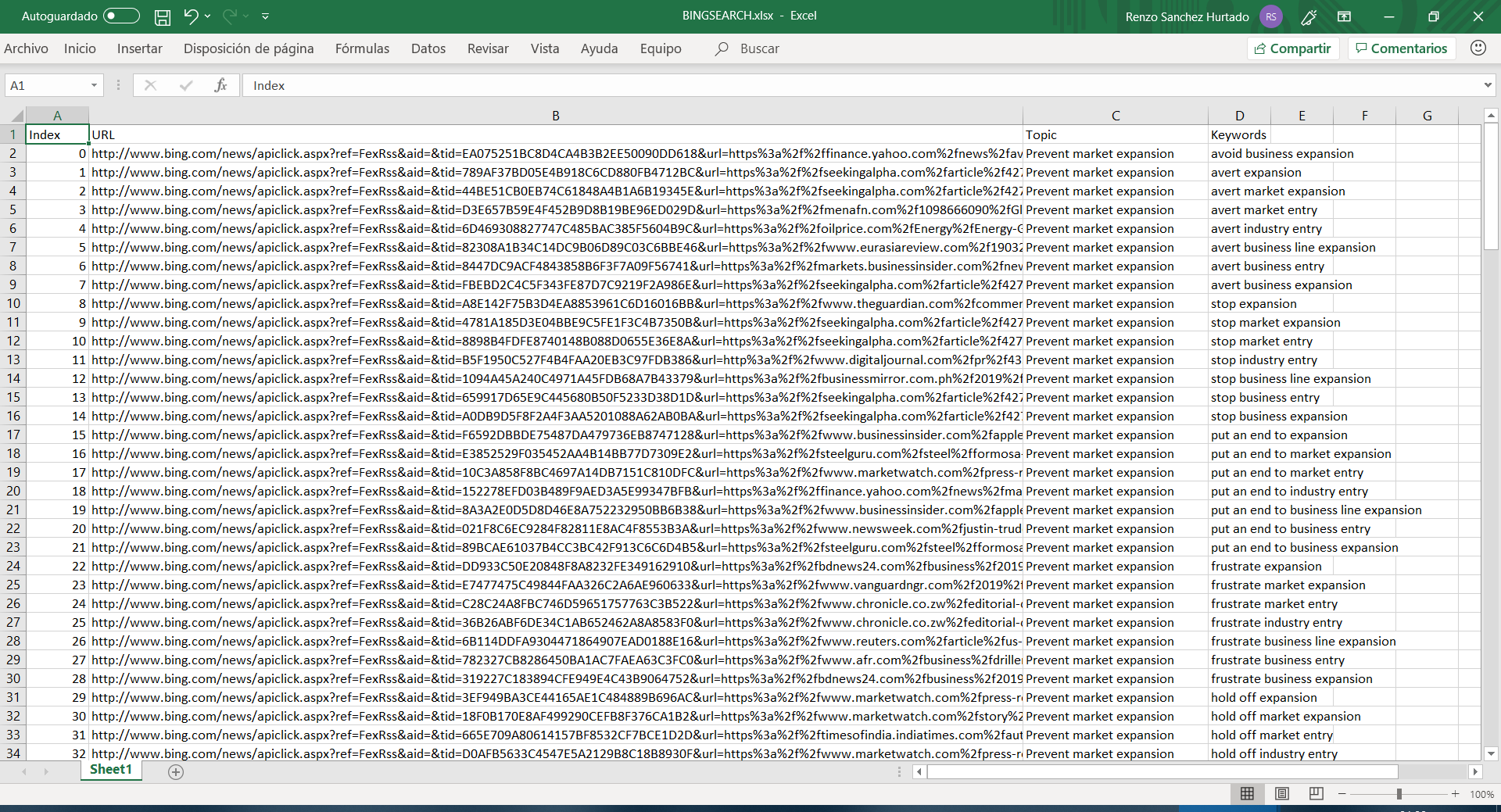
Tomando en cuenta que, al ejecutarse el programa, se ejecuta cada proceso en este orden: Google, Bing, SearchEngineWatch, NewsAdvance y TwitterRSS. Y terminan independientemente.  
Se puede concluir que el programa tuvo un rendimiento satisfactorio, siendo SearchEngineWatch la única excepción.

Los resultados para cada caso fueron éstos, según motor de búsqueda:

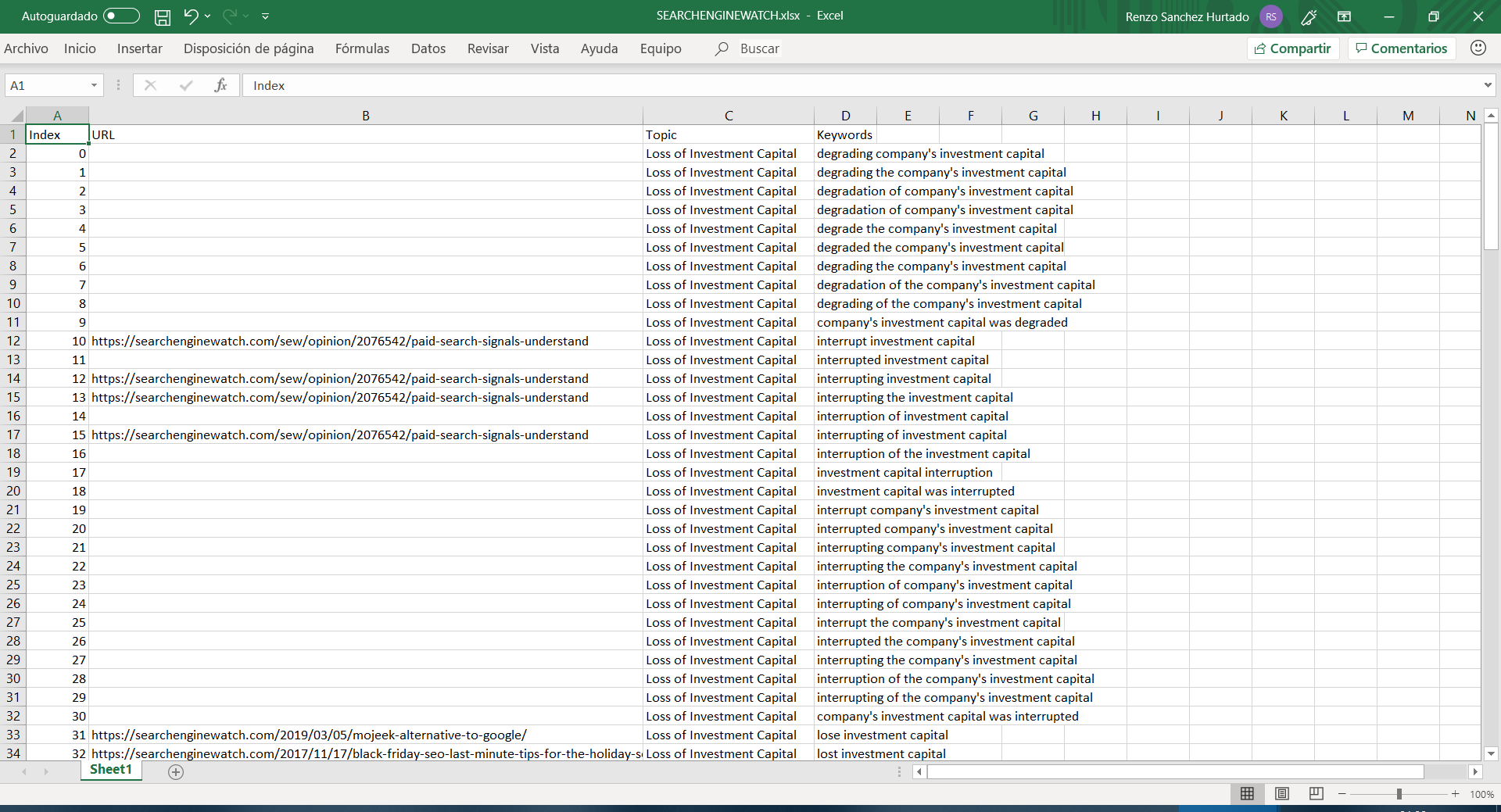
* Google



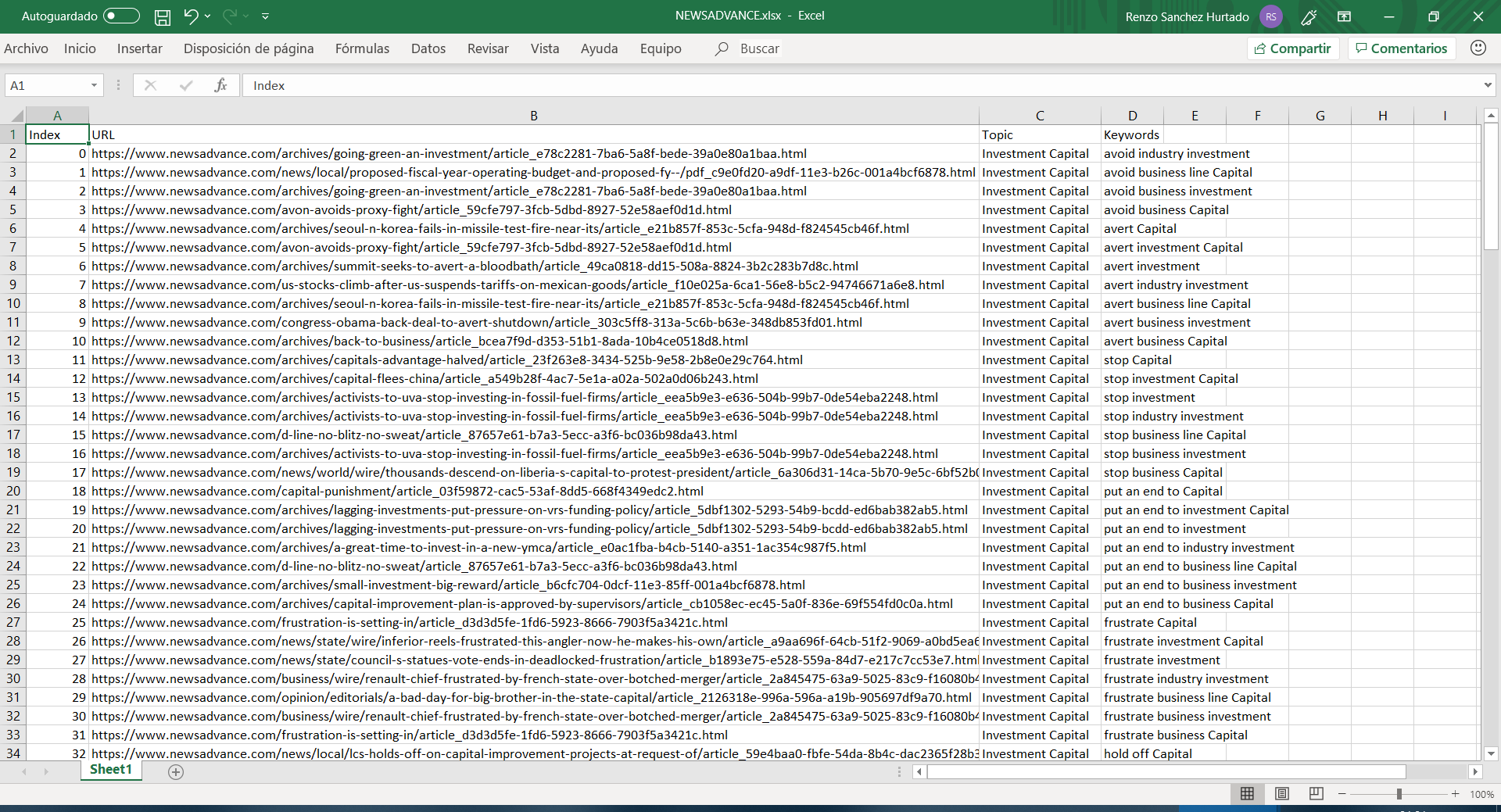
* Bing



* SearchEngineWatch



* The News & Advance



* Twitter RSS Feed

