Taller: Recuperación de Información con Scrapy

Este taller consiste en la aplicación del framework *Scrapy* para la extracción o recuperación de datos desestructurados desde la web. El taller consta de 3 partes. Donde cada parte debe implementarse como un proyecto Scrapy independiente.

Sobre la entrega

La entrega se debe realizar mediante Aula Virtual, en la fecha y hora que serán anunciadas en clase. Se debe realizar sólo 1 entrega por grupo.

Evaluación

Cada parte tendrá un puntaje máximo de 100 puntos. La nota final del taller corresponde al promedio de las 3 partes. La conversión del puntaje a la nota final se realizará con una escala de exigencia al 60%.

Parte 1: Quotes to Scrape

Esta parte consiste en extender el proceso de scraping al sitio http://quotes.toscrape.com. Para ello, debe realizar lo siguiente:

- (20 ptos) Defina un modelo relacional para representar la información de `Autor`, `Cita`, y `Etiqueta`. Un `Autor` puede tener 1 o más `Cita`; hay una relación muchos-a-muchos entre `Cita` y `Etiqueta`. Implemente este modelo en una base de datos `SQLite`.
- (25 ptos) Implemente el pipeline `MaxLengthPipeline` que normaliza el texto de las citas, dejándolo con un máximo de 255 caracteres. Este pipeline va primero en la cadena de procesamiento.
- (25 ptos) Implemente el pipeline `SQLiteCitasPipeline` que procesa un ítem insertándolo en la base de datos del punto anterior. Este pipeline va en segundo lugar en la cadena de procesamiento.
- (30 ptos) Implemente un nuevo formato de exportación, mediante la extensión de la clase `BaseItemExporter`, para que los items de salida sean escritos como consultas SQLite para inserción en la base de datos.

Parte 2:

En esta parte usted debe utilizar Scrapy para confeccionar un catálogo de todas las URLs del sitio web de la Escuela, comenzando por la página principal http://www.inf.ucv.cl. El objetivo es generar un set de datos de salida con las siguientes columnas:

- (15 ptos) URL
- (15 ptos) Status: indica el código de status retornado al tratar de acceder la URL. Para poder ver sitios no alcanzables, considere la configuración de este enlace
- (20 ptos) Content-Type: indica el tipo de contenido de la URL según lo retornado por el servidor. Usar "n/a" en caso de no existir la información.
- (20 ptos) Content-Length: el tamaño en bytes del recursom según lo retornado por el servidor. Usar 0 en caso de no existir la información.

- (30 ptos) Out-Links: el conjunto de URLs de salida (no repetidas) que salen desde este documento, si su contenido es de tipo HTML.

Para ello debe realizar lo siguiente:

- Defina una spider `InfSpider`, y confígurela para que comience desde la URL indicada, y que sólo visite sitios en el dominio `inf.ucv.cl`.
- Defina el método `parse`, en el cual se obtengan todos los links del documento obtenido. Para cada elemento, debe primero averiguar si su tipo es HTML o no. Para ello configure su objeto `Request` de forma que use el método `HEAD`. Llene el dataframe según corresponda, y siga solamente los enlaces que corresponden a documentos HTML.
- Determine si la implementación requiere el uso de pipelines, middleware, u otros elementos de Scrapy.

Parte 3:

Seleccione un sitio web dinámico y realice el scraping utilizando `scrapy-splash`. Documente en un archivo de texto el proceso, las dificultades, y la información a la que se le ha realizado el scraping. Considere los siguientes elementos de evaluación:

- (70 ptos) Su scraper debe procesar al menos 2 tipos de páginas distintas, que se generen dinámicamente, por ejemplo con Ajax o Angular.
- (15 ptos) Su scraper debe recolectar a lo menos 100 items. Además debe implementar a lo menos 3 items de pipeline para su procesamiento.
- (15 ptos) Al igual que en la parte 1, parte del procesamiento consiste en definir un modelo relacional y almacenar la información en él. Use nuevamente SQLite.