Sintaxis y semántica de los lenguajes
Trabajo Práctico Nro 2

Web Scraping Financiero
(30 de Junio 2022)

Comisión: Ing. Pablo Damián Mendez.
Fecha de entrega: 18/08/22

1. Introducción

El Web Scraping es una técnica que consiste en la utilización de programas, comúnmente denominados robots o simplemente “bots” (en spanglish también se les dice “scrapeadores”), para extraer información de páginas web automáticamente.

Algunos de los usos más comunes o prácticos para los que se utiliza el web scraping:

1. Analizar tendencias, por ejemplo, temas más posteados.
2. Extraer datos de contactos como por ejemplo cuentas de mail.
3. Seguir la evolución de precios de distintos productos.
4. Extraer los títulos y contenidos de un blog.
5. Realizar campañas de marketing, analizando webs propicias para publicitar productos.

1.1. Características del Web Scraping

- Se utiliza a través del protocolo HTTP: Naturalmente, como lo que se intenta analizar es un archivo HTML (El formato de las páginas web), el stream o flujo de bytes del archivo a analizar se obtiene a través de un request http. Básicamente, un request http es lo que se hace al ingresar una dirección web en el navegador y solicitar la página de dicha dirección.
- Analizan archivos que comúnmente tienen formato HTML (entre otras cosas como código Javascript, Jquery, imágenes, etc): Asociado con el punto anterior, el formato de una página web está dado por el lenguaje HTML (no es un lenguaje de programación imperativa, sino un formato de archivo de texto, una especificación de archivos XML). Cuando se conoce el formato de la página a analizar, como sucede en este trabajo, se pueden utilizar herramientas como expresiones regulares o búsqueda de texto para encontrar o acercarse rápidamente a la parte que contiene la información deseada.
- Existen muchas aplicaciones para realizar *web scraping*. Estas aplicaciones podrían reconocer automáticamente la estructura de cierta página o brindar una interfaz al usuario donde este pudiera seleccionar los campos que son de interés dentro del documento. Por supuesto, **está prohibido utilizar herramientas de este tipo para este trabajo práctico.**
- Legalidad y legitimidad: Su utilización no está normada. De todas maneras, existen cuestiones que deben ser tenidas en cuenta. Por ejemplo: si se realizan demasiados requests a un mismo servidor dejándolo temporalmente inhabilitado (DoS attack).

1.2. El lenguaje HTML

Para poder aplicar web scraping, obviamente es fundamental entender la materia prima a tratar, esto es: archivos en HTML.

El lenguaje HTML es el utilizado para la confección de páginas web. Sus siglas representan las iniciales de *HyperText Markup Language* (lenguaje de marcas de hipertexto). HTML es una instancia, una especificación de un lenguaje, utilizando XML.
Si bien, hoy en día, el HTML se utiliza junto a otros lenguajes (como javascript, JQUERY desde el cliente y algún otro lenguaje que pueda ejecutarse desde el servidor, como PHP, ASP.NET, JSP, etc), la estructura de las páginas está especificada puramente en HTML.
Como todo lenguaje formal cumple un estándar normado por *World Wide Web Consortium* (W3C) o Consorcio WWW (https://www.w3.org/), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Esta estandarización, permite que los navegadores web deban cumplir el estándar, permitiendo la existencia distintos productos para interpretar HTML.
En definitiva, un HTML es un archivo de texto (se puede crear con el notepad), cuya característica principal es el uso de etiquetas para crear los elementos que componen la página. Dichos elementos no pueden estar en cualquier posición, por ejemplo, el elemento raíz debe ser una etiqueta <html> y todo el contenido de la página debe cerrarse cerrando dicha etiqueta con la etiqueta </html> (todos los elementos deben ser cerrados, aunque algunos pueden ser cerrados dentro de la misma etiqueta que se abren).

1.3. Etiquetas html básicas

- <html>: define el inicio del documento HTML, le indica al navegador que lo que viene a continuación debe ser interpretado como código HTML. Esto es así *de facto*, ya que en teoría lo que define el tipo de documento es el DOCTYPE, que significa la palabra justo tras DOCTYPE el tag de raíz.
- <script>: incrusta un script en una web, o llama a uno mediante src="url del script". Se recomienda incluir el tipo MIME en el atributo type, en el caso de JavaScript text/javascript.
- <head>: define la cabecera del documento HTML; esta cabecera suele contener información sobre el documento que no se muestra directamente al usuario como, por ejemplo, el título de la ventana del navegador. Dentro de la cabecera <head> es posible encontrar:
- <title>: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana.
- <body>: define el contenido principal o cuerpo del documento. Esta es la parte del documento html que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo <body> es posible encontrar numerosas etiquetas. A continuación se indican algunas a modo de ejemplo:
- <h1> a <h6>: encabezados o títulos del documento con diferente relevancia.
- <table>: define una tabla.
    - <tr>: fila de una tabla.
    - <td>: celda de una tabla (debe estar dentro de una fila).

  Estas etiquetas se utilizan de forma anidada. Por ejemplo, una tabla de una sola fila que contiene una sola celda sería:

  <table><tr><td>Contenido de una celda</td></tr></table>

- <a>: hipervínculo o enlace, dentro o fuera del sitio web. Debe definirse el parámetro de pasada por medio del atributo *href*. Por ejemplo: Ejemplo se representa como ejemplo.

- <img>: imagen. Requiere del atributo *src*, que indica la ruta en la que se encuentra la imagen. Por ejemplo: . Es conveniente, por accesibilidad, poner un atributo alt="texto alternativo".
- : etiquetas para listas.
- <b>: texto en negrita (*Deprecado. Se reemplazó por la etiqueta* <*strong*>).
- <i>: texto en cursiva (*Deprecado. Se reemplazó por la etiqueta* <*em*>).
- <s>: texto tachado (*Deprecado. Se reemplazó por la etiqueta* <*del*>).
- <u>: Subrayado.

Nuevamente, la mayoría de etiquetas deben cerrarse como se abren, pero con una barra (/). Note, por ejemplo, la apertura y cierre del elemento
 del ejemplo de la figura 1.

1.4. Ejemplo HTML básico

El siguiente código muestra un ejemplo de un HTML básico que incluye en su cuerpo algunos elementos básicos listados en el punto anterior: un encabezado de tamaño 1, un párrafo, una tabla con dos filas y dos columnas y, finalmente, un vínculo a la página de la facultad. Por simplicidad, y para obviar por ahora los entities html, no se utilizan tildes dentro del código. Ud. puede crear este código, o uno similar, utilizado notepad u otro editor de texto (no procesador de texto como word o write, sino editor). Guarde el archivo con extensión .HTML para que el sistema operativo lo asocie directamente para ser abierto con el navegador predeterminado.

```
<html>
  <head>
    <title>Sintaxis y semantica de los lenguajes: Trabajo practico</title>
  </head>
  <body>
        <h1>Trabajo practico #1</h1>
    <p>Vamos a aplicar el conocimiento adquirido en nuestra materia a un lenguaje utilizado actualmente.</p>
        <table>
                <tr>
                <td>
                        Columna 1
                </td>
                <td>
                        Columna 2
                </td>
                </tr>
                <tr>
                <td>
                        1
                </td>
                <td>
                        2
                </td>
                </tr>
        </table>
        <br/> <!-- El br es un ejemplo de etiqueta que se abre y se cierra inmediatamente -->
        <a href="https://www.frba.utn.edu.ar/en/">Ir a pagina de la facu.</a>
  </body>
</html>
```

Figura 1 - Código HTML de la página de ejemplo

Al abrir el archivo descripto en la figura anterior se visualiza en el navegador web lo siguiente:

Trabajo practico #1

Vamos a aplicar el conocimiento adquirido en nuestra materia a un lenguaje utilizado actualmente.

Columna 1	Columna 2
1	2

Ir a pagina de la facu.

Figura 2 - Visualización del código de la figura 1

1.5. Lectura directa de un HTML desde su URL – WGET

Hay varias maneras de leer directamente una página de internet desde una URL. Una forma, quizás la más genérica, sería mediante la utilización de sockets, pero es tema de otras materias.
Para no tener que utilizar sockets en C, otra posibilidad es el uso de WGET para poder obtener el stream del archivo HTML correspondiente.
De esta manera se puede abrir el stream directamente en una variable FILE* y leer el archivo HTML de cualquiera de las maneras vistas en clase para leer archivos de texto. El siguiente ejemplo muestra la apertura y lectura de la página que se debe leer para este trabajo:

```
1   #include <stdio.h>
2
3   FILE *popen(const char *command, const char *mode);
4   int pclose(FILE *stream);
5
6   int main(void)
7   {
8       /* wget -q = silent mode */
9       FILE *cmd = popen("C:\\GnuWin32\\bin\\wget -q -O - https://www.bolsar.com/VistasDL/PaginaLideres.aspx --no-check-certificate", "r");
10      char result[1048576]; //1 MB de buffer, ajustar segun implementacion
11
12      while (fgets(result, sizeof(result), cmd) != NULL)
13          printf("%s", result);
14      pclose(cmd);
15      return 0;
16  }
17
18
19
```

Figura 3: Obtención del HTML de una página a través de internet utilizando wget

Para que se detecte el comando wget en windows tiene dos posibilidades: agregar wget en la variable de entorno PATH o incluir el ejecutable en la carpeta del proyecto.
Se puede ver en la figura 3 que la lectura es idéntica a un archivo de texto, en este caso con fgets, sin embargo, podría utilizarse cualquier función de lectura, como scanf, fgetc, etc.
En ese trabajo, leeremos un archivo localmente, para evitar inconvenientes de lectura remota, la mayoría de los servidores actuales bloquean por defecto las lecturas por wget, y se deben utilizar herramientas más complejas.

2. Objetivo del trabajo

Realizar el web scraping sobre un archivo de texto que simula la página de acciones líderes de bolsar.info (https://bolsar.info/lideres.php). Se deben emitir los siguientes reportes:

A. Listar en pantalla las acciones cuyo precio al contado al momento de apertura supera los $200.

B. Listado del promedio de cotización de compra y venta de todas las acciones al contado (“Cdo.” en la columna Vto) en un archivo .CSV (Comma Separated Values) importable desde excel. La hoja debe incluir encabezados y las columnas deben ser:

   Especie; Precio de compra; Precio de venta; Apertura; Promedio;.

   Un archivo CSV no es más que un archivo de texto con valores separados por caracteres ‘,’ o ‘;’. También se suelen utilizar tabulaciones como separadores. Debe tenerse en cuenta que debe establecer uno sólo de ellos como separador.

C. Mismo reporte que el listado A pero en una tabla html, indicando en color verde las filas de las especies cuyo precio de compra y precio de venta es menor al precio de apertura.

D. Mostrar por pantalla las acciones que tienen menor y mayor % de variación.

El programa debe contener un menú que permita elegir al usuario el reporte deseado.

3. Entregables

- Informe digital con carátula; explicación de la estrategia de resolución, sin omitir todas las herramientas o unidades que se hayan utilizado que no sean estándar del lenguaje C.
- Manual del usuario. No dar supuestos y aclarar todos los puntos para que el programa pueda ser ejecutado (desde el SO para el cual fue compilado).
- Casos de prueba: Captura de las salidas de los distintos puntos del enunciado. Pueden incluir archivos estáticos HTML con modificaciones sobre los valores del HTML de ejemplo. Código fuente: Se debe entregar el código fuente además del binario en repositorio git.