

Sintaxis y Semántica de los Lenguajes

Curso K2054

Manual de usuario Web Scraping

Alumno: Cesar Mejia

Legajo: 1418713

Correo: cmejia@frba.utn.edu.ar

Usuario Github: comejia

Repositorio: <https://github.com/comejia/utn-ssl-examples>

Docente: Ing. Pablo D. Mendez

Fecha de entrega:		Nota:		Fecha Aprobación:	
-------------------	--	-------	--	-------------------	--

Comentarios:

Tabla de contenido

Consideraciones	1
Estructura del proyecto	2
Interfaz de usuario	3
Configuración	3
Ejecución	4

Consideraciones

El presente manual se realizó con la siguiente configuración de trabajo:

- Sistema operativo: Linux
- Entorno de desarrollo (IDE): Visual Studio Code
- Terminal / consola bash
- wget

Si se utiliza otro sistema operativo que no sea Linux, es muy probable que algunos pasos no se puedan realizar. Por lo que deberá buscar cómo hacerlos en el sistema elegido.

El uso del IDE es opcional y no afecta en nada si lo utiliza o no (para programar en Lenguaje C con un editor de texto es suficiente). Aunque es importante considerar su uso ya que agiliza el desarrollo al incorporar plugins como autocompletado, resaltado de sintaxis, previsualizador de archivos .csv o .html, entre otras cosas que se pueden hacer con VSCode.

Este programa se puede descargar desde la página <https://code.visualstudio.com/> seleccionando el S.O. correspondiente.

La consola se utiliza para compilar y ejecutar el programa como se indica más adelante.

Wget es un programa que viene en la mayoría de los sistemas operativos Linux, y se utiliza dentro del programa para obtener la página en tiempo real.

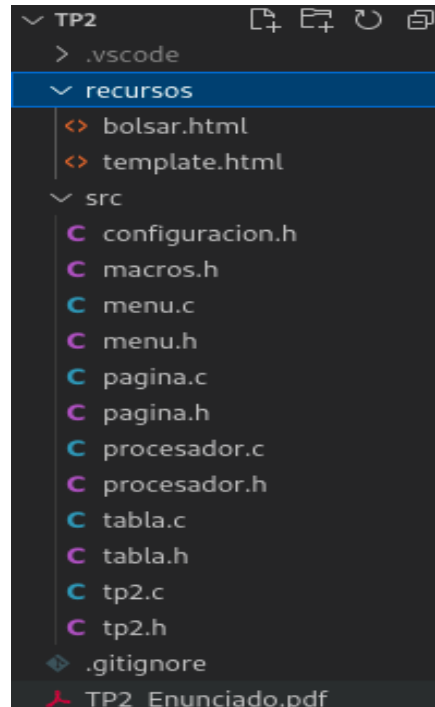
En caso de no tenerlo instalado wget puede hacerlo con:

```
sudo apt-get install wget
```

Este comando puede variar según la distribución que tenga, y no necesita ser configurado como en Windows.

Estructura del proyecto

Dentro de la carpeta TP2 se encuentran los siguientes archivos:



En “**recursos**” hay archivos que se utilizan para la ejecución del código:

- bolsar.html: contiene el código fuente de la página <https://bolsar.info/lideres.php>. Muy útil cuando se quiere trabajar de manera **offline** y hacer pruebas rápidas.
- template.html: tiene la estructura básica de una tabla en código HTML que luego se va a rellenar en tiempo de ejecución al seleccionar la opción 3. Si bien la plantilla es básica se puede mejorar el estilo. Una referencia muy útil es <https://www.w3schools.com/>

Carpeta “**src**”, está todo el código fuente desarrollado en C, dividido en distintos módulos para su mejor mantenimiento y comprensión. Esto facilita la mejora o ampliación del código, ya que con solo modificar el archivo correspondiente no debería afectar al resto

El resto de archivos corresponde a los informes del trabajo.

Interfaz de usuario

Al ejecutar el programa se cuenta con el siguiente menú:

```
Seleccione una opcion para generar el reporte deseado:
[1] - Listar en pantalla las especies cuyo porcentaje de variación es negativo
[2] - Generar reporte de las cotizaciones de compra y de venta en un archivo .CSV
[3] - Generar reporte de la opcion [1] en un archivo HTML
[4] - Imprimir codigo fuente de pagina (debug)
[0] - Salir de la aplicacion

Ingresar opcion: █
```

que es la forma de interactuar con el sistema y que al seleccionar una opción, la salida se va a mostrar en consola o por medio de un archivo, según corresponda.

Configuración

```
#define VENCIMIENTO_24HS    ">24hs<"
#define VENCIMIENTO_48HS    ">48hs<"

#define ACCION_VENCIMIENTO  VENCIMIENTO_48HS

#define RUTA_SALIDA_CSV      "compra_venta.csv"
#define RUTA_SALIDA_HTML     "index.html"
```

Previo a la ejecución del programa, se pueden modificar ciertos parámetros que se encuentran en el archivo configuración.h. A continuación se explica cada uno de ellos:

- ACCION_VENCIMIENTO: se puede elegir entre VENCIMIENTO_24HS o VENCIMIENTO_48HS (default).
- RUTA_SALIDA_CSV: la ruta puede ser absoluta o relativa. Si es relativa, como se muestra en la imagen, entonces el archivo de salida se va a crear en la misma carpeta. En una ruta absoluta se va a guardar donde se indique, ejemplo “/home/user/Documents/nombre.csv”
- RUTA_SALIDA_HTML: idem punto anterior. Nombre por defecto “index.html”

Ejecución

Para probar la aplicación se puede hacer de dos formas:

- Compilacion y ejecucion a “mano”
- Ejecución por medio de un script

Compilacion y ejecucion a “mano”

Para poder ejecutar el programa y ante la modificación del código, lo primero que se debe hacer es compilar el proyecto.

Para ello, hay que moverse a la carpeta *src* desde la consola mediante:

```
cd /ruta/al/proyecto/TP2/src
```

cd: comando linux para cambiar de directorio

Luego para compilar:

```
gcc *.c -o nombre_binario -Wall
```

gcc: es el compilador para C y C++

*.c: hace referencia a todos los archivos .c

-o: para especificar un nombre al ejecutable

-Wall: para que muestre los warning en el código, si es que hay.

Finalmente, para ejecutar se hace con:

```
./nombre_binario tipo_ejecucion
```

tipo_ejecucion: es el parámetro que se le pasa al programa, puede ser **--online** si se quiere ver datos en tiempo real o **--offline** para pruebas.

Ejecución por medio de un script

Mediante el script “**run.sh**” se logra lo mismo que el punto anterior, ya que contiene los mismos comandos. Pero para correr el programa solo es necesario ejecutar:

```
cd /ruta/al/proyecto/TP2/src
```

```
./run.sh tipo_ejecucion
```

Una vez ejecutado el programa se muestra la interfaz de usuario para poder interactuar con el sistema y generar los reportes que desee.

Para ver las distintas salidas que se obtienen, puede ver el informe TP2_Cesar_Mejia.pdf