

INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

APLICACIONES PARA SERVICIOS EN RED

PACTICA No. 5

PROFESOR:

MORENO CERVANTES AXEL ERNESTO

ALUMNOS:

GURROLA SANCHEZ JOEL

TENORIO ALVA ALAN OSMANI

GRUPO: 3CM16

Introducción

Wget es una herramienta informática creada por el Proyecto GNU. Puedes usarlo para recuperar contenido y archivos de varios servidores web. El nombre es una combinación de World Wide Web y la palabra get. Admite descargas a través de FTP, SFTP, HTTP y HTTPS.

Wget se crea en C portátil y se puede usar en cualquier sistema Unix. También es posible implementarlo en Mac OS X, Microsoft Windows, AmigaOS y otras plataformas populares.

Así que haciendo la copia de esta herramienta creamos un programa en Python que hace la petición a una url que le pasemos como parámetro e iniciara la descarga del sitio según una profundidad que le asignemos.

Peticiones HTTP

HTTP contiene un grupo de peticiones HTTP (también llamadas HTTP verbs por, el tipo de nombre que manejan casi todos ellos -pues si bien algunos son sustantivos, la gran mayoría no-) que nos ayudan a especificar la acción que se requiere realizar en un elemento determinado y aunque estas peticiones tienen distintas semánticas, también tienen muchas similitudes en las mismas que evitan que este grupo se extienda demasiado.

Clasificación de las peticiones HTTP

Estas peticiones las podemos clasificar en Safe e Idempotent:

Peticiones HTTP Safe: Un método HTTP es considerado safe o seguro si no altera el estado del servidor. En otras palabras, un método es seguro si conduce a una operación de 'sólo lectura'. Algunos de los métodos HTTP más comunes son seguros: OPTIONS, GET o HEAD. Todos los métodos seguros son también a su vez idempotent (así como también algunos, pero no todos, los métodos inseguros como DELETE o PUT).


Peticiones HTTP Idempotent: Así como un objeto cualquiera tiene la propiedad de idempotencia si al realizar una operación muchas veces da el mismo resultado cual si se hubiese realizado la operación una sola vez, un método HTTP es idempotente si una solicitud idéntica puede realizarse una

o demasiadas veces consecutivamente obteniendo el mismo resultado dejando al servidor en el mismo estado.

En la vida y en los números reales, podríamos llamar al 1 y al 0 como los únicos idempotentes para la operación de multiplicación, pues estos números al multiplicarse en muchas ocasiones, da como resultado él mismo ($1^n = 1$). Volviendo a HTTP, los métodos que (implementados correctamente) son idempotentes son el GET, HEAD, PUT y DELETE y como lo explicamos en los métodos Safe, todos éstos últimos son idempotentes.

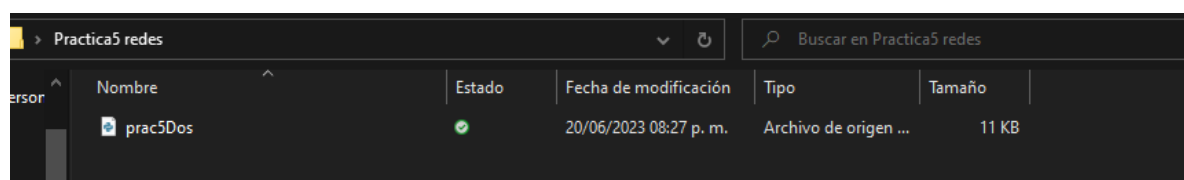
Desarrollo

Se decidió hacer esta práctica en Python para usar BeautifulSoup, es una librería que facilita mucho el Web Scrapping, esto nos sirve para obtener links de una página con una sola línea de código, sin embargo, hay otros retos como los posibles ciclos entre links encontrados.

A screenshot of a code editor with a dark background. The code is a Python script that uses the ThreadPoolExecutor from the concurrent.futures module to perform a web request using wpyget. The script is as follows:

```
1 if __name__ == "__main__":
2     try:
3         executor = ThreadPoolExecutor(max_workers=2)
4         pagina = "http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/"
5         wpyget(pagina)
6     except:
7         pass
```

Se utilizó la página que contiene una carpeta de prácticas y algunos zip de código. Notemos que antes de ejecutar el código, en la carpeta donde se ubica el Script, no hay más archivos:

A screenshot of a Windows File Explorer window. The address bar shows the path 'Practica5 redes'. The main area displays a table of files and folders. The table has columns for 'Nombre', 'Estado', 'Fecha de modificación', 'Tipo', and 'Tamaño'. There is one file listed: 'prac5Dos', which is a 'Archivo de origen ...' (Source file) and is 11 KB in size. The 'Estado' column shows a green checkmark.

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
prac5Dos	✓	20/06/2023 08:27 p. m.	Archivo de origen ...	11 KB

Ahora veamos que, al correrlo, se muestran los archivos que se están descargando y se crea la estructura de directorios como está en la URL que se ha puesto en el MAIN:

```
PS C:\Users\joelg> & C:/Users/joelg/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/joelg/OneDrive/Escritorio/Practica5 redes/prac5Dos.py"
Título de la página: IndexofaxelaplicacionessocketscnobloqueantesIPv4IPv6tmp
[HTML/DESCARGANDO] http://148.204.58.221/icons/back.gif back.gif [OK]
[SRC/DESCARGANDO] http://148.204.58.221/icons/blank.gif blank.gif [OK]
[SRC/DESCARGANDO] http://148.204.58.221/icons/text.gif text.gif [OK]
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=N;O=D
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=M;O=A
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=S;O=A
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=D;O=A
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/clienteNB.c
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/clienteNB2.c
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/clienteNBU.c
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/clienteNB_tmp.c
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/servidorNB.c
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/servidorNB2.c
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/servidorNBU.c
[LINK] http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/servidorNB_tmp.c
URI http://148.204.58.221/
['http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=N;O=D', 'http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=M;O=A', 'http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=S;O=A', 'http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/?C=D;O=A', 'http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/', 'http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/clienteNB.c', 'http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/clienteNB2.c', 'http://148.204.58.221/axel/aplicaciones/sockets/c/nobloqueantes/IPv4IPv6/tmp/clienteNBU.c', 'http://148.204.58.221/axel/apli
```

Practica5 redes					
Buscar en Practica5 redes					
	Nombre	Estado	Fecha de modificación	Tipo	Tamaño
	axel	✓	20/06/2023 08:58 p. m.	Carpeta de archivos	
	icons	✓	20/06/2023 08:58 p. m.	Carpeta de archivos	
	prac5Dos	✓	20/06/2023 08:27 p. m.	Archivo de origen ...	11 KB

Conclusiones

Esta práctica fue interesante de hacer, además de que fue la primera que hicimos en Python, fue divertido hacerla, aunque hubo varios detalles que tuvimos que cubrir como el que solo nos daba los link's de las páginas al principio, sin embargo, al final funcionó bien, y se pudieron descargar los contenidos de las carpetas.