



Instituto Politécnico Nacional
Escuela Superior de Cómputo



Aplicaciones para Comunicaciones en Red

Práctica 5: WGet

Alumnos:

Malagón Baeza Alan Adrian

Martínez Chávez Jorge Alexis

Profesor:

Moreno Cervantes Axel Ernesto

Grupo: 6CM1

1. Introducción

1.1 WGET

Wget es una herramienta informática creada por el Proyecto GNU. Puede usarse 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.

GNU Wget tiene muchas funciones para facilitar la recuperación de archivos grandes o la duplicación de sitios web o FTP completos, que incluyen:

- Puede reanudar las descargas canceladas, usando REST y RANGE
- Puede usar comodines de nombre de archivo y directorios espejo recursivamente
- Archivos de mensajes basados en NLS para muchos idiomas diferentes
- Opcionalmente, convierte enlaces absolutos en documentos descargados a relativos, de modo que los documentos descargados puedan vincularse entre sí localmente
- Se ejecuta en la mayoría de los sistemas operativos tipo UNIX, así como en Microsoft Windows
- Admite servidores proxy HTTP
- Soporta cookies HTTP
- Admite conexiones HTTP persistentes
- Operación desatendida en segundo plano
- Utiliza marcas de tiempo de archivos locales para determinar si los documentos deben volver a descargarse al duplicar

1.2 Concurrencia en Java

La concurrencia es la capacidad de hacer más de una cosa al mismo tiempo. A menudo a los desarrolladores de software se nos presenta el problema de, más allá de conseguir que una aplicación funcione correctamente, que lo haga de manera más rápida para satisfacer los requisitos del cliente.

Para esto, conviene diferenciar los conceptos de concurrencia y paralelismo. Concurrencia se da cuando dos o más tareas se desarrollan en el mismo intervalo de tiempo, pero que no necesariamente están progresando en el mismo instante. Es un concepto más general que el paralelismo, el cual consiste en llevar a cabo multitareas en el mismo instante literalmente.

Existen dos conceptos básicos asociados a la concurrencia:

- **Proceso:** es un programa en ejecución. Tiene su propio espacio de memoria, enlaces a recursos, I/O. Los procesos están aislados entre sí.
- **Hilo:** es un camino de ejecución dentro de un proceso. Cada proceso tiene al menos un hilo, llamado hilo principal. Los hilos comparten los recursos del proceso, incluida la memoria, por lo que pueden comunicarse entre sí. Cada hilo tiene su propia callstack.

Una vez que hemos decidido optimizar nuestro programa mediante concurrencia, y aunque no es estrictamente obligatorio conocer el nivel de paralelismo máximo que ofrece la máquina para la cual desarrollamos, es muy importante para hacer previsiones de optimización o para contrastar los resultados en pruebas.

Cada placa base de una computadora dispone de uno o varios sockets donde insertar procesadores. El número de “cores” o procesadores físicos puede ser uno o varios. Si nuestro equipo tiene un socket y n cores, significa que se podrían ejecutar literalmente n procesos realmente en paralelo.

Ahora bien, en Java el código que ejecuta un thread se define en clases que implementan la interfaz Runnable. Para la creación de hilos se cuenta con la clase Thread, la cual permite crear un hilo de ejecución en un programa. La clase Thread cuenta con los siguientes métodos

- run(): actividad del thread
- start(): activo run() y vuelve al llamante
- join(): espera por la terminación (timeout opcional)
- interrupt(): sale de un wait, sleep o join
- isInterrupted()
- yield()
- stop(), suspend(), resume() (deprecated)
- Métodos estáticos
- sleep(milisegundos)
- currentTread()
- Métodos de la clase Object que controlan la suspensión de threads
- wait(), wait(milisegundos), notify(), notifyAll()

2. Desarrollo

2.1 Clase WGet

Esta clase contiene el método principal (main) en el cual se crea el pool de conexiones, el tamaño del pool es establecido por el usuario. Se utiliza la interfaz `ExecutorService` que nos permite ejecutar tareas en hilos de forma asíncrona. `ExecutorService` está presente en el paquete `java.util.concurrent`. Esta interfaz, además ayuda a mantener un conjunto de hilos y les asigna tareas.

```
ExecutorService pool = Executors.newFixedThreadPool(tamPool);
```

Para crear un pool de hilos se utiliza el método `newFixedThreadPool()` de la clase `Executor`.

```
Runnable g = new wget2();  
ExecutorService pool = Executors.newFixedThreadPool(tamPool);  
pool.execute(g);  
pool.shutdown();
```

Luego se ejecuta una tarea con el método `execute()`, en este caso se ejecuta la clase `wget2` que implementa la interfaz `Runnable`. Al final se cierra el servicio ejecutor

2.2 Clase wget2 (implementa Runnable)

La clase `wget2` implementa la interfaz `Runnable`. En el método `run()` se le pide al usuario el URL del recurso a descargar con `WGET`, dándole la opción de descargar un solo recurso (ingresando únicamente la URL) o descargar recursivamente una serie de recursos, para ello se le solicita ingresar el comando `-r` y en seguida el URL en donde se encuentran los recursos a descargar.

Si únicamente se requiere descargar un recurso se utiliza el método `downloadResource()`, el cual recibe como parámetros el link o URL del recurso, la carpeta en donde se guardará y la última parte del URL (es decir el nombre del archivo con su extensión)

En caso de que se desee descargar recursivamente varios recursos, primero se utiliza el método `getData_Download()`, este método recibe como parámetros el URL de los recursos y las partes del URL separadas por el carácter "/" agrupadas en un arreglo llamado `partsURL`. Este método permite mapear cada uno de los recursos a un `HashSet` para su descarga.

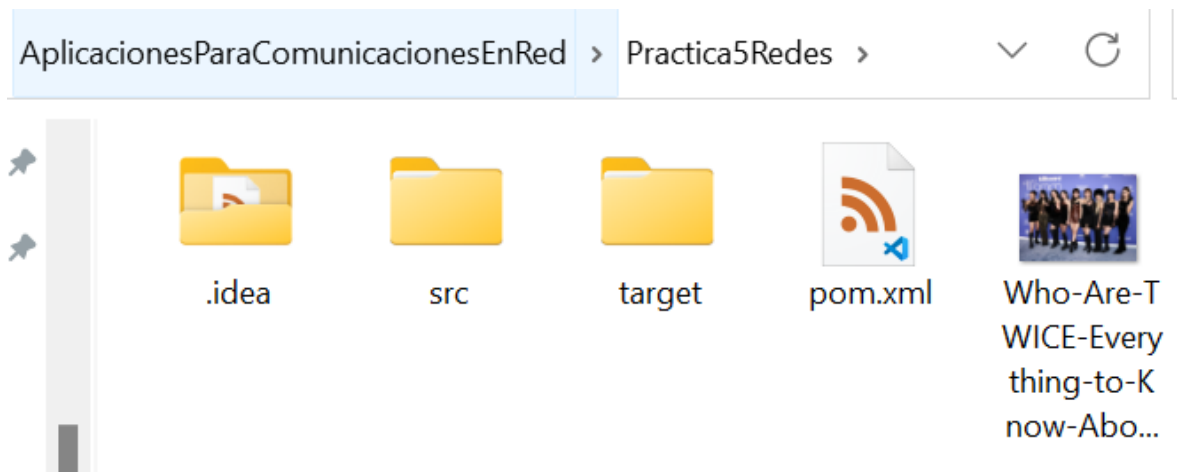
Cabe resaltar que en las descargas de cada uno de los recursos se verifica que el link a descargar sea válido o esté disponible, verificar que el nombre corresponda a un recurso, obtener el nombre del recurso correctamente y construir la ruta en la que se descargará el recurso (path). Además de que si se trata de imágenes, reescribir sus direcciones y descargarlas en el formato requerido. Todo esto se implementó para que la descarga de los recursos se llevará a cabo correctamente.

3. Pruebas

La primera prueba es descargar un recurso, en este caso una imagen. Primeramente se introduce el tamaño del pool de hilos, luego se introduce el URL <https://www.usmagazine.com/wp-content/uploads/2023/03/Who-Are-TWICE-Everything-to-Know-About-the-Breakthrough-K-Pop-Girl-Group.jpg>. En la imagen siguiente se observa que se realiza la descarga y se guarda en la carpeta del paquete Java.



```
WGet x
C:\Users\alanm\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\I
Introduzca el tamaño del pool de hilos:
12
NOTA -r: modo recursivo
Introduzca la URL:
https://www.usmagazine.com/wp-content/uploads/2023/03/Who-Are-TWICE-Everything-to-Know-Ab
Direccion: https://www.usmagazine.com/wp-content/uploads/2023/03/Who-Are-TWICE-Everything
Descargado: Who-Are-TWICE-Everything-to-Know-About-the-Breakthrough-K-Pop-Girl-Group.jpg
```



Ahora se procederá a descargar una serie de recursos de la página del profesor, para ello introducimos el tamaño del pool de conexiones, luego se introduce el modo recursivo `-r` y posteriormente el URL que para este caso es <http://148.204.58.221/axel/aplicaciones/sockets/java/canales/>. Inmediatamente después se despliega información sobre las descargas a realizar, como la dirección del servidor, y la profundidad de los archivos y carpetas, esto es para llevar un mejor control.

```
WGet x
C:\Users\alanm\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\
Introduzca el tamaño del pool de hilos:
12
NOTA -r: modo recursivo
Introduzca la URL:
-r http://148.204.58.221/axel/aplicaciones/sockets/java/canales/
-r
http://148.204.58.221/axel/aplicaciones/sockets/java/canales/
-- -- -- -- --
Links en donde se descargaran recursos: http://148.204.58.221/axel/aplicaciones/sockets/
Dirección anterior: http://148.204.58.221/axel/aplicaciones/sockets/java/
Dirección del servidor: http://148.204.58.221/
Dirección nueva: 148.204.58.221/axel/aplicaciones/sockets/java/canales/
Profundidad : 4
```

Personal

148.204.58.221 > axel > aplicaciones > sockets > java > canales

Buscar en canales

Nombre	Fecha de modificación	Tipo
ejemplo_NIO	30/05/2023 08:57 a. m.	Carpeta de arc
flujo	30/05/2023 08:57 a. m.	Carpeta de arc
Serializacion	30/05/2023 08:57 a. m.	Carpeta de arc
canales.zip	30/05/2023 08:57 a. m.	Carpeta comp
CNB.java	30/05/2023 08:57 a. m.	Archivo de ori
duke1.png	30/05/2023 08:57 a. m.	Archivo PNG
EchoClient.java	30/05/2023 08:57 a. m.	Archivo de ori
EchoClient2.java	30/05/2023 08:57 a. m.	Archivo de ori
EchoServer.java	30/05/2023 08:57 a. m.	Archivo de ori
EcoC.java	30/05/2023 08:57 a. m.	Archivo de ori
EcoS.java	30/05/2023 08:57 a. m.	Archivo de ori
index.html	30/05/2023 08:57 a. m.	Chrome HTML
input.txt	30/05/2023 08:57 a. m.	Documento d
MultiC.java	30/05/2023 08:57 a. m.	Archivo de ori
MulticastChannel.docx	30/05/2023 08:57 a. m.	Documento d

temas

iles

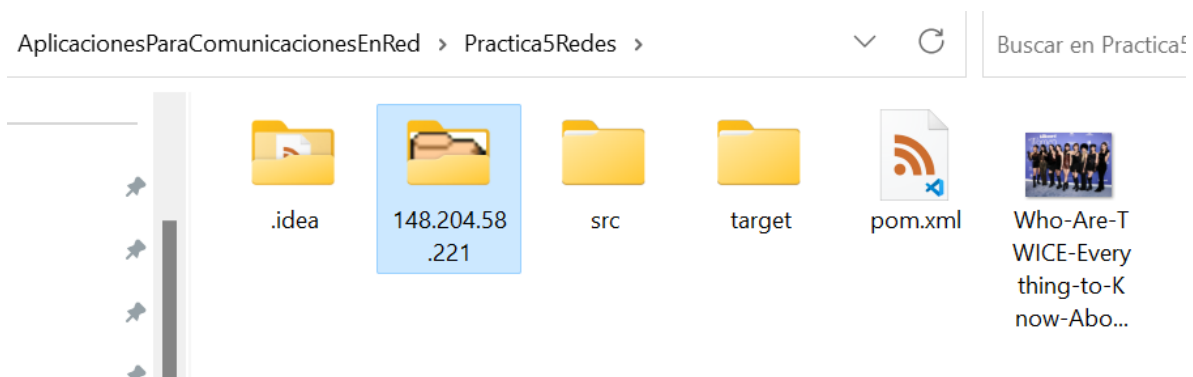
Como se puede observar se imprime cada una de las descargas que se están llevando a cabo, imprimiendo el nombre del recurso descargado y la dirección de donde se hizo la descarga. En la siguiente imagen por ejemplo, se observa que se descargaron archivos con extensiones java, zip y png.

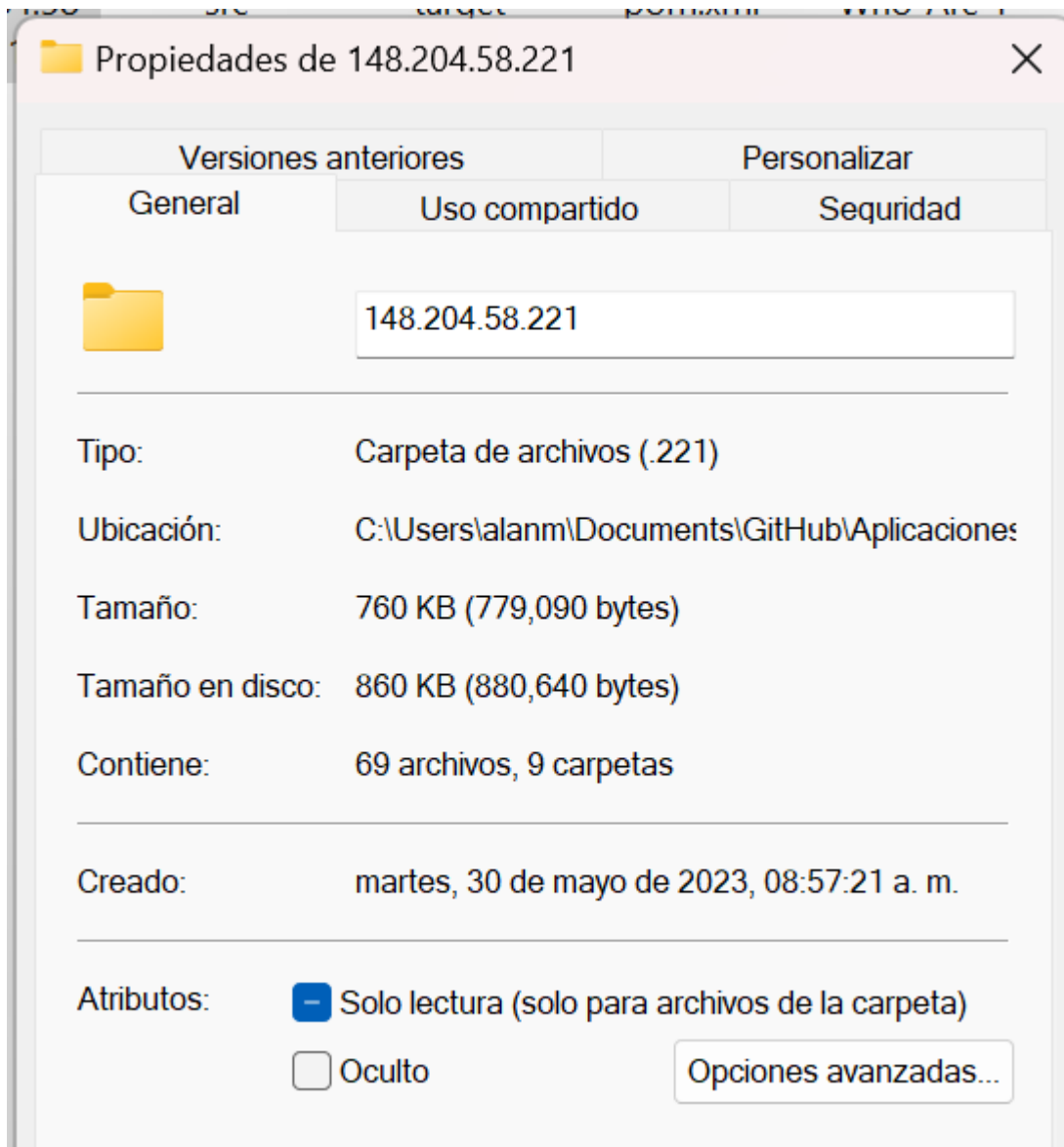
```
USNBC.java
----|http://148.204.58.221/axel/aplicaciones/sockets/java/canales/USNBC.java
Direccion: http://148.204.58.221/axel/aplicaciones/sockets/java/canales/USNBC.java
Descargado: USNBC.java

canales.zip
----|http://148.204.58.221/axel/aplicaciones/sockets/java/canales/canales.zip
Direccion: http://148.204.58.221/axel/aplicaciones/sockets/java/canales/canales.zip
Descargado: canales.zip

duke1.png
----|http://148.204.58.221/axel/aplicaciones/sockets/java/canales/duke1.png
Direccion: http://148.204.58.221/axel/aplicaciones/sockets/java/canales/duke1.png
Descargado: duke1.png
```















La carpeta se crea en la dirección en la que se encuentra el proyecto, en el ejemplo es la carpeta 148.204.58.221. Además, en este caso se crearon 9 carpetas con 69 archivos en total





Ahora si descargamos otra serie de recursos de la misma página del profesor, por ejemplo la siguiente URL <http://148.204.58.221/axel/aplicaciones/practicas/> , se creará la carpeta correspondiente en el path establecido descargando los recursos faltantes. Es decir en la carpeta aplicaciones se creará ahora la carpeta practicas con los recursos descargados. Eso se puede observar en la siguiente imagen

```
WGet x
C:\Users\alanm\.jdk\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\Int
Introduzca el tamaño del pool de hilos:
12
NOTA -r: modo recursivo
Introduzca la URL:
-r http://148.204.58.221/axel/aplicaciones/practicas/
-r
http://148.204.58.221/axel/aplicaciones/practicas/
-- -- -- -- --
Links en donde se descargarán recursos: http://148.204.58.221/axel/aplicaciones/practicas/
Dirección anterior: http://148.204.58.221/axel/aplicaciones/
Dirección del servidor: http://148.204.58.221/
Dirección nueva: 148.204.58.221/axel/aplicaciones/practicas/
Profundidad : 2
-- -- -- -- --
Dirección: http://148.204.58.221//icons/blank.gif
Descargado: blank.gif
```

> 148.204.58.221 > axel > aplicaciones > practicas ⌵ ↺			Buscar en practicas 🔍
Nombre	Fecha de modificación	Tipo	
 index.html	30/05/2023 09:02 a. m.	Chrome HTML	
 Pr%e1ctica%202_batalla_naval.docx	30/05/2023 09:02 a. m.	Documento de	
 Pr%e1ctica%203.doc	30/05/2023 09:02 a. m.	Documento de	
 Pr%e1ctica%205.doc	30/05/2023 09:02 a. m.	Documento de	
 Pr%e1ctica%206.doc	30/05/2023 09:02 a. m.	Documento de	
 Pr%e1ctica%207.doc	30/05/2023 09:02 a. m.	Documento de	
 Pr%e1ctica%208.doc	30/05/2023 09:02 a. m.	Documento de	
 pr%e1ctica_2_buscaminas.pdf	30/05/2023 09:02 a. m.	Documento Ad	
 practica1.doc	30/05/2023 09:02 a. m.	Documento de	
 practica2.doc	30/05/2023 09:02 a. m.	Documento de	
 practica2_carrito_compras.pdf	30/05/2023 09:02 a. m.	Documento Ad	
 practica3_chat.pdf	30/05/2023 09:02 a. m.	Documento Ad	
 practica4.doc	30/05/2023 09:02 a. m.	Documento de	
 practica5%20WGET.doc	30/05/2023 09:02 a. m.	Documento de	

De la misma forma se imprimen cada una de las descargas que se están llevando a cabo, imprimiendo el nombre del recurso descargado y la dirección de donde se hizo la descarga. En la siguiente imagen por ejemplo, se observa que se descargaron archivos con extensiones pdf.

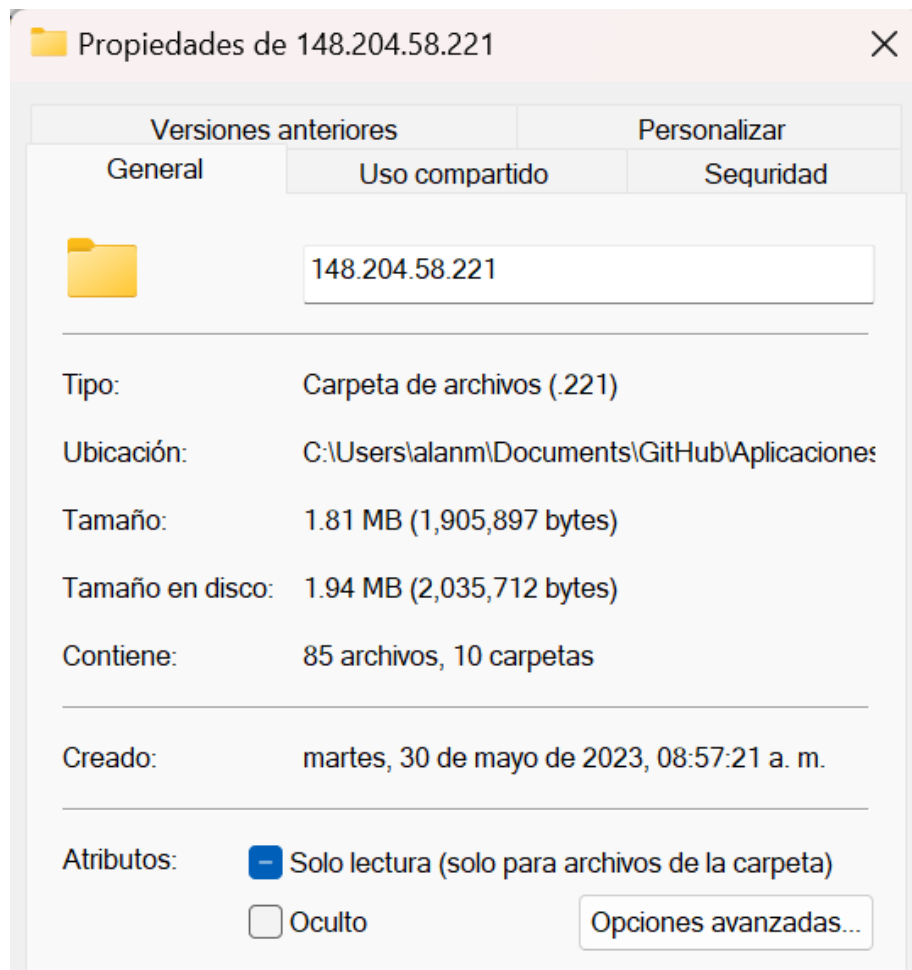
```

practica2_carrito_compras.pdf
----|http://148.204.58.221/axel/aplicaciones/practicas/practica2\_carrito\_compras.pdf
Direccion: http://148.204.58.221/axel/aplicaciones/practicas/practica2\_carrito\_compras.pdf
Descargado: practica2_carrito_compras.pdf

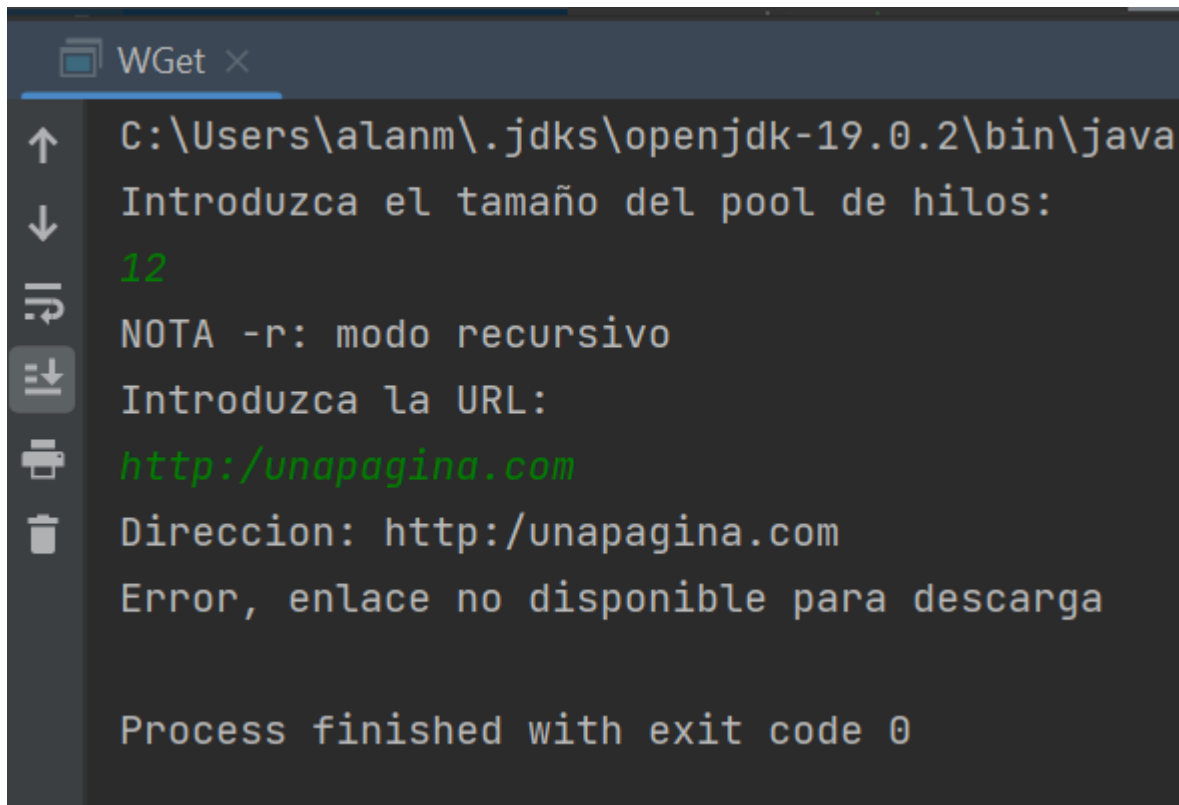
practica3_chat.pdf
----|http://148.204.58.221/axel/aplicaciones/practicas/practica3\_chat.pdf
Direccion: http://148.204.58.221/axel/aplicaciones/practicas/practica3\_chat.pdf
Descargado: practica3_chat.pdf

```

Ahora si vemos las propiedades de la carpeta se observa que se tiene un total de 85 archivos distribuidos en 10 carpetas.



Finalmente, si se tiene un enlace que no es válido se le muestra al usuario el error correspondiente y que el recurso no está disponible para descargar

A screenshot of a terminal window titled 'WGet'. The terminal shows the execution of a Java command to run WGet. It prompts for a thread pool size, which is set to 12. It then prompts for a URL, which is 'http://unapagina.com'. The output shows the direction and an error message: 'Error, enlace no disponible para descarga'. The process ends with an exit code of 0.

```
WGet x
C:\Users\alanm\.jdk\openjdk-19.0.2\bin\java
Introduzca el tamaño del pool de hilos:
12
NOTA -r: modo recursivo
Introduzca la URL:
http://unapagina.com
Direccion: http://unapagina.com
Error, enlace no disponible para descarga

Process finished with exit code 0
```

4. Preguntas

1. ¿Qué usos adicionales se le pueden dar a esta herramienta?

Se podría implementar métodos adicionales para la descarga de archivos en segundo plano, descargar y extraer archivos con un solo comando, rechazar tipos de archivos durante la descarga e incluso establecer la velocidad de descarga. De esta manera serviría como herramienta para sostener conexiones de red inestables y lentas. Además, si se produce un problema de red durante una descarga, se podría reanudar la recuperación de los archivos sin empezar desde cero.

2. ¿Se obtiene alguna ventaja al usar una alberca de hilos para implementar la descarga de los archivos?, ¿cuál?

Si se obtienen ventajas de rendimiento al implementar procesos multihilos , ya que es más rápido crear y eliminar un nuevo hilo que un nuevo proceso, además los hilos se pueden comunicar entre sí sin tener que invocar al kernel del sistema operativo.

5. Conclusiones

Al desarrollar esta práctica se logró el objetivo que era realizar una aplicación que implemente la descarga WGET para uno o más recursos a través de sus respectivos URL's. Cabe resaltar que al principio resultó complicado implementar la descarga recursiva de múltiples recursos, puesto que en ocasiones el programa se ciclaba infinitamente, por lo que se tuvieron que crear varios métodos para separar el URL, definir la ruta correspondiente a cada uno de los archivos a descargar y establecer la profundidad de los archivos descargados. Esto también nos sirvió para que los recursos mantuvieran el nombre y la estructura del archivo original. Además, también se incorporó la validación de los links respectivos. Finalmente para que el programa fuera concurrente se creó un pool de conexiones y de esta manera mejorar el rendimiento del programa.

Bibliografía

- Wget - GNU Project - Free Software Foundation. Gnu.org. (2022). Retrieved 1 June 2022, from <https://www.gnu.org/software/wget/>.
- B., G. (2022). ¿Qué es y cómo usar el comando wget?. Tutoriales Hostinger. Retrieved 2 June 2022, from <https://www.hostinger.es/tutoriales/usar-comando-wget/>.
- Díaz, D. (2022). Introducción a la Concurrencia en Java (I). Blog.softtek.com. Retrieved 2 June 2022, from <https://blog.softtek.com/es/java-concurrency>.