**--------------APLICACIONES EN COMUNICACIONES EN RED------------**

**PRÁCTICA 5:**

Wget

**Alumno:**

Meza Vargas Brandon David

**Grupo:**

3CM16

**Profesor:**

Moreno Cervantes Axel Ernesto

**Índice**

[**Introducción** 4](#_Toc104313020)

[**Desarrollo** 5](#_Toc104313021)

[**FileDownloader** 5](#_Toc104313022)

[**getFiles** 6](#_Toc104313023)

[**getFileName** 6](#_Toc104313024)

[**downloadFiles** 7](#_Toc104313025)

[**isDir** 7](#_Toc104313026)

[**Main** 8](#_Toc104313027)

[**Constants** 9](#_Toc104313028)

[**Pruebas de funcionamiento** 9](#_Toc104313029)

[**Conclusiones** 14](#_Toc104313030)

[**Bibliografía** 15](#_Toc104313031)

**Índice de ilustraciones**

[Ilustración 1. Primer constructor. de FileDownloader 5](#_Toc104313032)

[Ilustración 2. Segundo constructor de FileDownloader 6](#_Toc104313033)

[Ilustración 3. Método getfiles. 6](#_Toc104313034)

[Ilustración 4. Método getFileName 6](#_Toc104313035)

[Ilustración 5. Método downloadFiles 7](#_Toc104313036)

[Ilustración 6. Método isDir 8](#_Toc104313037)

[Ilustración 7. Clase Main 9](#_Toc104313038)

[Ilustración 8. Clase Constants 9](#_Toc104313039)

[Ilustración 9. Corriendo el programa 10](#_Toc104313040)

[Ilustración 10. Descargando archivo. 10](#_Toc104313041)

[Ilustración 11. Archivo descargado. 10](#_Toc104313042)

[Ilustración 12. Archivo visualizado 11](#_Toc104313043)

[Ilustración 13. Descargando archivos de una carpeta. 13](#_Toc104313044)

[Ilustración 14. Descarga de carpeta. 13](#_Toc104313045)

[Ilustración 15. Contenido de carpeta. 14](#_Toc104313046)

# **Introducción**

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

En la práctica actual se realizará una copia de este pequeño programa usando el lenguaje de programación Java e implementando flujos de entrada y salida para hacer la descarga de archivos, además de archivos se descargarán carpetas completas haciendo uso de la recursión y creando un hilo por cada archivo que haya para tener una descarga rápida, esto usando un pool de hilos.

# **Desarrollo**

A continuación se muestra el desarrollo de la práctica explicando el código que lo conforma.

## **FileDownloader**

Esta es la parte principal del programa de la práctica, en esta parte tenemos dos constructores de la clase, el primero recibe como parámetros un link, un path y la operación, esta operación indicará si se trata de una nueva descarga de un archivo o para descargar un html que tendrá el índice de los archivos contenidos en caso de que sea una carpeta.

En el primer constructor lo primero que se hace es crear una nueva URL con el link y si es la operación 1 se obtienen los archivos del link, en caso contrario se descarga el archivo.

Texto

Descripción generada automáticamente

Ilustración . Primer constructor. de FileDownloader

En el segundo constructor tenemos como parámetros un link, path, el pool de hilos y la operación, es exactamente lo mismo que el anterior, con la diferencia que aquí mandamos el pool de hilos para hacer la descarga de una manera más rápida.

Texto

Descripción generada automáticamente

Ilustración . Segundo constructor de FileDownloader

### **getFiles**

Este método verifica si el archivo es una carpeta o un solo archivo, en caso de que sea una carpeta se manda a llamar al método isDir(), en caso de que no se descarga el archivo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración . Método getfiles.

### **getFileName**

Este método solo retorna el nombre del archivo a descargar, esto lo hacemos cortando el nombre que viene de la url, para esto cortamos por cada / que se encuentre en la URL obteniendo así un array de strings, sabiendo que el nombre del archivo será el que viene hasta la última posición del arreglo, será esta la que regresaremos.

Texto

Descripción generada automáticamente

Ilustración . Método getFileName

### **downloadFiles**

Este método se encarga de descargar un archivo especificado por el usuario o los archivos que viene dentro de la carpeta, lo que se hace es crear un fulo de entrada y de salida del archivo que el usuario descargará, posteriormente solo iremos leyendo cada byte del archivo recibido por el flujo de entrada y lo iremos escribiendo hasta que no haya más bytes por leer.

Texto

Descripción generada automáticamente

Ilustración . Método downloadFiles

### **isDir**

Este método es muy importante en el programa, primeramente creamos un archivo a partir de la ruta y el nombre del archivo obtenido de la url ingresada por el usuario, posteriormente creamos una carpeta ya que este método nos indica que el archivo será una carpeta.

Posteriormente crearemos un flujo de entrada con el método openStream de la URL, este método crea una conexión TCP al servidor que la URL resuelve, de esta manera se manda una petición GET al servidor, que en este caso es nuestro navegador y devuelve el archivo, posteriormente leemos los bytes de ese flujo hasta que ya no haya datos, posteriormente vamos recorriendo cada archivo y ejecutando un hilo gracias a nuestro pool para que se encargue de ese archivo, se mandará entones el link del archivo actual, el path, el pool de hilos y la operación 0, esta indicará que se descargará el archivo.

. Texto

Descripción generada automáticamente

Ilustración . Método isDir

## **Main**

En esta clase solo se crea el scanner para que el usuario ingrese texto por el teclado y se manda a llamar a la clase FileDownloader para comenzar con el proceso de descarga de archivos.

Texto

Descripción generada automáticamente

Ilustración . Clase Main

## **Constants**

En esta clase solo se encuentran las constantes usadas a lo largo del programa principal.

Texto

Descripción generada automáticamente

Ilustración . Clase Constants

## **Pruebas de funcionamiento**

A continuación se mostrará el funcionamiento de la práctica a través de capturas del programa funcionando.

Al correr el programa, esta pregunta por un link al usuario para descargar archivos de el.

Texto

Descripción generada automáticamente con confianza media

Ilustración . Corriendo el programa

Una vez que el usuario ingrese un link con un archivo, el programa lo descargará y preguntará al usuario si desea ingresar otro link como se ve a continuación.

Texto

Descripción generada automáticamente

Ilustración . Descargando archivo.

El archivo lo podremos ver en la raíz del proyecto como se ve en la siguiente captura.



Ilustración . Archivo descargado.

Si lo abrimos, vemos que se puede visualizar sin ningún problema.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ilustración . Archivo visualizado

Ahora probaremos con un archivo que s una carpeta que contiene archivos dentro de ella así como más carpetas.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Ilustración . Descargando archivos de una carpeta.

Podemos ver que de igual forma la descarga se encuentra en la raíz del proyecto.



Ilustración . Descarga de carpeta.

Si abrimos la carpeta observamos que todos los archivos se encuentran ahí.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ilustración . Contenido de carpeta.

# **Conclusiones**

Esta práctica estuvo divertida e interesante, pues gracias a lo que hemos aprendido hasta el momento del curso podemos hacer una copia del programa wget de una manera rápida y sencilla usando lenguaje Java en mi caso y flujos de entrada y salida asociándolos a archivos creados a partir de una URL que el usuario ingresa.

La parte complicada vino cuando se tenía que crear un hilo por cada archivo en caso de que hubiera varios en una carpeta para acelerar el proceso de descarga, sin embargo, con recursión a la misma clase y ejecutando el pool de hilos por cada archivo se solucionó esta parte y se logro que la descarga sea muy rápida pues un hilo diferente se encarga de un archivo diferente.

Una buen práctica para poner a prueba ciertos conocimientos e investigar por nuestra parte ciertos aspectos.

# **Bibliografía**

1. G. Briones. (09 de marzo, 2022). Qué es y cómo usar el comando wget. [Online]. Available: <https://www.hostinger.mx/tutoriales/usar-comando-wget/#%C2%BFQue_es_el_Comando_Wget>