*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Aplicaciones para Comunicaciones de Red*

***Practica 1: Servicio de transferencia de archivos***

***Integrantes:***

* *Ornelas García Luis Ángel*
* *Sampayo Hernández Mauro*

***Grupo:***

***Profesor:*** *Moreno Cervantes Axel Ernesto*

**Introducción:**

El envío de archivos a través de la red es una característica importante para la gran mayoría de las aplicaciones que hoy día se utilizan (blogs, redes sociales, mensajería instantánea, Declaración de impuestos, educación en línea, etc.), sin embargo, no todas las aplicaciones disponibles permiten el envío de archivos de gran tamaño (p.e. El correo electrónico no permite enviar archivos de más de 10 o 20 MB). Esto hace necesario el desarrollo de aplicaciones que permitan transferir archivos sin importar el tamaño de éstos.

**Desarrollo:**

**Cliente**

Nuestro cliente al inicializar utiliza la clase **Client** la cual al pide el nombre de un cliente, si la carpeta existe, busca todos los documentos y carptetas y los muestra en la consola; si la carpte no existe, crea una nueva con el nombre del cliente mostrando un contenido **null** debido a que la carpeta esta vacia.

La función **enviarArch** recibe como parámetros un File el cual tiene la dirección de la carpeta del cliente, el nombre del cliente y un DataOutputStream para enviar archivos uno por uno

La función **envarAlgo** recibe de parámetros un File[], el nombre del cliente y los DataOutputStream y DataInputStream para comunicarse con el servidor. Si detecta que en el File hay un archivo envía al servidor un booleano en false para que se prepare para recibir un archivo, en caso contrario recibirá una carpeta

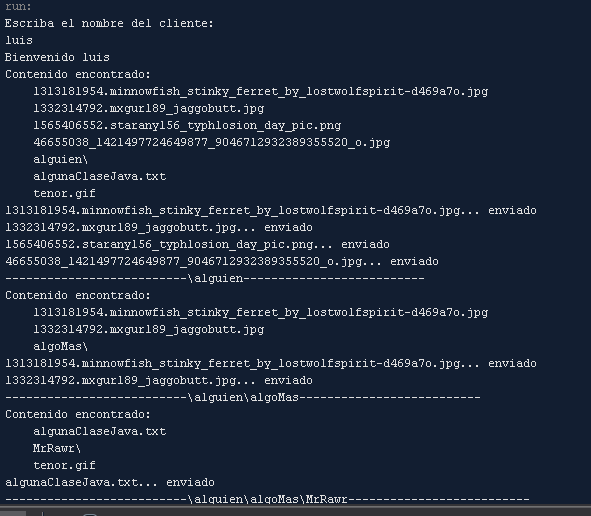
**Servidor**

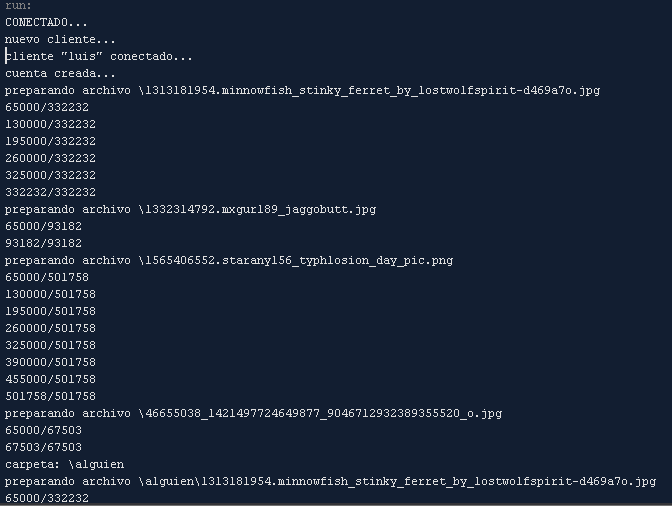
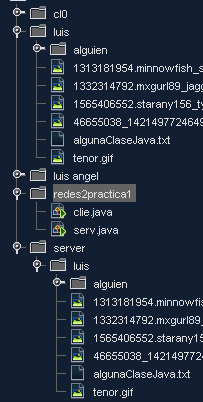
Nuestro servidor al inicializarse espera un cliente, cuando hace contacto con este le pide un nombre para crear en la raíz del servidor la carpeta para el cliente, utilzando el constructor de la clase **server**.

En la clase **server** tenemos el método **recibeArch** el cual recibe como parámetro un File el cual tiene la dirección de la carpeta del cliente en el servidor y un DataInputStream. Este método se utiliza para recibir archivos uno por uno.

El método **recibeAlgo** el cual se ejecuta después de hacer la conexión con el cliente, recibe como parámetro un File con la dirección de la carpeta del cliente y los DataOutputStream y DataInputStream. Este método recibe un valor entero para ir recorriendo la lista de carpetas que recibimos, utiliza el método **recibeArch** para recibir archivos y el método **recibeAlgo** para crear las carpetas y recibir archivos que están dentro de otras carpetas

**Pruebas:**



**Conclusión:**

Hoy en día este tipo de servicios resultan bastante útiles, pues nos permiten guardar o respaldar archivos importantes en una nube, ahorrándonos el tener que gastar dinero en una unidad de memoria externa y tener que mudar nuestros archivos de una unidad a otra si es que esta se queda obsoleta; además que nos facilita el pasar archivos desde un ordenador a otro.

**Bibliografía:**

* Oracle Help Center. 2020. *JDK 14 Documentation - Home*. [online] Available at: <https://docs.oracle.com/en/java/javase/14/>
* Docs.oracle.com. 2020. *Overview (Java Platform SE 7)*. [online] Available at: <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>