Práctica 2

Integrantes:

- **Flores Morales Aldahir Andrés**
- **Velasco Jiménez Luis Antonio**

Fecha de entrega: 14/11/2024

Materia: Aplicaciones para comunicaciones en red

Grupo: 6CM1

Introducción

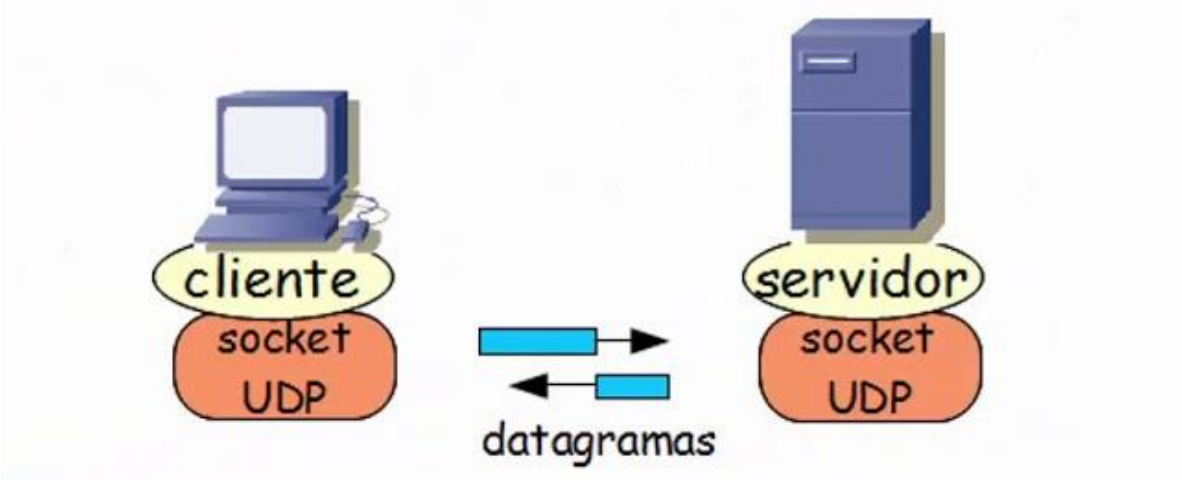
En el ámbito de las comunicaciones en redes, los sockets de datagrama juegan un papel crucial para implementar sistemas de comunicación rápida y ligera entre dispositivos. A diferencia de los sockets orientados a conexión (como TCP), los sockets de datagrama (basados en el protocolo UDP, User Datagram Protocol) no requieren de una conexión establecida y garantizada entre el emisor y el receptor, lo que los hace ideales para aplicaciones donde la velocidad y eficiencia son más importantes que la confiabilidad absoluta, como en transmisiones de audio o video en tiempo real, y sistemas de control en red.

Esta práctica se centra en comprender el funcionamiento de los sockets de datagrama. El servidor será capaz de recibir mensajes de uno o varios clientes, y los clientes, a su vez, podrán enviar datos en forma de paquetes al servidor sin necesidad de mantener una conexión permanente. Esto implica el uso de paquetes pequeños que se envían y reciben de manera independiente, permitiendo explorar conceptos como la fragmentación, el manejo de pérdida de paquetes y los mecanismos de retransmisión en caso de errores.

Durante la práctica, implementaremos un sistema de transferencia de archivos básico sobre UDP.

Sockets de Datagrama

Los sockets de datagrama son una herramienta fundamental en el ámbito de la comunicación de redes de computadoras y se utilizan ampliamente para implementar aplicaciones de red que requieren transmisión rápida y eficiente de datos sin la necesidad de establecer una conexión continua entre los dispositivos que participan en la comunicación. A diferencia de los sockets orientados a conexión (como los de TCP), los sockets de datagrama operan sobre el protocolo UDP (User Datagram Protocol), que es un protocolo de transporte no orientado a conexión. Esto significa que los datos se envían en paquetes independientes llamados datagramas, sin confirmación de llegada o garantías de reordenamiento en la entrega.



Características de los Sockets de Datagrama

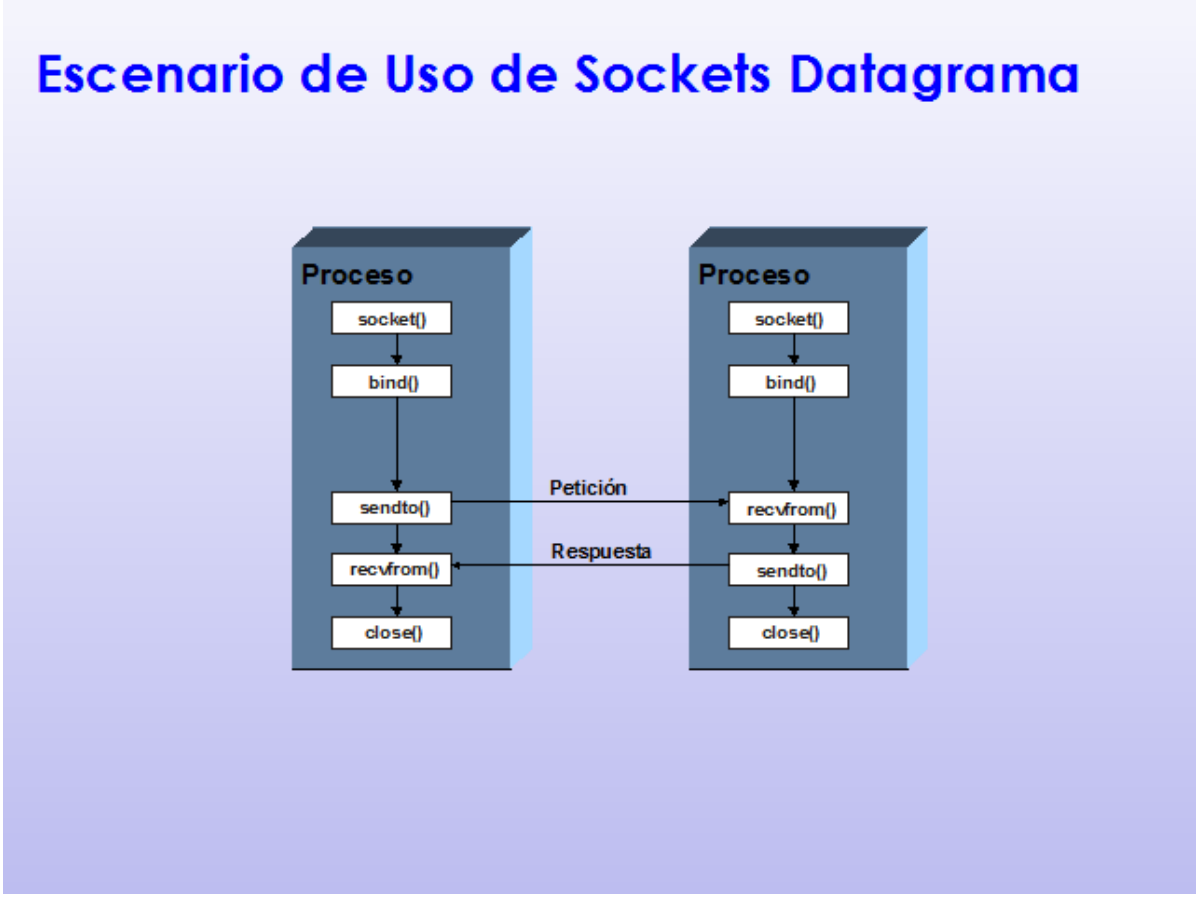
1. **Transmisión sin conexión** : Los sockets de datagrama permiten la transmisión de datos sin establecer una conexión continua entre el emisor y el receptor. Cada datagrama enviado es independiente y no necesita de una sesión o enlace entre el cliente y el servidor. Esto contrasta con TCP, que requiere establecer y mantener una conexión para el intercambio de información.
2. **Protocolo UDP (User Datagram Protocol)** : Los sockets de datagramas se basan en UDP, un protocolo de transporte que no garantiza la entrega confiable de los paquetes. Los paquetes enviados pueden llegar a su destino en el orden incorrecto, llegar duplicados o incluso perderse sin aviso. Aunque esto puede parecer una limitación, resulta ventajoso en aplicaciones donde la velocidad es más crítica que la confiabilidad, como en transmisiones de video en tiempo real, juegos en línea o sistemas de monitoreo en red.
3. **Unidad de Transmisión: el Datagrama** : En los sockets de datagrama, la información se empaqueta en pequeñas unidades llamadas datagramas. Cada datagrama es un paquete que contiene una porción de datos, así como la información de encabezado necesaria para ser dirigido al destino correcto. Los datagramas no tienen una longitud fija, aunque su tamaño generalmente se limita a 65.535 bytes debido a las restricciones del protocolo UDP. Los datagramas permiten a las aplicaciones de red dividir grandes cantidades de datos en paquetes más pequeños que se pueden enviar de manera independiente.
4. **Velocidad y Eficiencia** : La ausencia de mecanismos de control de flujo y verificación en los sockets de datagrama permite una transmisión de datos rápida y con baja latencia. Esto los hace ideales para aplicaciones en tiempo real que requieren que los datos lleguen rápidamente sin importar si algunos paquetes se pierden en el camino. Por ejemplo, en una transmisión de voz o video en tiempo real, perder algunos paquetes no suele afectar significativamente la calidad de la transmisión, mientras que retrasar los paquetes puede causar un impacto negativo.
5. **Simpleza de Implementación** : Los sockets de datagrama son simples de implementar, ya que no se requiere el código adicional para gestionar la creación, mantenimiento y cierre de una conexión. Solo es necesario crear el socket, especificar el destino y enviar el datagrama. Esto hace que los sockets de datagrama sean especialmente útiles para aplicaciones donde se requieren muchas comunicaciones breves con diferentes destinos, como en sistemas de detección de red o envío de comandos simples a Múltiples dispositivos.

Funcionamiento Básico de los Sockets de Datagramas

El proceso de envío y recepción de información a través de sockets de datagrama implica los siguientes pasos:

1. **Creación del Socket** : Tanto el emisor como el receptor deben crear un socket de datagrama. En la mayoría de los lenguajes de programación, este proceso es sencillo y solo requiere especificar el tipo de socket (en este caso, un socket de datagrama basado en UDP).
2. **Creación y Envío de Datagramas** : Para enviar datos, el emisor debe empaquetar los datos en un datagrama, que incluye tanto la información de los datos como la dirección IP y el número de puerto del receptor. Una vez configurado el datagrama, se envía al receptor sin esperar ninguna respuesta.

3. **Recepción del Datagrama** : El receptor, por su parte, escucha en un puerto específico, esperando recibir datagramas. Al recibir uno, el receptor desempaqueta el datagrama para obtener los datos y la información sobre el emisor. En la mayoría de las implementaciones, el receptor debe especificar el tamaño máximo del datagrama que espera recibir.
4. **Procesamiento de los Datos** : Una vez que el datagrama es recibido, la aplicación puede procesar los datos según sea necesario. Si el protocolo de la aplicación requiere algún tipo de confirmación de llegada o reordenamiento de los paquetes, estos mecanismos deben implementarse a nivel de aplicación, ya que UDP no los proporciona.
5. **Reintentos y Control de Errores (si es necesario)** : Si la aplicación requiere confiabilidad en la entrega de los datagramas, es necesario agregar mecanismos adicionales. Esto puede incluir la solicitud de reenvío de paquetes faltantes o la implementación de un número de secuencia para ordenar los paquetes en el receptor.



Ventajas y Desventajas de los Sockets de Datagramas

Ventajas :

- **Rapidez** : Al no requerir confirmación de entrega, los datagramas pueden ser enviados y recibidos con baja latencia, ideal para aplicaciones en tiempo real.
- **Bajo uso de recursos** : Sin la necesidad de establecer y mantener una conexión, el

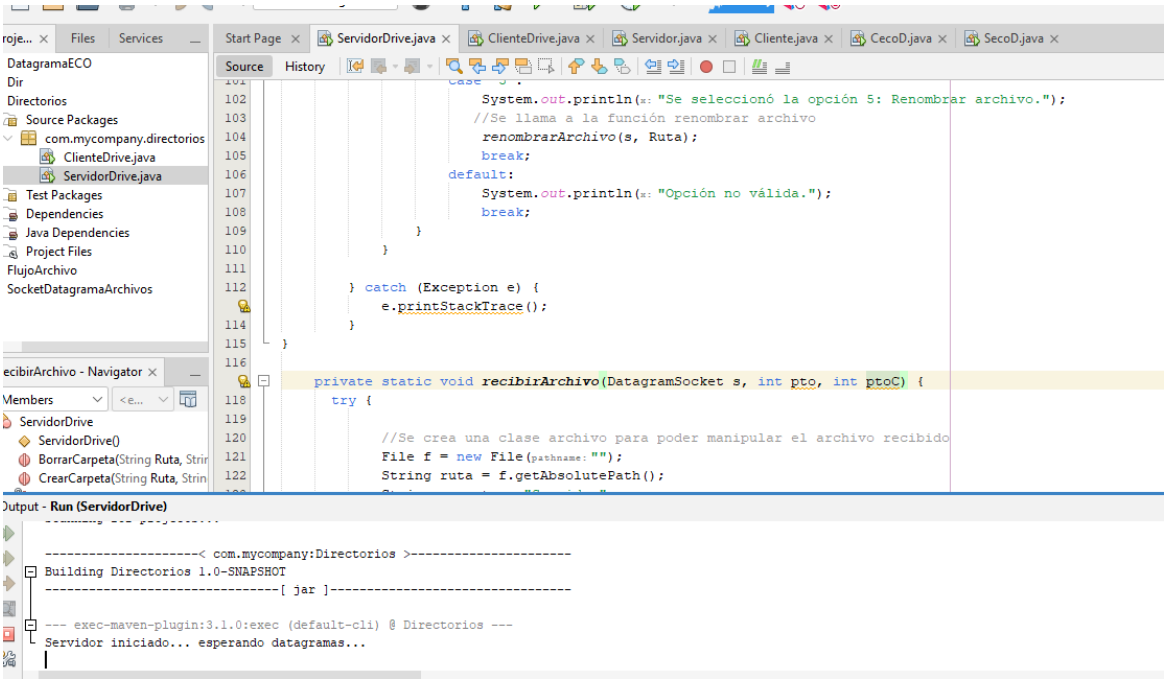
uso de CPU y memoria es menor.
- **Simpleza de diseño** : Menos código para manejar la conexión y la retransmisión, útil para aplicaciones ligeras o de corta duración.

Desventajas :
- **No confiable** : UDP no garantiza la entrega de paquetes, el reordenamiento ni la eliminación de duplicados, lo que implica que la aplicación debe implementar estos mecanismos si son necesarios.
- **Limitaciones de tamaño** : Los datagramas tienen un tamaño máximo de aproximadamente 65 KB, lo que puede ser un inconveniente para enviar grandes cantidades de datos en un solo paquete.
- **Sin control de congestión** : UDP no incluye mecanismos de control de congestión, lo que puede causar problemas en redes saturadas.

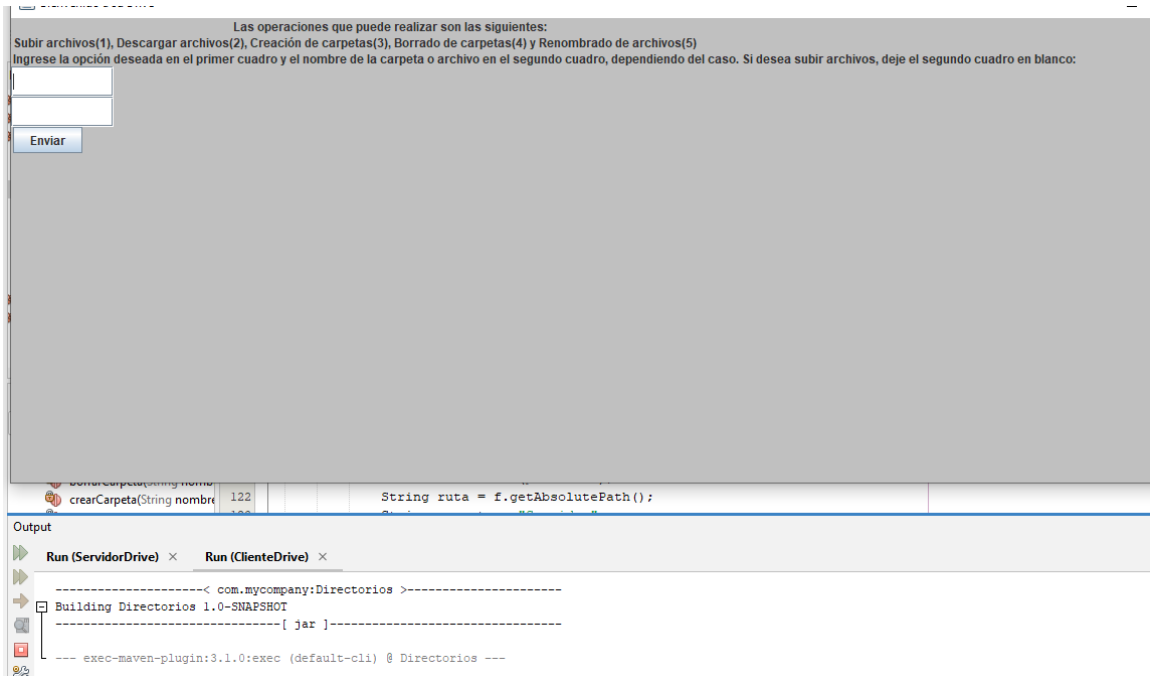
Ejemplos de uso de Sockets de Datagramas
Algunos ejemplos comunes de aplicaciones que utilizan sockets de datagrama son:
1. **Transmisiones de Audio y Video en Tiempo Real** : Aplicaciones como VoIP (Voice over IP) y transmisiones en vivo de video utilizan UDP para minimizar la latencia. En estos casos, algunos paquetes perdidos no afectan demasiado la experiencia del usuario, pero una alta latencia sí lo haría.
2. **Juegos en Línea** : En muchos juegos en línea, especialmente los juegos de disparos en primera persona y los juegos multijugador en tiempo real, la rapidez es crucial. UDP permite que los datos de la posición del jugador o los eventos de juego se envíen rápidamente sin necesidad de confirmación de entrega.
3. **DNS (Domain Name System)** : El protocolo DNS utiliza UDP para enviar consultas de nombres de dominio debido a que las consultas DNS son pequeñas y no necesitan una conexión sostenida. Además, la rapidez de UDP permite una resolución de nombres más eficiente.
4. **Protocolos de Descubrimiento de Redes** : Muchos protocolos de descubrimiento de dispositivos en la red, como el Protocolo de Configuración Dinámica de Host (DHCP), se basan en UDP para enviar mensajes de configuración de red debido a la rapidez y la falta de requisitos de conexión .

Pruebas:

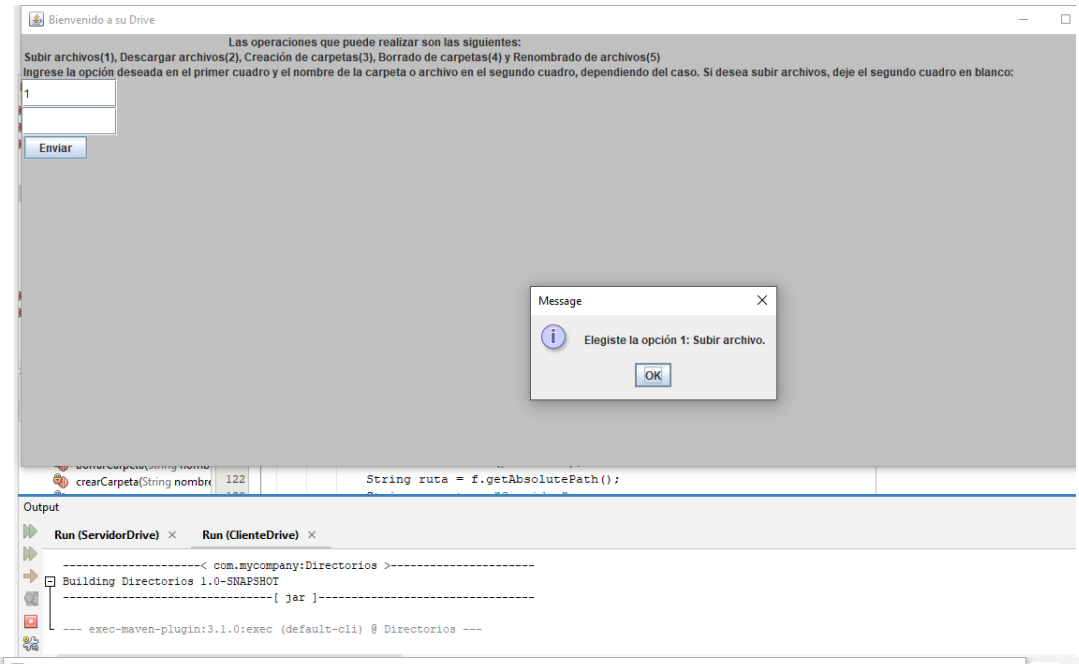
Una vez que inicia nuestro servidor, empieza a esperar los datagramas:

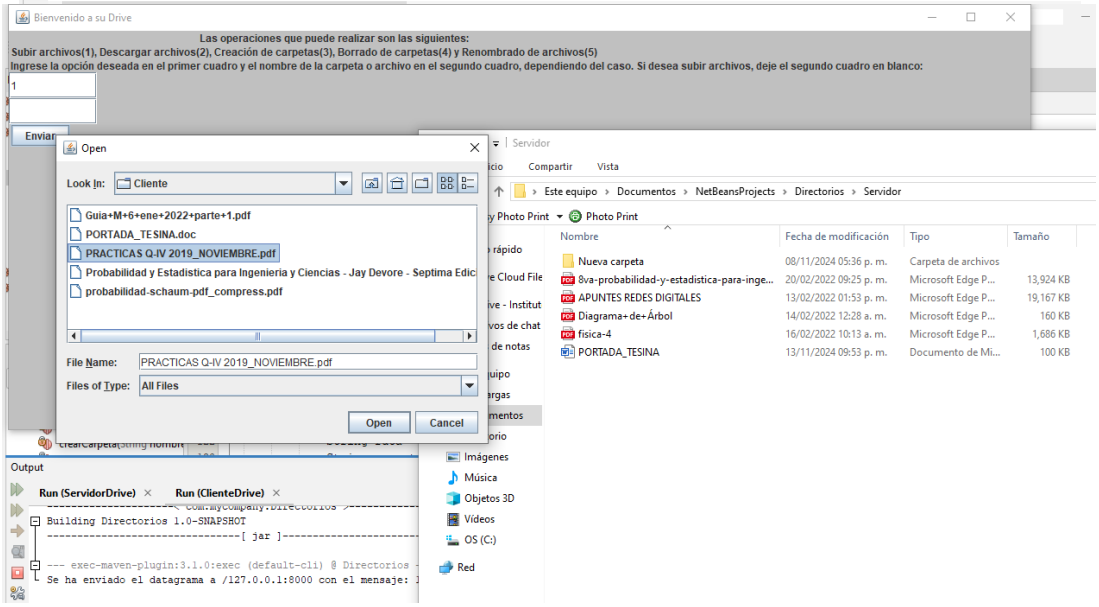


Cuando el cliente se ejecuta, despliega una interfaz para que el usuario pueda realizar la operación correspondiente:

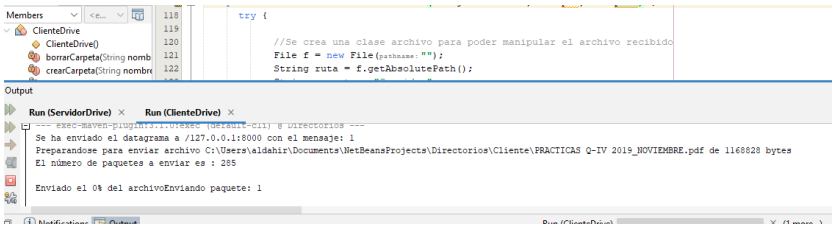


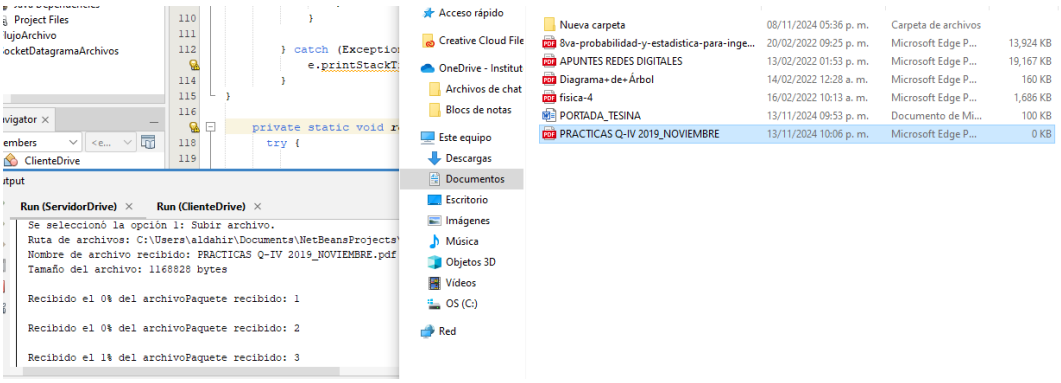
Subir archivos:
Para poder subir un archivo, se coloca la opción 1 en el primer recuadro, luego nos aparecen los archivos que se pueden subir, siempre y cuando se encuentren en la carpeta cliente.





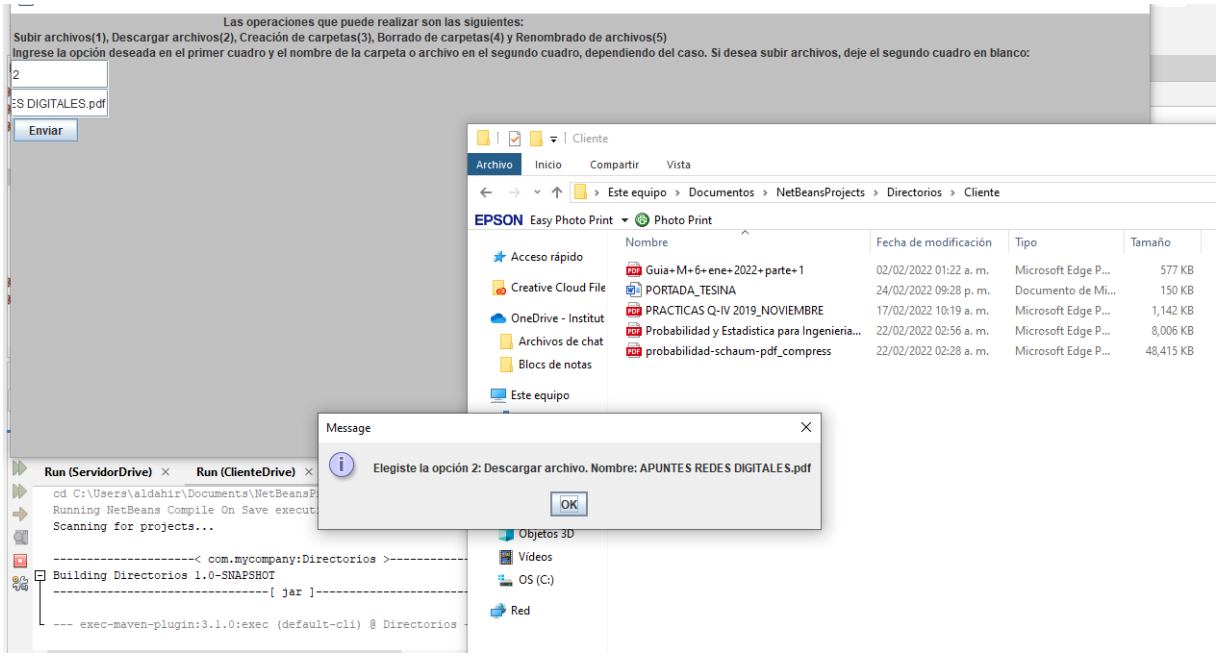
Una vez que se selecciona, se fragmenta el archivo en datagramas y se empiezan a mandar al servidor:



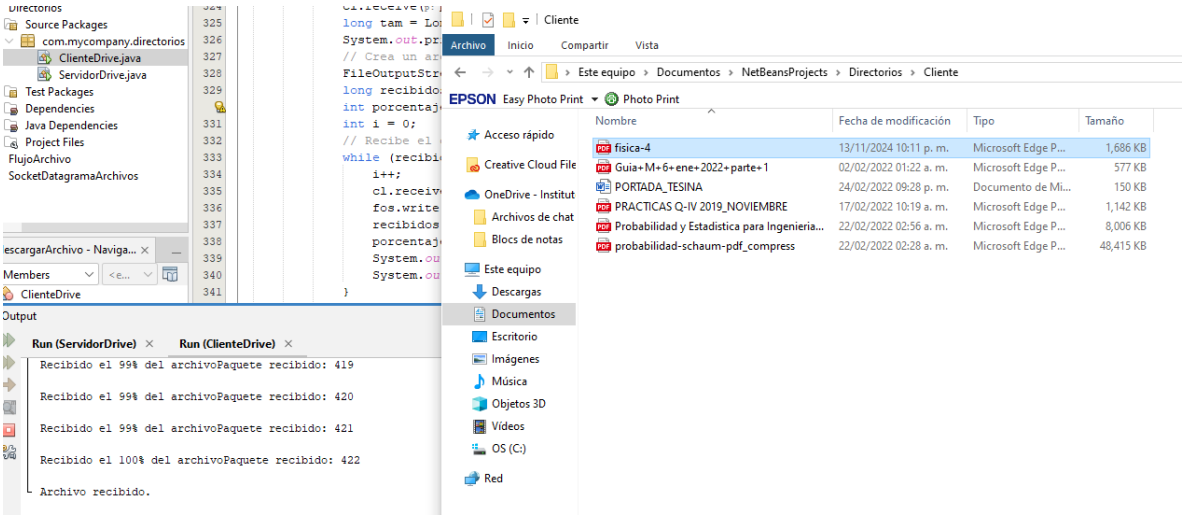


Descargar archivo

Para descargar archivos se coloca la opción 2 en el primer recuadro y el nombre del archivo a descargar en el segundo. Se debe agregar la extensión del archivo a descargar.

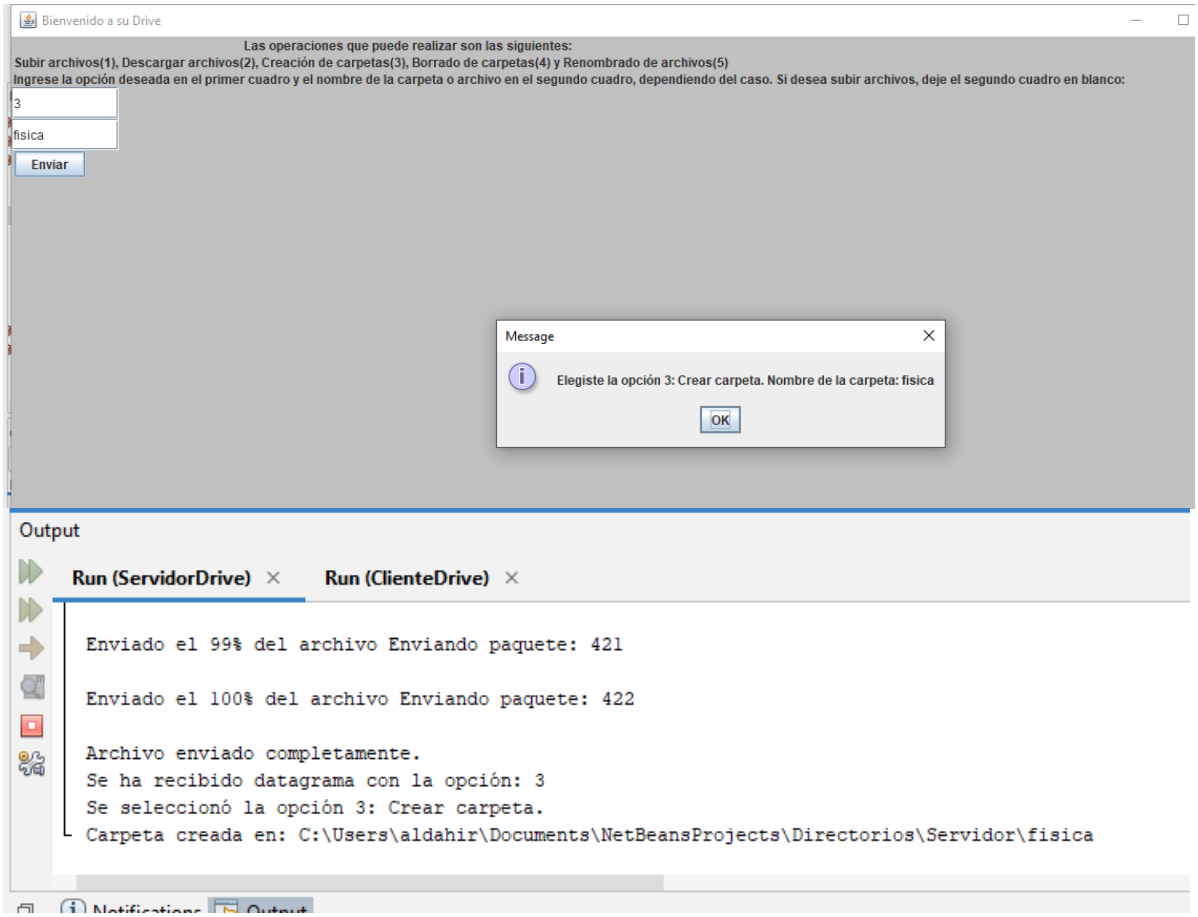


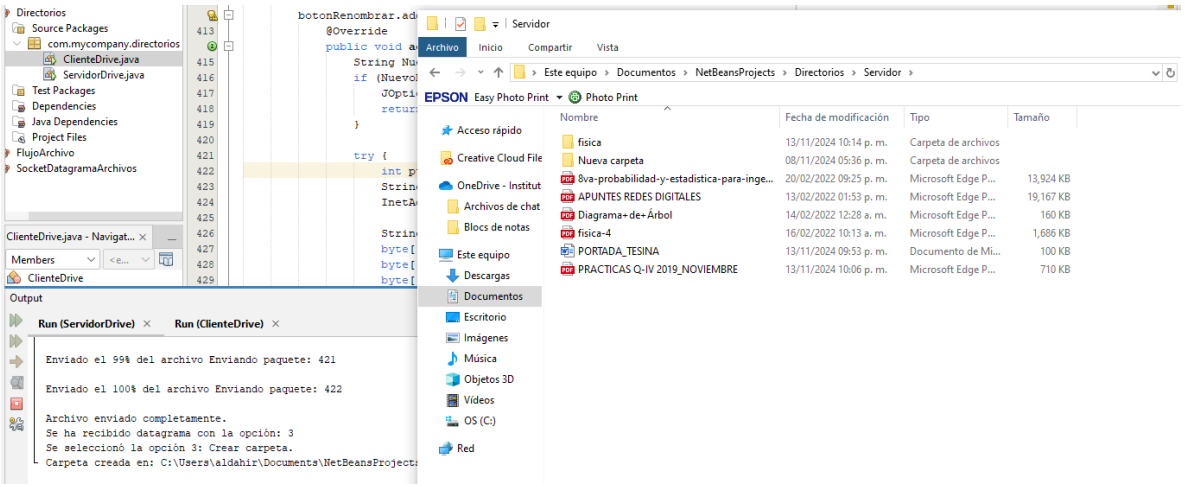
Luego de esto se empieza a fragmentar el archivo en datagramas y se empieza a enviar del servidor al cliente.



Crear carpeta:

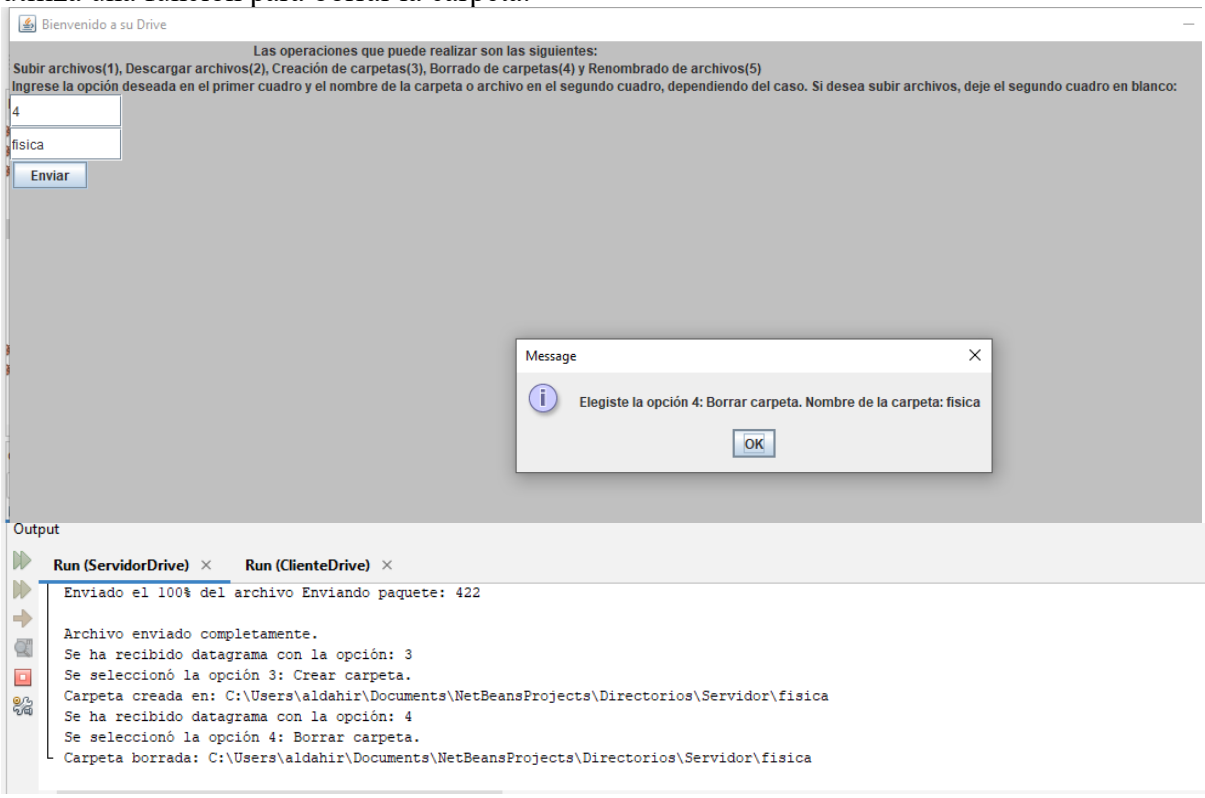
Para crear una carpeta, se coloca la opción 3 en el primer recuadro y en el segundo el nombre de la carpeta a crear, la cual se creará en la carpeta servidor.

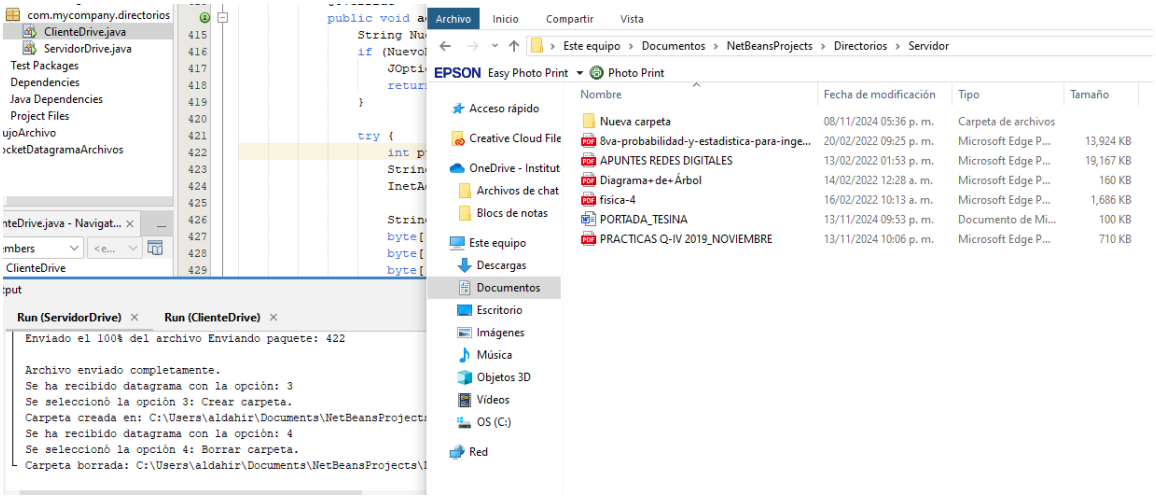




Borrar carpeta:

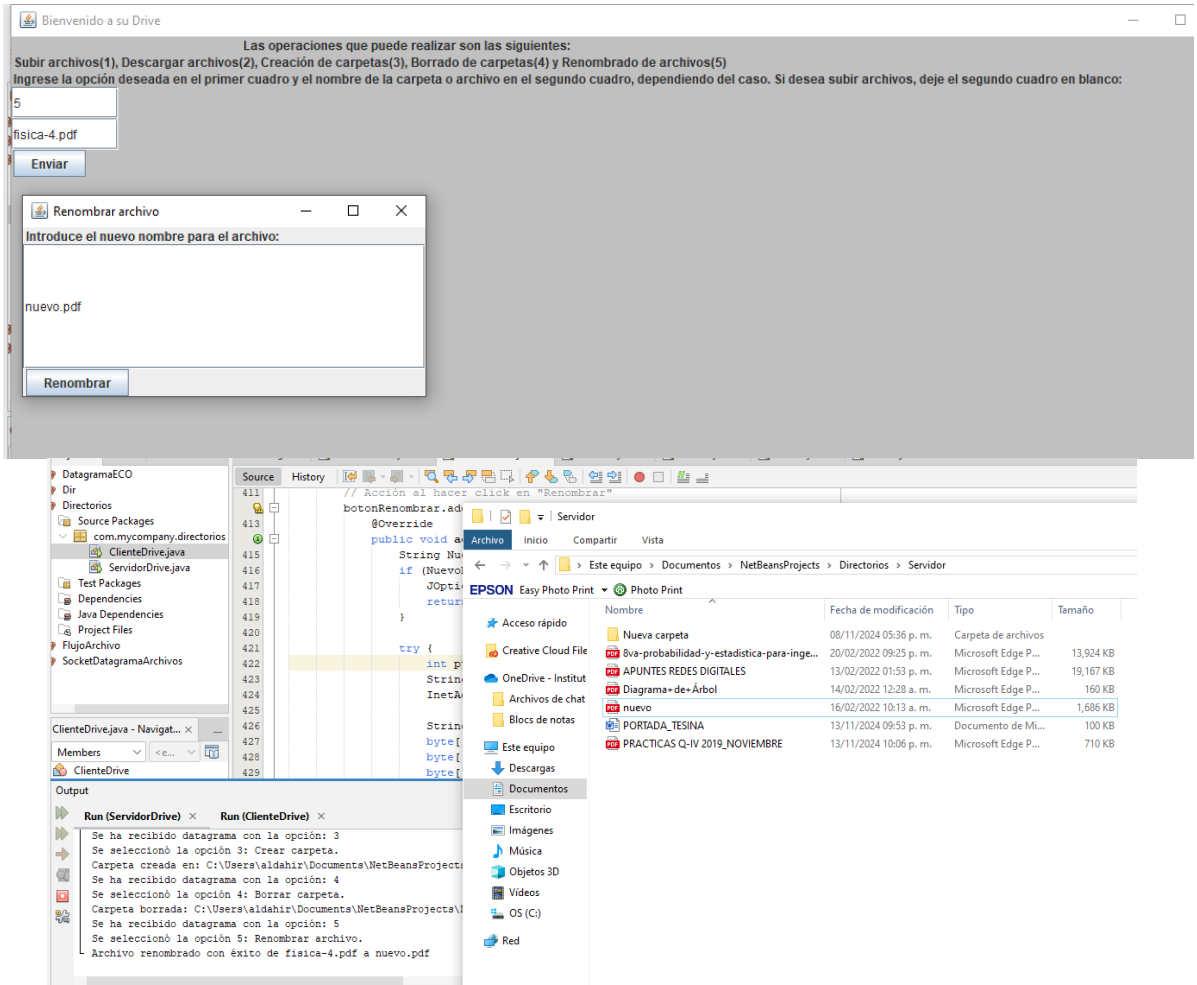
Para borrar una carpeta se coloca el número 4 en el primer recuadro y el nombre de la carpeta a borrar en el segundo, una vez hecho esto, se envía el nombre por medio de datagrama y se utiliza una función para borrar la carpeta.





Renombrar archivos:

Para poder renombrar un archivo, se coloca el número 5 en el primer cuadro, en el segundo se coloca el nombre del archivo a renombrar, seguido de un punto y su extensión. Luego de esto aparece un cuadro en donde se debe ingresar el nuevo nombre del archivo.



Conclusión

Los sockets de datagrama ofrecen una forma rápida y eficiente de comunicación en red para aplicaciones que requieren baja latencia y pueden tolerar la pérdida de algunos paquetes. Al no estar orientados a conexión, permiten que los desarrolladores diseñen aplicaciones ligeras y de alto rendimiento, aunque en caso de requerir confiabilidad, es necesario implementar un manejo adicional a nivel de aplicación. Esto hace que los sockets de datagrama sean una elección versátil para una amplia variedad de aplicaciones de red, desde streaming en tiempo real hasta videojuegos en línea y consultas de DNS, destacándose especialmente en entornos donde la rapidez y el bajo uso de recursos son prioritarios.