Ana María Espinosa Chaparro 201425031

Fabio Andres Lopez Corredor 201423782

Para desarrollar la solución de los requerimientos se realizaron las siguientes tareas:

**Etapa 1 : Implementación del cliente TCP.**

1. Desarrollar interfaz: El usuario accedera a una interfaz con botones que le permitan especificar el host y puerto por los que quiere comunicarse, le permitirá elegir entre tres archivos de 5, 20 y 50MB cual quiere recibir, detener el envío de los archivos o detener la comunicación con el servidor, y en un panel de texto visualizar el estado de su conexión.
2. Crear conexión: Por medio de un Socket en el que se especifica un host y un puerto, permitir al cliente conectarse al servidor. El sistema mostrará que la conexión fue exitosa o fallida.
3. Seleccionar archivo: Un JComboBox le mostrará al usuario los tres archivos y este podrá seleccionar el archivo que desea recibir.
4. Detener envío: Con un boton “Detener” el usuario podrá parar el envío del archivo.
5. Visualizar paquetes recibidor: En consola se le mostrará al cliente los paquetes recibidos, esto se lograra leyendo el flujo de entrada de bytes.
6. Recibir un archivo: El usuario recibirá el archivo que seleccionó y este se colocara en una carpeta que se le especifica al cliente, y además en un panel de texto se le informa al usuario la llegada del archivo.
7. Repositorio de los archivos recibidos: En la carpeta “recibidos” se almacenarán todos los archivos recibidos por parte del servidor.

Todo esto se realiza siguiendo el protocolo TCP.

**Etapa 2: Implementación del servidor TCP.**

1. Recibir conexiones TCP: Se crea un Server Socket que recibe conexiones TCP, a través de un puerto elegido(6789) para este propósito.
2. Contar 3 tipos de archivos disponibles para su descarga: El la carpeta docs se encuentran 3 archivos para mandar, uno de 5MB, otro de 20 MB, y otro de 50MB. El servidor cuenta con unas constantes que especifican la ruta de cada uno de estos archivos.
3. Definir el tamaño del buffer: Por medio de setReceiveBufferSize() se establece el tamaño del buffer, y posteriormente con las pruebas realizadas se llega a tomar la decisión que para esta aplicación la mejor opción es 50.
4. Definir el tamaño de los mensajes: Inicialmente se configura con un número de 1024 bytes, después se realizarán pruebas para definir si este valor es el indicado.
5. Definir un timeOut: Por medio de setSoTimeOut() se define un tiempo límite en el que el servidor cerrará la conexión para evitar el consumo de recursos innecesarios.
6. Realizar la transferencia de archivos: Según la elección del cliente se realiza el envío del archivo solicitado. Esto se informa en la interfaz del cliente y por consola.

Todo esto se realiza siguiendo el protocolo TCP.

**Etapa 3: Verificar el correcto funcionamiento.**

En esta etapa se realiza pruebas sobre el funcionamiento de los requerimientos, y que la interfaz sea amigable para el cliente. En caso de encontrar algún defecto se corregirá antes de proseguir a la siguiente etapa.

**Etapa 4: Pruebas de carga y desempeño sobre la arquitectura.**

1. Definición de los parámetros a utilizar en las pruebas.
2. Creación de pruebas de carga y desempeño.

Las pruebas se realizan con la ayuda de la herramienta JMeter.

**Etapa 5: Análisis de resultados.**

1. Elaboración del informe, respondiendo las preguntas solicitadas.