

Enunciado práctica 1

Desarrollo de Aplicaciones Distribuidas I

Miguel Ángel Guillén Navarro

Grado en Ingeniería Informática



Índice de contenidos

1. Introducción.....	3
2. Sobre el protocolo FTP	4
2.1 Tipos de Transferencia en FTP	4
2.2 Autenticación en FTP	5
2.3 FTP Seguro	5
2.4 Uso de FTP.....	6
3. Aspectos para tener en cuenta	7
4. Funcionalidad exigida.....	8
4.1 Entrega parcial.....	8
4.2 Entrega final enero	8
4.3 Entrega julio.....	8
4.4 Convocatoria especial	8
5. Entregas que realizar	9
5.1 Entrega parcial.....	9
5.2 Entrega final.....	9
6. Criterios de evaluación y calificación	10
6.1 Autoinforme	10

Desarrollo de Aplicaciones distribuidas

1. Introducción

El objetivo de la práctica es desarrollar los conceptos teóricos del desarrollo de aplicaciones distribuidas implementado, mediante sockets, un cliente FTP.

Elementos de la práctica

El equipo de directivo de la empresa para la que trabajamos ha contactado con el gerente de una empresa de desarrollo software y ha llegado a un acuerdo para que seamos nosotros los que implementemos la aplicación que comercializa. Entre otros muchos servicios la empresa oferta la posibilidad de crear una aplicación que permita gestionar mediante el protocolo FTP las carpeta y archivos alojados en el servidor.

La aplicación permitirá a los usuarios interactuar con uno o varios servidores FTP de una manera eficiente y segura. Este cliente debe ser ligero, seguro y altamente personalizable para futuros clientes.

La primera fase de este proyecto requiere la construcción de un prototipo que demuestre la capacidad de conectar, autenticar y comunicarse con un servidor FTP, utilizando Java y su enfoque de sockets para la comunicación en red. No se permite el uso de librerías externas.

2. Sobre el protocolo FTP

El Protocolo de Transferencia de Archivos (FTP, por sus siglas en inglés) es uno de los protocolos más antiguos y esenciales para la transferencia de archivos a través de una red. Su creación se remonta a los años 70 y está especificado en el RFC 959, diseñado para permitir la transferencia de datos entre sistemas, independientemente de la plataforma o del sistema operativo en uso. A lo largo de los años, FTP ha evolucionado, pero su propósito básico sigue siendo el mismo: permitir el intercambio eficiente de archivos en red, ya sea entre dos dispositivos o entre un servidor y un cliente.

FTP opera bajo un modelo cliente-servidor. Esto significa que hay dos partes involucradas: un cliente (la máquina que solicita o envía los archivos) y un servidor (la máquina que aloja o recibe los archivos). En esencia, el cliente se conecta al servidor FTP para descargar o subir archivos.

El protocolo FTP utiliza dos puertos distintos para comunicarse. Por lo general, el puerto 21 se usa para establecer la conexión y el puerto 20 se utiliza para la transferencia real de los datos. Este modelo de comunicación dual, que separa el control de la transferencia de los datos, es una de las características distintivas del FTP.

2.1 Tipos de Transferencia en FTP

El FTP permite dos modos de transferencia: **modo activo** y **modo pasivo**.

1. **Modo Activo:** En este modo, el cliente abre un puerto en su máquina y le informa al servidor que lo utilice para la transferencia de datos. El servidor FTP, a su vez, inicia una conexión desde su puerto 20 al puerto del cliente, transfiriendo los archivos en consecuencia. Sin embargo, este modo puede presentar problemas en situaciones donde hay cortafuegos o sistemas NAT

Desarrollo de Aplicaciones distribuidas

(Traducción de Direcciones de Red) que bloquean conexiones entrantes al cliente.

2. **Modo Pasivo:** Para solucionar los problemas del modo activo, se introdujo el modo pasivo, que es más seguro en redes protegidas. En este modo, el servidor abre un puerto y el cliente se conecta a ese puerto para la transferencia de datos. En este escenario, es el cliente quien establece las conexiones, evitando problemas con los cortafuegos y NAT.

2.2 Autenticación en FTP

El FTP puede funcionar en dos modalidades: con autenticación o sin autenticación. En la modalidad sin autenticación, denominada FTP anónimo, cualquier usuario puede conectarse al servidor sin necesidad de credenciales, lo que puede ser útil para distribuir archivos públicos, como actualizaciones de software o material didáctico. En cambio, en la modalidad con autenticación, se requieren un nombre de usuario y una contraseña para acceder a los archivos, lo que garantiza una mayor seguridad y control de acceso.

2.3 FTP Seguro

El protocolo FTP original no incluye mecanismos de cifrado, lo que significa que las transferencias de archivos, las credenciales y otros datos sensibles pueden viajar sin protección a través de la red, siendo susceptibles de ser interceptados por terceros. Para mitigar este riesgo, se introdujo una variante denominada **FTP Seguro (FTPS)**, que utiliza el protocolo SSL/TLS para cifrar tanto las credenciales como los datos transferidos. Otra alternativa es **SFTP (Secure File Transfer Protocol)**, que, aunque su nombre sugiere una relación con FTP, en realidad está más relacionado con el protocolo SSH. SFTP ofrece un nivel de seguridad elevado al integrar cifrado y autenticación robustos en la misma conexión.

2.4 Uso de FTP

FTP sigue siendo ampliamente utilizado en varios contextos, especialmente en la administración de servidores web, donde los administradores necesitan subir y descargar archivos al servidor de alojamiento. Sin embargo, el uso de FTP no se limita solo a servidores web. También es una herramienta importante en la gestión de archivos en grandes organizaciones, como universidades o empresas que requieren mover grandes cantidades de datos entre servidores y estaciones de trabajo.

La simplicidad del protocolo FTP lo hace especialmente útil cuando se trata de transferir grandes volúmenes de datos, ya que es eficiente en términos de uso de recursos. Además, muchas herramientas y clientes de FTP están disponibles tanto para sistemas operativos basados en UNIX como para Windows, lo que garantiza su versatilidad en distintos entornos.

Entre los clientes FTP más conocidos se encuentran **FileZilla**, **WinSCP** y **Cyberduck**, los cuales ofrecen interfaces gráficas fáciles de usar. También existen opciones de línea de comandos, como el comando `ftp` en sistemas UNIX y Windows, que proporcionan una mayor flexibilidad para usuarios avanzados.

Aunque el protocolo FTP ha sido desplazado en muchos casos por tecnologías más modernas y seguras, sigue siendo una herramienta valiosa para la transferencia de archivos en red. Su simplicidad y eficiencia en la gestión de grandes volúmenes de datos lo mantienen vigente, sobre todo cuando se combina con capas adicionales de seguridad como FTPS o SFTP. La elección entre FTP y otros protocolos dependerá del contexto específico y de los requisitos de seguridad de cada organización. Sin embargo, para muchas tareas cotidianas, FTP sigue siendo una solución rápida y confiable para el intercambio de archivos.

3. Aspectos para tener en cuenta

A la hora de realizar la práctica se deben tener en cuenta los siguientes aspectos de cara a una correcta implementación de esta.

- **Autenticación:** Se debe implementar un método que solicite al usuario su nombre y contraseña al inicio de la sesión. Este usuario/contraseña es para acceder a la aplicación desarrollada por el estudiante. No tiene nada que ver con los usuarios utilizados para conectarse con el servidor de FTP.
- **Roles:** La aplicación diferenciará entre dos tipos de usuarios: administradores y clientes. Sólo los primeros podrán crear, editar y eliminar usuarios de aplicación. Existirá el usuario administrador por defecto cuyas credenciales son admin/admin.
- **Configuración:** Todos los usuarios podrán configurar una o más cuentas FTP para conectarse al servidor correspondiente. Es decir, una configuración de cuenta tendrá, al menos, la dirección IP del servidor FTP, el puerto donde está desplegado y el usuario/contraseña para conectarse.
- **Persistencia:** Toda la configuración de conexión antes mencionada que introduzca un usuario será almacenada en fichero para que cada vez que acceda a la aplicación se cargue y no tenga que volver a introducirla.
- **Librerías:** No está permitido el uso de librerías que implementen un cliente FTP. Se debe realizar toda la funcionalidad mediante objetos de la clase Sockets y los correspondientes buffers de lectura y escritura.
- **Código:** Se debe crear una estructura de clases que permita la reutilización del código que accede al servidor de FTP. Es decir, la jerarquía de clases debe seguir el modelo en 3 capas (interfaz, lógica y persistencia).

Desarrollo de Aplicaciones distribuidas

4. Funcionalidad exigida

4.1 Entrega parcial

1. Control de acceso a la aplicación. Sólo los usuarios administradores pueden crear usuarios de la aplicación.
2. Cada usuario podrá configurar su servidor de FTP.
3. Se realiza la conexión con el servidor FTP y se listan los ficheros del directorio raíz.

4.2 Entrega final enero

1. La misma que en la entrega anterior y, además:
 - a. Descargar / Subir ficheros.
 - b. Crear/Eliminar carpetas.
 - c. Renombrar ficheros/carpetas.

4.3 Entrega julio

2. La misma que en la convocatoria anterior y, además:
 - a. Mover ficheros entre carpetas.
 - b. Cancelar una descarga/subida en curso.

4.4 Convocatoria especial

Dicha convocatoria se llevará a cabo en el curso 25-26.

1. La misma que en la convocatoria anterior y, además:
 - a. Se hace copia de seguridad de la estructura de ficheros/carpetas de tal forma que se puede trabajar offline.

Desarrollo de Aplicaciones distribuidas

5. Entregas que realizar

5.1 Entrega parcial

¿Qué? → Un zip nombrado con DNI y apellidos que incluya el código fuente y el diagrama de clases implementadas.

¿Cuándo? → Fecha tarea campus.

¿Dónde? → Campus virtual, tarea “Enero Práctica 1 Sockets – Entrega parcial”.

5.2 Entrega final

¿Qué? → Un zip nombrado con el DNI y apellidos que incluya: el código fuente, el autoinforme y un documento con la memoria.

¿Cuándo? → Fecha tarea campus.

¿Dónde? → Campus virtual, tarea “Enero Práctica 1 Sockets”.

En julio será “Julio Práctica 1 Sockets” y en convocatoria especial “Convocatoria Especial Práctica 1 Sockets”.

Desarrollo de Aplicaciones distribuidas

6. Criterios de evaluación y calificación

A la hora de evaluar la práctica se tendrán en cuenta varios criterios que determinan el grado en la calificación; estos son:

- ✓ La estructura de clases es correcta y cumple con los principios de la Programación Orientada a Objetos.
- ✓ Existe una explicación fundamentada para la estructura de clases diseñada.
- ✓ El código está optimizado.
- ✓ El código está comentado.
- ✓ Existe un correcto control de errores tanto a nivel de excepción en la ejecución como de interfaz de usuario.
- ✓ Se sigue el modelo en dos capas propuesto en esta memoria.

Antes de calificar se comprueba que:

- ✓ La memoria esté bien redactada y estructurada.
- ✓ La funcionalidad mínima haya sido implementada.

La puntuación que se puede obtener en cada entrega es 1 y 10 respectivamente. Siendo la nota final de la asignatura la suma de las dos. Por tanto, la primera entrega sirve como punto extra.

6.1 Autoinforme

En el autoinforme, sólo necesario para la segunda entrega, se especifica la funcionalidad a implementar, así como los puntos que se pueden obtener como máximo. En la tabla anterior se especifican los criterios por los que se evalúa la

Desarrollo de Aplicaciones distribuidas

funcionalidad exigida y siempre teniendo en cuenta el máximo posible a partir del autoinforme.

Por ejemplo, si en el autoinforme se incluye funcionalidad para optar a un 8 como máximo los porcentajes se aplican sobre esa nota como tope. Por tanto, si sacas un 10 en cada concepto (Memoria, Estructura de clases, etc.) tendrás un 8.

Además de los puntos básicos que sirven para establecer la máxima puntuación existen una serie de puntos extra. Estos se sumarán al resultado de aplicar el porcentaje. La suma nunca superará el máximo establecido. En el ejemplo anterior, aunque se hubiera implementado una funcionalidad extra no se pasaría del 8. En cambio, si al aplicar los porcentajes obtienes un 6 pero has hecho funcionalidad extra para sumar un punto la nota final será un 7.