

## FTP

El **Protocolo de Transferencia de Archivos** (FTP) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado.

La librería **Apache Commons Net** es a encargada de proporcionar las clases y utilidades necesarias para realizar operaciones sobre un servidor FTP o FTPS desde un cliente Java.

Las principales clases de **Apache Commons Net** son:

<b>FTP</b>	Implementa un cliente ftp
<b>FTPClient</b>	Funcionalidad para la descarga de ficheros
<b>FTPSCient</b>	Permite usar ftp mediante el protocolo seguro SSL
<b>FTPFile</b>	Información de los ficheros almacenados en el servidor
<b>FTPReplay</b>	Códigos de retorno del servidor ftp

FTPClient ofrece una serie de métodos de comunicación con el server. Los más usados son:

<b>connect</b>	Conecta con el servidor
<b>changeToParentDirectory</b>	Cambia del directorio de trabajo al directorio padre actual
<b>changeWorkingDirectory</b>	Cambia al directorio de trabajo
<b>deleteFile</b>	Borra un fichero
<b>disconnect</b>	Desconecta del servidor
<b>getReplyCode</b>	Obtiene un código de respuesta ftp
<b>listDirectories</b>	Obtiene los directorios del directorio de trabajo
<b>listFiles</b>	Obtiene los ficheros del directorio de trabajo
<b>login</b>	Accede al servidor mediante usuario / password
<b>logout</b>	Desconecta del servidor
<b>makeDirectory</b>	Crea un directorio en el servidor
<b>rename</b>	Renombra un fichero del servidor
<b>retrieveFile</b>	Descarga un fichero
<b>setFileType</b>	Permite indicar el tipo de fichero que se va a descargar
<b>storeFile</b>	Sube un fichero

## Cliente SMTP en Java

Para crear un cliente FTP en Java usaremos la librería **commons-net-3.8.0.jar**.

Para programar un método de envío de correos en Java:

- **Primero** es necesario establecer una conexión con el server. Es importante notar que, aunque se permita un acceso anónimo, se precisa suministrar un usuario y un password. Los valores por defecto para un usuario anónimo son: "anonymous" / "anonymous@domain.com"

```
private void conectar() throws SocketException, IOException {
    ftpClient.connect(SERVER, PORT);
    int respuesta = ftpClient.getReplyCode();

    // Trying to connect
    if (!FTPReply.isPositiveCompletion(respuesta)) {
        ftpClient.disconnect();
        throw new IOException("Error al conectar con " + SERVER + ":" + PORT);
    }

    boolean credencialesOk = ftpClient.login(USER, PASS);
    if (!credencialesOk) {
        throw new IOException("User or Pass incorrectos");
    }

    ftpClient.setFileType(FTP.BINARY_FILE_TYPE);
}
```

- **Lo segundo**, a partir de ahora vamos lanzando operaciones sobre el server. El directorio de trabajo será la carpeta actual del servidor sobre la que estamos trabajando. Podremos movernos por la estructura de fichero mediante otras operaciones. Será necesario usar listFiles() para que el cliente se informe de los ficheros que puede descargar. Cuando desee descargar un fichero, se lo ordena:

```
private boolean downloadFile(String fichero, String path) throws IOException {
    boolean ret = false;
    OutputStream outputStream = new BufferedOutputStream(new FileOutputStream(path));
    ret = ftpClient.retrieveFile(fichero, outputStream);
    outputStream.close();
    return ret;
}
```

- **Finalmente**, cierra las conexiones.