

PROGRAMACIÓN DE SERVICIOS Y PROCESOS**UT04. Generación de Servicios en Red***Práctica a entregar*

Práctica 1. Se cuenta con la siguiente librería **clienteFTP.java**:

```
import java.io.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import org.apache.commons.net.ftp.*;

/**
 * Clase que gestiona la conexión a un servidor y la de sus ficheros.
 *
 * @author Jonay Garcia Gonzalez.
 */
public class ClienteFTP {

    // Atributos
    private static FTPClient cliente = new FTPClient();

    /**
     * Constructor
     */
    public ClienteFTP() {

    }

    /**
     * Establece una conexión con el servidor FTP
     *
     * @param server
     *         Servidor al que nos queremos conectar
     * @param user
     *         Usuario para poder acceder
     * @param pwd
     *         Contraseña para poder acceder
     * @return True, si la conexión se estableció correctamente,
     *         False en caso contrario
     */
    public boolean conectar(String server, String user, String pwd) {

        try {
            // Conectarse y loguearse al servidor FTP
            cliente.connect(server);
            if (cliente.login(user, pwd)) {
                // Entrando en modo pasivo
                cliente.enterLocalPassiveMode();
                // Activar recibir/enviar cualquier tipo de archivo
                cliente.setFileType(FTP.BINARY_FILE_TYPE);

                // Obtener respuesta del servidor y acceder
                int respuesta = cliente.getReplyCode();
                if (FTPReply.isPositiveCompletion(respuesta) == true) {
```

```

        System.out.println("Conexión establecida con el servidor: " + server);
        return true;
    } else {
        return false;
    }
} else {
    System.out.println("Usuario o contraseña incorrectos.");
    return false;
}

} catch (IOException e) {
    System.out.println("Host del servidor incorrecto: " + server);
    e.printStackTrace();
    return false;
}

}

/**
 * Lista los ficheros del directorio actual del servidor FTP
 */
public void listarFicheros() {

    // lists files and directories in the current working directory
    FTPFile[] files;
    try {
        files = cliente.listFiles();

        // iterates over the files and prints details for each
        DateFormat dateFormatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

        for (FTPFile file : files) {
            String details = file.getName();
            if (file.isDirectory()) {
                details = "[" + details + "]";
            }
            details += "\t\t" + file.getSize();
            details += "\t\t" + dateFormatter.format(file.getTimestamp().getTime());
            System.out.println(details);
        }

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

/**
 * Sube un fichero al servidor FTP a un directorio especificado por parametro
 *
 * @param rutaCompleta:
 *     Ruta absoluta del fichero a subir, incluido el nombre del fichero,
 *     por ejemplo, "D:\\\\dir1\\file1.pdf"
 * @param nombreFichero:
 *     Nombre del fichero, incluida su extension, con el que se guardara
 *     en el servidor FTP
 * @return directorioFTP: Directoiro en el servidor FTP donde se guardara el
 *     fichero
 */
public void subirFichero(String rutaCompleta, String nombreFichero, String directorioFTP) {

    InputStream input;
    try {

```

```

        input = new FileInputStream(new File(rutaCompleta));
        cliente.storeFile(directorioFTP + nombreFichero, input);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

/**
 * Sube un fichero en el directorio actual del servidor FTP
 *
 * @param rutaCompleta:
 *     Ruta absoluta del fichero a subir, incluido el nombre del fichero,
 *     por ejemplo, "D:\\\\dir1\\file1.pdf"
 * @param nombreFichero:
 *     Nombre del fichero, incluida su extension, con el que se guardara
 *     en el servidor FTP
 */
public void subirFichero(String rutaCompleta, String nombreFichero) {

    InputStream input;
    try {
        input = new FileInputStream(new File(rutaCompleta));
        cliente.storeFile(nombreFichero, input);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

/**
 * Imprime el directorio actual del servidor FTP
 */
public void directorioActual() {

    try {
        String directorio = cliente.printWorkingDirectory();
        System.out.println("Directorio actual:" + directorio);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

/**
 * Nos posicionamos en el directorio raiz del servidor FTP
 */
public void cambiarADirectorioPadre() {

    try {
        cliente.changeToParentDirectory();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

/**
 * Nos posicionamos en un directorio en el servidor FTP especificado por
 * parametro, esta ruta es a partir del directorio actual

```

```
*
* @param directorio:
*     Directorio en el servidor FTP en el que nos queremos posicionar
*     a partir del directorio actual
*/
public void cambiarDirectorio(String directorio) {

    try {
        String directorioActual = cliente.printWorkingDirectory();
        cliente.changeWorkingDirectory(directorioActual + "/" + directorio);
        System.out.println("Cambiano al directorio: " + directorio);
    } catch (IOException e) {
        e.printStackTrace();
    }

}

/**
 * Borra un fichero en el directorio actual del servidor FTP
 */
* @param fichero:
*     Fichero a borrar del directorio actual en el servidor FTP
*/
public void borrarFichero(String fichero) {

}

/**
 * Borra un fichero del servidor FTP del directorio especificado por parametro
 */
* @param directorio:
*     Directorio en el servidor FTP donde se encuentra el fichero a borrar
* @param fichero:
*     Fichero a borrar en el servidor FTP
*/
public void borrarFichero(String directorio, String fichero) {

}

/**
 * Cerramos la sesion cno el servidor FTP
 */
public void desconectar() {

    if (cliente.isConnected()) {
        try {
            cliente.logout();
            cliente.disconnect();
            System.out.println("Desconectado del servidor FTP.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}

}
```

Que cuenta con los siguientes métodos.

- **conectar()**: permite conectarnos al servidor FTP.
- **listarFicheros()**: lista todos los ficheros del directorio actual del FTP.
- **subirFichero()**: permite subir un fichero al servidor FTP.
- **directorioActual()**: imprime el directorio donde nos encontramos en el servidor FTP.
- **cambiarADirectorioPadre()**: nos posicionamos en el directorio raíz del servidor FTP.
- **cambiarDirectorio()**: nos posicionamos en un directorio dentro del servidor FTP.
- **desconectar()**: se desconecta del servidor FTP.

Se pide realizar lo siguiente:

- Se pide crear una aplicación que permita, mediante un menú interactivo o una interfaz gráfica, gestionar todos estos métodos para poder subir un fichero al servidor FTP.
- Además, habrá que crear dos métodos:
 - **borrarFichero(String fichero)**: método que permite borrar un fichero del servidor FTP en el directorio actual.
 - **borrarFichero(String directorio, String fichero)**: método que permite borrar un fichero del servidor FTP en un directorio especificado por parámetro.

Nota: El objetivo de esta práctica es subir un fichero al servidor FTP utilizando los métodos de la clase clienteFTP.java y además, implementar los métodos para poder borrar el fichero que hemos subido al servidor.

El código para conectar con un servidor que está en la ip 192.168.1.38 con el usuario/password *formacion/formacion* sería:

```
ClienteFTP c = new ClienteFTP();  
  
c.conectar("192.168.1.38", "formacion", "formacion");  
c.desconectar();
```