

UF3. [PAC02] SOLUCIÓN

Actividades

Parte práctica

1. Comunicación con un servidor FTP

Generar un programa Java que se conecte a <ftp://ftp.rediris.es> y visualice todos los directorios del directorio raíz. Comentar el código indicando que se está realizando en cada paso.

```
import java.io.IOException;
import org.apache.commons.net.ftp.*;

/**
 *
 * @author Ana
 */
public class ClienteFTP {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        FTPClient cliente = new FTPClient();
        String servFTP="ftp.rediris.es";
        String usuario="";
        String pwd="";

        try {
            //Nos conectamos a servidor de FTP indicado
            cliente.connect(servFTP);
            //Obtenemos el código de respuesta
            int cod_resp = cliente.getReplyCode();
            //Comprobamos el código de respuesta
            if(FTPReply.isPositiveCompletion(cod_resp)){
                //Si es un código de respuesta positivo, seguimos con el
                //Logueamos el cliente con el usuario y pwd indicados mas
                boolean login = cliente.login(usuario, pwd);
                //Y comprobamos si el login ha sido o no correcto
                if(login){
                    //Si ha sido correcto, seguimos con el ejercicio
                    String dir_act=cliente.printWorkingDirectory();
                    //Comprobamos si estamos en el directorio raíz
                    if(!dir_act.equals("/"))
                        //En caso que no, cambiamos al directorio raíz
                        cliente.changeWorkingDirectory("/");
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

        System.out.println("Directorio actual: " +
cliente.printWorkingDirectory());
        //Obtenemos el listado de ficheros y directorios del
cliente
        FTPFile[] files = cliente.listFiles();
        System.out.println("Ficheros en el directorio actual:");

        String tipos[] = {"Fichero", "Directorio", "Elace
simb."};
        //Listamos por pantalla los ficheros y directorios del
directorio raiz, indicando de que tipo son
        for(int i=0; i<files.length; i++){
            System.out.println("\t" + files[i].getName() + " => "
+ tipos[files[i].getType()]);
        }
        //Realizamos el logout
        boolean logout = cliente.logout();
        if(!logout)
            System.out.println("Error al hacer el logout...");

        //Desconectamos del servidor FTP
        cliente.disconnect();
    }
    else{
        //Si no ha sido correcto, desconectamos y finalizamos el
programa
        cliente.disconnect();
        System.out.println("Login incorrecto");
        System.exit(1);
    }
}
else{
    //Si es un código de respuesta negativo, desconectamos y
finalizamos el programa
    cliente.disconnect();
    System.out.println("Conexión rechazada: " + cod_resp);
    System.exit(0);
}
} catch(IOException ioe){
    ioe.printStackTrace();
}
}
}
}

```

2. Comunicación con un servidor SMTP

Generar un programa Java que utilice el servidor SMTP de tu cuenta de correo (o podéis crear otra para tal fin) para enviar un mensaje a alguno de tus contactos. Los datos para conectarse al servidor SMTP (nombre, servidor, usuario, clave y puerto) deberán introducirse desde la línea de comandos. Comentar el código indicando que se está realizando en cada paso.

```
import java.io.IOException;
import java.io.Writer;
import java.util.Scanner;
import java.security.InvalidKeyException;
import java.security.KeyStoreException;
import java.security.NoSuchAlgorithmException;
import java.security.UnrecoverableKeyException;
import java.security.spec.InvalidKeySpecException;
import javax.net.ssl.KeyManager;
import javax.net.ssl.KeyManagerFactory;
import org.apache.commons.net.smtp.*;
/**
 *
 * @author Ana
 */
public class ClienteSMTP {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws NoSuchAlgorithmException,
    UnrecoverableKeyException, KeyStoreException, InvalidKeyException,
    InvalidKeySpecException {
        //Declaramos los datos del usuario y del servidor
        String server, username, pwd;
        int puerto; //Valor habitual: 587
        Scanner in = new Scanner(System.in);

        System.out.println("Introduce el nombre del servidor:");
        server = in.nextLine();
        System.out.println("Introduce el puerto de comunicacion con el
servicio SMTP del servidor:");
        puerto = in.nextInt();
        System.out.println("Introduce tu usuario (en el servidor antes
indicado):");
        username = in.next();
        System.out.println("Introduce tu password (en el servidor antes
indicado):");
        pwd = in.next();

        //Creamos un cliente SMTP Seguro
        AuthenticatingSMTPClient client = new AuthenticatingSMTPClient();

        try{
            int respuesta;
            //Creación de la clave para establecer un canal seguro
            KeyManagerFactory kmf =
            KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
```

```

kmf.init(null,null);
KeyManager km = kmf.getKeyManagers()[0];

//Nos conectamos al servidor SMTP
client.connect(server,puerto);

//Se establece la clave para la comunicación segura
client.setKeyManager(km);

respuesta = client.getReplyCode();
if(!SMTPReply.isPositiveCompletion(respuesta))
{
    client.disconnect();
    System.err.println("Conexion Rechazada");
    System.exit(1);
}

//Se envía a el comando EHLO
client.ehlo(server);

//Se ejecuta el comando STARTTLS y se comprueba si es True
if(client.execTLS()){
    //Se ejecuta la autenticación con el servidor
    if
(client.auth(AuthenticatingSMTPClient.AUTH_METHOD.PLAIN,username,pwd)){
        String destino = "agonzalez@ilernaonline.com";
        String asunto = "Prueba de SMTPClient";
        String mensaje = "Hola!!!\n\nNos vemos por las
redes!\n\nUn saludo,\n\nAna";
        //Creamos la cabecera con los datos arriba rellenados
        SimpleSMTPHeader cabecera = new
SimpleSMTPHeader(username, destino, asunto);

        //Indicamos la dirección del remitente y del
destinatario del correo
        client.setSender(username);
        client.addRecipient(destino);

        //Y enviamos el correo
        Writer writer = client.sendMessageData();
        if(writer==null){
            System.out.println("Fallo al enviar los datos");
            System.exit(1);
        }
        //Enviamos la cabecera
        writer.write(cabecera.toString());
        //Y enviamos el mensaje
        writer.write(mensaje);
        writer.close();

        //Verificamos que el servidor haya recibido
correctamente los datos enviados
        boolean exito = client.completePendingCommand();
        if(!exito){
            System.out.println("Fallo al finalizar la
transaccion");

```

```
        System.exit(1);
    }
}
else
    System.out.println("Usuario no autenticado");
}
else
    System.out.println("Fallo al ejecutar STARTTLS");
}catch(IOException e){
    System.err.println("No se ha podido conectar con el servidor");
    e.printStackTrace();
    System.exit(1);
}
try{
    client.disconnect();
}catch(IOException f){
    f.printStackTrace();
}
System.exit(0);
}
}
```