

Tarea PSP06

Roy Denys Calderón

Actividad 1

-Solicita el nombre del usuario que va a utilizar la aplicación. El login tiene una longitud de 8 caracteres y está compuesto únicamente por letras minúsculas

Para este ejercicio realice una aplicacion cliente-servidor en el que el cliente se conecta al servidor a partir del puerto 2024, este servidor no es capaz de soportar concurrencia de clientes. Una vez un cliente se conecta este le pide un nombre de usuario con las características propuestas en el ejercicio

```
private boolean login(Socket cliente) {
    try {
        in = new DataInputStream(in: cliente.getInputStream());
        out = new DataOutputStream(out: cliente.getOutputStream());
        sc = new Scanner(source: System.in);
        Pattern pat = Pattern.compile(regex: "[a-z]{8}");
        Matcher mat = null;

        while (true) {
            out.writeUTF("Bienvenid@ usuari@ \n"
                + "Por favor ingrese un nombre de usuario \n"
                + "***SEGUIR INSTRUCCIONES** \n"
                + "-Longitud maxima de 8 caracteres \n"
                + "-Nombre completo en minisculas");
            nombre = in.readUTF();
            mat = pat.matcher(input: nombre);
            if (mat.find()) {
                return true;
            }
        }

    } catch (IOException ex) {
        Logger.getLogger(name: Servidor.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
    }
    return false;
}
```

-Solicita al usuario el nombre de un fichero que quiere mostrar. El nombre del fichero es como máximo de 8 caracteres y tiene una extensión de 3 caracteres.

El usuario una vez supera el login(escribe un nombre de usuario con las características que se la ha pedido) puede solicitar el archivo que desea que se imprima por pantalla, eso si con similares características a las del login, este metodo buscarFichero() busca el nombre del fichero a partir de la ruta proporcionada por el clietne y lo imprime en la terminal

```
private void buscaFichero(Socket cliente) {
    try {
        in = new DataInputStream(in: cliente.getInputStream());
        out = new DataOutputStream(out: cliente.getOutputStream());
        sc = new Scanner(source: System.in);
        Pattern pat = Pattern.compile(regex: "[a-z0-9]{8}.[a-z]{3}");
        Matcher mt;
        out.writeUTF("Bienvenido " + nombre + "\n" +
            "Introduce la ruta del archivo que quieras visualizar \n" +
            + "***Instrucciones** \n"
            + "-El nombre del archivo debe ser de 8 caracteres \n"
            + "-La extension del archivo debe ser de 3 caracteres");
        String ruta = in.readUTF();
        mt = pat.matcher(input: ruta);

        if(mt.find()){
            BufferedReader br = new BufferedReader(new FileReader(fileName: ruta));
            String lectura;
            String archivo = "";
            while ((lectura = br.readLine()) != null) {
                archivo += lectura + "\n";
            }
            out.writeUTF(str: archivo);
        }else{
            out.writeUTF(str: "La ruta no contiene los parametros establecidos");
        }
    }

    catch (IOException ex) {
        Logger.getLogger(name: Servidor.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

- **Visualiza en pantalla el contenido del fichero.**

Una vez el cliente se conecta a nuestro servidor supera el login e introduce una ruta el servidor le envia como respuesta al cliente el texto que se encuentra en el archivo especificado

```

C:\Desarrollo\K> Run [C:\Desarrollo\K>
--- resourceDirectory C:\Users\pablo\Documents\Desarrollo\K\resources & Tarea2 ---
skip not existing resourceDirectory C:\Users\pablo\Documents\Desarrollo\K\resources

--- compile C:\Desarrollo\K\resources & Tarea2 ---
Nothing to compile - all classes are up to date

--- generateClasses & Tarea2 ---
Generando clases
Por favor ingrese un nombre de usuario
> ingresa incorrectamente
> longitud maxima de 8 caracteres
> nombre completo en minisculas
> incorrecto
Reingrese el nombre
Ingresame la ruta del archivo que quieras visualizar
> "Introducciones"
El nombre del archivo debe ser de 3 caracteres
La extension del archivo debe ser de 3 caracteres
Un ejemplo correcto es: "Introducciones.txt"
Actividad 1.1. Crea una aplicación que realice los siguientes pasos:

Solicita el nombre del usuario que va a utilizar la aplicación. El login tiene una longitud de 8 caracteres y será, compuesto únicamente por letras minúsculas.
Solicita al usuario el nombre de un fichero que quiere mostrar. El nombre del fichero es como máximo de 8 caracteres y tiene una extensión de 3 caracteres.
Realiza un control al contenido del fichero.
Es importante tener en cuenta que al tiempo que realizar una validación de los datos de entrada y llevar un registro de la actividad del programa.

```

- Es importante tener en cuenta que se tiene que realizar una validación de los datos de entrada y llevar un registro de la actividad del programa.

Para realizar una validacion de los datos de entrada que recibe el servidor nos apoyamos de las clases Pattern y Matcher, con la clase Pattern podemos crear expresion regulares y la clase Matcher que nos permite validar que lo que el usuario intrdujo fue lo que pediamos.

Para llevar el registro de la actividad de nuestro servidor usamos la clase Logger, y creamos un fichero en el que guardamos los registros que se van produciendo

```
private void registrar() {
    Logger logger = Logger.getLogger(name: "Registros");
    FileHandler fh;

    try {
        fh = new FileHandler(pattern: "C:\\Users\\roica\\Desktop\\registros.log", append: true);

        logger.addHandler(handler: fh);
        logger.setLevel(newLevel: Level.ALL);
        SimpleFormatter sf= new SimpleFormatter();
        fh.setFormatter(newFormatter: sf);

    } catch (IOException ex) {
        Logger.getLogger(name: Servidor.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    } catch (SecurityException ex) {
        Logger.getLogger(name: Servidor.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

Actividad 2

Firmar digitalmente la aplicación.

En este caso vamos a firmar únicamente la clase servidor para este ejemplo, lo primero que tenemos que hacer es dirigirnos a la ubicación donde esta nuestra clase y convertirla en un jar

```
C:\Users\roica\Documents\NetBeansProjects\Tarea6\src\main\java\roy\tarea6>javac Servidor.java

C:\Users\roica\Documents\NetBeansProjects\Tarea6\src\main\java\roy\tarea6>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 66D9-8A19

Directorio de C:\Users\roica\Documents\NetBeansProjects\Tarea6\src\main\java\roy\tarea6
24/02/2024  20:34  <DIR>          .
24/02/2024  19:10  <DIR>          ..
24/02/2024  20:11              1.469 Cliente.java
24/02/2024  20:34              5.167 Servidor.class
24/02/2024  20:28              4.980 Servidor.java
                3 archivos              11.616 bytes
                2 dirs 1.715.464.990.720 bytes libres

C:\Users\roica\Documents\NetBeansProjects\Tarea6\src\main\java\roy\tarea6>jar cvf Servidor.jar Servidor.class
added manifest
adding: Servidor.class(in = 5167) (out= 2736)(deflated 47%)
```

Continuamos creando las claves tanto privada como publica y aplicando la configuración necesaria para este paso(Nombre, ciudad, Empresa...)

```

C:\Users\roica\Documents\NetBeansProjects\Tarea6\src\main\java\roy\tarea6>keytool -genkey -alias roy -keypass roy123 -keystore Repositorio -storepass roy123 -keyalg rsa
What is your first and last name?
[Unknown]: Roy Calderon
What is the name of your organizational unit?
[Unknown]: IES jose Palomo
What is the name of your organization?
[Unknown]: DAM
What is the name of your City or Locality?
[Unknown]: Murcia
What is the name of your State or Province?
[Unknown]: Murcia
What is the two-letter country code for this unit?
[Unknown]: SP
Is CN=Roy Calderon, OU=IES jose Palomo, O=DAM, L=Murcia, ST=Murcia, C=SP correct?
[no]: yes

Generating 2.048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 90 days
for: CN=Roy Calderon, OU=IES jose Palomo, O=DAM, L=Murcia, ST=Murcia, C=SP

```

Y como ultimo paso nos queda firmar nuestra clase Servidor, que en mi caso le llame ServidorFirmado

```

C:\Users\roica\Documents\NetBeansProjects\Tarea6\src\main\java\roy\tarea6>jarsigner -keystore Repositorio -signedjar ServidorFirmado.jar Servidor.jar roy
Enter Passphrase for keystore:
jar signed.

Warning:
The signer's certificate is self-signed.

```

De esta forma ya podríamos compartir con la llave publica nuestros proyectos