

ACTIVIDAD 6.1

En este apartado de la tarea se nos pide que realicemos una aplicación que cumpla los siguientes requerimientos:

- **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.**

Dentro del proyecto que hemos creado para la tarea, añadimos una clase principal a la que llamaremos “Aplicación Principal” y que contendrá la lógica del programa. Dentro del método main vamos a incluir un bloque de código para solicitar el login al usuario, y evaluar si este tiene el formato requerido según un patrón dado:

```
public static void main(String[] args) {
    //Declaración de variables
    boolean correcto;
    Scanner teclado = new Scanner(System.in);
    //Declaración de los patrones
    final String patronLogin = "[a-z]{8}";
    final String patronNomFichero = "^([a-zA-Z0-9]{1,8}).[a-zA-Z]{3}$"; //Patrón indicando cuál es el inicio el final de la cadena

    /*-----*/
    /*Establecemos un fichero de registros de la aplicación*/
    Logger logger = creaFicheroRegistro();
    /*-----*/
    //Solicitamos login al usuario
    do {
        correcto = false;
        try {
            System.out.println("Introduzca el nombre de usuario: ");
            String nomUsuario = teclado.next();

            if (comprobarPatron(patronLogin, nomUsuario)) { //Si patrón coincide
                correcto = true;
                // Añado un mensaje al log
                String textoLog = "El patron (" + patronLogin + ") del usuario " + nomUsuario + " es correcto";
                logger.log(Level.INFO, textoLog);
                Thread.sleep(200); //Pausa de aplicacion
            } else {
                // Añado un mensaje al log
                String textoLog = "El patron (" + patronLogin + ") del usuario " + nomUsuario + " es INCORRECTO";
                logger.log(Level.WARNING, textoLog);
                Thread.sleep(200); //Pausa de aplicacion
            }
        } catch (Exception e) { //Recogemos excepción si las hubiera
            System.out.println("Se ha producido la excepción " + e.getMessage());
            correcto = false;
        }
    } while (!correcto);
}
```

Solicitamos el login, lo recogemos en una variable y este se lo pasamos como parámetro al método creado “comprobarPatron” que nos devolverá true o false según coincida o no el login dado con el patrón. A su vez contemplamos también cualquier excepción metida dentro de un bucle para que el programa regrese a solicitar usuario en caso de que salte alguna excepción en la recepción de los datos y escribimos en el log la información cada vez que se realiza alguna acción importante.

```
public static boolean comprobarPatron(String patron, String texto) {
    Pattern pat;
    Matcher mat;
    //Compilamos y comparamos el patrón con el texto
    pat = Pattern.compile(patron);
    mat = pat.matcher(texto);

    return mat.find(); //boolean true si coinciden patrón y texto
}
```

Este es el método que nos compara patrón con texto recibido, en este caso el login

- 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.

Creamos este bloque a continuación del anterior:

```
//Solicitamos nombre del fichero al usuario
do {
    correcto = false;
    try {
        System.out.println("Introduzca el nombre del fichero: ");
        String nomFichero = teclado.next();

        if (comprobarPatron(patronNomFichero, nomFichero)) { //Si patrón coincide
            correcto = true;
            // Añado un mensaje al log
            String textoLog = "El patron (" + patronNomFichero + ") del fichero " + nomFichero + " es correcto";
            logger.log(Level.INFO, textoLog);
            Thread.sleep(200); //Pausa de aplicacion
            correcto = visualizarFichero(nomFichero, logger);
        } else {
            // Añado un mensaje al log
            String textoLog = "El patron (" + patronNomFichero + ") del fichero " + nomFichero + " es INcorrecto";
            logger.log(Level.WARNING, textoLog);
            Thread.sleep(200); //Pausa de aplicacion
        }

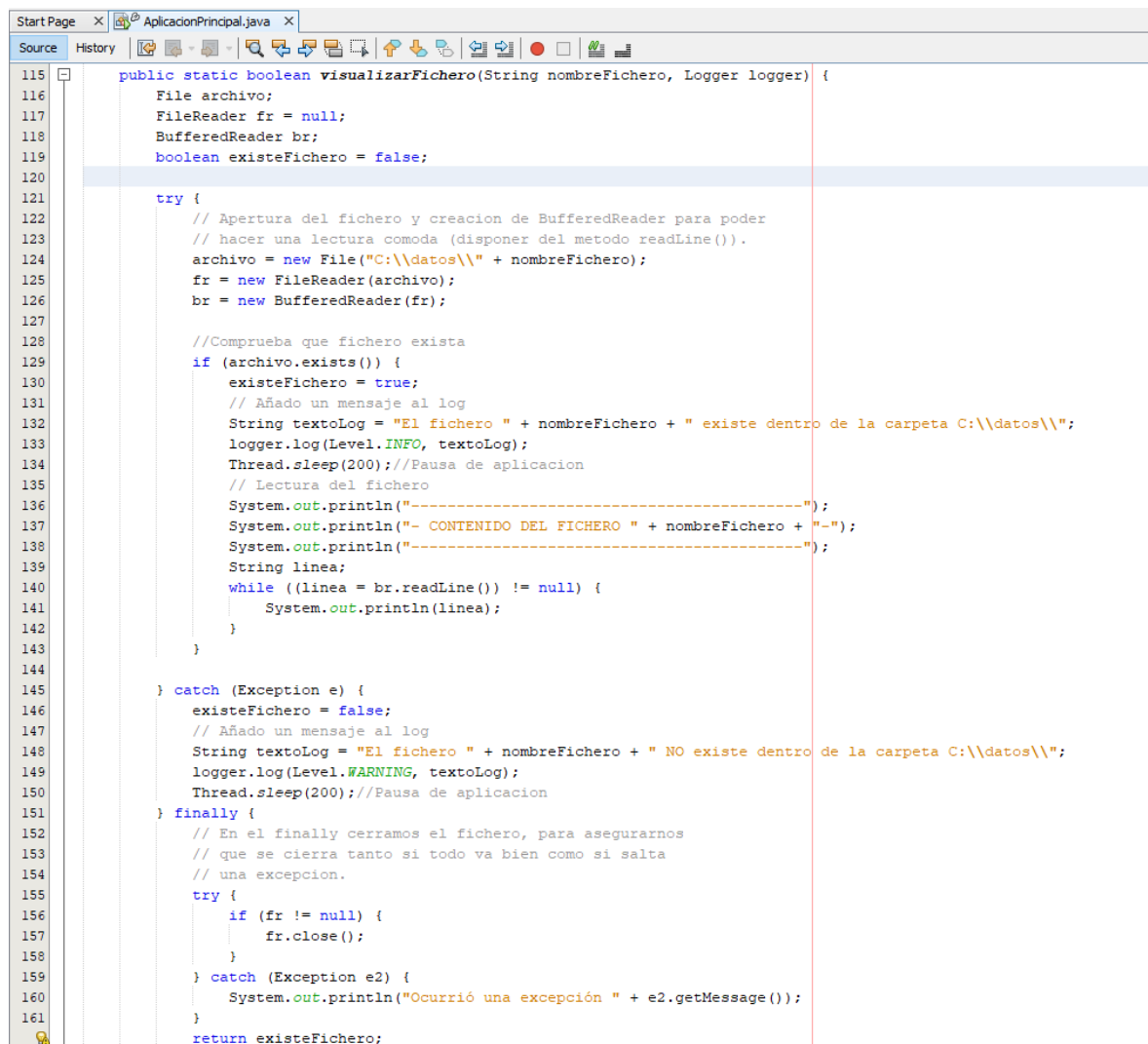
    } catch (Exception e) { //Recogemos excepción si las hubiera
        System.out.println("Se ha producido la excepción " + e.getMessage());
        correcto = false;
    }

} while (!correcto);
}
```

Como en caso anterior solicitamos el nombre del fichero, lo recogemos y se lo pasamos como parámetro al método “comprobarPatron” junto con el patrón de fichero propuesto en la tarea. Si el patrón es correcto llamamos al método “visualizarFichero” para que muestre su contenido. Contempla también cualquier excepción controlada por un bucle para que el programa sepa reponerse del error y escribe en el fichero log cualquier información importante que se vaya produciendo.

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

Este apartado lo he metido en un método al que he llamado “visualizarFichero”, recoge como parámetro el Logger y el nombre del fichero:



```

115 public static boolean visualizarFichero(String nombreFichero, Logger logger) {
116     File archivo;
117     FileReader fr = null;
118     BufferedReader br;
119     boolean existeFichero = false;
120
121     try {
122         // Apertura del fichero y creacion de BufferedReader para poder
123         // hacer una lectura comoda (disponer del metodo readLine()).
124         archivo = new File("C:\\datos\\" + nombreFichero);
125         fr = new FileReader(archivo);
126         br = new BufferedReader(fr);
127
128         //Comprueba que fichero exista
129         if (archivo.exists()) {
130             existeFichero = true;
131             // Añado un mensaje al log
132             String textoLog = "El fichero " + nombreFichero + " existe dentro de la carpeta C:\\datos\\";
133             logger.log(Level.INFO, textoLog);
134             Thread.sleep(200); //Pausa de aplicacion
135             // Lectura del fichero
136             System.out.println("-----");
137             System.out.println("- CONTENIDO DEL FICHERO " + nombreFichero + "-");
138             System.out.println("-----");
139             String linea;
140             while ((linea = br.readLine()) != null) {
141                 System.out.println(linea);
142             }
143         }
144     } catch (Exception e) {
145         existeFichero = false;
146         // Añado un mensaje al log
147         String textoLog = "El fichero " + nombreFichero + " NO existe dentro de la carpeta C:\\datos\\";
148         logger.log(Level.WARNING, textoLog);
149         Thread.sleep(200); //Pausa de aplicacion
150     } finally {
151         // En el finally cerramos el fichero, para asegurarnos
152         // que se cierra tanto si todo va bien como si salta
153         // una excepcion.
154         try {
155             if (fr != null) {
156                 fr.close();
157             }
158         } catch (Exception e2) {
159             System.out.println("Ocurrió una excepción " + e2.getMessage());
160         }
161     }
162     return existeFichero;

```

Evalúa si el fichero existe o no, si es que sí lo muestra por pantalla. Se recoge cualquier tipo de excepción y se recoge en el fichero log las acciones más importantes que van sucediéndose

PRUEBAS DE LA APLICACIÓN

Ejecutamos, nos va a solicitar nombre de usuario y nombre de fichero, probamos con 3 intentos (2 incorrectos y 1 correcto), así podemos hacernos una idea del comportamiento y si funciona correctamente:

```

Output - Tarea_6_PSP (run) X
compile:
run:
Introduzca el nombre de usuario:
daniel
mar 11, 2022 10:40:40 PM actividad_6_1.AplicacionPrincipal main
ADVERTENCIA: El patron ([a-z]{8}) del usuario daniel es INCORRECTO
Introduzca el nombre de usuario:
Danielin
mar 11, 2022 10:41:01 PM actividad_6_1.AplicacionPrincipal main
ADVERTENCIA: El patron ([a-z]{8}) del usuario Danielin es INCORRECTO
Introduzca el nombre de usuario:
danielon
mar 11, 2022 10:41:09 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron ([a-z]{8}) del usuario danielon es correcto
Introduzca el nombre del fichero:
refran.tx
mar 11, 2022 10:41:35 PM actividad_6_1.AplicacionPrincipal main
ADVERTENCIA: El patron (^[a-zA-Z0-9]{1,8}.[a-zA-Z]{3}$) del fichero refran.tx es INCORRECTO
Introduzca el nombre del fichero:
refran.txt
mar 11, 2022 10:41:42 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron (^[a-zA-Z0-9]{1,8}.[a-zA-Z]{3}$) del fichero refran.txt es correcto
mar 11, 2022 10:41:42 PM actividad_6_1.AplicacionPrincipal visualizarFichero
ADVERTENCIA: El fichero refran.txt NO existe dentro de la carpeta C:\datos\
Introduzca el nombre del fichero:
refranes.txt
mar 11, 2022 10:41:53 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron (^[a-zA-Z0-9]{1,8}.[a-zA-Z]{3}$) del fichero refranes.txt es correcto
mar 11, 2022 10:41:53 PM actividad_6_1.AplicacionPrincipal visualizarFichero
INFORMACIÓN: El fichero refranes.txt existe dentro de la carpeta C:\datos\

-----
- CONTENIDO DEL FICHERO refranes.txt-
-----
NO POR MUCHO MADRUGAR AMANECE MAS TEMPRANO
AUNQUE LA MONA SE VISTA DE SEDA MONA QUEDA
AL QUE MADRUGA DIOS LO AYUDA
DONDE HUBO FUEGO CENIZAS QUEDAN
EL CASADO CASA QUIERE
A CABALLO REGALADO NO SE LE MIRAN LOS DIENTES
EN CASA DE HERRERO CUCHILLO DE PALO
CAMARON QUE SE DUERME SE LO LLEVA LA CORRIENTE
DE TAL PALO TAL ASTILLA
DONDE MANDA CAPITAN NO MANDA MARINERO
MAS VALE PAJARO EN MANO QUE CIEN VOLANDO
MUCHO RUIDO Y POCAS NUECES
POR LA BOCA MUERE EL PEZ
NO HAY PEOR CIEGO QUE EL QUE NO QUIERE VER
PERRO QUE LADRA NO MUERDE
CRIA CUERVOS Y TE QUITARAN LOS OJOS
DIME CON QUIEN ANDAS Y TE DIRE QUIEN ERES
DIOS APRIETA PERO NO AHORCA
QUIEN MUCHO ABARCA POCO APRIETA
HAZTE LA FAMA Y ECHATE A DORMIR
BUILD SUCCESSFUL (total time: 1 minute 30 seconds)

```

Después de introducir los datos en rojo se muestran los mensajes de log para mostrar información de las acciones que se van sucediendo. A su vez estos mensajes se van a registrar en el fichero de log que hemos creado. Finalmente se muestra el contenido del fichero por pantalla

FicheroRegistros.log: Bloc de notas

Archivo Edición Formato Ver Ayuda

```

mar 11, 2022 10:40:40 PM actividad_6_1.AplicacionPrincipal main
ADVERTENCIA: El patron ([a-z]{8}) del usuario daniel es INcorrecto
mar 11, 2022 10:41:01 PM actividad_6_1.AplicacionPrincipal main
ADVERTENCIA: El patron ([a-z]{8}) del usuario Danielin es INcorrecto
mar 11, 2022 10:41:09 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron ([a-z]{8}) del usuario danielon es correcto
mar 11, 2022 10:41:35 PM actividad_6_1.AplicacionPrincipal main
ADVERTENCIA: El patron (^[a-zA-Z0-9]{1,8}.[a-zA-Z]{3}$) del fichero refran.tx es INcorrecto
mar 11, 2022 10:41:42 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron (^[a-zA-Z0-9]{1,8}.[a-zA-Z]{3}$) del fichero refran.txt es correcto
mar 11, 2022 10:41:42 PM actividad_6_1.AplicacionPrincipal visualizarFichero
ADVERTENCIA: El fichero refran.txt NO existe dentro de la carpeta C:\datos\
mar 11, 2022 10:41:53 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron (^[a-zA-Z0-9]{1,8}.[a-zA-Z]{3}$) del fichero refranes.txt es correcto
mar 11, 2022 10:41:53 PM actividad_6_1.AplicacionPrincipal visualizarFichero
INFORMACIÓN: El fichero refranes.txt existe dentro de la carpeta C:\datos\

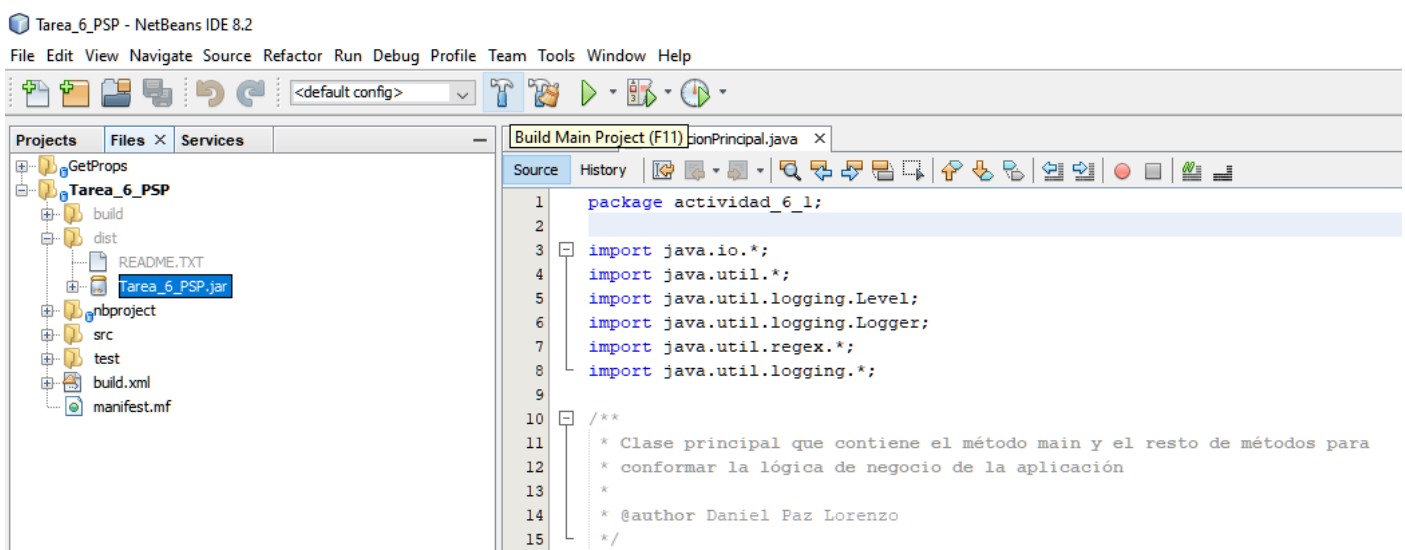
```

Este es el fichero de log que se ha ido generando cuando corríamos la aplicación

ACTIVIDAD 6.2

En este apartado se nos pide que utilizando la aplicación que hemos creado anteriormente, la firmemos digitalmente y que solo pueda leer los datos del directorio “c:/datos”.

Comenzamos compilando la aplicación en NetBeans:



Podemos ver en el navegador de la izquierda como se nos ha generado el paquete .jar

A continuación, abrimos la consola de comandos de Windows en modo administrador, nos dirigimos al directorio del proyecto y generamos un par de claves (privada y pública) para firmar el fichero con el comando "keytool":

```

Administrador: Símbolo del sistema
11/03/2022 12:55 <DIR>          build
10/03/2022 09:25          3.618 build.xml
11/03/2022 12:55 <DIR>          dist
10/03/2022 09:25          85 manifest.mf
10/03/2022 09:25 <DIR>          nbproject
10/03/2022 09:26 <DIR>          src
10/03/2022 09:29 <DIR>          test
                2 archivos          3.703 bytes
                7 dirs 244.689.059.840 bytes libres

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP>cd dist
D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>dir
El volumen de la unidad D es DATOS & ESTUDIO
El número de serie del volumen es: 1000-B5D6

Directorio de D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist
11/03/2022 12:55 <DIR>          .
11/03/2022 12:55 <DIR>          ..
11/03/2022 12:55          1.327 README.TXT
11/03/2022 12:55          6.283 Tarea_6_PSP.jar
                2 archivos          7.610 bytes
                2 dirs 244.689.059.840 bytes libres

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>keytool -genkey -alias daniel -keypass tarea6 -keystore RepoFirmaPSP -storepass dam
Error de herramienta de claves: java.lang.Exception: La contraseña del almacén de claves debe tener al menos 6 caracteres
D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>keytool -genkey -alias daniel -keypass tarea6 -keystore RepoFirmaPSP -storepass damDistancia
  
```

```

Administrador: Símbolo del sistema
[Unknown]: daniel paz
¿Cuál es el nombre de su unidad de organización?
[Unknown]: alumno dam
¿Cuál es el nombre de su organización?
[Unknown]: CIFP Camino de la Miranda
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Valladolid
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Valladolid
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: ES
¿Es correcto CN=daniel paz, OU=alumno dam, O=CIFP Camino de la Miranda, L=Valladolid, ST=Valladolid, C=ES?
[no]: si

Warning:
El almacén de claves JKS utiliza un formato propietario. Se recomienda migrar a PKCS12, que es un formato e
stándar del sector que utiliza "keytool -importkeystore -srckeystore RepoFirmaPSP -destkeystore RepoFirmaPS
P -deststoretype pkcs12".

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>
  
```

Rellenamos los datos que nos van solicitando

Este equipo > DATOS & ESTUDIO (D:) > Curso DAM > Repositorios GitHub > Asignaturas-2-DAM > PSP > Tareas > Tarea 6 > Tarea_6_PSP > dist

	Nombre	Fecha de modificación	Tipo	Tamaño
DAM	README.TXT	11/03/2022 12:55	Documento de te...	2 KB
	RepoFirmaPSP	11/03/2022 13:14	Archivo	2 KB
	Tarea_6_PSP.jar	11/03/2022 12:55	Executable Jar File	7 KB

Si entramos en el directorio ya vemos que se nos ha creado el almacenamiento de claves

Por último, vamos a firmar la aplicación con el certificado que hemos creado con el comando “jarsigner”:

```

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>dir
El volumen de la unidad D es DATOS & ESTUDIO
El número de serie del volumen es: 1000-B5D6

Directorio de D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist

11/03/2022 13:14 <DIR>          .
11/03/2022 13:14 <DIR>          ..
11/03/2022 12:55              1.327 README.TXT
11/03/2022 13:14              2.035 RepoFirmaPSP
11/03/2022 12:55              6.283 Tarea_6_PSP.jar
                        3 archivos          9.645 bytes
                        2 dirs 244.688.896 bytes libres

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>jarsigner -keystore
RepoFirmaPSP -signedjar signedTarea_6_PSP.jar Tarea_6_PSP.jar daniel
Enter Passphrase for keystore:
jarsigner error: java.lang.RuntimeException: keystore load: Keystore was tampered with, or password was inc
orrect

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>jarsigner -keystore
RepoFirmaPSP -signedjar signedTarea_6_PSP.jar Tarea_6_PSP.jar daniel
Enter Passphrase for keystore:
Enter key password for daniel:
jar signed.

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

Curso DAM > Repositorios GitHub > Asignaturas-2-DAM > PSP > Tareas > Tarea 6 > Tarea_6_PSP > dist

Nombre	Fecha de modificación	Tipo	Tamaño
README.TXT	11/03/2022 12:55	Documento de te...	2 KB
RepoFirmaPSP	11/03/2022 13:14	Archivo	2 KB
signedTarea_6_PSP.jar	11/03/2022 13:46	Executable Jar File	8 KB
Tarea_6_PSP.jar	11/03/2022 12:55	Executable Jar File	7 KB

Comprobamos que dentro del directorio se nos ha creado un nuevo paquete .jar ya firmado digitalmente

Ya solo nos queda exportar la llave pública de nuestro certificado para que un usuario externo pueda utilizar la aplicación validando mi firma digital. Para ello utilizamos nuevamente el comando “keytool” con el atributo “-export” de la siguiente manera:

```

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

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>dir
El volumen de la unidad D es DATOS & ESTUDIO
El número de serie del volumen es: 1000-B5D6

Directorio de D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist

11/03/2022 13:46 <DIR>          .
11/03/2022 13:46 <DIR>          ..
11/03/2022 12:55              1.327 README.TXT
11/03/2022 13:14              2.035 RepoFirmaPSP
11/03/2022 13:46              8.130 signedTarea_6_PSP.jar
11/03/2022 12:55              6.283 Tarea_6_PSP.jar
                        4 archivos          17.775 bytes
                        2 dirs 244.688.551.936 bytes libres

D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>keytool -export
-keystore RepoFirmaPSP -alias daniel -file Daniel.cert
Introduzca la contraseña del almacén de claves:
Certificado almacenado en el archivo <Daniel.cert>

Warning:
El almacén de claves JKS utiliza un formato propietario. Se recomienda migrar a PKCS12, que es un form
ato estándar del sector que utiliza "keytool -importkeystore -srckeystore RepoFirmaPSP -destkeystore R
epoFirmaPSP -deststoretype pkcs12".
  
```


Volvemos al directorio para comprobar que se nos ha generado el certificado:

h) > Curso DAM > Repositorios GitHub > Asignaturas-2-DAM > PSP > Tareas > Tarea 6 > Tarea_6_PSP > dist

Nombre	Fecha de modificación	Tipo	Tamaño
Daniel.cert	11/03/2022 13:54	Archivo CERT	2 KB
README.TXT	11/03/2022 12:55	Documento de te...	2 KB
RepoFirmaPSP	11/03/2022 13:14	Archivo	2 KB
signedTarea_6_PSP.jar	11/03/2022 13:46	Executable Jar File	8 KB
Tarea_6_PSP.jar	11/03/2022 12:55	Executable Jar File	7 KB

El siguiente apartado nos pide que la aplicación solo pueda leer los datos del directorio “c:/datos”, para ello tendremos que configurar la política de seguridad de nuestra aplicación firmada.

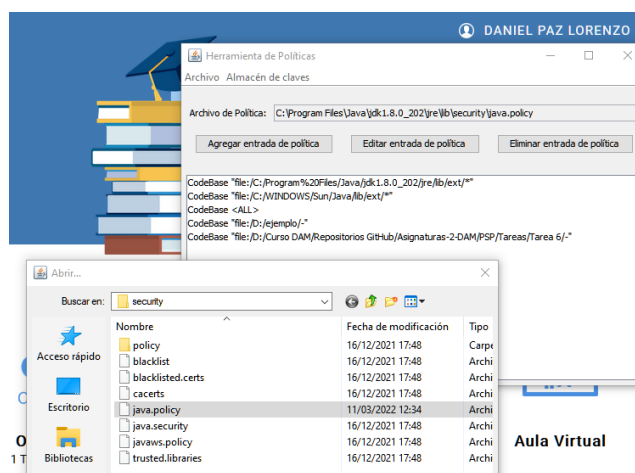
Lo primero que haremos será importar el certificado para validar la firma del fichero utilizando el comando “keytool -import”:

```

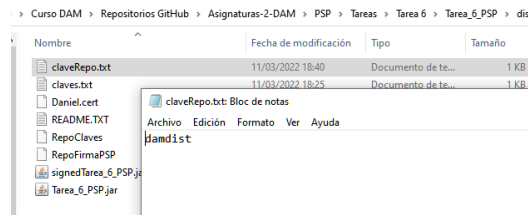
D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>keytool -import -alias daniel -file daniel.cert
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
Propietario: CN=daniel paz, OU=alumno dam, O=CIFP Camino de la Miranda, L=Valladolid, ST=Valladolid, C=ES
Emisor: CN=daniel paz, OU=alumno dam, O=CIFP Camino de la Miranda, L=Valladolid, ST=Valladolid, C=ES
Número de serie: 7f0d27a
Válido desde: Fri Mar 11 13:14:45 CET 2022 hasta: Thu Jun 09 14:14:45 CEST 2022
Huellas digitales del certificado:
MD5: BD:F4:40:D4:15:7C:5C:4A:5B:A5:BA:B6:CA:EA:9F:83
SHA1: CE:DA:E6:ED:22:17:E4:3C:8F:82:08:C4:CE:D1:DC:DC:03:D8:FD:18
SHA256: 97:09:9A:AD:49:56:2C:B0:88:6A:82:D9:E8:10:09:B9:6A:D5:76:68:C0:40:AF:46:EC:5E:B7:DF:BE:7B:F1:C2
Nombre del algoritmo de firma: SHA256withDSA
Algoritmo de clave pública de asunto: Clave DSA de 2048 bits
Versión: 3
Extensiones:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 8C 34 9B D0 43 C7 C0 76 E1 1B 61 D5 C0 EF 69 E5 .4..C..v..a...i.
0010: D4 B0 78 C6 ..x.
]
]
¿Confiar en este certificado? [no]: si
Se ha agregado el certificado al almacén de claves
  
```

Al ejecutar nos solicita contraseña para el nuevo almacén de claves creado

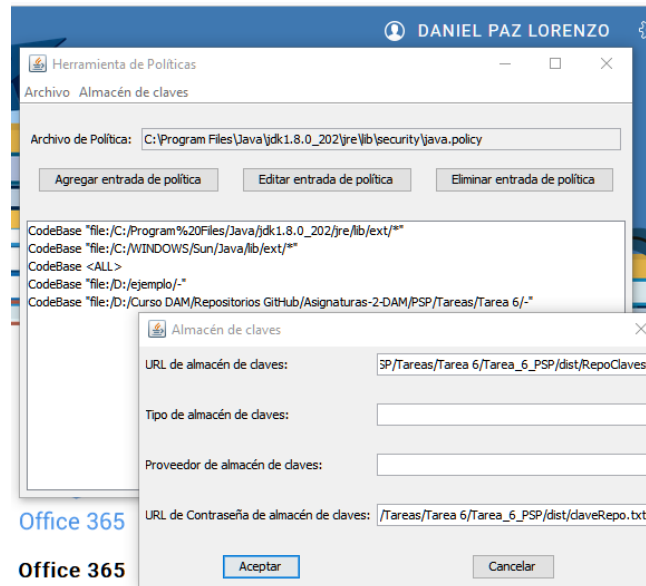
Pasamos a configurar la política de seguridad para que permita ejecutar el fichero firmado:



Abrimos la herramienta policytool, pulsamos Archivo->Abrir, seleccionamos java.policy y lo abrimos

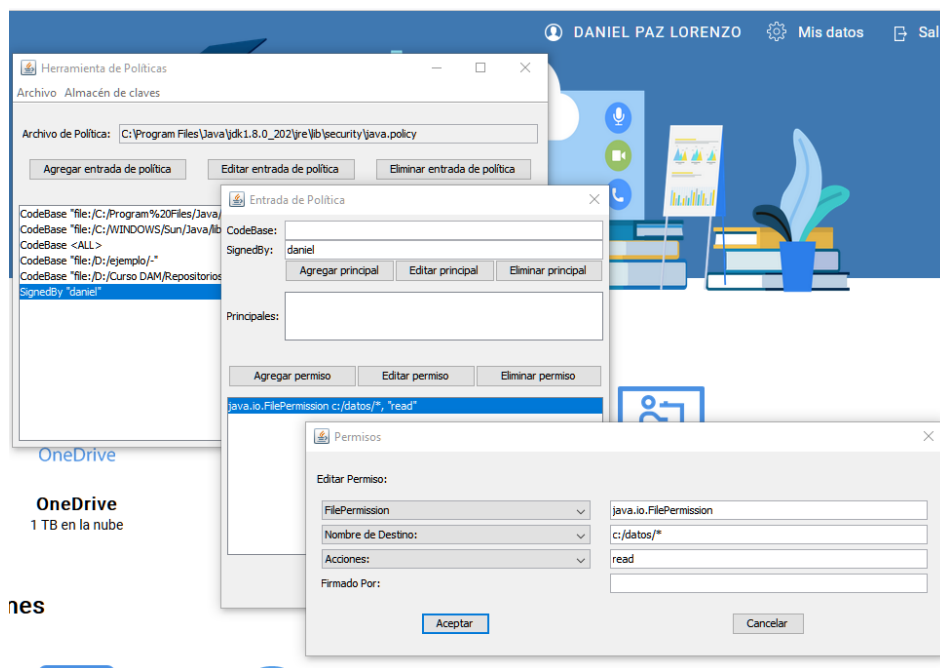


Creamos un .txt dentro del mismo directorio que contenga la clave del repositorio de claves



Abrimos el almacén de claves y metemos las 2 url del repositorio de las claves y del fichero con la contraseña del repositorio

Una vez tenemos importado el almacén de claves vamos a agregar una entrada de política para que la aplicación pueda acceder a la carpeta C:/datos:



Establecemos los parámetros necesarios y guardamos

Ya solo nos queda comprobar que hemos configurado correctamente la política de seguridad ejecutando la aplicación.

1. Este primer pantallazo corresponde a una ejecución anterior a configurar las políticas de seguridad donde podemos ver que tenemos denegado el acceso a c:/datos:

```
D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>java -Djava.security.manager -jar signedTarea_6_PSP.jar
mar 11, 2022 7:04:24 PM actividad_6_1.AplicacionPrincipal creaFicheroRegistro
GRAVE: null
java.security.AccessControlException: access denied ("java.io.FilePermission" "c:\datos" "write")
    at java.security.AccessControlContext.checkPermission(AccessControlContext.java:472)
    at java.security.AccessController.checkPermission(AccessController.java:884)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:549)
    at java.lang.SecurityManager.checkWrite(SecurityManager.java:979)
    at sun.nio.fs.WindowsPath.checkWrite(WindowsPath.java:799)
    at sun.nio.fs.WindowsFileSystemProvider.checkAccess(WindowsFileSystemProvider.java:372)
    at java.nio.file.Files.isAccessible(Files.java:2455)
    at java.nio.file.Files.isWritable(Files.java:2521)
    at java.util.logging.FileHandler.isParentWritable(FileHandler.java:428)
    at java.util.logging.FileHandler.openFiles(FileHandler.java:488)
    at java.util.logging.FileHandler.<init>(FileHandler.java:344)
    at actividad_6_1.AplicacionPrincipal.creaFicheroRegistro(AplicacionPrincipal.java:175)
    at actividad_6_1.AplicacionPrincipal.main(AplicacionPrincipal.java:31)
```

2. Después de configurar y guardar la nueva política de seguridad ejecutamos la aplicación y vemos como ahora si no nos muestra ningún mensaje de error y podemos ejecutar la aplicación sin restricciones:

```
Administrador: Símbolo del sistema
D:\Curso DAM\Repositorios GitHub\Asignaturas-2-DAM\PSP\Tareas\Tarea 6\Tarea_6_PSP\dist>java -Djava.security.manager -jar signedTarea_6_PSP.jar
Introduzca el nombre de usuario:
danielin
mar 11, 2022 7:06:37 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron ([a-z]{8}) del usuario danielin es correcto
Introduzca el nombre del fichero:
refranes.txt
mar 11, 2022 7:06:46 PM actividad_6_1.AplicacionPrincipal main
INFORMACIÓN: El patron (^[a-zA-Z0-9]{1,8}).[a-zA-Z]{3}$ del fichero refranes.txt es correcto
mar 11, 2022 7:06:46 PM actividad_6_1.AplicacionPrincipal visualizarFichero
INFORMACIÓN: El fichero refranes.txt existe dentro de la carpeta C:\datos\
-----
- CONTENIDO DEL FICHERO refranes.txt -
-----
NO POR MUCHO MADRUGAR AMANECE MAS TEMPRANO
AUNQUE LA MONA SE VISTA DE SEDA MONA QUEDA
AL QUE MADRUGA DIOS LO AYUDA
DONDE HUBO FUEGO CENIZAS QUEDAN
EL CASADO CASA QUIERE
A CABALLO REGALADO NO SE LE MIRAN LOS DIENTES
EN CASA DE HERRERO CUCHILLO DE PALO
CAMARON QUE SE DUERME SE LO LLEVA LA CORRIENTE
DE TAL PALO TAL ASTILLA
DONDE MANDA CAPITAN NO MANDA MARINERO
MAS VALE PAJARO EN MANO QUE CIEN VOLANDO
MUCHO RUIDO Y POCAS NUECES
POR LA BOCA MUERE EL PEZ
NO HAY PEOR CIEGO QUE EL QUE NO QUIERE VER
PERRO QUE LADRA NO MUERDE
CRIA CUERVOS Y TE QUITARAN LOS OJOS
DIME CON QUIEN ANDAS Y TE DIRE QUIEN ERES
DIOS APRIETA PERO NO AHORCA
QUIEN MUCHO ABARCA POCO APRIETA
HAZTE LA FAMA Y ECHATE A DORMIR
```