**ACTIVIDAD 2**

**Enunciado**

Realiza un programa en Java que verifique la validez de un fichero de firma previamente generado. A modo orientativo, la estructura del programa podría ser la siguiente:

* Leer la clave pública del fichero a verificar. Dicha clave está dentro de un fichero de nombre *Clave.publica* disponible en la plataforma.
* Recuperar la clave pública desde los datos codificados en formato X509 y mediante el algoritmo *DSA*.
* Leer el fichero que contiene la firma. Dicho fichero se llama *FICHERO.FIRMA* y está disponible en la plataforma.
* Inicializar el objeto *Signature* con la clave pública que se va a usar para verificar la validez del fichero y mediante el algoritmo *SHA1withDSA.*
* Leer el fichero que contiene los datos a verificar (*FICHERO.DAT*), el cual está disponible en la plataforma.
* Verificar la firma de los datos leídos del fichero *FICHERO.DAT.*
* Comprobar la verificación de los datos leídos. Si la verificación es correcta se mostrará el mensaje “LOS DATOS SE CORRESPONDEN CON SU FIRMA”. En caso contrario, se mostrará este otro mensaje: “LOS DATOS NO SE CORRESPONDEN CON SU FIRMA”.

El programa gestionará las posibles excepciones que se puedan producir durante su ejecución.

El nombre de la clase principal será Actividad2.

package ejercicio2;

import java.io.BufferedInputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.security.InvalidKeyException;

import java.security.KeyFactory;

import java.security.NoSuchAlgorithmException;

import java.security.PublicKey;

import java.security.Signature;

import java.security.SignatureException;

import java.security.spec.InvalidKeySpecException;

import java.security.spec.X509EncodedKeySpec;

public class Actividad2 {

public static void main(String[] args) {

// Abrimos los archivos en el try, para que se cierren solos

try (

// Clave publica

FileInputStream fisPublica = new FileInputStream(new File("src/ejercicio2/Clave.publica"));

// Ficehro con la firma

FileInputStream fisFirma = new FileInputStream(new File("src/ejercicio2/FICHERO.FIRMA"));

// Fichero con la información

BufferedInputStream bisDat = new BufferedInputStream(

new FileInputStream(new File("src/ejercicio2/FICHERO.DAT")));) {

// Leemos lca clave publica y la ponemos en un buffer

byte[] bufferPublica = new byte[fisPublica.available()];

fisPublica.read(bufferPublica);

// Ponemos la clave en formato x509

X509EncodedKeySpec clavePublicaSpec = new X509EncodedKeySpec(bufferPublica);

KeyFactory keyDSA = KeyFactory.getInstance("DSA");

PublicKey clavePublicaKey = keyDSA.generatePublic(clavePublicaSpec);

// Leemos la firma

byte[] bufferFirma = new byte[fisFirma.available()];

fisFirma.read(bufferFirma);

// Verificamos la clave publcia con SHA1withDSA

Signature signature = Signature.getInstance("SHA1withDSA");

signature.initVerify(clavePublicaKey);

// Leemos los datos

byte[] bufferDatos = bisDat.readAllBytes();

System.out.println(new String(bufferDatos));

signature.update(bufferDatos);

// Verificamos si los datos que tenemos y los que nos llegan son los mismos

if (signature.verify(bufferFirma)) {

System.out.println("LOS DATOS SE CORRESPONDEN CON SU FIRMA");

} else {

System.out.println("LOS DATOS NO SE CORRESPONDEN CON SU FIRMA");

}

} catch (IOException | NoSuchAlgorithmException | InvalidKeySpecException | InvalidKeyException

| SignatureException e) {

e.printStackTrace();

}

}

}

**REQUISITOS**

1. Entregar cada actividad en un archivo comprimido que contenga los ficheros \*.java empleados.
2. Los archivos comprimidos seguirán la siguiente nomenclatura:

*UD5\_Actividadn\_apellido*

donde:

*n*: valdrá 2, representando el número de la actividad

*apellido*: será el primer apellido del alumno

**CRITERIOS DE CORRECCIÓN**

1. Cumplimiento de los requisitos (15%).
2. Hacer lo que se indica en el enunciado (55%).
3. Claridad del código Java (15%).
4. Documentación del código Java (15%).
5. Explicación detallada de cómo se han resuelto problemas que hayan podido surgir en la realización de la actividad (hasta un 15% adicional sobre la nota obtenida). Dicha explicación se incluirá al principio de los archivos de código fuente Java mediante líneas de comentarios. La primera línea de comentarios sólo incluirá lo siguiente: // Resolución de problemas.
6. Variables no usadas (-15%).
7. Librerías no usadas (-15%).