PRÁCTICA 3

Implementación de un servicio básico de autenticación con DNIe

LAURA MORENO CHICHARRO

DAVID PADILLA ALVARADO

ÍNDICE

INTRODUCCIÓN

Esta práctica consiste en la implementación de un servicio de autenticación básico empleando el DNI.

El objetivo consiste que al introducir el dni en el lector de tarjeta e introduciendo el usuario y la contraseña a través de un formulario, éste permita acceder correctamente.

Para implementar este servicio hemos utilizado:

-Lector de tarjetas

-Software del dni electrónico (download página oficial)

- NetBeans

-JDK 1.7

-XAMPP

-Navegador web

-Github

-Tener una conexión a Internet.

En nuestra práctica disponemos de una parte que hace de cliente (Acceso) en java y de una parte que hace de servidor (dnie) en php.

CLIENTE

Para el cliente tenemos dos clases: la clase principal ‘Main.java’ y la clase ‘ObtenerDatos.java’.

Main.java

Es la clase principal del programa, en ella declaramos las variables a utilizar en toda la clase y además hemos incluido la función hash (sha1) que es la encargada de codificar los datos que le pasamos al servidor.

También creamos un objeto del método ObtenerDatos, el cual se encuentra dentro de la clase ObtenerDatos.java. para poder leer los datos del dni que introducimos en el lector de tarjetas y además leer el nombre.

Para ello una vez que llamamos al método LeerNif, realizamos un resumen de acorde a los datos que nosotros queremos obtener. Es decir, nosotros hemos obtenido la primera letra del nombre, el segundo apellido entero y la primera letra del segundo apellido, así formaríamos también el nombre de usuario con las mismas características.

De forma equivalente, seleccionamos el dni junto con la letra del mismo y una clave que será la contraseña que introduciremos por pantalla.

Todo esto lo concatenamos en una sola variable.

De manera que si el nombre es Moreno Chicharro, Laura con dni: 26248707A el resumen quedaría lmorenoc26248707a1234, siendo 1234 la contraseña que hemos introducido por pantalla.

De manera que le hacemos la función Hash a la variable final, es decir, a la variable que contiene todos estos datos concatenados.

Al no haber encontrado ningún código adaptable de base64 para JDK 1.7, cuando hacemos la petición HTTP al servidor, en vez de pasarle los datos en Base64, le pasamos directamente el hash.

ObtenerDatos.java

Esta clase es empleada para obtener los datos de la zona pública del dni.

Esta clase implementa cuatro métodos para obtener los datos que se encuentran en los certificados del dni. Los métodos implementados son para establecer la conexión con la tarjeta (utilizando lector de tarjetas), comprobación de la tarjeta, si es dni o no, leer el NIF del certificado y otro que recorre el dni.

SERVIDOR

El servidor en la zona encargada de realizar la autenticación, para ellos tenemos un formulario, y dos clases en php.

Hay dos métodos de autenticación:

-**Autenticación básica sin protección**, en la cual solo comprobamos los datos que introducimos con los de la base de datos creada, si son iguales entonces la autenticación es correcta.

-**Autenticación mediante un código de autenticación**, en la cual generamos comparamos dos hash, uno el producido de los datos de la tarjeta dni y otro el producido de los datos de la base de datos, si ambos coinciden entonces la autenticación es correcta.

Autentica.php

Es el encargado de hacer la autenticación básica, conectándose a la base de datos y comprobar que los datos que metemos en el formulario, tanto el dni, el usuario y la contraseña, coinciden con los datos que hay en la base de datos.

Si estos datos coinciden la autenticación es correcta, en caso contrario no se podrá acceder.

AutenticaMac.php

Es la parte del servidor encargada de hacer la autenticación segura, donde le pasamos los datos a través del método GET (los datos van en la URL).

Establecemos una conexión con la base de datos. Extraemos los datos, le hacemos un resumen de igual forma que hemos hecho en el main.java, de manera que al hacer el resumen, quedaría: lmorenoc26248707a1234, siendo 1234 la contraseña introducida.

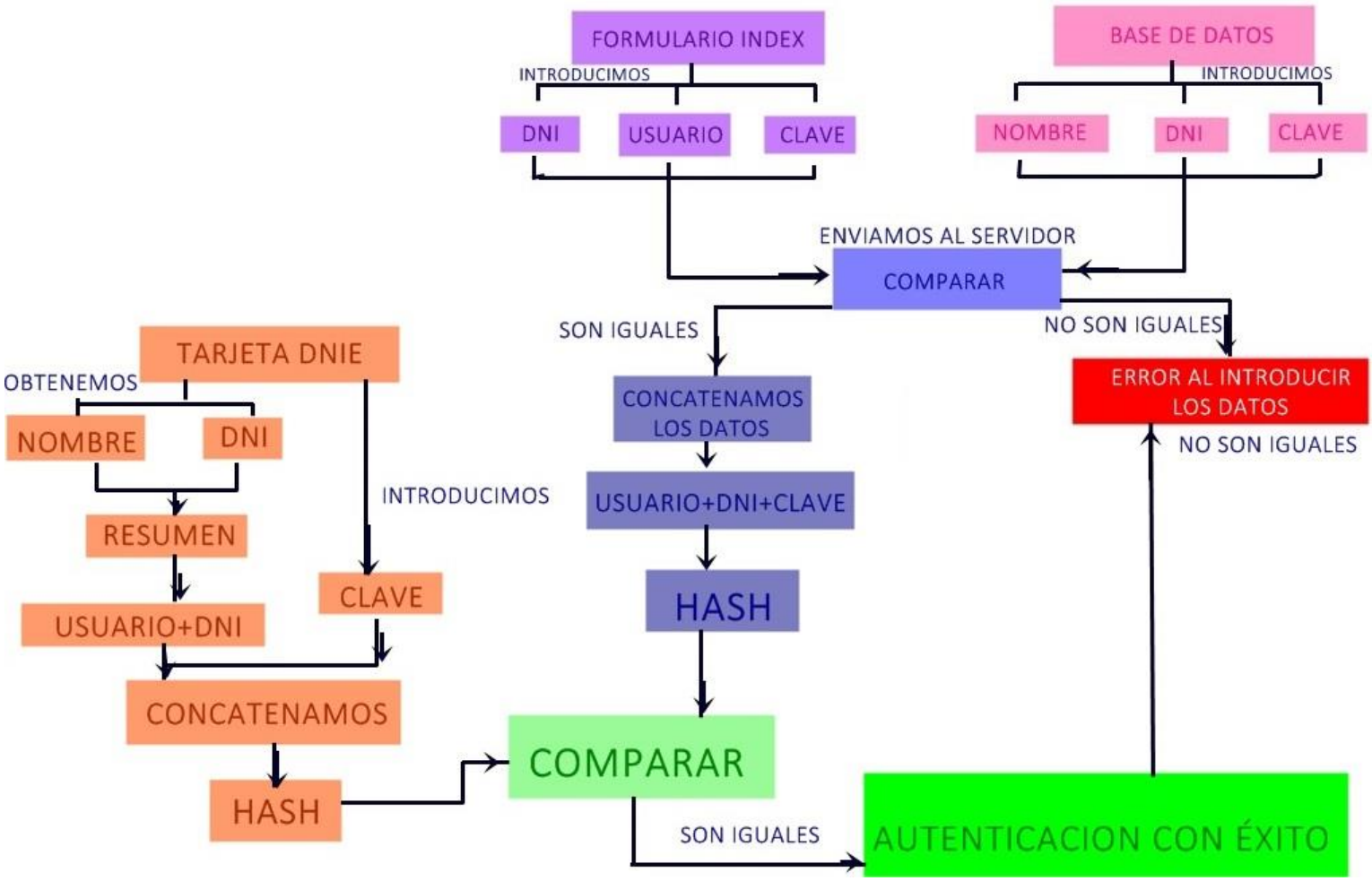
Una vez que hacemos el resumen, lo codificamos con Hash y finalmente comparamos ambos hash. El hash que hemos producido con los datos de la base de datos y el hash que le hemos pasado del cliente. Si ambos son iguales, entonces la autenticación es exitosa.

Index.php

Es el formulario utilizado para introducir los datos, éste conecta con autentica.php si queremos hacer una autenticación básica y con autenticaMac.php si lo que queremos es una autenticación segura.

AUTENTICACIÓN BÁSICA

AUTENTICACIÓN SEGURA



PROCESO LLEVADO A CABO PARA SU CORRECTO FUNCIONAMIENTO

-Empezamos introduciendo un nuevo usuario en la base de datos ‘dniauth’

```
insert into users value (1, "lmorenoc", "1234","26248707A" );
```

-Realizamos el cliente modificando el fichero ObtenerDatos.java, realizando así la lectura del dni para que nos extrajese el usuario y el dni dentro del método LeerDeCertificado.

```
    if ((byte) datos[offset] == (byte) 0xA1) {
        //El certificado empieza aquí
        byte[] r3 = new byte[20];

        byte[] r5 = new byte[30];

        /**
         * Recorreos el dni para que nos muestre el usuario (lmorenoc)
         * y el dni junto con la letra(26248707a)
         */

        //Nos posicionamos en el byte donde empieza el NIF y leemos sus 9 bytes
        for (int z = 0; z < 9; z++) { //lee el dni
            r3[z] = datos[109 + z];
        }


        for (int z = 0; z < 24; z++) {
            r5[z] = datos[162 + z];
        }
        nombre= new String(r5)+ new String(r3);

    }
}
return nombre;
```

- Implementamos el main.java
  - -Buscamos un código que implementase la función Sha1 [2]
  - -Buscamos código petición http con GET [4]
  - -Hacemos el resumen de los datos que leemos del dni
  - -Realizamos el hash a esos datos concatenados
  - -Los mostramos por pantalla

```
public static String sha1(String input) throws NoSuchAlgorithmException {
    MessageDigest mDigest = MessageDigest.getInstance("SHA1");
    byte[] result = mDigest.digest(input.getBytes());
    StringBuffer sb = new StringBuffer();
    for (int i = 0; i < result.length; i++) {
        sb.append(Integer.toString((result[i] & 0xff) + 0x100, 16).substring(1));
    }

    return sb.toString();
}
```

- Cuando acabamos el cliente empezamos a hacer el servidor
    - -Buscamos el código sha1 en php [3]
    - -Añadimos i a todos los mySQL, ya que era problema de la versión de MySQL
    - -Cambiamos de orden ('dniauth',$link) por ($link,'dniauth')
    - -Quitamos la contraseña del código de la base de datos, ya que al utilizar XAMPP por defecto te crea una instancia sin contraseña.
    - -Tuvimos problema al acceder a la base de datos porque en el usuario teníamos 26248704A y nosotros metíamos la letra en minúscula.
    - -El problema que también tenemos es que si queremos que lo muestre en el main, en el index no funciona, y viceversa.
    - -Otro problema es que al ejecutar el programa, nos muestra un código html por pantalla.
    - -No he podido implementar Base64 al utilizar JDK 1.7.
    - - Finalmente comparamos los datos que introducimos en el index con los datos de la base de datos, si son iguales les hace el hash y ya los compara con el hash que le pasa el cliente de la tarjeta del dni.

- Realización de JavaDoc

-Realización de la memoria

USOS EN LA VIDA COTIDIANA

-Acceder a Registros Telemáticos
-Consultar los puntos DGT
-Pedir cita médica

BIBLIOGRAFÍA

[1] Archivos (Acceso y DNI) descargados de docencia virtual

[2] http://www.sha1-online.com/sha1-java/

[3] http://php.net/manual/es/function.sha1.php

[4] https://lefunes.wordpress.com/2008/02/14/accediendo-a-un-sitio-httphttps-desde-java/