EPSL

Servicio básico de DNIe

Práctica 3

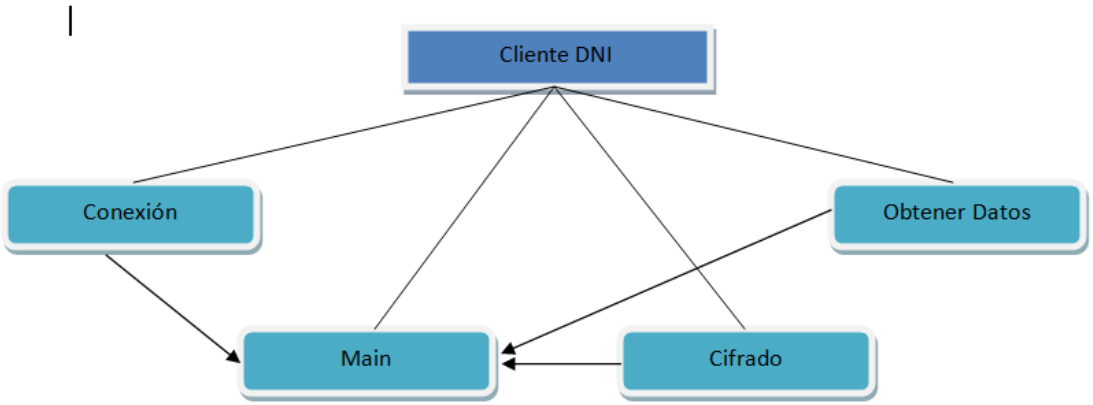
Emilio Sánchez Catalán

11/04/2016

En este documento se expondrá toda la información necesaria para entender el funcionamiento de los programas diseñados para dicha práctica.

Índice

Estructura principal del programa cliente.



El programa cliente consta de 3 clases:

- Obtener Datos: Clase cuya función consiste en extraer la información alojada en la parte publica del DNIe.
- Cifrado: Su función es manipular los datos introducidos y devolver el mensaje con su hash cifrado.
- Conexionhttp: Clase cuya función es crear y mandar los datos al servidor, para la autentificación.
- Main: Es la función principal del programa. Es la que hace uso de las dos clases anteriores.

Obtener Datos

Métodos:

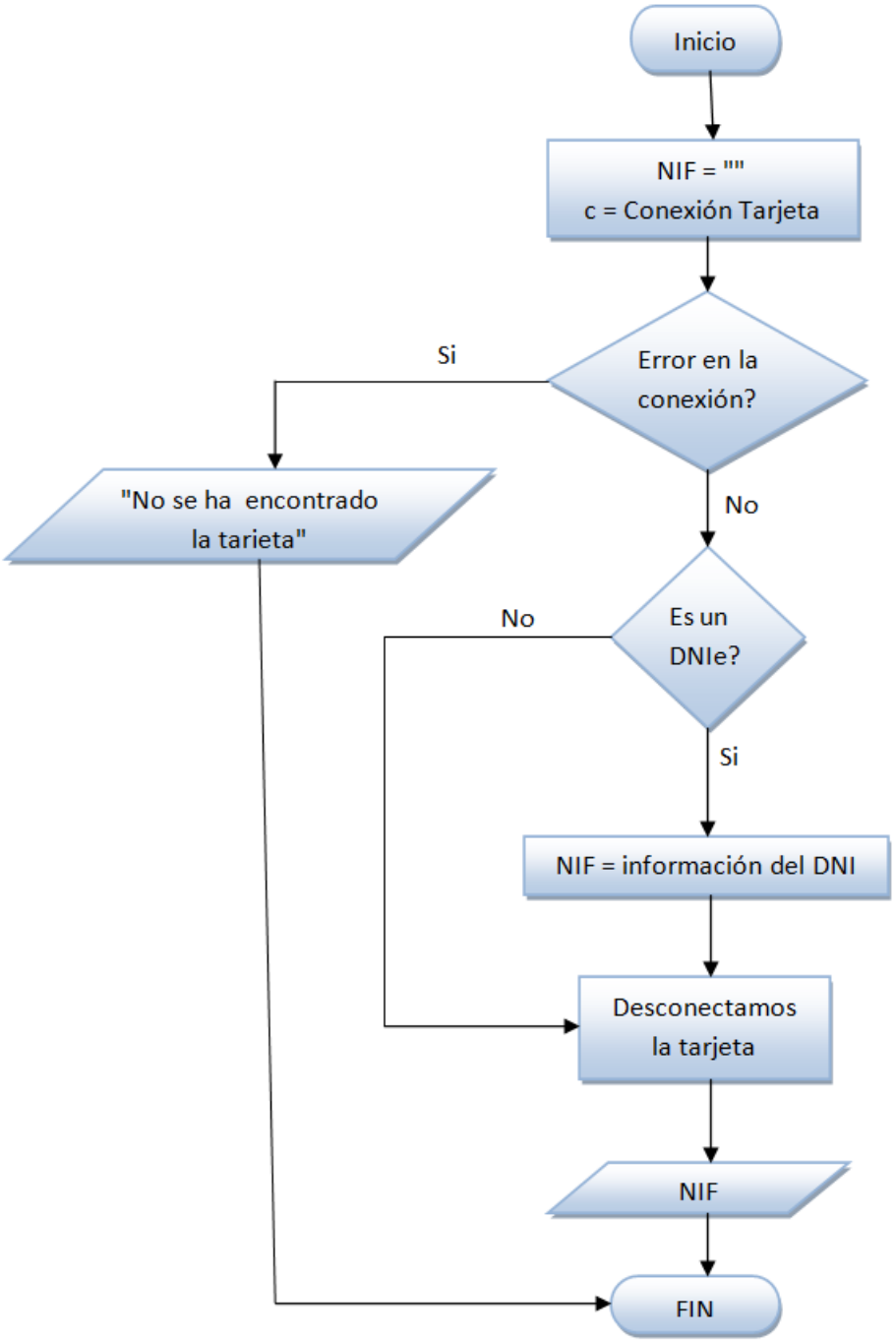
- oBtenerDatos: Método principal de la clase ya que es aquel que se encarga en hacer uso del resto de los métodos para realizar la conexión con el DNIe y obtener los datos y devolverlos.
- leerDeCertificado: Selección y lectura de los bytes del DNIe.
- ConexionTarjeta: Comprueba que exista la conexión de una tarjeta en el lector.
- esDNIe: Da a conocer el tipo de tarjeta que estamos leyendo.

Forma de uso:

```
#Obtenerdatos obj_Obtenerdatos = new Obtenerdatos();

#String var = obj_Obtenerdatos.oBtenerDatos();
```

Diagrama de flujo:



Cifrado

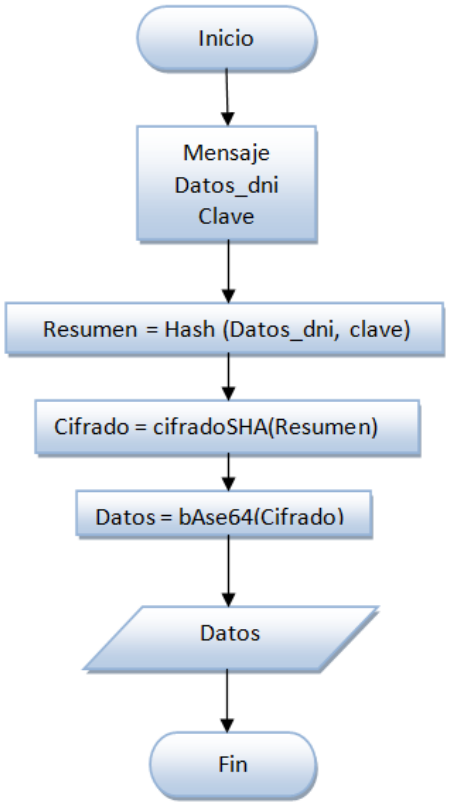
Métodos:

- hash(datos, clave): Método el cual coge los datos recibidos y les procesa un resumen de los mismos y se le suma clave. La estructura de los datos debe ser: "var1 var2 var3 var4".
- cifradoSHA(mensaje): Método que coge el mensaje recibido le aplica el cifrado SHA-1.
- bAse64(codehas): Método que aplica al codehas una codificación base64.

- cifrar(mensaje, datos, clave): Método principal el cual hace uso de los métodos anteriores y obtiene en base64 el mensaje más el hash generado por el resumen de los datos del usuario y su clave.

Forma de uso:
#String var = Cifrado.cifrar(mensaje, datosnif, clave);

Diagrama de flujo:



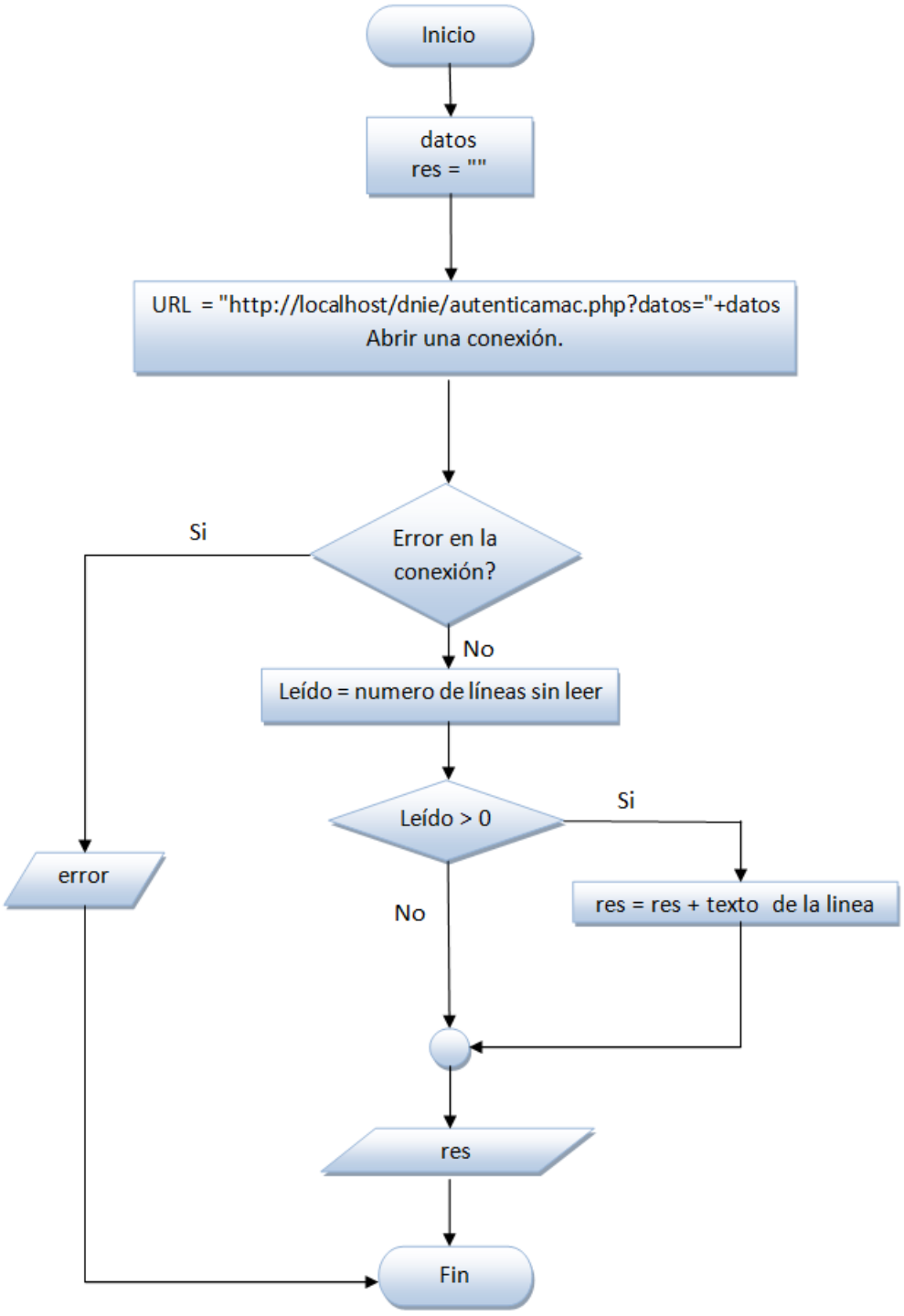
Conexionhttp

Métodos:

- oPen(datos): Método el cual abre la conexión http con el servidor y manda los datos mediante el método GET.

Forma de uso:
#String var = Conexionhttp.oPen(datos);

Diagrama de flujo:



Main

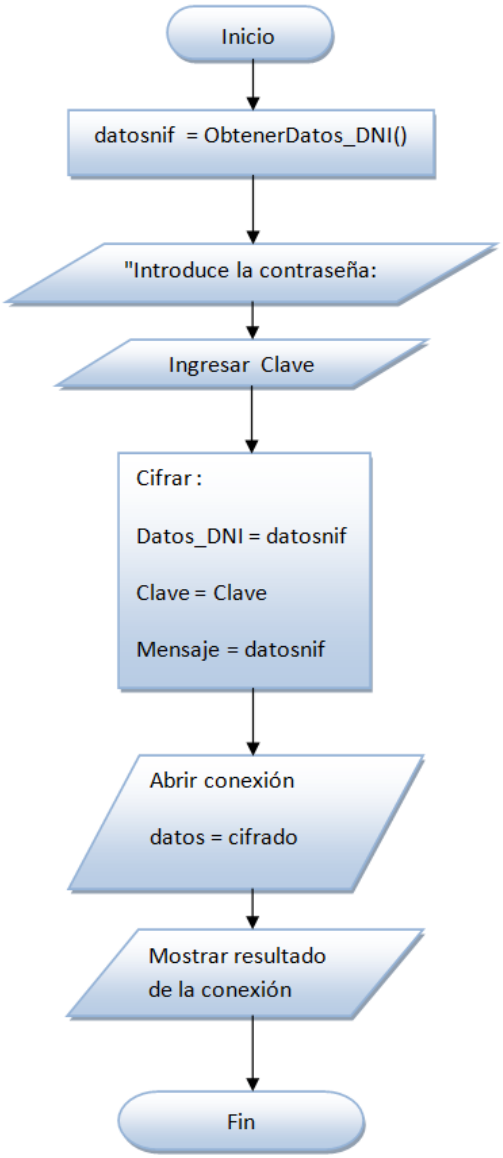
Métodos:

- main: método principal y arranque del programa.

Forma de uso:

Llevar a ejecutar. No hace uso de los argumentos del main.

Diagrama de flujo:



Estructura principal del programa servidor

Se ha programado un servicio php usando programación estructurada. Aunque en el programa se pueden diferenciar tres partes: recepción de datos, acceso a la base de datos, comprobación de datos.

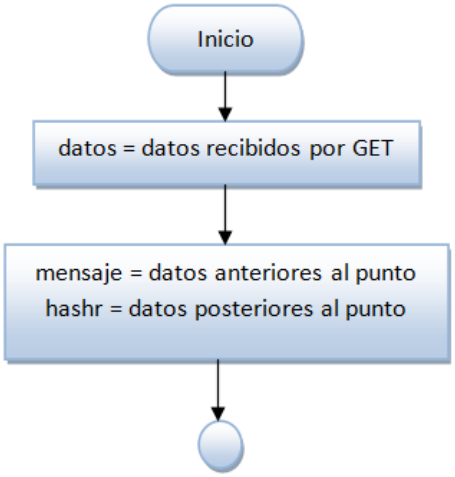
Para comentar al mismo, usaremos dicha diferenciación:

Recepción de datos

Funcionamiento:

El servidor recoge los parámetros recibidos por el método GET con nombre de variable datos. Posteriormente decodifica en base64 la información recibida y divide la información en dos variables una anterior a un punto (mensaje) y otra posterior (hash).

Diagrama de flujo:

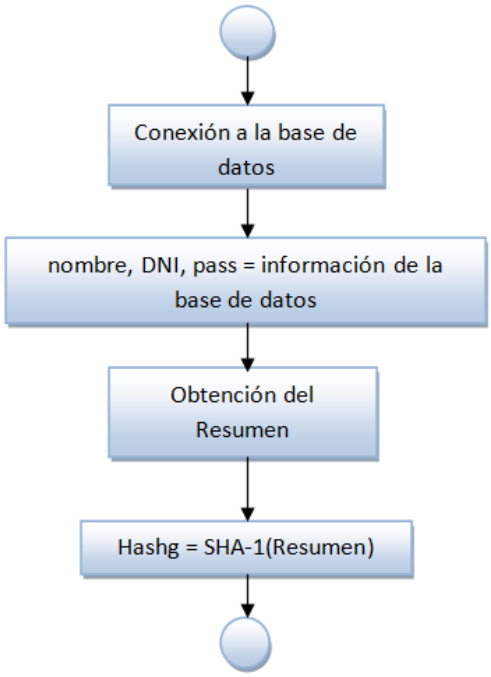


Acceso a la base de datos

Funcionamiento:

El servicio php realiza una conexión MySql a la base de datos y pide los usuarios. De la información obtenida se guardan el nombre, DNI y contraseña. Posteriormente se manipulan dicha información para obtener el mismo resumen que obtuvimos con el DNIe y ha este resumen sumado a la clave obtendremos el hash cifrado en SHA-1.

Diagrama de flujo:

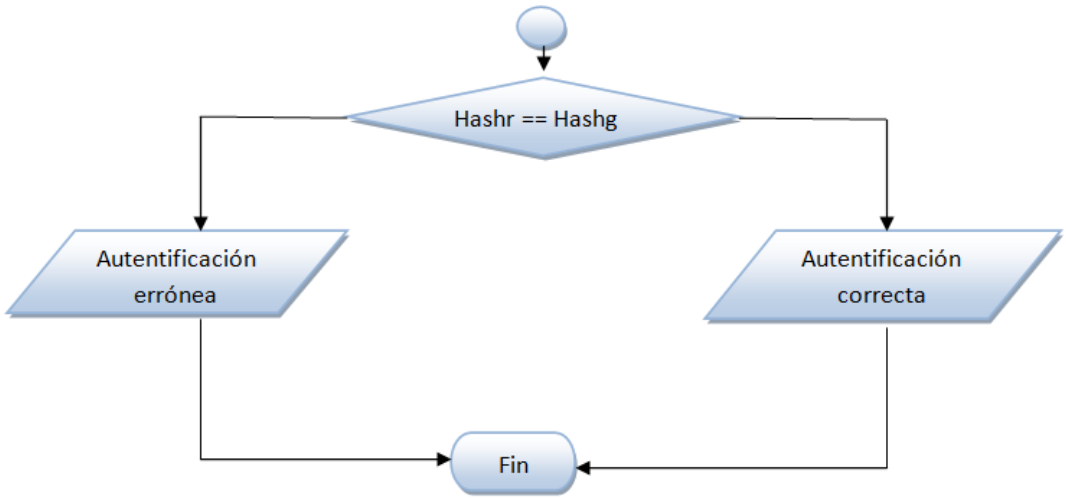


Comprobación de datos

Funcionamiento:

Comparamos los hash obtenidos: hashr y hashg en caso de no ser iguales dará fallo en la autentificación. En caso de si serlos, será una autentificación correcta.

Diagrama de flujo:



Cronograma

Primera sesión: familiarización con el código cliente y obtención de nombre, apellidos y DNI del DNIe

Segunda sesión: instalación de Wamp, funcionamiento del código php entregado en la práctica y creación de la base de datos.

Tercera sesión: métodos de generación del hash y cifrado.

Cuarta sesión: Creación del fichero autenticamac.php y su código.

Quinta sesión: Código de la conexión http con el servidor y depuración del código autenticamac.php

Sexta sesión: Comentar el código y generar la documentación.