Servicio básico de autenticación basado en datos firmados con DNIe

Practica 4

Emilio Sánchez

11/05/2016

Índice

Aplicación Cliente

El programa cliente consta de 3 clases:

- FirmarDatos: Clase cuya función es acceder y generar una firma con el nombre de usuario, el DNI, la fecha y la clave.
- Autentica: Clase cuya función es realizar la conexión con él servidor.
- Main: Es la función principal del programa.

FirmarDatos.java

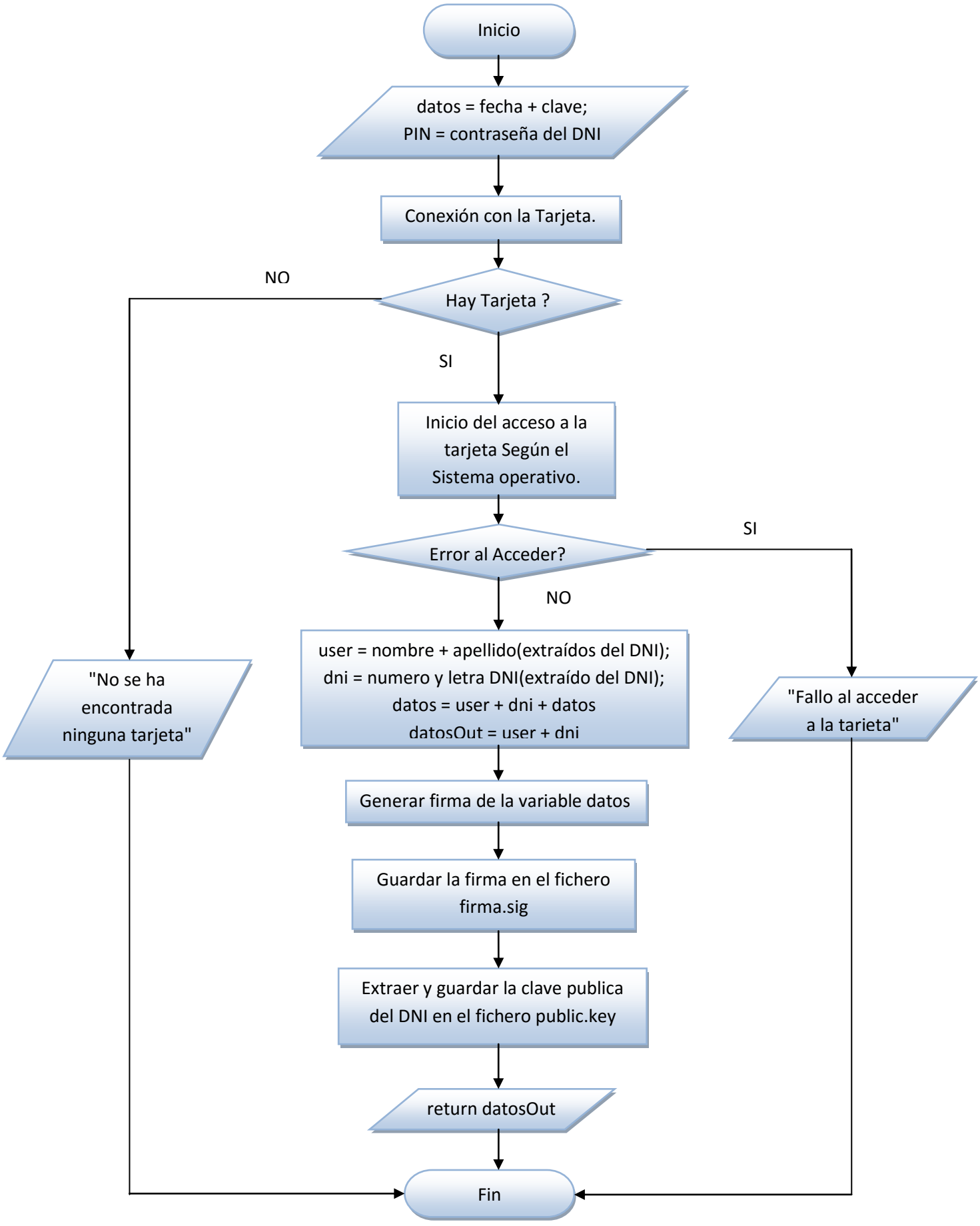
Métodos:

firmarDatos:

Descripción:

- Se crea una conexión con la tarjeta con el método conexionTarjeta.
- Comprueba cual es el sistema operativo que se está usando, para elegir el correcto archivo de configuración PKCS11 y se añade al proveedor de seguridad.
- Accede a los certificados del DNI y obtenemos la clave privada.
- Extrae los datos del DNI, usuario y número del DNI, se pasan a minúscula, se almacenan para los datos de salida y los añadimos a los datos a firmar junto con la clave y la fecha.
- Firmamos los datos mencionados anteriormente.
- Guardamos la firma generada en el fichero firma.sig.
- Guardamos la clave publica del DNI en el fichero public.key.

Diagrama de flujo:


Inicio
datos = fecha + clave;
PIN = contraseña del DNI
Conexión con la Tarjeta.
NO
Hay Tarjeta ?
SI
Inicio del acceso a la tarjeta Según el Sistema operativo.
Error al Acceder?
SI
NO
"No se ha encontrada ninguna tarjeta"
user = nombre + apellido(extraídos del DNI);
dni = numero y letra DNI(extraído del DNI);
datos = user + dni + datos
datosOut = user + dni
"Fallo al acceder a la tarieta"
Generar firma de la variable datos
Guardar la firma en el fichero firma.sig
Extraer y guardar la clave publica del DNI en el fichero public.key
return datosOut
Fin

Forma de uso:

#FirmarDatos varObj = new FirmarDatos();

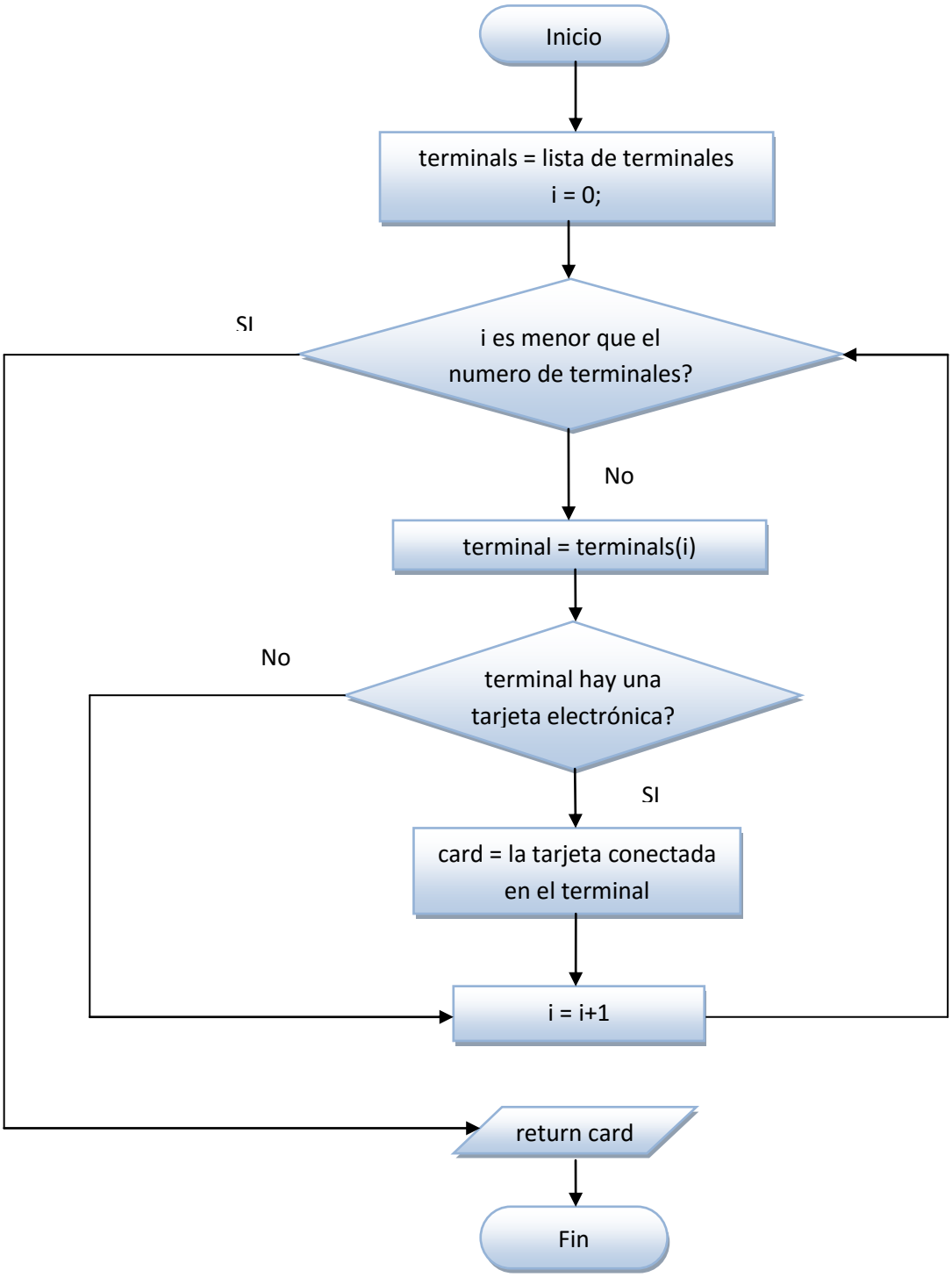
#String [] var = varObj.firmarDatos (PIN_del_dni, fecha_formato_String+clave_formato_String);

conexionTarjeta:

Descripción:

- Crea un objeto terminal con la lista de los terminales del ordenador
- Revisa los toda la lista en busca de una tarjeta electrónica.

Diagrama de flujo:



Forma de uso:

// Clase privada solo se puede usar en la misma clase java FiramarDatos.java

Card c = conexionTarjeta();

compruebaFirma

Descripción:

- Recibe como parámetros la firma, los datos a los que se le ha realizado la firma y la clave publica.
- Comprueba que la firma es correcta. Devuelve true en caso de serlo y false en caso de que no

Forma de uso:

compruebaFirma(user + dni + date + clave, firma, clavepublica);

Autentica.java

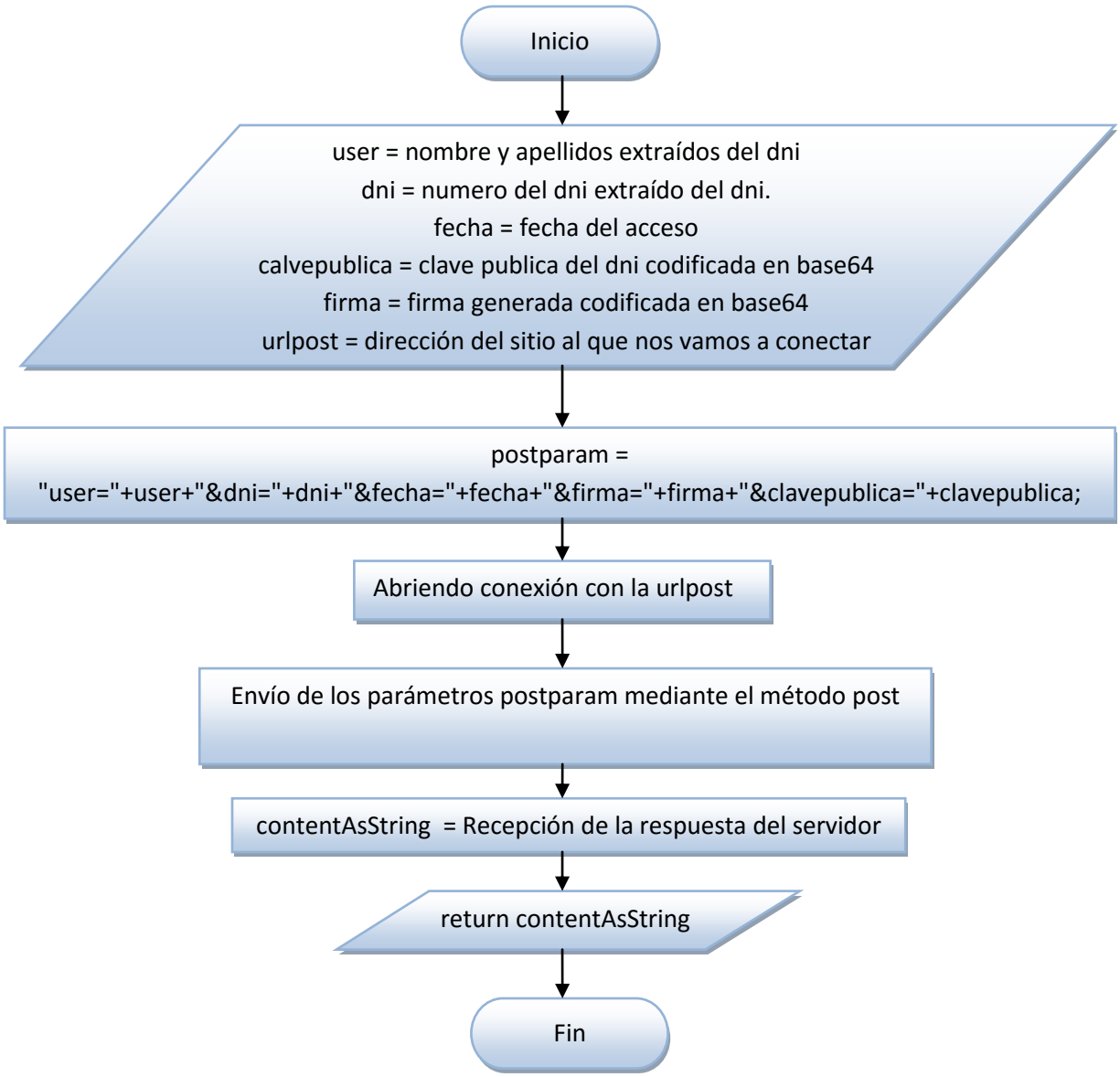
Métodos:

enviarCredencialesPost

Descripción:

- Establece la conexión con el servidor indicado en la variable urlpost.
- Envía lo parámetros mediante el método post de http (fecha, user, dni, firma y clave publica).
- Recibe la respuesta del servidor.

Diagrama de flujo:



Forma de uso:

Autentica varObj = new Autentica();

String respuesta = varObj. enviarCredencialesPost(url, nombre_y_apellido, dni, fecha, firma, clave_publica);

Main.java

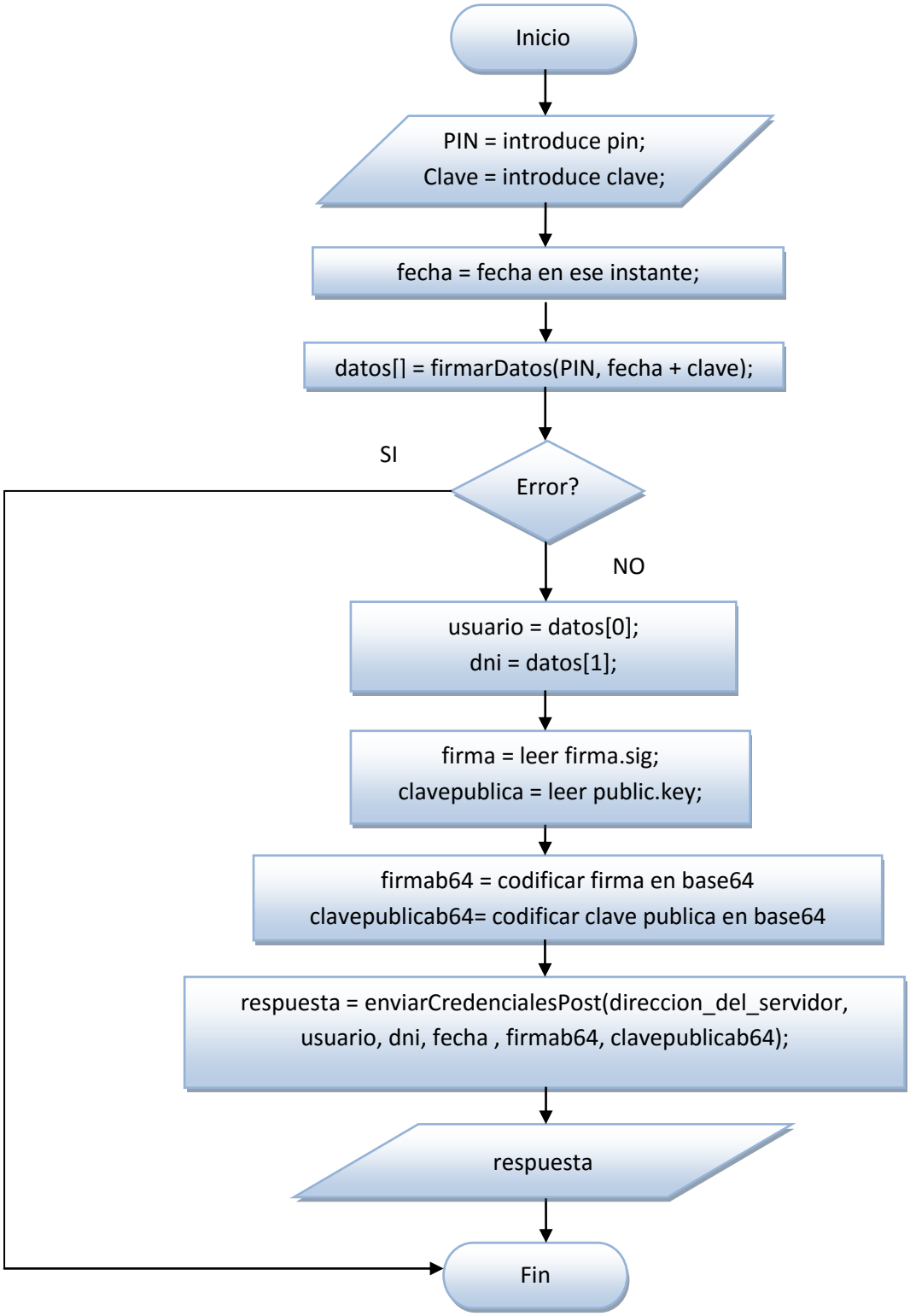
Métodos:

main

Descripción:

- Pide al usuario el pin del dni y la clave.
- Coge la fecha actual.
- firma los datos haciendo uso del método firmarDatos.
- lee la firma guardada en el fichero firma.sig y la codifica en base64
- lee la clave publica guardada en el fichero public.key y la codifica en base64
- establece la conexión pasando los parámetros recogidos

Diagrama de flujo:



Forma de uso:

Es la función principal que arranca el proyecto

Aplicación Servidor

El programa servidor es un servlet en java que consta de dos clases:

- DniDatabase: clase que realiza la conexión a la base de datos.
- autentica: clase que recibe la respuesta del cliente la procesa, verifica la firma y genera una respuesta.

DniDatabase.java

Atributos:

- String driver = "com.mysql.jdbc.Driver"; // driver mysql
- String url = "jdbc:mysql://localhost:3306/dniauth"; // dirección de la base de datos
- Connection conn; // objeto que establece la conexión.

Métodos:

cargarDriver

Descripción:

- Crea una nueva instancia del driver de la librearía MySql

Forma de uso:
Es un método privado por lo que solo se puede usar en la misma clase.

#cargaDriver();

getConnection

Descripción:

- abre una conexión con la base de datos indicada previamente

Forma de uso:
Es un método privado por lo que solo se puede usar en la misma clase.

#getConnection();

//para cerrar la conexión usaremos la variable conn para cerrarla.

#conn.close();

cogerClave

Descripción:

- Se le pasa el parámetro dni al método y en función a él genera una sentencia sql.
- Se ejecuta la sentencia sql, la cual devuelve la clave de usuario del dni introducido.

- Finalmente devuelve dicha clave.

Forma de uso:

DniDatabase db = new DniDatabase();

String var = db.cogerClave(Dni);

autentica.java

hereda de la clase HttpServlet.

Atributos:

- byte clavepublica[] ; // clave publica del dni.
- String user ; // nombre + apellidos.
- String dni ; // numero del dni.
- String date ; // fecha en la que se realiza la conexión.
- String clave ; // clave del usuario.
- byte firma[] ; // firma de los datos enviados
- final DniDatabase db = new DniDatabase();
- String firmab64 ; // firma de los datos enviados codificados en base64.
- String clavepublicab64; // clave publica del dni codificada en base64.

Métodos:

doPost

Descripción:

- Recibe todas las variables y las guarda en los atributos de la clase.
- Decodifica la firmab64 y la clavepublicab64 en base 64.
- Comprueba la que la firma sea correcta.
- En caso de que la firma sea correcta genera el mensaje de respuesta autentificación correcta.
- En caso contrario genera el mensaje de respuesta autentificación incorrecta.

Forma de uso:

Es un metodo sobrescrito de la clase servlet que se encuentra a la espera de recibir un mensaje del tipo post.

processRequestOK

Descripción:

responde con un mensaje de Autentificación correcta.

Forma de uso:

processRequestOK(request, response);

processRequestER

Descripción:

- responde con un mensaje de Autentificación incorrecto.

Forma de uso:

processRequestOK(request, response);

compruebaFirma

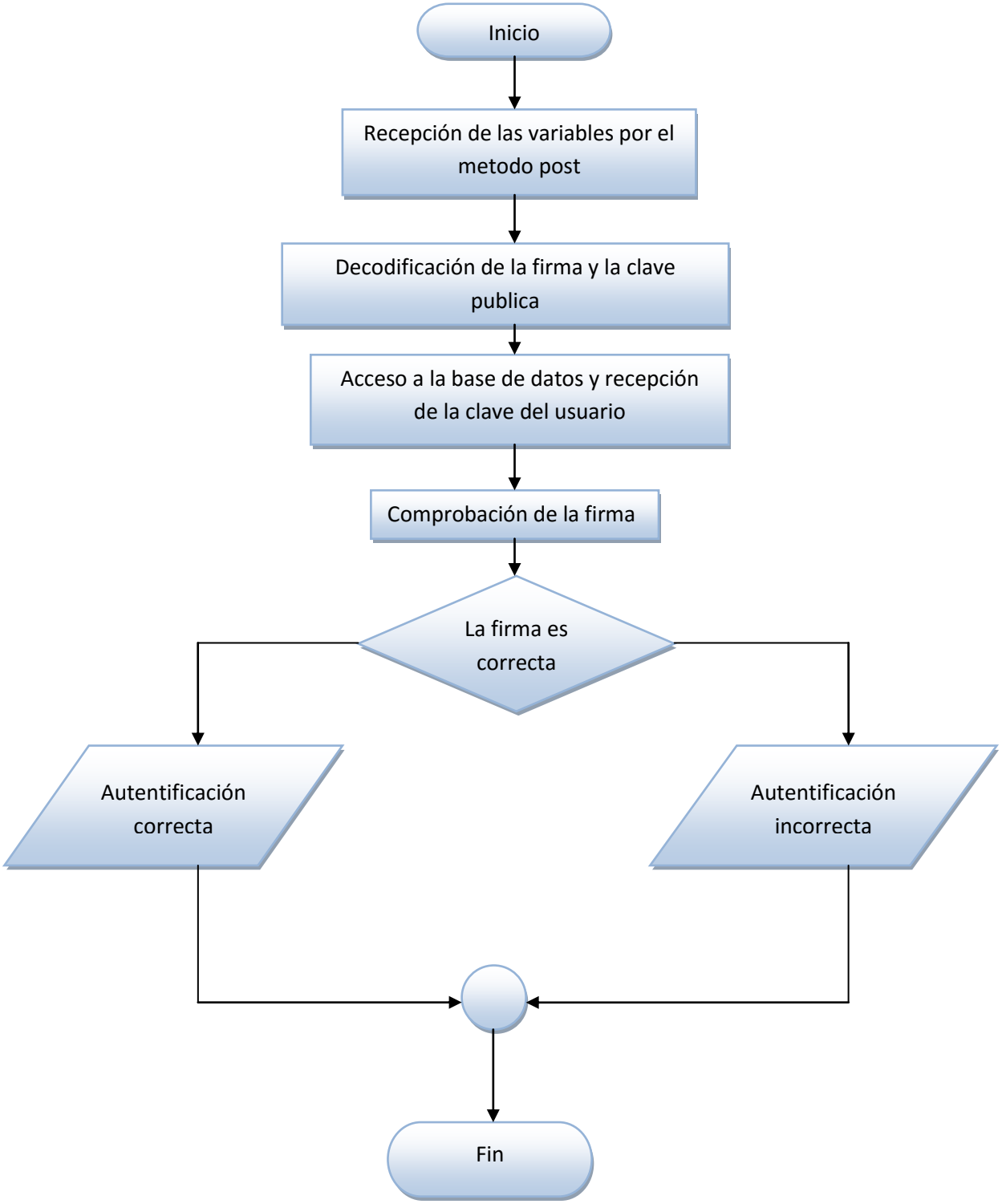
Descripción:

- Recibe como parámetros la firma los datos recibidos y la clave publica.
- Comprueba que la firma es correcta. Devuelve true en caso de serlo y false en caso de que no

Forma de uso:

compruebaFirma(user + dni + date + clave, firma, clavepublica);

Diagrama de flujo:


Inicio
Recepción de las variables por el metodo post
Decodificación de la firma y la clave publica
Acceso a la base de datos y recepción de la clave del usuario
Comprobación de la firma
La firma es correcta
Autentificación correcta
Autentificación incorrecta
Fin

Cronograma

Primera sesión y segunda sesión: Creación del proyecto e instalación de los programas y configuraciones necesarias para su funcionamiento.

Tercera sesión: Familiarización con el código y el objetivo de la practica

Cuarta sesión: Búsqueda de los campos del dni, cambio del contenido al cual le generaba la firma, adaptación del envío de credenciales mediante Post, creación del código para el acceso a la base de datos y el código de recepción de parámetros. También se ha verificado la firma y generado los códigos de respuesta en función de si la firma es correcta o no.

Quinta sesión: Depuración y cementación del código, y creación de la memoria