Programación de

Servicios y Procesos

CLIENTE-SERVIDOR

Diego González Martínez | 8/02/2023

# Ejercicio 7

**¿De qué trata el Ejercicio 7?**

De igual manera a lo visto en el tema, ahora te proponemos un ejercicio que genere una cadena de texto y la deje almacenada en un fichero encriptado, en la raíz del proyecto hayas creado, con el nombre fichero.cifrado.

Para encriptar el fichero, utilizarás el algoritmo Rijndael o AES, con las especificaciones de modo y relleno siguientes: Rijndael/ECB/PKCS5Padding.

La clave, la debes generar de la siguiente forma:

* A partir de un número aleatorio con semilla la cadena del nombre de usuario + password.
* Con una longitud o tamaño 128 bits.

**¿De qué trata?**

El código que he hecho es un programa de Java que se encarga de cifrar y descifrar archivos. La ejecución del programa comienza preguntando por el usuario y la contraseña. Si la combinación es correcta (usuario: "diego", contraseña: "1234"), el programa genera una clave de cifrado a partir de la concatenación de usuario y contraseña. Luego, el programa cifra un archivo llamado "fichero.txt" y guarda el resultado en un archivo llamado "fichero.txt.cifrado". Finalmente, el programa descifra el archivo cifrado y muestra su contenido en pantalla.

El cifrado y descifrado se realiza usando el algoritmo AES (Advanced Encryption Standard) en modo ECB (Electronic Codebook) y relleno PKCS5.

**¿Cuál es la funcionalidad?**

El código es un programa de Java que realiza la encriptación de un archivo de texto con el algoritmo AES (Advanced Encryption Standard). Antes de realizar la encriptación, el usuario debe proporcionar un nombre de usuario y una contraseña. Si el nombre de usuario es "diego" y la contraseña es "1234", se procede a realizar la encriptación y descifrado del archivo. Finalmente, el contenido del archivo descifrado se muestra en la consola.

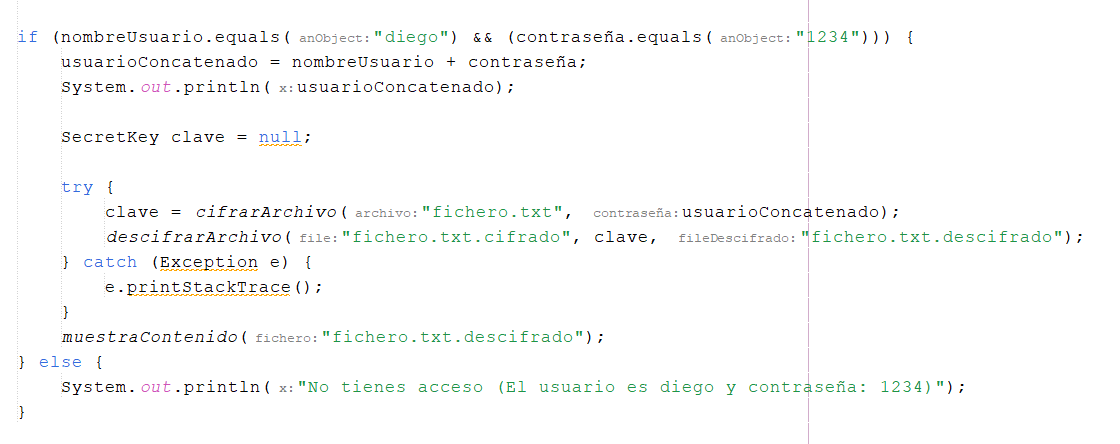
**¿Qué hace paso por paso?**

Este código Java hace lo siguiente:

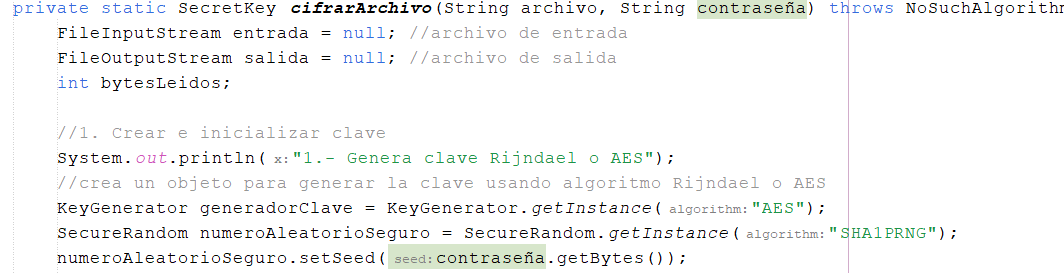
1. Pide al usuario que indique su nombre de usuario y contraseña.
2. Verifica si el nombre de usuario es "diego" y la contraseña es "1234". Si la verificación es correcta, concatena el nombre de usuario y la contraseña en una cadena.
3. Usa el algoritmo AES para generar una clave secreta a partir de la cadena obtenida en el paso 2.
4. Usa el cifrado AES para cifrar un archivo "fichero.txt".
5. Usa la clave secreta para descifrar el archivo cifrado y guardar el resultado en "fichero.txt.descifrado".
6. Muestra el contenido del archivo descifrado "fichero.txt.descifrado".
7. Si la verificación del nombre de usuario y contraseña en el paso 2 falla, muestra un mensaje que indica que no se tiene acceso.

**Código explicado paso a paso:**

El cliente pide el usuario y contraseña. Luego, verifica si los datos ingresados son "diego" y "1234". Si los datos son correctos, se concatena el nombre de usuario y la contraseña, se utiliza para cifrar un archivo "fichero.txt" y luego descifrarlo. Finalmente, se muestra el contenido del archivo descifrado. Si los datos ingresados no son correctos, se imprime un mensaje indicando que el usuario no tiene acceso.

Se inicializa un objeto KeyGenerator para generar una clave de cifrado AES. Luego, se crea un objeto SecureRandom para generar números aleatorios seguros, y se establece la semilla con los bytes de la contraseña dada.

Finalmente, la clave se genera inicializando el KeyGenerator con un tamaño de 128 bits y el objeto SecureRandom.

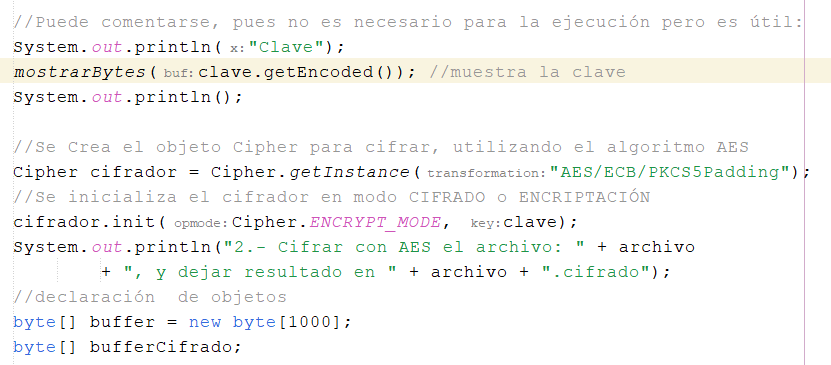
****

Se genera una clave AES utilizando un KeyGenerator con una semilla basada en la contraseña proporcionada.

Luego, se crea un objeto Cipher utilizando el algoritmo AES/ECB/PKCS5Padding, que representa el cifrador que se utilizará para encriptar el archivo.

Después, se inicializa el cifrador en modo de encriptación (Cipher.ENCRYPT\_MODE) utilizando la clave AES generada previamente.

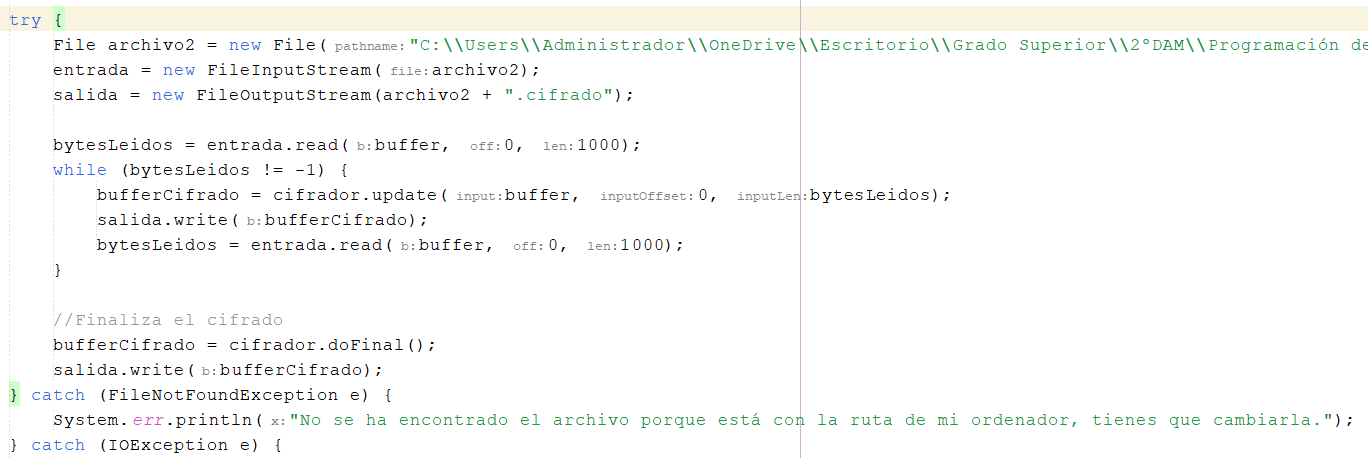
Se declaran dos arrays de bytes, un buffer y un bufferCifrado, que se utilizarán para leer y escribir los datos del archivo cifrado.

****

Creo dos objetos de tipo FileInputStream y FileOutputStream para leer y escribir en un archivo de texto, respectivamente. El archivo a cifrar se encuentra en la ruta "C:\Users\Administrador\OneDrive\Escritorio\Grado Superior\2ºDAM\Programación de Servicios y Procesos\Tema 7\PSP\_07\fichero.txt".

Utilizo un bucle while para leer el archivo de texto en bloques de 1000 bytes y cifrar cada bloque usando el objeto Cipher inicializado previamente.

El cifrado de cada bloque se guarda en un archivo de salida con nombre "fichero.txt.cifrado".

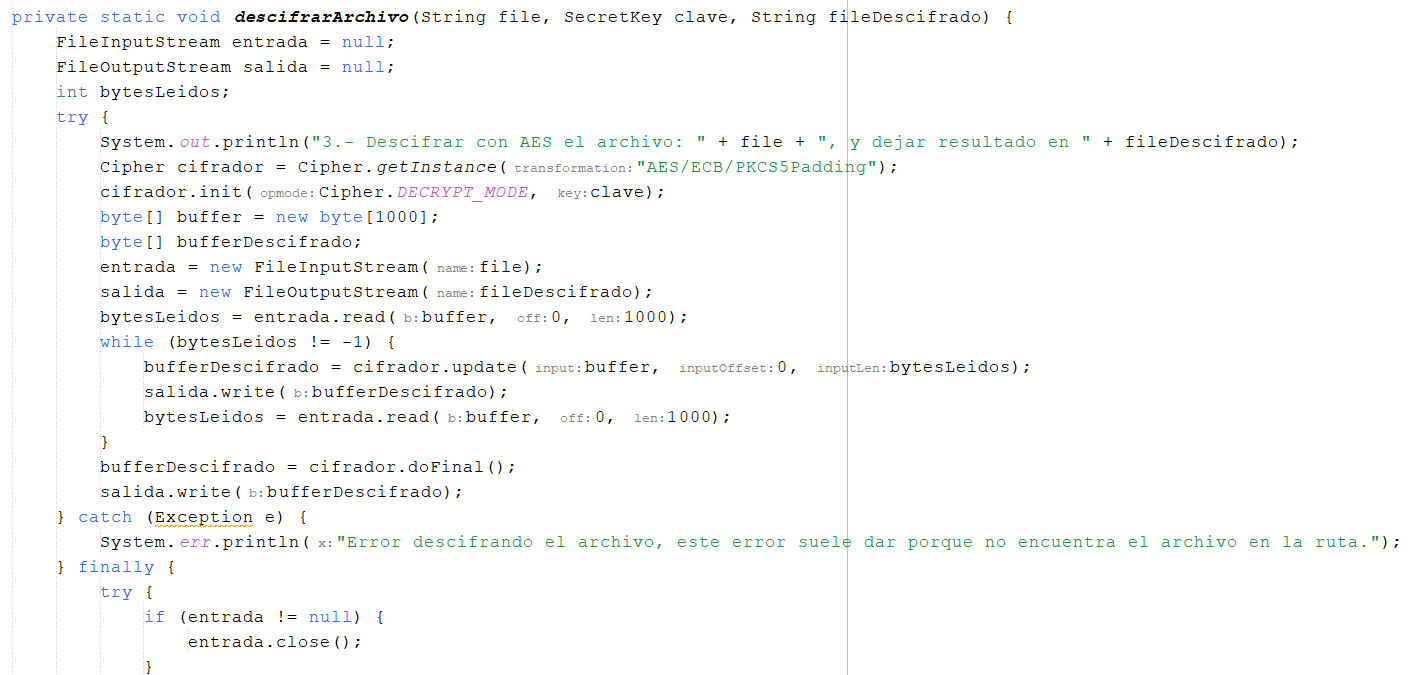
****

Esta parte del código describe un método llamado "descifrarArchivo" que descifra un archivo cifrado en AES y escribe el resultado en un archivo nuevo. Utiliza una clave secreta especificada para el descifrado y maneja posibles excepciones que pueden ocurrir durante el proceso. Finalmente, se asegura de cerrar los streams de entrada y salida.

El método llamado "descifrarArchivo" comienza creando un objeto Cipher y configurándolo para descifrar con AES utilizando el modo ECB y el padding PKCS5. Luego, abre dos streams, uno de entrada y otro de salida, con los nombres especificados en los parámetros de entrada.

A continuación, se leen bloques de 1000 bytes del archivo de entrada y se descifran con el método "update" de Cipher. Los datos descifrados se escriben en el archivo de salida. Finalmente, se ejecuta el método "doFinal" para completar el descifrado y escribir cualquier byte que quedó sin escribir.

Si ocurre algún error durante la ejecución, se imprimirá un mensaje de error en la consola. Finalmente, se cierran los streams de entrada y salida para liberar recursos y evitar problemas.

****

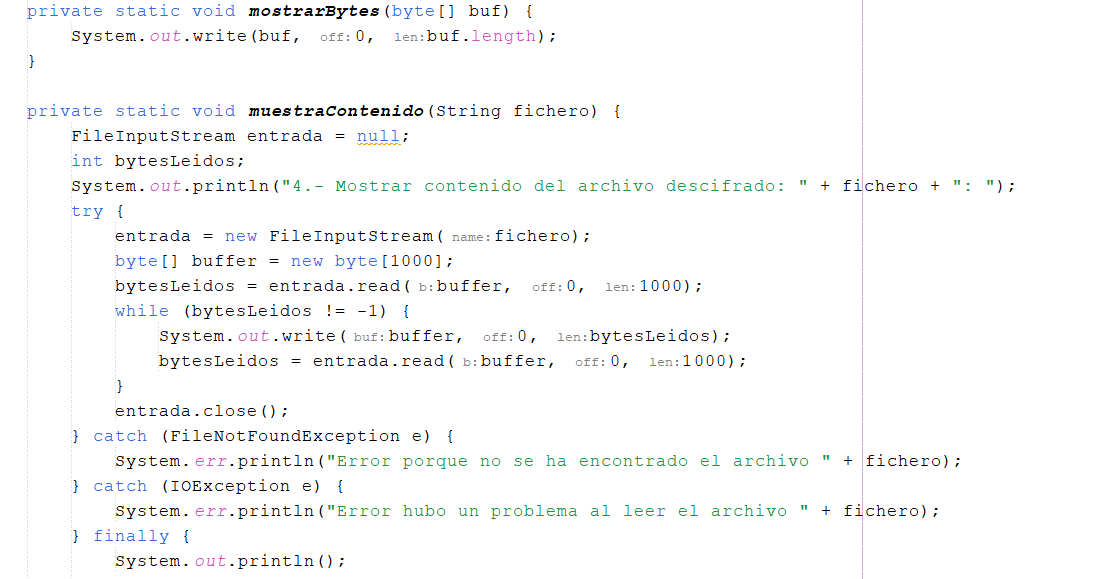
Por último, el método "mostrarBytes(byte[] buf)" imprime el contenido de un array de bytes en la salida estándar.

El método "muestraContenido(String fichero)" muestra el contenido de un archivo especificado por el parámetro "fichero".

Para leer el contenido del archivo, se crea un objeto de la clase FileInputStream, se leen los datos en bloques de tamaño 1000 bytes y se escriben en la salida estándar.

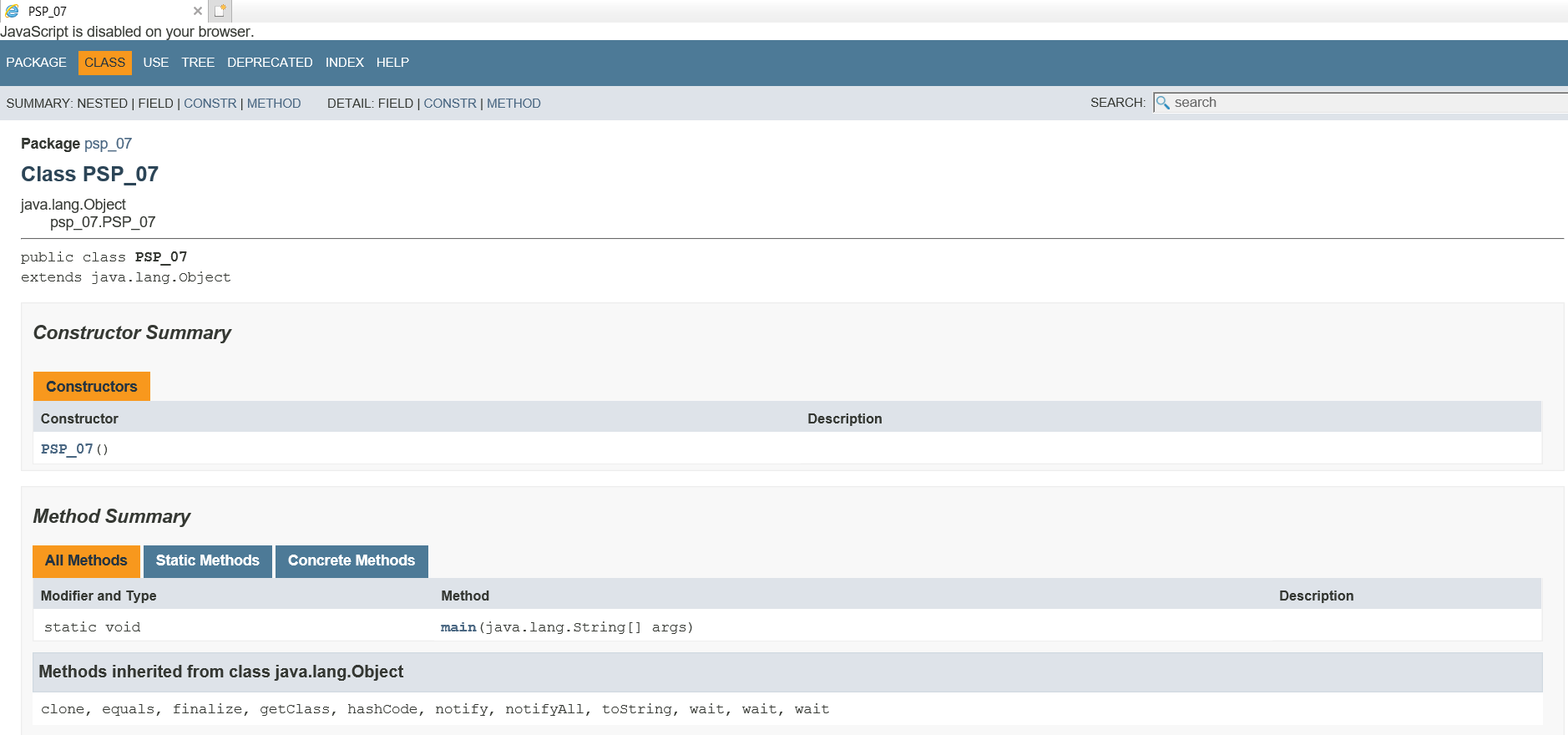
Si ocurre algún error durante la ejecución, se imprimirá un mensaje de error en la consola.

Finalmente, se cierra el objeto FileInputStream y se imprime un salto de línea.

****

**JAVADOC:**

**C:\Tema 7\PSP\_07\dist\javadoc**

****