Alumno: Kevin Zamora Amela

Tarea: PSP07

Asignatura: Programación de Servicios y Procesos

Ejercicio 1.

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.

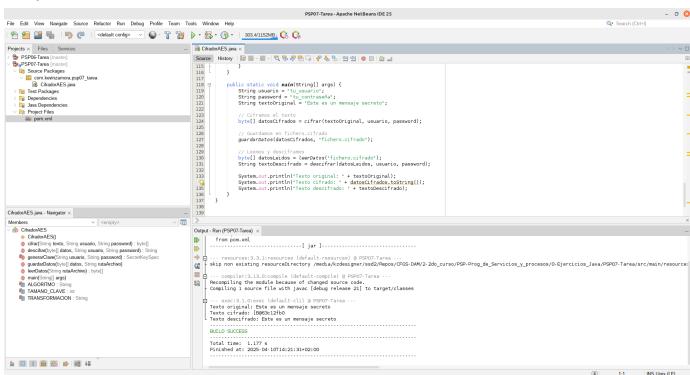
Para encriptar el fichero, utilizarás el algoritmo Rijndael o AES, con las especificaciones de modo y relleno siguientes: Rijndael/ECB/PKCS5Padding.

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.

Para probar el funcionamiento, el mismo programa debe acceder al fichero encriptado para desencriptarlo e imprimir su contenido.

Respuesta:



PSP04 Kevin Zamora

Validación del funcionamiento requerido:

```
Output-Run (PSP07-Tarea) ×

from pom.xml

| jar | ...
| resources:3,3,1:resources (default-resources) @ PSP07-Tarea ...
| skip non existing resourceDirectory /media/kzdesigner/ssd2/Repos/CFGS-DAM/2-2do_curso/PSP-Prog_de_Servicios_y_procesos/0-Ejercicios_Java/PSP07-Tarea/src/main/resources/
| compiler3.13.0:compile de_Servicios_y_procesos/0-Ejercicios_Java/PSP07-Tarea/src/main/resources/
| compiler3.13.0:compile de_Servicios_y_procesos/0-Ejercicios_Java/PSP07-Tarea/src/main/resources/
| compiler3.13.0:compile de_Servicios_y_procesos/0-Ejercicios_Java/PSP07-Tarea/src/main/resources/
| compiler3.13.0:compile de_Servicios_y_procesos/0-Ejercicios_Java/PSP07-Tarea/src/main/re
```

Programa/Aplicación implementado/a:

```
* Clase que maneia las operaciones de cifrado/descifrado usando AES
  public class CifradorAES {
      private static final String ALGORITMO = "AES";
   private static final String TRANSFORMACION = "AES/ECB/PKCSSPadding";
      private static final int TAMANO CLAVE = 128; // bits
public static void main(String[] args) {
          String usuario = "tu_usuario";
          String password = "tu contraseña";
          String textoOriginal = "Este es un mensaje secreto";
          // Ciframos el texto
          byte[] datosCifrados = cifrar(textoOriginal, usuario, password);
          // Guardamos en fichero.cifrado
          guardarDatos(datosCifrados, "fichero.cifrado");
          // Leemos y desciframos
          byte[] datosLeidos = leerDatos("fichero.cifrado");
          String textoDescifrado = descifrar(datosLeidos, usuario, password);
          System.out.println("Texto original: " + textoOriginal);
          System.out.println("Texto cifrado: " + datosCifrados.toString());
          System.out.println("Texto descifrado: " + textoDescifrado);
```

Importaciones de los paquetes o bibliotecas necesarias:

```
package com.kevinzamora.psp07_tarea;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

Método para generar "clave" fija
Bibliotecas de métodos de cifrado
```

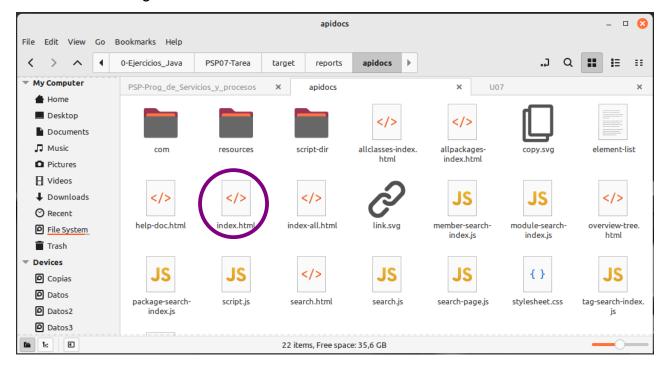
PSP04 Kevin Zamora

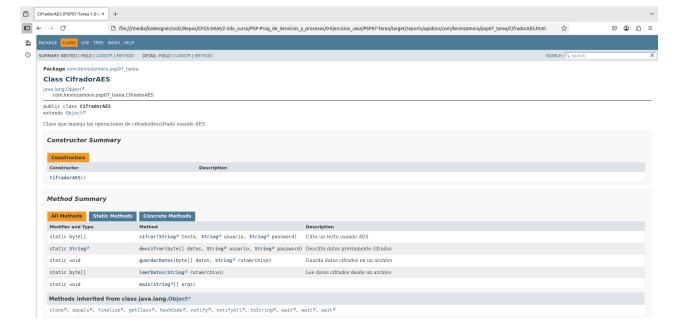
Métodos implementados:

```
* Cifra un texto usando AES
       * @param texto Texto a cifrar
       * @param usuario Usuario para generar la clave
       * @param password Password para generar la clave
       * @return Datos cifrados
      public static byte[] cifrar(String texto, String usuario, String password) {
曱
          try { ◀
                                                                          Control de excepciones
              SecretKeySpec clave = generarClave(usuario, password);
              Cipher cipher = Cipher.getInstance(TRANSFORMACION);
              cipher.init(Cipher.ENCRYPT_MODE, clave);
              return cipher.doFinal(texto.getBytes(StandardCharsets.UTF 8));
                                                                          Control de excepciones
          } catch (Exception e) {
              throw new RuntimeException("Error al cifrar", e);
豆
       * Descifra datos previamente cifrados
       * @param datos Datos cifrados
       * @param usuario Usuario usado para generar la clave
       * @param password Password usada para generar la clave
       * @return Texto descifrado
       */
早早
      public static String descifrar(byte[] datos, String usuario, String password) {
          try {
              SecretKeySpec clave = generarClave(usuario, password);
              Cipher cipher = Cipher.getInstance(TRANSFORMACION);
              cipher.init(Cipher.DECRYPT_MODE, clave);
              return new String(cipher.doFinal(datos), StandardCharsets.UTF 8);
          } catch (Exception e) {
              // throw new RuntimeException("Error al descifrar", e);
              e.printStackTrace(); // Imprime el error original
              throw new RuntimeException("Error al descifrar", e);
      }
       * Guarda datos cifrados en un archivo
       * @param datos Datos cifrados
       * @param rutaArchivo Buta del archivo
       public static void guardarDatos(byte[] datos, String rutaArchivo) {
          try (FileOutputStream fos = new FileOutputStream(rutaArchivo)) {
               fos.write(datos);
          } catch (IOException e) {
               throw new RuntimeException("Error al guardar el archivo", e);
      }
       * Lee datos cifrados desde un archivo
       * @param rutaArchivo Buta del archivo
        * @return Datos leídos
早早
       public static byte[] leerDatos(String rutaArchivo) {
           try (FileInputStream fis = new FileInputStream(rutaArchivo)) {
               byte[] datos = fis.readAllBytes();
               return datos;
           } catch (IOException e) {
               throw new RuntimeException("Error al leer el archivo", e);
                                                                                  Kevin Zamora
```

Comentarios relacionados con el proceso de generación de la documentación con JavaDoc del proyecto resultante:

- Primeramente y como ya deberíamos saber, para ejecutar la herramienta JavaDoc y proceder a generar la documentación de nuestro proyecto, debemos hacer clic derecho sobre el directorio raíz de nuestro proyecto con java y seleccionar la opción "Generar JavaDoc/Generate JavaDoc", según tengamos en Español o en Inglés, respectivamente, la interfaz de nuestro IDE Netbeans.
- Una vez realizado el citado proceso, accedemos a la carpeta "target" (ubicada en el directorio raíz de nuestro proyecto). Y tras el citado paso, accedemos a las carpetas anidadas "reports" y "apidocs". Al acceder a la última carpeta, nos apareceran una serie de carpetas y de archivos, aunque deberemos enfocarnos en el archivo "index.html" y abrirlo en un navegador web.





PSP04 Kevin Zamora