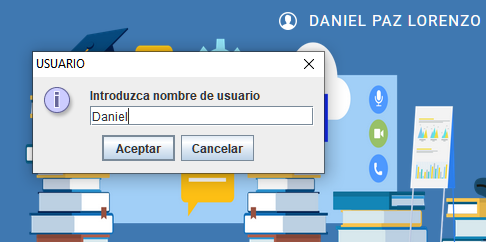
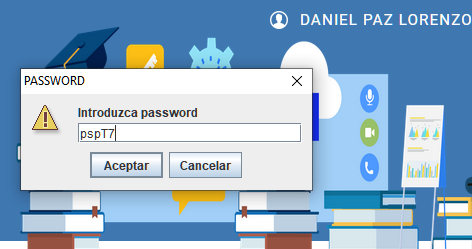
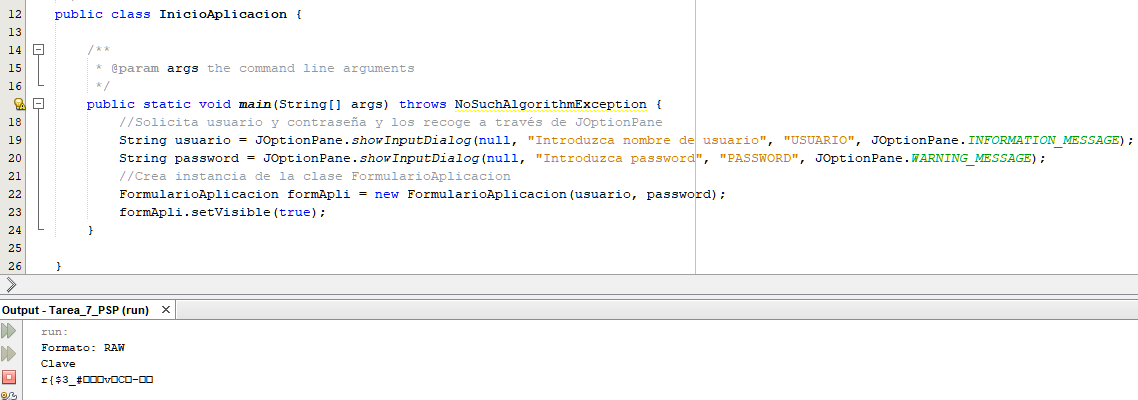
**EJERCICIO 1**

En esta tarea se nos pide que cifremos un texto dado con algoritmo "AES/ECB/PKCS5Padding" y con una clave de número aleatorio con semilla la cadena (usuario + password) y de longitud 128 bits.

Para realizar esta tarea he decidido realizar una aplicación con interfaz gráfica y así poder interactuar con el usuario de una manera sencilla y visual. A grandes rasgos, la aplicación solicitará al usuario un nombre y contraseña, luego mostrará un formulario con distintas opciones: introducir texto, guardar fichero encriptado, mostrar fichero encriptado y mostrar fichero desencriptado.

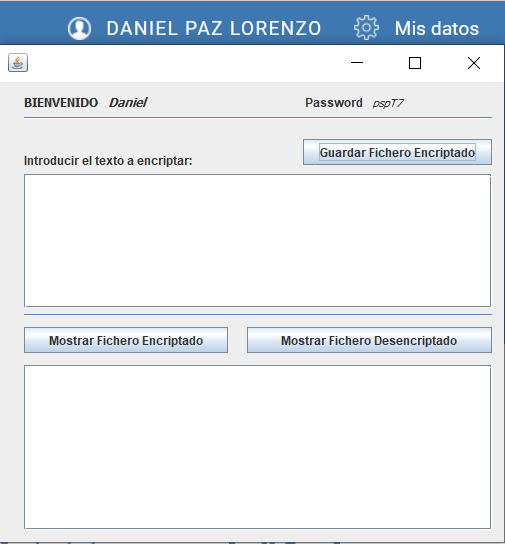
La clave va a ser privada y la misma tanto para encriptar como para desencriptar, es decir vamos a realizar una encriptación simétrica en este caso con algoritmo AES como se nos pide en la tarea.

A continuación, voy a mostrar el funcionamiento de mi aplicación poniendo como ejemplo una ejecución tipo y explicando cada uno de los pasos:

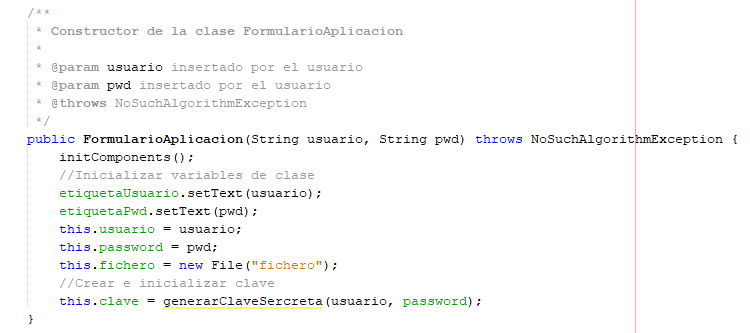
1. Se nos va a solicitar usuario y contraseña mediante 2 cuadros de diálogo “JOptionPane”, una vez introducidas serán recogidas en ambas variables guardadas para tal fin:

*Este es el código de la clase que inicia la aplicación y en ejecución podemos ver que la salida nos muestra ya la clave encriptada*

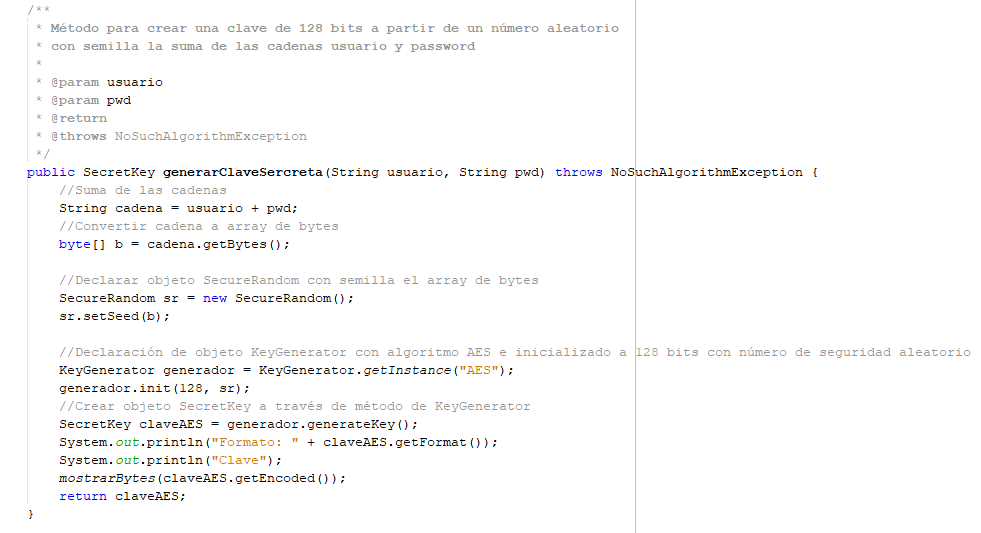
1. Una vez recogidos los datos y generada ya la clave se muestra el formulario principal de la aplicación con todas las opciones disponibles:



*Pantalla de opciones de la aplicación*

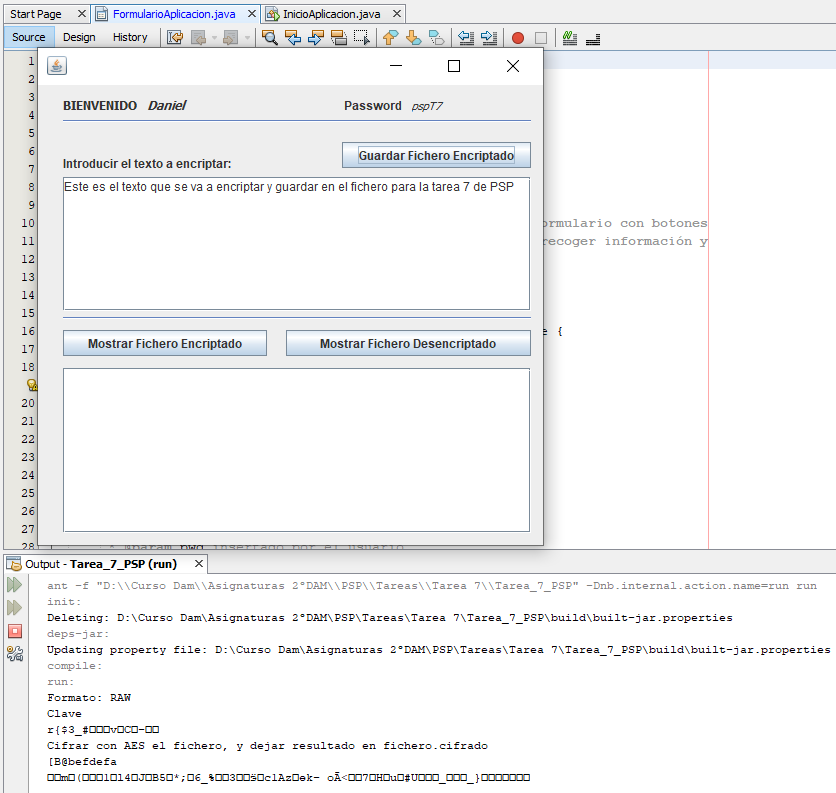


*Código del constructor que genera la pantalla de opciones cuando instanciamos la clase*

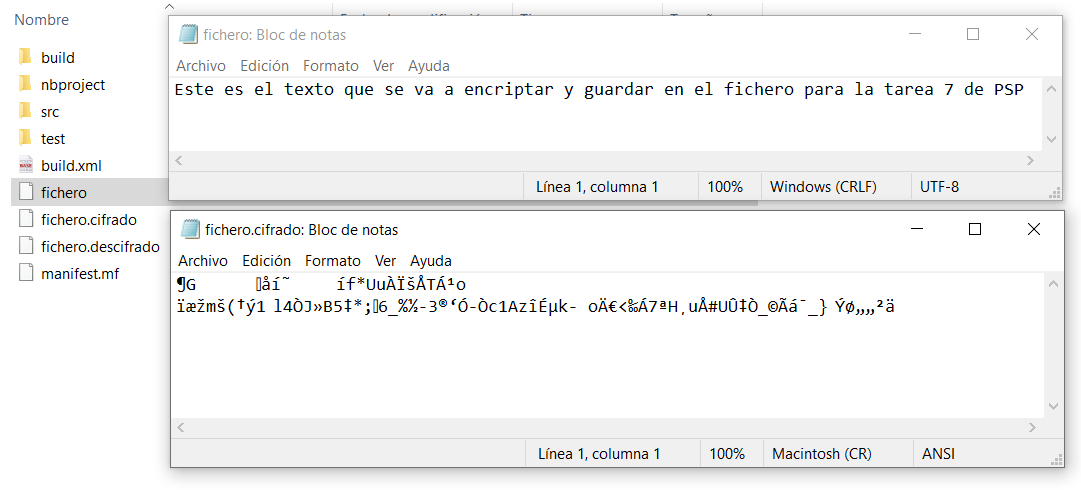


*Código para generar la clave de 128 bits, encriptada con el algoritmo AES y configurada a partir de un número aleatorio con semilla la cadena del nombre de usuario + password*

1. Escribimos en el recuadro el texto que queremos encriptar y cuando ya lo tengamos pulsamos el botón de “guardar fichero encriptado”:

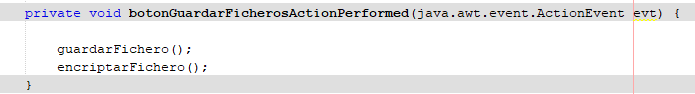


*Una vez se guarda el fichero encriptado muestra la clave y el texto, ya cifrados y un mensaje de verificación*

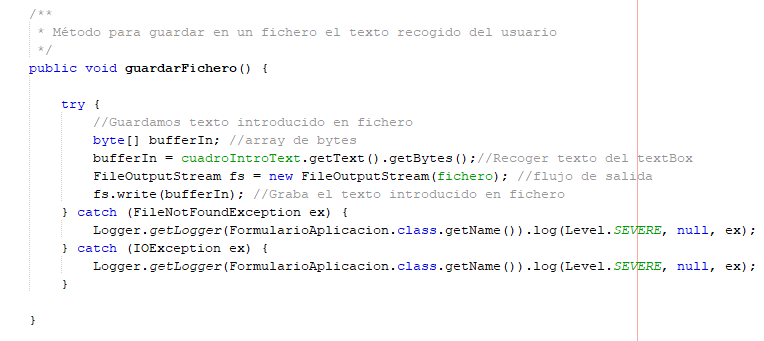


*Si abrimos la carpeta del proyecto podemos ver que se nos han creado los 2 ficheros y si abrimos cada uno de ellos veremos su contenido*

Cuando nosotros pulsamos el botón se genera un evento y le diremos que encripte el texto, con el algoritmo “Rijndael/ECB/PKCS5Padding” y posteriormente lo guarde en un fichero (fichero.cifrado). Para ello he creado 2 métodos:



*Recoge evento y llama a los métodos*

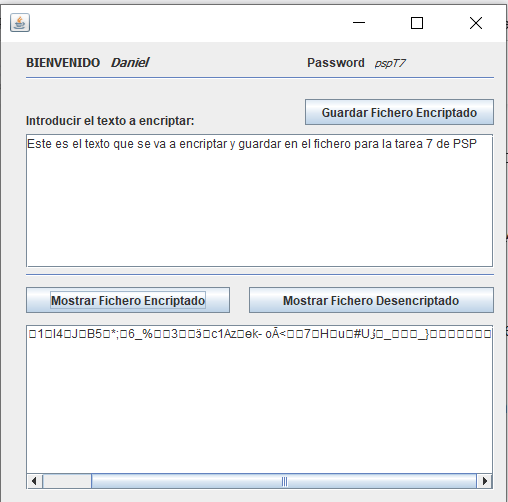


*Recoge el texto introducido y lo guarda en un fichero llamado “fichero” dentro del proyecto*



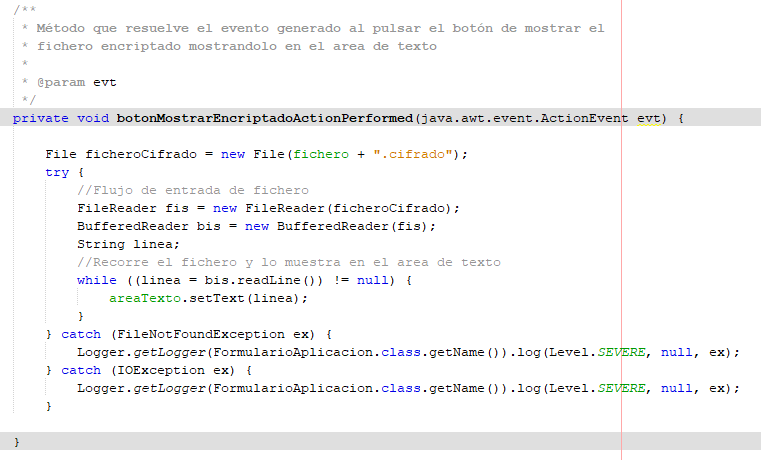
*Lee el fichero anterior, lo encripta y lo guarda en un fichero llamado “fichero.cifrado” dentro del proyecto*

1. A continuación, pulsamos en “mostrar fichero encriptado” lo que hará que se muestre en el área de texto el fichero cifrado anteriormente:

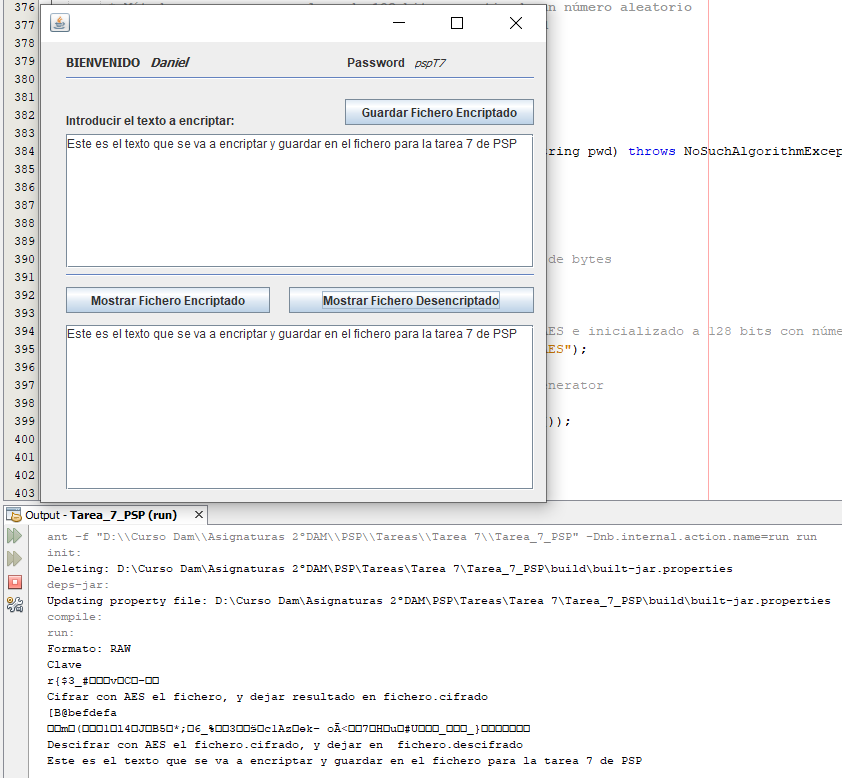


*En el cuadro inferior vemos como quedaría el texto cifrado con el algoritmo AES*

El código que responde al evento generado por el botón y que lee el fichero para mostrarlo al usuario sería el siguiente:

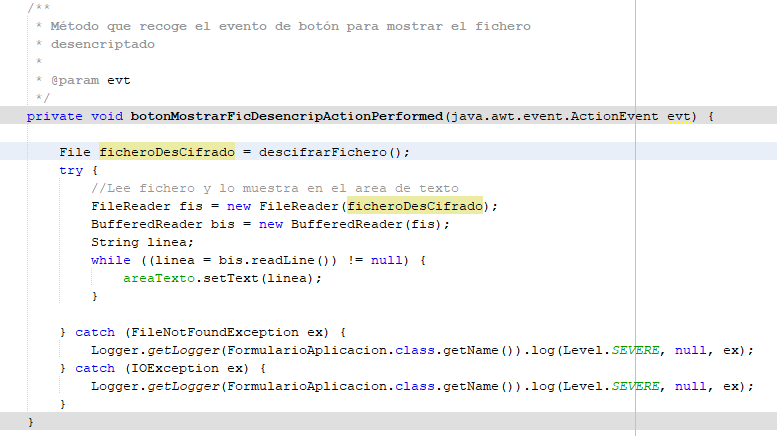


1. La última de las opciones sería desencriptar el fichero cifrado, guardarlo en “fichero.descifrado” dentro del proyecto y a mostrar, en el área de texto de la aplicación, el resultado. Todo ello lo haremos pulsando el botón de “Mostrar fichero desencriptado”:



*Observamos que el texto mostrado coincide con el inicial por lo que se ha desencriptado correctamente, a la vez muestra un mensaje de verificación en la salida por consola*

En el código que recoge el evento del botón, recogemos en un objeto File el fichero ya descifrado que nos devolverá el método “descifrarFichero()” que hemos creado para ello. Una vez lo tenemos recogido solo queda leerlo e ir mostrando al usuario el resultado por pantalla:



Este es el método creado para descifrar el fichero en el que le indicamos que fichero queremos desencriptar, con que algoritmo y cuál es la clave para hacerlo:

