

Módulo Profesional 09:
Programación de servicios y procesos
Actividad UF1

CICLO FORMATIVO DE GRADO SUPERIOR EN

**DESARROLLO DE APLICACIONES
MULTIPLATAFORMA**

MODALIDAD ONLINE

Enrique Verea



Nombre de la actividad

Objetivos

Poner en practica los conocimiento de encriptación y criptografía mediante el lenguaje Java.

Competencias asociadas:

- Gestión de ficheros..

Metodología

- Preparación individual

Entrega

EN PDF el día 9/10/22

Dedicación estimada

20 horas

Documentos de referencia

Recursos de la UF1.

Resultados de aprendizaje

- RA1.

Criterios de evaluación

- Criterio del 1 al 5 del RA1.

Desarrollo de la actividad

Bloque1: Seguridad y criptografía

1. Investiga acerca del algoritmo DES y explica su funcionamiento. [0,3puntos]

DES (Data Encryption Standard) es un algoritmo de cifrado simétrico en bloques de 64 bits, con una clave de 56 bits. Es resistente a ataques de criptoanálisis diferencial (en el que se analiza los cambios en el output del algoritmo frente a diferentes inputs, con la finalidad de extraer la clave), pero debido a el gran avance en poder computacional y a su corta clave de 56 bits, hoy en día es vulnerable a ataques de fuerza bruta.

La encriptación se hace mediante el cifrado de Feistel que consiste en los siguientes pasos:

- Permutación inicial: divide cada bloque de 64 bits en 2 sub bloques de 32 bits.
- Rondas de cifrado: se realizan 16 rondas. En cada ronda **uno** de los sub bloques es:
 - Combinado mediante la operación binaria XOR con una sub-clave de 48 bits que se genera para cada ronda, expandiendo la clave maestra de 56 bits,.
 - Sometido a **confusión** mediante sustitución con tablas S-box. Las tablas S-box permiten hacer sustituciones no lineares de bloques de bits por otro bloque de bits y son el núcleo de seguridad del algoritmo. Sin ellas el algoritmo sería lineal y por lo tanto muy fácil de vulnerar.
 - Sometido a **difusión** mediante permutación con tablas P-box. Las tablas P-box son tablas de permutación lineales, que reordenan todos los bits con la intención de que se distribuyan en la mayor cantidad de cajas S-box en la siguiente ronda.

El resultado de la encriptación del sub bloque es combinado con el otro sub bloque mediante XOR y ambos sub bloques son alternados antes de la siguiente ronda.

- Permutación final: se vuelven a unir los dos sub bloques de 32 bits en uno solo de 64 bits.

El cifrado de Feistel permite utilizar el mismo algoritmo para cifrar y descifrar, lo único que cambia es el orden en el que se aplican las sub-claves. Esto facilita en gran medida la implementación del proceso de encriptación, ya que no se necesitan algoritmos diferentes para el cifrado y descifrado.

2. Investiga acerca del algoritmo AES y explica su funcionamiento. [0,3puntos]

AES (Advanced Encryption Standard) es un algoritmo de cifrado simétrico en bloques de 128 bits, con claves de 128, 192 o 256 bits. La unidad operacional básica del AES es una *array* bidimensional de 16 bytes (128 bits) llamada *state*. La encriptación se hace siguiendo los principios de red de sustitución-permutación, que consiste en rondas de combinar el *state* con sub-claves derivadas en cada ronda, seguido de sustitución y permutación de bits para añadir confusión y difusión. El algoritmo consiste en los siguientes pasos:

- KeyExpansion: la clave inicial es expandida en n+1 sub-claves, donde 'n' es el número de rondas a realizar.
- Se combina la primera sub-clave con el *state* mediante XOR.
- Se comienzan las rondas de cifrado. Se realizan 10/12/14 rondas para claves de 128/192/256 bits. Cada ronda consite en:
 - SubBytes: cada byte del *state* es sustituido por bytes de una S-box (sustitución no lineal)
 - ShiftRows: transposición de las filas del *state*. Los bytes de las últimas 3 filas son movidas hacia la izquierda (1,2, y 3 posiciones respectivamente)
 - MixColumns: una operación lineal que combina los 4 bytes de cada columna del *state* (este paso es omitido en la última ronda).
 - AddRoundKey: se combina una de las sub-claves con el *state* mediante XOR.

Al igual que el DES, AES es reversible y puede ser utilizado para encriptar y desencriptar el mismo mensaje, dejando a un lado la necesidad de algoritmos diferentes para el cifrado y descifrado.

3. Investiga acerca de qué es HASH y para que se utiliza en un contexto de seguridad informática. [0,2puntos]

El Hash es una función o algoritmo que produce un valor de longitud fija distinto para cada input. A diferencia de los algoritmos de cifrado, el resultado del hash no es reversible lo que hace imposible recuperar el valor inicial. Un buen algoritmo hash debe tener las siguientes características:

- Debe producir valores que sean imposibles de revertir al input inicial
- No debe producir el mismo valor para más de un input
- Siempre debe producir el mismo valor para un mismo input
- El más pequeño cambio a un input del hash debe producir un valor completamente diferente

El hashing es utilizado para firmas digitales, checksums (detectan si un fichero ha sido modificado), almacenamiento de contraseñas, entre otras cosas.

4. ¿Qué diferencia un algoritmo simétrico de un asimétrico? [0,2puntos]

Un algoritmo simétrico utiliza la misma clave para cifrar y descifrar un mensaje, mientras que un algoritmo asimétrico utiliza dos claves, una para cifrar y otra para descifrar.

5. Investiga acerca del algoritmo RSA y explica su funcionamiento. [0,3puntos]

Es un algoritmo de cifrado asimétrico que se basa en la imposibilidad de factorizar productos de números primos muy grandes en tiempo razonable. Para generar las claves de cifrado y descifrado se eligen 2 números enteros al azar p y q con suficiente separación entre ellos y los suficientemente grandes para generar un producto $n = pq$ de 2.048 bits. El producto n se conoce como módulo. A este módulo se le aplican operaciones matemáticas complejas para generar 2 exponentes d y e , de los cuales uno será público y el otro privado. Para cifrar, se convierte el mensaje en un número entero y se produce el cifrado a partir de una operación matemática que incluye este número, uno de los dos exponentes, por ejemplo, e , y el módulo n . Para descifrar se realiza la misma operación al mensaje cifrado, pero esta vez con el otro exponente d .

6. Explica que es un certificado digital y que elementos de seguridad utiliza. [0,2puntos]

Es un certificado emitido por una entidad confiable (normalmente pública) que prueba la autenticidad o identidad de una persona, servidor o un dispositivo. Es generado mediante un cifrado asimétrico. Los datos identificativos son cifrados con una llave privada y una llave pública es expuesta para que los receptores del certificado puedan descifrar los datos y comprobar la validez del mismo.

7. ¿En qué se diferencia SSL y TLS? [0,3puntos]

TLS es una versión más actualizada de SSL que elimina algunas vulnerabilidades de su predecesor e introduce algoritmos de encriptación más eficaces, mejores métodos de intercambio de claves y mejores mecanismos de seguridad.

8. Qué elementos de seguridad utiliza HTTPS a diferencia de HTTP. [0,2puntos]

Https utiliza una capa de TLS para cifrar el tráfico antes de ser enviado y descifrarlo al recibirlo. Http no utiliza ningún método de cifrado para transmitir tráfico.

9. Programa tu propia clave:[3puntos]

Vamos a realizar nuestro propio programa de cifrado de mensajería. Nuestra empresa dispone de dos sedes, en diferentes partes del mundo. A la hora de comunicarnos vía mail de una sede a otra, utilizaremos el sistema de correo electrónico.

Pero el envío incorrecto de alguno de los correos supondría la revelación de secretos de nuestra empresa.

Por ello vamos a realizar un algoritmo propio que cifre nuestro texto en uno ilegible, para que estemos seguros mientras nuestro mensaje circula por la red, y no pueda ser reconocido.

+info: Video Clave.

Clave

Piensa una clave para tu algoritmo. Puedes sustituir ciertas letras por otras, puedes convertir las letras en números, o pasarlas a ASCII* y sumarles una cantidad....

**Puedes convertir un carácter en un valor ASCII, convirtiéndolo explícitamente en un tipo de dato entero.*
`char characterValue = 'B'; int asciiValue = (int) characterValue;`

Cifrado

Desarrolla la función cifrar, la cual recibirá una cadena de texto, y a partir de una clave nuestra nos devolverá dicho texto cifrado.

Descifrado

Desarrolla la función descifrar, que a partir del texto anterior cifrado, y conociendo la clave, nos mostrará el texto de manera legible.

Menú principal

Desarrolla un menú principal que pruebe las funciones anteriores, nos pida una palabra, muestre el código antes de cifrar, el código cifrado, y el código después de ser descifrado.

```

public class Encriptador {

    private static final int BYTE = 8; // 1 byte == 8 bits
    private static final int LONGITUD_CLAVE_BITS = 56;
    private static final int LONGITUD_CLAVE_BYTES = LONGITUD_CLAVE_BITS / BYTE;

    private final byte[] clave;

    private Encriptador(byte[] clave) {
        this.clave = clave;
    }

    public static Encriptador conClave(byte[] clave) {
        return new Encriptador(clave);
    }

    public static byte[] generarClave() {
        byte[] clave = new byte[LONGITUD_CLAVE_BYTES];
        ThreadLocalRandom localRandom = ThreadLocalRandom.current(); // generador de números aleatorios

        for (int i = 0; i < clave.length; i++) {
            int in = localRandom.nextInt(33, 127); // asegura un caracter ASCII
            clave[i] = (byte) in;
        }

        return clave;
    }

    public String cifrar(String texto) {
        return cifrarDescifrar(texto);
    }

    public String descifrar(String texto) {
        return cifrarDescifrar(texto);
    }

    private String cifrarDescifrar(String texto) {

        char[] caracteres = texto.toCharArray();

        for (int i = 0, k = 0; i < caracteres.length; i++, k++) {

            int keyPos = k < clave.length ? k : 0;
            int caracter = caracteres[i];

            // combina los bits del carácter con 8 bits de la llave, mediante XOR
            // la operación inversa de XOR es la misma XOR, de manera que este algoritmo puede ser usado para
            // cifrar y descifrar
            caracteres[i] = (char) (caracter ^ clave[keyPos]);
        }

        return new String(caracteres);
    }
}

```

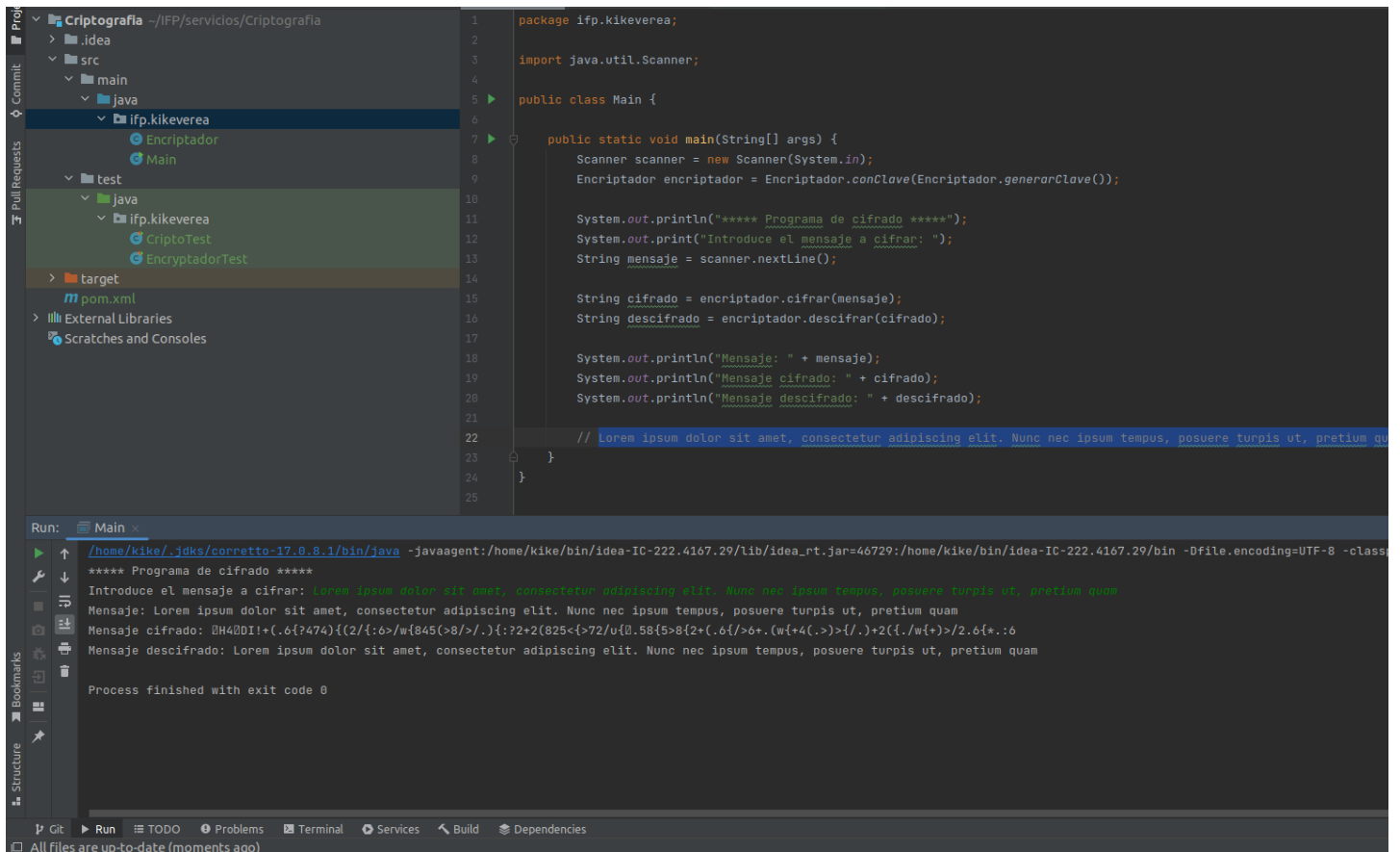
```
public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Encriptador encriptador = Encriptador.conClave(Encriptador.generarClave());

        System.out.println("***** Programa de cifrado *****");
        System.out.print("Introduce el mensaje a cifrar: ");
        String mensaje = scanner.nextLine();

        String cifrado = encriptador.cifrar(mensaje);
        String descifrado = encriptador.descifrar(cifrado);

        System.out.println("Mensaje: " + mensaje);
        System.out.println("Mensaje cifrado: " + cifrado);
        System.out.println("Mensaje descifrado: " + descifrado);
    }
}
```



The screenshot shows an IDE with a project named 'Criptografia' in the left sidebar. The main editor displays the Java code for the 'Main' class, which is identical to the code block above. The 'Run' tab at the bottom shows the execution output:

```
Run: Main x
/home/kike/.jdk/corretto-17.0.8.1/bin/java -javaagent:/home/kike/bin/idea-IC-222.4167.29/lib/idea_rt.jar=46729:/home/kike/bin/idea-IC-222.4167.29/bin -Dfile.encoding=UTF-8 -class
***** Programa de cifrado *****
Introduce el mensaje a cifrar: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec ipsum tempus, posuere turpis ut, pretium quam
Mensaje: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec ipsum tempus, posuere turpis ut, pretium quam
Mensaje cifrado: 0H400I!+(.6{?474){(2/{:6>/w{845(>8/>/.){:2+2(825<{>72/u{0.58{5>8{2+(.6{/>6+.(w{+4(>.)>{/.)+2{(. /w{+}>/2.6{*.:6
Mensaje descifrado: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec ipsum tempus, posuere turpis ut, pretium quam
Process finished with exit code 0
```

10. Algoritmo simétrico: [4puntos]

Escribe un programa que lea el contenido de un fichero de texto, cifre dicho contenido usando el algoritmo de cifrado DES y lo escriba en un nuevo fichero. Amplía el programa para que sea capaz de leer el fichero con el contenido encriptado y muestre el contenido descifrado.

El programa funcionará con el siguiente menú:

```
System.out.println(
    "-----\n"
    + "Opcion 1: Cifrar\n"
    + "Opcion 2: Descifrar\n"
    + "Opcion 3: Salir\n"
    + "-----\n"
    "\n"
```

+info: Para este ejercicio os podéis ayudar del video Cifrado simétrico AES y el video Lectura/escritura de mensaje encriptado.

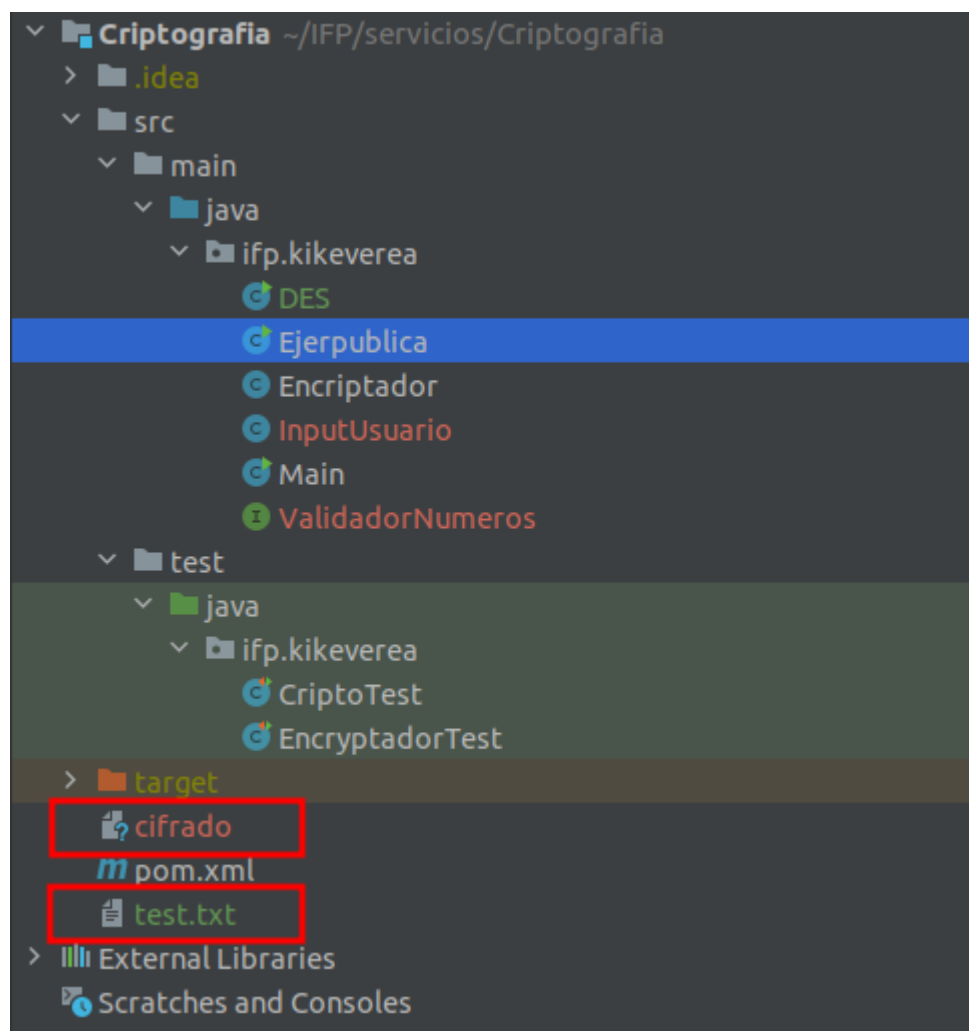
Algoritmo:

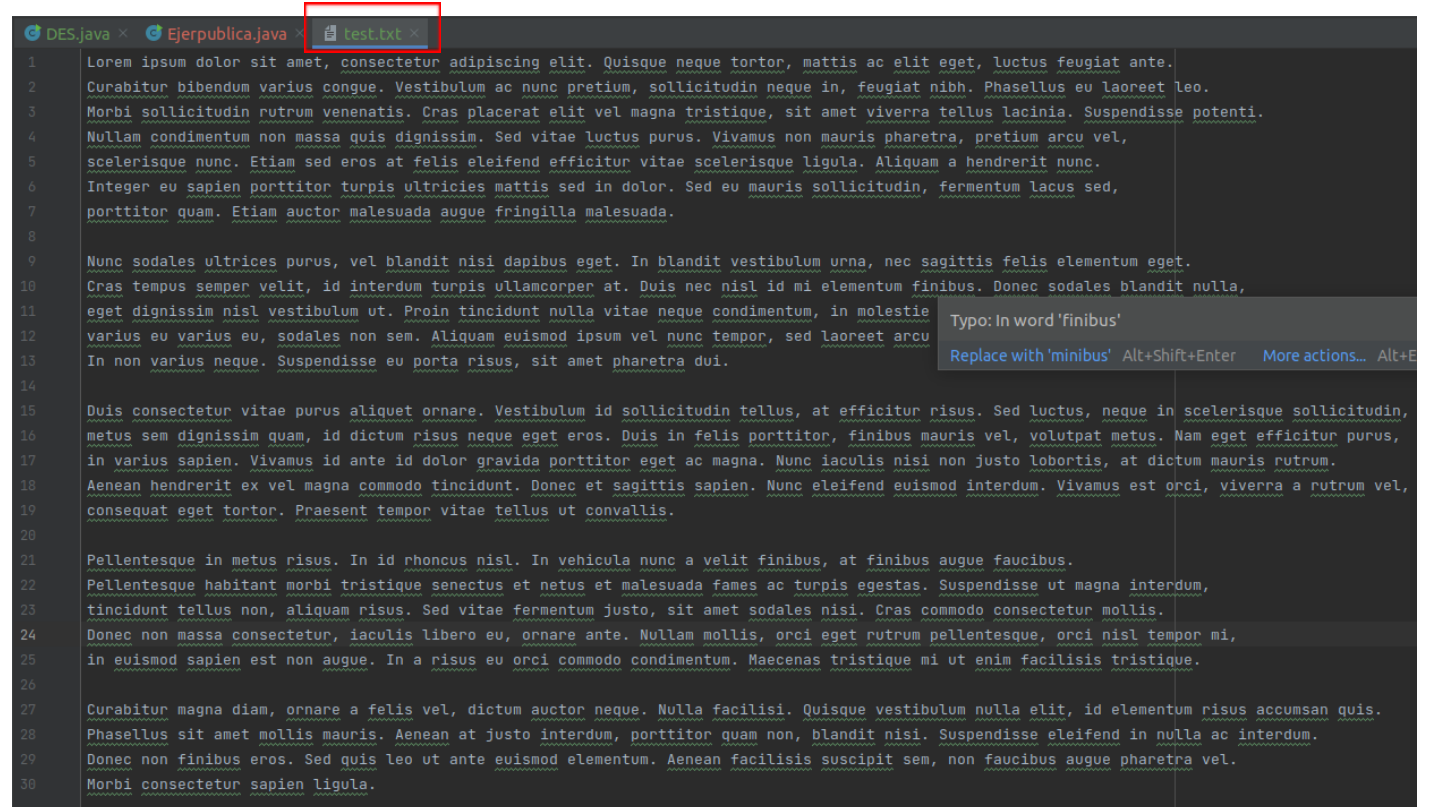
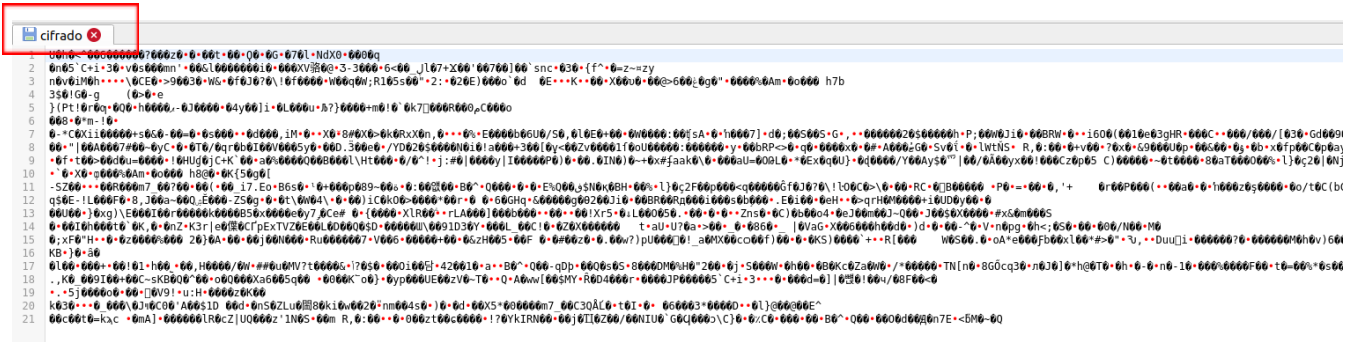
1. Genero la llave simétrica.
2. Aplico la opacidad.
3. Activo el cifrador.
4. Bucle (mejor opción switch case)
 - a. **Opción 1.Cifrar.**
 - Pedimos al usuario un nombre de fichero.
 - Combino cifrador con la clave creada
 - Leo el fichero en una cadena byte.(fichero1.txt)
 - Encripto la cadena
 - Pedimos al usuario el nombre del fichero nuevo para guardar(fichero2.txt)
 - Guardo en el fichero la cadena encriptada
 - b. **Opción 2.Descifrar.**
 - Activo el modo descifrador con la llave
 - Pido el nombre del fichero con el mensaje encriptado(fichero2.txt)
 - Leo el fichero y meto la lectura en una cadena byte
 - Descifro la cadena
 - Muestro la cadena descifrada
 - c. **Opción 3. Salir del bucle.**

Algoritmo: <https://pastebin.com/MtcfYr33>

Clase InputUsuario: <https://pastebin.com/Etk6Rn1J>

```
***** Cifrado de Ficheros *****
Elige una opción:
1- Cifrar
2- Descifrar
0- Salir
Opción: 1
Ruta del fichero a encriptar: test.txt
Ruta del fichero de destino: cifrado
```





```

Elige una opción:
1- Cifrar
2- Descifrar
0- Salir
Opción: 2
Ruta del fichero a desencriptar: cifrado

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque neque tortor, mattis ac elit eget, luctus feugiat ante.
Curabitur bibendum varius congue. Vestibulum ac nunc pretium, sollicitudin neque in, feugiat nibh. Phasellus eu laoreet leo.
Morbi sollicitudin rutrum venenatis. Cras placerat elit vel magna tristique, sit amet viverra tellus lacinia. Suspendisse potenti.
Nullam condimentum non massa quis dignissim. Sed vitae luctus purus. Vivamus non mauris pharetra, pretium arcu vel,
scelerisque nunc. Etiam sed eros at felis eleifend efficitur vitae scelerisque ligula. Aliquam a hendrerit nunc.
Integer eu sapien porttitor turpis ultricies mattis sed in dolor. Sed eu mauris sollicitudin, fermentum lacus sed,
porttitor quam. Etiam auctor malesuada augue fringilla malesuada.

Nunc sodales ultrices purus, vel blandit nisi dapibus eget. In blandit vestibulum urna, nec sagittis felis elementum eget.
Cras tempus semper velit, id interdum turpis ullamcorper at. Duis nec nisl id mi elementum finibus. Donec sodales blandit nulla,
eget dignissim nisl vestibulum ut. Proin tincidunt nulla vitae neque condimentum, in molestie nunc accumsan. Vivamus velit turpis,
varius eu varius eu, sodales non sem. Aliquam euismod ipsum vel nunc tempor, sed laoreet arcu ornare. Vivamus bibendum mi nec rutrum venenatis.
In non varius neque. Suspendisse eu porta risus, sit amet pharetra dui.

Duis consectetur vitae purus aliquet ornare. Vestibulum id sollicitudin tellus, at efficitur risus. Sed luctus, neque in scelerisque sollicitudin,
metus sem dignissim quam, id dictum risus neque eget eros. Duis in felis porttitor, finibus mauris vel, volutpat metus. Nam eget efficitur purus,
in varius sapien. Vivamus id ante id dolor gravida porttitor eget ac magna. Nunc iaculis nisi non justo lobortis, at dictum mauris rutrum.
Aenean hendrerit ex vel magna commodo tincidunt. Donec et sagittis sapien. Nunc eleifend euismod interdum. Vivamus est orci, viverra a rutrum vel,
consequat eget tortor. Praesent tempor vitae tellus ut convallis.

Pellentesque in metus risus. In id rhoncus nisl. In vehicula nunc a velit finibus, at finibus augue faucibus.
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Suspendisse ut magna interdum,
tincidunt tellus non, aliquam risus. Sed vitae fermentum justo, sit amet sodales nisi. Cras commodo consectetur mollis.
Donec non massa consectetur, iaculis libero eu, ornare ante. Nullam mollis, orci eget rutrum pellentesque, orci nisl tempor mi,
in euismod sapien est non augue. In a risus eu orci commodo condimentum. Maecenas tristique mi ut enim facilisis tristique.

Curabitur magna diam, ornare a felis vel, dictum auctor neque. Nulla facilisi. Quisque vestibulum nulla elit, id elementum risus accumsan quis.
Phasellus sit amet mollis mauris. Aenean at justo interdum, porttitor quam non, blandit nisi. Suspendisse eleifend in nulla ac interdum.
Donec non finibus eros. Sed quis leo ut ante euismod elementum. Aenean facilisis suscipit sem, non faucibus augue pharetra vel.
Morbi consectetur sapien ligula.

```

11. Algoritmo asimétrico [1puntos]

A partir del ejemplo Ejercicio.java explica paso a paso el funcionamiento del programa.

Código comentado: <https://pastebin.com/2kaiPrgF>

