Manual de usuario de la File Encoder Application v1.0

# Índice

[0. Índice 1](#_Toc415437488)

[1. Introducción 1](#_Toc415437489)

[2. Inicio rápido 2](#_Toc415437490)

[3. Funcionamiento de la aplicación 3](#_Toc415437491)

[3.1. Pantalla principal 3](#_Toc415437492)

[3.2. Menú Archivo 5](#_Toc415437493)

[3.3. Menú Herramientas 6](#_Toc415437494)

[3.4. Menú Acerca de 6](#_Toc415437495)

[3.5. Lista de configuraciones de encriptación 7](#_Toc415437496)

[3.5.1. Conceptos básicos 7](#_Toc415437497)

[3.5.2. El formulario 7](#_Toc415437498)

[3.6. Configuración de encriptación 8](#_Toc415437499)

[3.6.1. Parámetros de la configuración de encriptación 9](#_Toc415437500)

[3.7. Configuración de la aplicación 12](#_Toc415437501)

[4. Encriptación por la línea de comandos 14](#_Toc415437502)

[4.1. Scripts para Windows 14](#_Toc415437503)

[4.2. Scripts para Linux y para Mac 15](#_Toc415437504)

[4.3. Parámetros de encriptación como argumentos 15](#_Toc415437505)

[5. Método de encriptación 16](#_Toc415437506)

[5.1. Cabecera del fichero encriptado 16](#_Toc415437507)

[5.2. Algoritmo de encriptación 17](#_Toc415437508)

[5.3. Generador pseudoaleatorio utilizado 18](#_Toc415437509)

# Introducción

La File Encoder Application es una aplicación que permite encriptar ficheros.

Básicamente permite:

* Encriptar fichero.
* Desencriptar fichero (encriptado con esta misma aplicación).
* Abrir fichero encriptado (desencriptándolo previamente)

La aplicación permite operar con ficheros encriptados (con extensión .jfe) y desencriptados con su interfaz gráfica.

Así mismo el motor de la aplicación puede ser invocado a través de la interfaz de comandos obteniendo de esta forma la misma funcionalidad en cuando a encriptación/desencriptación que la aplicación gráfica.

La única desventaja de utilizar el interfaz de comandos es que hay que escribir la contraseña en claro, y puede ser vista por algún curioso.

La aplicación está escrita en Java, por lo que es portable a distintas plataformas y debería ser compatible. Es decir, debería ser posible encriptar un fichero en una plataforma y desencriptarlo en otra distinta.

Ha sido probado sólo con procesadores Intel, en Windows 8, Linux y OS-X pero también debería ser compatible con cualquier plataforma que tenga instalada la máquina virtual de Java (JRE).

Existen varios parámetros de configuración de la encriptación que modificados convenientemente pueden permitir:

* incrementar la rapidez de los procesos de encriptación/desencriptación.
* incrementar la robustez de la encriptación.
* minimizar el uso de memoria.

Desgraciadamente no pueden optimizarse todos los puntos a la vez. Así pues, si se quiere incrementar la robustez de la encriptación a través de los valores de los parámetros de encriptación, eso va en contra de incrementar la rapidez o en contra de minimizar el uso de la memoria.

Si no tienes ganas de aprender los parámetros, puede dejarse la configuración básica que incluye configuración de parámetros basados en el tamaño del fichero, para permitir que los ficheros mayores sean encriptados con una configuración más rápida que los pequeños.

Independientemente de los parámetros utilizados para encriptar, una buena contraseña ayuda a que la robustez de la encriptación sea mayor.

Por ello se aconseja utilizar una contraseña muy larga (puede ser una frase), para que un intento de romper la encriptación por fuerza bruta se haga más complicado.

Por ejemplo, una buena contraseña debería tener mayúsculas, minúsculas, números y no estar compuesta únicamente por palabras de diccionario. Si a ello añadimos que la contraseña puede tener una longitud larga, por ejemplo entre 30 y 40 caracteres, el sistema de encriptado se hace más fuerte.

# Inicio rápido

Antes de empezar, indicar que la aplicación utiliza el micrófono para recoger bytes aleatorios, por lo que sería necesario tener un micrófono conectado para que la aplicación funcione de forma óptima. De otra forma, la encriptación sería mucho más débil.

Para comenzar a utilizar la aplicación, sigue los siguientes pasos:

1. Abre la File Encoder Application, haciendo doble click en: el .jar de la aplicación: que está en .../\_binary/FileEncoderApplication.jar
2. Abre una ventana nueva de fichero (Menú: Archivo->Nuevo).
3. Fichero. Introduce el fichero que deseas encriptar (cualquier nombre, excepto extensión .jfe) o desencriptar (fichero con extensión .jfe). Puedes pulsar el botón "..." para abrir el diálogo de selección de fichero.

Ten en cuenta que para encriptar o desencriptar un fichero, no debe existir el archivo encriptado o desencriptado. Si existe este archivo, la operación fallará, ya que la aplicación no va a sobreescribir el archivo.

1. Contraseña. Introduce la contraseña que va a tener el fichero encriptado, o que tiene el fichero a desencriptar.
2. Repetir contraseña. Vuelve a introducir la contraseña. Ambas contraseñas deben coincidir para poder encriptar/desencriptar.

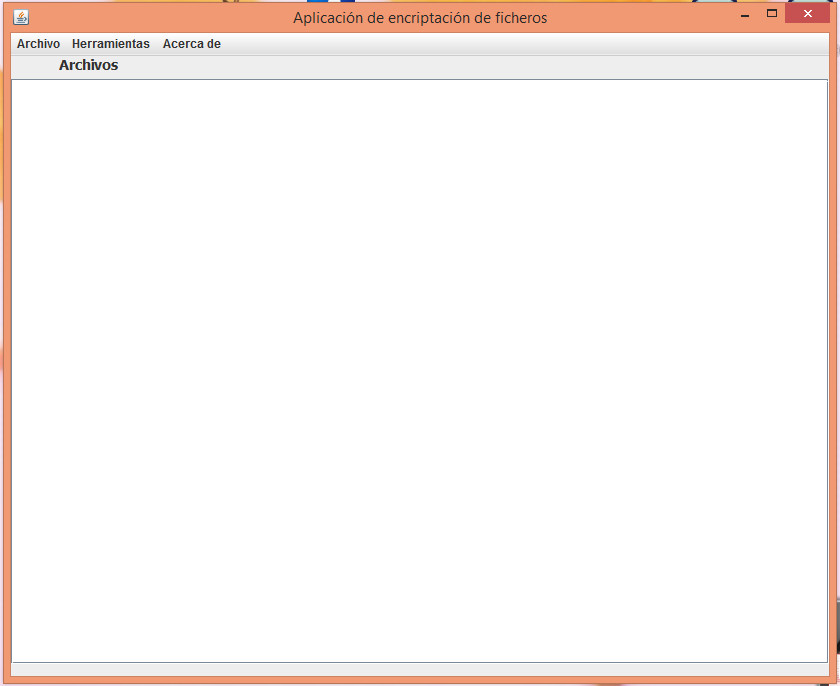
Tras estos pasos, ya puedes utilizar los botones de Encriptar, Desencriptar o abrir fichero encriptado.

# Funcionamiento de la aplicación

En este capítulo se verá como encriptar, desencriptar y abrir ficheros encriptados y también se verán las opciones más comunes a la hora de ejecutar estas acciones.

## Pantalla principal

Al arrancar la aplicación gráfica, aparece una pantalla como la siguiente:



En ella puede observarse que hay un menú principal y un área grande principal donde estarán las ventanas MDI de la aplicación.

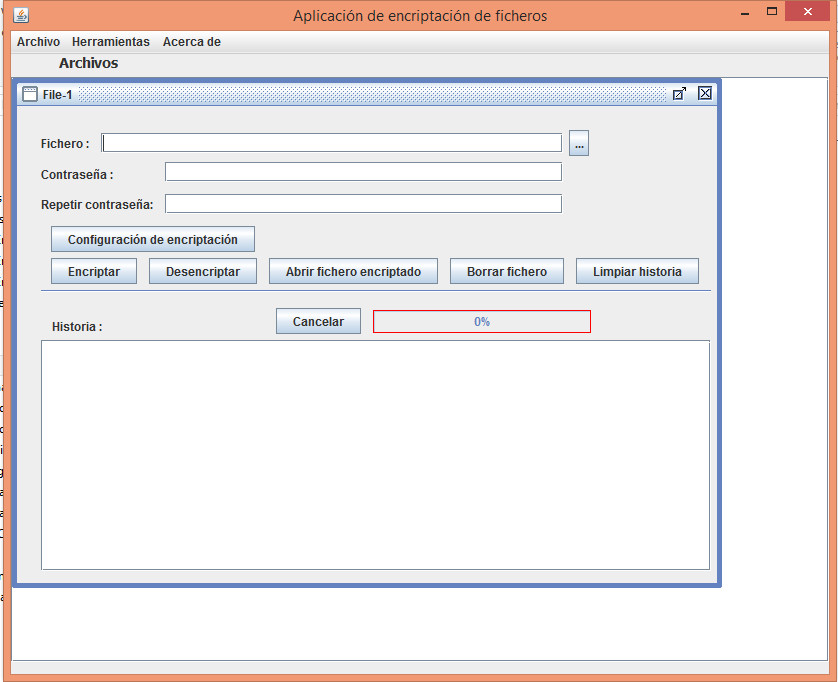
En el área donde se ponen las ventanas de ficheros, existe un menú emergente que aparece al pulsar botón derecho sobre ella.

En ese menú aparece una sola opción: Nuevo

Seleccionando esa opción aparecerá una nueva ventana de fichero.

Se pueden abrir tantas ventanas de fichero como sea necesario.

El aspecto de la ventana de fichero es este:



En ella hay varios parámetros que deben rellenarse antes de acceder a la funcionalidad.

* "Fichero". En este cuadro de texto debe aparecer el nombre del fichero sobre el que se aplicarán las acciones.

El fichero puede ser o bien un fichero sin encriptar (con cualquier extensión), o bien un fichero encriptado (con extensión .jfe).

Para rellenarlo se puede hacer:

* + Pulsando el botón de su derecha "...". Cuando se pulse este botón, se abrirá un diálogo para navegar por los discos del equipo y seleccionar el fichero.
  + editándolo con el teclado (esto puede ser útil cuando ya tienes en el cuadro de texto el nombre del fichero y quieres pasar de trabajar con el fichero sin encriptar a trabajar con el fichero encriptado, añadiendo la extensión .jfe o borrándola).
* "Contraseña" y "Repetir Contraseña". En estos dos cuadros de texto debes escribir la contraseña que se va a aplicar al encriptar o desencriptar el fichero.

Los dos cuadros de texto deben contener la misma contraseña. Esto es así para que, en caso de haber escrito mal la contraseña (sobre todo al encriptar), puedas darte cuenta. De otra forma se encriptaría con una contraseña desconocida y luego no podría desencriptarse el fichero.

Obviamente la contraseña utilizada para encriptar y para desencriptar debe ser la misma para que se realice el proceso de desencriptación correctamente.

Una vez rellenos los parámetros anteriores, ya puede utilizarse la funcionalidad.

Con la nueva ventana de fichero se puede acceder a las funcionalidades pulsando los siguientes botones:

* "Encriptar". Encripta el fichero
* "Desencriptar". Desencripta el fichero
* "Abrir un fichero encriptado". Desencripta y abre el fichero con la aplicación predefinida en el sistema para ese tipo de ficheros.
* Modificar la "Configuración de encriptación". Sirve para modificar los parámetros de encriptación. Ver punto: 3.6-Configuración de encriptación
* "Borrar fichero". Borra el fichero
* "Limpiar historia". Borra la historia del cuadro de texto Historia.
* "Cancelar". Cancela la encriptación o desencriptación.

Hay un cuadro de texto (Historia), en el que aparece el estado y resultado de las acciones que vamos tomando sobre los ficheros.

También existe una barra que indica el progreso de la acción en curso y cuando finaliza la encriptación/desencriptación de un fichero, existe una etiqueta a la derecha de la barra de progreso que indica el tiempo transcurrido en milisegundos desde el inicio de la acción.

## Menú Archivo

El menú Archivo tiene este aspecto:

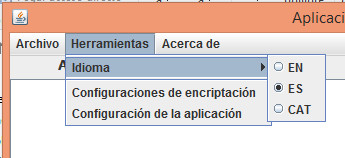


Este menú tiene dos opciones:

* Nuevo. Al pulsar sobre este elemento del menú crearemos una nueva ventana de fichero, vista en el punto anterior.
* Salir. Permite salir de la aplicación.

## Menú Herramientas

El menú Herramientas tiene este aspecto:



Este menú tiene tres opciones:

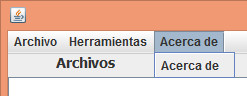
* Idioma. Al abrir este submenú aparecen los idiomas disponibles. Sirve para poder cambiar el idioma de la aplicación.

Por defecto hay disponibles los siguientes idiomas:

* "EN". Inglés
* "ES". Castellano
* "CAT". Catalán
* Configuraciones de encriptación. Cuando se pulsa este elemento de menú, aparece un formulario en el que se permite modificar las configuraciones de encriptación utilizadas según los tamaños de fichero a encriptar. Ver punto: 3.5-Lista de configuraciones de encriptación
* Configuración de la aplicación. Cuando se pulsa este elemento de menú, aparece un formulario en el que se permite modificar los parámetros de configuración generales de la aplicación. Ver punto: 3.7-Configuración de la aplicación

## Menú Acerca de

El menú Acerca de tiene este aspecto:



Este menú sólo tiene una opción:

* Acerca de. Cuando se pulsa este elemento de menú, aparece un formulario con los datos de la aplicación, los agradecimientos y los datos de contacto.

## Lista de configuraciones de encriptación

Este formulario permite crear, modificar y borrar configuraciones de encriptación o revertir a la lista de configuraciones de encriptación por defecto.

### Conceptos básicos

Como se comentó en la introducción, existen varios parámetros para la encriptación que inciden sobre la rapidez, el uso de memoria y la robustez de la encriptación. Ver punto: 3.6.1-Parámetros de la configuración de encriptación

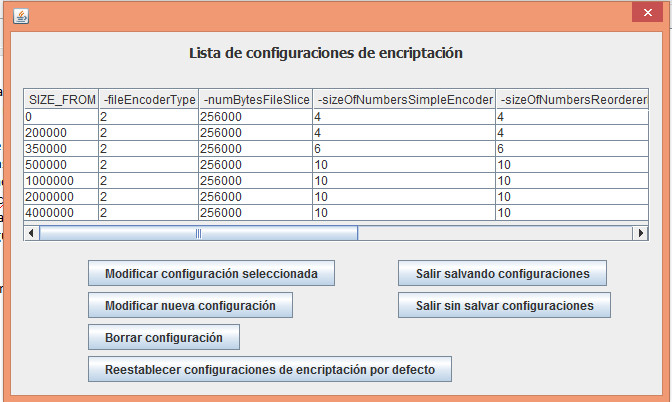
Dado que los parámetros de encriptación inciden sobre todo en la rapidez de la encriptación/desencriptación parece lógico pensar que en función del tamaño del fichero debería utilizarse una configuración de encriptación u otra.

Para facilitar las cosas al usuario final, la File Encoder Application permite tener configuraciones de encriptación predefinidas para los distintos rangos de tamaño de fichero, pudiendo escoger automáticamente una configuración de encriptación en función del tamaño del fichero a encriptar.

Este formulario permite mantener esas configuraciones de encriptación definidas para cada rango de tamaño de fichero.

### El formulario

El formulario tiene el siguiente aspecto:



En él existe una tabla resumen de las configuraciones de encriptación ordenadas de menor a mayor tamaño de fichero, de la que sólo puede seleccionarse una fila a la vez.

Cada fila se corresponde con una configuración de encriptación, asociada a un rango de tamaños de fichero.

La configuración de encriptación aplicaría al rango de tamaños de fichero desde el campo SIZE\_FROM de la propia configuración de encriptación en octetos, hasta el campo SIZE\_FROM de la siguiente configuración de encriptación (la siguiente fila), o bien ilimitado si no existe ninguna fila posterior.

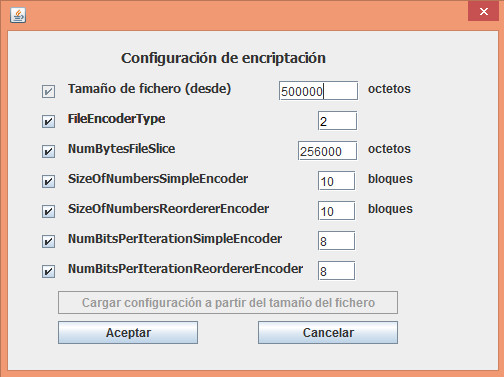
También existen una serie de botones que aportan la funcionalidad del formulario:

* Modificar configuración seleccionada. Abre el formulario que permite modificar los parámetros de encriptación de la configuración de encriptación seleccionada de la tabla. Ver punto: 3.6-Configuración de encriptación
* Modificar nueva configuración. Abre el formulario que permite crear una configuración de encriptación desde cero, teniendo que incluir el parámetro SIZE\_FROM que permitirá definir el rango de tamaños de fichero para los cuales aplica esa configuración.
* Borrar configuración. Borra la configuración de encriptación seleccionada de la tabla.
* Restablecer configuraciones de encriptación por defecto. Revierte todos los cambios que se hayan realizado sobre las configuraciones de encriptación por rango de tamaño de fichero, dejando la lista de configuraciones de encriptación original (la que venía inicialmente en la aplicación).
* Salir salvando configuraciones. Permite salir del formulario salvando los cambios realizados en las configuraciones de encriptación.
* Salir sin salvar configuraciones. Permite salir del formulario sin salvar ni aplicar los cambios realizados en las configuraciones de encriptación por rango de tamaño de fichero.

## Configuración de encriptación

Este formulario permite modificar los parámetros de encriptación para una configuración de encriptación determinada, ya sea para aplicarla directamente sobre un fichero a encriptar (desde un formulario de fichero) o bien para modificar los parámetros de configuración de encriptación de un rango de tamaño de fichero determinado.

El formulario tiene el siguiente aspecto:



A este formulario puede llegarse o bien desde la ventana de fichero, vista en el punto: 3.1-Pantalla principal o bien desde el formulario de lista de configuraciones de encriptación en el punto: 3.5.2-El formulario.

Existe un botón para cargar los parámetros por defecto en función del tamaño del fichero. Este botón sólo está habilitado en caso de que el acceso al formulario haya sido a través de la ventana de fichero. En caso de haber accedido a este formulario a través del formulario de lista de configuraciones de encriptación, este botón estará deshabilitado.

Este botón sólo ofrece funcionalidad en caso de que esté habilitado y además tiene que existir el fichero configurado en la ventana de fichero. En otro caso al pulsarlo la aplicación no hará nada.

### Parámetros de la configuración de encriptación

Se pueden configurar los siguientes parámetros:

* Tamaño de fichero (desde). Este valor no es un parámetro de configuración propiamente dicho y sólo está habilitado en caso de que el acceso al formulario haya sido a través del formulario de lista de configuraciones de encriptación.

Esto es así, porque en ese caso se pretende crear o editar una configuración de encriptación que aplicará a un rango de tamaños de fichero. El tamaño de fichero (desde) indicará el inicio del rango de tamaño de fichero en bytes. El final del rango al que aplicará la configuración de este formulario, dependerá del parámetro tamaño de fichero (desde) de la siguiente configuración de encriptación en la lista de configuraciones de encriptación.

En caso de que se haya accedido a este formulario desde la ventana de fichero, la configuración de encriptación que se está editando, en caso de ser aceptada, podrá ser aplicada directamente en los parámetros de la próxima encriptación, dependiendo de la configuración de prioridades de configuración de encriptación explicada en el punto: 3.7-Configuración de la aplicación

* FileEncoderType. Este parámetro indica qué clase de FileEncoder se va a utilizar para encriptar. Es un valor numérico que, para la v1.0, puede tomar los valores 1 o 2.
* 1 (obsoleto). Clase de FileEncoder que utiliza un generador pseudoaleatorio caótico basado en la case BigDecimal de Java para hacer los cálculos.

Como para hacer las operaciones de encriptación se utiliza esta clase a bajo nivel, es posible que algún cambio de implementación en esta clase en alguna actualización de Java, pudiera romper la compatibilidad hacia atrás, por lo que se decidió crear una clase propia numérica larga menos dependiente de Java.

* 2 (aconsejado). Este es el valor aconsejado para esta versión de la aplicación. Indica que se utilice la clase de FileEncoder que utiliza un generador pseudoaleatorio caótico basado en una clase numérica propia para hacer los cálculos.

Esta clase trabaja con números de precisión variable y no es genérica. Está muy enfocada al rango numérico alcanzado con el generdor pseudoaleatorio caótico.

Contiene la información del número en un array de long de los que por razones de optimización sólo se utilizan los 31 bits inferiores de cada elemento del array.

Cuando se hable de "bloques" en los siguientes parámetros de encriptación aplicables a este FileEncoderType, nos referimos a un elemento del array de long, es decir un long (31 bits).

* NumBytesFileSlice. Este parámetro de configuración de encriptación indica el número de bytes de rodaja que se utilizarán para encriptar.

Para ficheros grandes no es viable cargar todo el fichero en memoria y hacer los cálculos para la encriptación, ya que para hacer esos cálculos se utiliza una memoria entre 40 y 100 veces mayor que la del tamaño en bytes que se está encriptando, por lo que para, por ejemplo, un fichero de 100 MiB, tendríamos que la cota superior de la memoria utilizada serían 10 GiB, cantidad de memoria que no está disponible en muchas máquinas actuales.

Este parámetro hace que el contenido del fichero se divida en rodajas, y para cada una de esas rodajas, primero se lea el contenido de la rodaja en un array de bytes, se aplique la encriptación, y se escriba en el fichero encriptado el contenido de la rodaja encriptada. Para encriptar la siguiente rodaja ya no será necesario tener en memoria las rodajas anteriores, por lo que no sería necesario utilizar tanta memoria como si se encriptara de una vez el fichero completo.

Entonces bajar este parámetro hace que baje el consumo de memoria de la aplicación.

Sin embargo no sería bueno bajar mucho este parámetro, ya que la encriptación completa del fichero puede romperse por fuerza bruta si se rompe la encriptación de la primera rodaja, por lo que para poner las cosas más costosas al que quiera romper esta encriptación, este parámetro debería estar lo más alto posible en función de la memoria de que dispongamos.

Habrá que tener en cuenta que si ponemos este parámetro muy alto, el archivo podría no poder ser desencriptado en una máquina con menos memoria.

* SizeOfNumbersSimpleEncoder. Este parámetro de configuración de encriptación aplica al tamaño numérico aplicado en el generador pseudoaleatorio caótico.

Ver punto: 5.2-Algoritmo de encriptación

El parámetro SizeOfNumbersSimpleEncoder hace referencia al tamaño de los números en el generador pseudoaleatorio utilizado para el primer paso de la encriptación (XOR).

Las unidades de este parámetro es ligeramente diferente para los FileEncoderType 1 y 2.

* Para el FileEncoderType 1, este parámetro está expresado en bytes.
* Para el FileEncoderType 2, este parámetro está expresado en bloques de 31 bits. En este caso sería igual al número de elementos long que hay en el array que tiene los datos del número. (En cada elemento long del array se almacenan 31 bits).
* SizeOfNumbersReordererEncoder. Este parámetro de configuración de encriptación aplica al tamaño numérico aplicado en el generador pseudoaleatorio caótico del segundo paso de la encriptación (reordenación).

Las unidades de este parámetro es ligeramente diferente para los FileEncoderType 1 y 2.

* Para el FileEncoderType 1, este parámetro está expresado en bytes.
* Para el FileEncoderType 2, este parámetro está expresado en bloques de 31 bits. En este caso sería igual al número de elementos long que hay en el array que tiene los datos del número. (En cada elemento long del array se almacenan 31 bits).
* NumBitsPerIterationSimpleEncoder. Este parámetro de configuración de encriptación aplica al número de bits que devuelve el generador pseudoaleatorio en cada iteración.

Este parámetro aplica al generador pseudoaleatorio que se utiliza en el primer paso (XOR).

Este parámetro influye drásticamente en el tiempo de computación de la encriptación.

Para ficheros pequeños a encriptar, es aconsejable utilizar un valor muy bajo de este parámetro, para que la fortaleza de la encriptación sea mayor.

Para ficheros grandes o muy grandes, será mejor utilizar valores grandes para este parámetro, ya que de lo contrario el tiempo para encriptar/desencriptar sería muy grande y se haría pesado tener que esperar cada vez que se necesite encriptar/desencriptar el fichero.

Los valores aceptados por el FileEncoderType 1 y el 2 son ligeramente distintos:

* Para el FileEncoderType 1, este parámetro puede aceptar los valores 1, 2 o 4. El valor 4 es el que produce encriptaciones más rápidas, sin embargo no se aconseja utilizar con SizeOfNumbersSimpleEncoder menores a 16 bytes, ya que en ese caso se obtendrían números entre 0 y 15 en cada iteración que no serían equiprobables. A partir de tamaños de números de 16 bytes sí que podrían utilizarse aunque mejor a partir de números de 32 bytes.
* Para el FileEncoderType 2, este parámetro puede aceptar los valores 1, 2, 4 o múltiplos de 8 hasta 64 (64 es el valor máximo). No todos los valores están permitidos.

Para el valor de 4 es aconsejable no utilizar tamaños de números (configurados en SizeOfNumbersSimpleEncoder) menores a 5 bloques. Mejor aún a partir de 8 bloques.

Para valores mayores que 4 (múltiplos de 8 hasta 64), el valor configurado debe cumplir esta condición:

NumBitsPerIterationSimpleEncoder <= 3 x SizeOfNumbersSimpleEncoder

* NumBitsPerIterationReordererEncoder. Este parámetro de configuración de encriptación aplica al número de bits que devuelve el generador pseudoaleatorio en cada iteración.

Este parámetro aplica al generador pseudoaleatorio que se utiliza en el segundo paso (reordenación).

Este parámetro influye drásticamente en el tiempo de computación de la encriptación.

Para ficheros pequeños a encriptar, es aconsejable utilizar un valor muy bajo de este parámetro, para que la fortaleza de la encriptación sea mayor.

Para ficheros grandes o muy grandes, será mejor utilizar valores grandes para este parámetro, ya que de lo contrario el tiempo para encriptar/desencriptar sería muy grande y se haría pesado tener que esperar cada vez que se necesite encriptar/desencriptar el fichero.

Los valores aceptados por el FileEncoderType 1 y el 2 son ligeramente distintos:

* Para el FileEncoderType 1, este parámetro puede aceptar los valores 1, 2 o 4. El valor 4 es el que produce encriptaciones más rápidas, sin embargo no se aconseja utilizar con SizeOfNumbersReordererEncoder menores a 16 bytes, ya que en ese caso se obtendrían números entre 0 y 15 en cada iteración que no serían equiprobables. A partir de tamaños de números de 16 bytes sí que podrían utilizarse aunque mejor a partir de números de 32 bytes.
* Para el FileEncoderType 2, este parámetro puede aceptar los valores 1, 2, 4 o múltipos de 8 hasta 64 (64 es el valor máximo). No todos los valores están permitidos.

Para el valor de 4 es aconsejable no utilizar tamaños de números (configurados en SizeOfNumbersReordererEncoder) menores a 5 bloques. Mejor aún a partir de 8 bloques.

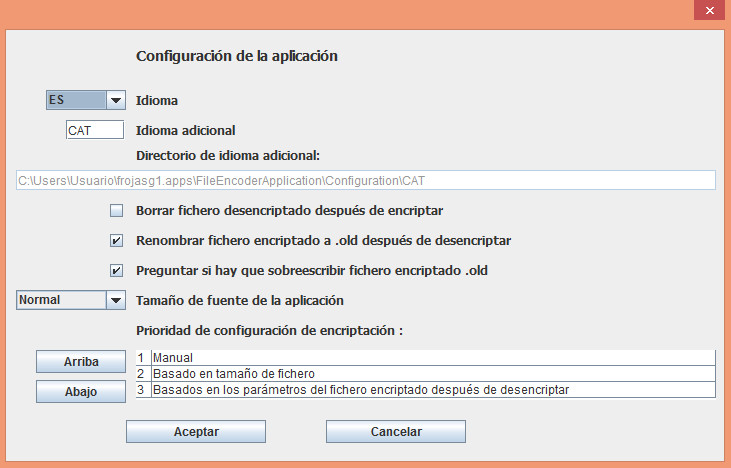
Para valores mayores que 4 (múltiplos de 8 hasta 64), el valor configurado debe cumplir esta condición:

NumBitsPerIterationReordererEncoder <= 3 x SizeOfNumbersReordererEncoder

## Configuración de la aplicación

Este formulario permite modificar los parámetros de configuración generales de la aplicación.

El formulario tiene el siguiente aspecto:



Los parámetros que pueden configurarse son los siguientes:

* Idioma. Es el idioma de los textos que escribe la aplicación.

Los idiomas que pueden escogerse son:

* EN. Inglés
* ES. Español
* El idioma adicional que por defecto es CAT
* Idioma adicional. Es el idioma adicional que aparece a la hora de poder cambiar de idioma.

Si quieres poner un idioma del que no existen textos en la aplicación, puedes cambiar el idioma adicional a tu idioma y traducir los ficheros que hay dentro del directorio que se crea cuando pulsas el botón de Aceptar.

Los ficheros a traducir se copian al directorio indicado en: "Directorio de idioma adicional"

El formato de estos ficheros de textos es el de properties de java.

Para los que no conozcáis este formato, os diré que un fichero properties, tiene un título, y tras este, un número variable de etiquetas con su valor, parecido a lo que sigue:

# TITULO

# xxxxxxxxxx

ETIQUETA1=texto 1

ETIQUETA2=texto 2

...

Las etiquetas deben permanecer sin modificar, y los textos deben modificarse dependiendo de la traducción en el idioma escogido.

Si creas un idioma adicional distinto a los que hay disponibles en la aplicación, si quieres puedes enviármelo (frojasg1@hotmail.com) y lo incluiré en las siguientes versiones de la aplicación.

* Tamaño de la fuente de la aplicación. Este parámetro permite controlar el tamaño de las fuentes que aparecen en la aplicación, existiendo la posibilidad de escoger el tamaño normal, o bien el tamaño grande.
* Borrar fichero encriptado después de encriptar. Este check box permite configurar precisamente eso, si tras encriptar un fichero, deseas borrar automáticamente el fichero sin encriptar.

El borrado que se realiza no es un borrado normal, sino que se borra poniendo ceros en cada uno de los bytes del fichero, y tras esto, se hace un borrado normal.

De esta forma, es imposible recuperar el fichero sin encriptar, a menos que desencriptes el fichero encriptado de la forma habitual, utilizando la contraseña.

* Renombrar fichero encriptado a .old después de desencriptar. Este check box permite configurar si tras una desencriptación deseas guardar el fichero encriptado que acaba de ser desencriptado como .old. De esta forma te aseguras tener almacenada una versión encriptada del fichero, por si acaso ocurre algo inesperado.
* Preguntar si hay que sobreescribir fichero encriptado .old. En caso de que la opción anterior esté activa, esta opción permite configurar si deseas ser preguntado para confirmar el sobreescribir un fichero encriptado .old con el nuevo fichero encriptado pendiente de ser almacenado como .old.
* Tamaño de fuente de la aplicación. Este parmámetro de configuración permite cambiar el tamaño de fuente utilizado en la aplicación.

Tiene dos opciones:

* Normal. En este caso las fuentes que aparecen en pantalla son las originales.
* Grande. En este caso las fuentes que aparecen en pantalla son 1,4 veces más grandes que las originales.
* Prioridad de configuración de encriptación. En esta tabla se puede configurar el orden de prioridad de la configuración de encriptación.

Por defecto las opciones tienen esta prioridad:

* (1) - Manual. Se refiere a cuando aplicas una configuración de encriptación manualmente.

Lo lógico es ponerla como primera opción, ya que si no la pones como primera opción, es posible que hagas la modificación de la configuración de encriptación y la aceptes, pero esta configuración no sea aplicada cuando realizas la encriptación.

* (2) - Basado en tamaño de fichero. Esta opción se refiere a que la configuración de encriptación se obtiene de la lista de configuraciones de encriptación que hay configuradas dependiendo del tamaño del fichero. Ver punto: 3.5-Lista de configuraciones de encriptación
* (3) - Basados en los parámetros del fichero encriptado después de desencriptar. Se refiere a recoger la configuración de encriptación basada en la configuración de encriptación existente en el fichero encriptado que acaba de ser desencriptado.

La manera de configurar este orden de prioridades, es seleccionar en la tabla la fila donde está la opción a la que deseas cambiarle la prioridad, y pulsar los botones Arriba y Abajo para subir o bajar la prioridad.

# Encriptación por la línea de comandos

Además de poder encriptar y desencriptar desde la interfaz gráfica de la aplicación, también es posible invocar a las funciones de encriptar y desencriptar mediante la línea de comandos.

Esto no es muy recomendable, ya que para encriptar/desencriptar hay que poner la password en claro, y es posible que alguien pueda verla mirando la historia de comandos, o incluso viendo los procesos que están ejecutándose en el sistema. También es posible que tengas que teclear la contraseña y que esta se vea en tu pantalla, por lo que no es un sistema muy seguro.

Sin embargo es posible hacerlo.

Existe un directorio (\_scripts) en el que se han generado unos scripts para facilitar la tarea.

Hay dos versiones de los scripts una para Windows (scripts \*.bat) y otra que sirve tanto para Linux como para Mac (scripts \*.sh)

## Scripts para Windows

La versión para Windows de los scripts se encuentra en el directorio:

...\\_scripts\windows

Los scripts disponibles son estos:

* command.interface.FileEncoder.bat
* decodeFile.bat
* encodeFile.bat
* example.decodeFile.bat
* example.encodeFile.bat

Para invocar a estos scripts, hay que hacerlo desde el mismo directorio en el que se encuentran ubicados.

El primer script contiene la llamada a la aplicación de Java, recogiendo los argumentos que le son pasados y añadiéndolos a los argumentos para la aplicación de Java.

El segundo script contiene los comandos para desencriptar un fichero encriptado. Toma como parámetros el nombre del fichero encriptado y la contraseña.

El tercer script contiene los comandos para encriptar un fichero. Toma como parámetros el nombre del fichero y la contraseña.

El cuarto script es un ejemplo de cómo invocar al script que encripta.

El quinto script es un ejemplo de cómo invocar al script que desencripta.

## Scripts para Linux y para Mac

La versión para Linux y Mac de los scripts se encuentra en el directorio:

.../\_scripts/Mac.or.Linux

En un principio los scripts están comprimidos en el fichero: scripts.tar.gz

Hay que descomprimir este archivo para poder trabajar con los scripts. Para ello, desde el directorio donde se encuentra el archivo scripts.tar.gz, teclear el siguiente comando:

tar -xvzf scripts.tar.gz

Una vez descomprimidos en el directorio tendremos los siguientes scripts:

* command.interface.FileEncoder.sh
* decodeFile.sh
* encodeFile.sh
* example.decodeFile.sh
* example.encodeFile.sh

Para invocar a estos scripts, hay que hacerlo desde el mismo directorio en el que se encuentran ubicados.

El primer script contiene la llamada a la aplicación de Java, recogiendo los argumentos que le son pasados y añadiéndolos a los argumentos para la aplicación de Java.

El segundo script contiene los comandos para desencriptar un fichero encriptado. Toma como parámetros el nombre del fichero encriptado y la contraseña.

El tercer script contiene los comandos para encriptar un fichero. Toma como parámetros el nombre del fichero y la contraseña.

El cuarto script es un ejemplo de cómo invocar al script que encripta.

El quinto script es un ejemplo de cómo invocar al script que desencripta.

## Parámetros de encriptación como argumentos

De los scripts anteriores el principal es este:

command.interface.FileEncoder

Desde él se invoca a la aplicación Java y como parámetros toma principalmente los que ya se han descrito en el punto: 3.6.1-Parámetros de la configuración de encriptación

Los parámetros disponibles para encriptar/desencriptar son estos:

* -password Seguido por un argumento con la password
* -encodedFileName Seguido por un argumento con el nombre de fichero encriptado.
* -decodedFileName Seguido por un argumento con el nombre de fichero desencriptado.
* -encode o -decode Para seleccionar entre encriptación y desencriptación.
* -fileEncoderType Seguido por un argumento con el identificador del fileEncoderType (1 obsoleto or 2) a utilizar en la encriptación (sólo para encriptación)
* -sizeOfNumbersSimpleEncoder Seguido por el tamaño de los números para el paso de encriptación 1 (XOR) (sólo para encriptación).
* -sizeOfNumbersReordererEncoder Seguido por el tamaño de los números para el paso 2 (reordenación) (sólo para encriptación).
* -numberOfBitsPerIterationSimpleEncoder Seguido por el número de bits devueltos por el generador pseudoaleatorio en cada iteración para el paso 1 (XOR) (sólo para encriptación).
* -numberOfBitsPerIterationReordererEncoder Seguido por el número de bits devueltos por el generador pseudoaleatorio en cada iteración para el paso 2 (reordenación). (sólo para encriptación).
* -numBytesFileSlice Seguido por el número de bytes de la rodaja. Se recomienda utilizar un valor alto para este paramétro. La memoria utilizada por la aplicación está limitado aproximadamente por 100 veces el tamaño de rodaja. (sólo para encriptación).
* -useFileSizeForEncryptingParams Si este parámetro está presente, los parámetros de encriptación (definidos por los 5 parámetros anteriores) serán calculados basándose en el tamaño del fichero a encriptar y los 5 parámetros anteriores serán sobreescritos.

# Método de encriptación

Los algoritmos de encriptación y desencriptación son muy similares, e idénticos en tiempo de computación (encriptación simétrica).

En los siguientes puntos se verá un resumen del método de encriptación utilizado en la aplicación.

## Cabecera del fichero encriptado

Se calcula una hash de la contraseña de encriptación (SHA-256) que devuelve una clave de 256 bits que llamaremos clave-1.

El tamaño de clave-1 es siempre el mismo.

Si utilizamos el generador pseudoaleatorio con diferentes parámetros de configuración, es posible que una clave de 256 bits no sea óptima.

El tamaño óptimo de la clave a ser utilizada para inicializar los generadores pseudoaleatorios para la encryptación es calculado y lo llamamos tamaño-2

El micrófono es utilizado como fuente de bytes aleatorios, y se cogen de él tamaño-2 bytes que serán utilizados para formar la clave-2 que será utilizada para encriptar.

**Por ello es importante tener el micrófono conectado**, para poder obtener bytes para la clave lo más aleatorios posible.

Si el micrófono no está conectado, estos bytes aleatorios son obtenidos utilizando la clase SecureRandom de Java, utilizando como mínimo un byte casi aleatorio como semilla.

Esto no es mucho, por lo que es muy importante tener conectado el micrófono para poder obtener bytes realmente aleatorios.

La clave-2 se utiliza para encriptar el fichero.

La clave-1 se utiliza para encriptar la clave-2

La clave-2 encriptada, los parámetros de configuración de la encriptación (sin encriptar) y una hash (MD5) (encriptada) del fichero original serán escritos en la cabecera del fichero encriptado.

Esta información será utilizada para desencriptar el fichero. La hash MD5 se utilizará para comprobar que el proceso de desencriptación ha ido correctamente.

## Algoritmo de encriptación

Esta aplicación realiza la encriptación de ficheros en rodajas, esto es dividiendo el fichero en partes más pequeñas. Para cada encriptar cada una de las rodajas se siguen dos pasos:

* Encriptación de la rodaja estilo XOR. En esta parte de la encriptación, generando bits pseudoaleatorios y haciendo un XOR con los datos originales, se obtiene una rodaja encriptada cuyos bytes son bastante aleatorios.

Si dejáramos aquí la encriptación, sería relativamente rápido intentar romper esta encriptación si se conocen algunos bytes del inicio del fichero sin encriptar, como es posible que se conozcan en el caso de ficheros que puedan abrirse con alguna aplicación (por ejemplo Word o Excel), ya que generalmente esos ficheros tendrán algún tipo de cabecera que hace la encriptación de este tipo más vulnerable.

Debido a ello la aplicación genera un segundo paso en la encriptación:

* Reordenación pseudoaleatoria de los bytes de la rodaja. En esta parte de la encriptación, se generan posiciones pseudoaleatorias, donde se van a ir moviendo uno a uno los bytes de la rodaja, reordenando el conjunto de los bytes de la rodaja.

Para obtener los bytes iniciales del fichero desencriptado, es necesario desencriptar una rodaja completa, por lo que eso hace que se haga más difícil romper la encriptación, ya que es mucho más costoso en térmimos de tiempo de computación desencriptar una rodaja completa (cuyo tamaño es configurable, y por defecto está a 256.000 bytes), que desencriptar los 100 o 1000 primeros bytes y compararlos con una cabecera válida de word o excel , por ejemplo.

Cada uno de los pasos anteriores, para realizar su parte de la encriptación utilizan un generador pseudoaleatorio caótico, que en la implementación actual están caracterizados básicamente por un tamaño de los números a utilizar en el generador y el número de bits pseudoaleatorios a devolver en cada iteración. Dependiendo de estos dos parámetros el generador pseudoaleatorio calcula el número de bytes óptimo que debe tener la contraseña.

Cuanto mayor sea el tamaño de los números, mayor será el tamaño óptimo de la contraseña con la que hay que inicializar el generador pseudoaleatorio, y mayor será el tiempo de computación de cada iteración. Así pués, un mayor tamaño de los números hace que el tiempo de encriptación/desencriptación sea mayor. El efecto sobre el tiempo de computación de este parámetro es mayor cuanto menor sea el número de bits por iteración del generador pseudoaleatorio.

La configuración ideal sería una que produjera una encriptación robusta y que fuera rápida.

Para conseguir este objetivo, podríamos utilizar tamaños de número grandes (más robusto) con número alto de bits devueltos por iteración (más rápido) y combinado con tamaños de rodaja lo más grandes posible (mucho más robusto, pero cuanto mayor es este número, más memoria se utiliza).

Para ficheros pequeños se debería utilizar un número muy pequeño de bits devueltos por iteración (por ejemplo: 1) y un tamaño de números grande. El tamaño de la rodaja debería ser como mínimo igual al tamaño del fichero a encriptar.

## Generador pseudoaleatorio utilizado

El generador pseudoaleatorio utilizado en esta aplicación, es un generador pseudoaleatorio basado en secuencias caóticas.

Es un generador pseudoaleatorio muy sencillo y de muy fácil implementación.

La descripción de este generador pseudoaleatorio fue obtenida del libro:

"Secuencias pseudoaleatorias para telecomunicaciones" Edicions UPC (1996) Ernesto J. Forner Cruselles y José L. Melús Moreno

Un agradecimiento para ellos por escribir un libro tan interesante.

Citando del propio libro:

"

*Este generador, propuesto por M. Romera, I. Jiménez y J. Negrillo [ROM90], está basado*

*también en la generación de secuencias pseudoaleatorias mediante el empleo de funciones caóticas.*

*[ROM90] ROMERA, M.; JIMENEZ, I. NEGRILLO, J. Generación de Secuencias Cifrantes Mediante*

*Funciones Caóticas. I Reunión Española de Criptología. Mallorca, 1990.*

*...*

*La secuencia caótica de este generador (de números reales) está definida por:*

*Xo = 0*

*Xi+1 = Xi2 + K*

*Si -2 <= K <= 0,25, entonces los números de esta secuencia están acotados: -2 <= Xi <= 2*

*Unas secuencias son cíclicamente convergentes (periodos 1, 2, 3 ...) y otras son caóticas.*

*Cuando K es muy próximo a -2, por ejemplo K = -1,99999XXXXXXXXXXX, las series que se obtienen a partir de la iteración real de Mandelbrot son casi siempre caóticas.*

*...*

*La secuencia real obtenida puede convertirse fácilmente en una secuencia pseudoaleatoria binaria, tomando el signo de cada número de la serie (la distribución es simétrica respecto a cero), o aplicando un criterio de paridad a los dígitos de cada número de la serie.*

"

Basándonos en este generador de secuencias caóticas, hemos creado nuestro generador muy similar a este, que a grandes rasgos cumple con las siguientes condiciones:

-1 <= Xo  <= 1

K = -1,99999999xxxxxxxxxxxxxxxxxxxxxxxx

Para inicializar el generador pseudoaleatorio necesitaremos una semilla (un número de bytes), que utilizaremos para inicializar Xo y la parte variable de K

Tanto X como K deben ser números de precisión configurable que permita una precisión muy grande, ya que si utilizamos los tipos básicos de números de mayor precisión que existen en Java (double), el rango de la semilla sería muy limitado y por fuerza bruta, podría romperse el sistema pseudoaleatorio.

Las opciones que tenemos son:

* Utilizar la clase BigDecimal de Java (Utilizado en el FileEncoderType=1, obsoleto)
* Programar una clase nueva que maneje números de precisión variable (Utilizado en el FileEncoderType=2).

Los bits pseudoaleatorios que se derivan de la secuencia caótica son obtenidos de estas maneras:

* Cuando el número de bits producidos por iteración es 1, 2 o 4 (FileEncoderType 1 y 2):

Se cuentan los bits "1" del Xi y se hace el módulo (2 si el número de bits por iteración era 1, 4 si era 2 o 16 si era 4).

Esta es la versión más robusta del generador pseudoaleatorio (más costosa en tiempo de computación, y menos dependiente del valor concreto de Xi)

De esa forma se obtienen los bits pseudoaleatorios de cada iteración de la secuencia caótica.

* Si se utiliza 1 bit por iteración, no hay limitación de tamaño de número, ya que los posibles valores producidos (0 o 1) son siempre equiprobables independientemente del número de bits utilizado.
* Si se utilizan 2 bits por iteración, es mejor utilizar números de tamaños superiores o iguales a 16 bits, ya que por ejemplo haciendo el módulo 4 de la cuenta de bits 1 de un número caótico 8 bits (resultando 0, 1, 2 o 3), esos valores no serían equiprobables, pero con tamaños de número de 16 bits, la diferencia de probabilidad ya es despreciable.
* Si se utilizan 4 bits por iteración, es mejor utilizar números de tamaños superiores o iguales a 256 bits.

En la siguiente tabla pueden verse las probabilidades de cada valor posible en función del tamaño del número cuando se producen 4 bits pseudoaleatorios por iteración:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Valor | números de 16 bits | números de 32 bits | números de 64 bits | números de 128 bits | números de 192 bits | números de 256 bits |
| 0 | 0,00305176% | 13,99499346% | 9,93997199% | 7,29371224% | 6,55137753% | 6,33706399% |
| 1 | 0,02441406% | 13,17175925% | 9,64196694% | 7,21415681% | 6,52843586% | 6,33043664% |
| 2 | 0,18310547% | 10,97647697% | 8,80338313% | 6,98766512% | 6,46310389% | 6,31156353% |
| 3 | 0,85449219% | 8,08803737% | 7,57614083% | 6,64887061% | 6,36532878% | 6,28331793% |
| 4 | 2,77709961% | 5,25798649% | 6,17124246% | 6,24950378% | 6,24999687% | 6,24999998% |
| 5 | 6,66503906% | 3,00877392% | 4,81247919% | 5,85042763% | 6,13466680% | 6,21668205% |
| 6 | 12,21923828% | 1,52314156% | 3,69661687% | 5,51233488% | 6,03689611% | 6,18843647% |
| 7 | 17,45605469% | 0,73142946% | 2,96941304% | 5,28654495% | 5,97156857% | 6,16956339% |
| 8 | 19,63806152% | 0,48979651% | 2,71754309% | 5,20728020% | 5,94862872% | 6,16293605% |
| 9 | 17,45605469% | 0,73142946% | 2,96941304% | 5,28654495% | 5,97156857% | 6,16956339% |
| 10 | 12,21923828% | 1,52314156% | 3,69661687% | 5,51233488% | 6,03689611% | 6,18843647% |
| 11 | 6,66503906% | 3,00877392% | 4,81247919% | 5,85042763% | 6,13466680% | 6,21668205% |
| 12 | 2,77709961% | 5,25798649% | 6,17124246% | 6,24950378% | 6,24999687% | 6,24999998% |
| 13 | 0,85449219% | 8,08803737% | 7,57614083% | 6,64887061% | 6,36532878% | 6,28331793% |
| 14 | 0,18310547% | 10,97647697% | 8,80338313% | 6,98766512% | 6,46310389% | 6,31156353% |
| 15 | 0,02441406% | 13,17175925% | 9,64196694% | 7,21415681% | 6,52843586% | 6,33043664% |

Como se puede observar en la tabla, si el tamaño de los números es pequeño no obtendremos unos bits retornados lo suficiente equiprobables como para encriptar.

* Cuando el número de bits producidos por iteración es un múltiplo de 8 entre 8 y 64 (ambos incluidos) (FileEncoderType 2):

En ese caso se toman bytes completos que forman parte de Xi para utilizarlos como si fueran números pseudoaleatorios. Esto no es exacto, pero para que vaya más rápido hay que sacar los bits pseudoaleatorios de algún sitio. Seguramente no es tan robusto como contar el número de bits "1" del número Xi, pero creo que puede servir.

Obtener los bits pseudoaleatorios por este método es muy rápido, sólo hay que coger los bytes pseudoaleatorios directamente. Además, permite coger un número grande de bits por iteración.

La condición que debe cumplirse entre el número de bits producidos por iteración y el tamaño en bloques (bloques de 31 bits, cada bloque se almacena en un long, que se almacena en un array de longs de longitud variable) es esta:

NumeroBitsPorIteracion / 8 <= 3 x NumeroBloquesPorNumero

Ya que se cogen como mucho 3 bytes de cada bloque.