

Copyright 2015 Francisco Javier Merchán Macías.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

**Francisco Javier Merchán Macías**

Máster Universitario de Investigación en Ingeniería y Arquitectura, especialidad en Tecnologías Informáticas y de Comunicaciones.

[javier@javiermerchan.com](mailto:javier@javiermerchan.com)

+34 XXXXXXXXX



# Seguridad de la información

## Práctica 2

Profesor: Juan Arias Masa

### Manual de usuario

**Francisco Javier Merchán Macías**

Máster Universitario de Investigación en Ingeniería y Arquitectura, especialidad en Tecnologías  
Informáticas y de Comunicaciones.

[javier@javiermerchan.com](mailto:javier@javiermerchan.com)

+34 XXXXXXXXX



# Práctica 2

Francisco Javier Merchán Macías

Máster Universitario de Investigación en  
Ingeniería y Arquitectura, especialidad en  
Tecnologías Informáticas y de Comunicaciones.

---

## ÍNDICE

---

1. Sobre la práctica.....	5
Objetivos .....	5
Requisitos .....	5
Documentación .....	5
Se pide.....	5
2. Algoritmo de cifrado Gronsfield (práctica 1) .....	5
3. Algoritmo de cifrado DES simplificado .....	6
4. Funcionamiento del programa.....	7
Ejecutar el programa por primera vez .....	7
Obtener ayuda .....	8
Opción de prueba de cifrado y descifrado del algoritmo Gronsfield.....	8
Ejecutar el programa cargando el fichero de configuración.....	9
Ejemplo de cifrado Gronsfield.....	10
Ejemplo de descifrado Gronsfield.....	11
Ejemplo de cifrado DES simplificado.....	12
Ejemplo de descifrado DES simplificado .....	14
Ejemplo de cifrado DES simplificado con traza activa.....	15
Ejemplo de descifrado DES simplificado con traza activa .....	18
Errores en el fichero de configuración .....	21

5. Conclusiones .....	22
6. Bibliografía y referencias utilizadas.....	23

## 1. Sobre la práctica

### Objetivos

- Utilizar el cifrado y descifrado Gronsfeld (añadido de la práctica 1).
- Utilizar el cifrado y descifrado DES simplificado.

### Requisitos

- PC/Mac con sistema operativo Windows u OSX con Java instalado.
- Java JRE versión 8 update 60.

### Documentación

- Se incluye la aplicación en formato .JAR (P2\_SI\_2015\_64\_frmerchan.jar), la cual se podrá ejecutar con el siguiente comando:

```
java -jar P2_SI_2015_64_frmerchan.jar
```

- Se incluye la documentación de programador en formato JAVADOC.
- Manuales de usuario y programador.

### Se pide

Implementar un programa codificado en JAVA, que ha de funcionar como codificador o decodificador según lo que tenga declarado el fichero de configuración (permitirá usar los algoritmos Gronsfeld y DES simplificado). El nombre del fichero de configuración se obtiene como parámetro del programa. Desde el fichero de configuración se especificará el fichero a codificar o descodificar, clave de cifrado o descifrado, salida en pantalla o fichero de salida con el proceso del programa.

Se pretende codificar o decodificar un texto que vendrá escrito en un fichero de texto.

## 2. Algoritmo de cifrado Gronsfeld (práctica 1)

El algoritmo de cifrado está basado en el cifrado de Vigenère y es un cifrado del tipo polialfabéticos. Esto significa que se usa más de un alfabeto de cifrado.

El nombre del cifrado se debe a que el autor de dicho cifrado fue el diplomático belga José de Bronckhorst, conde de Gronsfeld, y se creó sobre el año 1744. Se trata de un cifrado resistente al análisis de frecuencia. Este cifrado fue roto por Friedrich Kasiski en 1863.

El cifrado se basa en el uso de alfabetos desplazados, conformando una tabla de cifrado. Esta tabla está conformada en sus filas por los números de la clave y en cada columna por los caracteres del alfabeto que pretendemos cifrar.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
1	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
2	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
3	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
4	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
5	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
6	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
7	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
8	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
9	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Tabla 1. Tabla de cifrado Gronsfeld.

El código de cifrado corresponde con la primera columna y va desde el 0 al 9.

Los caracteres a cifrar corresponden con el índice de cada columna y que corresponden al alfabeto en inglés y corresponden en los siguientes caracteres (ABCDEFGHIJKLMNOPQRSTUVWXYZ).

El proceso de cifrado sería el siguiente:

- Clave: 924
- Texto a cifrar: cifrando
- Criptograma: ENQTFYFT

En el caso de encontrar caracteres no incluidos en el grupo de caracteres no cifrables, la clave de cifrado no rotará hasta el momento que encuentre un carácter cifrable. A modo de ejemplo sería lo siguiente:

- Clave: 924
- Texto a cifrar: cifrando '924'. Quiero seguridad.
- Criptograma: ENQTFYFT 924. BWNPTT DGLFTNOCI.

No cifrará caracteres como el espacio, números y caracteres de puntuación.

### 3. Algoritmo de cifrado DES simplificado

El algoritmo de cifrado DES simplificado es una versión reducida del algoritmo DES. Se trata de un algoritmo de cifrado por bloques. Tiene una estructura Feitsel de 2 vueltas, y tanto al principio como al final del algoritmo realiza una función de permutación.

Este algoritmo DES simplificado fue creado por el profesor Edward Shaefer de la Universidad de Santa Clara, para mostrar de una manera sencilla el funcionamiento general de DES, se basa totalmente en el funcionamiento de DES pero con menos procesos, se ingresa un texto en Bits (8 bits) y una clave (10 bits). El siguiente esquema describe el proceso de cifrado y descifrado que emplea el DES simplificado (Ilustración 1). (Gama-Moreno & Mejía-Mancilla, 2009)

En la Ilustración 2 se muestra el esquema de generación de claves para el algoritmo S-DES. Su funcionamiento es similar al DES, emplea módulos iguales pero simplificados para la fácil visualización y comprensión del algoritmo.

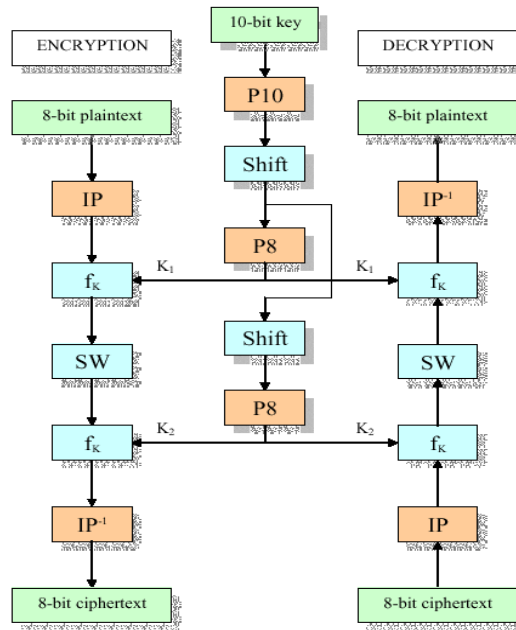


Ilustración 1 Generación de claves para el SDES

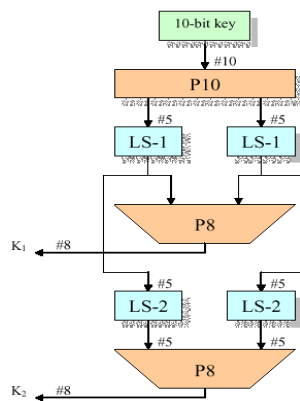


Ilustración 2 Generación de claves para el SDES.

#### 4. Funcionamiento del programa

Ejecutar el programa por primera vez

El funcionamiento del programa es muy sencillo. Debemos ejecutar el siguiente comando:

```
java -jar P2 SI 2015 64 frmerchan.jar
```

Ejecutando el comando anterior obtendremos la siguiente salida:

```

C:\> Símbolo del sistema
F:\Pruebas>java -jar P2_SI_2015_64_frmerchan.jar
No has tecleado parámetros... Usa: 'java -jar P2_SI2015_64.jar -h' para obtener ayuda.
F:\Pruebas>

```

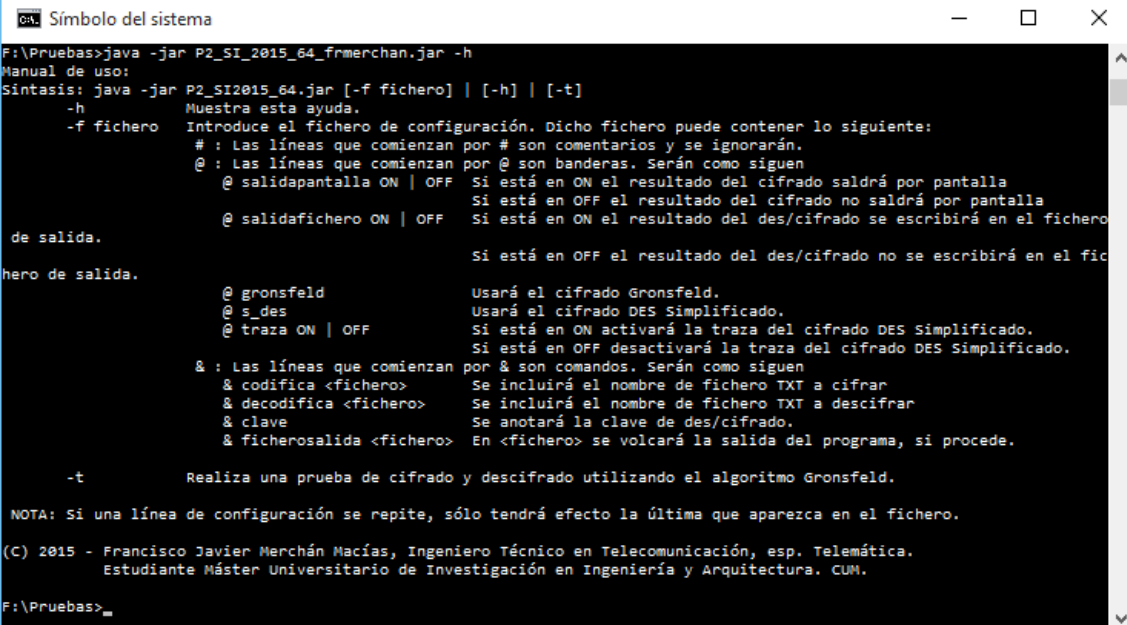
Esta salida nos dará información para obtener ayuda para poder usar el programa.

## Obtener ayuda

Para ello ejecutaremos el siguiente comando

```
java -jar P2_SI_2015_64_frmerchan.jar -h
```

Y obtendremos la siguiente información a modo de ayuda.



```

F:\Pruebas>java -jar P2_SI_2015_64_frmerchan.jar -h
Manual de uso:
Sintaxis: java -jar P2_SI2015_64.jar [-f fichero] | [-h] | [-t]
-h          Muestra esta ayuda.
-f fichero  Introduce el fichero de configuración. Dicho fichero puede contener lo siguiente:
             # : Las líneas que comienzan por # son comentarios y se ignorarán.
             @ : Las líneas que comienzan por @ son banderas. Serán como siguen
                 @ salidapantalla ON | OFF Si está en ON el resultado del cifrado saldrá por pantalla
                                           Si está en OFF el resultado del cifrado no saldrá por pantalla
                 @ salidafichero ON | OFF  Si está en ON el resultado del des/cifrado se escribirá en el fichero
                                           Si está en OFF el resultado del des/cifrado no se escribirá en el fichero de salida.
             @ gronsfeld                   Usará el cifrado Gronsfeld.
             @ s_des                       Usará el cifrado DES Simplificado.
             @ traza ON | OFF              Si está en ON activará la traza del cifrado DES Simplificado.
                                           Si está en OFF desactivará la traza del cifrado DES Simplificado.
             & : Las líneas que comienzan por & son comandos. Serán como siguen
                 & codifica <fichero>      Se incluirá el nombre de fichero TXT a cifrar
                 & decodifica <fichero>     Se incluirá el nombre de fichero TXT a descifrar
                 & clave                   Se anotará la clave de des/cifrado.
                 & ficherosalida <fichero> En <fichero> se volcará la salida del programa, si procede.
-t          Realiza una prueba de cifrado y descifrado utilizando el algoritmo Gronsfeld.

NOTA: Si una línea de configuración se repite, sólo tendrá efecto la última que aparezca en el fichero.

(C) 2015 - Francisco Javier Merchán Macías, Ingeniero Técnico en Telecomunicación, esp. Telemática.
          Estudiante Máster Universitario de Investigación en Ingeniería y Arquitectura. CUM.

F:\Pruebas>
  
```

## Opción de prueba de cifrado y descifrado del algoritmo Gronsfeld

He implementado como opción extra al programa, una prueba de cifrado y descifrado del algoritmo Gronsfeld. Para lanzarla debemos ejecutar el siguiente comando:

```
java -jar P1_SI_2015_64_frmerchan.jar -t
```

Obtendremos la siguiente salida, la cual muestra distintas pruebas de cifrado y descifrado utilizando el algoritmo Gronsfeld:



```

C:\> Símbolo del sistema
F:\Pruebas>java -jar P2_SI_2015_64_frmerchan.jar -t

-----
---- TESTEO DEL DES-CIFRADO GRONSFELD ----
-----

- PruebaGronsfeld-Cifrado carácter: Letra de entrada: Q, con valor de contraseña: 7. Letra codificada: J
- PruebaGronsfeld-Descifrado carácter: Letra de entrada: Q, con valor de contraseña: 7. Letra descodificada: X
- El texto a cifrar es: 'seguridad prueBas@desEgUri/dad.es||ek12345'
  La clave de des-cifrado es: 2587
  El texto cifrado es: 'XRDNMVATI CONJOKL@IRPXLHOB/INA.XX||RH12345'
  El texto descifrado despues de cifrar es: 'SEGURIDAD PRUEBAS@DESEGURI/DAD.ES||EK12345'
- El texto a cifrar es 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.
  - El texto cifrado con clave 0 es: 'CDEFGHIJKLMNOPQRSTUVWXYZAB'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 1 es: 'DEFGHIJKLMNOPQRSTUVWXYZABC'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 2 es: 'FGHIJKLMNOPQRSTUVWXYZABCDE'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 3 es: 'HIJKLMNOPQRSTUVWXYZABCDEF'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 4 es: 'LMNOPQRSTUVWXYZABCDEFGHIJK'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 5 es: 'NOPQRSTUVWXYZABCDEFGHIJKLM'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 6 es: 'RSTUVWXYZABCDEFGHIJKLMNOPQ'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 7 es: 'TUVWXYZABCDEFGHIJKLMNQRS'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 8 es: 'XYZABCDEFGHIJKLMNQRSTUWV'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 9 es: 'CDEFGHIJKLMNOPQRSTUVWXYZAB'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - El texto cifrado con clave 0123456789 es: 'CEHKPSXAFLMORUZHCKPVMYBEJM'
    El texto descifrado es: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

F:\Pruebas>

```

### Ejecutar el programa cargando el fichero de configuración

Para ejecutar el programa utilizando un fichero de configuración, podremos ejecutar el siguiente comando:

```
java -jar P2 SI 2015 64 frmerchan.jar -f fichconf.txt
```

En este momento ejecutaremos el programa cargando un fichero de configuración llamado 'fichconf.txt'. En dicho fichero de configuración declararemos las opciones que usará el programa. Las opciones disponibles serán las siguientes:

- Carácter '#': Será para incluir comentarios en el fichero de configuración.
- El carácter '@': Sirve para incluir banderas. Las banderas disponibles son las siguientes:
  - '@ salidapantalla': tenemos las opciones 'ON' y 'OFF' para activar o desactivar la salida del programa en pantalla.
  - '@ salidafichero': tenemos las opciones 'ON' y 'OFF' para activar o desactivar la salida del programa y guardarla en un fichero.
  - '@ gronsfeld': Usará el cifrado Gronsfeld.
  - '@ s\_des': Usará el cifrado DES simplificado.
  - '@ traza': tendremos las opciones 'ON' y 'OFF' para activar o desactivar la traza del algoritmo DES simplificado.
- El carácter '&': Sirve para incluir comandos. Los comandos disponibles son las siguientes:
  - '& codifica <fichero>': donde podremos incluir el nombre de un <fichero> donde tenemos el texto que queremos cifrar. Su salida podrá salir en pantalla y/o fichero.
  - '& decodifica <fichero>': donde podremos incluir el nombre de un <fichero> donde tenemos el texto que queremos descifrar. Su salida podrá salir en pantalla y/o fichero.

- '& clave': Se anotará la clave de cifrado/descifrado que se utilizará para cifrar o descifrar el fichero de entrada.
- '& ficherosalida <fichero>': Se guardará en el <fichero> la salida del programa, incluida la salida cifrada o descifrada.

El programa detectará que opción se encuentra en último lugar a la hora de '& codificar <fichero>' o '& decodificar <fichero>', es decir, si a la hora de cargar el fichero de configuración, el programa encuentra dos líneas '& codifica <fichero>' o '& decodifica <fichero>', se tendrá en cuenta la última línea que encontremos, dando igual si el programa encuentra varias líneas iguales.

### Ejemplo de cifrado Gronsfield

Podemos ver un ejemplo de fichero de configuración:

```
@ salidapantalla ON

@ salidafichero ON

& ficherosalida fichero_descodificar.txt

& descodifica fichero_descodificar.txt
& codifica fichero_codificar.txt

@ s_des
@ gronsfeld

#& clave 2587
#& clave 1111011010
& clave 924

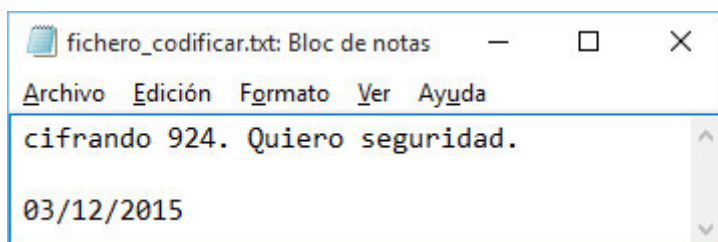
@ traza OFF
```

Ejecutaremos el siguiente comando:

```
java -jar P2_SI_2015_64_frmerchan.jar -f config_cod.txt
```

En este ejemplo, el programa mostrará su salida en pantalla y se guardará en el fichero 'fichero\_descodificar.txt'. También realizará la codificación del fichero 'fichero\_codificar.txt'. Si se activa la opción de '@ salidafichero ON' y no ponemos la opción '& ficherosalida <fichero>', se definirá un nombre por defecto al fichero de salida.

El fichero 'fichero\_codificar.txt' es el siguiente:



Obtendremos la siguiente salida en pantalla:

```

C:\ Símbolo del sistema
F:\Pruebas>java -jar P2_SI_2015_64_frmerchan.jar -f config_cod.txt

---- PROGRAMA DE DES-CIFRADO CREADO PARA LA ASIGNATURA 'SEGURIDAD DE LA INFORMACIÓN' - CURSO 2015-2016 ----
- Bienvenid@ al programa de la segunda práctica de Seguridad de la información, correspondiente a la -
- asignatura SEGURIDAD DE LA INFORMACIÓN del MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA Y -
- ARQUITECTURA, ESPECIALIDAD EN TECNOLOGÍAS INFORMÁTICAS Y DE COMUNICACIONES. -
- Utilizamos los algoritmos: Gronsfield y DES simplificado. -
---- (C) 2015 FRANCISCO JAVIER MERCHÁN MACÍAS - INGENIERO TÉCNICO EN TELECOMUNICACIÓN, ESP. TELEMÁTICA ----

- Salida a pantalla activada.
- Salida a fichero activada.
- Vamos a CIFRAR el fichero fichero_codificar.txt
- Clave de cifrado: '924'
- El fichero de salida es: fichero_descodificar.txt
- Usamos el cifrado Gronsfield.
- La traza está desactivada.

-- INICIO DEL PROCESO DE CIFRADO A CONTINUACIÓN -----
ENQTFYFT 924. BWNPTT DGLFTNOCI.

03/12/2015

-- FIN DEL PROCESO DE CIFRADO -----

>>> Hasta pronto... <<<

F:\Pruebas>

```

Y la siguiente salida en el fichero:

```

fichero_descodificar.txt: Bloc de n...
Archivo Edición Formato Ver Ayuda
ENQTFYFT 924. BWNPTT DGLFTNOCI.

03/12/2015

```

### Ejemplo de descifrado Gronsfield

En este caso utilizaremos el siguiente fichero de configuración:

```

@ salidapantalla ON

@ salidafichero ON

& ficherosalida fichero_codificar.txt

& codifica fichero_codificar.txt
& descodifica fichero_descodificar.txt

@ s_des
@ gronsfeld

# & clave 2587
# & clave 1111011010
& clave 924

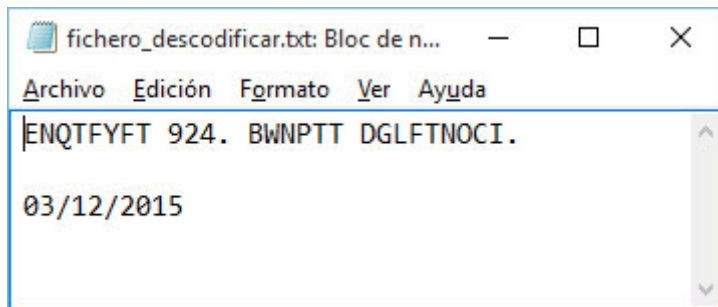
@ traza OFF

```

Ejecutaremos el siguiente comando:

```
java -jar P2 SI 2015 64 frmerchan.jar -f config desc.txt
```

El fichero a decodificar es la salida de la codificación anterior. Es el siguiente:



Y obtendremos la siguiente salida en pantalla:

```

C:\. Símbolo del sistema
F:\Pruebas>java -jar P2_SI_2015_64_fmmerchan.jar -f config_desc.txt

---- PROGRAMA DE DES-CIFRADO CREADO PARA LA ASIGNATURA 'SEGURIDAD DE LA INFORMACIÓN' - CURSO 2015-2016 ----
-----
- Bienvenid@ al programa de la segunda práctica de Seguridad de la información, correspondiente a la -
- asignatura SEGURIDAD DE LA INFORMACIÓN del MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA Y -
- ARQUITECTURA, ESPECIALIDAD EN TECNOLOGÍAS INFORMÁTICAS Y DE COMUNICACIONES. -
- Utilizamos los algoritmos: Gronsfeld y DES simplificado. -
-----
---- (C) 2015 FRANCISCO JAVIER MERCHÁN MACÍAS - INGENIERO TÉCNICO EN TELECOMUNICACIÓN, ESP. TELEMÁTICA ----

- Salida a pantalla activada.
- Salida a fichero activada.
- Vamos a DESCIFRAR el fichero fichero_descodificar.txt
- Clave de descifrado: '924'
- El fichero de salida es: fichero_codificar.txt
- Usamos el cifrado Gronsfeld.
- La traza está desactivada.

-- INICIO DEL PROCESO DE DESCIFRADO A CONTINUACIÓN -----
CIFRANDO 924. QUIERO SEGURIDAD.

03/12/2015

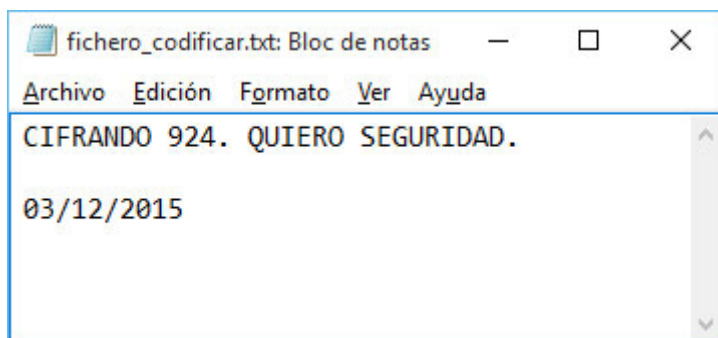
-- FIN DEL PROCESO DE DESCIFRADO -----

>>> Hasta pronto... <<<

F:\Pruebas>

```

Y obtendremos la siguiente salida en el fichero 'fichero\_codificar.txt'.



### Ejemplo de cifrado DES simplificado

Vamos a realizar el proceso de cifrado del mismo texto, pero con una clave de cifrado diferente al ejemplo de Gronsfeld. El fichero de configuración a usar será el siguiente:

```
@ salidapantalla ON

@ salidafichero ON
```

```
& ficherosalida fichero_descodificar.txt

& descodifica fichero_descodificar.txt
& codifica fichero_codificar.txt

@ gronsfeld
@ s_des

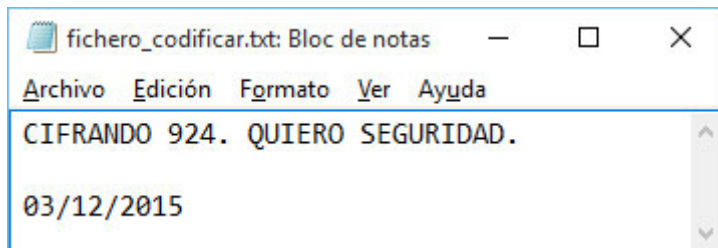
#& clave 2587
& clave 1111011010

@ traza OFF
```

Ejecutaremos el siguiente comando:

```
java -jar P2_SI_2015_64_frmerchan.jar -f config_cod.txt
```

El fichero a codificar es el siguiente:



Y obtendremos la siguiente salida en pantalla:

```

F:\Pruebas>java -jar P2_SI_2015_64_frmerchan.jar -f config_cod.txt

---- PROGRAMA DE DES-CIFRADO CREADO PARA LA ASIGNATURA 'SEGURIDAD DE LA INFORMACIÓN' - CURSO 2015-2016 ----
- Bienvenid@ al programa de la segunda práctica de Seguridad de la información, correspondiente a la -
- asignatura SEGURIDAD DE LA INFORMACIÓN del MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA Y -
- ARQUITECTURA, ESPECIALIDAD EN TECNOLOGÍAS INFORMÁTICAS Y DE COMUNICACIONES. -
- Utilizamos los algoritmos: Gronsfeld y DES simplificado. -
---- (C) 2015 FRANCISCO JAVIER MERCHÁN MACÍAS - INGENIERO TÉCNICO EN TELECOMUNICACIÓN, ESP. TELEMÁTICA ----

- Salida a pantalla activada.
- Salida a fichero activada.
- Vamos a CIFRAR el fichero fichero_codificar.txt
- Clave de cifrado: '1111011010'
- El fichero de salida es: fichero_descodificar.txt
- Usamos el cifrado DES Simplificado.
- La traza está desactivada.

-- INICIO DEL PROCESO DE CIFRADO A CONTINUACIÓN -----
11001001001011000001110000111111100001010110111101100100110001011101011000000110000100010010111110101100000111
10000010001011000100011100111110110001011101011000000100100011110001100100000100011111100101100101100001010110010
01001011

011101001001111011111101110111000001111111010000011011101001011101100011110

-- FIN DEL PROCESO DE CIFRADO -----

>>> Hasta pronto... <<<

F:\Pruebas>
```

Y obtendremos la siguiente salida en el fichero 'fichero\_descodificar.txt'.

### Ejemplo de descifrado DES simplificado

Vamos a realizar el proceso de descifrado de la salida del cifrado del ejemplo anterior. El fichero de configuración a usar será el siguiente:

```
@ salidapantalla ON

@ salidafichero ON

& ficherosalida fichero_codificar.txt

& codifica fichero_codificar.txt
& descodifica fichero_descodificar.txt

@ gronsfeld
@ s_des

# & clave 2587
& clave 1111011010

@ traza OFF
```

Ejecutaremos el siguiente comando:

```
java -jar P2 SI 2015 64 frmerchan.jar -f config desc.txt
```

El fichero a descodificar es el siguiente:

Y obtendremos la siguiente salida en pantalla:

```

F:\Pruebas>java -jar P2_SI_2015_64_frmerchan.jar -f config_desc.txt

---- PROGRAMA DE DES-CIFRADO CREADO PARA LA ASIGNATURA 'SEGURIDAD DE LA INFORMACIÓN' - CURSO 2015-2016 ----
- Bienvenid@ al programa de la segunda práctica de Seguridad de la información, correspondiente a la
- asignatura SEGURIDAD DE LA INFORMACIÓN del MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA Y
- ARQUITECTURA, ESPECIALIDAD EN TECNOLOGÍAS INFORMÁTICAS Y DE COMUNICACIONES.
- Utilizamos los algoritmos: Gronsfeld y DES simplificado.
-
---- (C) 2015 FRANCISCO JAVIER MERCHÁN MACÍAS - INGENIERO TÉCNICO EN TELECOMUNICACIÓN, ESP. TELEMÁTICA ----

- Salida a pantalla activada.
- Salida a fichero activada.
- Vamos a DESCIFRAR el fichero fichero_descodificar.txt
- Clave de descifrado: '1111011010'
- El fichero de salida es: fichero_codificar.txt
- Usamos el cifrado DES Simplificado.
- La traza está desactivada.

-- INICIO DEL PROCESO DE DESCIFRADO A CONTINUACIÓN -----
CIFRANDO 924. QUIERO SEGURIDAD.

03/12/2015

-- FIN DEL PROCESO DE DESCIFRADO -----

>>>  Hasta pronto...  <<<

F:\Pruebas>

```

Y obtendremos la siguiente salida en el fichero 'fichero\_codificar.txt'.

```

fichero_codificar.txt: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
CIFRANDO 924. QUIERO SEGURIDAD.
03/12/2015

```

Ejemplo de cifrado DES simplificado con traza activa

Vamos a cifrar un sólo carácter con la traza activa para observar el proceso completo que realiza el programa:

El carácter a cifrar será:

```

fichero_codificar.txt: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
@

```

El fichero de configuración para cifrar ese carácter será:



```
config_cod.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
@ salidapantalla ON
@ salidafichero ON
& ficherosalida fichero_descodificar.txt
& descodifica fichero_descodificar.txt
& codifica fichero_codificar.txt
@ gronsfeld
@ s_des
#& clave 2587
& clave 1111011010
@ traza ON|
```

Ejecutaremos el siguiente comando:

```
java -jar P2_SI_2015_64_frmerchan.jar -f config_cod.txt
```

Y obtendremos la siguiente salida en pantalla:



```

C:\ Símbolo del sistema
F:\Pruebas>java -jar P2_SI_2015_64_fmmerchan.jar -f config_cod.txt

---- PROGRAMA DE DES-CIFRADO CREADO PARA LA ASIGNATURA 'SEGURIDAD DE LA INFORMACIÓN' - CURSO 2015-2016 ----
-----
- Bienvenid@ al programa de la segunda práctica de Seguridad de la información, correspondiente a la
- asignatura SEGURIDAD DE LA INFORMACIÓN del MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA Y
- ARQUITECTURA, ESPECIALIDAD EN TECNOLOGÍAS INFORMÁTICAS Y DE COMUNICACIONES.
- Utilizamos los algoritmos: Gronsfeld y DES simplificado.
-----
---- (C) 2015 FRANCISCO JAVIER MERCHÁN MACÍAS - INGENIERO TÉCNICO EN TELECOMUNICACIÓN, ESP. TELEMÁTICA ----

- Salida a pantalla activada.
- Salida a fichero activada.
- Vamos a CIFRAR el fichero fichero_codificar.txt
- Clave de cifrado: '111011010'
- El fichero de salida es: fichero_descodificar.txt
- Usamos el cifrado DES Simplificado.
- La traza está activada.

-- INICIO DEL PROCESO DE CIFRADO A CONTINUACIÓN -----
##### Comenzamos a generar las claves...
-Entrada P10: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada P8: Bit0[0], Bit1[1], Bit2[1], Bit3[1], Bit4[1], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida P8: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
-Entrada P10: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada LS1 izq y der: Bit0[0], Bit1[1], Bit2[1], Bit3[1], Bit4[1], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida LS1: Bit0[0], Bit1[1], Bit2[1], Bit3[1], Bit4[1], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
-Entrada LS2 izq y der: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[0], Bit8[1], Bit9[1]
Salida LS2: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[0], Bit8[1], Bit9[1]
-Entrada P8: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[0], Bit8[1], Bit9[1]
Salida P8: Bit0[0], Bit1[1], Bit2[1], Bit3[0], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
-La clave K1 generada es: 1110101
-La clave K2 generada es: 01100111

##### Iniciamos cifrado del caracter: @, que corresponde al Plaintext [01000000]
-Entrada Plaintext: Bit0[0], Bit1[1], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
Salida IP izq y der: Bit0[1], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
-Iniciamos una Vuelta
-Entrada IP derecha o Permutador 1 vuelta derecha: Bit0[0], Bit1[0], Bit2[0], Bit3[0]
Salida EP: Bit0[0], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
-Entrada EP para hacer XOR: Bit0[0], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
Clave de entrada hacer XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
Salida XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
-Inicio cajas S0 y S1
-La posición de la Fila S0 es: 3
-La posición de la Columna S0 es: 3
-La posición de la Fila S1 es: 1
-La posición de la Columna S1 es: 2
-La salida de la caja S0 es [10] y la salida de la caja S1 es [01]
-Entrada SalidaCajas: Bit0[1], Bit1[0], Bit2[0], Bit3[1]
Salida P4: Bit0[0], Bit1[1], Bit2[0], Bit3[1]
-Entrada P4 para hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[1]
IP Izquierda hacer XOR: Bit0[1], Bit1[0], Bit2[0], Bit3[0]

```

```

C:\> Símbolo del sistema

-Entrada P8: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[0], Bit8[1], Bit9[1]
  Salida P8: Bit0[0], Bit1[1], Bit2[1], Bit3[0], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
-La clave K1 generada es: 11110101
-La clave K2 generada es: 01100111

##### Iniciamos cifrado del caracter: @, que corresponde al Plaintext [01000000]
-Entrada Plaintext: Bit0[0], Bit1[1], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
  Salida IP izq y der: Bit0[1], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
-Iniciamos una Vuelta
-Entrada IP derecha o Permutador 1 vuelta derecha: Bit0[0], Bit1[0], Bit2[0], Bit3[0]
  Salida EP: Bit0[0], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
-Entrada EP para hacer XOR: Bit0[0], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
  Clave de entrada hacer XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
  Salida XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
-Inicio cajas S0 y S1
-La posición de la Fila S0 es: 3
-La posición de la Columna S0 es: 3
-La posición de la Fila S1 es: 1
-La posición de la Columna S1 es: 2
-La salida de la caja S0 es [10] y la salida de la caja S1 es [01]
-Entrada SalidaCajas: Bit0[1], Bit1[0], Bit2[0], Bit3[1]
  Salida P4: Bit0[0], Bit1[1], Bit2[0], Bit3[1]
-Entrada P4 para hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[1]
  IP Izquierda hacer XOR: Bit0[1], Bit1[0], Bit2[0], Bit3[0]
  Salida XOR2: Bit0[1], Bit1[1], Bit2[0], Bit3[1]
-Salida Primer Paso Permutado: Bit0[0], Bit1[0], Bit2[0], Bit3[0], Bit4[1], Bit5[1], Bit6[0], Bit7[1]
-Iniciamos una Vuelta
-Entrada IP derecha o Permutador 1 vuelta derecha: Bit0[1], Bit1[1], Bit2[0], Bit3[1]
  Salida EP: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[1]
-Entrada EP para hacer XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[1]
  Clave de entrada hacer XOR: Bit0[0], Bit1[1], Bit2[1], Bit3[0], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
  Salida XOR: Bit0[1], Bit1[0], Bit2[0], Bit3[0], Bit4[1], Bit5[1], Bit6[0], Bit7[0]
-Inicio cajas S0 y S1
-La posición de la Fila S0 es: 2
-La posición de la Columna S0 es: 0
-La posición de la Fila S1 es: 2
-La posición de la Columna S1 es: 2
-La salida de la caja S0 es [00] y la salida de la caja S1 es [01]
-Entrada SalidaCajas: Bit0[0], Bit1[0], Bit2[0], Bit3[1]
  Salida P4: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
-Entrada P4 para hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
  IP Izquierda hacer XOR: Bit0[0], Bit1[0], Bit2[0], Bit3[0]
  Salida XOR2: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
-Salida Segundo Paso Permutado: Bit0[0], Bit1[0], Bit2[0], Bit3[1], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
=> El caracter cifrado es: 00010111

00010111

-- FIN DEL PROCESO DE CIFRADO -----

>>>  Hasta pronto...  <<<

F:\Pruebas>

```

Y obtendremos la siguiente salida en el fichero 'fichero\_descodificar.txt'.

```

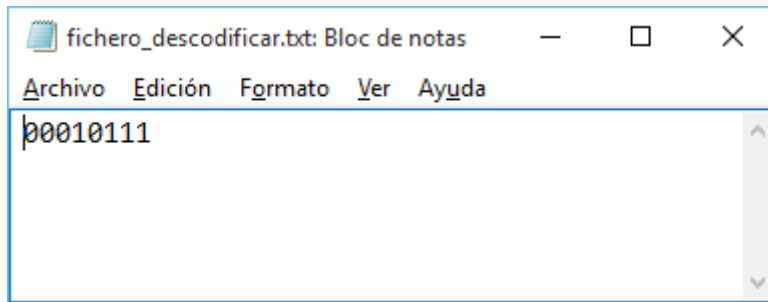
fichero_descodificar.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
00010111

```

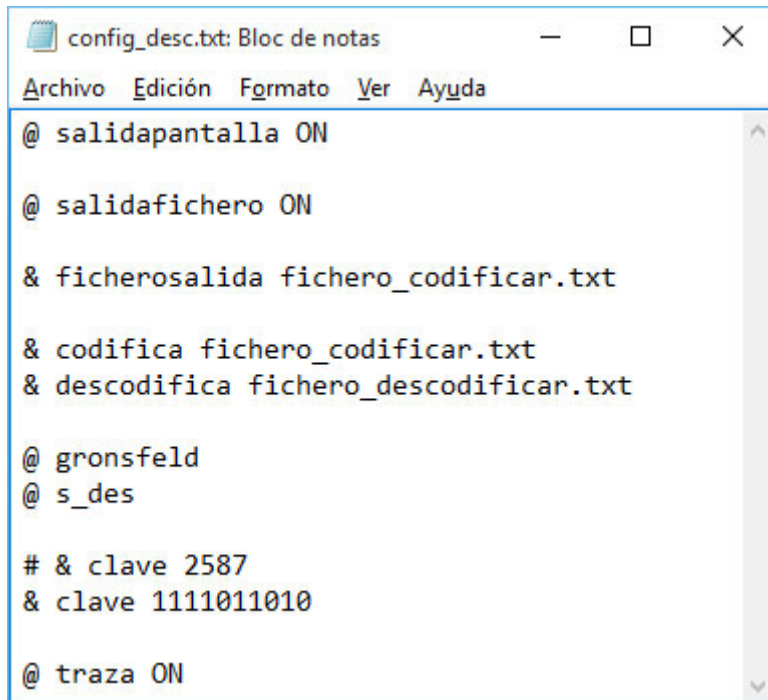
Ejemplo de descifrado DES simplificado con traza activa

Vamos a descifrar un sólo carácter para observar el proceso completo que realiza el programa:

El carácter a descifrar será:



El fichero de configuración para descifrar ese carácter será:



Ejecutaremos el siguiente comando:

```
java -jar P2_SI_2015_64_frmerchan.jar -f config_desc.txt
```

Y obtendremos la siguiente salida en pantalla:

```

C:\ Símbolo del sistema
F:\Pruebas>java -jar P2_SI_2015_64_fmmerchan.jar -f config_desc.txt

---- PROGRAMA DE DES-CIFRADO CREADO PARA LA ASIGNATURA 'SEGURIDAD DE LA INFORMACIÓN' - CURSO 2015-2016 ----
-----
- Bienvenid@ al programa de la segunda práctica de Seguridad de la información, correspondiente a la
- asignatura SEGURIDAD DE LA INFORMACIÓN del MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA Y
- ARQUITECTURA, ESPECIALIDAD EN TECNOLOGÍAS INFORMÁTICAS Y DE COMUNICACIONES.
- Utilizamos los algoritmos: Gronsfeld y DES simplificado.
-----
---- (C) 2015 FRANCISCO JAVIER MERCHÁN MACÍAS - INGENIERO TÉCNICO EN TELECOMUNICACIÓN, ESP. TELEMÁTICA ----

- Salida a pantalla activada.
- Salida a fichero activada.
- Vamos a DESCIFRAR el fichero fichero_descodificar.txt
- Clave de descifrado: '1111011010'
- El fichero de salida es: fichero_codificar.txt
- Usamos el cifrado DES Simplificado.
- La traza está activada.

-- INICIO DEL PROCESO DE DESCIFRADO A CONTINUACIÓN -----
##### Comenzamos a generar las claves...
-Entrada P10: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada P8: Bit0[0], Bit1[1], Bit2[1], Bit3[1], Bit4[1], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida P8: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
-Entrada P10: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
Salida P10: Bit0[1], Bit1[0], Bit2[1], Bit3[1], Bit4[1], Bit5[0], Bit6[1], Bit7[1], Bit8[0], Bit9[1]
-Entrada LS1 izq y der: Bit0[0], Bit1[1], Bit2[1], Bit3[1], Bit4[1], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
Salida LS1: Bit0[0], Bit1[1], Bit2[1], Bit3[1], Bit4[1], Bit5[1], Bit6[1], Bit7[0], Bit8[1], Bit9[0]
-Entrada LS2 izq y der: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[0], Bit8[1], Bit9[1]
Salida LS2: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[0], Bit8[1], Bit9[1]
-Entrada P8: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[0], Bit8[1], Bit9[1]
Salida P8: Bit0[0], Bit1[1], Bit2[1], Bit3[0], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
-La clave K1 generada es: 11110101
-La clave K2 generada es: 01100111

##### Iniciamos descifrado del caracter que corresponde al Chiptext [00010111]
-Entrada Plaintext: Bit0[0], Bit1[0], Bit2[0], Bit3[1], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
Salida IP izq y der: Bit0[0], Bit1[1], Bit2[0], Bit3[0], Bit4[1], Bit5[1], Bit6[0], Bit7[1]
-Iniciamos una Vuelta
-Entrada IP derecha o Permutador 1 vuelta derecha: Bit0[1], Bit1[1], Bit2[0], Bit3[1]
Salida EP: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[1]
-Entrada EP para hacer XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[1]
Clave de entrada hacer XOR: Bit0[0], Bit1[1], Bit2[1], Bit3[0], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
Salida XOR: Bit0[1], Bit1[0], Bit2[0], Bit3[0], Bit4[1], Bit5[1], Bit6[0], Bit7[0]
-Inicio cajas S0 y S1
-La posición de la Fila S0 es: 2
-La posición de la Columna S0 es: 0
-La posición de la Fila S1 es: 2
-La posición de la Columna S1 es: 2
-La salida de la caja S0 es [00] y la salida de la caja S1 es [01]
-Entrada SalidaCajas: Bit0[0], Bit1[0], Bit2[0], Bit3[1]
Salida P4: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
-Entrada P4 para hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
IP Izquierda hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[0]

```

```

C:\> Símbolo del sistema

Salida P8: Bit0[0], Bit1[1], Bit2[1], Bit3[0], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
-La clave K1 generada es: 11110101
-La clave K2 generada es: 01100111

##### Iniciamos descifrado del caracter que corresponde al Chiptext [00010111]
-Entrada Plaintext: Bit0[0], Bit1[0], Bit2[0], Bit3[1], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
Salida IP izq y der: Bit0[0], Bit1[1], Bit2[0], Bit3[0], Bit4[1], Bit5[1], Bit6[0], Bit7[1]
-Iniciamos una Vuelta
-Entrada IP derecha o Permutador 1 vuelta derecha: Bit0[1], Bit1[1], Bit2[0], Bit3[1]
Salida EP: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[1]
-Entrada EP para hacer XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[0], Bit4[1], Bit5[0], Bit6[1], Bit7[1]
Clave de entrada hacer XOR: Bit0[0], Bit1[1], Bit2[1], Bit3[0], Bit4[0], Bit5[1], Bit6[1], Bit7[1]
Salida XOR: Bit0[1], Bit1[0], Bit2[0], Bit3[0], Bit4[1], Bit5[1], Bit6[0], Bit7[0]
-Inicio cajas S0 y S1
-La posición de la Fila S0 es: 2
La posición de la Columna S0 es: 0
-La posición de la Fila S1 es: 2
La posición de la Columna S1 es: 2
La salida de la caja S0 es [00] y la salida de la caja S1 es [01]
-Entrada SalidaCajas: Bit0[0], Bit1[0], Bit2[0], Bit3[1]
Salida P4: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
-Entrada P4 para hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
IP Izquierda hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[0]
Salida XOR2: Bit0[0], Bit1[0], Bit2[0], Bit3[0]
-Salida Primer Paso Permutado: Bit0[1], Bit1[1], Bit2[0], Bit3[1], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
-Iniciamos una Vuelta
-Entrada IP derecha o Permutador 1 vuelta derecha: Bit0[0], Bit1[0], Bit2[0], Bit3[0]
Salida EP: Bit0[0], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
-Entrada EP para hacer XOR: Bit0[0], Bit1[0], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
Clave de entrada hacer XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
Salida XOR: Bit0[1], Bit1[1], Bit2[1], Bit3[1], Bit4[0], Bit5[1], Bit6[0], Bit7[1]
-Inicio cajas S0 y S1
-La posición de la Fila S0 es: 3
La posición de la Columna S0 es: 3
-La posición de la Fila S1 es: 1
La posición de la Columna S1 es: 2
La salida de la caja S0 es [10] y la salida de la caja S1 es [01]
-Entrada SalidaCajas: Bit0[1], Bit1[0], Bit2[0], Bit3[1]
Salida P4: Bit0[0], Bit1[1], Bit2[0], Bit3[1]
-Entrada P4 para hacer XOR: Bit0[0], Bit1[1], Bit2[0], Bit3[1]
IP Izquierda hacer XOR: Bit0[1], Bit1[1], Bit2[0], Bit3[1]
Salida XOR2: Bit0[1], Bit1[0], Bit2[0], Bit3[0]
-Salida Segundo Paso Permutado: Bit0[0], Bit1[1], Bit2[0], Bit3[0], Bit4[0], Bit5[0], Bit6[0], Bit7[0]
--> El carácter descifrado en BINARIO es: 01000000
--> El carácter descifrado en ASCII es: @

@

-- FIN DEL PROCESO DE DESCIFRADO -----
>>> Hasta pronto... <<<

F:\Pruebas>

```

Y obtendremos la siguiente salida en el fichero 'fichero\_codificar.txt'.

```

fichero_codificar.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
@

```

### Errores en el fichero de configuración

El programa detecta si en el fichero de configuración no tiene alguna opción importante como la clave de cifrado/descifrado o el fichero de salida, en cuyo caso se define un fichero de salida con el nombre 'FJMM-TrazadoXDefecto.txt'.

En el algoritmo DES simplificado puede detectar si la clave de cifrado no tiene la longitud correcta o no es válida. También detecta si el fichero a descifrar con este algoritmo es válido, es decir, detecta si todos sus caracteres son 0s o 1s.

## 5. Conclusiones

A través de este programa podremos codificar y decodificar cualquier fichero de texto con el cifrado Gronsfeld o DES simplificado, ambos se han explicado en clase de Seguridad de la Información del Máster Universitario de Investigación de Ingeniería y Arquitectura y en los Grados de Informática y Telemática del Centro Universitario de Mérida de la Universidad de Extremadura.

Resulta muy interesante la programación de un algoritmo de cifrado y descifrado en lenguaje Java ya que permite la compresión de dicho algoritmo y nos anima en el estudio de cualquier tipo de algoritmo de cifrado a un nivel muy alto ya que debemos seguir paso a paso todo el algoritmo para realizarlo completamente en el programa. A su vez, debemos controlar muchos parámetros complementarios que nos ayudan a mejorar la programación en general.

Todo esto hace que el estudio de la asignatura Seguridad de la Información nos ayude mucho al desarrollo de nuestros conocimientos en el área de seguridad.

## 6. Bibliografía y referencias utilizadas

- Apuntes de la asignatura Seguridad de la Información de Juan Arias Masa. Centro Universitario de Mérida.
- Alfonso Muñoz Muñoz y Jorge Ramió Aguirre. Cifrado de las comunicaciones digitales. De la cifra clásica al algoritmo RSA. Editorial OxWORD. ISBN: 9788461631247. 2013.
- Gama-Moreno, L. A., & Mejía-Mancilla, J. I. (2009). Implementación de una Capa de Encriptamiento de Campos (CEC) mediante el algoritmo S-DES. (p. 6). Presentado en VII Congreso Internacional en Innovación y Desarrollo Tecnológico, Cuernavaca, Morelos, México. Recuperado a partir de [http://www.lgama.com.mx/images/publicaciones/CEC04sep09\\_final.pdf](http://www.lgama.com.mx/images/publicaciones/CEC04sep09_final.pdf)