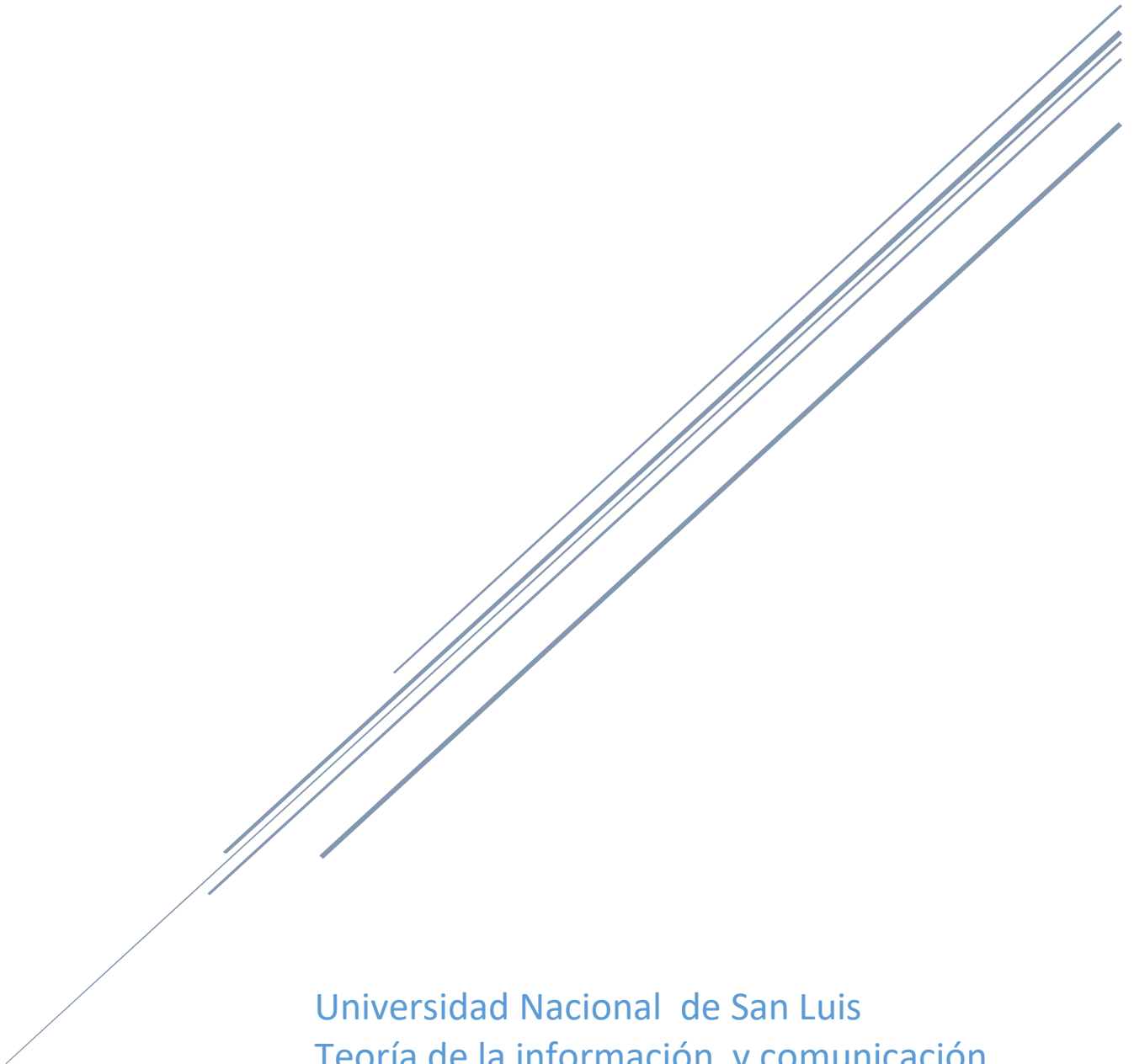


PRACTICO DE MAQUINA

Santiago Bordarampé

Sebastian Riccardo

Gonzalo Barroso



Universidad Nacional de San Luis

Teoría de la información y comunicación

Profesores:

Dr. Germán Montejano

Lic. Mario A. Silvestri

INDICE

1 INTRODUCCIÓN	2
2 CODIFICACIÓN CON HAMMING	3
2.1 PROTEGER	3-5
2.2 INTRODUCIR ERRORES	6
2.3 DESPROTEGER	7
3 COMPACTACIÓN CON HUFFMAN	8
3.1 COMPACTAR	8-9
3.2 DESCOMPACTAR	10
3.3 VER ARCHIVOS	11
3.4 ESTADÍSTICAS	12-14
4 CONCLUSIONES	15-20
4.1 CODIFICACION CON HAMMING	15
4.1.1 CODIFICACIÓN	15-16
4.1.2 DECODIFICACIÓN	17
4.2 COMPACTACION CON HUFFMAN	18-20
4.2.1 COMPRESIÓN	18
4.2.2 GRÁFICO DE TAMAÑO	19
4.2.3 GRAFICO DE TIEMPO	20

Introducción

El siguiente informe detalla el funcionamiento del programa huffman-hamming el mismo fue desarrollado como práctico final de máquina de la materia Teoría de la información y comunicación.

Las funcionalidades principales del programa son la protección de archivos de texto mediante el método de codificación de hamming y la compresión de archivos de texto mediante codificación huffman.

El programa permite codificar con hamming utilizando bloques de tamaño 256, 1024, 2048 o 4096 bits. Por otro lado, el programa compacta archivos de texto usando el radix 2 y generando una tabla de frecuencias específica del propio texto seleccionado.

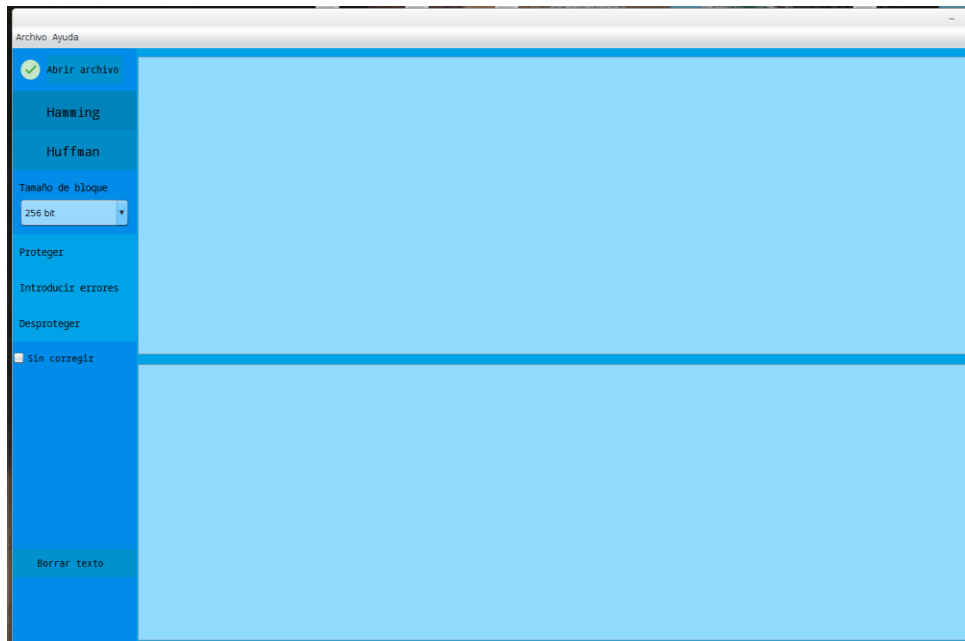
Codificación con hamming

Proteger:

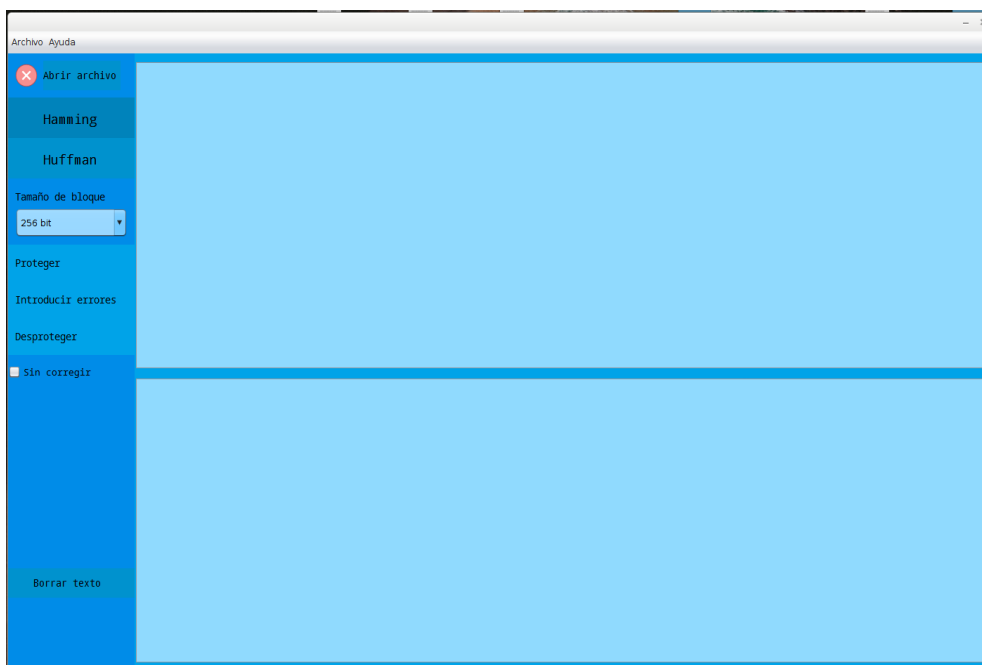
Lo primero que encontramos al abrir el programa es la siguiente pantalla, luego se debe seleccionar un archivo de texto el cual vamos a proteger con el botón “abrir archivo”.



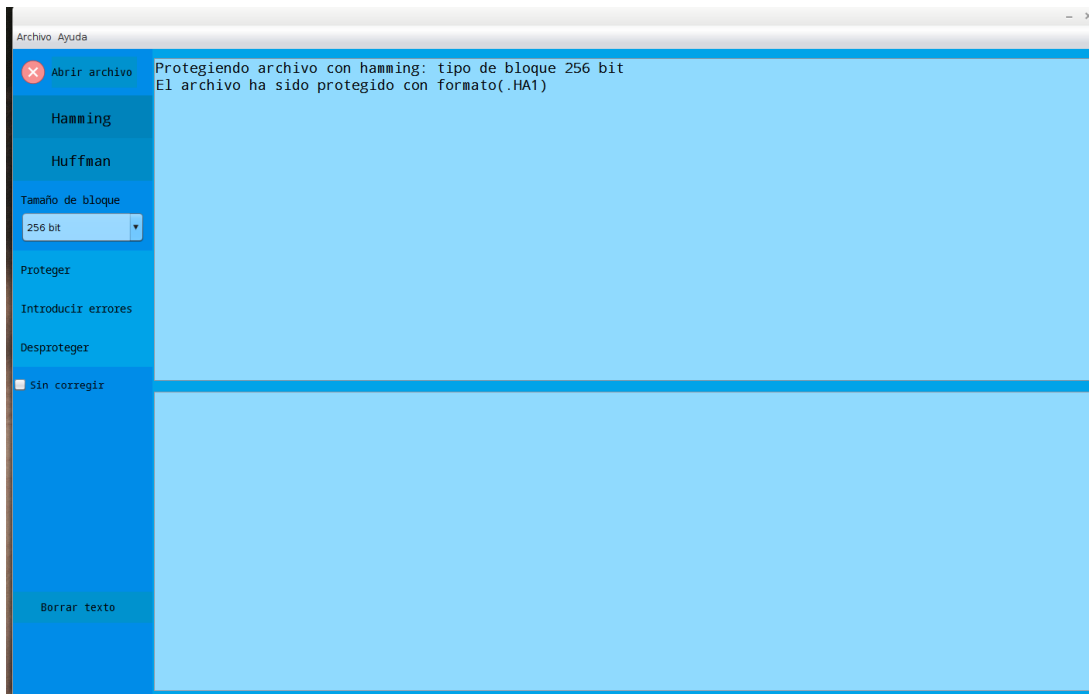
Cuando el archivo haya sido seleccionado aparecerá un check en color verde. Para realizar la protección con hamming se debe presionar el botón “hamming” en la esquina superior izquierda. El botón “hamming” abrirá un menú con las distintas funciones que se pueden realizar con codificación hamming.



En la pantalla que se muestra debajo se deberá seleccionar el tamaño de bloque que se desea utilizar para la protección del archivo con el combo box "Tamaño de bloque" y luego presionar proteger.



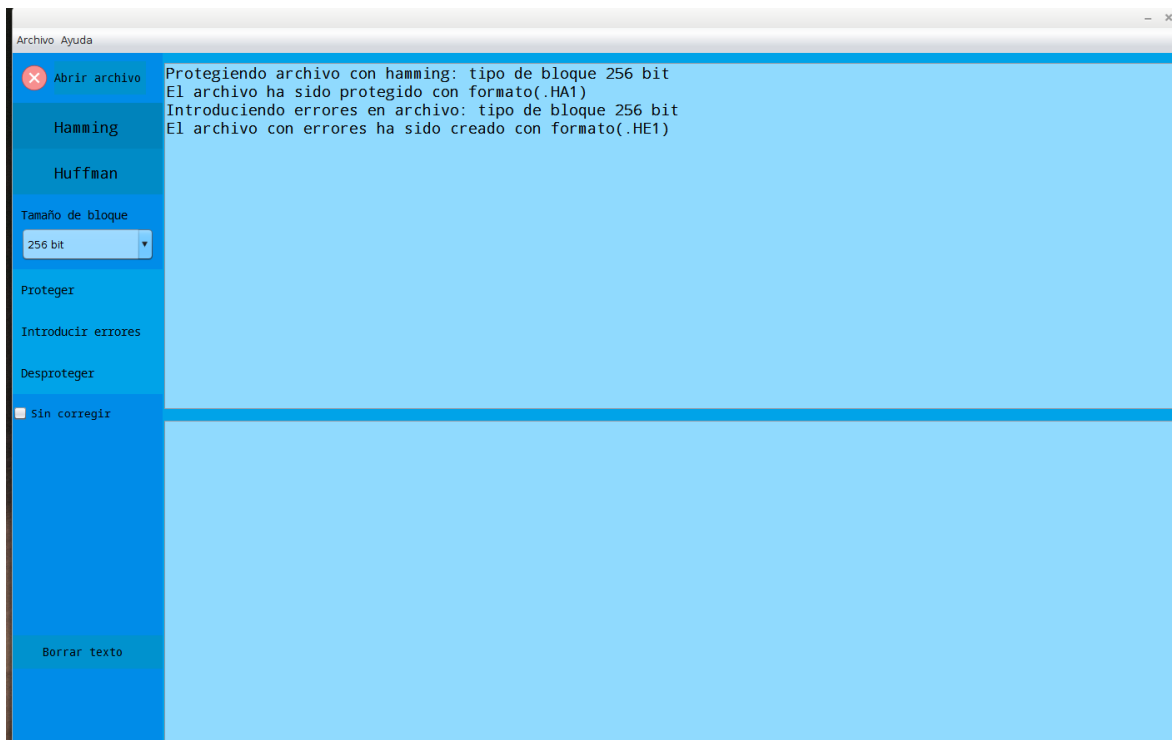
Luego el programa comienza la protección del archivo la cual puede durar desde unos segundos hasta unos minutos dependiendo del tamaño del archivo. Una vez finalizada la protección se crea un archivo en el escritorio con el mismo nombre que el archivo original pero con el formato (.HAX) donde **X=1,2,3,4** dependiendo del tamaño de bloque que se seleccionó para codificar.



Introducir errores:

Para introducir errores debemos seleccionar un archivo con formato HAX buscando el mismo con el botón abrir archivo y luego presionar el botón introducir errores.

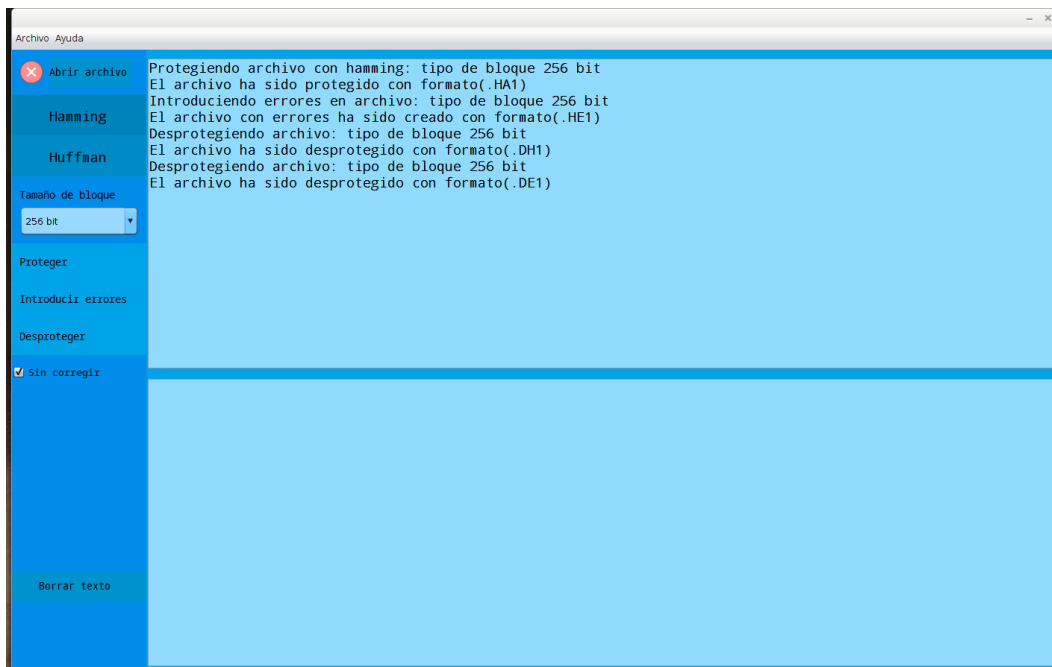
Si el archivo seleccionado es del tipo (.HA1) se deberá seleccionar el tamaño de bloque que le corresponde, en este caso seria 256 bit. Cada bloque tiene un 50% de probabilidad de que se le introduzca un error y en caso de tenerlo, el bit es seleccionado de forma aleatoria. Esta funcionalidad va a generar un archivo con el mismo nombre que el archivo original pero con formato (.HEX) donde X= 1,2,3,4 dependiendo del tamaño de bloque con el cual fue protegido el archivo.



Desproteger:

Para la desprotección de un archivo podemos seleccionar tanto archivos con formato (.HAX) o archivos con formato (.HEX), a la hora de la desprotección podemos seleccionar queremos corregir los errores del archivo o desprotegerlo sin corregir. Por defecto el programa desprotegerá con corrección, en caso de no querer corregir el archivo se deberá seleccionar el check “sin corregir”.

El programa creara un archivo con el mismo nombre que el original pero con formato (.DHX) si el archivo fue desprotegido con corrección y si fue desprotegido sin corrección el formato del archivo será (.DE1) donde **X** = 1,2,3,4.



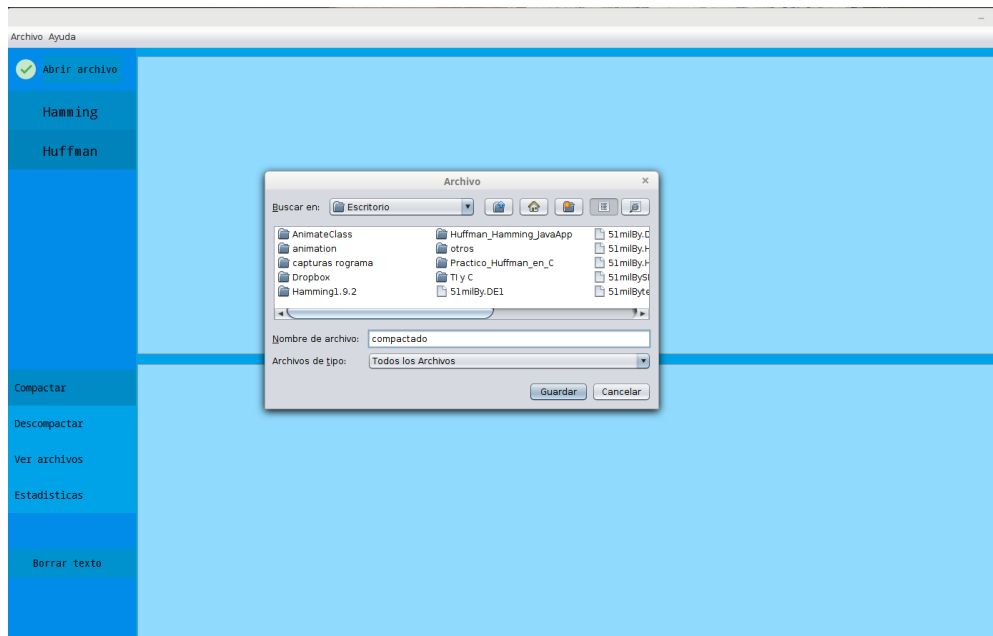
Compactación con Huffman

Compactar:

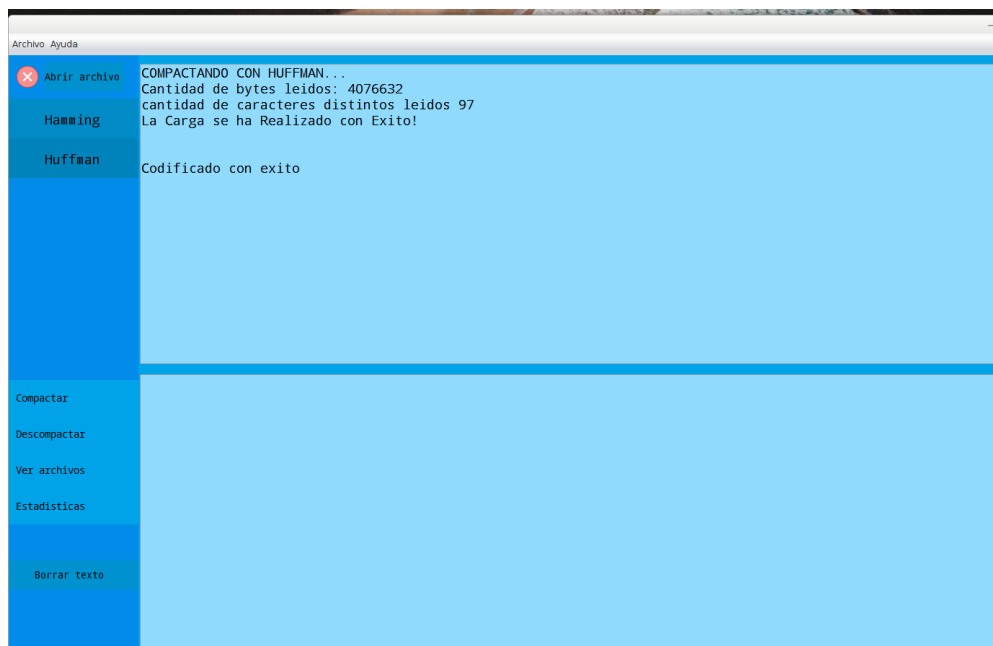
Para compactar un archivo en huffman primero se deberá seleccionar un archivo de texto presionando el botón “Abrir archivo”, una vez seleccionado el archivo deberá aparecer un check en color verde en la esquina superior izquierda. A continuación, tocamos el botón “huffman” el cual abrirá un menú con las distintas funciones que podemos realizar con la compactación huffman.



Cuando el archivo ya fue seleccionado tocamos “Compactar” y se abrirá un File Chooser en donde se debe seleccionar la ruta en donde se quiere guardar el archivo compactado, el programa automáticamente le asignara el formato (.HUF).



Una vez que se presiona guardar el programa comenzará la compactación de archivo la cual podrá durar desde unos segundos a unos minutos dependiendo del tamaño del archivo. Cuando termina de compactar el programa escribe en el área de texto la cantidad de bytes leídos y la cantidad de bytes distintos leídos.



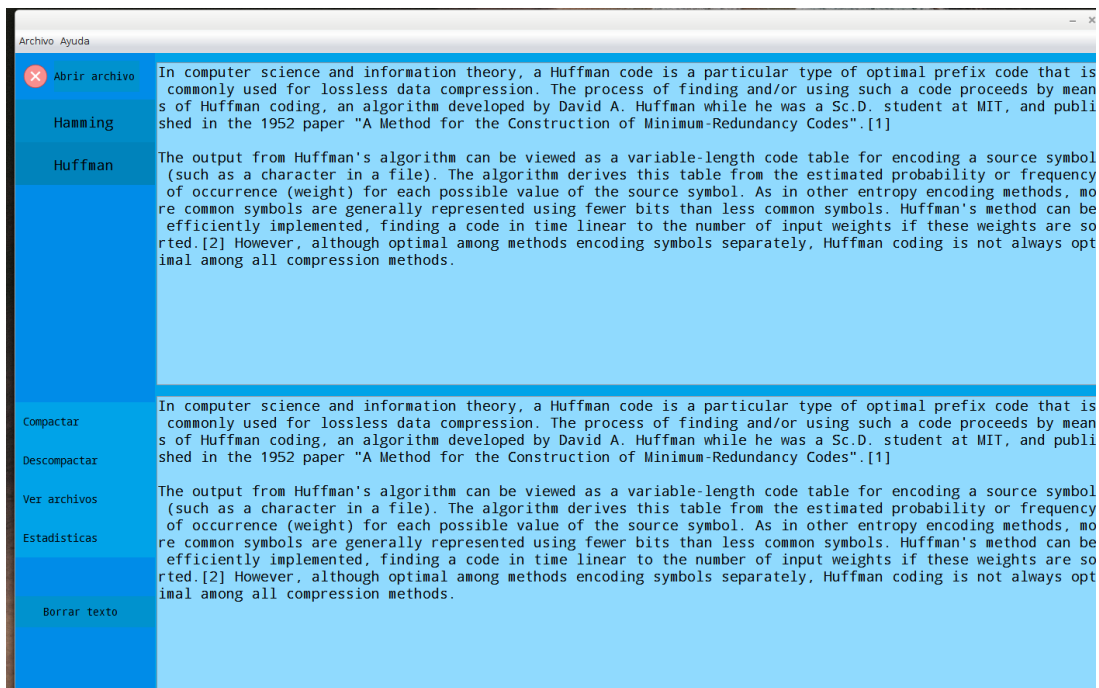
Descompactar:

Primero se deberá seleccionar nuevamente un archivo con el botón “Abrir archivo” pero este deberá tener formato (.HUF). Luego presionamos el botón “Descompactar” lo que abrirá un File Chooser el cual nos permitirá seleccionar donde queremos guardar el archivo descompactado, el programa automáticamente le agregara el formato (.DUF). Se mostrará por medio del área de texto que el programa termino la descompactación del archivo con éxito.



Ver archivos:

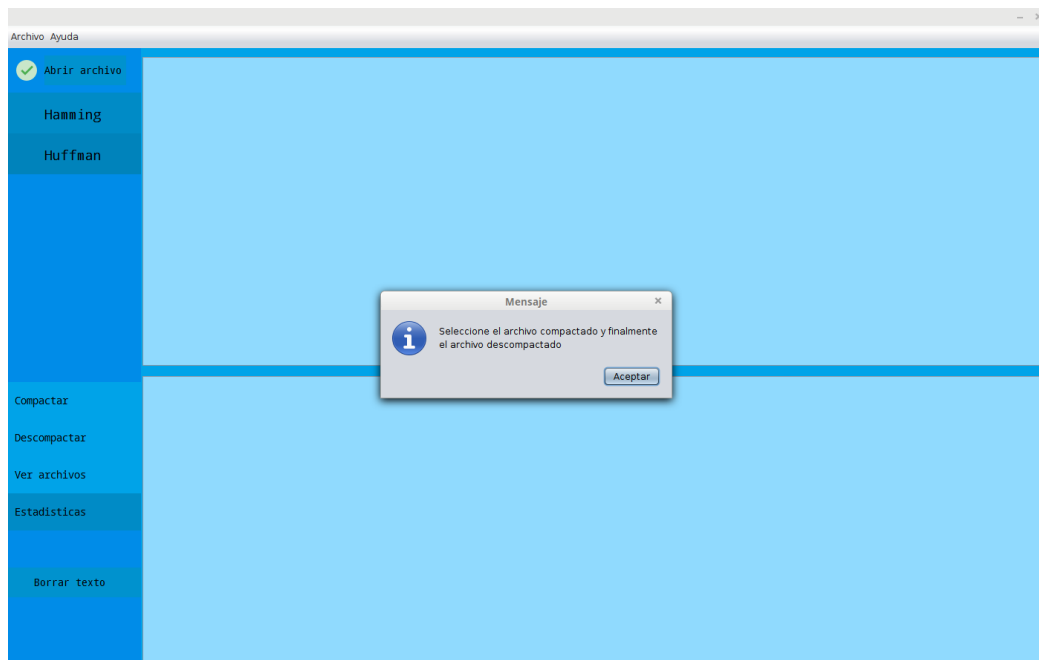
La función “ver archivos” nos permite visualizar dos archivos uno en cada área de texto para así poder compararlos y buscar las diferencias. Una vez que se presiona el botón se abrirá un File Chooser con el cual podremos seleccionar el archivo que se va a visualizar en la primera área de texto(arriba) y luego de seleccionar el primer archivo automáticamente se abrirá otra vez el File Chooser así podemos elegir el segundo archivo que se desea comparar con el primero el cual se abrirá en la segunda área de texto (abajo).



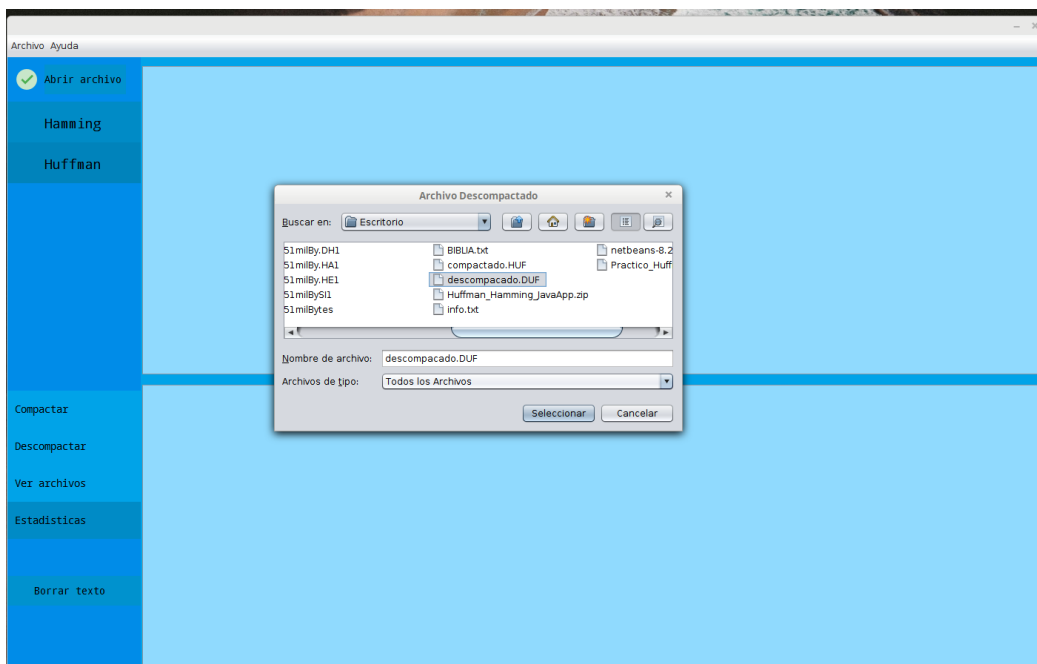
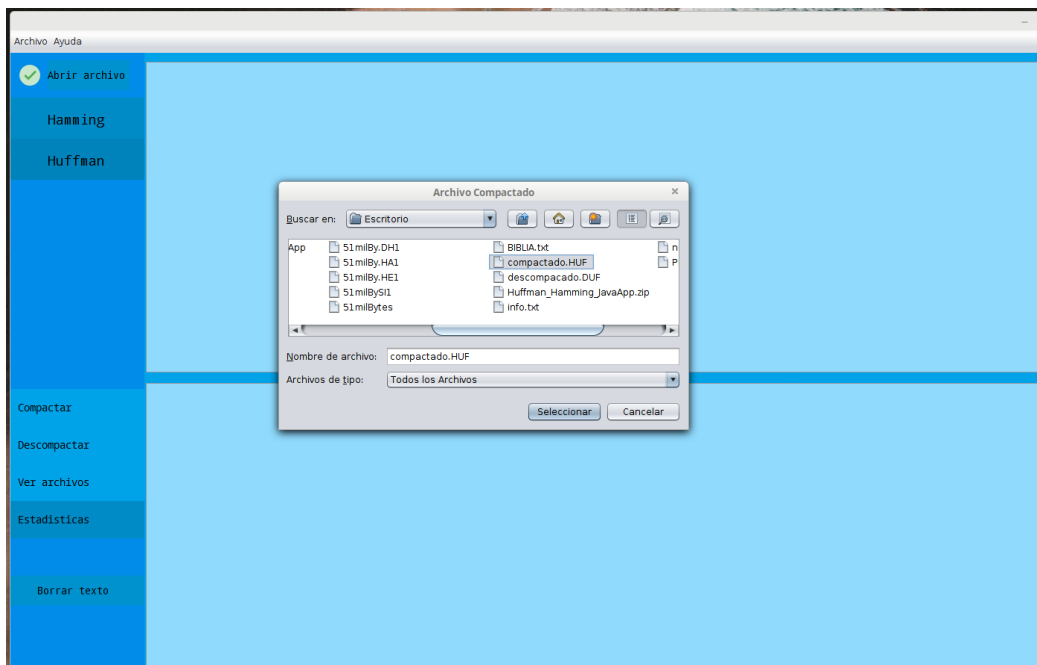
Estadísticas:

La función estadística sirve para mostrar el tamaño de los archivos original, compactado y descompactado.

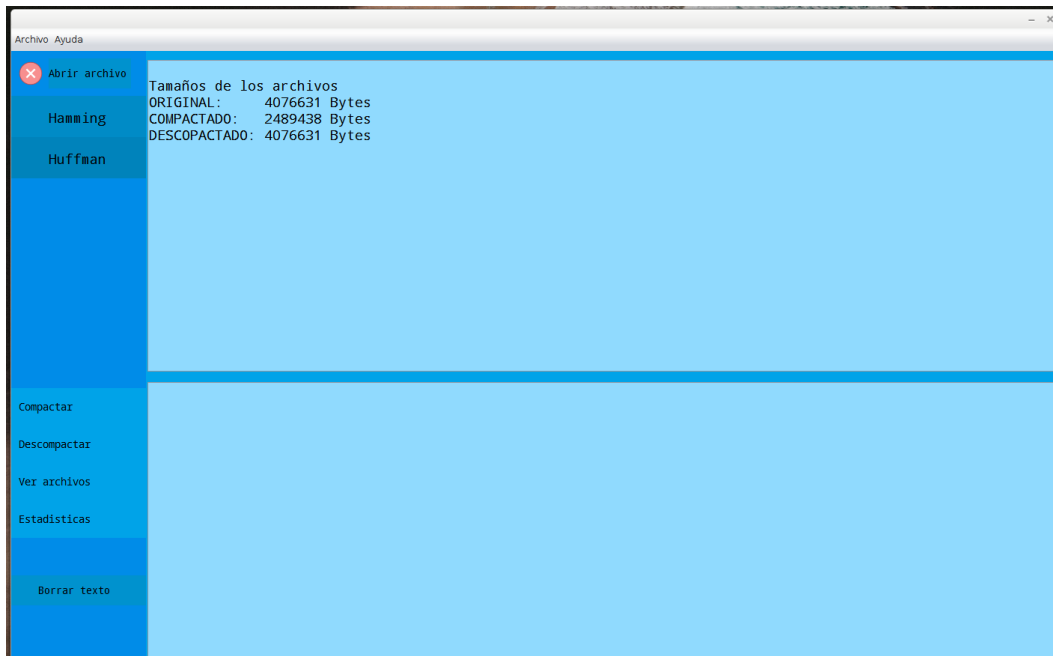
Primero se deberá seleccionar el archivo original con el que se realizó la compactación mediante el botón “Abrir archivo”, una vez seleccionado el archivo presionar el botón “Estadísticas” y el programa abrirá un File Chooser para seleccionar el archivo compactado primero y luego el programa abrirá el File Chooser nuevamente para poder seleccionar el archivo descompactado.



A continuación se muestran los File Choosers con el cuales se deben buscar los archivos compactado y descompactado con formato (.HUF) y (.DUF) respectivamente.



Finalmente, el programa escribirá en el área de texto superior los datos del tamaño de cada archivo, como se muestra en la siguiente imagen.



Conclusiones

Codificación Hamming:

Para recolectar los siguientes datos se utilizó como archivo de prueba un archivo de texto de tamaño de 4.1 Megabytes (4.076.632 bytes), las ejecuciones se realizaron sobre el sistema operativo Linux mint y con un procesador Intel Core i7 HQ 4710. Se realizaron 30 pasadas de cada ejecución y luego se calculó un promedio de tiempo.

Codificación:

Bloque 256

Tiempo de ejecución: 27.4 segundos

Tamaño final del archivo codificado: 4.225.174 bytes

Bloque 1024

Tiempo de ejecución: 36.2 segundos

Tamaño final del archivo codificado: 4.120.901 bytes

Bloque 2048

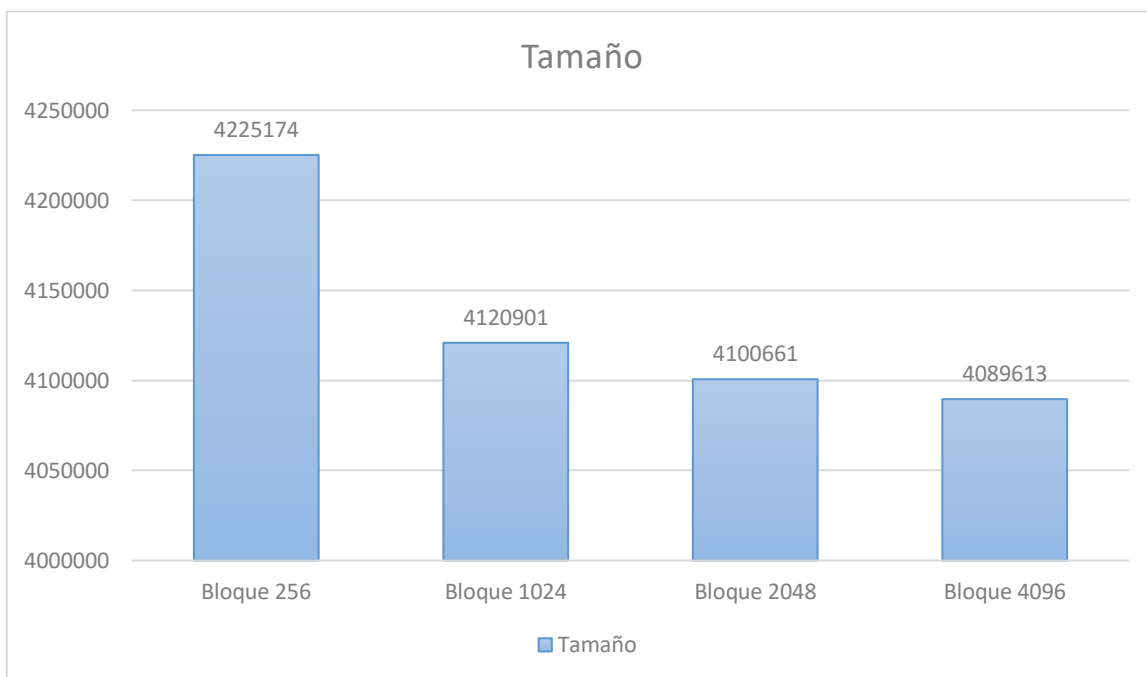
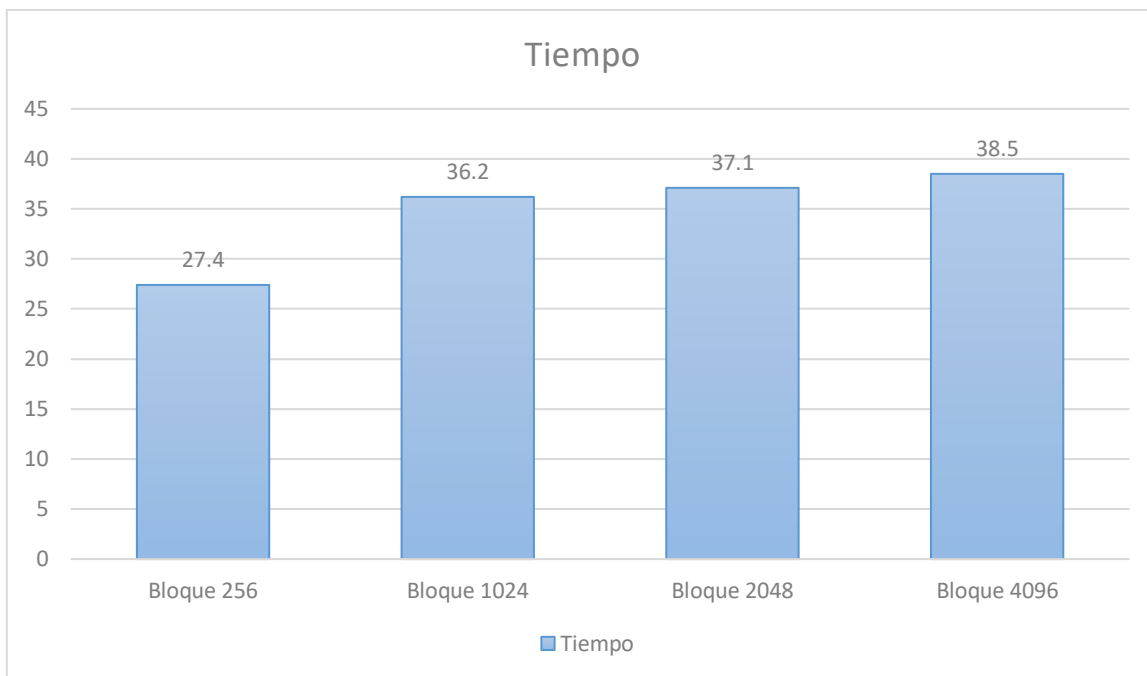
Tiempo de ejecución: 37.1 segundos

Tamaño final del archivo codificado: 4.100.661 bytes

Bloque 4096

Tiempo de ejecución: 38.5 segundos

Tamaño final del archivo codificado: 4.089.613 bytes



Lo que podemos observar después de analizar los siguientes gráficos es que a medida que se aumenta el tamaño del bloque el tiempo de ejecución de la codificación aumenta, mientras que el tamaño del archivo codificado disminuye.

Decodificación:

Bloque 256

Tiempo de ejecución: 12.4 segundos

Bloque 1024

Tiempo de ejecución: 18.5 segundos

Bloque 2048:

Tiempo de ejecución: 20.2 segundos

Bloque 4096:

Tiempo de ejecución: 22.9 segundos



Se observa que mientras el tamaño de bloque aumente el tiempo de ejecución para su decodificación también lo hace.

Compactación con huffman:

Para realizar las ejecuciones se toman tres archivos de textos distintos, el primer archivo tiene un tamaño de 4.1 Megabytes (4.076.632 bytes), el segundo archivo tiene un tamaño de 8.3 Megabytes (8.292.953 bytes) y el ultimo archivo tiene un tamaño de 2.1 Megabytes (2.149.446 bytes).

Compresión:

Archivo 1 (4.076.632 bytes)

Tiempo de ejecución: 7.20 segundos

Tamaño 2.489.438 bytes

Cantidad de caracteres distintos: 97

Archivo 2 (8.292.953 bytes)

Tiempo de ejecución: 14,75 segundos

Tamaño: 5.100.939 bytes

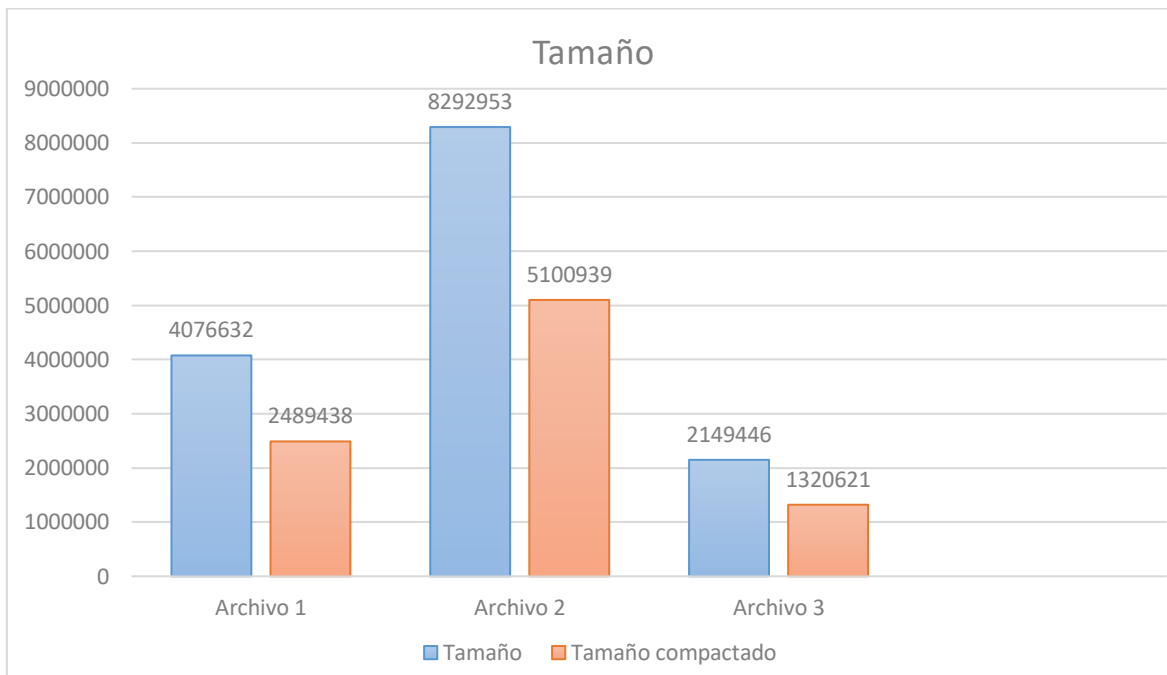
Cantidad de caracteres distintos: 97

Archivo 3 (2.149.446 bytes)

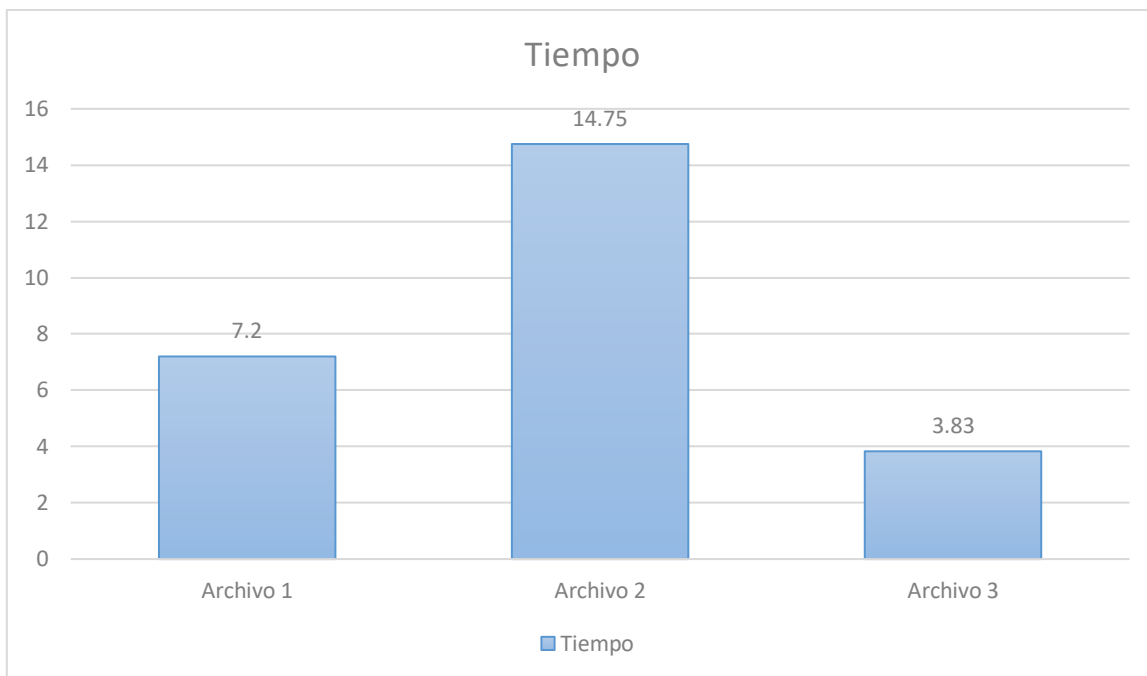
Tiempo de ejecución: 3.83 segundos

Tamaño: 1.320.621 bytes

Cantidad de caracteres distintos: 95



Se puede observar que en el archivo 1 la compresión total fue de un 39% aproximadamente, en el archivo 2 es de 38.5 % y con el archivo 3 la compresión fue de 38.6% por lo que no se puede apreciar una relación entre el tamaño del archivo y el porcentaje de compresión. El porcentaje de compresión está más relacionado con la cantidad de caracteres distintos que existen en el texto.



Como se observa en la gráfica a medida que aumenta el tamaño de los archivos también lo hace el tiempo de compresión. El archivo 2 es 50.9% más pesado que el archivo 1 y su tiempo de compresión es aproximadamente un 51,2% mayor. Por otro lado, comparando el archivo 1 con el archivo 3 vemos que este último es solo el 52.7 % del tamaño del archivo 1 y su tiempo de compresión es 53% más rápido respecto al archivo 1.