



## HuffmanEncoder: Una aplicación informática para codificar mensajes por el método de Huffman<sup>\*</sup>

**Carlos Ernesto Velázquez Rodríguez**

Carrera: Ingeniería Informática

Instituto Superior Minero Metalúrgico (Cuba).

**Resumen:** El artículo introduce la aplicación informática HuffmanEncoder, que implementa el algoritmo de Huffman, un algoritmo para construir códigos de Huffman, creado por David Huffman en 1952 y descrito en "*A Method for the Construction of Minimum-Redundancy Codes*". La aplicación ha sido creada para apoyar el proceso de enseñanza-aprendizaje para la asignatura de Redes de computadoras en el Instituto Superior Minero Metalúrgico. Se abordan elementos del entorno de la telemática, la teoría que subyace bajo el método de Huffman, la situación en la que surgió, y otras interesantes nociones sobre la definición informal y formal del problema, la técnica básica a aplicar, cómo se construye el árbol de Huffman, aspectos sobre una comparación entre códigos ASCII y códigos Huffman, y los principios sobre el software presentado.

**Palabras clave:** Telemática; Telemática; Huffman; Shannon-Fano; redes de computadoras; ASCII; codificación de información.

---

<sup>\*</sup> Trabajo tutorado por el Ing. Eloy Rafael Jiménez Iglesias.  
Recibido: 12 octubre 2013 / Aceptado: 15 junio 2014.

## HuffmanEncoder: a software for coding message

**Abstract:** The article introduces the computer application HuffmanEncoder that implements the algorithm of Huffman, an algorithm to build codes of Huffman, created by David Huffman in 1952 and described in "A Method for the Construction of Minimum-Redundancy Codes". The application has been created to support the teaching-learning process for the subject of Computers Networks in the ISMMM. Elements of the environment of the telematic are approached, the theory that underlies beneath the method of Huffman, the situation in which arose, and other interesting notions about the informal and formal definition of the problem, the basic technique to apply, how the tree of Huffman is built, aspects on a comparison between ASCII codes and Hufmann codes, and the principles on the presented software.

**Key words:** Telematics; Huffman; Shannon-Fano; computers networks; ASCII; code of information.

## Introducción

La Teleinformática surgió como ciencia en los años 70 del siglo pasado y en las últimas décadas sus aplicaciones han logrado una importancia relevante, dada la capacidad de esta para descentralizar mediante redes los recursos ofrecidos por la informática.

Conceptualmente esta disciplina comprende el conjunto de elementos y técnicas que permiten la transmisión automática de datos desde un origen hacia un destino, distanciados geográficamente. Es automática porque no se requiere la intervención humana para llevar a cabo la comunicación. Los datos enviados son entendidos como entidades susceptibles de ser tratados por un ordenador.

La Teleinformática fue recomendada por la *Association of Computing Machinery (ACM)* y el *Institute for Electrical and Electronics Engineers (IEEE)* para su enseñanza en los programas de carreras afines a las ciencias de la informática. En Cuba se lleva a cabo su instrucción en las facultades en las que se preparan futuros profesionales de la informática, y también en otras carreras asociadas.

La Teleinformática se basa en el modelo propuesto inicialmente por Claude Shannon en 1948. Este modelo es un sistema general de comunicación. El mismo parte de una fuente de información desde la que, a través de un transmisor, se emite una señal que viaja por un canal, en este punto puede ser interferida por algún ruido. La señal sale del canal y va a un receptor que la decodifica y la convierte en un mensaje que es entregado a un destinatario. El modelo de la teoría de la información trata de establecer la forma más económica, segura y rápida de codificar un mensaje, sin que la presencia de ruidos complique de alguna manera su transmisión. La codificación puede referirse tanto a la conversión de los datos que se envían por el canal en señales eléctricas o electromagnéticas, como al cifrado para su seguridad.

Una fuente es todo lo que emite mensajes de algún tipo, por ejemplo, un dispositivo de transmisión de datos y los mensajes de datos que envía. Una fuente es, en realidad, un conjunto finito de mensajes.

Un mensaje es un conjunto de ceros y unos. El concepto general de mensaje puede ser aplicado a alfabetos de orden mayor que 2 o binario, pero se restringe en la información digital, como es el caso de las computadoras.

Un código es un conjunto de ceros y unos usados para representar mensajes de acuerdo a cierta regla preestablecida. Esta regla es lo que se llama método o función de codificación.

La información se codifica para ser transmitida. El conjunto total de símbolos que se transmite es llamado alfabeto. La codificación es usada considerablemente durante la transmisión, conservación y procesamiento de la información y es imprescindible para lograr la adaptación óptima de la fuente de información con el canal de comunicación. Este tratamiento que se le da a la información se ocupa, entre otras cosas, de eliminar la redundancia.

La forma en la cual se codifican los mensajes es arbitraria. Existen muchos métodos de codificación. La codificación permite convertir símbolos de un alfabeto de un lenguaje dado en un símbolo de otro sistema de representación, como números o secuencia de pulsos eléctricos de un sistema electrónico.

En la asignatura de Teleinformática (o Redes de Computadoras, como se conoce en el Instituto Superior Minero-Metalúrgico de Moa) se introducen los métodos Shannon-Fano y Huffman como técnicas de codificación de mensajes emitidos por fuentes de información en el modelo de Shannon.

El primero es una técnica para construir un código prefijo que está basado en un conjunto de símbolos y sus probabilidades de ocurrencia estimadas o medidas (como el de Huffman). Este primer método no es eficiente porque no consigue la menor longitud esperada de palabra código posible, como ocurre con el procedimiento de Huffman. El método Shannon-Fano fue propuesto por el primero, pero fue atribuido a Robert Fano, pues fue este quien lo publicó como un informe técnico. No se debe confundir la técnica de Shannon-Fano con la codificación Shannon, que es un método de codificación usado para probar el teorema de Shannon de la codificación sin ruido. Tampoco debe confundirse con la codificación Shannon-Fano-Elias o codificación Elias, que fue el primero en proponer una compresión aritmética.

Como dato curioso, la codificación Shannon-Fano se usa en el método de compresión IMPLUDE, que es parte del formato de los archivos ZIP.

Huffman resolvió la mayor parte de los errores presentes en el método Shannon-Fano. Los precursores enfocaban la solución construyendo un árbol binario desde arriba y

hacia abajo. Huffman le dio vuelta al problema y lo construyó desde abajo y hacia arriba, comenzando por las hojas o nodos terminales. Hizo significativas aportaciones en otras áreas, incluyendo diseño de señales para aplicaciones de radar y comunicación, así como procedimientos de diseño para circuitos lógicos asíncronos (Ibañez, 2002).

Este método de codificación (y compresión) de información aportado por Huffman y su implementación en el software HuffmanEncoder es el tema de este trabajo.

En la asignatura de Redes de Computadoras no se cuenta con herramientas que muestren a los estudiantes de la carrera ejemplos prácticos de los distintos algoritmos que se introducen durante el curso. Es vital para los alumnos tener contacto con estos procedimientos, siguiendo la idea de que la práctica es el criterio de la verdad. En el caso específico de la codificación por el método de Huffman siempre es interesante comparar los resultados de procesar texto por esta técnica y la forma en que se almacenan los caracteres de texto en una computadora, es decir, mediante el *American Standard Code for Interchange of Information* (ASCII). La necesidad de comparar la codificación mediante una técnica y otra consiste en evidenciar el ahorro de espacio para almacenar el texto.

El objetivo general reside en elaborar una aplicación informática que muestre a los educandos un mensaje de texto cualquiera codificado mediante la técnica de Huffman y, además, su representación binaria de su equivalente ASCII. Adicionalmente, mostrar una comparación entre las longitudes de los mensajes codificados obtenidos.

La investigación se ha desarrollado mediante métodos empíricos, fundamentalmente revisión y análisis de varios documentos y sitios webs en internet que cubren varios ángulos del tema de la compresión y codificación por Huffman, la teleinformática y el modelo de comunicación propuesto por Shannon, además de la historia ocurrida en torno a los científicos fundadores y sistematizadores de la telemática como ciencia y los métodos referidos en el presente documento.

### **Basamento teórico subyacente**

Una definición informal del problema sería cómo obtener un código binario prefijo con longitud de palabra código esperada mínima (en términos de árboles, se trata de un árbol con longitud de camino mínimo) dados un conjunto de símbolos y sus pesos

(proporcionales a sus probabilidades de ocurrencia). Un código binario prefijo consiste en obtener cadenas de símbolos para caracteres de tal forma que la cadena de símbolos que representa a un símbolo del alfabeto fuente en particular no será prefijo de otra cadena que representa a otro símbolo de este alfabeto.

Para obtener la solución óptima es necesario usar el método de Huffman. La optimalidad del procedimiento en cuestión fue demostrada por el inventor de la técnica en su artículo "*A Method for the Construction of Minimum-Redundancy Codes*". Tal método consiste en el algoritmo de Huffman, que toma un alfabeto de  $n$  símbolos, además de las frecuencias de aparición de cada símbolo del alfabeto, y produce un código de Huffman para tales datos, otorgando a los símbolos más probables las cadenas de símbolos de longitud más corta que a los menos probables. Estos últimos, obviamente, obtienen longitudes de cadenas de símbolos más largas.

### **Técnica básica**

La técnica es en sí misma el propio algoritmo de Huffman, que radica en construir un árbol binario desde las hojas o nodos terminales (los cuales contienen los caracteres y sus frecuencias y/o probabilidades de ocurrencia relativas) hacia la raíz. Esta construcción se desarrolla en un proceso reiterado de unir, en un nuevo nodo, cada vez, los nodos actuales de menores frecuencias relativas encontrados en la lista de nodos. Este nuevo nodo contendrá la suma de las frecuencias y/o probabilidades de los nodos que lo han formado. El algoritmo finaliza cuando solo queda uno, el cual es la raíz del árbol binario que se espera obtener. Durante todo el transcurso del procesamiento se van etiquetando los hijos izquierdos de cada subárbol con 0, y los derechos, con 1.

Cada carácter, finalmente, estará implícito en las hojas del árbol binario obtenido, y su codificación correspondiente mediante la técnica de Huffman puede ser conseguida formando una cadena con los valores de las etiquetas recorridas desde la raíz del árbol binario hasta el nodo terminal donde se encuentra el carácter en cuestión.

Para ejemplificar de manera tolerable el proceso, se parte a continuación del siguiente adagio latino: "*Post tenebras lux.*", sin comillas.

Los datos obtenidos del mensaje se evidencian en la Tabla 1.

Tabla 1. Datos obtenidos del mensaje

<b>Longitud del Mensaje: 18</b>		
<b>Carácter</b>	Frecuencia	Probabilidad
<b>t</b>	2	0,11111111
<b>&lt;espacio&gt;</b>	2	0,11111111
<b>s</b>	2	0,11111111
<b>e</b>	2	0,11111111
<b>b</b>	1	0,05555556
<b>P</b>	1	0,05555556
<b>o</b>	1	0,05555556
<b>l</b>	1	0,05555556
<b>u</b>	1	0,05555556
<b>r</b>	1	0,05555556
<b>a</b>	1	0,05555556
<b>n</b>	1	0,05555556
<b>x</b>	1	0,05555556
<b>.</b>	1	0,05555556

La primera columna contiene los caracteres presentes en el mensaje. En el caso de la segunda columna, se trata de la frecuencia relativa a cada carácter de la columna precedente. La última contiene las probabilidades de ocurrencia de cada carácter en el mensaje dado. Esta probabilidad se calcula de manera sencilla: es la frecuencia del carácter correspondiente dividida por la longitud del mensaje contenedor.

### **Construyendo el árbol binario**

Se parte desde una lista con nodos, uno por cada carácter presente en la primera columna de la Tabla 1. Estos nodos contienen las informaciones relativas a cada carácter asociado en la tabla. La lista debe estar ordenada por criterio de probabilidad o de frecuencia en forma decreciente.

El algoritmo a seguir es el siguiente:

1. Mientras la longitud de la lista de nodos sea mayor que 1:

(1). Crear un nuevo nodo con una frecuencia (probabilidad de ocurrencia) igual a la suma de las frecuencias (probabilidades) de los dos nodos actuales de menor frecuencia (probabilidad de ocurrencia) en la lista.

(2). Asignarle al nuevo nodo, los nodos escogidos anteriormente. Como hijo derecho el de menor frecuencia (probabilidad) de los dos, y como hijo izquierdo, el restante. Eliminar de la lista los nodos sumados.

(3). Insertar el nuevo nodo en la lista siguiendo el criterio de orden de esta (frecuencia o probabilidad decreciente).

2. El nodo que queda será la raíz del árbol de Huffman.

Para el texto dado, el procedimiento sería el que se muestra en la Tabla 2.

Tabla 2. Procedimiento para el algoritmo de Huffman (1)

<b>Longitud del Mensaje: 18</b>							
<b>Carácter</b>	Frecuencia	Probabilidad					
<b>t</b>	2	0,11111111	2	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>
<b>&lt;espacio&gt;</b>	2	0,11111111	2	2	2	2	2
<b>s</b>	2	0,11111111	2	2	2	2	2
<b>e</b>	2	0,11111111	2	2	2	2	2
<b>b</b>	1	0,05555556	1	2	2	2	2
<b>P</b>	1	0,05555556	1	1	2	2	2
<b>o</b>	1	0,05555556	1	1	1	2	2
<b>l</b>	1	0,05555556	1	1	1	1	2
<b>u</b>	1	0,05555556	1	1	1	1	1
<b>r</b>	1	0,05555556	1	1	1	1	1
<b>a</b>	1	0,05555556	1	1	1	1	
<b>n</b>	1	0,05555556	1	1	1		
<b>x</b>	1	0,05555556	1	1			
<b>.</b>	1	0,05555556	1				
<b>Suma total de probabilidades</b>		<b>1</b>					
			<b>Suma parcial de frecuencias</b>		<b>2</b>	<b>2</b>	<b>2</b>

Tabla 2. Procedimiento para el algoritmo de Huffman (2)

<b>2</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>6</b>	<b>8</b>	<b>10</b>	<b>18</b>
2	2	4	4	4	4	6	8	
2	2	2	4	4	4	4		
2	2	2	2	4	4			
2	2	2	2					
2	2	2						
2	2							
2								
<b>2</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>6</b>	<b>8</b>	<b>10</b>	<b>18</b>

Como observación interesante se puede resaltar que la suma de las probabilidades de ocurrencia de los caracteres será siempre 1, y la suma de las frecuencias, indubitablemente, será igual a la longitud del texto. Para hacer las iteraciones se pueden sumar las frecuencias o las mismas probabilidades de ocurrencia. Esto es irrelevante pues estos valores son proporcionales, dado que la probabilidad de cada carácter depende de su frecuencia en el texto, que tiene una longitud fija, es decir:



$$P(i) = \frac{F(i)}{L(T)}$$

donde  $P(i)$  indica la probabilidad de ocurrencia del carácter  $i$  en el texto  $T$ ,  $F(i)$  indica la frecuencia de ocurrencia del carácter  $i$  en el texto  $T$  (un conteo de las veces que aparece el carácter en el texto) y  $L(T)$  es la longitud del texto  $T$ .

Las probabilidades pueden ser siempre las mismas para cada carácter, de ser así estarán basadas en un caso promedio: normalmente se ha tomado alguna obra literaria y se ha procesado para obtener los datos necesarios (frecuencia y probabilidad). Una opción consiste en procesar dinámicamente los mensajes que se pasan para obtener las frecuencias propias de cada carácter en el mensaje actual (Rueda *et al.*, 1995).

Si se siguiese la primera forma podría pensarse en usar *El Ingenioso Hidalgo Don Quijote de La Mancha*, pieza cumbre de la literatura española escrita por Miguel de Cervantes, pero no puede asegurarse que el lenguaje usado en la obra sea estándar actualmente. Así las cosas, se ha propuesto que se use *La Regenta*, de Leopoldo Alas, dado que esta novela es un poco más actual y está escrita con un lenguaje un poco más vigente que la primera opción, aunque fue escrita en el siglo XIX. Esta obra está considerada la mejor novela española de esa centuria.

En la situación de recurrir a una tabla precalculada, el descompresor/decodificador debe contar también con la misma. Si se calcularan dinámicamente las frecuencias, la tabla ha de almacenarse con el texto codificado para, a la hora de decodificar, poder completar el proceso exitosamente.

Durante todo el procedimiento se hubo de otorgar un valor de 0 o 1 para cada hijo, izquierdo y derecho, respectivamente, de cada nuevo nodo formado. Al final del proceso, como ha sido explicado antes, se obtendrá un árbol binario, cuyas hojas contienen los caracteres. Para formar el código propio de cada carácter no se hace más que recorrer desde la raíz del árbol binario hasta la hoja que contenga al carácter buscado, e ir tomando los caracteres de codificación encontrados.

En la ilustración 4 se puede observar un posible árbol de Huffman (que es en sí mismo un árbol binario) generado para este mensaje. Si se desea encontrar el código correspondiente al carácter espacio, representado en el árbol por el carácter

underscore ( \_ ), hay que situarse en la raíz del árbol y hacer un recorrido hacia la hoja que contiene tal carácter. Durante el desplazamiento se va formando el código con las etiquetas de los arcos superados en el recorrido. Finalmente el código obtenido será el correspondiente al carácter contenido en el nodo terminal al que se llega.

Así se comprueba que el código Huffman asignado al carácter espacio es 0000. En la Tabla 4 se ofrece la lista de caracteres y sus codificaciones correspondientes según el árbol obtenido por este método sumando las frecuencias.

Es necesario resaltar que esta codificación obtenida no es la única posible. Esto depende del orden de inserción de los nuevos nodos obtenidos en la lista de nodos. De esta forma es posible obtener distintos árboles de Huffman. Sin embargo, cualquier árbol de Huffman genera un código óptimo; es decir, cualquier árbol de Huffman codificará un texto que tenga las frecuencias de la misma tabla, exactamente en el mismo espacio (óptimo).

### **ASCII y códigos Huffman**

El método más habitual para representar caracteres internamente en una computadora es usar cadenas de bits de longitud fija. Este es el caso ASCII.

ASCII es un código de caracteres basado en el alfabeto latino, de uso amplio en el idioma inglés y otras lenguas de occidente. Surgió en el año 1963, en el Comité Estadounidense de Estándares. Fue una evolución de los conjuntos de códigos utilizados por aquella época en la telegrafía. Su primera publicación como estándar fue en 1967, y su última actualización fue en 1986.

ASCII simboliza cada carácter mediante una cadena de siete bits. El problema que presenta ASCII es que no puede codificar más de  $2^7 = 128$  caracteres, de ellos, solo imprimibles los últimos 95, dado que los primeros 32 eran caracteres no imprimibles, la mayor parte, caracteres de control en desuso que influyen en cómo se procesa el que usaban para marcar paquetes de datos, o para controlar protocolos de transmisión de datos.

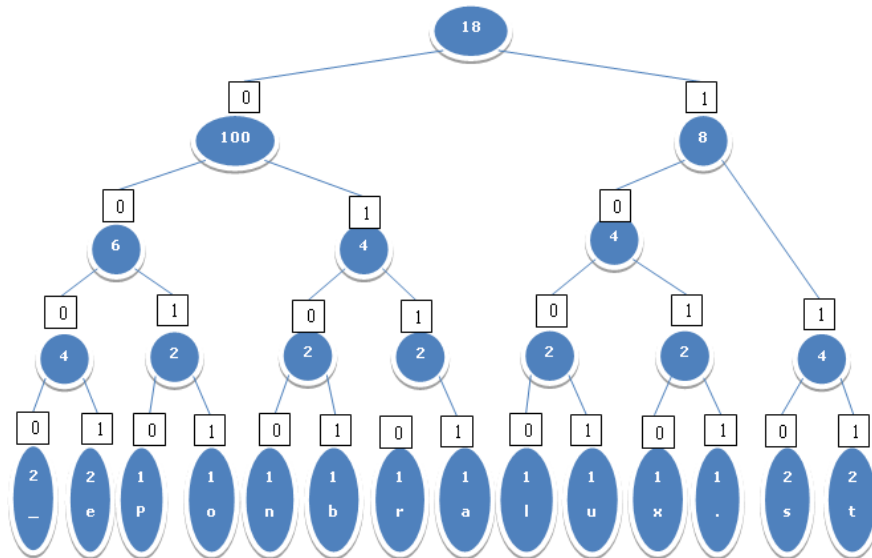


Figura 1. Árbol de Huffman para el ejemplo de la Tabla 1.

ASCII tiene muchas variantes que han surgido para facilitar la escritura en lenguas que usan el alfabeto latino distintas del inglés. Esto ha sido una necesidad a medida que la tecnología de la informática se difundió a lo largo del mundo. Estas variedades se agrupan usualmente bajo el término "ASCII extendido", que a veces se aplica erróneamente a otras que incluso no mantienen el conjunto de caracteres de 7 bits de ASCII.

Tabla 3. Tabla de códigos Huffman obtenidos usando el árbol de la ilustración 1

Relación de códigos Huffman obtenidos	
Carácter	Código Huffman
<espacio>	0000
e	0001
s	110
t	111
n	0100
b	0101
r	0110
a	0111
l	1000
u	1001
x	1010
.	1011
P	0010
o	0011

Los códigos Huffman representan los caracteres mediante cadenas de bits de longitud variable, en contraposición a ASCII, que usa cadenas de bits de longitud fija. Es por esto que los códigos Huffman representan una alternativa para ASCII y muchos otros códigos de longitud fija, sin querer esto decir que sea factible usar Huffman para sustituir ASCII en la representación de caracteres en las computadoras. Con Huffman, por lo general, es posible representar cadenas de caracteres, como textos o programas, en un espacio mucho menor, comparado con el que se necesita al usar ASCII.

La longitud del texto original del ejemplo ("*Post tenebras lux.*", sin comillas) es de 18. La codificación por ASCII del mensaje tendría una longitud de 111, y la codificación por Huffman tendría apenas una longitud igual a 60. En términos de bytes, la representación ASCII del mensaje sería de aproximadamente unos 15 bytes. Este mismo mensaje ocuparía en códigos Huffman cerca de apenas unos 8 bytes, para una razón de compresión ASCII/Huffman de 1.85.

### **Aplicación HuffmanEncoder**

#### **Un acercamiento a HuffmanEncoder**

La aplicación HuffmanEncoder es un software de escritorio desarrollado en el lenguaje Java, usando el IDE NetBeans 7.0, para el apoyo a la enseñanza de la asignatura de Redes de Computadoras en el Instituto Superior Minero-Metalúrgico de Moa.

Permite insertar un texto directamente por teclado o a través de un archivo, que puede ser de cualquier formato (será interpretado como texto plano), con la única prescripción de que debe tener una longitud menor de 18 000 caracteres (incluidos saltos de línea, tabulaciones y retornos de carro o *enter*).

La herramienta no usa una tabla precalculada de frecuencias basada en un caso promedio. En su lugar se calculan las frecuencias de cada carácter en cada nuevo mensaje insertado.

La aplicación muestra por separado datos relevantes sobre el texto ofrecido para codificar, tales como su longitud, cantidad de caracteres únicos, frecuencia y probabilidad de ocurrencia de cada uno de ellos en el mensaje, y su codificación por el método de Huffman y mediante ASCII. A la par brinda la redacción entregada traducida completamente en ASCII y Huffman, además de otras informaciones útiles

sobre la compresión alcanzada con Huffman en perspectiva con ASCII. El sistema propone, del mismo modo, salvar las codificaciones obtenidas por una u otra vía en archivos.

La codificación Huffman se encuentra todavía usada en la generalidad, sobre todo gracias a su simplicidad, alta velocidad e inexistencia de dificultades relacionados con las patentes, debido a que el profesor Huffman, su creador, no insistió nunca en inscribir comercialmente su descubrimiento.

### Aspectos del análisis y diseño de HuffmanEncoder

Se ha de recibir un mensaje de texto que puede contener caracteres alfanuméricos y signos de puntuación, llaves, corchetes, comillas, paréntesis, signos de admiración e interrogación, de operaciones aritméticas y otros.

A partir de este mensaje, especificado por el usuario a través de un archivo o por teclado, se obtiene una lista con los datos necesarios para la entrada del algoritmo de Huffman que producirá un árbol binario desde el cual se obtendrán los códigos relativos a cada carácter contenido en el mensaje.

Para tomar el texto se requiere una interfaz visual en función de la entrada del mensaje, por teclado o desde un archivo que habrá que cargar en el software. Se decidió diseñar una para cada una, las que se muestran en las Figuras 2 y 3.

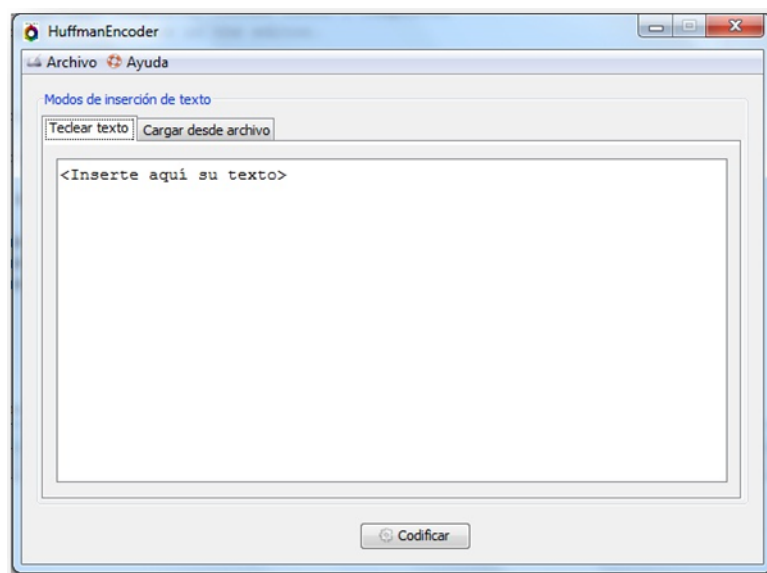


Figura 2. GUI para inserción de texto por teclado.

El tratamiento del texto de entrada se lleva a cabo luego de pulsar el botón "Codificar", en la GUI de inserción de texto correspondiente. En este punto, luego de realizar las validaciones pertinentes, el control del flujo del programa es cedido a la clase Huffman, que se encarga de construir una tabla de frecuencias en consecuencia a cada carácter presente en el mensaje.

A continuación se procede a construir el árbol de Huffman. Durante la conformación de la estructura se van asignando los códigos propios de cada rama.

Con el propósito de apoyar el progreso de la conversión de caracteres a códigos Huffman, la clase estática Huffman hace uso de otras clases auxiliares (ArbolBinarioHuffman y NodoHuffman). ArbolBinarioHuffman es un tipo de estructura de dato basado en los árboles binarios, como su nombre indica. Una estructura de esta especie no es más que un árbol con raíz cuyos nodos o vértices tienen, a lo sumo, 2 hijos, llamados cada uno hijo izquierdo e hijo derecho. Un árbol es un grafo conexo y acíclico.

Cuando los vértices tienen datos asociados y están ordenados de manera que para cada nodo  $n$  en el árbol, cada dato contenido en el hijo izquierdo de  $n$  es menor que el contenido en  $n$ , y cada dato en el hijo derecho de  $n$  es mayor que el dato en  $n$ , la estructura se llama árbol de búsqueda binaria. Este tipo de árbol es especialmente útil para localizar datos rápidamente.

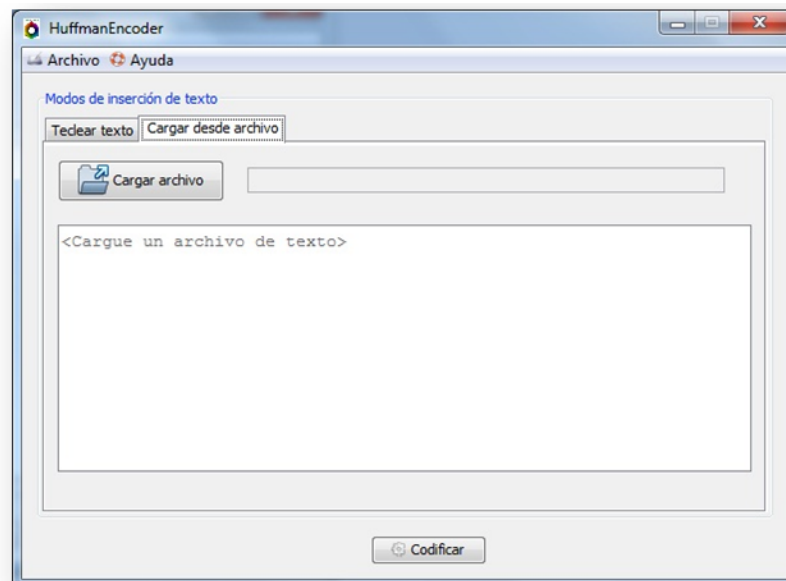


Figura 3. GUI para inserción de texto desde un archivo.

La clase `ArbolBinarioHuffman` se soporta en la clase `NodoHuffman`, como indica la imagen 4.



Figura 4. Diseño de clase `ArbolBinarioHuffman`.

Un objeto de tipo `NodoHuffman` contiene los siguientes campos:

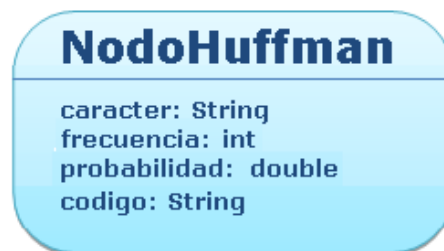


Figura 5. Diseño de clase `NodoHuffman`.

De modo similar que para la inserción, es necesaria una interfaz gráfica para mostrar el resultado de la computación realizada sobre el texto tributado a la aplicación. El prototipo se muestra en la imagen 6 con los resultados obtenidos para el texto del ejemplo (*"Post tenebras lux."*, sin comillas).

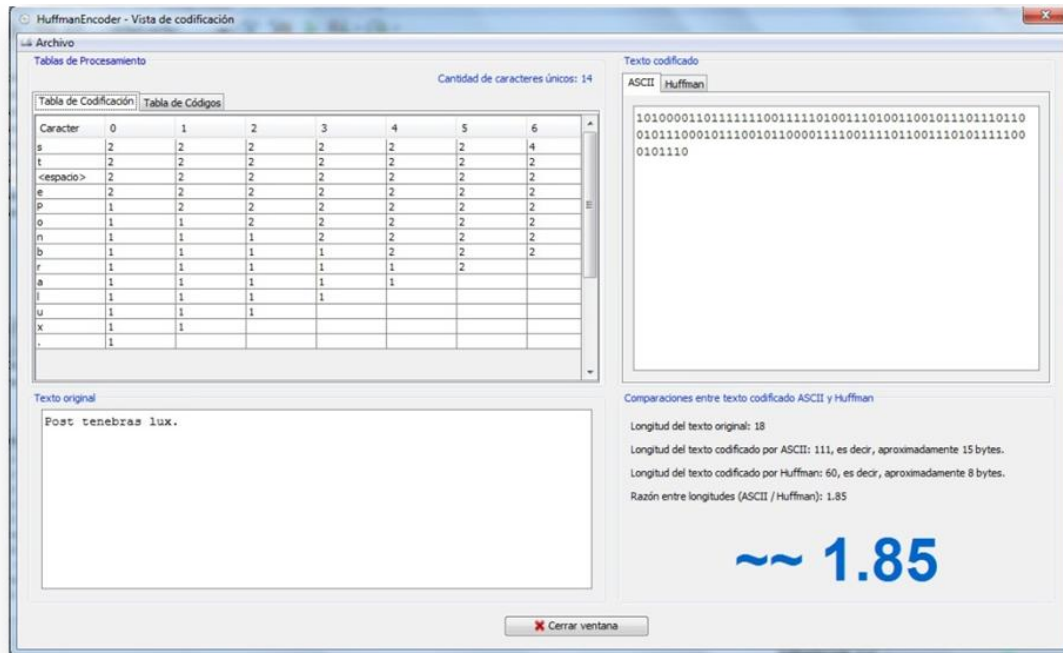


Figura 6. GUI para mostrar resultados obtenidos (1).

Como puede observarse, esta GUI ofrece varios elementos de análisis del texto codificado. El panel Tablas de Procesamiento contiene las tablas del proceso de codificación y la de códigos. En esta última se muestran los caracteres presentes en el texto, sus frecuencias, sus probabilidades de ocurrencia, su codificación por Huffman y por ASCII, como se observa en la imagen 7.

También puede apreciarse en ambas iconografías las codificaciones del texto en ASCII y Huffman, respectivamente. Estos datos se encuentran en el panel de Texto codificado.



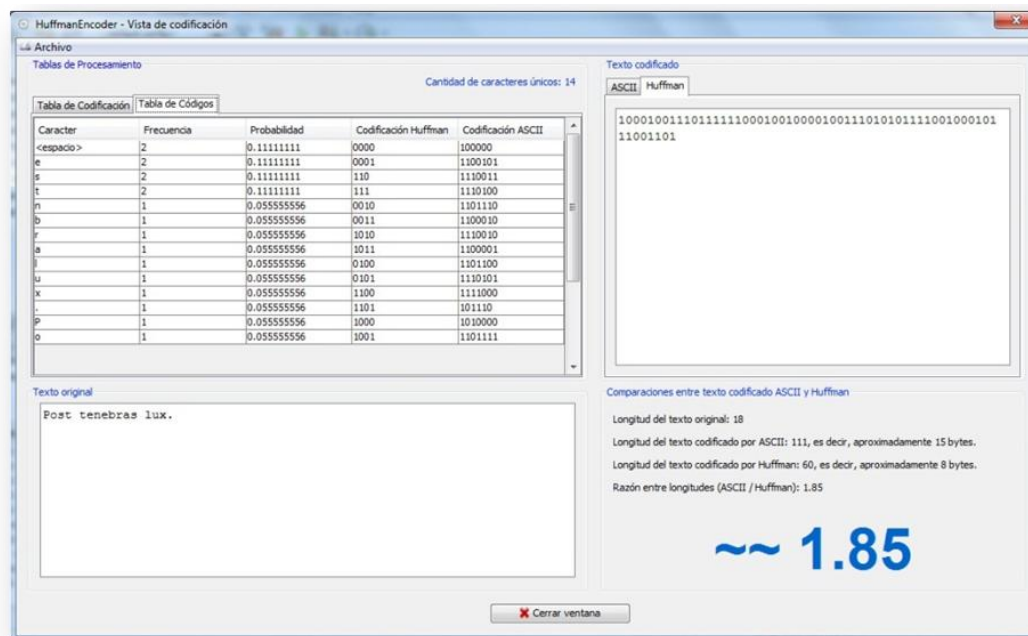


Figura 7. GUI para mostrar resultados obtenidos (2).

La GUI ofrece igualmente una sección en la que se encuentra el texto original, y por último, otro compartimiento en el que se brinda información relativa a la eficiencia de la codificación por Huffman. Las reseñas propuestas incluyen longitud del texto original, longitud de texto codificado por ASCII, longitud de texto codificado por Huffman y la razón de compresión entre el texto codificado por ASCII y el codificado por Huffman.

## Conclusiones

La aplicación de escritorio HuffmanEncoder fue diseñada para su empleo en la asignatura de Redes de Computadoras, como apoyo a su enseñanza en el Instituto Superior Minero Metalúrgico, donde se instruye la carrera de Ingeniería informática. Ha sido realizada en Java a través del entorno de desarrollo NetBeans 7.0.

HuffmanEncoder habilita a sus usuarios, los estudiantes, para tener una idea mejor formada sobre el proceso de codificación por Huffman y cuánto de significativo tiene la optimización que surge de simbolizar óptimamente los caracteres de un mensaje dado para transmitirlo por un medio cualquiera, en términos de ahorro de tiempo y espacio.

El desarrollo de HuffmanEncoder ha redundado en un conocimiento un poco más efectivo de las bases de la teleinformática, una mejor concepción del ámbito en que surgió tal ciencia, los principales personajes científicos que lograron instituir la

otorgándole su calidad de disciplina bien establecida y ampliamente usada en la actualidad.

Comúnmente esta codificación se usa en otros métodos de compresión. La deflación y códec multimedia como JPEG y MP3 tienen una cuantificación digital basada en la codificación Huffman.

### Referencias bibliográficas

HUFFMAN, D. 1952: A method for the construction of minimum-redundancy codes. *Proceedings of the I.R.E.*, sept, p. 1 098-1 102.

IBAÑEZ, D. & SILVERA, M. 2002: Procesadores digitales de señales. Comprensión de datos utilizando el algoritmo de Huffman. Proyecto final. [En línea]. Consultado: 6 may 2012. Disponible en: <http://iie.fing.edu.uy/ense/asign/dsp/proyectos/2002/compresion/comhuff.htm>.

RUEDA, L. *ET AL.* 1995: Un modelo de codificación dinámica del método de Huffman para la compresión de datos en línea. En: I Congreso Argentino de Ciencias de la Computación Red de Universidades con Carreras en Informática (RedUNCI). p. 165-174.

SHANNON, C. E. 1948: A Mathematical Theory of Communication. *Bell System Technical Journal* 27(3): 379-423, July.