

Documentación Segunda Tarea Programada

Danny Chaves Chaves, dxnnx@me.com, Minor Sancho Valverde, tivin.minor10@gmail.com, Oscar Chavarria Campos, ocach14@hotmail.com

September 29, 2016

1 Introducción

Los sistemas de almacenamiento y comunicación digitales con múltiples propósitos, necesitan optimizar el uso de recursos, debido a esto se han formulado múltiples algoritmos para comprimir y descomprimir la información contenida en una señal. La compresión de una señal digital (compuesta por 1's y 0's) consiste en aplicar algún método que permita disminuir el tamaño original de la señal. Existen varios algoritmos, los cuales permiten reconstruir dicha señal pero algunos no logran el 100 de la reconstrucción. Un algoritmo utilizado para lograr la compresión en su totalidad es conocido como algoritmo de Huffman.

2 Análisis del Problema

Primeramente para este proyecto debimos investigar sobre el algoritmo de Huffman, como este funciona, ya que se debe de crear un árbol binario, pero empezando desde abajo para arriba su construcción, luego verificar la cantidad de apariciones de los nodos hijos, en cada nodo. Pero para lograr implementar el algoritmo, primero se tenia que investigar sobre el lenguaje, ya que prolog es un lenguaje lógico, el cual es muy poco utilizado se podría decir que solo se usa en casos específicos, como motivos académicos.

3 Diseño de la solución

Para implementar el algoritmo de Huffman se siguieron los pasos mencionados en la especificación del proyecto dados por el profesor.

3.1 Diccionario de Huffman

Se analiza todo el texto, para construir el diccionario de frecuencias de aparición, donde un token constituye la llave, y el valor de la entrada es definido por el número de veces que el token sucede en el texto.

3.2 Árbol de Huffman

Los pasos consisten en.

- Crear un nodo por cada token.
- Tomar y remover los dos nodos de menor frecuencia de la lista y unirlos en un nuevo nodo. Este nuevo nodo tendrá como etiqueta un token nulo, y como cantidad de apariciones, la suma de la cantidad de apariciones de los dos nodos hijos. El nuevo nodo se inserta en la lista de nodos.
- Repetir el paso 2 hasta que exista un solo nodo dentro de la lista.

3.3 Compresión

El texto comprimido se obtiene entonces al reemplazar cada token por el código de Huffman especificado en el diccionario de Huffman.

4 Modo de Uso

Los pasos para ejecutar el programa son:

1. ?- crearDiccionario('RUTA').
2. ?- comprimirTexto('RUTA').
3. ?- descomprimir.

5 Pruebas

5.1 Ejecución

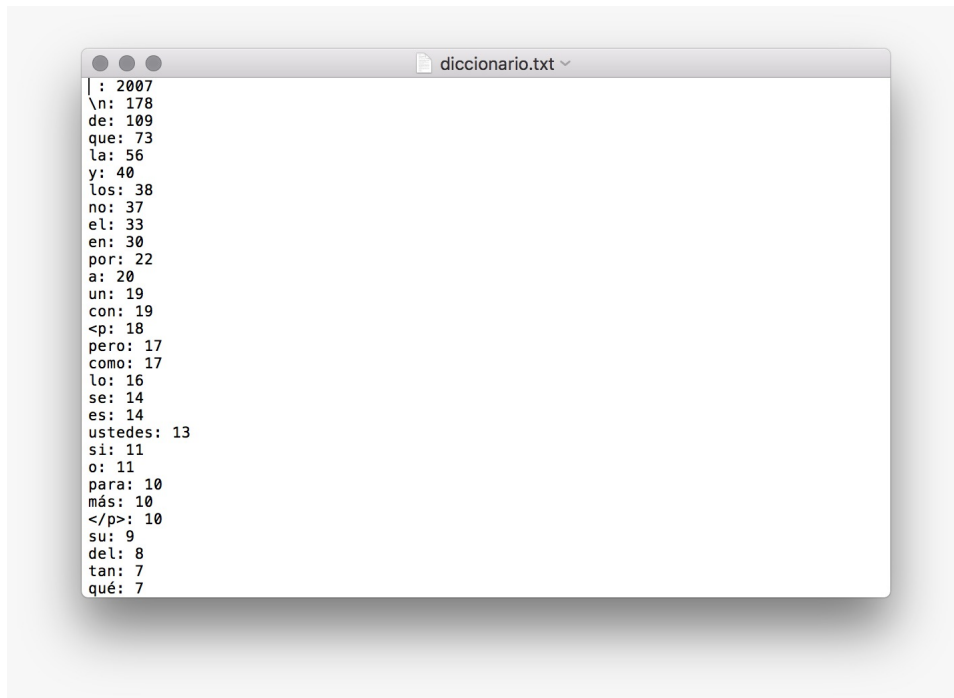


Figure 1: Diccionario de Huffman

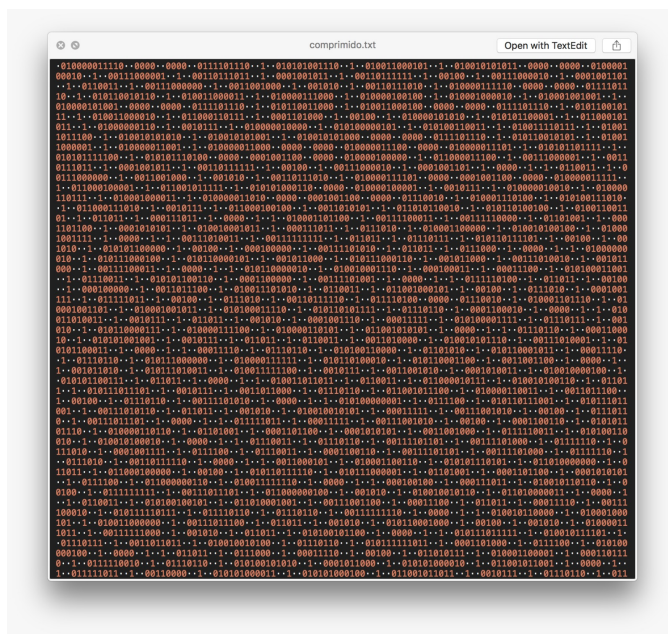


Figure 2: Compresión

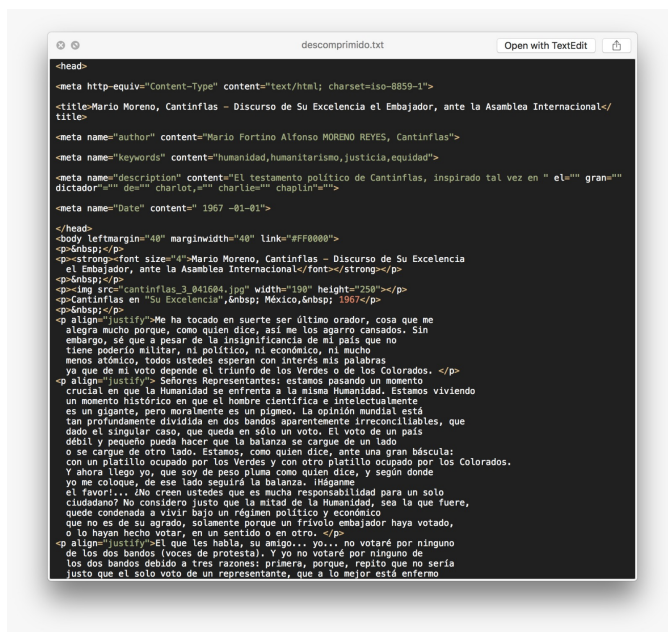


Figure 3: Descompresión

5.2 Archivos

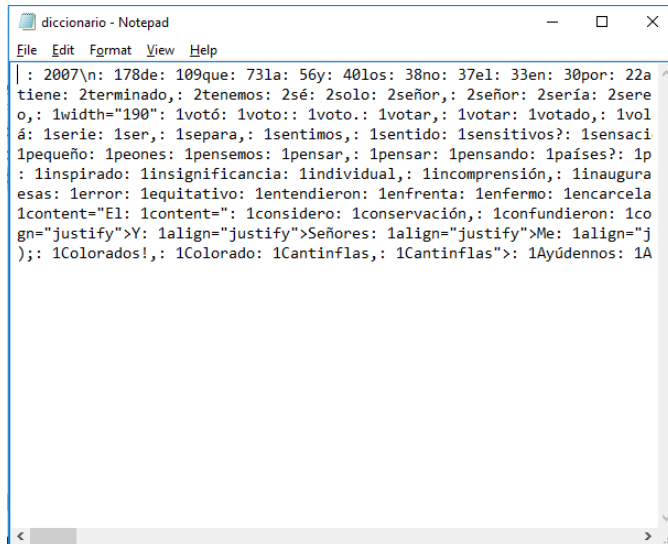


Figure 4: Diccionario de Huffman

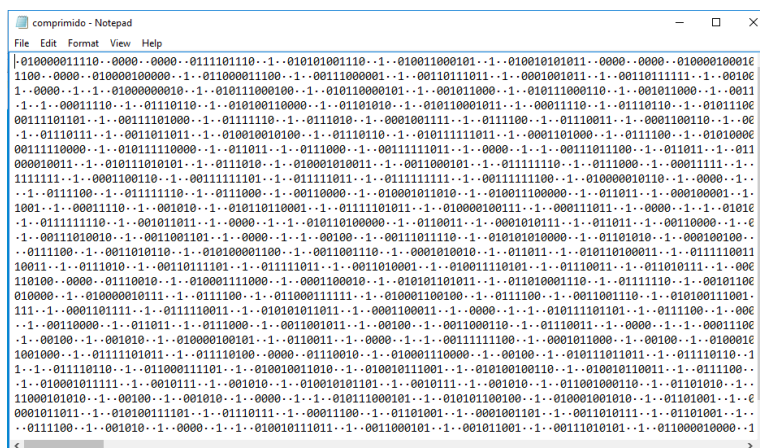
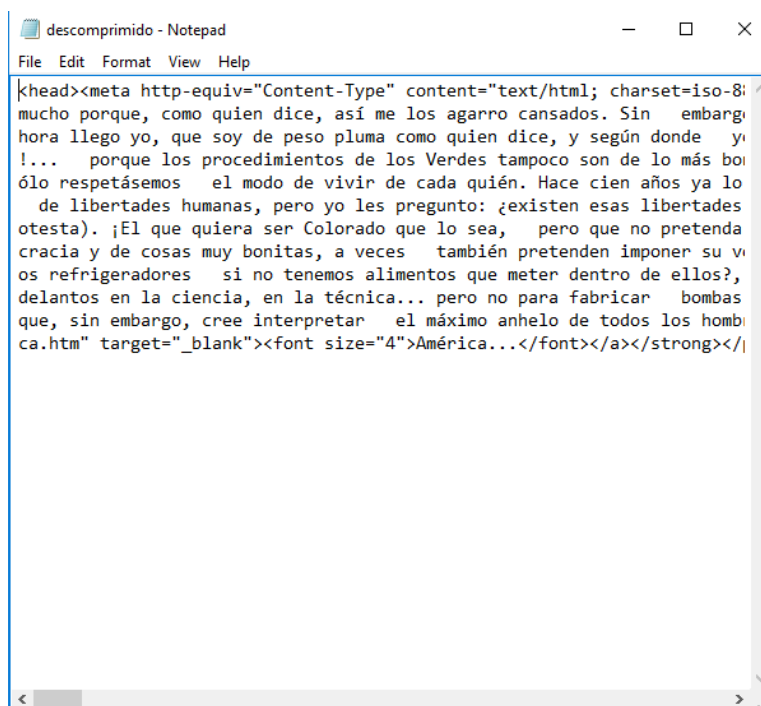


Figure 5: Compresión



```
descomprimido - Notepad
File Edit Format View Help
<head><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
mucho porque, como quien dice, así me los agarro cansados. Sin embargo,
hora llego yo, que soy de peso pluma como quien dice, y según donde yo
!... porque los procedimientos de los Verdes tampoco son de lo más boni
ólo respetásemos el modo de vivir de cada quién. Hace cien años ya lo
de libertades humanas, pero yo les pregunto: ¿existen esas libertades
otesta). ¡El que quiera ser Colorado que lo sea, pero que no pretenda
cracia y de cosas muy bonitas, a veces también pretenden imponer su vi
os refrigeradores si no tenemos alimentos que meter dentro de ellos?,
delantos en la ciencia, en la técnica... pero no para fabricar bombas
que, sin embargo, cree interpretar el máximo anhelo de todos los hombr
ca.htm" target="_blank"><font size="4">América...</font></a></strong></p>
```

Figure 6: Descompresión

6 Conclusión

Durante la realización de este proyecto logramos apreciar el gran potencial de Prolog, el cual brinda muchas ventajas con herramientas como hechos y funtores, que otros lenguajes de programación no brindan. Entre las mayores dificultades que encontramos, la creación del árbol, ya que manejamos muy poco este lenguaje y tiene muy poca información en internet para su desarrollo, se puede encontrar algunos sitios que dan algunos ejemplo, pero no son claros. En cuanto sus características específicas, se tuvo que aprender como es que prolog utiliza el backtracking para instanciar variables, con base a hechos. Con esta tarea logramos ampliar nuestros conocimientos en un lenguaje tan poderoso como prolog pero tan poco conocido y utilizable a comparación con otros lenguajes.