Trabajo Práctico 2: Codificación de Huffman

I206 - Inferencia y Estimación

Grupo 2: Abril Diaco, Agustina Barrena, Pilar Otero

Nros. de legajo: 36763, 36282, 36389
Información de contacto: adiaco@udesa.edu.ar, abarrena@udesa.edu.ar, potero@udesa.edu.ar

Segundo semestre 2025

Resumen

En este informe se estudiaron conceptos fundamentales de la *Teoría de la Información* aplicados al análisis de tres textos mediante el algoritmo de Huffman. Se estimaron las distribuciones de probabilidad de los símbolos presentes en `adn.txt`, `mitología.txt` y `tabla.txt`, se generaron sus codificaciones binarias y se evaluaron la longitud promedio de los códigos L, la entropía $H(X)$, la reducción de memoria respecto de una codificación uniforme y la entropía normalizada η.

Los resultados evidenciaron que la estructura estadística de cada fuente determina directamente su compresibilidad. En `adn.txt`, con símbolos casi equiprobables, se obtuvo $H(X) \approx 2{,}00$ bits/símbolo, $L = 2{,}00$ bits/símbolo, $\eta \approx 1$, y una reducción prácticamente nula. Para `mitología.txt`, característico del lenguaje natural, se obtuvo $H(X) = 4{,}31$ bits/símbolo, $L = 4{,}33$ bits/símbolo, $\eta \approx 0{,}73$ y una reducción del 27,76 %. En `tabla.txt`, con alta repetitividad y símbolos dominantes, se obtuvo $H(X) = 3{,}38$ bits/símbolo, $L = 3{,}40$ bits/símbolo, $\eta \approx 0{,}60$ y la mayor disminución de memoria: 43,26 %.

La comparación de $H(X)$ con L confirmó que los códigos generados se aproximan al límite teórico de compresión sin pérdida. Asimismo, la entropía normalizada permitió comparar fuentes con alfabetos diferentes, concluyéndose que a mayor desigualdad en las probabilidades de los símbolos, mayor redundancia y mayor eficiencia de compresión.

Se concluyó que la experiencia validó la relación entre la estructura probabilística de un texto y la eficiencia del algoritmo de Huffman, demostrando que la optimización de la representación de la información depende no solo del tamaño del mensaje, sino de la distribución de sus símbolos.

1. Introducción

El objetivo principal de la práctica consistió en analizar cómo la distribución de probabilidades de los símbolos de un texto influye en su entropía, en la longitud promedio de codificación y en la eficiencia de compresión alcanzada.

El algoritmo de Huffman constituye un método de codificación óptimo para fuentes discretas sin memoria, es decir, modelos que emiten un conjunto finito de símbolos en los cuales la probabilidad de cada elemento es independiente de las anteriores. El proceso estudiado asigna códigos binarios más cortos a los símbolos más frecuentes de un archivo y más largos a los menos frecuentes, minimizando así la longitud promedio L.

El procedimiento consiste en construir un árbol binario ponderado a partir de las probabilidades de los caracteres, donde en cada paso, se combinan los dos nodos de menor probabilidad hasta formar una única estructura. Finalmente, el recorrido desde la raíz hasta cada hoja determina el código binario correspondiente, que además, es libre de prefijos, lo que significa que ninguna secuencia binaria asignada a un símbolo coincide con el inicio de la secuencia de otro.

A partir de esto, resulta fundamental determinar la probabilidad de aparición de los caracteres por medio del cálculo de la frecuencia relativa, definida como:

$$p(x_i) = \frac{f(x_i)}{M} \tag{1}$$

donde el numerador $f(x_i)$ representa la cantidad de apariciones del caracter x_i, y el denominador M la cantidad total de símbolos que constituyen el texto.

Para evaluar la eficiencia de una codificación no uniforme, se calcula la longitud promedio de bits mediante la fórmula

$$L = \sum_{i=1}^{N} p(x_i) l(x_i) \tag{2}$$

donde N es la cantidad de símbolos diferentes en el texto, x_i el i-ésimo elemento, $p(x_i)$ la probabilidad de aparición del mismo x_i y $l(x_i)$ la longitud (cantidad de bits) asignada por el código de Huffman a dicho elemento.

En una codificación uniforme, donde todos los símbolos son representados con una misma longitud de secuencia, la cantidad mínima de bits necesarios para representar los N caracteres distintos es:

$$L_{\text{uniforme}} = \log_2(N) \tag{3}$$

El porcentaje de reducción de memoria logrado con la codificación de Huffman se expresa como:

$$\text{Reducción} = \frac{L_{\text{uniforme}} - L}{L_{\text{uniforme}}} \times 100 \tag{4}$$

lo cual cuantifica la eficiencia obtenida en relación con una codificación uniforme de igual alfabeto.

Por otro lado, la entropía $H(X)$ mide la incertidumbre asociada a una variable aleatoria X. Desde el punto de vista teórico, se define como la esperanza de la información puntual:

$$H(X) = E[I(X)] = -\sum_{x} p(x) \log_2(p(x)) \tag{5}$$

Esta magnitud representa la cantidad media de información contenida en cada símbolo y establece el límite teórico inferior promedio de bits necesarios para codificar el archivo seleccionado sin pérdida de información.

Para comparar textos con distintos tamaño, se emplea la entropía normalizada:

$$\eta = \frac{H(X)}{\log_2(N)} \tag{6}$$

la cual permite evaluar cuán uniformemente distribuida está la información en un texto, de forma independiente al número total de elementos.

Finalmente, se implementaron las funciones necesarias en Python para estimar la probabilidad de aparición de los caracteres, generar los respectivos códigos de Huffman y calcular las magnitudes teóricas y experimentales correspondientes, sobre tres textos de referencia: `mitología.txt`, `tabla.txt` y `adn.txt`. A partir de los resultados obtenidos, se compararon la eficiencia de compresión alcanzada con la entropía teórica y la longitud promedio de codificación uniforme, evaluando así el desempeño del algoritmo en distintos contextos de distribución simbólica.

2. Metodología

2.1 Frecuencia de aparición

En esta primera etapa, mediante la implementación de la función `PMF(path)`, se determinó la frecuencia de aparición de cada carácter para los textos provistos porla cátedra.

Para ello, se construyó un diccionario de ocurrencias, donde las claves representan los caracteres y los valores corresponden a la cantidad de apariciones de cada uno de los mismos en el texto. A partir de esta estructura, se calculó la frecuencia relativa de aparición de cada símbolo utilizando la expresión presentada en la Ecuación 1.

Posteriormente, se generaron gráficos de barras para facilitar la interpretación de los resultados obtenidos, como la identificación de símbolos predominantes y la distribución de probabilidades de cada texto.

2.2 Operaciones con bits

Esta sección constituye la aplicación central del algoritmo de Huffman, implementado a partir de la función `huffman_code(dict[str:float])`. En este contexto, se desarrolló la función `huffman_texto(str, dict[str:float])`, cuyo propósito es generar, para un texto determinado, el diccionario de codificación resultante de aplicar el algoritmo de Huffman sobre las frecuencias de sus caracteres.

El procedimiento seguido consiste en abrir el archivo correspondiente, determinar el código asociado a cada símbolo y, finalmente, concatenar las secuencias codificadas para obtener el mensaje comprimido.

A partir de estos resultados, se calculó la longitud promedio de bits mediante la función `L(dict[str:float])`, correspondiente a la codificación no uniforme obtenida. Esta magnitud, representada por L, se determina como la suma ponderada de las longitudes asignadas a cada símbolo según su probabilidad de aparición, de acuerdo con la Ecuación 2.

Luego, se buscó la cantidad mínima de bits necesarios para una codificación uniforme, en la que, tal como se mencionó previamente, todos los símbolos son representados con la misma longitud. El proceso de obtención de la misma se baso en la expresión teórica de la Ecuación 3, por medio de `cantidad_bits(dict[str:float])`.

Finalmente, se estimó el porcentaje de reducción de memoria logrado con la codificación de Huffman respecto de la uniforme. Para ello, se implementó la función `porcentaje_reduccion(float, float)`, la cual utiliza la Ecuación 4. Esta última ofrece la comparación entre la dichas codificaciones, cuantificando así, la eficiencia obtenida mediante la compresión y reflejando cuánto se reduce el tamaño del mensaje sin la pérdida de información.

2.3 Entropías

Utilizando las frecuencias relativas de cada carácter en los textos, se calculó la entropía de cada uno a través de la función `entropia(dict)` la cual utiliza la Ecuación 5 y hace uso de las probabilidades halladas previamente.

Los resultados obtenidos fueron posteriormente comparados con las longitudes promedio de codificación (L) de cada archivo con el fin de analizar la relación entre la entropía teórica y la eficiencia del algoritmo de Huffman.

2.4 Entropías normalizadas

En último lugar, en esta etapa se calculó la entropía normalizada con el fin de obtener para cada texto, un valor comprendido en el intervalo $[0, 1]$. Para ello, se empleó la Ecuación 6, implementada en la función `entropia_normalizada(dict)`.

La entropía en cuestión permite comparar directamente las distribuciones de los distintos textos, incluso cuando difieren en longitud o tamaño del alfabeto. En particular, un valor cercano a 1 indica una distribución más uniforme de probabilidades (mayor incertidumbre), mientras que valores cercanos a 0 reflejan una fuente más predecible y con símbolos dominantes.

3. Resultados

Al calcular la probabilidad de aparición estimada de cada carácter para los textos analizados, se obtuvieron las distribuciones de frecuencia representadas en las Figuras 1, 2 y 3. Estas gráficas permiten visualizar de manera comparativa cómo varía la probabilidad relativa de los símbolos según el contenido y la estructura de cada archivo.

Con el objetivo de resaltar las diferencias en las frecuencias individuales, todas las representaciones se realizaron en escala logarítmica, lo cual permite distinguir con mayor claridad las variaciones entre símbolos poco frecuentes y aquellos más dominantes.

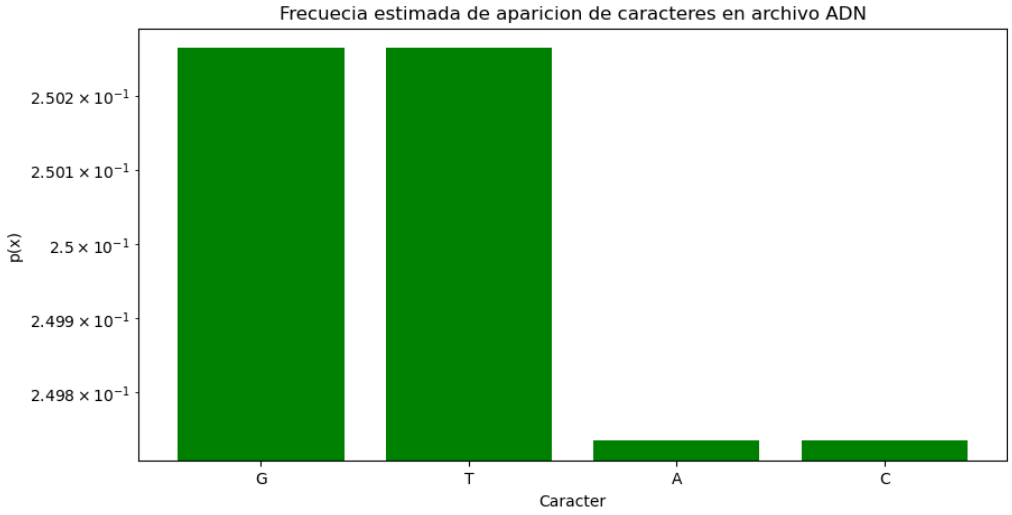

Figura 1: Frecuencia estimada de aparición de caracteres en el archivo `ADN`. La representación fue realizada en escala logarítmica. Se observa una baja cantidad N de símbolos distintos, con probabilidades relativamente altas para los caracteres `G` y `T`, y considerablemente menores para `A` y `C`.

La Figura 1 muestra la distribución de frecuencias para el archivo `ADN`. El número de símbolos distintos es reducido ($N = 4$), correspondiente a los componentes del ADN : `A`, `T`, `C` y `G`. Las probabilidades resultan cercanas entre sí, aunque con una ligera predominancia de `G` y `T`. Las pequeñas diferencias en las frecuencias entre los símbolos pueden deberse a la composición particular del fragmento de ADN analizado. Así, el conjunto es casi equiprobable, con escasa redundancia estadística, consistente con una secuencia donde no hay predominio extremo de un símbolo por sobre el resto.

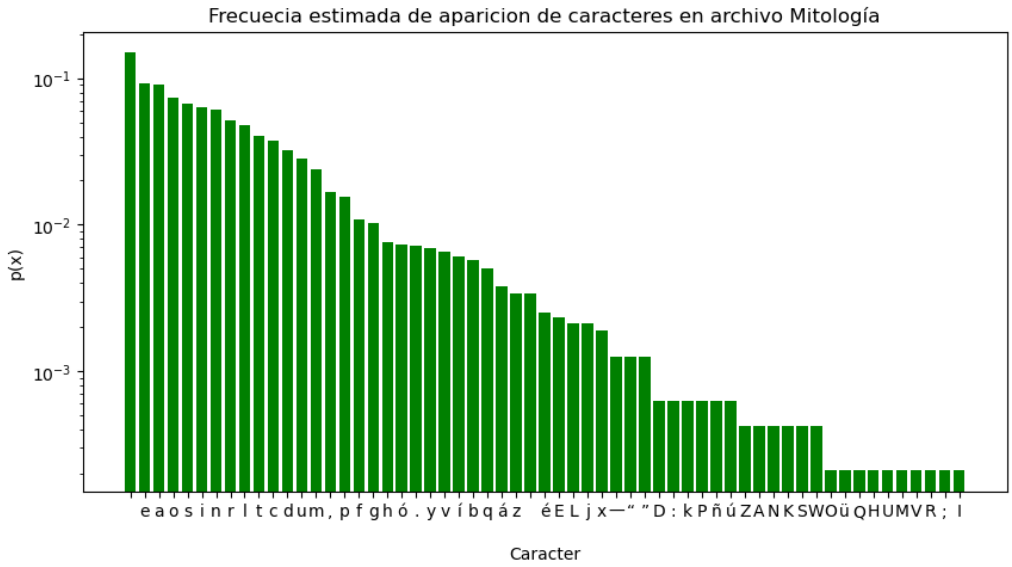

Figura 2: Frecuencia estimada de aparición de caracteres en el archivo `Mitología`. La representación se realizó en escala logarítmica. Se aprecia una amplia variedad de símbolos con frecuencias que decrecen de manera aproximadamente exponencial. Los caracteres más frecuentes corresponden a las vocales y consonantes más comunes del idioma español, los signos de puntuación, caracteres acentuados y mayúsculas presentan baja frecuencia.

En contraste, la Figura 2 muestra una distribución fuertemente más amplia. El patrón de frecuencias decrece de manera aproximadamente exponencial: unas pocas letras concentran gran parte de la probabilidad total, mientras que el resto —signos, mayúsculas o caracteres especiales, como los acentuados— tiene menor participación. Este patrón surge por la alta frecuencia de las vocales y consonantes más comunes del español, comportamiento esperado debido a que el texto en cuestión involucra un amplio vocabulario.

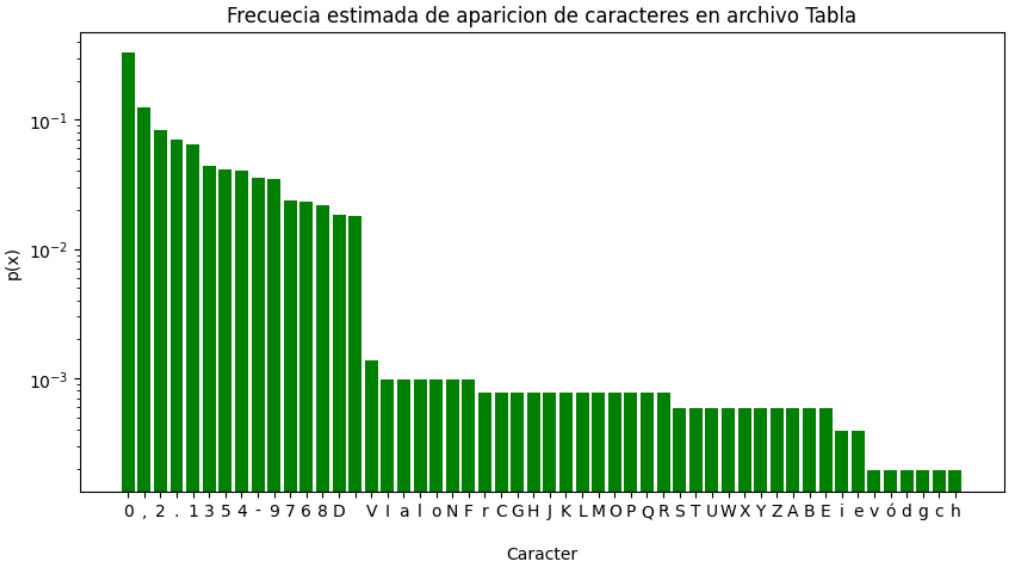

Figura 3: Frecuencia estimada de aparición de caracteres en el archivo `Tabla`. Representación en escala logarítmica. La distribución muestra una cantidad moderada de símbolos distintos. Se destacan los dígitos y separadores (como el espacio o la coma) con alta frecuencia relativa, seguidos por letras asociadas a los elementos de la tabla. Esto evidencia un alfabeto mixto compuesto por números, letras y símbolos, reflejando la estructura tabular del archivo.

Por último, la Figura 3 presenta la distribución de frecuencias del archivo `Tabla`. En este caso, el alfabeto observado es mixto, incluyendo letras, dígitos y separadores, lo cual refleja la estructura tabular del texto. Se observa una clara predominancia de los separadores, como el punto, la coma y los espacios, seguidos por los caracteres alfabéticos, sobretodo las mayúsculas. La caída de las frecuencias mantiene una tendencia decreciente escalonada, indicando una mezcla entre regularidad y dispersión en la aparición de los símbolos: encontramos elementos repetidos, asi como otros con una menor proporcion de apariciones.

Con respecto a los ejercicios dos y tres se extrajeron los resultados presentados a continuación para los diferentes textos.

En el archivo `adn.txt`, se obtuvo una longitud promedio del código de Huffman de $L = 2{,}00$ bits por símbolo, mientras que un código uniforme requeriría también 2 bits por símbolo. El porcentaje de reducción fue prácticamente nulo ($\approx 1{,}1 \times 10^{-14}\,\%$), y la entropía calculada fue $H(X) = 1{,}999999$ bits por símbolo.

Al comparar la entropía $H(X)$ con la longitud media de codificación L, se observa que ambos valores son prácticamente iguales. Esto indica que el código de Huffman obtenido es óptimo para una fuente casi equiprobable, ya que su longitud promedio se aproxima al límite teórico inferior de compresión sin pérdida.

Bajo lo postulado por la *Teoría de la Información* esto implica que la fuente, `adn.txt`, posee una distribución de probabilidades muy uniforme, lo que hace que el

código fijo de 2 bits por símbolo sea ya prácticamente óptimo, sin margen apreciable de compresión adicional, es decir, sin poder reducir significativamente el tamaño del archivo sin perder información.

Para el texto `mitología.txt`, la longitud promedio del código de Huffman fue de $L = 4{,}33$ bits/símbolo, frente a los 6 bits requeridos por un código fijo, lo que representa una reducción del 27,76 %. La entropía calculada fue $H(X) = 4{,}31$ bits/símbolo.

Al comparar la entropía $H(X)$ con la longitud promedio L, se observa que ambas magnitudes son muy cercanas, lo que indica que el código de Huffman se aproxima notablemente al límite teórico de compresión sin pérdida. En este caso, a diferencia del archivo `adn.txt`, la fuente presenta una distribución de símbolos fuertemente desigual, propia del lenguaje natural: unas pocas letras concentran la mayor parte de la probabilidad, mientras que otras aparecen con muy baja frecuencia. Esta estructura introduce redundancia, la cual le permite al código de Huffman asignar códigos más cortos a los símbolos más frecuentes, logrando así una compresión efectiva sin pérdida de información.

En términos de *Teoría de la Información*, este comportamiento refleja que el texto *Mitología* constituye una fuente de complejidad intermedia, es decir, ni completamente uniforme ni extremadamente diversa, por lo que hay un cierto grado de redundancia aprovechable. El algoritmo de Huffman logra explotar dicha redundancia eficientemente, generando un código cuya longitud promedio se encuentra muy próxima al límite teórico óptimo.

La comparación entre $H(X)$ y L confirma que el código de Huffman se comporta casi de forma ideal para este tipo de fuentes, permitiendo una compresión efectiva y sin pérdida de información, reflejando los patrones de frecuencia y redundancia típicos del lenguaje natural.

Por último, para el archivo `Tabla.txt`, se obtuvo una longitud promedio de $L = 3{,}40$ bits por símbolo, mientras que el código uniforme requeriría 6 bits, lo que implica una reducción del 43,26 %. La entropía correspondiente fue $H(X) = 3{,}38$ bits por símbolo.

Tal como en los casos anteriores, se verifica que $H(X)$ y L son valores proximos, lo que indica una buena aproximación al límite teórico. No obstante, la fuente presenta una marcada desigualdad en las frecuencias de los símbolos, producto de la estructura repetitiva del texto (números, comas y puntos). Esto se traduce en una menor entropía absoluta, ya que la aparición de ciertos caracteres puede predecirse con alta probabilidad. La alta redundancia del conjunto permite una compresión más eficiente, otorgando al algoritmo de Huffman un desempeño superior frente a los otros textos analizados.

Dado que los tres textos analizados contienen diferentes cantidades de símbolos, se recurrió al cálculo de la entropía normalizada, que ajusta el valor de $H(X)$ al tamaño del alfabeto y posibilita una comparación directa entre las fuentes.

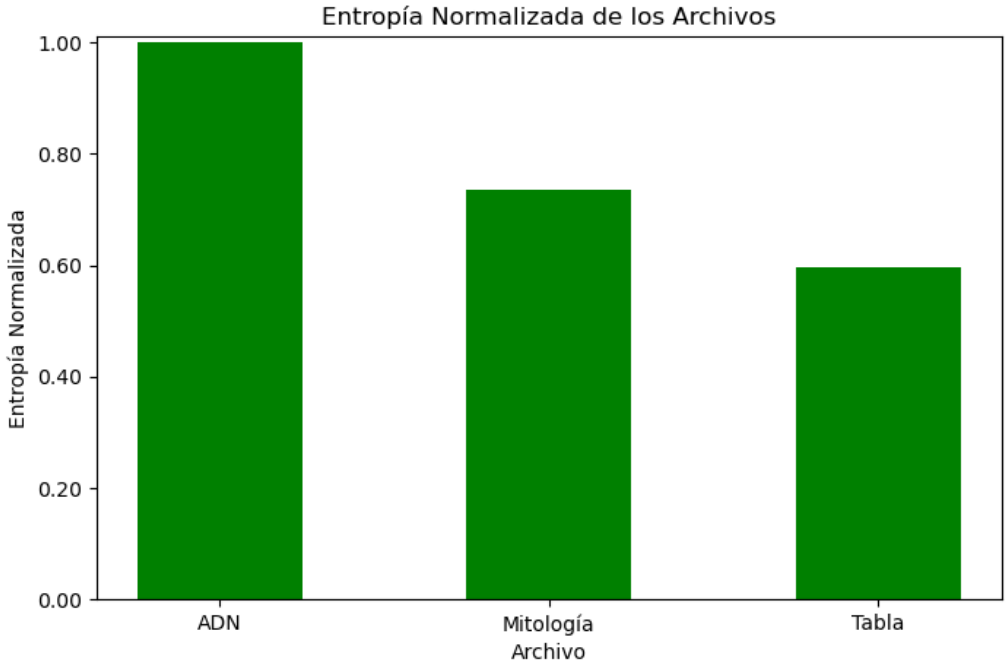

Figura 4: Comparación de las entropías normalizadas de los archivos estudiados. El archivo *ADN* presenta la mayor incertidumbre, mientras que *Tabla* la menor de las tres.

En la Figura 4 se comparan las entropías normalizadas de los archivos analizados. Los valores obtenidos fueron: para `adn.txt` $\approx$ 0,9999996 bits por símbolo, para `mitología.txt`$\approx$ 0,7349469 bits por símbolo y en `tabla.txt` $\approx$ 0,5953473 bits por símbolo.

Queda en evidencia de esta manera, que las tres fuentes presentan entropías normalizadas superiores a 0,5, lo cual indica que ninguna es completamente uniforme, pero sí lo suficientemente equilibrada para permitir una compresión eficiente al hacer uso del código de Huffman.

Siguiendo esta línea de análisis, se halla aquí la razón por la cual la entropía del texto `adn.txt` alcanza un valor prácticamente unitario: a diferencia de los otros archivos, los cuales presentan una mayor variedad de caracteres con cierta regularidad en su aparición, este texto, al estar constituido por 4 simbolos (A,C,G,T) con probabilidades muy parecidas, tiene una distribucion homogénea de los mismos. Como consecuecia, la fuente refleja una menor redundancia en su contenido y su compresión resulta mínima.

Por su parte, el archivo `mitología.txt`, presenta una entropía normalizada intermedia, lo que refleja una distribución de símbolos más desigual, típica del lenguaje natural. Algunas letras (como las vocales y las consonantes más utilizadas) dominan la frecuencia total, mientras que otras, como los signos de puntuación, tienen apariciones escasas. Esta heterogeneidad en la distribución de los símbolos implica que algunos caracteres aparecen con mayor probabilidad que otros, lo que genera un cierto grado de redundancia y predictibilidad en el texto. En otras palabras, la aparición de un símbolo no es completamente aleatoria, sino que puede anticiparse en función de su frecuencia global.

Finalmente, el archivo `Tabla` presenta la menor de las entropias entre los archivos analizados, lo cual sugiere una estructura repetitiva y una distribución desigual de sus elementos. En este tipo de fuentes, la información se concentra en unos pocos

elementos dominantes, los cuales aparecen con una alta frecuencia, mientras que el resto ocurre muy esporádicamente. Esta concentración provoca que la incertidumbre promedio caiga notablemente. Desde el punto de vista de la Teoría de la Información, esto se traduce en una entropía más baja ya que, la aparición de los símbolos más comunes puede predecirse con mayor facilidad.

En conjunto, los resultados confirman que la entropía normalizada está inversamente relacionada con la redundancia, entendida como el grado de previsibilidad dentro de una fuente: a menor entropía, mayor redundancia, lo que implica que una parte del mensaje puede predecirse, y por lo tanto, representarse con una menor cantidad de bits sin pérdida de información.

Así, identificamos que los tres archivos son de distintos grados de complejidad informacional: desde la aleatoriedad casi perfecta del `adn.txt`, pasando a la riqueza lingüística y estructurada de `mitología.txt`, hasta la regularidad fuertemente predecible de `Tabla.txt`.

4. Conclusiones

El desarrollo de este informe permitió aplicar y comprender en profundidad el proceso detrás del reconocido algoritmo de Huffman, junto con los beneficios que este conlleva, en particular la reducción del espacio ocupado en memoria por los archivos a través de una codificación eficiente. Su implementación práctica permitió comprobar cómo las longitudes de los códigos asignadas a cada símbolo se ajustan directamente a sus probabilidades de aparición, alcanzando así una representación óptima sin pérdida de información.

Un aspecto destacado fue la comparación entre textos de distintos formatos, alfabetos y longitudes, lo cual representó el mayor de los desafíos tanto conceptual como práctico. Esta diversidad implicó que los valores obtenidos para cada caso particular -como la longitud promedio, la entropía y el porcentaje de reducción- no fueran inmediatamente comparables entre sí. Por tal motivo, fue necesario analizarlos de forma relativa mediante medidas adimensionales, como la entropía normalizada, que permitió establecer un criterio común para la comparación de fuentes de información heterogéneas.

Simultáneamente, el análisis detallado de textos con estructuras y contenidos diversos permitió observar de manera más clara y visual cómo las diferencias en la distribución de los caracteres impactan directamente en la compresibilidad de la información. Se evidenció así que, cuanto más desigual es la distribución de probabilidades de los símbolos, mayor es la reducción de memoria alcanzada por el algoritmo de Huffman. En cambio, para textos con distribuciones casi uniformes —como el caso de `adn.txt`— el margen de compresión es prácticamente nulo, ya que el código uniforme se aproxima al límite teórico de eficiencia.

Desde el punto de vista teórico, los resultados verifican la coherencia entre la entropía teórica $H(X)$ y la longitud promedio de codificación L. Esta correspondencia demuestra que Huffman no sólo minimiza el largo esperado de los códigos, sino que también refleja con precisión la estructura probabilística subyacente de la fuente.

En síntesis, el desarrollo de este informe permitió integrar teoría y práctica al vincular conceptos de probabilidad, entropía e información con su aplicación computacional concreta. Los resultados obtenidos reforzaron la comprensión del vínculo

entre la estructura estadística de los datos y la eficiencia de su representación, mostrando que la información contenida en un mensaje no solo depende de su longitud, sino también de la distribución con que aparecen sus símbolos. En consecuencia, el estudio constituye una validación empírica de los principios de la Teoría de la Información y evidencia la relevancia del algoritmo de Huffman como herramienta fundamental en los sistemas modernos de compresión sin pérdida.