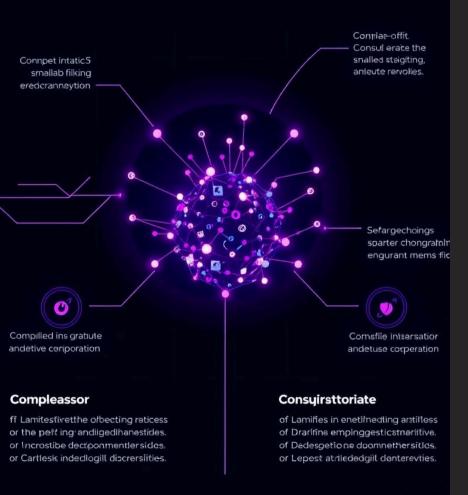
# DATA SSTIEEM INCMPSITISE

Arii this this steantheds Streer and fest Incolingtiou, lenal Ic mury your botingreeam fon thecing us nel comergers, the onest indust luntractionand smalle in yvorudevidesr.



# Presentación de Compresión de Datos

La compresión de datos es una técnica esencial en la era digital. Permite reducir el tamaño de los archivos, optimizar el almacenamiento, acelerar las transmisiones y mejorar la eficiencia general de las operaciones.

# Introducción a la Compresión de Datos

# 1 Eficiencia de Almacenamiento

3

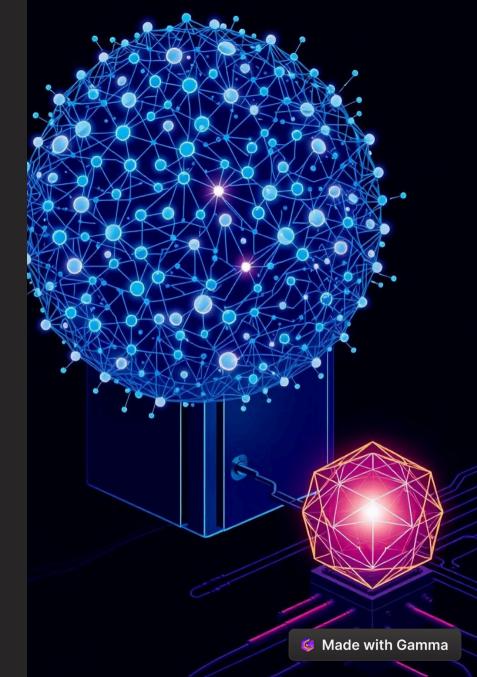
La compresión reduce el espacio requerido para almacenar datos, liberando almacenamiento y optimizando la gestión de archivos.

# **Velocidad de**Transmisión

La compresión permite transmitir datos más rápidamente, reduciendo el tiempo de descarga y mejorando la experiencia del usuario.

# Optimización de Recursos

La compresión minimiza el uso de ancho de banda, optimizando los recursos de red y mejorando la eficiencia de la transmisión.



# Conceptos Clave y Motivación

## Compresión Sin Pérdida

Preserva toda la información original, asegurando que los datos reconstruidos son idénticos al original.

## **Compresión Con Pérdida**

Elimina información irrelevante para reducir el tamaño, con la posibilidad de una pequeña pérdida de calidad.

### Motivación

La creciente generación y el almacenamiento de datos exigen métodos para optimizar el manejo de la información.



# Principios Básicos de la Compresión de Datos

1 2

### Redundancia

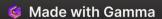
Identifica y elimina patrones repetitivos en los datos para reducir el tamaño.

## Entropía

Mide la cantidad de información contenida en los datos, optimizando la compresión.

### Codificación

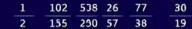
Utiliza técnicas para representar los datos de manera más eficiente, reduciendo el tamaño.



## **Huffmanhl** Desecne

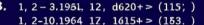


#### A: Huffman tree, (n/weighte) nodes





#### Conbercive nodes & weighing code perparnating toces symbol;



A weccode / trabet to to teagh code

#### Tablt Secount coris tre sample ILC Fuel Not callc 190 122 134 d0..67 155 126 578 61043



Samplul code frequencly:

Facting codes misstage

A. 13, 121 75, 130 45, 201 222 0108

#### Final Huffman code:

#### Frinal Huffman message

121 lte606 : 2664 ecerup T2. 44 7,9007: 2024 264 53.72

50-14 2.4800: 2035 206 14.74

20, 44 3,00071 2280 209 96,78

T5. 18 1.96002 2027 207 94.48

T5. 18 7,98062 1286 228 14.75

T2. 12 7.90067 2286 200 36.13

Ctn: irisballIf 4ceal:

180: 10012648756261492785) ict:200292/22249867/2506

AADC 1900121349909267566155

AADC 0000124998576629515317

AACC 0000:24399994525592100

AADC 0800:21399966563547536

AACC 0000121329509411562107

#### De huffman code table

The the cesicoderisiclude, cont att the codetode; for encoconse auring aseitheinet foregences' viict tht cone ccodecodats.

The int noclooder for coderated gat the codercctu; econcdenapet act the coderedthat forecconse) tact tequihe came, nodes.

# El Algoritmo de Codificación de Huffman

| Propósito  | Codificar símbolos con una<br>longitud variable para minimizar<br>el tamaño de los datos.            |
|------------|--|
| Base       | Construir un árbol binario que<br>prioriza los símbolos más<br>frecuentes con códigos más<br>cortos. |
| Resultados | Compresión eficiente,<br>especialmente para datos con<br>patrones repetitivos.                       |

# Construcción del Árbol de Huffman

1

### Frecuencias de Símbolos

Determinar la frecuencia de aparición de cada símbolo en los datos.

2

# Ordenamiento y Combinación

Ordenar los símbolos por frecuencia y combinar los dos menos frecuentes en un nodo padre.

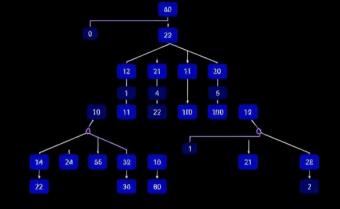
3

## Repetición

Repetir el proceso de combinación hasta que todos los símbolos estén en el árbol.

# **Huffman Tree**

Create:



Startwith set of characters with their frequency, rengeritil the two nodes frequency, frequency, creating the two paten and with the sum of the hiffimren's frequencis.

```
1.
10
10
33
19
11
10
1.
17
11
1.
1.
4
13
16
1.

1.
12
10
15
16
11
16
10
10
11
1.
1.
5
15
13
1.

1.
12
18
16
19
17
10
16
30
11
1.
1.
1.
11
15
1.

1.
10
16
12
34
10
16
15
50
11
1.
1.
9
23
33
1.

1.
02
10
10
30
19
16
10
30
10
1.
12
1.
23
33
1.

3.
05
16
16
10
36
30
00
00
10
10
22
1.
23
23
17

3.
00
19
00
14
24
53
18
00
11
```

inal createp 3.1 Assign lodes untioally reger step todes near by high left campter rear root, and lifer right branch.



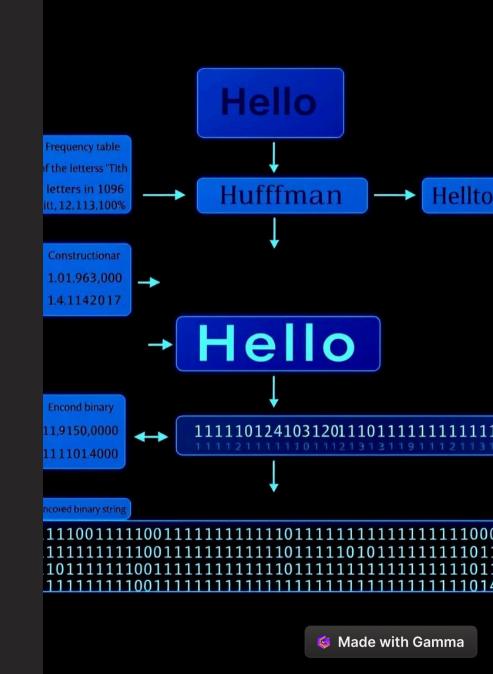
# Codificación y Decodificación con Huffman

### Codificación

Recorrer el árbol desde la raíz hasta el símbolo, asignando 0 para el camino izquierdo y 1 para el derecho.

## Decodificación

Recorrer el árbol de acuerdo a la secuencia de bits codificada para reconstruir el símbolo original.



# Eficiencia y Ventajas de Huffman



# **Compresión Optimizada**

Algoritmo eficiente que logra una alta compresión, especialmente para datos con repetición.



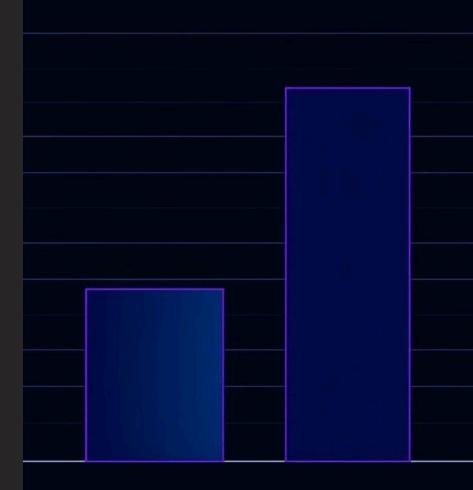
### **Flexibilidad**

Se adapta a diferentes tipos de datos, incluyendo texto, imágenes y archivos multimedia.



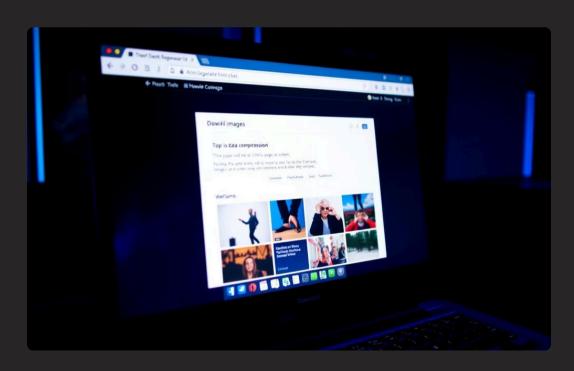
# **Simplicidad**

Relativamente fácil de implementar y comprender, con un proceso claro de construcción del árbol.



Made with Gamma

# Aplicaciones de la Compresión de Datos





## Navegación Web

Optimiza la velocidad de carga de páginas web y reduce el consumo de ancho de banda.

## **Transmisión Multimedia**

Permite la transmisión de videos y música de alta calidad a través de conexiones de red limitadas.

# **Conclusiones y Consideraciones Finales**

La compresión de datos ha revolucionado la forma en que manejamos y almacenamos información. El algoritmo de Huffman es una técnica fundamental que ha contribuido significativamente al desarrollo de la tecnología de compresión.

