****

**UNIVERSIDAD TÉCNICA DE ORURO**

**FACULTAD NACIONAL DE INGENIERÍA**

**INGENIERÍA INFORMÁTICA**



PROYECTO INF – 2610 A

“TEORIA DE LA INFORMACION”

**COMPRESION Y DESCOMPRESION DE FICHEROS**

**Docente:** Ing. Franz Chinche Imaña

**Estudiante:** Alex Baltazar Castellón

Oruro – Bolivia

2023

**ÍNDICE GENERAL DE CONTENIDOS**

[1 ASPECTOS GENERALES 3](#_Toc152440060)

[1.1 Objetivo 3](#_Toc152440061)

[2 MARCO TEORICO 3](#_Toc152440062)

[2.1 ALGORITMO DE HUFFMAN 3](#_Toc152440063)

[2.2 CODIFICACION DE HUFFMAN 4](#_Toc152440064)

[2.3 DISEÑO E IMPLEMENTACION 6](#_Toc152440065)

[3 CONCLUSIONES 8](#_Toc152440066)

[3.1 CONCLUSION 8](#_Toc152440067)

[4 ANEXOS 9](#_Toc152440068)

# **ASPECTOS GENERALES**

## Objetivo

Diseñar un software (SW) que permita comprimir y descomprimir un archivo .txt aplicando codificación de Huffman.

# **MARCO TEORICO**

## ALGORITMO DE HUFFMAN

El algoritmo de Huffman es un algoritmo para la construcción de códigos de Huffmann, desarrollado por David A. Huffmann en 1952 y descrito en “A Method for the Construction of Minimum-Redundancy Codes”. Este algoritmo toma un alfabeto de n símbolos, junto con sus frecuencias de aparición asociadas, y produce un código de Huffman para ese alfabeto y esas frecuencias.

El algoritmo consiste en la creación de un árbol binario que tiene cada uno de los símbolos por hoja, y construido de tal forma que siguiéndolo desde la raíz a cada una de sus hojas se obtiene el código Huffman asociado.

1. Se crean varios árboles, uno por cada uno de los símbolos del alfabeto, consistiendo cada uno de los árboles en un nodo sin hijos, y etiquetado cada uno con su símbolo asociado y su frecuencia de aparición.

2. Se toman los dos árboles de menor frecuencia, y se unen creando un nuevo árbol. La etiqueta de la raíz será la suma de las frecuencias de las raíces de los dos árboles que se unen, y cada uno de estos árboles será un hijo del nuevo árbol. También se etiquetan las dos ramas del nuevo árbol: con un 0 la de la izquierda, y con un 1 la de la derecha.

3. Se repite el paso 2 hasta que sólo quede un árbol.

Con este árbol se puede conocer el código asociado a un símbolo, así como obtener el símbolo asociado a un determinado código.

Para obtener el código asociado a un símbolo se debe proceder del siguiente modo:

1. Comenzar con un código vacío

2. Iniciar el recorrido del árbol en la hoja asociada al símbolo

3. Comenzar un recorrido del árbol hacia arriba

4. Cada vez que se suba un nivel, añadir al código la etiqueta de la rama que se ha recorrido

5. Tras llegar a la raíz, invertir el código

6. El resultado es el código Huffman deseado

Para obtener un símbolo a partir de un código se debe hacer así:

1. Comenzar el recorrido del árbol en la raíz de éste

2. Extraer el primer símbolo del código a descodificar

3. Descender por la rama etiquetada con ese símbolo

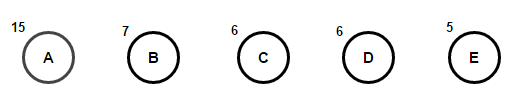
4. Volver al paso 2 hasta que se llegue a una hoja, que será el símbolo asociado al código

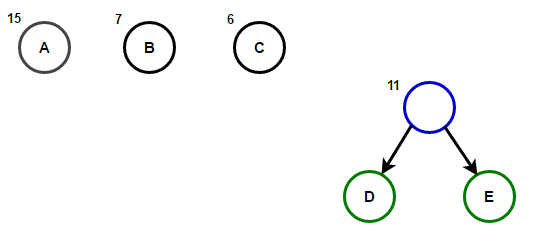
En la práctica, casi siempre se utiliza el árbol para obtener todos los códigos de una sola vez; luego se guardan en tablas y se descarta el árbol.

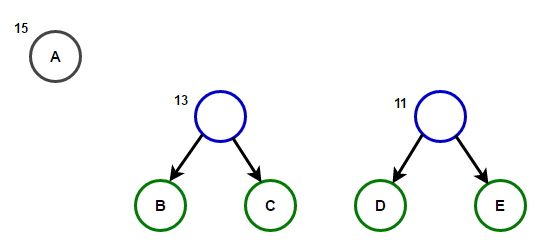
## CODIFICACION DE HUFFMAN

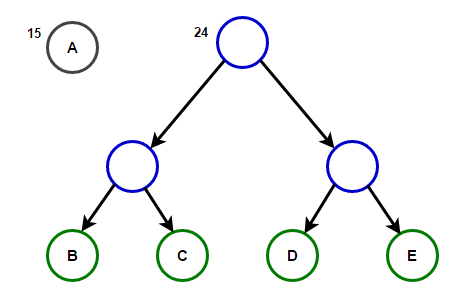
La codificación Huffman (también conocida como Codificación Huffman) es un algoritmo para realizar la compresión de datos y forma la idea básica detrás de la compresión de archivos. Esta publicación habla sobre la codificación de longitud fija y de longitud variable, los códigos decodificables únicos, las reglas de prefijo y la construcción del árbol de Huffman.

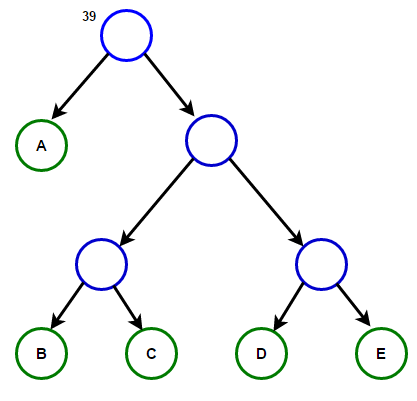
Considere un texto que consta de sólo 'A', 'B', 'C', 'D', y 'E' caracteres y sus frecuencias son 15, 7, 6, 6, 5, respectivamente. Las siguientes figuras ilustran los pasos seguidos por el algoritmo:



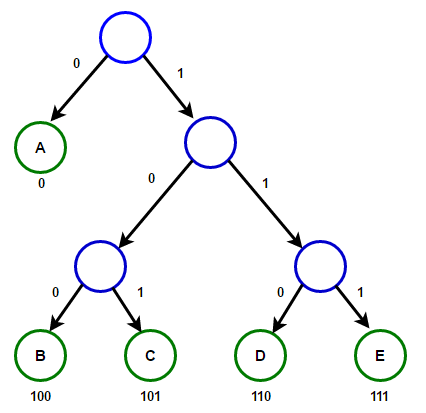








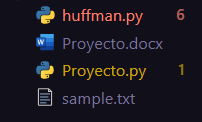
La ruta desde la raíz hasta cualquier nodo hoja almacena el código de prefijo óptimo (también llamado código Huffman) correspondiente al carácter asociado con ese nodo hoja.



## DISEÑO E IMPLEMENTACION

Dicho software lo implementaremos en lenguaje de programación Python, el proyecto contara con 3 archivos:

* Proyecto.py: Es donde estará el main del programa y llamara funciones.
* huffman.py: Contendra todas las clases, funciones para lograr la codificación Huffman.
* sample.txt: Sera un archivo de texto donde contiene el texto que deseamos comprimir (véase anexo2).



Al compilar y ejecutar el programa este creara dos archivos:

* sample.bin: Donde contendrá el código en binario (.bin) del texto en comprimido, lo cual el archivo estará para formato de grabar CD/DVD (véase anexo 3)
* sample\_descomprimido.txt: Sera el archivo descomprimido.



# **CONCLUSIONES**

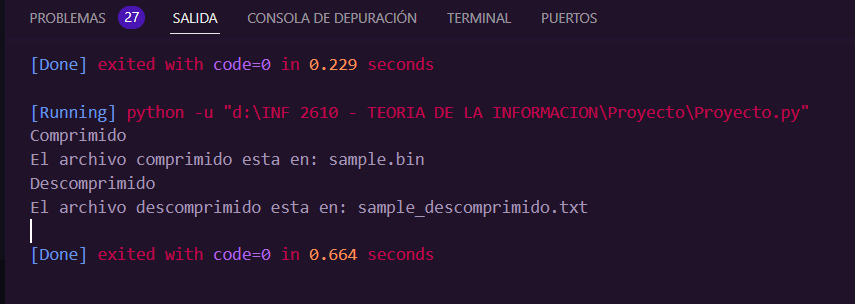
## CONCLUSION

Se logro diseñar con satisfacción el software (SW) el cual tiene la funcionalidad de comprimir y descomprimir ficheros, usando la codificación Huffman y aplicando el algoritmo de Colas de Montículo (heap) haciendo uso de las librerías correspondientes.

# **ANEXOS**

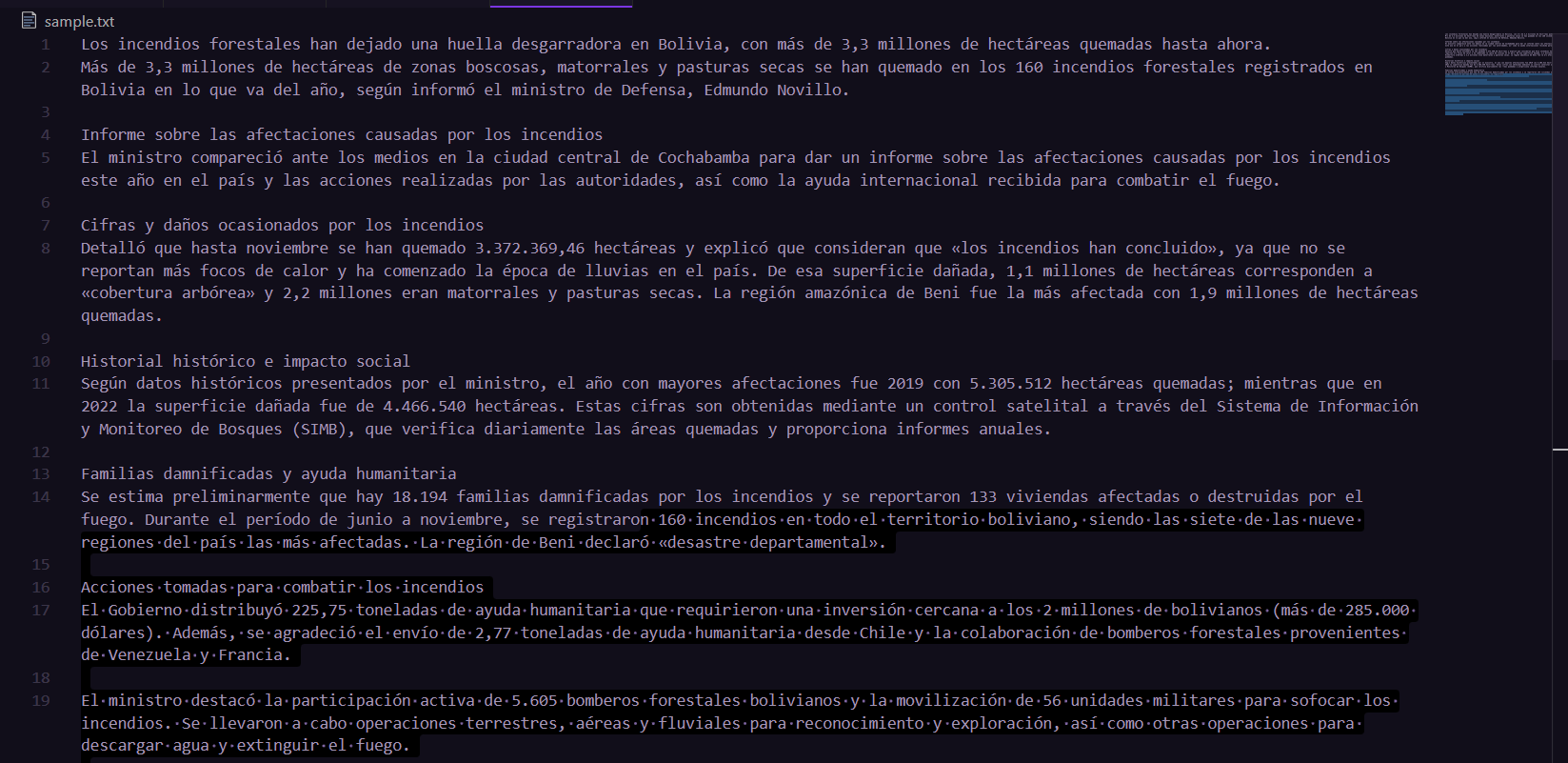
**ANEXO 1**

**EJECUCIÓN DEL PROGRAMA**

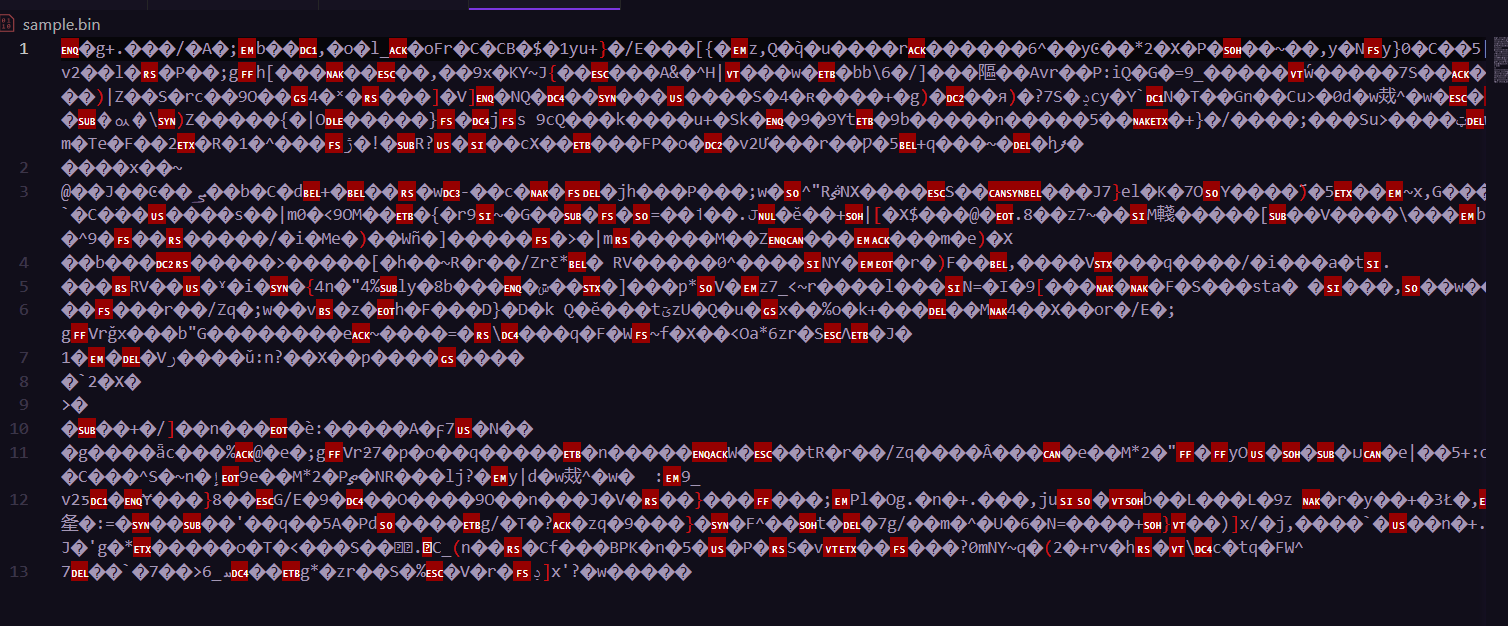


**ANEXO 2**

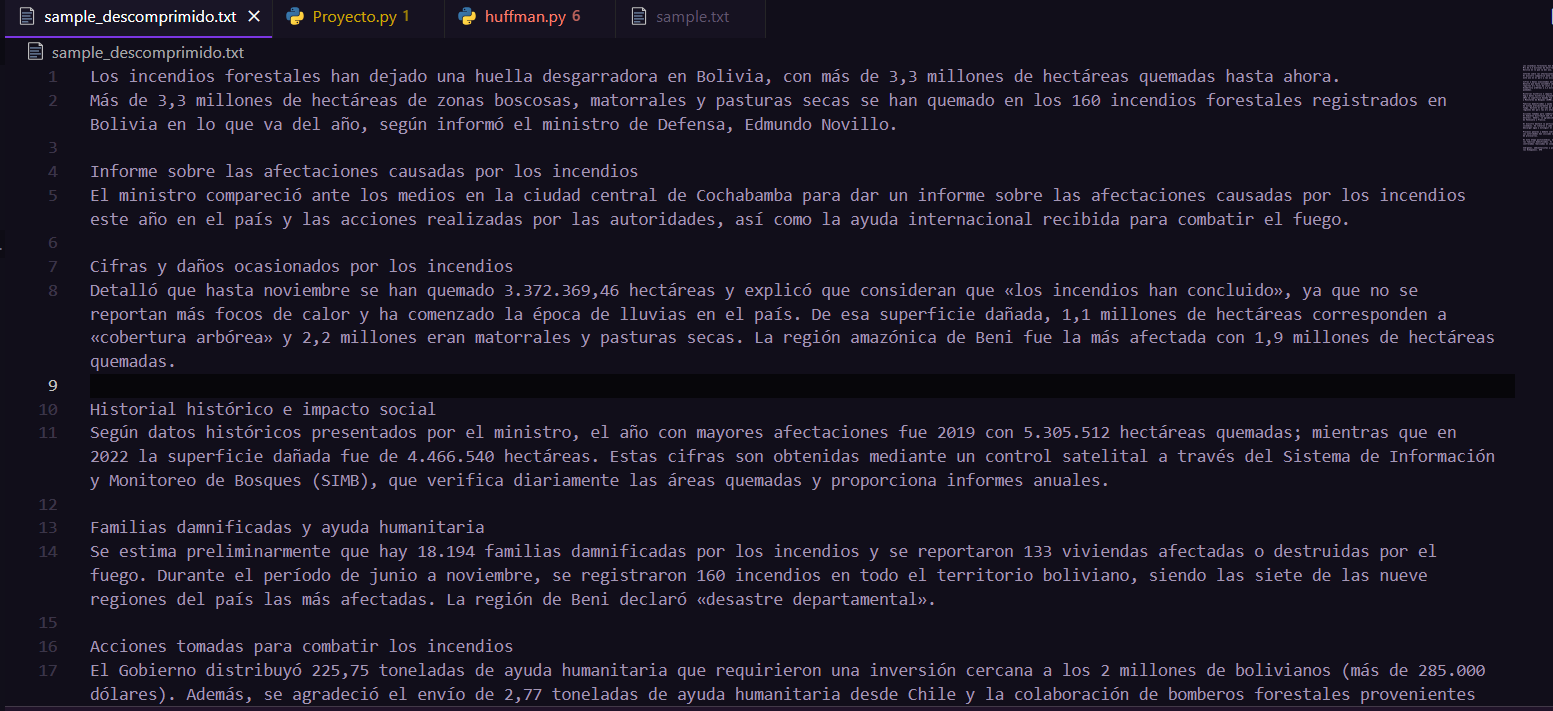
**CONTENIDO DE SAMPLE.TXT**



**ANEXO 3**

**SAMPLE.BIN**

**ANEXO 5**

**SAMPLE.DESCOMPRIMIDO**