



Centro Universitario de Ciencias Exactas e  
Ingenierías

## ANÁLISIS DE ALGORITMOS



### Actividad 8: desarrollo backend del algoritmo de Hoffman

**Carrera:** Ingeniería en computación.

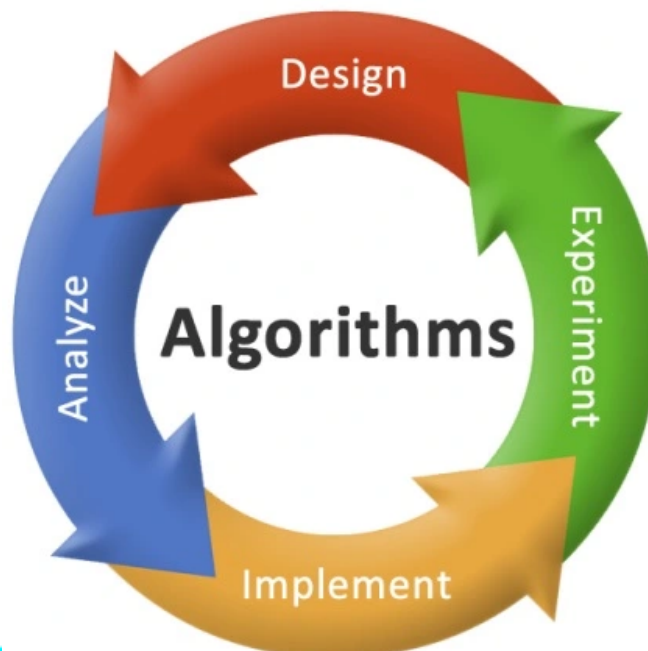
**Mtro.** Lopez Arce Delgado Jorge Ernesto

**Alumno:** Escalante Fernández Fabián Joheshua

**Código:** 222966855

**Sección:** D01

**Fecha:** 28/04/2024



## Actividad 8: desarrollo backend del algoritmo de Hoffman

### **Introducción**

Este es un programa simple que permite analizar archivos de texto utilizando el algoritmo de Huffman, 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. Es una técnica de compresión sin pérdida que asigna códigos de longitud variable a diferentes caracteres, consta de la creación de un árbol binario que tiene cada uno de los símbolos por hoja como visto en clase.

### **Objetivo**

Contar la frecuencia de caracteres y proporcionar funciones básicas de compresión y descompresión utilizando el algoritmo de Huffman. La implementación de esto va a ser con la librería de tkinter, utilizando tres botones y una interfaz gráfica.

### **Requerimientos**

tkinter

heapq

collections

## Funcionalidades

- Examinar: Permite seleccionar un archivo de texto para analizar
- Compresión: Una vez que se ha seleccionado un archivo, lo vamos a comprimir utilizando el algoritmo de compresión de Huffman y el archivo comprimido se guarda como .huf en la misma ubicación que el archivo original.
- Descompresión: Si se selecciona un archivo comprimido con extensión huf lo descomprime utilizando el algoritmo de Huffman de nuevo y guarda el contenido descomprimido en un nuevo archivo sin .huf.

## Desarrollo

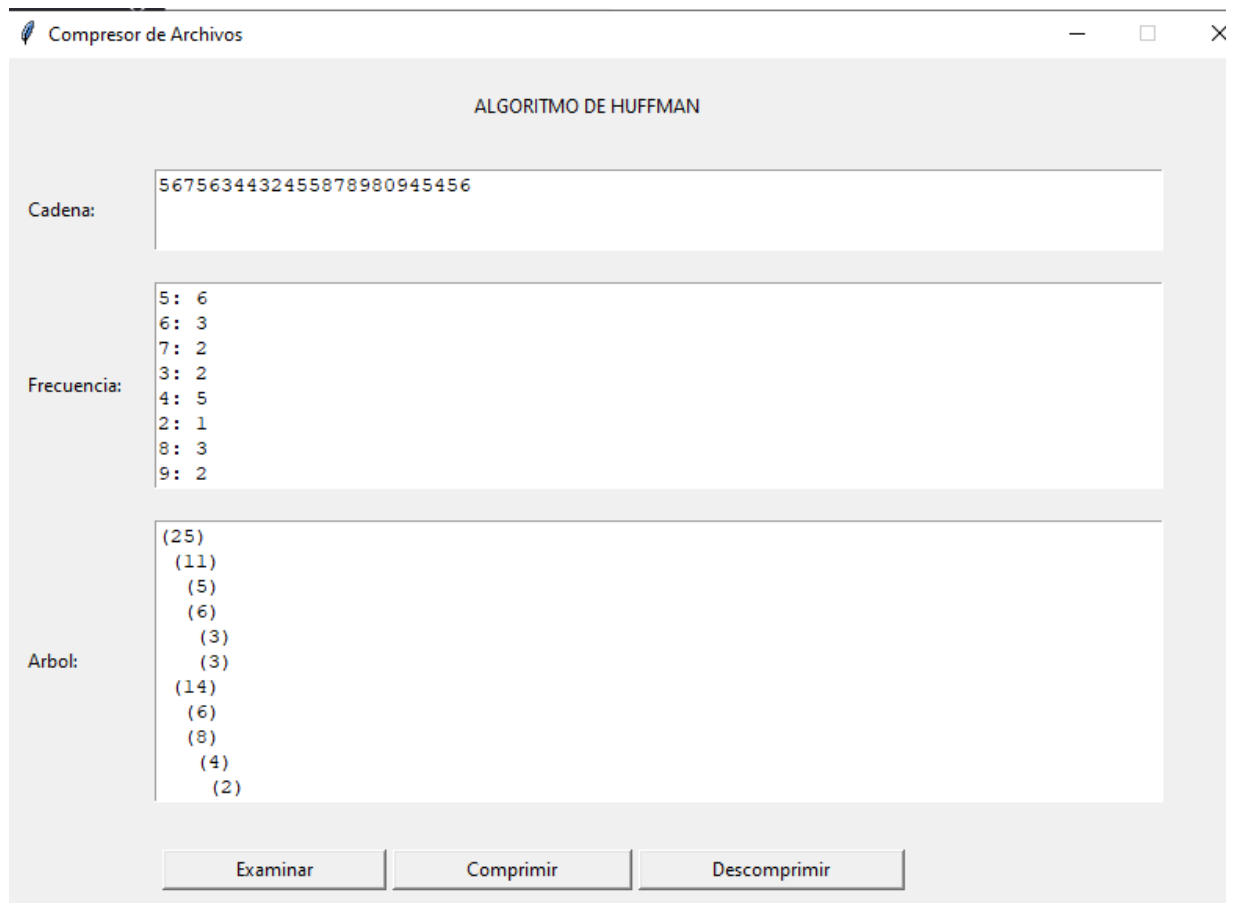
Lo primero que tuve que determinar es cómo realizar los procesos con la ventana que me dio la compañera, en este caso se muestra la cadena original, la frecuencia de cada dígito y esta información plasmada como un árbol de búsqueda binario como vimos en clase. Las dos primeras cosas la verdad fueron fáciles porque solo era usar un mostrar en pantalla y el otro solo fue usar un acumulador, pero para mostrar como árbol si fue un poco difícil hacerlo, en realidad no era requerimiento de la actividad pero en el código del front end se determinaba que era algo a implementar. Esto se logro con la propiedad de que el árbol de Huffman se puede visualizar de manera vertical, donde los nodos izquierdos se encuentran debajo del nodo padre y los nodos derechos se encuentran a la misma altura que el nodo padre pero a la derecha y para hacer esto se uso un widget de texto que llameentradaArbol donde cada nodo se representa mediante un par de paréntesis que contienen la frecuencia del nodo, los nodos se dibujan en el widget de texto utilizando espacios para mantener y simular su posición dentro del árbol.

Para examinar el archivo usamos un método llamado examinar\_archivo llamada por el evento de clic en el botón, despliega un cuadro de diálogo que permite al usuario seleccionar

### Actividad 8: desarrollo backend del algoritmo de Hoffman

un archivo de texto desde su sistema de archivos local, la ruta de ese archivo se guarda en una variable, se verifica si se ha seleccionado un archivo y se habilitan los botones de comprimir y Descomprimir para permitir que el usuario realice estas acciones sobre el archivo seleccionado, se lee el contenido del archivo seleccionado y se muestra en un widget de texto de la interfaz gráfica.

### Resultados



## **Conclusión**

A través de una interfaz gráfica desarrollada con Tkinter en Python, los usuarios pueden seleccionar archivos de texto, comprimirlos y descomprimirlos según sea necesario. La interfaz proporciona una experiencia intuitiva y fácil de usar, permitiendo a los usuarios interactuar con el algoritmo de Huffman sin necesidad de conocer detalles técnicos complejos. Sin embargo, es importante tener en cuenta que el programa actual presenta algunas limitaciones y posibles mejoras, como la gestión adecuada de caracteres especiales y la optimización del rendimiento para archivos de gran tamaño. En general, el programa cumple su propósito de ofrecer una herramienta simple y funcional para la compresión y descompresión de archivos de texto utilizando el algoritmo de Huffman.

## Actividad 8: desarrollo backend del algoritmo de Hoffman