**Informe de Implementación del Compresor y Descompresor Huffman**

**Desarrollo**:

Se implementó un sistema de compresión y descompresión de archivos de texto utilizando el algoritmo de Huffman, el cual es ampliamente reconocido por su eficiencia para compresión sin pérdida. La solución se desarrolló en Java y consta de tres componentes principales:

Construcción del árbol de Huffman: Se analiza el archivo de entrada para contar la frecuencia de cada carácter. A partir de estas frecuencias, se construye un árbol binario utilizando una cola de prioridad (PriorityQueue), de modo que los caracteres con menor frecuencia tengan mayor profundidad y, por tanto, códigos binarios más largos.

Codificación y escritura de bits: Usando el árbol generado, se asignan códigos binarios únicos a cada carácter. Luego, el archivo original se convierte en una secuencia de bits comprimidos y se guarda usando una clase personalizada BitOutputStream, que permite escribir a nivel de bit.

Descompresión y validación: El proceso inverso comienza con la lectura del archivo .hufftree que contiene el árbol serializado, seguido por la lectura del archivo .huff. A medida que se leen los bits, se recorre el árbol hasta encontrar hojas, reconstruyendo así el texto original.

Se generaron y verificaron archivos comprimidos y descomprimidos usando al menos dos textos distintos. En todos los casos, la descompresión devolvió exactamente el contenido original.

**Decisiones de diseño:**

Se empleó serialización de objetos (ObjectOutputStream) para guardar el árbol, permitiendo una reconstrucción precisa durante la descompresión.

Se desarrollaron clases auxiliares (BitInputStream y BitOutputStream) para manejar la lectura y escritura bit a bit, fundamentales para lograr compresión efectiva.

El código fue modularizado para mantener claridad, facilidad de pruebas y futuras extensiones.

**Conclusión:**

El compresor Huffman desarrollado cumple con los objetivos de eficiencia, organización y precisión en la compresión y descompresión de archivos de texto. Se respetaron los principios del algoritmo, aplicando estructuras de datos adecuadas. El sistema fue probado exitosamente con distintos archivos, y puede integrarse fácilmente a flujos de trabajo más complejos. La compresión es efectiva y la fidelidad en la recuperación del texto original fue total.