Informe análisis: *"Desafío 1"*

Estudiantes:

Harlin David Palacios Bermúdez

Sebastian Arango Sánchez

Docente Informática ll:

Aníbal José Guerra Soler

Universidad De Antioquia

Septiembre 16 del 2025

Segundo semestre

A) Análisis del problema y consideraciones para la alternativa de solución propuesta.

Como petición de la empresa “Informa2”, se pide realizar un proyecto que lleve a cabalidad la desencriptación y la descompresión de un mensaje contenido en un respectivo archivo de texto e identificando el método de compresión y los parámetros que llevaron a cabo para esto.

La desencriptación se dará por medio de rotación de bits y la operación XOR con una clave “k”.

La descompresión se dará utilizando dos métodos, los cuales son: LZ78 y RLE. La primera de estas se basa en crear un arreglo dinámico que ubica cada letra nueva, avanza en el archivo y la va guardando en este. El segundo método nos cuenta la cantidad de veces que encontramos una letra de manera consecutiva, guardándola en una lista, con el valor y el carácter correspondiente.

Se nos otorga un fragmento aleatorio del mensaje para poder buscar la coincidencia, después de realizar la desencriptación y descompresión, para poder comparar y verificar la veracidad del contenido obtenido de dicho proceso.

Debemos de considerar lo siguiente al momento de codificar: la restricción de caracteres, no se distingue la clave ni la rotación de bit a la hora de desencriptación ni su descompresión del código; por tanto se realizará un método inverso en un respectivo algoritmo el cual desarrolle los procesos mencionados anteriormente de forma que llegue a la solución adecuada o

mensaje de texto que queremos tener, también se ha de tener en cuenta; la eficiencia y la memoria dinámica mediante la implementación de punteros.

Nuestro objetivo final es entregar al usuario la desencriptación y descompresión de los archivos anteriormente brindados por este como material de trabajo.

B) Esquema donde describa las tareas que usted definió en el desarrollo de los algoritmos.

1- Leer los archivos de entrada: Lo primero que hicimos fue leer el mensaje encriptado y comprimido desde un archivo. También leímos el fragmento conocido del mensaje original desde un archivo de texto. Esto nos dio el insumo material para trabajar.

2- Desencriptar el mensaje: Como el mensaje se encriptó con una rotación de bits a la izquierda y luego una operación XOR con una clave, para desencriptar tuvimos que hacer el proceso inverso: primero aplicamos XOR con la clave y luego rotamos los bits a la derecha. Creamos funciones especiales para manejar estas operaciones a nivel de bits, ya que son fundamentales para recuperar el mensaje comprimido.

3- Descomprimir el mensaje: Sabíamos que el mensaje comprimido podía estar en uno de dos formatos: RLE o LZ78. Así que implementamos ambos métodos para estar seguros:

4- RLE: Este método consiste en números seguidos de letras, ir juntando los dígitos para formar números completos y luego repetir la letra tantas veces como indicara el número.

5- LZ78: Este método usa un diccionario. La cadena desencriptada contiene pares de bytes que representan un índice y un carácter. Fuimos reconstruyendo el diccionario a medida que leíamos estos pares, lo que nos permitió formar el texto original poco a poco.

6- Probar todas las combinaciones: Como no conocíamos los valores de la rotación (n) ni de la clave (K), probamos todas las posibilidades: n de 1 a 7 bits y K de 0 a 255. Para cada combinación, desencriptamos el mensaje y luego intentamos descomprimirlo con ambos métodos. Si el resultado incluía el fragmento conocido y además usábamos toda la cadena de entrada, significaba que habíamos encontrado la solución correcta.

7- Manejar la memoria: Como no podíamos usar los tipos de variable tipo string, utilizamos arreglos de caracteres con memoria dinámica (usando new[] y delete[]). Esto requirió que nos aseguramos de liberar la memoria después de usarla para no gastar recursos innecesarios y evitar problemas de rendimiento.

8- Probar con ejemplos: Antes de usar el mensaje real, probamos todas las funciones con ejemplos pequeños y conocidos. Esto nos ayuda a verificar que cada parte del código funciona bien y a corregir errores antes de enfrentarnos al problema completo.

Este enfoque nos permitió abordar el problema de manera ordenada y asegurarnos de cubrir todos los aspectos necesarios para encontrar la solución.

C) Algoritmos implementados.

Para la elaboración de la solución, los Algoritmos a utilizar serán diversos cuerpos de código que representarán el desarrollo de funciones o el cuerpo principal donde se ejecutará todo el código. Específicamente, cada cuerpo del código cumplirá una labor explícita que será explicada a continuación:

- Main o Cuerpo Principal: En este se aplicará el llamado de las funciones creadas con anterioridad en otro módulo, para llevar a cabo las operaciones inversas de encriptación y compresión. Además en este se llevará a cabo el ingreso y salida de datos hacia el usuario. En el ingreso se recibirá la cantidad y los archivos a pasar por los procesos de solución. Mientras que, en la salida se dará el texto decodificado, junto a los métodos que se usaron para hallar el resultado final.
- Función Descompresión RLE: Primeramente se realizará la descompresión del texto por este método, recibiendo como parámetro arreglos dinámicos, de forma que bajo el uso de memoria dinámica se irá trabajando sobre este, cambiando el valor que codifica en tabla ASCII por el que verdaderamente se desea obtener del texto original sin encriptar, por tanto, me entregará el arreglo con sus datos modificados.
- Función Descompresión LZ78: Realizará la descompresión del texto, en caso de que se haya evidenciado el hecho de que haya sido posible su compresión por módelo LZ78, recibiendo como parámetro arreglos dinámicos, de forma que bajo el uso de memoria dinámica se irá trabajando sobre este, cambiando el valor que codifica en

tabla ASCII por el que verdaderamente se desea obtener del texto original sin encriptar, por tanto, me entregará el arreglo con sus datos modificados.

- Función Desencriptación XOR: Esta empleará el papel de desencriptar el valor de cada dato que se encuentre en el arreglo dinámico, llevando a cabo la operación XOR bit a bit, lo cual me alterará la información del dato, pero esta operación, nos dará como resultado objetivamente la información que contenía dicho dato antes de su encriptación por operación XOR. Al final la función retornará el arreglo dinámico con sus datos variados, listos para realizar la desencriptación por rotación de Bits.
- Función Desencriptación Rotación de Bits: Esta función se ejecutará, luego de la Función de Desencriptación por Operación XOR, en la cual se dará la rotación de bits de cada dato char en dirección opuesta (Derecha) a la que se dio la encriptación (Izquierda), por tanto, después de llevar a cabo esta operación tendremos los datos del texto en su representación comprimida. A su vez con esta función hallaremos el valor con el cual se llevó a cabo la rotación de bits del dato original. Siendo así, que esta función nos retornará el número que representa la cantidad de bits que se rotaron y el arreglo dinámico con el texto comprimido.
- Función Lectura del Archivo: se realizará una función la cual nos permite la entrada de la cantidad de archivos a decodificar, por ende antes de esta se le pedirá al usuario que nos solicite la cantidad de archivos que va a ingresar. Una vez ingresada la cantidad de textos se aplicará la separación de estos. Luego, nos vamos al final del

archivo para ver cuánto mide, así sabemos cuánta memoria necesitamos. Con ese tamaño, creamos un arreglo, donde guardaremos toda la información. Después, leemos todo el contenido del archivo de una vez y lo metemos en nuestro arreglo. Finalmente, cerramos el archivo.

- Función Comparación Pista-Resultado: Con esta lograremos determinar si con respecto al resultado de nuestra operación de rotación, hemos encontrado una relación entre la pista y nuestro prospecto de solución, determinando si debemos seguir realizando la descompresion por otro método o si simplemente ya hemos terminado, de forma tal que describirá el fin de nuestro modelo de decodificación del texto encriptado. La función retornará valores Booleanos.

D) Problemas de desarrollo que afrontó.

Al momento de iniciar el desarrollo nos topamos con varios problemas, los cuales no nos permitían continuar con el respectivo avance del código, entonces después de analizarlos y entenderlos pudimos llegar a sus soluciones. Los problemas que nos afectaron fueron los siguientes:

1- Fuga de memoria: a veces al momento de hacer un new[], se nos olvidaba complementarlo con un delete[], entonces teníamos una parte de memoria sin utilizar, lo cual nos provocaba que hubiera menos eficiencia.

2- Lectura de los archivos: algunos de los archivos fueron codificados y comprimidos en un sistema operativo diferente al de windows, lo que nos generó una confusión al principio, aparte de eso la manera de cargarlos todos conjuntos.

3- Falsas soluciones: al momento de realizar la desencriptación y descompresión, y probarlo con la pista pudimos notar que existe la posibilidad de que el fragmento coincide hasta cierto punto del texto, pero que después de eso ya no sea legible, lo que provoca que ese camino no sea el correcto.

4- Buscar: en este caso entran n y k, porque pueden tomar cualquier valor , entonces encontrarlo nos llevaría bastante tiempo, así que toca optimizar el código.

5- Diccionarios: este caso específico se da para LZ78 el cual nos dice el uso de diccionarios, y C++ no cuenta con muchos comandos para el uso de estos, entonces toca ingeniárselas para realizarlo de una manera satisfactoria.

6- El uso de Git: debido a nuestra poca experiencia en git, al momento de utilizar nos confundiamos o lo realizabamos de la manera que no era la indicada, provocando que no pudiéramos entender mucho de lo que estábamos haciendo o si lo estábamos llevando correctamente.

7- Ubicación desconocida: no sabemos la ubicación exacta de la pista por ende, toca hacer un método que nos encuentre la ubicación que necesitamos y que coincida.

E) Evolución de la solución y consideraciones para tener en cuenta en la implementación.

Para llegar a la solución se siguió el proceso de desarrollo establecido, el cual se dividió en las siguientes fases:

1- El problema y su contextualización: para empezar el desarrollo del programa se llevó a cabo su respectiva contextualización. En la cual comprendimos la necesidad del usuario a satisfacer con respecto a la problemática que lo aquejaba, correspondiente a la decodificación de un mensaje que se encuentra comprimido y encriptado con anterioridad, contando que desconocemos la clave con que se encripto y con el modelo en el cual este fue comprimido.

2- El análisis: se realizará, tras lo entendido sobre el problema un programa el cual sea eficiente y eficaz, cumpliendo los requisitos del usuario. Se nos entregará uno o varios textos encriptados y comprimidos. Lo primero a notar es una rotación de bits hacia la izquierda, además, cada uno cuenta con una distintiva clave. El segundo, son dos posibles métodos en el cual fue comprimido el texto entregado LZ78 y RLE, existe la posibilidad de que ninguno de estos dos procedimiento sea con el cual fue comprimido. Aparte de todo lo anterior contaremos con una pista la cual nos sirve para saber si vamos por el camino correcto, y poder dar con el

texto exacto. Al finalizar todos los procedimientos anteriores se le entregará al usuario todos los textos recibidos, ya desencriptados y descomprimidos.

3- Diseño: se garantizará el uso de memoria dinámica y estática, uso de variables que estén establecidas en los ámbitos de las memorias anteriormente dichas, implementando funciones las cuales en un principio estarán enfocadas en la desencriptación y descompresión del archivo y en mayor proporción hallar las técnicas con las que se dio el proceso inverso de lo precitado.

4- Consideraciones: Luego del entendimiento de todos los requisitos evidenciados a desarrollar en el proyecto, tenemos diferentes aspectos a tratar con suficiente minuciosidad, desde temas de abstención con diversos mecanismos, librerías o inclusive estrategias que acorten demasiados los procesos, por tanto tenemos que desarrollar funciones propias que realicen estos procesos de lectura e implementación de estructuras; hasta de conductas del manejo de los datos, entre los que podemos encontrar, los caracteres a identificar, claves y valor de operaciones para descifrar, y propiedades inherentes al uso de los punteros.

5- Implementación: se inicia con el proceso de programación, nuestra implementación se realizará por medio de que ambos miembros del equipo realizarán cierta parte del código de una manera discontinua. Dicho de una mejor manera harán partes del código por separado pero siguiendo la misma estructura, por ejemplo: Harlin

realizará hasta cierto punto la desencriptación y Sebastian la termina de complementar, así mismo con los métodos de descompresión, esto nos permitirá trabajar con la misma estructura y no confundirnos al momento de juntar todas las partes del programa.

6- Pruebas:llevado de la mano con la implementación, (aunque en la realización del código se harán pruebas para ver que todo funcione correctamente) se probarán textos ejemplos para ver el correcto funcionamiento del programa, y que no haya ningún error que pueda llevar al mal funcionamiento de este.

7- Conclusión: ya con el programa debidamente hecho, se le hará entrega al usuario final, el cual quedará satisfecho con lo solicitado. De igual manera se realizarán pruebas junto a él para que pueda ver el correcto funcionamiento del sistema. Después de eso, llegaremos a un acuerdo para la ***remuneración***.