Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

ANÁLISIS COMPARATIVO ENTRE FORMATOS DE ARCHIVOS DE COMPRESIÓN (GZ, BZ2, ZIP Y LZH)

Ludin Estuardo De León González

Asesorado por: Inga. Elizabeth Domínguez Alvarado

Guatemala, septiembre de 2006

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ANÁLISIS COMPARATIVO ENTRE FORMATOS DE ARCHIVOS DE COMPRESIÓN (GZ, BZ2, ZIP Y LZH)

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

LUDIN ESTUARDO DE LEÓN GONZÁLEZ

ASESORADO POR: INGA. ELIZABETH DOMÍNGUEZ ALVARADO

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, SEPTIEMBRE DE 2006

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soría
VOCAL II	Ing. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Inga. Virginia Victoria Tala Ayerdy
EXAMINADOR	Ing. Pedro David Tzoc Tzoc
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIO	Ing. Carlos Humberto Pérez Rodríguez

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ANÁLISIS COMPARATIVO ENTRE FORMATOS DE ARCHIVOS DE COMPRESIÓN (GZ, BZ2, ZIP Y LZH),

tema que me fuera asignado por la dirección de la escuela de Ingeniería en Ciencias y Sistemas, en febrero de 2004.

Ludin Estuardo De León González

DEDICATORIA A

DIOS	Mi Padre Celestial, Señor y Creador a quién dedico especialmente este logro, pues sin Él nada de esto fuera posible.
MIS PADRES	Manuel De Jesús De León Muñoz y María Virginia González Ramos, a quienes dedico este logro, porque gracias a sus incansables desvelos, esfuerzos y sabios consejos he logrado este triunfo.
MIS HERMANOS	Manuel Lisandro y Jordi Sullivan, por todo su amor y apoyo en todo lo que he realizado hasta el día de hoy.
MIS CATEDRÁTICOS	Por sus enseñanzas, dedicación y esmero.
MI ASESORA	Inga. Elizabeth Domínguez, por su tiempo, paciencia y orientación en la elaboración de este trabajo de graduación.
MIS COMPAÑEROS DE EXAMEN PRIVADO	Edgar, Neftalí Mario y Willian, porque juntos pudimos pasar esa prueba tan difícil y en quienes encontré un gran apoyo para poder llegar a la meta.

MIS AMIGOS

En especial a Fernando Omar Sequen, Manuel Alberto Girón, Pedro Pablo Hernández y Sergio Alfredo Calvillo. Por todos y cada uno de los momentos imborrables que convivimos.

MIS AMIGOS EN GENERAL

Gracias por cada una de sus lágrimas, por cada una de sus sonrisas y por cada una de sus ayudas brindadas.

AGRADECIMIENTO A

MI SEÑOR JESÚS, mi Salvador, mi Señor y mi Guía, quién en todo momento ha estado conmigo y me ha colmado de bendiciones, muestra de ello, este triunfo. Gracias Señor.

MI PATRIA GUATEMALA, país que me vio nacer y al cual ahora le contribuyo con un grano de arena para sacarlo adelante.

LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, por ser mi casa de estudios y brindarme la oportunidad de lograr este sueño.

LA FACULTAD DE INGENIERIA, por brindarme todos los medios y facilidades para poder optar a este título.

TODAS LAS PERSONAS, que de una u otra forma Dios puso en mi camino y contribuyeron para que pudiera llegar a la meta, gracias.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

Algoritmo Secuencia de pasos escritas en pseudo código que resuelven un problema determinado.

Árbol binario Estructura formada por nodos, los cuales todos ellos contienen dos enlaces (ninguno, uno o ambos). Por lo general está formado de un nodo raíz y dos nodos hijos.

Archivo *ASCII* Estos son los denominados archivos de texto. La abreviación *ASCII* significa *American Standard Code for Information Interchange*, un código de 7 bits que sustituye las letras del alfabeto romano por cifras y otros caracteres informáticos.

Archivo binario Un archivo binario, contrariamente a un fichero *ASCII*, contiene más que simplemente texto. Puede contener fotos, sonido, hojas de cálculo, o documentos concebidos para el procesamiento de texto. Los ficheros binarios están formados de unos y ceros.

Archivo *BMP* Archivo binario cuya información son imágenes. Es popular en el sistema operativo *Windows*. En comparación a otros formatos gráficos, su tamaño es considerablemente grande.

Archivo comprimido Archivo al cual se le ha aplicado el proceso de compresión, por medio de un compresor determinado y que generalmente es reducido en tamaño en comparación a su tamaño original.

Archivo *JPG* Archivo binario cuya información son imágenes. Es popular en el Internet debido a su corto tamaño. Podemos decir que es una mejora de formatos como el *BMP*.

Archivo *MP3* Archivo binario cuya información es sonido. Es popular en el Internet debido a su corto tamaño. Es sin duda una mejora de formatos tales como los archivos *WAV*.

Archivo WAV Archivo binario cuya información es sonido. Por la forma de su compresión, es un formato con un tamaño considerablemente grande en comparación a los formatos de audio actuales. Su tamaño puede ser reducido, pero esto afecta la calidad a obtener.

Archivo WMA Archivo binario cuya información es sonido. Popular en el sistema operativo *Windows*. Son archivos comprimidos de sonido muy similares en tamaño a los archivos de formato *MP3*.

Bajar archivos Término mejor conocido como *download*. Este proceso consiste en transferir un archivo de un computador remoto hacia el computador que estamos operando.

Buffer Dispositivo de almacenamiento temporal, regularmente de acceso rápido y formado de varias posiciones correlativas.

Bit Unidad de medida más pequeña existente en cuanto a mediciones de tamaños de información. Está formada por un uno (1) o un cero (0).

Byte	Grupo de 8 bits (unos y ceros) que sirve como unidad de medida cuando se habla de la capacidad de almacenamiento de un dispositivo o en su defecto de la información que almacenará, en este caso los archivos.
Clic	Acción de pulsar una sola vez un botón del ratón.
Codificación	Proceso de proteger un grupo de información aplicando una serie de pasos algorítmicos.
Compresión	Es una manera de reducir la talla de un archivo para que no ocupe demasiado espacio en un servidor o en un disco duro y que pueda viajar más rápidamente por la red. La compresión se realiza por medio de un software que utiliza ecuaciones matemáticas (algoritmos). Se necesita otro software para descomprimir los datos.
Compresor	Herramienta o comando utilizado, que esta construido de tal forma que aplica un algoritmo para poder realizar el proceso de compresión.
Compresión reversible	Proceso de poder regresar al estado original del archivo comprimido esto es: poder realizar la descompresión.
Compresión irreversible	Proceso que al ser realizado no existe forma alguna de poder regresar al estado original del archivo.
Disco duro	Dispositivo fijo de almacenamiento de información. Su unidad de medida es el *byte*.

Empaquetar Proceso de agrupar un conjunto de archivos en un solo archivo.

Extensión archivo Secuencia de caracteres que identifica el tipo de un archivo. Comúnmente está formado de tres caracteres, aunque pueden ser más. Por ejemplo, si un archivo fuera de texto, lo más seguro es que su extensión fuera *TXT*.

Modo gráfico Mejor conocido como el modo protegido en el sistema Windows. Este modo se opera desde una interfase gráfica, que hace más amigable el sistema para los usuarios.

Modo real Mejor conocido como el modo *DOS* en el sistema *Windows*. O modo símbolo del sistema, en donde se trabaja con base a la ejecución de comandos.

Multiplataforma Término que define que una aplicación es soportada, sin realizarle variaciones, por más de un sistema operativo.

Parámetros Serie de términos agregados, en este caso en específico, a un comando para realizar tareas extras.

Pseudo código Notación de un algoritmo a un nivel muy general, que después puede ser implementado o aplicado a un lenguaje determinado de programación en su propia sintaxis.

Ratón Dispositivo de entrada utilizado por una computadora para realizar eventos diversos.

Unidad Central de Procesos	Unidad encargada de realizar una serie de micro-operaciones de escritura y lectura, así como control de flujos de los datos al más bajo nivel.
Shell	Modo originalmente utilizado por el sistema *UNIX*, adoptado por *Linux*. También llamado “modo consola” (en apariencia similar al modo *DOS* en *Windows*) en donde todo es trabajado con base a la ejecución de comandos.
Sistema operativo	Conjunto de programas que administran los dispositivos de entradas y salidas, procesos e interfaces de un computador. Entre los más utilizados se encuentran *Windows*, *UNIX*, *Linux*, entre otros.
Tamaño real	Tamaño calculado con base en los *bytes*. Se dice real, debido a que este es verdaderamente el tamaño ocupado por un archivo en un dispositivo de almacenamiento.
Tamaño resumido	Tamaño que la mayoría de los sistemas operativos calculan y presentan con base al tamaño original y es menor a éste, omitiendo cierto nivel de exactitud. Entiéndase que hace el equivalente 1*Kb* = 1000 *bytes*, cuando en realidad son 1024 *bytes*.

RESUMEN

La compresión de archivos sin lugar a duda es uno de los temas más importantes cuando tratamos de realizar un eficiente manejo de la información, lo cual se reduce a un sólo término: El espacio. Sin duda en la actualidad, con la incursión del Internet, el envío, carga y descarga de los archivos requiere para el usuario se haga en el menor tiempo posible y con la menor cantidad de espacio a ocupar.

Hoy en día, existen diferentes tipos de formatos de compresión (unos comerciales otros no), incluso para diferentes tipos de sistemas operativos, y en algunos casos solo se conocen algunos muy tradicionales o populares. En este trabajo de investigación, se trata de incluir cuatro formatos, no muy conocidos para algunos, pero que al igual que los formatos mencionados anteriormente, poseen la capacidad de realizar una buena compresión.

Por lo tanto, este trabajo de investigación busca servir como una ayuda para poder observar, tanto las ventajas, como desventajas de estos formatos de compresión, así como la eficiencia de cada uno para hacer ésta tarea. También observar las mejoras que existen entre los diferentes formatos y también en relación a sus versiones anteriores.

Y finalmente, tener un punto de referencia y apoyo al realizar pruebas reales con diferentes tipos de archivos y observar el resultado que arrojará el análisis realizado.

OBJETIVOS

- **General**

  Elaborar un documento que haga un análisis comparativo de los diferentes formatos de compresión y que sirva como una guía cuando se necesite hacer la elección de uno de ellos y realizar la compresión de archivos.

- **Específicos**

  1. Presentar los diferentes formatos de compresión, para que éstos sean conocidos y puedan ser aplicados.
  2. Realizar un análisis comparativo entre los formatos de compresión, y, así observar sus diferentes ventajas y desventajas propias de cada uno en relación con los demás.
  3. Elegir el correcto formato de compresión, según sea el grupo de archivos a comprimir.

INTRODUCCIÓN

Los formatos de compresión a estudiar en este trabajo de investigación no serán a lo mejor muy comerciales o conocidos, pero por su misma situación se presentan para poder conocer un poco más acerca de ellos, en relación a su manejo, ventajas y desventajas entre sus versiones anteriores, y ventajas y desventajas entre los distintos formatos en estudio.

Además, se hace un estudio acerca de los algoritmos de compresión que estos formatos utilizan y de las mejoras que se han realizado entre versiones actuales y versiones anteriores. Pues a pesar de que existe cierta compatibilidad entre los distintos formatos, se puede observar que se ha hecho una mezcla de diferentes algoritmos o simplemente se ha mejorado el algoritmo original.

Finalmente, se hace un estudio con base en un conjunto real de archivos y haciendo comparaciones tanto en tiempo como en espacio, éste último es el aspecto quizás más importante a buscar a la hora de comprimir archivos. Aunque en la actualidad lo que se busca es un compresor ideal (reducir la mayor cantidad de espacio en el menor tiempo), realmente no se ha logrado conseguir, pero lo que si se ha logrado es ir mejorando la tecnología existente y así realizar de la manera más eficiente este proceso, la compresión.

1. FORMATOS DE ARCHIVOS DE COMPRESIÓN

En la actualidad existen diferentes tipos de formatos de archivo de compresión, algunos muy conocidos o comerciales y otros, como algunos de los incluidos en este estudio, no tanto. Entre los formatos de compresión más conocidos o más manejados por su popularidad, facilidad de manejo o comercialización se pueden mencionar archivos extensión: ZIP, Z, RAR, CAB, ARJ, ZOO, etc. La utilización de los formatos de compresión mencionados anteriormente se puede deber a varias causas, posiblemente por la utilización de un solo sistema operativo, por costumbre de utilización, u otras razones; la razón que sea no importa, pues independientemente de esto, se darán a conocer algunos formatos de archivos de compresión no muy conocidos pero que de igual forma tienen una gran capacidad y eficiencia para poder comprimir archivos.

El estudio se realizará con base en cuatro formatos de archivos de compresión, los cuales son de extensión: ZIP, GZ, BZ2 y LZH. La utilización de estos últimos tres es más común en sistemas operativos que los usuarios desconocen o no están acostumbrados a utilizar, pero siempre es importante abrirse a nuevas alternativas y sobre todo en algo tan importante como el tema de la compresión de archivos.

Antes de realizar el análisis principal, existen términos básicos que son necesarios conocer para poder entender las características, funcionamiento y resultados que envuelven a los formatos de archivos de compresión.

1.1 ¿Qué es la compresión de archivos?

La idea principal de la compresión de archivos se basa en el espacio: un archivo comprimido ocupa menos espacio que uno solamente archivado. La compresión de archivos consiste en tomar uno o más archivos y convertirlos en un único archivo el cual ocupará menos espacio que la suma original de los tamaños de todos ellos.

Una analogía, para poder entender mejor el concepto de compresión, es un mapa de bolsillo. Cuando el mapa se encuentra extendido, su tamaño es enorme y sería imposible guardarlo en nuestro bolsillo; por lo tanto se procede a hacerle una serie de dobleces que hacen que su tamaño original disminuya y así de esta forma pueda ingresar a nuestro bolsillo, es decir se hace mas compacto, pues sucede lo mismo cuando se habla de la compresión de archivos, solo que este proceso se hace a través de algoritmos de compresión pero la tarea que realizan es básicamente similar. Pues el objetivo es dejar el mayor espacio libre posible en el disco duro comprimiendo un conjunto de archivos en uno solo y así poder almacenar más información.

Además con la incursión del Internet, la compresión de archivos es un tema tan común pues ayuda a los usuarios a transferir sus archivos de forma más rápida, en un espacio reducido y por lo tanto en una forma más eficiente. Otra ventaja que nos da la compresión es la multiplicidad de envío, pues podemos enviar muchos archivos simultáneamente en uno solo, en lugar de enviar uno por uno, pues esto sería una tarea muy lenta y tediosa.

Por otra parte, lo más común es comprimir aquellos archivos que no van a ser utilizados en un tiempo considerable como por ejemplo una copia de seguridad de información importante, archivos que se desean conservar pero que no se utilizarán en un tiempo considerable o simplemente el hecho de querer comprimirlo para liberar más espacio en el disco duro.

1.1.1 Compresión reversible

La compresión reversible se refiere al proceso de poder regresar al estado original un archivo comprimido, esto es: poder realizar la descompresión. Poder realizar este proceso es muy importante cuando se trata de comprimir un grupo de archivos que poseen información que pueda ser utilizada en un futuro, pues de lo contrario la información que contienen estos archivos se perdería, debido a que no habría manera alguna de ser recuperada.

1.1.2 Compresión irreversible

La compresión irreversible entonces a diferencia de la reversible es que no podemos regresar la información después de ser codificada a su estado original. Este tipo de compresión es muy utilizada en el manejo de archivos de video, música, etc., y lo que se logra es un grado de distorsión que en el caso, por ejemplo de una imagen, se logra al reducir su calida o tamaño.

Para entender mejor este concepto existe un ejemplo muy simple y fácil de entender: la diferencia entre un archivo gráfico BMP y un JPG es que el archivo JPG es un archivo comprimido y que en algunos casos es irreversible su conversión al formato original. En ocasiones se puede regresar a su estado original, pero a través de herramientas especiales encargadas de realizar este proceso y aún si se logra realizar con éxito el proceso, puede perderse cierto grado de integridad de la información contenida en el archivo en relación a su estado original.

1.2 ¿Qué es un archivo binario?

Un archivo binario a diferencia de un archivo ASCII, además de contener simplemente texto puede contener sonido, video, etc. Es decir, con este tipo de archivo se posee una gran ventaja en relación a su contenido pues debido a su formato nos da la pauta para almacenar una gran diversidad de tipo de información, ya que como su nombre lo indica está formado simplemente de unos y ceros.

La mayor parte de los formatos de archivos de compresión adoptan este tipo de formato por las ventajas mencionadas anteriormente, pues es imposible guardar sonido, video, etc. en un archivo de texto como el formato ASCII.

1.3 Formato GZ

Este formato de archivo de compresión es más popular en sistemas operativos como UNIX o *Linux.* Reduce el tamaño de un archivo por medio del algoritmo denominado *Lempel-Ziv* o también conocido como LZ77 que será analizado más adelante.

La desventaja principal de este tipo de formato consiste en que únicamente se puede comprimir un solo archivo simultáneamente. Es decir, si se deseara comprimir un conjunto de archivos contenidos en una carpeta, no se puede realizar de forma directa. Para poder realizar dicha tarea se tendrían que utilizar comandos que realizan el proceso de empaquetar varios archivos en uno solo, por mencionar algunos de estos comandos tenemos: “tar”, “cpio”, etc.

Aunque existen herramientas que realizan la compresión y descompresión de este tipo de archivos, la realización de un archivo GZ por lo general se hace con el comando “gzip” en su forma más simple. Una forma sencilla de realizar una compresión y desde el *shell* o modo consola como también se le conoce, sería la siguiente:

```
# gzip archivo.tar
```

Como se verá en capítulos posteriores, existe una serie de parámetros adicionales que pueden ser utilizados con este comando y que nos proporcionan ayuda en el proceso de compresión de los archivos.

1.4 Formato BZ2

Este formato de archivo de compresión se obtiene por medio de la utilización de varios algoritmos de compresión y codificación, los cuales son:

- *Burrows Wheeler*
- La codificación de *Huffman*

Estos algoritmos serán explicados en un capítulo posterior. El hecho es que debido a la utilización de estos algoritmos de compresión y codificación el formato *BZ2* adquiere una mejor compresión que la obtenida por los formatos que utilizan el algoritmo *Lempel-Ziv*, como es el caso del formato *GZ* que se mencionó anteriormente.

Aunque el análisis de este archivo se hará en un capítulo posterior, es importante mencionar que la desventaja principal de este formato de compresión es que requiere una mayor cantidad de trabajo de la *CPU* (Unidad Central de Proceso) y por lo tanto se requerirá más tiempo para poder comprimir los archivos, pero el resultado obtenido será mejor si se considera el aspecto espacio, que es uno de los más importantes cuando se trata de la compresión de información.

El comando básico para utilizar este formato de compresión se denomina “bzip2”. Y de igual forma que el formato de compresión *GZ* requiere de la ayuda de la utilidad “tar” para la compresión de más de un archivo, en otras palabras “tar” los empaqueta y “bzip2” los comprime.

1.5 Formato ZIP

Este formato de archivo de compresión es el más comúnmente utilizado en el sistema operativo *Windows*. Para lograr su compresión se utiliza el comando “zip” y a su vez para su descompresión se utiliza el comando unzip. El algoritmo de compresión utilizado para lograr crear un archivo con este formato es uno del tipo *Lempel-Ziv*.

Este formato de compresión es quizás el más utilizado debido a la popularidad dada por *Windows*. Pues desde un inicio cuando únicamente se utilizaba el sistema operativo DOS comercialmente, este formato se comenzó a hacer popular. El comando utilizado en ese entonces para poder comprimir era el “pkzip” y su correspondiente “pkunzip” para descomprimir.

De forma efímera se puede decir que en comparación a los formatos *BZ2* y *GZ* este formato posee una ventaja directa, esta es: comprimir más de un archivo a la vez.

Aunque existen muchas herramientas que soportan este tipo de formato, también puede ser manipulado desde el modo consola como originalmente se trabajaba cuando se utilizaba el comando “pkzip”. En el próximo capítulo, cuando se haga el análisis con base a la facilidad de aplicación se podrá observar como se realiza una compresión de este tipo de formato desde el modo consola.

1.6 Formato LZH

El formato de compresión *LZH* es un formato que para poder ser creado utiliza el comando de compresión “lha”. La combinación de algoritmos que utiliza para lograr este formato de compresión pertenecen a las familias de algoritmos *Lempel-Ziv* y *Huffman*.

Entre sus principales ventajas están la creación de archivos con una composición ligeramente más firme por la estructura que logran dichos algoritmos y por otra parte la distribución del *LHA* que se hace de forma libre.

De igual forma que el formato *ZIP* y a diferencia de los otros dos formatos *GZ* y *BZ2*, este formato permite la compresión de más de un archivo a la vez sin la necesidad de un utilitario extra, pues como se mencionó en el capítulo anterior era necesario utilizar el comando “tar” para empaquetarlos.

Es importante mencionar que los cuatro formatos de archivos de compresión que son el motivo de este estudio, permiten un manejo de uno o más archivos a la hora de comprimir ya sea con ayuda o sin ella de una utilidad de empaquetamiento y que cuando se menciona que solamente permite comprimir un archivo a la vez es a la hora de hacerlo desde el modo consola en el caso de *UNIX* o *Linux* y si fuera compatible con el sistema operativo *Windows* haríamos referencia al modo real o como bien se le conoce el *DOS* o modo consola.

Pero existen herramientas que se presentarán en capítulos posteriores y que a lo mejor poseen la ventaja de manejar más de un archivo simultáneamente, pues son herramientas gráficas que permiten el manejo de varios formatos de compresión en una sola aplicación.

1.7 La utilidad TAR

Es una utilidad que permite agrupar varios archivos en uno único. Es decir lo que comúnmente se conoce como empaquetar un grupo de archivos.

Para fines de este estudio se utilizará este empaquetador, ya que su nivel de facilidad de uso es alto, además provee muchos parámetros para poder realizar varias tareas, y como se verá más adelante cuando se realice el análisis comparativo y se realicen compresiones reales, se podrá conocer la sintaxis y utilización de dicha utilidad.

Por ahora lo importante es hacer énfasis en que dicha utilidad reúne a un grupo de archivos en uno solo sin comprimirlos y que el siguiente paso a esto sería utilizar el comando “gzip” ó “bzip2” para poder crear un archivo de compresión de extensión *GZ* ó *BZ2* respectivamente, el cual entonces ya se puede reconocer como un archivo empaquetado y comprimido.

Si no se utiliza esta utilidad la extensión del archivo sería simple, es decir quedaría de la forma *GZ* o *BZ2*; mientras que si se utiliza se agregaría a la extensión la expresión TAR, por lo tanto el archivo quedaría en su totalidad con la extensión *TAR.GZ* ó *TAR.BZ2*.

1.8 Otros formatos de compresión

En la actualidad existen formatos que presentan ventajas en el proceso de la compresión de archivos. Entre ellos se encuentra el formato RK. Este formato aplica un grado de compresión mayor en comparación con los formatos presentados en este trabajo de investigación.

Adicionalmente ofrece mejores medidas de seguridad, como la encriptación y además, soporta mucho más que algunos de los formatos mencionados en este trabajo. Existen varias herramientas que manipulan este tipo de archivo, la más conocida, *winrk*. Una ventaja sorprendente de este tipo de archivo es que se puede trabajar con archivos de cualquier tamaño, incluso generar copias de seguridad de unidades de disco duro completas. Su principal desventaja, es el tiempo que toma para generar las compresiones en comparación con otros formatos de compresión.

Otro formato que en la actualidad esta revolucionando el proceso de compresión es el formato *UDA*. Ofrece un alto grado de compresión en relación a sus competidores. Pero, de igual forma que el formato *RK*, si lo que buscamos es un tiempo rápido de compresión, sin duda alguna este tipo de formato no es la mejor elección.

Aunque en comparación con el formato *RK* su tiempo de compresión es menor, no deja de ser muy lento.

Sin duda alguna que en sus versiones próximas se buscará encontrar un punto intermedio, esto es, un tiempo menor en la compresión y reducir el tamaño de los archivos lo mayormente posible; o la incursión de nuevos formatos que busquen dar una solución eficaz y lo más eficientemente posible.

Para la realización de este estudio el deseo era incluir a estos dos tipos de archivos u otros, para poder compararlos contra los formatos incluidos en este trabajo de investigación, pero no se encontró herramienta disponible, en este caso paquetes o herramientas en modo *shell* para *Linux* y así poder dar igualdad de condiciones en el proceso de compresión.

2. ANÁLISIS COMPARATIVO CON BASE EN LA EFICIENCIA Y MANEJO DE LOS FORMATOS DE COMPRESIÓN

2.1 Ventajas y desventajas

Todos los formatos en estudio en este trabajo de investigación poseen, a la hora de ser comparados con otros formatos de compresión, un grupo de ventajas y desventajas que por lo tanto dependiendo de la tarea o resultado requerido así será la selección del formato a utilizar.

Las comparaciones que se pueden realizar van desde el tamaño y tiempo de compresión hasta cosas simples como la facilidad de aplicación de un comando ó la utilización de una herramienta de compresión. Estos últimos aspectos parecieran ser de poca importancia, pero cuando se trata de compresión de archivos dejan de serlo, pues influirán en el resultado final deseado.

Pueden existir casos como querer comprimir un grupo de archivos en un determinado formato y no tener las herramientas adecuadas, no poseer el sistema operativo que soporta dicho formato ó simplemente no saber cómo generar el archivo de compresión.

Por ello, ahora se realizará un análisis de las principales ventajas y desventajas desde varias perspectivas, tales como la facilidad de aplicación, la multiplataforma, la disponibilidad de herramientas para su compresión, la cantidad de archivos permitida de comprimir y la utilización de parámetros en los comandos de compresión.

2.1.1 Facilidad de aplicación

En esta sección se analizará uno a uno los formatos de compresión en estudio con base en la facilidad de aplicación, esto es: cuantos pasos se deben realizar para llegar a la compresión y la sintaxis de los comandos para obtener el formato de compresión deseado. Se debe hacer hincapié una vez más en un aspecto muy importante, el análisis a realizar en esta sección será con base en comandos propios de los sistemas operativos que soportan dichos formatos, pues como se mencionó anteriormente, por ejemplo, en *UNIX* o *Linux* existe el "modo consola" o "*shell*" lo que vendría siendo más o menos un equivalente al "modo *DOS*" o "modo real" en *Windows*.

Como se sabe el formato GZ necesita de la utilidad "tar" para poder comprimir más de un archivo; pero en el caso contrario, si es un solo archivo el que se va a comprimir la sintaxis es la siguiente:

```
# gzip archivo.mpg
```

Luego de esto entonces se sustituirá el archivo original *ARCHIVO.MPG* por uno nuevo denominado ahora *ARCHIVO.MPG.GZ*.

Para descomprimir un archivo se debe utilizar el comando “gunzip”, de la siguiente forma:

gunzip archivo.mpg.gz

Regresando así a su estado original el archivo *MPG*. Por otra parte, si el formato que se quiere obtener es un *BZ2* la secuencia es similar, solo que ahora se deberá utilizar el comando “bzip2” y “bunzip2” para comprimir y descomprimir respectivamente un archivo. Así, para comprimir se haría de la siguiente forma:

bzip2 archivo.mpg

Generando el archivo compreso *ARCHIVO.MPG.BZ2* sustituyendo al archivo original. Y para descomprimir entonces ser haría de la siguiente forma:

bunzip2 archivo.mpg.bz2

Regresando el archivo a su estado original, es decir como *MPG*. Hasta ahora se ha comprimido un solo archivo a la vez. Para poder comprimir más de un archivo entonces es necesaria la ayuda del comando “tar” para empaquetarlos y después comprimirlos, ya sea con “bzip2” o “gzip”. La secuencia entonces para empaquetar un grupo de archivos sería la siguiente:

tar –cvf archivo.tar directorio/fichero

El parámetro “c” indica que se va a empaquetar, “v” se utiliza para mostrar lo que está realizando y “f” indica que lo que sigue es el nombre del archivo a empaquetar. Seguidamente el nombre del archivo extensión *TAR* que se va a generar y por último el directorio donde están los archivos a empaquetar.

Una vez generado el archivo *TAR* que contiene el conjunto de archivos que se desea comprimir, se puede aplicar cualquiera de los comandos de compresión según sea el formato a generar, es decir el "bzip2" o "gzip". Independientemente del comando que se utilice en esta parte del proceso ya se puede decir que el conjunto de archivos se encuentran comprimidos.

Para invertir el proceso se puede utilizar el comando "gunzip" o "bunzip2", según sea el formato generado. Al aplicar cualquiera de estos comandos se regresará el archivo a su origen, es decir un archivo *TAR*. Y para finalmente desempaquetar todos los archivos únicamente cambiaremos el parámetro "c" por el parámetro "x" que indica que se va a proceder a desempaquetar:

```
# tar –xvf archivo.tar
```

Existen varias formas de hacer todos estos procesos, pero se eligió hacerlo de esta forma para poder explicar paso a paso la secuencia de cómo empaquetar y finalmente cómo comprimir y viceversa.

Ahora se explicará como generar archivos de compresión *ZIP*, de la forma más sencilla, y con base en el sistema operativo *Linux*, se realizará al igual que los dos formatos anteriores desde el *shell*.

Para la compresión se utilizará el comando "zip" y para su respectiva descompresión se utilizará el comando "unzip". Entonces, para comprimir un archivo se hace de la siguiente forma:

zip archivo.zip archivo.mpg

De esta forma se generará un nuevo archivo comprimido en formato *ZIP*. A diferencia de los dos formatos anteriores, el archivo original permanece intacto y se genera uno totalmente nuevo. Colocar el nombre del archivo con extensión *ZIP* no es necesario, pues si solamente se colocara “archivo” en lugar de “archivo.zip” automáticamente el comando le agregará la extensión *ZIP*.

La descompresión correspondiente se hace con el comando “unzip”, de la siguiente forma:

unzip archivo.zip

De igual forma, no es necesario indicarle la extensión *ZIP*, pues el comando entiende que es un archivo de compresión de este tipo. Por otra parte, si se deseara comprimir un grupo de archivos se haría de la siguiente manera:

zip todos.zip *

De esta forma todos los archivos que se encuentren en el directorio actual, serán comprimidos en un solo archivo llamado *TODOS.ZIP*.

Finalmente, para generar los archivos de formato *LZH* por medio de la aplicación “lha” se realiza de la siguiente manera, en esta ocasión se hará desde el *DOS* o “modo consola” de *Windows*:

C:\ lha a archivo.lzh archivo.mpg

Si se desea comprimir más de un archivo y todos de un mismo tipo, se hace de la siguiente forma:

C:\ lha a archivo.lzh *.mpg

Pero si lo que se desea es comprimir archivos de diferentes formatos, se hace de la siguiente forma:

C:\ lha a archivo.lzh uno.mpg dos.mp3 tres.txt

El resultado a obtener es comprimir los archivos *UNO.MPG*, *DOS.MP3* y *TRES.TXT* en uno solo llamado *ARCHIVO.LZH*. Para el proceso inverso, es decir la descompresión, se haría de la siguiente manera:

C:\ lha x archivo.lzh

De esta forma se descomprime el archivo *ARCHIVO.LZH* extrayendo todos los archivos contenidos en él.

2.1.2 Plataforma de aplicación

Aunque en la actualidad existen herramientas muy buenas para el soporte de los formatos de archivos de compresión, éstos fueron creados originalmente para un sistema operativo determinado o simplemente algunos sistemas operativos los tomaron como su estándar.

Los formatos *GZ* y *BZ2* desde un inicio han sido creados para sistemas operativos como *UNIX* y *Linux*. Aunque originalmente en *UNIX* se utilizaba el formato Z, se ha descontinuado su uso y ahora se utiliza el formato *GZ*. De igual forma el formato BZ2 es uno de los más utilizados y originalmente creado para *Linux*. Un punto importante que debe aclararse es que cuando se hace referencia a *Linux* inherentemente se hace referencia a *UNIX*, pues no es secreto para nadie que *Linux* está basado en *UNIX*.

Por otra parte, el formato *ZIP* originalmente se utilizaba para el sistema operativo *DOS*, obviamente continuó su uso *Windows* quién lo ha popularizado bastante, y lo ha vuelto uno de sus archivos de compresión estándar.

Aunque antes se utilizaban los comandos "pkzip" y "pkunzip" para comprimir y descomprimir respectivamente, ahora *Windows* soporta una gran variedad de aplicaciones que manejan este formato de compresión. También es soportado por *Linux* a través de los comandos "zip" y "unzip" para comprimir y descomprimir respectivamente.

También se encuentra el formato *LZH* el cual es comprimido por el *LHARC* un programa con base en un algoritmo de empaquetamiento, creado por *Harayasu Yoshizaki* cuyo fuente es de libre distribución. Y aunque en un inicio no se construyó para una plataforma específica, este tipo de archivo soporta ambas plataformas, tanto *Windows* como *UNIX*.

Finalmente debe hacerse una aclaración, algunos de estos formatos pueden ser soportados por *Mac*, *OS/2*, u otros sistemas operativos. Pero en este trabajo de investigación, cuando se menciona que un formato es multiplataforma, se refiere a que por lo menos es soportado por *Windows* y *UNIX*.

2.1.3 Disponibilidad de herramientas para la compresión

Simultáneamente con el avance de la tecnología surgen nuevas herramientas que pueden dar soporte a estos formatos de archivos de compresión. En la actualidad, existe una variedad de herramientas comerciales y no comerciales, pero el hecho es mencionarlas para poder conocer cuánta disponibilidad poseemos para el manejo de estos formatos.

Para el formato gz existen herramientas sencillas como el *WINZIP*, *WINACE* que pueden ser "bajados" fácilmente del Internet. Esto en el sistema operativo *Windows*, por el lado de *Linux* en el modo consola está el comando "gzip". En modo gráfico, existen herramientas propias de *Linux* como el *Ark*, *File Roller* y *Karchiver* entre otros. Para el formato *BZ2* existe muy poca herramienta que lo soporta directamente. Lo único que soportan algunas de estas herramientas es el formato *TAR*, pero la extensión *BZ2* no la reconocen, todo esto en el sistema operativo *Windows*. Entre la poca herramienta que lo soporta se puede mencionar *IZarc* y que puede ser utilizada en *Windows*. Por el lado de *Linux* está directamente en el "modo consola" el comando "bzip2" y en el ambiente gráfico lo soportan las herramientas *Ark*, *File Roller* y *Karchiver* de igual forma que el formato GZ.

El formato *ZIP* sin duda alguna es el formato más soportado por las herramientas de compresión. Desde el modo DOS con "pkzip" y "pkunzip" para comprimir y descomprimir respectivamente hasta el ya conocido WINZIP, también lo soporta *WINRAR*, *WINACE*, *ANACONDA*, *DIRECTORY TOOLKIT*, *FILZIP*, *UTIZIP*, *ZIPGENIOUS*, etc., solo por mencionar algunos.

En la plataforma Linux está el comando en modo consola "zip" y "unzip" para comprimir y descomprimir respectivamente, y en el ambiente gráfico lo soportan las herramientas *Ark*, *File Roller* y *Karchiver*.

Finalmente el formato *LZH* en *Windows* existe un programa que se puede bajar muy fácilmente del Internet llamado *LHA*. También herramientas como *WINZIP*, *WINRAR*, *WINACE* y *ZIPGENIOUS* entre otros lo soportan. Existe una versión de igual forma en *Linux* con el comando *LHA* que funciona de la misma manera que el de *DOS*. En el ambiente gráfico también lo soportan *Ark*, *File Roller* y *Karchiver*.

Un punto importante que debe ser tomado en cuenta es que todos los comandos de "modo consola" y programas de herramientas gráficas mencionados y utilizados para las pruebas en este trabajo de investigación corresponden a los sistemas operativos *Linux SuSE 9.00* y *Windows XP Professional*.

2.1.4 Cantidad de archivos permitidos de comprimir

Como se vio en la sección anterior existe un buen número de herramientas que soportan estos formatos de compresión. La comparación que se realizará de ahora en adelante será con base en el "modo consola"; se tomó la decisión de hacerlo de esta forma pues originalmente estos formatos se hicieron para este modo y todos presentaban ventajas y desventajas desde entonces.

La cantidad de archivos a comprimir tanto en el formato *GZ* como en el *BZ2* por medio de sus comandos respectivos "gzip" y "bzip2" es de un solo archivo. Ambos comandos necesitan la ayuda del utilitario "tar" que empaqueta a un grupo de archivos contenidos en una carpeta convirtiéndolos así en un archivo del tipo *TAR* para posteriormente ser comprimida, solamente de esta forma se puede comprimir más de un archivo con uno de estos comandos. O como se verá más adelante, también podrán realizar esta tarea a través de la aplicación de parámetros en los comandos.

Mientras tanto, la cantidad de archivos que permite comprimir el comando "zip" para generar archivos de este formato va de uno en adelante. Siempre y cuando los archivos a comprimir se encuentren dentro de la misma carpeta todos pueden ser comprimidos en uno solo.

Finalmente, utilizando el comando “lha”, ya sea en *Windows* ó *Linux*, permite comprimir más de un archivo a la vez de igual forma que el formato *ZIP*, incluso nos da la opción de poder comprimir un grupo de archivos de la misma extensión. Como se puede observar claramente, en este aspecto los formatos *ZIP* y *LZH* poseen ventaja sobre los formatos *GZ* y *BZ2*.

2.1.5 Ventajas adicionales con la utilización de parámetros

Hasta ahora solamente se han visto aspectos como facilidad de aplicación, plataforma de aplicación, disponibilidad de herramientas y cantidad de archivos permitida de comprimir, pero también existe otro tipo de comparación a tomar en cuenta: los parámetros de los comandos de compresión; con este nuevo tipo de comparación se podrán observar las ventajas ofrecidas por cada uno de estos formatos a la hora de comprimir, y por qué no, a la hora de descomprimir los archivos.

Estos son algunos de los parámetros ofrecidos por el comando “gzip” para el manejo de archivos *GZ*:

-d permite descomprimir en lugar de utilizar el comando gunzip
-r permite comprimir los archivos contenidos dentro de una carpeta
-v muestra mensajes de lo que está realizando
* comprime todos los archivos contenidos dentro de una carpeta (individualmente)
-# donde # está entre el rango 1 (mayor rapidez) a 9 (mayor compresión)

Este último parámetro es uno de los más interesantes, ya que con el mismo comando se puede indicar el grado de velocidad y el grado de compresión. El número estándar utilizado es el número 6.

Ahora, estos son algunos de los parámetros ofrecidos por el comando “bzip2” para el manejo de archivos *BZ2*:

-k se mantiene el archivo original y se crea uno nuevo con la extensión *BZ2*
-d se descomprime un archivo *BZ2*
-z comprime un archivo al formato *BZ2* (no es necesario colocarlo)
-t ofrece chequeo del archivo en formato *BZ2*
-s se utiliza para reducir el tamaño de memoria a utilizar para manejar los bloques generados del archivo de compresión
-v muestra mensajes de lo que está realizando
-# donde # está entre el rango 1 (mayor rapidez) a 9 (mayor compresión)
-q trabaja en modo silencioso, es decir no muestra ningún mensaje al usuario

Como se puede observar también posee este parámetro interesante de mayor compresión o mayor velocidad, según sea el número que se coloque. Otro aspecto interesante es que se puede manejar la memoria a utilizar a la hora de comprimir, entre otras cosas.

Por otra parte, los parámetros ofrecidos por el comando “zip” son los siguientes:

-r permite comprimir los archivos contenidos dentro de una carpeta
-b permite utilizar el directorio indicado para crear el archivo temporal
-f actualiza un archivo comprimido
-q trabaja en modo silencioso, es decir no muestra ningún mensaje al usuario
-u actualiza un archivo comprimido, permitiendo agregar nuevos archivos
-x esta opción permite excluir uno o más archivos al comprimir

Finalmente estos son algunos de los parámetros ofrecidos por el comando “lha” en el manejo de archivos *LZH*:

a agrega archivos al archivo de compresión
x descomprime el contenido de un archivo lzh
t chequea la integridad del archivo comprimido lzh
l lista él o los archivos contenidos en el archivo comprimido lzh
d borra un archivo que se encuentre dentro del archivo comprimido lzh
q trabaja en modo silencioso, es decir no muestra mensaje alguno al usuario

2.1.6 Matriz comparativa entre formatos de compresión

Tomando en cuenta las comparaciones anteriores se procederá a realizar un análisis y así comparar entre cada formato sus ventajas y desventajas propias de cada formato, para esto observe la Tabla I.

Conjuntamente, para entender entonces los resultados obtenidos y la calificación dada a cada aspecto veamos el siguiente punto que comprende los comentarios y discusión de los resultados.

Tabla I. Comparación con base en la eficiencia y manejo de los archivos de compresión

	Facilidad de Aplicación	**Plataformas**	**Disponibilidad de Herramientas**	**Cantidad de archivos permitidos**	**Utilización de parámetros**
GZ	Regular	**Original:** Linux **Otras:** Windows	**Windows:** Regular **Linux:** Buena	**Directa:** 1 archivo **Indirecta:** 1 ó más archivos	Buena
BZ2	Regular	**Original:** Linux	**Windows:** Regular **Linux:** Buena	**Directa:** 1 archivo **Indirecta:** 1 ó más archivos	Buena
ZIP	Buena	**Original:** DOS **Otras:** Windows, Linux	**Windows:** Buena **Linux:** Buena	**Directa:** 1 o más archivos	Buena
LZH	Buena	**Original:** ninguna **Otras:** DOS, Windows, Linux	**Windows:** Buena **Linux:** Buena	**Directa:** 1 o más archivos	Buena

2.2 Comentarios y discusión de resultados

En el atributo “facilidad de uso” se tomó en cuenta el número de pasos para poder comprimir un grupo de archivos. Tomando en cuenta que tanto “gzip” como “bzip2” necesitan de la ayuda de la utilidad “tar”, su facilidad entra en el rango de “regular”. Mientras que los otros dos formatos lo pueden hacer de forma directa, por lo tanto se les aplicó la calificación de “buena”.

En “plataformas” únicamente el formato *LZH* aparece en su plataforma original como ninguna, pues en un inicio no se desarrollo para alguna en específico, pero luego se distribuyó para poder ser utilizado en *DOS* (*Windows*), *Linux* y otras más.

En “disponibilidad de herramientas” se tomó en cuenta si en dicho sistema operativo se puede realizar en modo consola. Además, si las herramientas gráficas cuentan con el soporte para dicho formato. El criterio tomado para la calificación fue el siguiente: si existe una cantidad considerable de herramientas para poder manejar el formato, se calificó como “buena”; si únicamente lo manejan muy pocas herramientas en modo gráfico se calificó como “regular” y si el formato no se puede manejar en modo consola y tampoco en modo gráfico se calificó como “mala”.

En cantidad de archivos se colocó quienes lo hacen de forma directa para comprimir un grupo de archivos y quienes lo hacen de forma indirecta. Es decir, los que lo hacen en forma indirecta, es porque necesitan de algún comando extra para realizarlo.

Finalmente, la “utilización de parámetros” no se calificó por la cantidad de opciones que posee cada comando de compresión; sino por la función que provee a la hora de comprimir. Y aunque algunos poseen funciones que otros no poseen, cada uno fue considerado y aceptado que proveía buenas opciones con la ayuda de estos parámetros para comprimir, y por eso aparecen todos con una calificación de “buena”.

Porque mientras unos ofrecen descomprimir con el mismo comando, chequear el archivo comprimido, otros ofrecen manipular la memoria a la hora de comprimir, u ofrecen el grado de compresión, en fin, cada uno posee sus propias opciones con sus ventajas y desventajas.

Entonces, analizando la Tabla I, con base únicamente en estos aspectos, pues más adelante se evaluarán otras características, se puede ver claramente que el formato que mejor ventaja ofrece sobre los demás es el formato *ZIP* y el formato *LZH*. Los puntos que les favorece son la cantidad simultánea de archivos que comprimen y la facilidad de uso.

3. ANÁLISIS COMPARATIVO CON BASE EN LA ESTRUCTURA Y SUS MEJORAS ENTRE FORMATOS

3.1 Algoritmos utilizados para la compresión

Hasta el día de hoy existe una infinidad de algoritmos para poder realizar el proceso de compresión y siguen surgiendo nuevas alternativas. En este capítulo se explicarán los algoritmos sobre los cuales están basados los formatos de compresión en estudio. Cada uno de estos algoritmos es la base sobre la cual está construida la versión final de compresión de cada formato, es decir, pueden tener variaciones, para poder mejorar el proceso de compresión; pero su esencia radica en ellos.

3.1.1 LZ77

Este algoritmo está basado en un diccionario para la compresión de texto, creado por *Abraham Lempel* y *Jacob Ziv* en el año de 1977 (por ello recibe el nombre de *LZ77*).

Su funcionamiento tiene un razonamiento muy sencillo, semejante al *LZ78* utiliza una especie de diccionario, pero éste es sustituido por una "ventana" también denominada "ventana corrediza", para entender mejor este concepto, si estuviéramos hablando de un programa, entonces sería un buffer.

Simplemente lo que hace es ir encontrando ocurrencias hacia atrás a partir del caracter que está siendo analizado en ese momento. De esta forma, es buscada en la "ventana" y si es encontrada la ocurrencia, entonces simplemente se genera una salida formada por una dupla: desplazamiento y longitud. Es decir, en que posición se encontró la repetición y que longitud posee, esto obviamente ocupará menos espacio que la cadena original. Si no se llega a encontrar una coincidencia, entonces se generará una copia literal de la entrada.

Se le da el nombre de "ventana corrediza" a este tipo de algoritmos pues la ventana se va desplazando sobre la entrada, en longitud si hubo coincidencia (pues se agrega la dupla desplazamiento y longitud) o bien en carácter si no se encontró (se agrega las literales, que no son más que caracteres sin comprimir).

Para entender mejor, observe el siguiente ejemplo. Si tuviéramos la cadena '123 123', entonces vamos a tomar caracter por caracter:

1) Tomamos '1'; V = ' '. No hay coincidencia.
2) Tomamos '2'; V = '1'. No hay coincidencia.
3) Tomamos '3'; V = '12'. No hay coincidencia.
4) Tomamos ' '; V = '123'. No hay coincidencia.
5) Tomamos '1' V = '123 '. Si hay coincidencia, entonces

desplazamiento = 0 y longitud = 3. (que al final el desplazamiento será de 3, pues vamos de la posición actual hacia atrás)

Como se puede ver, entonces inicialmente se guarda la literal ‘123’ (entiéndase literal como una secuencia de *bytes* sin comprimir) y al encontrar la siguiente cadena como es una ocurrencia, entonces ya no guardará ‘123’ sino la dupla desplazamiento (3) y longitud (3), para este ejemplo. El desplazamiento es tres en este caso, porque la búsqueda se hace a partir del caractér actual hacia atrás, en este caso su longitud es 3, pues es la cadena ‘123’.

Como podemos observar este algoritmo realiza una compresión considerable, con respecto al espacio, pues sustituye la cadena original por una simple dupla. Pero depende mucho del tamaño de la “ventana”, pues se entiende que mientras más grande sea la ventana mayor compresión lograremos, pues al ingresar mayor cantidad de literales, se podrán encontrar más ocurrencias y por lo tanto su proceso será más rápido.

3.1.2 Huffman

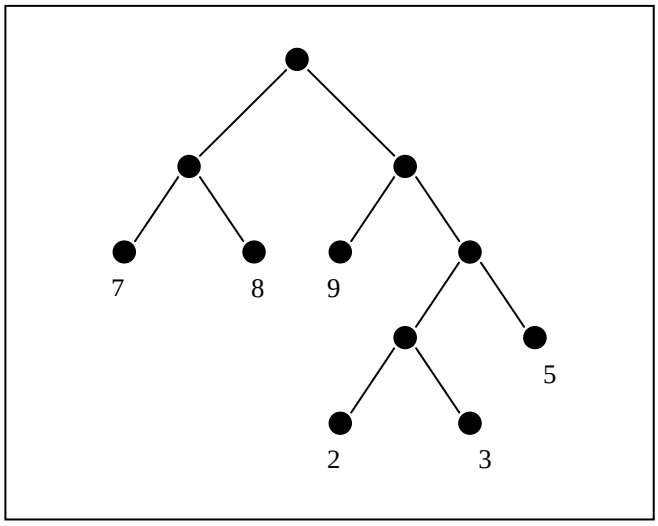
Algoritmo creado por *David Huffman* en el año de 1953 que entra en la categoría de algoritmos estadísticos, pues utiliza las propiedades estadísticas de la fuente para mejorar la codificación, que consiste en ordenar las probabilidades de los caracteres después de combinarlas, generando códigos de longitud variable para cada caracter en particular. Este algoritmo tiene un buen grado de eficiencia ya que su funcionamiento se basa en árboles binarios.

Para comprender mejor el funcionamiento de éste método, observe el siguiente ejemplo, se tomará en cuenta un cierto grado de conocimiento sobre árboles binarios.

Suponga que se tiene un texto en donde al hacer el conteo de sus caracteres generó la siguiente salida de ocurrencias: A = 2, B = 3, C = 5, D = 7, E = 8 y F = 9. El árbol binario generado quedaría como lo muestra la figura 3.1.

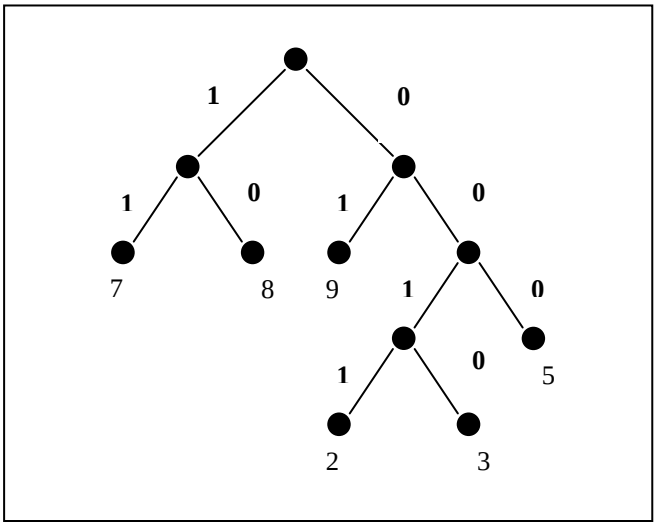
El algoritmo de *Huffman* dice que en base a nuestra lista de probabilidades, en este caso: 2, 3, 5, 7, 8 y 9, se suman las dos probabilidades más pequeñas, se extrae cada uno de los valores sumados y se ingresa el nuevo valor, esto es la suma de ambos. Observando entonces la figura 1, se puede ver que al sumar 2 + 3 se genera una nueva cantidad, es decir 5, quedando la lista ahora: 5, 5, 7, 8 y 9, ahora se suman las dos cantidades más pequeñas, esto es 5 + 5 y así sucesivamente.

Figura 1. Árbol binario



Si se agrega entonces ahora a las aristas los valores 1 y 0 para la izquierda y derecha respectivamente, entonces el árbol quedaría como lo muestra la figura 2.

Figura 2. Árbol binario con peso



Entonces al generar los códigos, recorriendo el árbol, para cada letra, quedarán como lo muestra la Tabla II.

Tabla II. Códigos generados

Caracter	Código
A	0011
B	0010
C	000
D	11
E	10
F	01

Se sustituyen entonces cada uno de los caracteres por su código correspondiente generado, por ejemplo si la secuencia original fuera:

A A B B B C C C C C D D D D D D D E E E E E E E E F F F F F F F F F

Entonces el nuevo código quedaría de la siguiente forma:

00110011001000100010000000000000000… y así sucesivamente.

Es decir, sustituimos el caractér “A” por el código generado “0011” por ejemplo, y así con cada uno de los caracteres. Como se puede ver, la ventaja obtenida es que mientras la ocurrencia de un caracter sea mayor el código generado para él será de menor longitud.

3.1.3 Burrows Wheeler

Este algoritmo fue creado por *Michael Burrows* y *David Wheeler*, método conocido también como *BWT*, método de codificación que en combinación con un buen compresor realiza una excelente tarea.

Como ejemplo, tomemos la cadena ‘LUDIN’; debido a que este método se basa en una matriz de n x n, donde n es la longitud de la cadena que se está analizando, en este caso entonces se generará una matriz de 5 x 5.

Lo primero a realizar es asignar a la primera fila de esta matriz, la cadena original, es decir 'LUDIN'. Ahora, mientras se incremente el número de fila, se coloca la cadena con un corrimiento de un caracter hacia la izquierda. Es decir mientras se incrementa de fila, se realiza un corrimiento de la cadena. Quedará entonces como lo muestra la figura 3.

Figura 3. Matriz de caracteres

	1	**2**	**3**	**4**	**5**
1	L	U	D	I	N
2	U	D	I	N	L
3	D	I	N	L	U
4	I	N	L	U	D
5	N	L	U	D	I

El siguiente paso es ordenar las filas alfabéticamente como se puede ver en la figura 4.

Figura 4. Matriz de caracteres ordenada

	1	**2**	**3**	**4**	**5**
1	D	I	N	L	**U**
2	I	N	L	U	**D**
3	**L**	**U**	**D**	**I**	**N**
4	N	L	U	D	**I**
5	U	D	I	N	**L**

Ahora, observe qué es lo que se ha generado en la columna 5, que aparece resaltada en la figura 4.

Pues esta es la cadena que se utilizará, es decir 'UDNIL', además se puede ver en qué fila quedó la cadena original (LUDIN), que también aparece de forma resaltada, en este caso es la fila 3. Bien entonces la dupla generada para este caso, según *Burrows Wheeler*, es **(UDNIL, 3).**

3.2 Primeras versiones de algoritmos de compresión

La compresión como un proceso formal, ha venido siendo utilizada aproximadamente desde el año 1980. Los primeros compresores que surgieron, por mencionar algunos, fueron *COMPACT* y *COMPRESS*, los cuales fueron utilizando metodologías que hasta esa fecha estaban comenzando a ser mencionadas y también comenzando a ser implementadas en programas. *COMPACT* estaba basado en *Huffman*, por lo mismo resultaba un tanto lento en su procesamiento; y *COMPRESS* estaba basado en la familia de *Lempel-Ziv*, más específicamente en el algoritmo *LZW*.

También fueron surgiendo empaquetadores tales como *TAR*, que lo único que hace es reunir a un grupo de archivos en una misma carpeta para después aplicarle un comando de compresión.

Todo fue migrando de *Huffman* (y sus variaciones) a *LZW*, debido a que este presentaba mejoras en su proceso de compresión. Esta migración se dio por medio del surgimiento del *ARC*.

Esta nueva metodología realizaba ambas tareas en una, es decir empaquetaba y comprimía a la vez, revolucionando así el mundo de las aplicaciones de compresión. La compresión que implementó este nuevo método fue el algoritmo de *Run-Length*, *Huffman estático* y *LZW*.

Luego fueron surgiendo metodología como *LZSS* y *LZARI*, pero debido a la velocidad de los procesadores que existían en este tiempo, eran metodologías que realizaban un proceso muy lento. Después de tanto intento, entonces surgió el *LHARC* (*Harayasu Yoshizaki*) y *ARJ* (*Robert Jung*). Ambos de libre distribución, el primero dio pauta su autor para ser utilizado para cualquier propósito y el segundo para propósitos no comerciales.

Luego apareció el *PKZIP* (*Phil Katz*) tras su intento fallido de "copiar" el método que utilizaba *ARC*, para hacerle frente al popular *ARJ* en ese entonces. Después de esto, fueron surgiendo metodologías como *HA* y otros. En la actualidad han surgido compresores como *RAR*, *LZOP*, *RK*, etc.

3.3 Mejoras realizadas

Como se ha visto los compresores han evolucionado conforme ha pasado el tiempo. Y los algoritmos muy antiguos han ido prevaleciendo, pues la esencia se mantiene. Aunque es obvio que los nuevos compresores realizarán mejoras a estos algoritmos realizando versiones o variaciones, éstas no son más que mejoras en capacidad de manejo de información o una simple mezcla con otras metodologías existentes.

Una mejora evidente a la fecha, es que han pasado de ser metodologías estáticas a metodologías dinámicas, esto es: el manejo de su información ha aumentado en gran capacidad, la compresión por lo tanto ha mejorado y la compatibilidad entre formatos se ha logrado hacer gracias a las potentes herramientas que hoy en día existen.

Los formatos en estudio en este trabajo de investigación, han logrado mantener la esencia de algoritmos que han permanecido a través del tiempo, por supuesto con sus respectivas variaciones. Por ejemplo el formato *GZ*, que es una mejora del formato *Z* utilizado en *UNIX*, utiliza un algoritmo del tipo *Lempel-Ziv*. El formato *BZ2* utiliza el famoso *Burrow-Wheeler* y *Huffman* entre otros. El formato *ZIP* que también utiliza una variación de *Lempel-Ziv* y finalmente el formato *LZH* que utiliza una combinación entre *Huffman* y *Lempel Ziv*.

En la actualidad los expertos se han dedicado a mejorar los compresores de tal forma que su resultado sea el más eficaz y eficiente; pues lo ideal es obtener una máxima compresión en un tiempo mínimo.

Finalmente, también están surgiendo nuevas herramientas que además de ser fáciles de utilizar, están soportando la mayoría de los formatos de compresión y con esto está comenzando a surgir una gran competencia en el campo de la compresión de archivos.

3.4 Compatibilidades entre formatos de compresión

Como se sabe existe una gran afinidad entre los formatos de compresión en estudio, ya que el origen de cada uno de ellos esta basado en la esencia de algún algoritmo de compresión ya conocido, podemos decir entonces que existe cierto grado de compatibilidad entre ellos.

Esto se puede ver claramente en la actualidad, pues están surgiendo una infinidad de herramientas de compresión que soportan una gran cantidad de formatos de compresión simultáneamente y además dichas herramientas son multiplataforma.

Por si esto fuera poco, además de soportar una gran gama de formatos de compresión, tienen la capacidad de realizar conversiones entre formatos, es decir pasar de un formato a otro con la misma herramienta.

Para el usuario entonces cada vez es más fácil conocer y manejar los diferentes formatos de compresión. Su única tarea es tomar la decisión entre obtener una mayor rapidez o una menor cantidad de espacio, con tan solo un *click* del ratón.

Esta compatibilidad se viene dando, también por causa de las mejoras en las versiones, pues los primeros programas que se venían manejando en modo consola, con tan solo agregar un parámetro podíamos generar formatos antiguos, por ejemplo, tenemos el “gzip” que es para los archivos *GZ* y también soporta o puede crear archivos *Z*, que es su antecesor.

3.5 Comentarios

El análisis que se ha descrito a lo largo de este capítulo, tiene como fin dar a conocer los diferentes tipos de algoritmos que utiliza cada uno de estos formatos de compresión y tratar de entender un poco las ventajas y desventajas que presenta cada uno de ellos. En este momento, no se puede llegar con certeza a una conclusión y decidir cual de ellos utilizar y en que situación, pero se puede saber que ventajas tiene cada uno de ellos a la hora de generar sus códigos (codificación) y conocer un poco del proceso de compresión que maneja cada uno de ellos.

Lo importante es tener muy en cuenta que los algoritmos descritos hasta ahora son como los “originales” y que con el tiempo han ido surgiendo variaciones o incluso mezclas entre ellos para poder generar nuevos algoritmos que en realidad son sucesores de los originales.

En el siguiente capítulo, se hará un análisis a fondo y podremos llegar a la toma de decisiones y realizar conclusiones acerca de los formatos de compresión. Se hará un estudio con archivos reales, en tiempo real; observaremos los resultados obtenidos e incluso realizar nuestras propias pruebas y comprobar que las pruebas realizadas en este trabajo se han realizado de una forma cuidadosa y con el menor grado de errores. Además se tratará de detallar todo paso por paso, para que al final se pueda llegar a una razonable conclusión y tomar decisiones propias.

4. APLICACIÓN DE LOS DISTINTOS FORMATOS DE COMPRESIÓN: CASOS REALES

4.1 Selección del grupo de archivos como objeto de estudio

Para realizar las pruebas con casos reales, se tomará un grupo de archivos que poseen ciertas características y un predeterminado comportamiento cuando se les aplica el proceso de compresión. Los casos a estudiar en este trabajo de investigación se clasificarán con base en la cantidad de archivos a comprimir: la primera parte será una compresión unitaria (un único archivo), para observar el porcentaje real de compresión obtenido por los diferentes formatos en estudio. La segunda parte estará formada por la compresión múltiple (más de un archivo simultáneamente), para poder observar el comportamiento masivo al aplicar el proceso de compresión; sólo en esta última prueba se aplicará un control sobre el tiempo de compresión, pues en la compresión unitaria es muy difícil poder realizar una medición debido a la velocidad con que se da dicho proceso. Este estudio nos guiará hacia una conclusión final y por lo tanto la elección del mejor formato que se acople a cualquier necesidad de eficiencia que se esté buscando.

Los formatos de archivos que serán motivo de estudio en este trabajo de investigación serán los siguientes: archivos de texto (*TXT*) que por su formato tienden a obtener un alto porcentaje de compresión. Archivos gráficos (*JPG*) que por su composición obtienen un bajo nivel de compresión.

Archivos gráficos de mapa de bits (*BMP*) que a pesar de ser gráficos, se obtiene un alto porcentaje de compresión. Archivos de audio (*MP3* y *WMA*) que son archivos ya comprimidos, pero en formato de sonido, comúnmente transformados del formato *WAV*.

4.2 Aplicación de los diferentes formatos de compresión

La aplicación de los diferentes formatos de compresión a los formatos mencionados en la sección anterior, será primeramente de forma unitaria, es decir un solo archivo a la vez, y luego se aplicará a un grupo de archivos del mismo formato.

Como se vio en el capítulo 2, en la sección plataforma, los formatos de compresión en estudio son soportados tanto en el sistema operativo *Windows* como en *UNIX* (*Linux*). Para fines de este estudio y con base en que el sistema operativo *Linux*, en el modo consola, posee los comandos necesarios para realizar la compresión de cada uno de ellos, se utilizará este sistema operativo para realizar las pruebas. La finalidad de realizar estas pruebas es el hecho de comparar dos aspectos que prácticamente son las principales características cuando se trata de formatos de compresión: espacio y tiempo.

Ciertamente, como se pudo ver en capítulos anteriores, también existe una serie de parámetros adicionales, que pueden ser utilizados según sea el formato de compresión a obtener, pero en las pruebas que se realizarán en este capítulo se tratará de utilizar lo más "natural" cada comando, es decir, únicamente se utilizarán aquellos parámetros que sean los básicos y necesarios para realizar el proceso de compresión, para que no exista ventaja alguna entre ellos. Y así observar de forma genérica las ventajas y desventajas de cada uno de los formatos de compresión en estudio.

4.2.1 Pruebas con un único archivo (compresión unitaria)

Las primeras pruebas se realizarán a través de la aplicación de los distintos formatos de compresión en estudio, aplicándolos a un único archivo. Las características de los cuatro archivos a ser estudiados se dan a conocer en la Tabla III:

Tabla III. Tipos de archivos a comprimir

Nombre archivo	Formato	Tamaño
texto.txt	Archivo de texto	105.80 KB
audio.mp3	Archivo de audio comprimido	9.40 MB
mapa.bmp	Archivo gráfico de mapa de bits	6.10 MB
Imagen.jpg	Archivo gráfico comprimido	345.90 KB

A cada uno de ellos se le aplicará su comando correspondiente para comprimirlos a los formatos *GZ*, *BZ2*, *ZIP* y *LZH*. El parámetro a comparar es el porcentaje de compresión obtenido con cada uno de los formatos, obviamente por medio de un análisis entre el tamaño original y el tamaño obtenido. La Tabla IV describe a detalle cada uno de los comandos a utilizar para lograr la compresión en su respectivo formato, utilizando como ejemplo el archivo *TEXTO.TXT*.

Tabla IV. Comandos aplicados para el proceso de compresión

Formato Original	Comando Aplicado	Formato obtenido
texto.txt	gzip texto.txt	texto.txt.gz
texto.txt	bzip2 texto.txt	texto.txt.bz2
texto.txt	zip texto.zip texto.txt	texto.zip
texto.txt	lha a texto.lzh texto.txt	texto.lzh

La misma secuencia será aplicada a las otras pruebas unitarias a realizar, para obtener también los respectivos formatos de compresión. Una vez aplicada esta secuencia de comandos, se procederá a realizar una clasificación según el formato original del archivo, es decir, entre los archivos con alto porcentaje de compresión (*TXT* y *BMP*) y los archivos con bajo porcentaje de compresión (*MP3* y *JPG*).

4.2.1.1 Archivos con porcentaje bajo de compresión

Se entiende que estos tipos de formatos de archivo son bajos en su porcentaje de compresión, debido a que previamente han sido comprimidos en sus propios formatos. Es decir, se les ha aplicado un tipo de compresión de tal forma que se reduce su tamaño pero permanecen en su estado de presentación original. Por ejemplo, un archivo *JPG*, es una formato gráfico comprimido por lo general del formato *BMP*, en pocas palabras, ambos son formatos gráficos, solamente que el formato *JPG* ocupa menos espacio que el formato *BMP*.

Entendido lo anterior, entonces se procederá a observar la Tabla V y la Tabla VI, que presentan los resultados obtenidos al aplicar el proceso de compresión al archivo *IMAGEN.JPG* y al archivo *AUDIO.MP3* con cada uno de los comandos descritos en la Tabla IV y que generarán los formatos de compresión que son el objeto de este estudio:

Tabla V. Resultados del porcentaje de compresión obtenido para el archivo imagen.jpg

Archivo Gráfico: IMAGEN.JPG					
Compresor	**Tamaño Original Resumido (KB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(KB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**
Gzip	345.90	354,233	344.20	352,457	0.50%
Bzip2	345.90	354,233	344.10	352,371	0.53%
Zip	345.90	354,233	344.30	352,580	0.47%
Lha	345.90	354,233	344.40	352,622	0.45%

Tabla VI. Resultados del porcentaje de compresión obtenido para el archivo audio.mp3

Archivo de Sonido: AUDIO.MP3					
Compresor	**Tamaño Original Resumido (MB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(MB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**
Gzip	9.40	9,857,024	9.30	9,765,069	0.93%
bzip2	9.40	9,857,024	9.30	9,723,820	1.35%
Zip	9.40	9,857,024	9.30	9,765,191	0.93%
Lha	9.40	9,857,024	9.30	9,787,110	0.71%

Aunque la conclusión con base en los resultados se hará posteriormente, podemos observar rápidamente en la Tabla V y Tabla VI que la mejor compresión se obtuvo al generar el archivo del formato *BZ2*, tanto en el caso del archivo de formato *JPG* como en el caso del archivo del formato *MP3*.

4.2.1.2 Archivos con porcentaje alto de compresión

A diferencia de los formatos de la sección anterior, estos formatos debido a la estructura de su composición, permiten un porcentaje alto de compresión tanto en su mismo formato (como se mencionó anteriormente, por ejemplo transformar del formato gráfico *BMP* al formato gráfico *JPG*) como al comprimirlos a cualquiera de los formatos de compresión en estudio.

Veamos entonces los resultados obtenidos al aplicar la misma secuencia de comandos aplicados a los formatos de la sección anterior, y observe lo que sucede con estos formatos altos en porcentaje de compresión, que en este caso son los archivos *TEXTO.TXT* y *MAPA.BMP*.

Tabla VII. Resultados del porcentaje de compresión obtenido para el archivo texto.txt

Archivo de Texto: TEXTO.TXT					
Compresor	**Tamaño Original Resumido (KB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(KB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**
Gzip	105.80	108,376	27.00	27,632	74.50%
bzip2	105.80	108,376	24.10	24,704	77.21%
zip	105.80	108,376	27.10	27,752	74.39%
lha	105.80	108,376	29.50	30,191	72.14%

Tabla VIII. Resultados del porcentaje de compresión obtenido para el archivo mapa.bmp

Archivo Gráfico: MAPA.BMP					
Compresor	**Tamaño Original Resumido (MB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(MB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**
gzip	6.10	6,365,238	3.60	3,777,431	40.66%
bzip2	6.10	6,365,238	2.70	2,811,663	55.83%
zip	6.10	6,365,238	3.60	3,777,552	40.65%
lha	6.10	6,365,238	4.00	4,201,973	33.99%

De igual forma, que con los archivos de bajo porcentaje de compresión, el resultado indica nuevamente en la Tabla VII y Tabla VIII que el mayor porcentaje de compresión se obtuvo con el formato *BZ2*.

Hasta ahora, solo se ha visto la ventaja de compresión sobre el tamaño de un archivo; observe a continuación al realizar las pruebas con un conjunto de archivos, el tiempo que tarda cada uno de los compresores al realizar el proceso de compresión.

4.2.2 Pruebas con más de un archivo (compresión masiva)

Una vez observado el comportamiento tanto para los archivos de alto porcentaje de compresión como para los de bajo porcentaje de compresión, se realizarán pruebas con estos mismos formatos de archivos, pero masivamente. En esta parte del estudio además de observar el porcentaje de compresión obtenido, en relación al tamaño, se medirá un parámetro adicional que tiene un nivel alto de importancia en el proceso de compresión, este es: el tiempo.

Pues lo deseado, al tratar el tema de la compresión de archivos, es una solución eficiente, y un aspecto que juega un papel importante en dicha eficiencia, es el tiempo que tomará comprimir un buen número de archivos.

Por otra parte, como se mencionó en el primer capítulo dos de los formatos de compresión en estudio necesitan de la ayuda del utilitario *TAR*, que es necesario para empaquetar el grupo de archivos a comprimir.

Por lo tanto, también hay que considerar el tiempo necesario para realizar el empaquetado de los archivos y así poder decidir en base a espacio y tiempo, cual de los formatos nos ofrecen mejores ventajas.

Observe la Tabla IX y la Tabla X, que muestran los resultados obtenidos al comprimir un considerable número de archivos, tanto de alto como de bajo porcentaje de compresión, y observe este nuevo aspecto que es el tiempo total, que toma realizar dicho proceso:

Tabla IX. Resultados del porcentaje y tiempo de compresión obtenidos para archivos TXT

ARCHIVOS DE TEXTO (TXT); CANTIDAD: 14600 ARCHIVOS								
Compresor	**Tamaño Original Resumido (MB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(MB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**	**T. E. (Seg.)**	**T. C. (Seg.)**	**Tiempo Total**
gzip	59.50	62,388,650	15.90	16,707,301	73.22%	26	10	36
bzip2	59.50	62,388,650	13.10	13,708,464	78.03%	26	66	92
zip	59.50	62,388,650	21.60	22,617,077	63.75%	0	38	38
lha	59.50	62,388,650	19.70	20,680,032	66.85%	0	53	53

Como se puede observar en las dos tablas anteriores y que corresponden a los formatos con alto grado de compresión, el formato *BZ2* es el que obtiene un mayor porcentaje de compresión; pero en relación al tiempo total de compresión, podemos ver que su tiempo es considerablemente mayor con relación a los demás formatos. Veamos entonces las siguientes dos tablas que nos muestran los resultados obtenidos con los formatos con bajo porcentaje de compresión:

Tabla X. Resultados del porcentaje y tiempo de compresión obtenidos para archivos BMP

ARCHIVOS DE IMAGEN (BMP); CANTIDAD: 1648 ARCHIVOS								
Compresor	**Tamaño Original Resumido (MB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(MB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**	**T. E. (Seg.)**	**T. C. (Seg.)**	**Tiempo Total**
gzip	339.20	355,643,442	194.50	203,933,578	42.66%	50	65	115
bzip2	339.20	355,643,442	173.80	182,248,373	48.76%	50	572	622
Zip	339.20	355,643,442	194.20	203,679,205	42.73%	0	97	97
Lha	339.20	355,643,442	198.80	208,444,149	41.39%	0	221	221

Tabla XI. Resultados del porcentaje y tiempo de compresión obtenidos para archivos MP3 y WMA

ARCHIVOS DE AUDIO (MP3 Y WMA); CANTIDAD: 114 ARCHIVOS								
Compresor	**Tamaño Original Resumido (MB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(MB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**	**T. E. (Seg.)**	**T. C. (Seg.)**	**Tiempo Total**
Gzip	473.70	496,685,230	466.80	489,433,547	1.46%	59	119	178
bzip2	473.70	496,685,230	466.00	488,662,320	1.62%	59	912	971
Zip	473.70	496,685,230	466.80	489,452,850	1.46%	0	135	135
Lha	473.70	496,685,230	468.30	491,027,587	1.14%	0	236	236

Tabla XII. Resultados del porcentaje y tiempo de compresión obtenidos para archivos JPG

ARCHIVOS DE IMAGEN (JPG); CANTIDAD: 1648 ARCHIVOS								
Compresor	**Tamaño Original Resumido (MB)**	**Tamaño Original Real (B)**	**Tamaño Compreso Resumido(MB)**	**Tamaño Compreso Real (B)**	**Porcentaje De Compresión**	**T. E. (Seg.)**	**T. C. (Seg.)**	**Tiempo Total**
gzip	23.60	24,794,508	22.20	23,264,208	6.17%	1	4	5
bzip2	23.60	24,794,508	22.30	23,427,687	5.51%	1	50	51
zip	23.60	24,794,508	22.60	23,648,496	4.62%	0	8	8
lha	23.60	24,794,508	22.50	23,629,119	4.70%	0	20	20

Como se puede observar desde la Tabla IX hasta la Tabla XII, se agregaron tres columnas, estas corresponden al tiempo. La columna con el encabezado “T.E. (Seg.)” se refiere al tiempo tomado para el empaquetamiento de los archivos a comprimir; la columna con el encabezado “T.C. (Seg.)” se refiere al tiempo tomado para realizar la compresión, y finalmente está la columna del “Tiempo Total” (también en segundos). Es fácil observar que tanto el formato *ZIP* como el formato *LZH* no necesitan pasar por el proceso de empaquetado, debido a que la compresión se realiza de forma directa.

Para los otros dos formatos, *GZ* y *BZ2*, como muestra la Tabla XIII y tomando como ejemplo la carpeta *BMP* que contiene los archivos de este formato, se realizó el proceso de empaquetamiento de la siguiente forma:

Tabla XIII. Proceso de empaquetamiento para una carpeta

Formato Original	Empaquetamiento	Nuevo Formato
/bmp	tar –cvf bmp.tar /pruebas/bmp	bmp.tar

Después de realizar el empaquetamiento, para cada carpeta que contiene los distintos formatos de archivos, se aplican los correspondientes comandos para generar los formatos *GZ* y *BZ2*, de igual forma como cuando se realizó la prueba unitaria; de esta forma al aplicar los compresores "gzip" y "bzip2", se generarán los archivos *BMP.TAR.GZ* y *BMP.TAR.BZ2* respectivamente.

4.3 Análisis y comparación entre los diferentes formatos de compresión

Realizadas las pruebas, se procederá a realizar un análisis y discusión de los resultados obtenidos y así de esta forma poder determinar cual de los formatos de compresión aplicados es el más óptimo en una determinada situación y también poder identificar el formato más estándar, por llamarlo de alguna manera, que será aquel formato, que se logre adaptar a cualquiera de las necesidades requeridas por un usuario a la hora de realizar el proceso de compresión.

4.3.1 Comparación entre el tamaño original y el tamaño comprimido

Esta parte del análisis únicamente se enfocará al tamaño obtenido con cada uno de los formatos de compresión. Como se puede observar desde la Tabla V hasta la Tabla VIII para los archivos únicos comprimidos, la diferencia de compresión entre los archivos de bajo porcentaje y los archivos de alto porcentaje de compresión es considerable.

Tomemos, como ejemplo el compresor *BZIP2*. Al comprimir el archivo de texto, se obtuvo un 77% de compresión. Para verlo mas claro, su tamaño original real es de 108,376 B (*bytes*) y se logró reducirlo a 24,704 B. El comportamiento de los otros compresores está muy cercano al presentado por el compresor *BZIP2*, siendo la diferencia de 2%, 3% y 5% menos obtenido, para el “gzip”, “zip” y “lha” respectivamente.

Pero podemos determinar que para un único archivo en compresión, estas diferencias son mínimas si estamos hablando de espacio ocupado en un dispositivo de almacenamiento.

El mismo comportamiento, se puede observar para los archivos bajos en porcentaje de compresión, pues aunque el compresor “bzip2” sigue a la cabeza, la diferencia con respecto a los otros compresores, es mínima.

4.3.2 Comparación del tiempo de ejecución en la compresión de archivos

Ahora el análisis se enfocará al otro aspecto importante en la compresión: el tiempo. Si bien es cierto algunos compresores son muy buenos en relación al porcentaje de compresión obtenido, pueden presentar ciertas deficiencias en el tiempo tomado para realizar dicho proceso.

Desde la Tabla IX hasta la Tabla XII se pueden observar los resultados para compresiones masivas, claramente, se ve que el compresor *BZIP2* comprime mejor que los demás compresores. Pero si nos enfocamos al tiempo, se puede ver que este aspecto no favorece en nada a dicho compresor. Tomemos como ejemplo los resultados para el grupo de archivos de imagen *BMP* que fueron comprimidos.

El mejor tiempo obtenido lo tiene el compresor *ZIP* (si se habla de tiempo total, tomemos en cuenta, que tanto el compresor “bzip2” y “gzip” necesitan empaquetar primero los archivos) con 97 segundos, que representa un 15% de lo obtenido con el formato “bzip2” que realizó todo el proceso en 622 segundos. Luego le sigue el compresor “gzip” y por último el compresor “lha“ que les tomó un 18% y 35% del tiempo tomado por el compresor “bzip2”.

Al observar los resultados obtenidos con los archivos con porcentaje bajo de compresión (*JPG*, *MP3* y *WMA*) los porcentajes calculados son muy similares a los obtenidos con los archivos con porcentaje alto de compresión, pues aunque sigue

prevaleciendo en ocasiones, el compresor “bzip2” con respecto al parámetro espacio, pues con relación al tiempo de compresión tiene un valor alto.

4.3.3 Comparación espacio – tiempo

Ahora de forma simultánea analicemos los parámetros espacio y tiempo, tomando en cuenta únicamente las compresiones masivas, pues como se explicó anteriormente solo en esta parte de las pruebas se tomó el tiempo del proceso de compresión realizada por cada uno de los compresores de los diferentes formatos de archivos de compresión en estudio, se puede ver que el formato que toma la mejor ventaja ante estos dos aspectos es el *GZ*.

Aunque la comparación global se analizará en la sección siguiente, observando los resultados presentados por las Tablas IX a la Tabla XII es fácil inclinarse por el formato *GZ*. Porque se busca un espacio reducido y un tiempo corto en el proceso de compresión, y el formato *GZ* se acerca a la rapidez del formato *ZIP* y al espacio reducido del formato *BZ2*.

Como se vio en la sección de ventajas adicionales, en donde se explicó el manejo de lo parámetros, existe un parámetro que ayuda a obtener un mejor tiempo de respuesta o un mejor tamaño de compresión; lo seguro es que al tratar de manipular este parámetro haciendo que el formato *GZ* trabaje más rápido de forma similar como el formato *ZIP*, disminuiría su eficiencia en el tamaño obtenido; de igual forma si se manipulara el parámetro de tal forma que se obtuviera un mejor tamaño como *BZ2*, el tiempo ya no sería tan eficiente como lo hace en su estado natural.

Es por ello que el proceso de compresión efectuado en este trabajo de investigación, para obtener mejor información y no entrar en confusiones, se ha realizado sin la ayuda de estos parámetros, para poder decidir realmente cual de los formatos posee la mejor eficiencia en cuanto a rendimiento global, que hasta este momento lo posee el formato *GZ*.

4.3.4 Comparación entre los formatos de compresión

Comparando los formatos, según los resultados mostrados desde la Tabla V hasta la Tabla XII tomando en cuenta tanto la compresión masiva como la compresión unitaria sin duda alguna el compresor que obtuvo el mejor tamaño de compresión fue el "bzip2"; sin embargo, en relación al tiempo los compresores que realizaron mejor su trabajo con respecto a esta característica fueron tanto el "gzip" como el "zip", pues la diferencia entre ambos, con respecto al tiempo, es mínima.

También, observando las diversas tablas de resultados, para nadie resulta difícil ver que el formato más deficiente sin lugar a duda es el *LZH*, tanto en tiempo como en compresión.

Acertadamente se puede llegar a la conclusión de que depende mucho de la situación en la que estemos, para poder aplicar así un determinado formato de compresión; pero también podemos, con base a todas las características presentadas por cada uno de los formatos determinar que uno de estos cuatro formatos en estudio, resulta ser el más eficiente, sin importar dicha situación, y que puede llegar a satisfacer casi todas las necesidades que busca un usuario a la hora de comprimir.

En la próxima sección se llegará a la toma de una decisión con respecto al formato que al parecer es el que mejor se adapta en cuanto a todas las características que se buscan cubrir en el proceso de compresión de un archivo.

4.4 Discusión y conclusiones de los resultados de las pruebas

Para llegar a una conclusión definitiva, se tomará como base los resultados que se han obtenido con las pruebas realizadas con los distintos formatos de compresión; se tratará de tomar un criterio general, y como se ha mencionado en ocasiones anteriores, le elección se realizará sobre aquel formato que presentó un comportamiento de tal forma que cubrió la mayor parte de las necesidades requeridas por un usuario.

Tomando en cuenta aspectos tan simples como su facilidad de uso, la cantidad de herramientas disponibles, su soporte en varias plataformas, etc. hasta aspectos mas complejos como espacio obtenido de compresión y tiempo para realizar este proceso, se ha podido llegar a la conclusión que el formato que mejor se adapta a la mayoría de las necesidades básicas de un usuario definitivamente es el formato *GZ*.

Hasta antes de realizarse las pruebas masivas, se estaba creando la idea de que definitivamente el formato *BZ2* era el mejor para realizar la compresión, porque siempre obtenía el mejor porcentaje de compresión; pero después de realizarse las pruebas masivas y de observar el tiempo tomado por este formato para realizar el proceso de compresión, definitivamente para un usuario resultaría un tanto desesperante utilizar este tipo de compresión, si lo que busca es un tiempo mínimo.

El formato *ZIP*, definitivamente presentó una proximidad al formato *GZ*, en cuanto a tiempo, pero su compresión con respecto al espacio, presentaba cierta desventaja. Aún siendo un formato muy comercial y sin duda alguna el más conocido y manejado, presenta ciertas desventajas en cuanto a espacio y tiempo. Y qué decir del formato *LZH*, que se mantuvo en último plano, en comparación con los otros tres formatos, en relación a todos los parámetros comparados. Lo cierto es que este es un formato deficiente, en relación a cualquier aspecto, el cual fue incluido en este trabajo de investigación para darlo a conocer y realmente presentar su comportamiento en el proceso de compresión.

Finalmente, con el paso del tiempo definitivamente los compresores están mejorando la forma de realizar la compresión, haciendo nuevas ediciones y versiones para realizar este proceso de la forma más eficiente. Incluso construyendo nuevas herramientas de tal forma que sean los más globales posibles, es decir, abarcando y soportando la mayor cantidad de formatos de compresión y haciéndole las cosas mas sencillas en cuanto a manejo y soporte a los usuarios.

CONCLUSIONES

1. Existe una diversa cantidad de formatos de archivos de compresión, que por el motivo que sea, no son muy conocidos, pero que presentan una igual o mayor eficiencia en el proceso de compresión que los formatos más conocidos o comerciales.

2. Cada formato de archivo de compresión presenta sus propias ventajas y desventajas correspondientes debido al algoritmo utilizado para realizar el proceso de compresión. Esto hace que cada formato sea utilizado según sea la necesidad a cubrir por el usuario.

3. Aunque cada formato presente una serie de ventajas y desventajas en relación con los demás formatos, existen formatos que pueden adaptarse a cualquier necesidad a ser cubierta a favor del usuario, desde cualquier punto de vista que se quiera comparar.

RECOMENDACIONES

1. Determinar qué herramientas hay disponibles para el formato que se va a utilizar.

2. Determinar si el compresor a utilizar es soportado por el sistema operativo en uso.

3. Determinar si la compresión la realiza de forma directa o si necesita de algún otro programa o utilitario.

4. Investigar y determinar si el compresor a utilizar tiene un alto grado de facilidad de uso.

5. Buscar las ediciones más recientes de los compresores a utilizar, para así obtener información y conocer las ventajas que presentan con respecto a ediciones anteriores.

6. Investigar sobre el surgimiento de nuevos formatos o formatos desconocidos, pues pueden presentar ventajas sobre los formatos tradicionales o con los cuales se ha vuelto costumbre trabajar.

BIBLIOGRAFÍA

1. Kenneth A. Ross – Charles R.B. Wright. Matemáticas Discretas. (2ª Edición; México: Editorial PRENTICE-HALL HISPANOAMERICANA, S.A., 1988), pp. 486-492

2. **Extensiones varias**
   http://www.red-spring.com.ar/ExtensionesVarias.doc (03/03/2004)

3. **Comandos de almacenamiento**
   http://www.recursos-as400.com/comolinux07.shtml (26/03/2004)

4. **Manual del principiante de *Red Hat Linux***
   http://linux-cd.com.ar/manuales/rh9.0/rhl-gsg-es-9/s1-managing-compressing-archiving.html (26/03/2004)

5. **Manual oficial del principiante de *Red Hat Linux***
   http://www.europe.redhat.com/documentation/rhl7.3/rhl-gsg-es-7.3/s1-managing-compressing-archiving.php3 (26/03/2004)

6. **Manual oficial del principiante de *Red Hat Linux***
   http://www.europe.redhat.com/documentation/rhl7.2/rhl-gsg-es-7.2/s1-zip-tar.php3 (26/03/2004)

7. **Comandos de almacenamiento**
   http://www.recursos-as400.com/comolinux07.shtml (26/03/2004)

8. **Descomprimir archivos *.zip encriptados en *Linux***
http://bulma.net/body.phtml?nIdNoticia=641 (26/03/2004)

9. **Juntando y comprimiendo archivos**
http://acm.asoc.fi.upm.es/documentacion/lpractico/node11.html (26/03/2004)

10. **Formatos y extensiones de archivos**
http://www.learnthenet.com/spanish/html/34filext.htm (26/03/2004)

11. **Glosario - *ASCII***
http://www.learnthenet.com/spanish/glossary/ascii.htm (26/03/2004)

12. **Glosario – Fichero Binario**
http://www.learnthenet.com/spanish/glossary/binary.htm (26/03/2004)

13. **Formateo de documentos para la interoperabilidad**
http://www.mexicoextremo.com.mx/ayuda/librolinux/index.php?nombre=linux_capitulo4-5.html (26/03/2004)

14. **La compresión de los archivos**
http://www.donde.net/elzip.html (30/03/2004)

15. ***TAR/GZIP/BZIP2***
http://linuca.org/body.phtml?nIdNoticia=76 (26/03/2004)

16. **Algoritmos principales más usados**
http://www.galeon.com/odiseus/info/algoritm.htm (01/04/2004)

17. Redes en *Linux*

http://www.htmlweb.net/linux/redes/redes_linux_7.html (01/04/2004)

18. Breve historia de las herramientas de compresión

http://www.galeon.com/odiseus/info/intro.htm (01/04/2004)

19. *Burrows Wheeler Decoder*

http://acm.uva.es/problemset/v7/741.html (01/04/2004)

20. *Data Compression with the Burrows-Wheeler Transform*

http://www.dogma.net/markn/articles/bwt/bwt.htm (01/04/2004)

21. *WinUDA*

http://guti.bitacoras.com/index.php?entry=entry050926-212839 (10/05/2006)

22. Nuevo formato de compresión

http://www.forospyware.com/t12417.html (10/05/2006)

23. *WinRK*, compresión profesional para todos

http://www.terra.es/tecnologia/articulo/html/tec12989.htm (10/05/2006)