¿Existe el **COMPRESOR** perfecto?

Es común recibir archivos comprimidos (típicamente por mail), cuando es necesario minimizar todo lo posible el tamaño de los archivos enviados y así superar la limitación impuesta por los sistemas de correo. Para muchos, el procedimiento es misterioso: ¿qué significa comprimir? Y además, ¿no podríamos comprimir nuevamente un archivo comprimido y así reducirlo aún más? Veremos más de cerca esta cuestión y sus inesperadas consecuencias.

Por Ing. Edgardo García

Gerente de Sistemas y Tecnología, Editorial Atlántida SA.
Profesor Titular de Procesos Digitales y Gestión de Color, Fundación Gutenberg

¿QUÉ ES COMPRIMIR?

Para la mayoría de las personas, la palabra comprimir tiene un claro significado mecánico, y tiende a imaginar que un archivo puede "comprimirse" en el mismo sentido que uno "comprime" un abrigo para que entre de una buena vez en el placard. Claro que un archivo no es algo "plástico" (ni siquiera es una cosa sólida) que uno pueda "apretar", a pesar de lo cual uno de los programas más populares que usamos para ello tiene por ícono una pequeña prensa. Así que… ¿qué le pasa a un archivo cuando se comprime?

En pocas palabras, se trata de guardar la misma información utilizando menos bits que el archivo original, mediante un proceso reversible que me permita más tarde recuperar ese archivo original. Imaginemos un archivo de texto. En general cada carácter de ese texto requiere 8 bits de información. Recordemos que cada bit representa una de dos posibilidades (un "0" o un "1"), así que una serie de 8 bits se puede hacer de 28 = 256 maneras distintas. Para ver la razón, pensemos que una secuencia de un solo bit daría dos posibilidades, "0" y "1"; en una de dos bits,

GRÁFICO I

Archivo	Tamaño	Compresión relativa al anterior
Original	22.404.024 bytes	-
Primera compresión	14.925.081 bytes	-33%
Segunda compresión	14.928.614 bytes	+0,02%
Tercera compresión	14.933.345 bytes	+0,03%

GRÁFICO II

Cantidad n de bits del archivo original	Cantidad de archivos de n de bits de longitud (Conjunto A)	Total de archivos de longitud menor a n bits (Conjunto B)	"Déficit" de B comparado con A
1	21 = 2	Ninguno	2-0= 2
2	22 = 4	2	4-2= 2
3	23 = 8	2+4= 6	8-6= 2
4	24 = 16	2+4+8= 14	16-14= 2
5	25 = 32	2+4+8+16= 30	32-30= 2
6	26 = 64	2+4+8+16+32= 62	64-62= 2
7	27 = 128	2+4+8+16+32+64= 126	128-126= 2

para cada posible primer bit hay dos posibles segundos bits, es decir 2x2 posibilidades ("00", "01", "10", "11"); luego, 8 bits podrán generar 2x2x2x2x2x2x2x2 = 28 secuencias distintas, y cada una de ellas representará un carácter. Por ejemplo, la "J" se representa por la secuencia 01001010.

Usar secuencias de 8 bits me permite abarcar un "alfabeto" de 256 caracteres (letras, números, símbolos de puntuación), pero en la mayoría de los textos no se usan TODOS esos símbolos. En muchos casos podríamos arreglarnos con menos, y usar así secuencias de bits más cortas (7 bits alcanzarían para 128 símbolos, 6 bits para 64, etc.). Comprimir es, sencillamente, usar la cantidad más justa posible de bits que me alcancen para representar los caracteres que efectivamente necesito. Además, ciertos caracteres aparecen con más frecuencia que otros (en el castellano la letra "E" es la más frecuente), y es posible aprovechar este hecho empleando menos bits para los caracteres que aparecen más seguido.

¿Y en una imagen? En RGB, cada pixel utiliza 24 bits de información, lo que alcanza para 224 = 16.777.216 colores diferentes posibles. ¿Se usan todos esos colores? En general no; por lo tanto sería posible representar la misma imagen usando menos bits, tomando nota de cuántos colores se utilizan realmente en esa imagen particular, y cuáles se usan con más frecuencia.

Uno puede legítimamente sospechar que este

proceso de compresión no puede repetirse logrando siempre comprimir más en cada paso; si así fuera, podría reducir cualquier archivo a unos pocos bits. Sería genial tener todas las imágenes, páginas web, videos y música que existen en internet en un pen drive… Entonces, comprimido ya el archivo, ¿qué pasa si lo comprimimos otra vez?

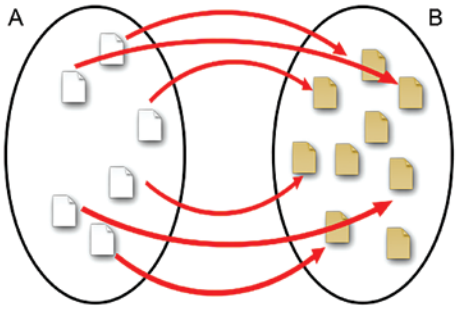
SEGUNDAS PARTES NUNCA FUERON BUENAS

Para nuestra sorpresa, la segunda compresión no sólo no reduce el tamaño del archivo, sino que lo aumenta. La razón es que un método de compresión bien diseñado debe ser capaz de encontrar una mínima secuencia de bits a partir de la cual el archivo original puede reconstruirse; luego debe "empaquetar" esta secuencia en el archivo comprimido, pero además debe incluir otros datos, por ejemplo el nombre del archivo original, su fecha de creación o modificación, etc., es decir sus metadatos. Tenemos entonces que:
Archivo comprimido = Secuencia mínima de bits del archivo original + metadatos

Si comprimimos nuevamente, la secuencia ya comprimida no puede comprimirse más (ya que obtuvimos una secuencia mínima); los metadatos quizás puedan comprimirse, pero debo agregarle los nuevos metadatos de esta segunda compresión, por lo que obtendríamos:
Segundo archivo comprimido = Secuencia mínima de bits de (secuencia mínima + metadatos) + metadatos

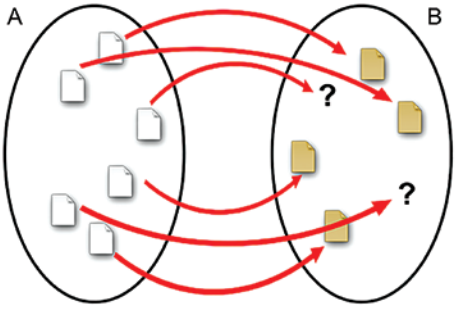
lo que muestra claramente que este segundo archivo comprimido tendrá un tamaño superior al primero. El aumento es relativamente pequeño, ya que los metadatos suelen ser una ínfima cantidad de datos comparados con el archivo original. En el siguiente experimento tomé un archivo Photoshop .psd (que suelen comprimir bien, es decir, la primera compresión logra una buena reducción) y lo sometí a compresiones sucesivas. Este es el resultado (gráfico I).
Claro que aquí la diferencia es pequeña, pero los archivos vueltos a comprimir efectivamente aumentan su tamaño.
Uno podría objetar que la culpa la tienen los metadatos. En parte sí, pero, si los omitiéramos, a lo sumo lograríamos que el tamaño de archivo se mantuviera constante, no logrando ninguna compresión efectiva.

GRÁFICO III



Un método de compresión visto como un "mapeo" entre los archivos "originales" (A) y los archivos comprimidos (B). Necesitamos que en B haya por lo menos tantos elementos como en A.

GRÁFICO IV



Lo que sucede en realidad es que en B siempre hay menos elementos que en A, entonces no puede haber una relación 1 a 1 entre archivos originales y archivos comprimidos distintos.

¿ES POSIBLE CREAR UN COMPRESOR PERFECTO?

Lo notable de este asunto es que, por bueno que sea el método de compresión elegido, siempre habrá un archivo que ese método no pueda comprimir. Esto sugiere que no es posible diseñar un compresor que comprima cualquier archivo. Esta drástica afirmación puede demostrarse sencillamente mediante lo que se llama el principio del palomar: si tenemos "n" nidos de palomas y "p" palomas, si hay más palomas que nidos y todas las palomas están en algún nido, entonces en un nido (por lo menos) hay dos o más palomas. Por ejemplo, en una empresa donde trabajan

400 personas, es seguro que por lo menos dos de ellas cumplen años el mismo día. Aquí las personas son las palomas y los 365 días del año son los nidos; como hay más personas que cumpleaños posibles, se aplica el principio.

Supongamos entonces un archivo de n bits de tamaño; comprimirlo significa crear otro que pueda revertirse al original y que tenga menos de n bits. Planteado así, un método de compresión cualquiera equivale a establecer una relación entre todos los archivos de hasta n bits (que llamaríamos el conjunto A de los archivos originales) y todos los archivos de menos de n bits (que sería el conjunto B de todos los archivos comprimidos posibles a partir de A). Nuestra condición de que el método sea reversible implica que cada elemento del primer conjunto tiene uno y sólo un representante en el otro; de esta forma, dado un archivo comprimido, siempre podremos invertir el procedimiento para recuperar el archivo que le dio origen (gráfico III).

Hasta acá todo bien, pero ¿es posible hacer esto siempre? Para ello es necesario que en el conjunto B haya (por lo menos) tantos elementos como en el A. Así que vamos a contarlos. Para simplificar, limitémonos a archivos de tamaño fijo. Un archivo de n bits se puede crear de 2n maneras diferentes: ésa es la cantidad de elementos de A.

Por otro lado, ¿cuántos archivos de menos de n bits existen? Hagamos un cálculo simple aprovechando el análisis anterior. Contemos cuántos archivos hay de exactamente 1, 2, 3, etc. bits y sumemos. La tabla siguiente muestra este cálculo (gráfico II).

¿Qué significa este resultado? Como el conjunto B de archivos comprimidos (los nidos) tiene 2 elementos menos que el conjunto A de archivos originales (las palomas), para un tamaño fijo de bits y cada conjunto de archivos de ese tamaño, dos de ellos (como mínimo) no podrán comprimirse; de lo contrario, existirían al menos dos archivos originales (dos palomas) distintos que darían como resultado el mismo archivo comprimido (nido) y no habría forma de revertir el proceso: no podríamos diferenciar los originales (gráfico IV).

Por lo tanto, sin importar cuál sea el algoritmo, aplicando el principio del palomar podemos demostrar que no puede existir un compresor "perfecto" que comprima cualquier archivo.

CONSECUENCIAS FILOSÓFICAS

Si nos atenemos a la leyenda, la manzana que cayó en la cabeza de Newton le permitió a éste elaborar su teoría de la gravedad. Pensemos ahora en un libro donde se ha consignado todas las veces que una manzana cayó de un árbol: la altura de la caída, su duración y la velocidad del impacto. Ese libro sería más o menos voluminoso, pero podría ser reemplazado sin pérdida por la ley que Newton descubrió, que escrita ocuparía apenas una página, ya que esa ley es capaz de explicar o predecir esas cantidades.

Desde este punto de vista, una teoría no es más que la versión comprimida de todos los eventos que esa teoría es capaz de explicar. Podríamos decir que "entendemos" la gravedad desde que existe esa teoría; luego, entender es comprimir.

El resultado notable de que no existe el compresor perfecto, llevado a este plano, tiene una profunda e inquietante consecuencia: deben existir hechos físicos reales que, sin embargo, no seremos nunca capaces de explicar o predecir. En otras palabras, existen verdades que no se pueden entender.

Y no termina aquí. En matemáticas siempre se ha supuesto que cualquier afirmación podría demostrarse ser verdadera o falsa. Y si aún no lo estaba, sería cuestión de tiempo que alguien diera con la demostración correcta. Ahora sabemos que quizás no puedan demostrarse; algunas afirmaciones podrían ser ciertas, pero no lo sabríamos. Nunca.