

(LA IMPOSIBILIDAD DE) EL COMPRESOR INFINITO

Desde hace muchos años vemos anuncios de empresas que dicen haber inventado compresores de propiedades asombrosas: ratios de compresión de 100 a 1, de 1000 a 1, etc. ¿Por qué no los usamos si las ventajas son tan evidentes?

Bit, byte, compresión, tipo de datos... terminología que hace diez años hubiera dejado boquiabierto al más templado es hoy en día de uso corriente. Gracias al avance de la Sociedad de la Información, la informática y su vocabulario se han expandido como la pólvora, a la par que Internet se introducía en miles de hogares y empresas.

Además de hacer la declaración de la renta y comprar libros, Internet ofrece muchas otras posibilidades. El deseo de transmitir o almacenar cada vez un mayor volumen de datos sin cambiar el medio (línea telefónica, tarjeta de memoria, etc.) hace que se trabaje intensamente en el campo de la Teoría de la Información, especialmente, en la compresión de datos.

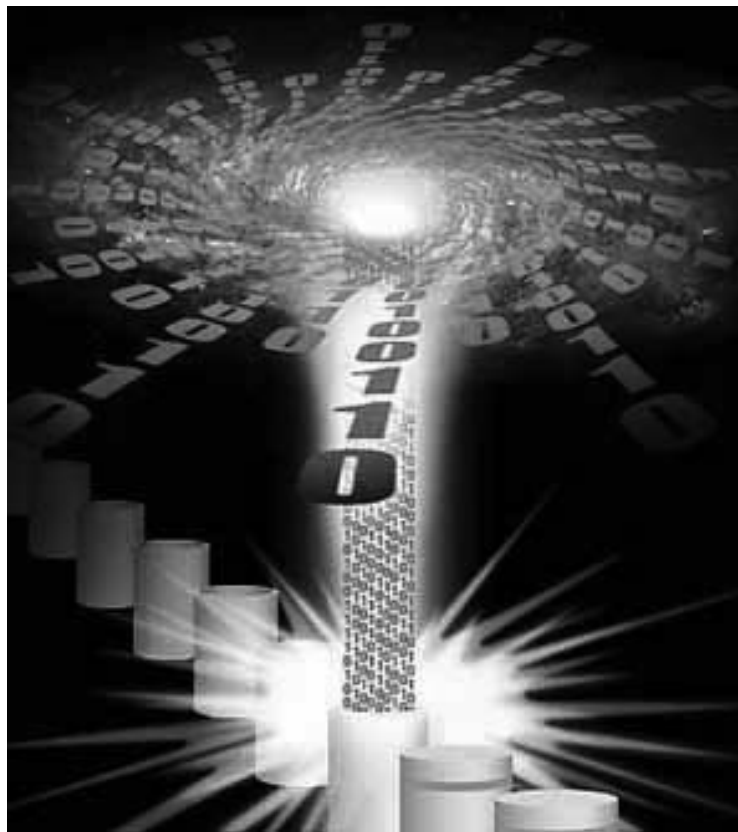
Un compresor es un algoritmo que disminuye el espacio ocupado por cierta información. Cada año vemos en los medios científicos y especializados (New Scientist, Nature, Byte, ZDNet, etc.) varios anuncios de empresas que aseguran haber superado el límite teórico (límite de Shannon) de compresión de datos. Generalmente, estos anuncios van asociados a peroratas como "necesitamos más dinero

para terminar las investigaciones", "tenemos la tecnología pero hace falta un socio interesado en comercializar la tecnología" (ergo gastar dinero en la empresa), etc. ¿Les suena de algo?

Un estudiante irlandés de 16 años consiguió engañar a un jurado de doce personas y ganó el Premio al Joven Científico del Año 2002 con su "invento", un navegador de Internet llamado XWebs, que supuestamente multiplicaba por cuatro la velocidad de navegación.

ZeoSync anunciaba en su página nada menos que cinco tecnologías relacionadas con la compresión de datos: codificador (BinaryAccelerator™), compresor (BitPerfect™), secuenciador (Zero Space Tuner™), transformador de dominio (Relational Differentiation Encoding™) y el conjunto de todo (TunerAccelerator™).

Pegasus Technologies afirmaba que podía guardar 1,28 Gigabytes en un disquete de 3,5" (cuya capacidad normal es 1,44 Megabytes) gracias a su tecnología HyperDrive™ (de la



que llegaron a obtener una patente —cosa fácil en EEUU, todo sea dicho—. Web Technologies nos asombraba con su compresor DataFiles/16™, que comprimía cualquier archivo mayor de 64 KBytes a un dieciseisavo de su tamaño. David C. James obtuvo

Comprimir consiste en eliminar la redundancia.

una patente en EEUU para su tecnología HyperSpace™, que alardeaba de comprimir datos aleatorios. Premier Research Corporation hizo una aparición estelar con su compresor Minc, otro que decía poder comprimir datos aleatorios. Pixelon se aprovechó de la burbuja de las puntocom para vender humo y dilapidar 35 millones de dólares estadounidenses de sus inversores, mientras decía haber inventado un compresor de vídeo y audio de propiedades casi mágicas. Todas estas empresas han desaparecido.

Para entrar en materia, veamos un método sencillo de compresión. Si sustituimos las repeticiones de una letra por el número de veces que se repite podríamos comprimir esta secuencia:

AAAAAAAAAAAAAAAAABAAAC
DEEEEEEECCCBFFF
WCCAAAAJJLLLLL

como:

15AB3ACD6E4C3BF4TW2C5A3J5L

Lo que antes ocupaba 55 caracteres, ahora ocupa sólo 26: ¡una disminución del 53%!

En el ejemplo anterior hemos usado una versión bastante pedestre de un algoritmo de codificación por longitud de recorrido (*Run-Length Encoding*, RLE), pero que muestra la idea básica: eliminar aquello que se repite.

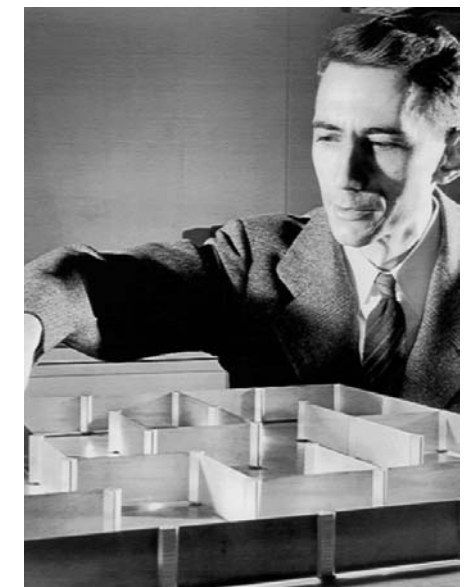
Los compresores eliminan la redundancia, posibilitando que una conexión a Internet lenta permita transmitir grandes volúmenes de datos, que se puedan realizar llamadas telefónicas y videoconfe-

rencias por Internet, etc. En resumen: permiten exprimir más nuestra capacidad de transmisión y almacenamiento de datos.

CONCEPTOS BÁSICOS DE TEORÍA DE LA INFORMACIÓN

La teoría de la información es la rama de la matemática que estudia qué es la información, cómo medirla, cómo representarla y cómo transmitirla. Este campo es uno de los más jóvenes de las matemáticas: su origen es el artículo Una teoría matemática de la comunicación, publicado en 1948 por Claude Shannon

Tipos de compresores: *lossy* y *lossless*



Claude E. Shannon, el padre de la Teoría de la Información. (Bell Labs)

En compresión de datos se distingue entre dos tipos de compresores, los compresores sin pérdida (*lossless*) y los compresores con pérdida (*lossy*).

La diferencia es evidente: cuando comprimimos un fichero de ordenador con un compresor *lossless*, al descomprimirlo

obtenemos exactamente el mismo fichero que teníamos en origen, porque el compresor no produce pérdida de información. Con un compresor *lossy*, en cambio, es imposible recuperar el fichero tal y como era en origen, porque el compresor ha producido pérdida de información. Como contrapartida, los compresores *lossy* alcanzan niveles de compresión mucho mayores que los compresores *lossless* (a costa de la calidad del fichero obtenido).

Hay aplicaciones en las que es imprescindible usar compresores *lossless*, por ejemplo, en un documento de un procesador de texto: si usáramos un compresor *lossy*, perderíamos información, y al descomprimirlo, podríamos encontrarnos con que nos faltan frases, párrafos o imágenes.

En otras aplicaciones, en cambio, se puede usar sin problemas un compresor *lossy*. Los ejemplos más claros son la voz y la imagen. Como ni nuestro oído ni nuestro ojo son perfectos, una disminución moderada de la calidad es inapreciable. Los conocidos formatos MP3, MPEG, AVI, etc. son formatos de compresión *lossy*: se consigue que el audio o el vídeo ocupen poco espacio a costa de disminuir su calidad.

Después de ver en qué consiste la compresión *lossy* y la compresión *lossless*, podemos intuir que:

- La compresión *lossless*, al ser sin pérdida de información, tiene un límite máximo de compresión.

- La compresión *lossy*, al ser con pérdida de información, no tiene

