

Cálculo básico do espazo ocupado por distintos tipos de datos

Sumario

- 1 [Introdución](#)
- 2 [Números](#)
- 3 [Texto](#)
- 4 [Imaxes](#)
- 5 [Audio](#)
- 6 [Vídeo](#)

Introdución

Un sistema informático manexa información de todo tipo (números, texto, imaxes, son, vídeo, etc.), dándolle entrada, saída ou procesándoa. Para iso utilizará distintos mecanismos especiais de representación, almacenamento e presentación tal e como veremos a continuación.

Baixo a nosa perspectiva humana é fácil diferenciar rapidamente o que son números do que é texto, imaxe, son ou vídeo, utilizando para isto os órganos sensoriais e o cerebro.

O ordenador no seu funcionamento trata de emular o comportamento humano, pero, ao ser unha máquina dixital, que ten como soporte a electrónica, só é capaz de representar información binaria, polo que os ordenadores necesitan codificar a información do mundo real ao equivalente binario e utilizar mecanismos para a súa presentación.

Dende os inicios da informática a codificación foi sempre problemática entre outras cousas pola falta de acordo na representación desa información. Hoxe en día existen numerosos [estándares](#) para tal fin.

Números

Os números se almacenan dependendo do tipo de valor que sexa (natural, enteiro ou real), utilizando diferentes sistemas de representación numérica como é o caso do complemento á base ou o exceso á base para números enteiros ou estándar [IEEE 794](#) para números reais.

Texto

Para o caso do texto, o que se fai é codificar cada carácter da cadea a almacenar empregando unha serie de valores binarios cós que se corresponde de acordo a un determinado código.

O código [ASCII](#) foi, durante moito tempo, o mais empregado.

Pouco despois apareceu un problema: este código é suficiente para os caracteres da lingua inglesa, pero non para outras linguas. Entón engadiuse un octavo bit para representar outros 128 caracteres que son distintos segundo idiomas (Europa Occidental usa uns códigos que non utiliza Europa Oriental), chamándose así [ASCII Estendido](#).

Unha ampliación deste método de codificación é o código [Unicode](#) que pode utilizar ata 4 bytes (32 bits) para representar cada carácter, có que é capaz de codificar calquera símbolo en calquera lingua do planeta utilizando o mesmo conxunto de códigos.

Pouco a pouco, o Unicode, é o código que se vai estendendo pero o dominio histórico que levou o código ASCII complica a súa popularidade.

No caso do almacenamento de outro tipo de datos, como imaxes, vídeo ou audio, necesítase unha codificación moito mais complexa, non chegando unha correlación símbolo-secuencia de bits. Ademais neste tipo de información non hai un patrón que se repita, polo que hai decenas de formas de codificar.

Imaxes

Para as imaxes, unha forma básica de codificalas en binario son as chamadas **imaxes rasterizadas, matriciais ou de mapa de bits**, nas que se almacena a información de cada píxel (cada un dos puntos distinguibles na imaxe) descompóndoo en tres cores primarias (R ou nivel de vermello, G ou nivel de verde, B ou nivel de azul), con valores de tamaño dependendo do número de cores que admita a representación; o que se coñece como "profundidade de cor".

-Establécese un código formado por ceros e uns para cada cor.

-Por exemplo, se temos unha profundidade de cor de catro cores (branco, vermello, amarelo e negro), poderíamos establecer este código: 00-Branco 01-Vermello 10-Amarelo 11-Negro

Empregamos así dous bits para cada código e a imaxe representárase mediante o código da cor de cada punto da imaxe de forma ordenada.

Neste exemplo, a cantidade de información que supón unha imaxe ven dada por:

$$\text{Tamaño Total} = \text{bits para cada cor} * \text{resolución horizontal} * \text{resolución vertical}$$

Naturalmente nunha imaxe non só se graban os píxeles, senón tamén o tamaño da imaxe, o modelo de cor, etc. De aí que representar estes datos sexa tan complexo para o ordenador (e é tan complexo entendelo nós).

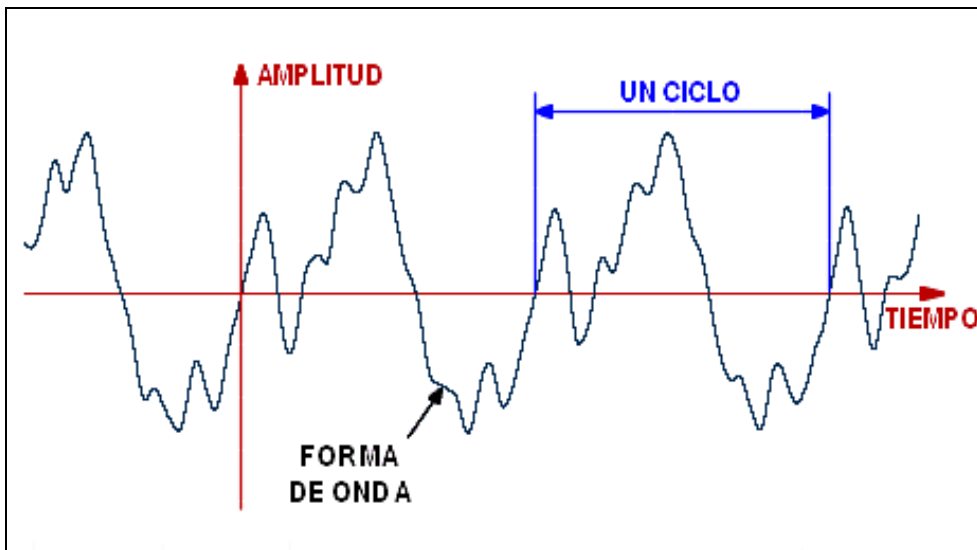
Existen outro tipo de imaxes chamadas **imaxes vectoriais**, vectorizadas ou escalables onde a imaxe se constrúe a partir de vectores, que son obxectos formados matematicamente como segmentos, polígonos, arcos e outras figuras, almacenándose distintos atributos matemáticos dos mesmos (por exemplo, un círculo branco se define pola posición do seu centro, o tamaño do seu radio, o grosor e a cor da liña e, por último, a cor de recheo).

Un dos maiores atractivos dos gráficos vectoriais é poder ampliar o tamaño dunha imaxe a vontade sen sufrir o efecto do escalado que sufren os gráficos rasterizados. Tamén permiten mover, estirar ou transformar imaxes de forma moi sinxela, o que o levou ao uso moi estendido na xeración de imaxes en tres dimensións tanto dinámicas como estáticas. Na actualidade todos os gráficos vectoriais son facilmente traducibles a gráficos rasterizados.

Audio

No caso do audio (ou son), que é, por natureza, información analóxica ou continua, trátase dunha onda que transcorre durante un tempo. Para almacenar o son haberá que representar, dalgunha forma, esa onda para que despois se poida mandar o sinal axeitado a dispositivos de saída de audio.

A onda de son adoita ter un aspecto similar ao seguinte:



Para gardar o son tómase o valor da amplitude (altura da onda) en binario cun número de bits, chamado **calidade da mostraxe**, que determinará a calidade do mesmo, habitualmente 16 ou 32 bits. Esta operación faise cada certo tempo, tomándose un número de puntos por segundo chamada **frecuencia** que se mide en hercios (puntos por segundos). Son frecuencias habituais as da telefonía (8kHz, 8.000 puntos por segundo), a radio (22 kHz) ou o CD (44,1 kHz).

Para reproducir o son reconstrúese a onda a partires dos valores almacenados. É evidente que a maior frecuencia (máis cerca se atopan os puntos uns dos outros), a onda formada parecerase máis á orixinal.

Hai que ter tamén en conta que a música pode ser son **mono**, cunha soa canle de son; son **estéreo**, o que implica dúas ondas ou dúas canles, un para cada altofalante; ou son **Dolby** que, a súa versión máis estándar, o Dolby 5.1 emprega 6 altofalantes e, polo tanto, 6 canles de son.

Así, o tamaño dun son almacenado viría dado por:

$$\text{Tamaño} = \text{Número de canles} * \text{Calidade da mostraxe} * \text{Frecuencia} * \text{Duración}$$

Por exemplo, se temos 30 segundos de son estéreo cunha calidade de 32 bits e cunha frecuencia de 22KHz, o tamaño que ocupará será:

$$\text{Tamaño} = 2 * 32 * 22.000 * 30 = 5,035 \text{ MiB}$$

Vídeo

Por último, almacenar vídeo é moi sinxelo se partimos da base de que é unha representación de imaxes, ou **frames**, e son nun tempo determinado.

Unha película non é máis que unha serie de cadros despregados uns tras outros para crear unha ilusión de movemento. O ritmo de imaxes por segundo é unha característica do vídeo chamada **frames por segundo (fps)**.

Imos ver un exemplo cun vídeo de 30 segundos gravado a unha resolución 640x480 e 32 bits de profundidade de cor a 30fps con son en estéreo de 32 bits de calidade con frecuencia de 22kHz. Canto ocupa?

Comezamos calculando o tamaño das imaxes:

$$\text{Tamaño da imaxe} = 640 * 480 * 32 = 9.830.400 \text{ bits} = 1.200 \text{ kiB}$$

$$\text{O número total de imaxes será: } 30 \text{ (fps)} * 30 \text{ (segundos)} = 900 \text{ imaxes}$$

$$\text{A secuencia ocupará: } 1.200 \text{ kiB} * 900 \text{ (imaxes)} = 1.080.000 \text{ kiB} = 1.054,68 \text{ MiB}$$

Por outro lado calculamos o tamaño do son:

$$\text{Tamaño do son} = 2 \text{ (estéreo)} * 32 \text{ (bits)} * 22.000 \text{ (Hz)} * 30 \text{ (segundos)} = 42.240.000 \text{ (bits)} = 5,03 \text{ MiB}$$

O tamaño total do vídeo sería a suma das dúas cantidades:

$$\text{Tamaño total} = 1.054,68 \text{ MiB} + 5,03 \text{ MiB} = 1,034 \text{ GiB}$$