

# Gràfics digitals

David Gómez Fontanills  
Alba Ferrer Franquesa

PID\_00191343



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-Compartir igual (BY-SA) v.3.0 España de Creative Commons. Se puede modificar la obra, reproducirla, distribuirla o comunicarla públicamente siempre que se cite el autor y la fuente (FUOC. Fundació per a la Universitat Oberta de Catalunya), y siempre que la obra derivada quede sujeta a la misma licencia que el material original. La licencia completa se puede consultar en: <http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.ca>

# Índice

<b>1. Introducción a los gráficos digitales.....</b>	<b>5</b>
1.1. Imagen digital frente a imagen analógica .....	5
1.1.1. Introducción: gráficos codificados .....	5
1.1.2. Cambios para los grafistas .....	7
1.1.3. Dispositivos de entrada y salida de datos .....	7
1.2. Digitalización .....	8
1.2.1. Escáner .....	8
1.2.2. Cámaras digitales .....	9
1.3. Dispositivos de entrada para creación de gráficos .....	11
1.4. Dispositivos de salida de gráficos .....	12
1.4.1. Impresora .....	12
1.4.2. Pantalla del monitor .....	13
<b>2. Gráficos de mapa de bits.....</b>	<b>15</b>
2.1. Una retícula de píxeles .....	15
2.2. Resolución de la imagen .....	15
2.3. Resolución de pantalla .....	18
2.4. Profundidad de color .....	20
2.5. Herramientas de edición de gráficos de mapa de bits .....	22
2.5.1. Herramientas de dibujo .....	23
2.5.2. Herramientas de selección .....	25
2.5.3. Filtros algorítmicos .....	29
2.5.4. Ajustes de imagen .....	29
<b>3. Gráficos vectoriales.....</b>	<b>31</b>
3.1. Dibujo basado en objetos .....	31
3.1.1. Objetos Bézier .....	32
3.1.2. Bézier en el software gráfico .....	33
3.1.3. Trabajo de precisión con las curvas Bézier .....	35
3.2. Propiedades de los objetos vectoriales .....	36
3.2.1. Propiedades de la línea o filete .....	37
3.2.2. Transformaciones de los objetos vectoriales .....	38
3.2.3. Composición de gráficos vectoriales .....	40
3.3. Mano alzada: dibujar vectores sin nodos ni manejadores .....	41
3.4. La resolución de las imágenes vectoriales .....	42
3.5. Cuándo usar gráficos vectoriales .....	43
<b>4. Formatos gráficos.....</b>	<b>44</b>
4.1. Codificar gráficos .....	44
4.1.1. Tipos de formato .....	45
4.2. Formatos para gráficos vectoriales y metaficheros .....	45
4.2.1. Formatos para artes gráficas .....	46

4.2.2.	Formatos para Internet .....	47
4.3.	Formatos para mapa de bits .....	48
4.3.1.	TIFF, el formato de impresión .....	49
4.3.2.	Formatos para la web: GIF, JPEG y PNG .....	49
4.3.3.	Sistemas de compresión .....	51
4.3.4.	La polémica LZW .....	53
4.4.	Software y formatos .....	54
4.5.	Información rápida sobre formatos .....	55
<b>5.</b>	<b>Software de creación y tratamiento gráfico.....</b>	<b>58</b>
5.1.	Software bitmap y vectorial .....	58
5.1.1.	Entornos de trabajo vectorial .....	59
5.1.2.	Entornos de trabajo bitmap .....	60
5.1.3.	Confluencias .....	60
5.2.	Software orientado a impresión y orientado a la web .....	62
5.2.1.	Software orientado a impresión .....	63
5.2.2.	Software orientado a web .....	64
5.3.	Software propietario y <i>open-source</i> .....	65
5.3.1.	El software .....	65
5.3.2.	La aparición del software propietario .....	66
5.3.3.	El proyecto GNU .....	66
5.3.4.	La licencia GPL .....	67
5.3.5.	El movimiento del software libre .....	67
5.3.6.	Software libre para gráficos .....	68
<b>Bibliografía</b> .....		<b>69</b>

# 1. Introducción a los gráficos digitales

## 1.1. Imagen digital frente a imagen analógica

### 1.1.1. Introducción: gráficos codificados

La digitalización de datos introduce un cambio significativo en sus posibilidades de manipulación y tratamiento. También trabajar con gráficos digitales es significativamente distinto a hacerlo con gráficos analógicos. Una imagen analógica es, por ejemplo, la que podemos trazar con un lápiz o un pincel sobre un papel. La llamamos analógica desde la perspectiva de lo digital. Es analógica porque no ha sido generada por un proceso codificado ni se almacena de forma codificada.

#### Sketchpad de Ivan Sutherland

Los sistemas de diseño asistido por ordenador, también conocidos por sus siglas inglesas CAD, son sistemas de software y, a veces, hardware, muy avanzados. Pero hace mucho tiempo que rondan por la cabeza de los programadores, no son algo nuevo. He aquí al abuelo de los CAD actuales, el sistema *Sketchpad*.

El Sketchpad fue desarrollado en el año 1963 por Ivan Sutherland como parte de su tesis doctoral. Introducía algunos avances que tardarían en desarrollarse como herramientas de uso común, pues ya contaba con una interfaz gráfica de usuario, GUI, muchas décadas antes de que se extendieran por todo el planeta informático.

El sistema Sketchpad fue creado en el Laboratorio Lincoln del MIT sobre un ordenador TX-2, una de las mejores máquinas de entonces pues contaba con 320Kb de memoria base y 8Mb de memoria externa en forma de cintas magnéticas. En cuanto a dispositivos gráficos, estaba dotado con un monitor de siete pulgadas 1024×1024, además de un puntero óptico y un remedo de ratón.



Fotografía desconocida. Esta imagen se reproduce acogiéndose al derecho de cita o reseña (art. 32 LPI), y está excluida de la licencia por defecto de estos materiales.  
Sketchpad; un software de dibujo de gráficos de ingeniería para el ordenador TX-2. Creado en 1963 por Ivan Sutherland como parte de su doctorado en el Massachusetts Institute of Technology, fue la primera interfaz gráfica de usuario para ordenadores.

Durante años, siglos, milenios, los gráficos han sido principalmente analógicos. Una imagen digital es una imagen codificada a través de dígitos (símbolos arbitrarios que no guardan una analogía con el referente). Antes de la informática ha habido imágenes codificadas de esta forma como, por ejemplo, el tejido de *jacard* que usaba tarjetas perforadas para los telares (y que son el antecedente directo de las tarjetas perforadas de los primeros ordenadores) o el bordado a punto de cruz que usa retículas y símbolos para codificar los colores.

Pero cuando se habla de gráficos digitales se suele hacer referencia a la tecnología informática. Ésta codifica la información a través de un sistema numérico **binario** basado sólo en ceros y unos. Su unidad básica es el **bit** (del inglés *binary digit*) que puede tener uno de los dos valores. Agrupando los bits en paquetes de ocho (**bytes**) y asignando a cada uno de ellos un valor numérico se empieza una cadena de sofisticación creciente a través de la cual es posible almacenar cualquier información. En consecuencia lo que tenemos es un sistema de codificación de la información que es independiente del soporte (puede almacenarse en cualquier soporte que pueda dar ceros y unos, sea magnético, óptico o de cualquier otro tipo) y la información puede ser copiada con total fidelidad. A diferencia de la imagen analógica, en la imagen digital la copia es igual al original porque se copian números y no hay "ruido" que interfiera en el proceso.

### **Bit, la base del sistema binario**

En informática, "bit" es la contracción de la expresión *binari digit* (**dígito binario**). Pero a la vez significa en inglés "pedacito" o "porción". El bit representa una variable binaria en la que cada dígito toma el valor de 0 o el valor de 1. Dentro del ordenador, constituye, por lo tanto, la unidad más pequeña en la que se puede codificar la información (palabra, número, instrucción). En consecuencia, cualquier información se mide por el número de bits necesarios para su representación en un sistema binario.

El sistema binario trabaja sobre la base de 2, es decir, utiliza únicamente 2 cifras para expresar cualquier cantidad. Estas cifras son los dos valores que puede tener un bit: 0 y 1. El sistema en el que trabajan los ordenadores es binario. A nivel electrónico la máquina considera:

- **valor 0** = transmisión de una señal eléctrica de tensión entre 0 y 0,0 voltios.
- **Valor 1** = transmisión de una señal eléctrica de tensión entre 0,8 y 4,5 voltios.

Del bit derivan el resto de unidades de medición. Así el elemento byte, conjunto de 8 bits, seguido de sus múltiplos, el kilobyte, el megabyte, el gigabyte, el terabyte. El aumento de la escala viene determinado por la potencia de 2 más próxima a 1.000 que expresa la fórmula:

$$2^{10} = 1.024$$

Cada unidad mantiene esta relación con la anterior y la siguiente:

1 bit = 0/1  
1 byte = 8 bits  
1 kilobyte = 1.024 bytes  
1 Megabyte = 1.024 kilobytes  
1 Gigabyte = 1.024 megabytes

### 1.1.2. Cambios para los grafistas

A mediados de la década de los ochenta los ordenadores personales y el software de autoedición entraron en los estudios de diseño transformando por completo la forma de trabajar de los profesionales de las artes gráficas.

Los programas de autoedición –o publicación de escritorio, conocida en inglés como *desktop publishing* (DTP)– eran programas orientados a la maquetación de libros y revistas o al diseño de carteles y folletos. El valor añadido se daba principalmente por el hecho de que lo que se veía en pantalla se aproximaba bastante al resultado final impreso y permitía modificar un diseño con facilidad.

La dimensión conceptual del cambio la expresa muy bien Elena Fuenmayor de la siguiente forma:

"...el sólo hecho de haber transformado las rutinas de trabajo en tantos sentidos: visualizar las ideas al instante, recombinar los elementos de la página con mayor libertad, recomponer en varias soluciones diferentes una misma idea, experimentar personalmente con otras especialidades afines al diseño (...) ha influido indiscutiblemente en el replanteamiento de la naturaleza de la página. Replanteamiento que no se limita a la investigación de nuevas soluciones formales, sino que llega al terreno de lo simbólico al ser capaz de generar un lenguaje visual original."

Elena Fuenmayor (2001). *Ratón, ratón...* (ed. original 1997). Barcelona: Gustavo Gili.

Durante la década de los noventa la publicación en formato digital fue cobrando fuerza. El soporte final ya no es sólo el papel; el gráfico será visto en pantalla (del ordenador, del móvil, etc.). Especialmente con la expansión de Internet, el ordenador, además de instrumento de creación, pasa a ser el dispositivo donde se visualiza lo publicado en la red.

Diseñar algo para que sea visto en pantalla y a través de un dispositivo informático es distinto, técnica y conceptualmente, a hacerlo para ser impreso en papel. En pocos años el trabajo del grafista ha sufrido dos grandes transformaciones que le han aportado flexibilidad y nuevas posibilidades.

### 1.1.3. Dispositivos de entrada y salida de datos

A grandes rasgos un ordenador suele dividirse en tres partes:

- 1) **Unidad Central de Proceso** (CPU en sus siglas inglesas) que incluye el procesador y el resto de circuitos y chips que lo complementan.
- 2) **Memoria de acceso directo** (RAM) y **de almacenamiento** (módulos de memoria con circuitos integrados, disco duro y diversidad de discos removibles).
- 3) **Dispositivos de entrada y salida de datos** (además de sus tarjetas controladoras).

Resulta obvio que estos últimos son los que permiten a una persona trabajar (interactuar) con el ordenador. Unos son la vía de entrada de información e instrucciones y los otros la forma de verificar el trabajo realizado y visualizar resultados.

En referencia al trabajo con gráficos digitales los **dispositivos de entrada** son aquellos que permiten digitalizar gráficos analógicos (escáner, cámaras digitales, tarjetas digitalizadoras A/V) y aquellos que permiten generar imágenes en el propio ordenador (ratón, teclado, tableta gráfica). Los principales **dispositivos de salida** de datos son la impresora y el monitor.



Fuente: [www.sxc.hu](http://www.sxc.hu). Esta imagen se reproduce acogiéndose al derecho de cita o reseña (art. 32 LPI), y está excluida de la licencia por defecto de estos materiales.

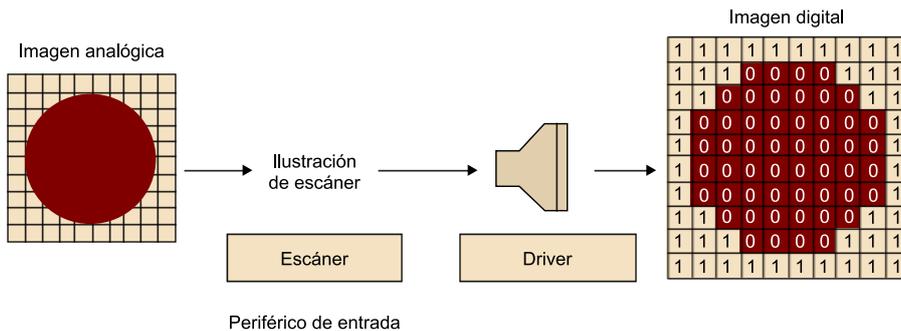
## 1.2. Digitalización

### 1.2.1. Escáner

**Digitalizar** implica transcribir estructuras analógicas a nuevas estructuras basadas en zeros y unos.

El **escáner** es uno de los dispositivos más usados para este proceso y va a ser un ejemplo útil para entender cómo puede realizarse esta transcripción.

Supongamos que se quiere digitalizar un dibujo sencillo en blanco y negro. El escáner hará un barrido con luz de la imagen punto por punto. Una matriz interpretará las diferencias de luz en impulsos eléctricos que luego pasarán a código binario. Las zonas oscuras se almacenarán como ceros y las zonas claras como unos. Los datos del escáner pasan al ordenador pero este necesita tener de un programa "conductor" (*driver*) que envíe las instrucciones al dispositivo y reciba la información. Además será necesaria una aplicación con la que el usuario pueda controlar los parámetros de escán.



Fuente: adaptado de E. Fuenmayor (2003).

Si la imagen original tiene distintos valores de gris que se quieren codificar, en vez de un **bit** de información para cada punto se usarán ocho, agrupados en un *byte* (lo que nos dará hasta 256 tonos distintos). La matriz transformará las diferencias lumínicas en diferencias de voltaje a las que luego se asignará un valor numérico. En la digitalización de imágenes en color el escáner hará tres barridos (uno para cada uno de los colores primarios) con filtros que se encargan de dejar pasar sólo el espectro de luz que corresponda. En cada uno de ellos se usarán ocho bits, codificando las diferencias de intensidad. La suma de los tres colores compone la imagen en color en la que cada punto tiene 24 bits (8+8+8) de información.

Así pues, cada punto se puede codificar con un solo bit (para la imagen en blanco y negro), con ocho (para la imagen con distintos valores de gris, escala de grises) o con veinticuatro (para la imagen en color, color real). A cada uno de estos puntos, la unidad básica de la imagen digital obtenida, lo llamamos **píxel** (contracción de *picture element*).

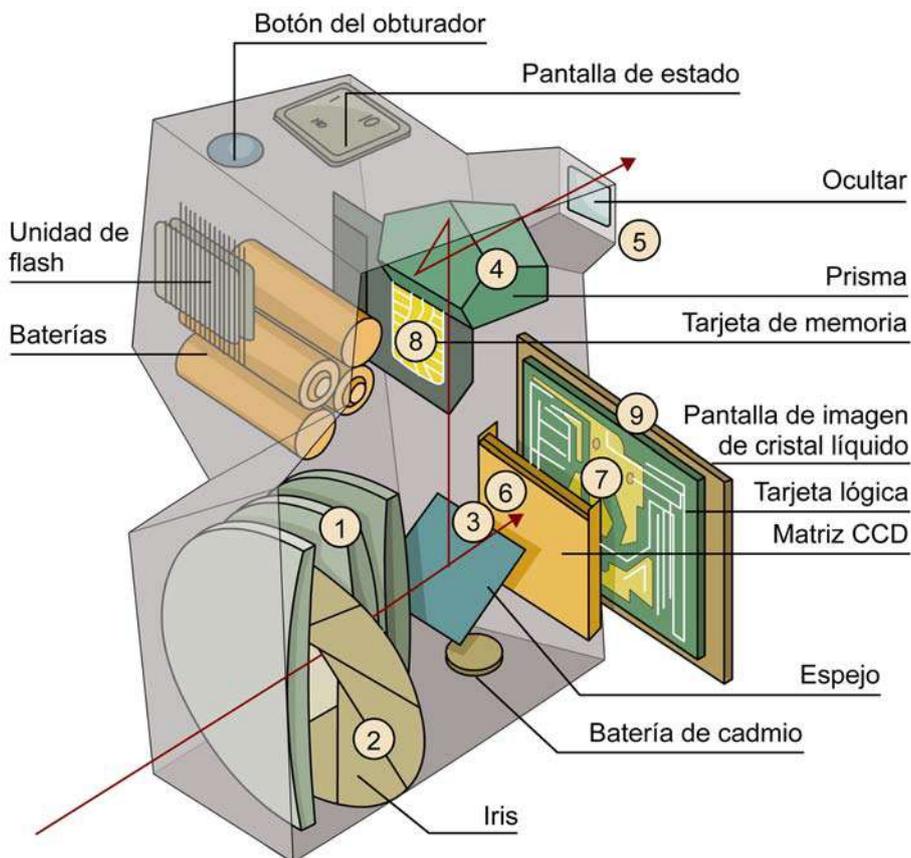
### 1.2.2. Cámaras digitales

Las **cámaras digitales** son otro dispositivo de digitalización muy común. Con ellas fotografiamos o grabamos imágenes del mundo real y luego las copiamos al disco duro del ordenador. En la fotografía química tradicional la luz que atraviesa la lente queda impresionada en una película (con tres capas superpuestas de productos químicos que reaccionan a distintas longitudes de onda). En las cámaras digitales, la luz que entra por la lente es captada por una retícula de puntos de fósforo que convierten en corriente eléctrica la luz recibida.

Estos captosres fotosensibles dan mayor carga eléctrica cuanto más intensa sea la luz y/o más prolongada la exposición. Estas cargas se transfieren a una unidad de memoria registrando digitalmente la imagen.

Para obtener una imagen en color se utilizan filtros, de forma que cada punto de fósforo reciba una de las tres longitudes de onda. En función del fabricante (Sony, Nikon, Casio, Minolta, Panasonic, Epson, HP, etc.) y de si son para imagen en movimiento (vídeo) o imagen estática (fotografía) se utiliza una cinta de vídeo digital o alguna otra unidad de almacenamiento como discos duros o tarjetas de memoria que varían según el fabricante. Para la entrada de los datos al ordenador se utilizan conectores como *Firewire* (nombre comercial de IEEE-1394, también conocido como *iLink*) o USB. También se utilizan lectores específicos para cada tipo de tarjeta (lectores de tarjetas).

El esquema siguiente explica cómo se obtiene una imagen con una cámara reflex digital.



Pulsando parcialmente el botón del obturador se dispara el foco automático y los mecanismos de exposición ajustando el objetivo (1) y la apertura del iris (2). En una cámara reflex, como la que se muestra en la ilustración, la luz que entra en el objetivo es desviada por un espejo (3), a través de un prisma (4), a una pequeña pantalla de visor acoplada a un ocular (5). Cuando se pulsa completamente el botón del obturador el espejo gira dejando libre el camino de la luz y se activa la matriz CCD (6). La lectura de la matriz CCD es procesada por una placa lógica (7), donde se comprime. La imagen procesada es enviada a la tarjeta de memoria (8) para su almacenamiento y a la pantalla de la imagen de cristal líquido (9). Después de unos segundos la cámara está ya preparada para tomar otra imagen.

### 1.3. Dispositivos de entrada para creación de gráficos

Si se va a crear la imagen directamente en el ordenador (**imagen sintética**) se usarán programas específicos de creación de gráficos. Pero para entrar datos a éstos programas son necesarios también dispositivos de entrada.

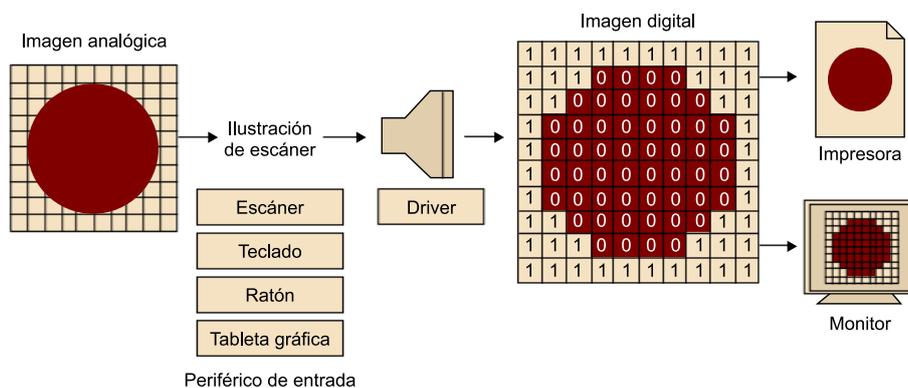


Teclado, ratón y tableta gráfica.  
Fuente: www.sxc.hu. Esta imagen se reproduce acogiéndose al derecho de cita o reseña (art. 32 LPI), y está excluida de la licencia por defecto de estos materiales.

Uno de los primeros dispositivos que se añadió al ordenador es el **teclado**. Su propósito inicial era introducir instrucciones alfanuméricas (texto y números) aunque el sistema operativo y el software le sacan el máximo partido, siendo fuente de entrada de todo tipo de instrucciones.

Desde la década de los ochenta los ordenadores personales (y especialmente los destinados al diseño gráfico) incorporan el *mouse* –el **ratón**. Se trata de un dispositivo que traduce los movimientos de una bola sobre una superficie en movimiento de un "cursor" en pantalla. Es un elemento clave para la interacción con la interfaz gráfica de usuario (señalar, seleccionar, clicar, arrastrar y soltar) y del manejo de programas de gráficos. Posteriormente la bola se ha sustituido por tecnología óptica.

Para trabajos de ilustración las **tabletas gráficas** mejoran sensiblemente la forma de dibujar respecto al mouse. Constan de un "lápiz" que se desplaza sobre una superficie que recoge los puntos de presión ejercidos sobre ella. La mayoría de tabletas son sensibles a las diferencias de presión permitiendo que el software de gráficos las traduzca en diferencias de grosor de línea.

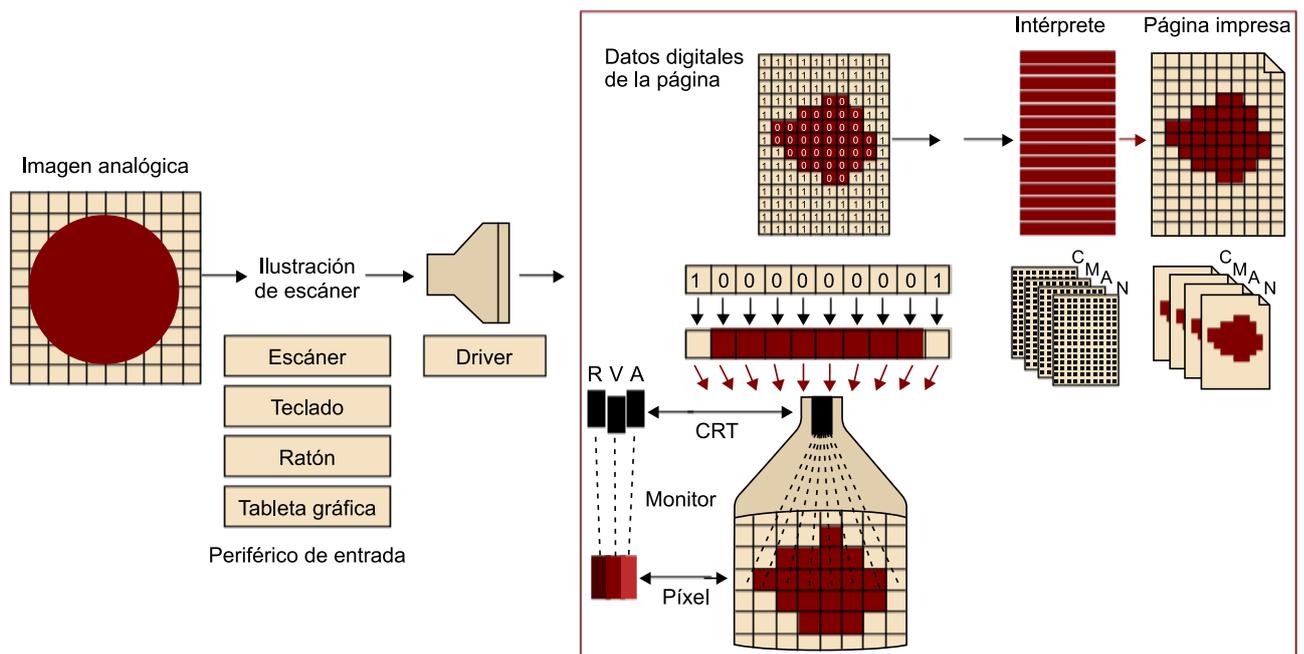


Fuente: adaptado de E. Fuenmayor (2003).

## 1.4. Dispositivos de salida de gráficos

### 1.4.1. Impresora

En lo referente a gráficos, los principales dispositivos de salida de datos son, como se ha dicho, la **impresora** y la **pantalla del monitor**. En este caso el proceso es el inverso a la entrada de datos. La información digital tiene que transcribirse en información analógica: tinta sobre papel a través de las impresoras y ondas luminosas en el caso del monitor.



La **impresora** es uno de los primeros dispositivos de salida que se incorporó a los ordenadores. Los primeros ordenadores que recibían los datos en tarjetas perforadas en la década de los cincuenta del siglo XX ya imprimían sus resultados en papel. Desde la década de los ochenta tenemos "impresoras de escritorio" conectadas directamente al ordenador, a través de uno de sus puertos de entrada o de una red, que nos permiten obtener rápidamente una imagen impresa del trabajo realizado. Las tecnologías de impresora (matriciales, inyección de tinta, láser, etc.) están en constante evolución y actualmente la calidad de la mayoría de ellas es buena, variando en rapidez y prestaciones.

Las impresoras a color usan cuatro colores básicos (cian, magenta, amarillo y negro) para componer la imagen a través de algún tipo de trama. También han evolucionado los *plóters* o impresoras de gran formato, convergiendo en parte con las tecnologías de escritorio, mejorando en calidad y admitiendo todo tipo de soportes de impresión: papel, plásticos, textil. Otro dispositivo de "impresión" es la filmadora de fotolitos que traduce la información digital en impresión química sobre acetato. Éste fue uno de los elementos clave del cambio en la forma de trabajar en el sector de las artes gráficas ya que el fo-

tolito obtenido directamente del ordenador sirve para insolar la plancha de impresión (*offset*) o la pantalla de estampación (serigrafía), evitando el paso por la fotomecánica convencional. Actualmente ya existen dispositivos que insolan directamente sobre la plancha (*direct to plate*) sin necesidad de fotolitos. Destinadas a sustituir la imprenta tradicional *offset* en tirajes cortos (200, 500, 1000) existen impresoras basadas en las tecnologías de escritorio capaces de imprimir múltiples ejemplares con calidad y rapidez.



### Evolución de los procesos de impresión

El esquema muestra cómo se han reducido o integrado los procesos para obtener un producto acabado impreso. Se ha simplificado esta evolución en tres fases, desde la regrafía previa a la digitalización, pasando por un proceso basado en la filmación de fotolitos desde el ordenador y finalmente un proceso de impresión directa de múltiples ejemplares sobre papel.

#### 1.4.2. Pantalla del monitor

Otro dispositivo de visualización es la **pantalla del monitor**. Fue un dispositivo clave en la aparición del ordenador personal, en la mejora de la interacción persona/máquina y, evidentemente, en la posibilidad de creación y tratamiento de gráficos por ordenador.

Para visualizarse en pantalla la información digital tiene que generarse una señal luminosa. Por ello, cualquiera que sea su tecnología, los monitores tienen una retícula de puntos, equivalentes al píxel, que puedan emitir luz. El conjunto de píxeles iluminados compone la imagen. Las tecnologías más frecuentes son el tubo de rayos catódicos (CRT en sus siglas inglesas) y las pantallas de matriz activa (TFT de *Thin Film Transistor*).

Los **monitores CRT** tienen en su interior un tubo de rayos catódicos con varios cañones que disparan electrones contra el material fosforescente de la pantalla. Éste está formado por partículas de fósforo que al recibir el impacto del electrón emiten radiación visible. La distinta composición del fósforo de la pantalla determinará la longitud de onda y en consecuencia el color percibido. Cada punto (píxel) de la pantalla tiene partículas de 3 tipos que emitirán luz correspondiente al azul, rojo o verde. Para encenderlas se utilizan simultáneamente 3 haces de electrones. La intensidad de cada haz determina la intensidad (amplitud) de la onda emitida. Como las partículas sólo emiten luz

durante un corto período, los haces de electrones barren la pantalla varias veces por segundo encendiéndolas. La alta frecuencia del "refresco" de la pantalla permite que se perciba una imagen estable y continua. Las principales desventajas de este tipo de monitor son que ocupan mucho espacio en la mesa, por el espacio necesario para el tubo de rayos catódicos, que consumen mucha energía y que emiten radiaciones perjudiciales para el usuario. Por ello se han buscado tecnologías alternativas que en buena parte las han sustituido.

En los **monitores TFT** no hay barrido de una superficie fosforescente. Cada punto de la pantalla es una pequeña celda fija de cristal líquido que deja pasar parte de la radiación visible generada por lámparas de luz blanca en la parte posterior. Cada celda corresponde a un píxel y recibe una carga variable de un electrodo alimentado por un transistor. Hay tres transistores para cada celda (uno para cada color) y en función de la carga recibida las moléculas del cristal líquido varían su ondulación girando los rayos de luz. Un filtro polarizado impide que pase la luz que no está alineada con el mismo.

## 2. Gráficos de mapa de bits

### 2.1. Una retícula de píxeles

En términos generales existen dos formas de codificar digitalmente la información gráfica, que dan lugar a dos tipos de gráficos: los llamados **mapa de bits** y los **gráficos vectoriales**.

En los gráficos de mapa de bits la información se almacena punto por punto, dividiendo la imagen en casillas cuadradas formando una especie de retícula. Cada casilla es un píxel y de él se conoce su posición y se le asigna un valor correspondiente a su color. No se registra geoméricamente la forma de la imagen, ésta será percibida cuando se muestre cada píxel en la retícula. Se trata del mismo sistema que hemos explicado para la digitalización del gráfico a través los escáneres y las cámaras digitales. El número de bits usados para la codificación del gráfico determina el número máximo de colores que puede tener cada píxel. El tamaño de la retícula determina la calidad (resolución) de la imagen.

#### Mapa de bits, también llamados...

La terminología en el campo de los gráficos digitales a veces es diversa para un mismo concepto. Otras formas de referirse a las imágenes de mapa de bits son:

- **Imagen rasterizada:** por el inglés *raster*, referido a la rejilla rectangular formada por píxeles.
- **Imagen matricial:** por estar formada por una "matriz" de píxeles.
- **Pixmap:** por contracción de la expresión inglesa *pixel map*.

### 2.2. Resolución de la imagen

La **resolución de una imagen** es el número de unidades gráficas por unidad de superficie.

En una imagen impresa es el número de puntos por unidad de superficie. En una imagen digital es el número de píxeles por unidad de superficie.

Como se ha explicado, un **píxel** (abreviatura de *Picture Element*) es la unidad básica de una imagen digital de mapa de bits. Distingue el elemento mínimo sobre el que podemos definir el color. No tiene nada que ver con alguna medida del mundo físico. Podemos decir que el píxel es una unidad de división de la imagen sin un tamaño real concreto.

Con **unidad de superficie** sí nos referimos a medidas del mundo físico. Podemos usar los centímetros pero, por influencia de la cultura anglosajona, solemos usar las pulgadas. Una pulgada es pues una unidad de medida física del sistema imperial británico de medidas que corresponde a 2,54 centímetros (o 25,4 milímetros) del sistema métrico decimal.

La resolución de una imagen digital suele expresarse en píxeles por pulgada (abreviado como ppp) –*pixels per inch* (ppi) en inglés–. ¿Cuántos píxeles hay en ese recuadro de una pulgada? Depende de la resolución de entrada dada: escaneado, cámara digital, imagen creada o modificada en el programa de gráficos. Cuantos más píxeles haya en una pulgada, más pequeños serán y mejor será la calidad y la definición (detalle) de la imagen resultante (mayor resolución).

### Resolución e impresión

Del mismo modo que procuramos que una imagen no aparezca en su salida por pantalla pixelada, resulta igualmente importante que tampoco se obtenga su impresión final pixelada o falta de nitidez. Este objetivo, sin embargo, resulta más crítico en impresión porque debemos controlar varios tipos de resolución para garantizar una impresión óptima. Así, para una impresión de un original digital, tal como se contempla actualmente en las artes gráficas, debemos gestionar tres tipos diferenciados de resolución, que no son excluyentes sino dependientes entre sí:

- **Resolución de entrada o digitalización:** corresponde al número de píxeles por pulgada de una imagen, establecida al ser captada por un dispositivo de digitalización como el escáner o la cámara fotográfica digital. La unidad de resolución de este parámetro son los píxeles por pulgada (ppp). Esta resolución corresponderá con la resolución digital de la imagen en su tratamiento a través de una aplicación para la edición gráfica.
- **Resolución de filmación:** concepto bastante complejo, y únicamente necesario para la impresión convencional y por tanto, excluyente para la impresión digital, que fundamentalmente parametriza los puntos (conocidos como *spots*) por pulgada que puede exponer un dispositivo de filmación al generar el fotolito o la impresora para impresión offset, rotográfica, flexográfica o serigráfica. Para diferenciar esta resolución de la digital que utiliza la unidad ppp, acompañamos el valor de la resolución de filmación de la expresión anglosajona *dpi* (*dots per inch*, es decir, puntos por pulgada). Esta resolución determinará, en definitiva, si es posible o no filmar a la lineatura de impresión final deseada.
- **Lineatura de impresión:** corresponde al número de puntos de impresión por unidad métrica cuadrada del soporte y se mide mediante la unidad lpp (líneas por pulgada) o lpc (líneas por centímetro). Así 150 lpp corresponden a 60 lpc. A mayor número de líneas de puntos por pulgada o centímetro, imprimiremos más puntos, lógicamente de menor diámetro, con lo que la impresión ganará en detalle, definición y rango cromático. Sin embargo, la lineatura máxima, en lpp o lpc, estará limitada tanto por las condiciones de máquina y soporte, como por la capacidad de exposición de la filmadora (dpi). Tal y como hemos comentado anteriormente, la lineatura, en última instancia, determinará la resolución digital puesto que esta corresponderá a la lineatura multiplicada por el factor de reproducción.

Es importante conocer todas estas abreviaciones para no confundir unas con otras

### Píxeles por pulgada

Los píxeles por pulgada se refieren a la **pulgada lineal**. Por ejemplo:

Si tenemos un cuadrado con las dimensiones de 1x1 pulgada con una resolución de 120ppp tendremos: 120x120 píxeles (120x120 = 14.400 píxeles en toda el área).

Dicho de otro modo: si tenemos una imagen de 120ppp de resolución eso significa que tomando un fragmento de 1x1 pulgada de esa imagen podremos contar 14.400 píxeles en ella.

Si la imagen es de 300ppp de resolución en un fragmento de 1x1 pulgada tenemos 300 x 300 píxeles, lo que son 90.000 píxeles en esa área.

La **calidad** de una imagen digital va directamente relacionada a su resolución, ya que cuanto mayor sea la cantidad de píxeles por pulgada que presente, mayor será su calidad, pero también será mayor el espacio que requiere para ser almacenada, ya que presentará más bits de información.

Mayor cantidad de ppi = Mayor calidad = Mayor tamaño

En el ámbito de la **producción impresa**, la resolución digital a utilizar depende de varios factores:

- dimensiones de la impresión final;
- sistema de impresión, ya sea digital o convencional. Dentro de este último grupo, según sea offset, huecogrado, flexografía o serigrafía;
- soporte de impresión final.

En cuanto a las dimensiones, será necesario multiplicar la resolución de impresión correspondiente por un índice matemático, conocido como factor de reproducción. Este factor resulta del cociente entre las respectivas dimensiones finales y las del original, siempre que sean proporcionales, como sigue:

$$\begin{aligned} \text{Factor de reproducción} &= \\ \text{Altura final} / \text{altura original} &= \\ \text{Anchura final} / \text{anchura original} & \end{aligned}$$

En cuanto al sistema de impresión, si trabajamos para impresión digital podemos convenir que una resolución digital estandarizada en torno a 300 ppp puede ser suficiente (factor de reproducción aparte). Si por el contrario trabajamos para impresión convencional, deberemos aplicar la siguiente fórmula genérica:

$$\begin{aligned} \text{Resolución digital} &= \\ \text{Lineatura del sistema de impresión} \times \text{factor de calidad} & \end{aligned}$$

La lineatura del sistema de impresión será la densidad de trama que utiliza tal sistema, determinada por sus condiciones de máquina y por el soporte final. Así pues, varía para cada sistema de impresión y soporte utilizado.

Por lo que respecta al factor de calidad de la fórmula, podemos establecerlo de forma genérica y para facilitar también el cálculo de la resolución en 2.

#### Ejemplo

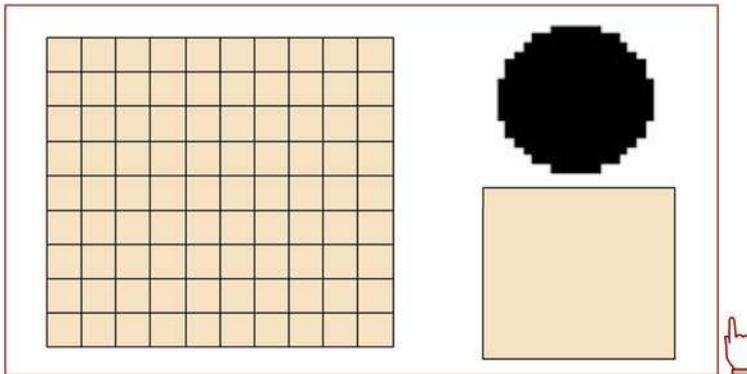
Factor de reproducción = 10 cm / 5 cm = 20 cm / 10 cm = 2 (o lo que es lo mismo, la imagen se reproducirá al 200% o el doble de grande)

Así pues, por ejemplo, para una impresión estándar en offset (sistema de impresión convencional de mayor cuota de mercado) sobre papel offset que trabaja con una lineatura de 150 lpp (líneas por pulgada o *lines per inch*), estableceremos una resolución digital de 300 ppp para el original a imprimir, como resultado de la fórmula mencionada ( $150 \text{ lpp} \times 2$ ).

Sistema de impresión	Trama	Factor de calidad	Resolución imagen
Impresión digital	-	X 2	600 ppp
Offset (papel offset)	150 lpp	X 2	600 ppp

### Actividad

A continuación dispones de una demostración interactiva que puede ayudarte a entender el concepto de **resolución**. Los cuadrados de las retículas emulan la función de un píxel. El objetivo es experimentar de manera práctica el trabajo a diferentes resoluciones y ver las limitaciones de las imágenes de baja resolución.



A partir de las siguientes cuadrículas, intentar representar el modelo original. Para ello pulsar sobre cada uno de los cuadrados para que cambien de color. Copiar píxel a píxel.

### 2.3. Resolución de pantalla

La **resolución de pantalla** determina la calidad de las imágenes que se representan en el monitor.

Las pantallas vectoriales fueron las estándar durante las décadas de los años sesenta y setenta. Representaban la imagen a través de un osciloscopio que movía un haz de electrones dibujando la imagen en pantalla. Fueron desechadas a principios de los ochenta en favor de las pantallas de barrido basadas en una retícula de píxeles que, a partir de la década de los noventa, serían progresivamente sustituidas por pantallas planas de plasma y cristal líquido basadas también en una retícula de píxeles. Por lo que, tanto las imágenes de mapa de bits como las vectoriales, tienen que representarse finalmente sobre una pantalla basada en mapas de bits.

La resolución de una pantalla viene determinada por la cantidad de píxeles que puede representar en una unidad de superficie. Así, si tomáramos un recuadro de 1x1 pulgada de un monitor y contáramos los píxeles que hay en él podríamos determinar su resolución.

Durante años la resolución de los monitores estuvo limitada. Los ordenadores Macintosh de Apple iban acompañados de pantallas que representaban 72 píxeles por pulgada. El resto de ordenadores personales –clónicos compatibles con el PC de IBM– usaban pantallas que representaban 96 píxeles por pulgada. Así el tamaño físico de una imagen digital en pantalla podía ser relativamente estable y la llamada "resolución de pantalla" a 72ppp o 96ppp se consideraba establecida. Y aún se sigue tomando como referencia, por inercia, en el ámbito del diseño.

La mejora tecnológica de la definición de los monitores ha hecho que esto deje de ser así. Los monitores –especialmente por el empuje de los dispositivos de pantalla pequeña como móviles, netbooks o PDA– han dejado atrás las resoluciones de 72ppp y 96ppp, actualmente la resolución de pantalla es un valor variable que depende del monitor que estemos usando. Hay un límite en el tamaño físico del píxel en pantalla que depende del tamaño de los puntos de fósforo (en pantallas de rayos catódicos) o de la celdilla (en pantallas de cristal líquido). Pero este límite depende de la definición del monitor.

En consecuencia no hay una relación estable entre la resolución de la imagen digital y el tamaño físico en el que esa imagen va a mostrarse en pantalla.

Una imagen, pongamos por ejemplo de 400 x 300 píxeles, veremos que ocupará la misma memoria y se mostrará al mismo tamaño en un navegador web tanto si se ha definido una resolución de 72ppp como si se ha definido de 300ppp. ¿Por qué?

Pues porque lo importante en pantalla no es la relación (ppp) píxeles/pulgada definida en la propia imagen sino las propias dimensiones que se han dado a la imagen (píxeles de ancho x píxeles de alto) y la resolución que hemos elegido en nuestro sistema operativo así como el tamaño físico del monitor donde visualizamos dicha imagen.

### **Ejemplo de resolución de pantalla**

Una imagen de 800x600 píxeles ocupará toda la pantalla si hemos elegido 800x600 como resolución en las preferencias del sistema operativo. Tanto en una pantalla de 14" como en una de 17" como en una de 19". La diferencia es que en la pantalla de 19" físicamente la imagen se verá más grande puesto que cada píxel será de mayor tamaño (ocupará más espacio en la superficie de la pantalla). Si cambiamos la resolución de 800x600 a 1024x768 píxeles o más esa imagen de 800x600 ya no ocupará toda la pantalla, sólo ocupará una parte.

Por lo tanto, lo importante en las imágenes que se muestran en pantalla no es la resolución (ppp) sino el tamaño (píxeles de alto por píxeles de ancho) que determina su calidad, ya que con más píxeles puede representarse mejor un elemento gráfico. Algo que queda claro si pensamos en una fotografía. Aún así el diseñador no puede controlar el tamaño físico en el que se va a mostrar ese tamaño en píxeles. Es el usuario quien, a través de la configuración de su equipo, tiene el control final de su experiencia.

## 2.4. Profundidad de color

La **calidad del color** de una imagen de mapa de bits la determina la cantidad de información utilizada para representar cada píxel. Nos referimos a la información que se guarda para cada píxel como **profundidad de color del píxel** o también **profundidad del píxel**.

Las posibilidades de codificar información de color dependen del número de bits que dediquemos para cada píxel. Así, a una imagen que disponga de mayor número de bits tendrá mayores posibilidades cromáticas pero también ello incidirá en que ocupará más memoria de almacenamiento en el ordenador, será un archivo con mayor información y mayor peso. Se dice que una imagen tiene una profundidad de color de 1 bit o de 8 bits, en función de los bits asignados a la codificación del color para cada píxel. De esta manera podemos determinar la siguiente escala:

Profundidad de color		
<b><math>2^1 = 2</math> tonos</b>	<b>1 bit</b>	Imágenes en blanco y con negro.
<b><math>2^8 = 256</math> tonos</b>	<b>8 bits</b>	Se pueden crear imágenes con una escala tonal de calidad. Usado para escala de grises. O una imagen con una representación del color limitada a una paleta de 256 tonos como máximo (color indexado).
<b><math>2^{24} = 16.777.216</math> tonos</b>	<b>24 bits o color verdadero</b>	Se dispone de 8 bits por canal de luz primaria (rojo, verde y azul). El resultado es una gama cromática aparentemente ilimitada para el ojo humano. Las imágenes de calidad deberán tener, como mínimo, esta cantidad de tonos. Se conoce también como "color real" ( <i>true color</i> ).
<b><math>2^{32} = 4.294.967.296</math> tonos</b>	<b>32 bits o color verdadero</b>	Difiere del anterior por el hecho de disponer de un canal de 8 bits para funciones gráficas avanzadas, como sería el empleo de manipulaciones especiales. También usado en el procesamiento de imágenes en cuatricromía que tienen cuatro canales (cyan, magenta, amarillo y negro) y necesitan también 8 bits por canal.

En función del contexto de trabajo tendremos que llegar a un compromiso entre la calidad que deseamos para la imagen y la cantidad de memoria que va a necesitar el archivo. Esto es especialmente importante en la creación de gráficos para Internet donde, debido al ancho de banda, reducir la memoria que ocupan los gráficos puede ser importante.



8 bits son suficientes para representar este icono que identifica el repositorio de imágenes y archivos libres del proyecto "Commons" de la Fundación Wikimedia en el que se utilizan colores planos.

© Fundación Wikimedia. Esta imagen se reproduce acogiéndose al derecho de cita o reseña (art.32 LPI) y se excluye de la licencia por defecto de estos materiales.



8 bits son insuficientes para esta fotografía.

© Chris Brown (zoonabar) 2010 – UOC 2010. Creative Commons Reconocimiento Compartitrgual 3.0 – Publicada originariamente en Flickr.



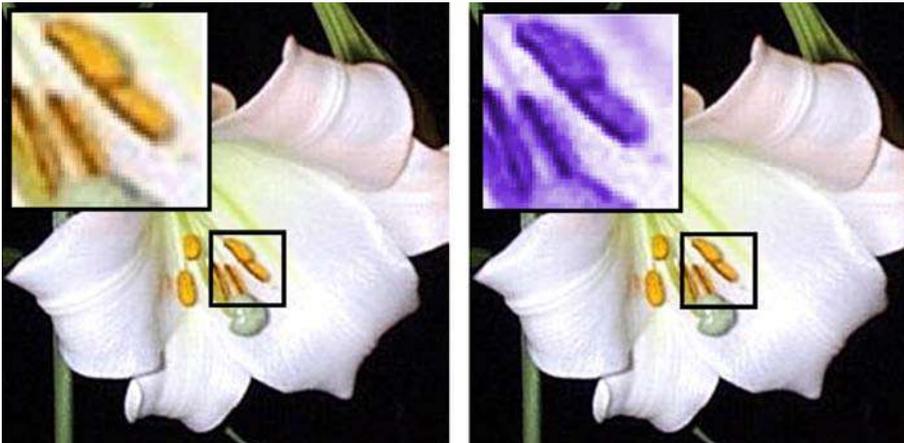
24 bits son suficientes para representar esta fotografía en buena calidad.

© Chris Brown (zoonabar) 2010. Creative Commons Reconocimiento Compartir Igual 2.0 – Publicada originariamente en Flickr.

### Ved también

Ampliad información sobre la profundidad de color con relación a los sistemas de codificación y a los modos de color en el subapartado "Codificación digital del color" del apartado "Color y diseño gráfico" del módulo "Conceptos básicos de diseño gráfico".

## 2.5. Herramientas de edición de gráficos de mapa de bits



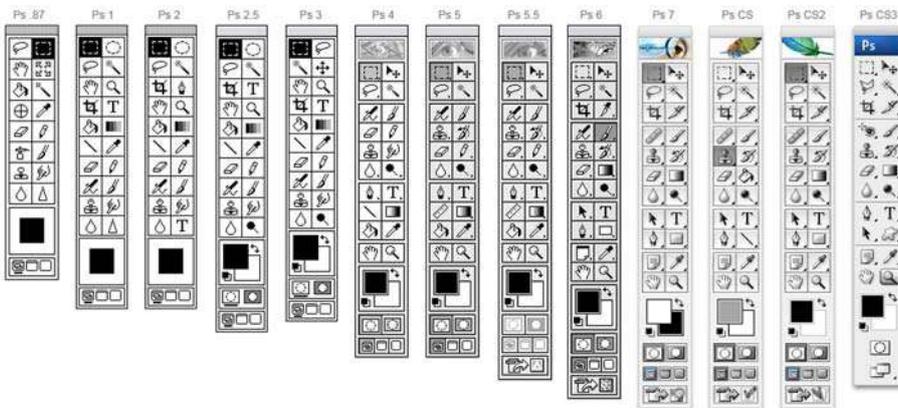
En el primer gráfico ampliamos un fragmento para ver como la imagen está constituida por píxeles. En el segundo gráfico mostramos el tratamiento aplicado sólo sobre ese fragmento. Se le han modificado los valores de color de los píxeles que lo constituyen. En este ejemplo, se ha realizado un tinte en azul que sólo afecta a los píxeles seleccionados. El programa de gráficos no puede reconocer una flor, para él son grandes cantidades de píxeles.

La mayoría de programas de gráficos usan la convención de una caja de herramientas, disponible como ventana del sistema, en la que el usuario puede escoger distintas formas de crear o manipular los gráficos.

En un gráfico de mapa de bits cualquier actuación supone la modificación de los colores de los píxeles, es decir, se modifican los píxeles que constituyen su suma, la imagen. La forma más simple de editar sería por ejemplo usar el cursor a modo de lápiz que va cambiando a negro los píxeles por donde pasa, dibujar a mano alzada. Pero evidentemente las herramientas actuales se han sofisticado mucho más y permiten conseguir efectos muy distintos y de mayor concreción. Las herramientas son además parametrizables. Ello permite que, en la mayoría de programas (es así tanto en GIMP, como en Photoshop y en Fireworks, por citar algunos), un doble clic sobre el icono de la herramienta abre una ventana en la que es posible ajustar sus parámetros para conseguir el efecto deseado.



El aspecto visual de los elementos del entorno de trabajo de un programa como el Gimp pueden adaptarse integrándose en el sistema operativo –Windows, Linux, MacOS– y al "tema" visual elegido. El kit de herramientas se mantiene estable y es muy similar al de Photoshop.



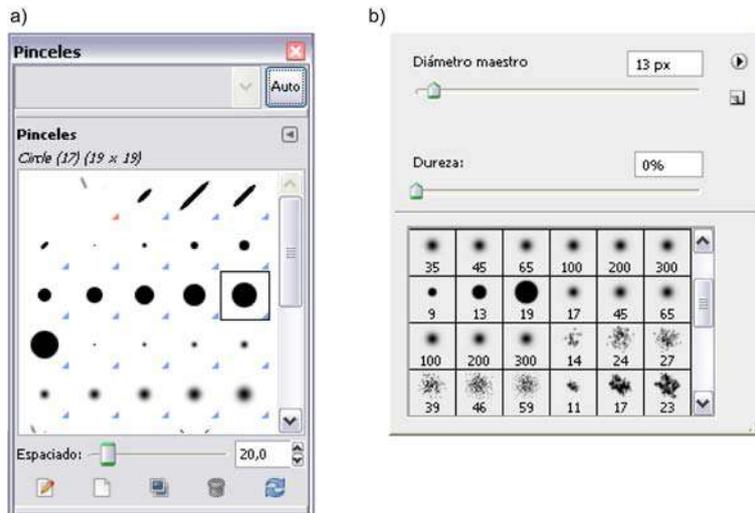
© Adobe. Esta imagen se reproduce acogido al derecho de cita o reseña (art. 32 LPI), y está excluida de la licencia por defecto de estos materiales.

Herramientas de Photoshop: evolución de la caja de herramientas de Photoshop desde sus primeras versiones. Como puede observarse, desde su primera versión de los ochenta tampoco ha variado tanto. El conjunto principal de herramientas siguen siendo las mismas, en este y en otros editores de gráficos. Pero son más parametrizables.

### 2.5.1. Herramientas de dibujo

Los programas de gráficos han evolucionado para intentar obtener resultados similares a los que se pueden conseguir con cualquier herramienta de dibujo. Se han desarrollado algoritmos que simulan el comportamiento de, por ejemplo, lápices, carboncillo, pinceles, etc.

Las herramientas más habituales son el **lápiz** y el **pincel o brocha**. Son dos herramientas muy similares que suelen diferenciarse porque el **lápiz** genera un trazo de un solo color, lo que en baja resolución produce un dentado en la línea, y el **pincel** genera trazo suavizado en sus bordes, a través de un escalado de color entre la línea y el fondo. De todas formas este suele ser un parámetro ajustable que también puede incluir la herramienta de pincel.



a) Selección de pincel en GIMP 2.6.8. b) Selección de brocha/pincel en Photoshop CS4.

Para el "lápiz" y el "pincel" pueden escogerse distintos groesos y formas que determinarán el trazo resultante. También es posible crear nuestro propio pincel. En tal caso deberá determinarse en píxeles el **diámetro** del mismo, la **dureza** (que determina el número de píxeles del suavizado en los bordes), la **forma** (puede ser circular, elíptica, cuadrada o de cualquier otra forma) y el **ángulo de inclinación**. Otros parámetros ajustables son la **opacidad**. Por debajo del 100% el trazo aumentará en transparencia y el color resultante será una mezcla del color escogido y el del "fondo". Y, por último, lo que llamamos **modo**, que determina de qué forma se aplica el trazo con relación a lo que ya hay en la imagen. Tenemos, por ejemplo, el modo "oscurecer" que hace que nuestro pincel sólo pinte sobre los colores más claros al escogido.

El pincel también puede tener otro parámetro llamado **desvanecimiento**, que hace que a medio trazo nos quedemos sin "pintura", como si se tratara de un pincel que debemos volver a mojar en el bote. Para simular este efecto la pérdida de color es progresiva.

Las herramientas de lápiz y pincel suelen estar preparadas para recibir parámetros de cambio de presión. Si disponemos de una paleta con lápiz óptico sensible a presión, el trazo puede cambiar de grueso en función de ella, obteniendo una línea modulada.

En la tabla siguiente se muestran distintos efectos en función de los parámetros de herramienta utilizados:



Otras herramientas de dibujo importantes en la edición gráfica son:

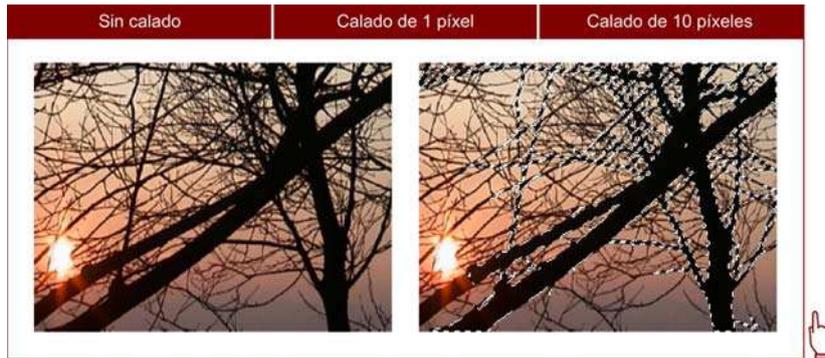
- la **goma**: que borra el dibujo y que es parametrizable en su grueso y opacidad. La goma puede borrar al color elegido como color de fondo o a transparente si hemos elegido esta opción como fondo del gráfico.
- el **aerógrafo**: que simula esta herramienta con un trazo difuminado que se superpone al color existente.
- el **bote de pintura**: que pinta del mismo color los píxeles contiguos al punto en el que se aplica.

### 2.5.2. Herramientas de selección

Las herramientas de selección son también un elemento clave en el software de mapa de bits. Permiten seleccionar una zona de píxeles para aplicarles un efecto, desplazarlos o copiarlos para luego pegarlos en otro archivo. Hay tres herramientas básicas de selección:

- la selección regular en forma de rectángulo o elipse
- el lazo o selección irregular
- la varita mágica que selecciona los píxeles con un valor de color similar

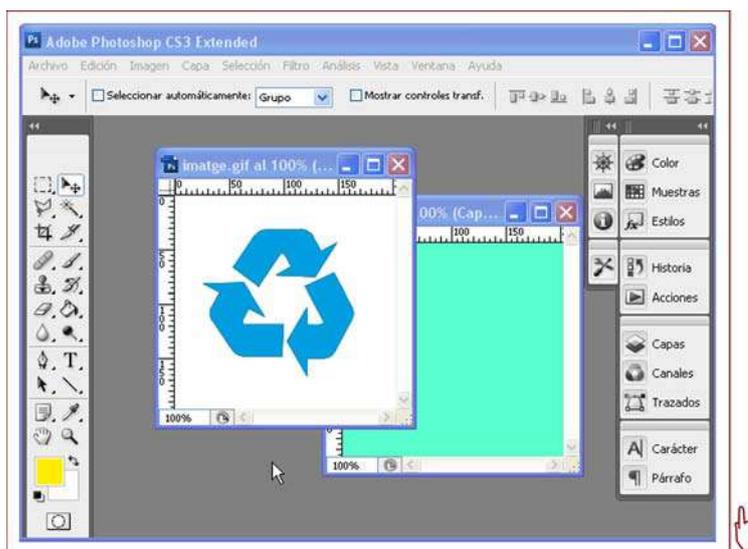
Las herramientas de selección también son parametrizables. En general los tres tipos de herramientas permiten determinar un "calado" o "suavizado" de la selección, de manera que si aplicamos color éste se integrará con el fondo en los límites de la selección.



Selección sobre la imagen original. Se puede aplicar un calado a la selección y luego copiar el contenido para pegarlo sobre otro fondo.

La herramienta de **selección regular** permite seleccionar libremente o bien de forma restringida. Se puede restringir el uso de la herramienta a un tamaño fijo o bien usar una proporción restringida. Las dos opciones son muy útiles. Si, por ejemplo, tenemos que recortar varias fotos a un mismo tamaño usaremos el tamaño fijo para encuadrarlas y luego recortar la imagen. Si la resolución de partida es distinta podemos usar la proporción restringida para seleccionar lo que queramos y luego adaptar el tamaño de la imagen.

La **varita mágica** se suele usar para seleccionar áreas del dibujo que tienen un mismo color. Por defecto se seleccionan los píxeles iguales que sean contiguos. Pero también está disponible, como parámetro o como opción del menú, la posibilidad de escoger todos los píxeles de la imagen que tengan el mismo color que el píxel que se ha clicado. Otro parámetro editable es la llamada **tolerancia** (en Photoshop y en Fireworks) o **umbral** (en The GIMP). Permite seleccionar, no sólo los píxeles del mismo color, sino los que son de un color similar. El umbral de tolerancia es ajustable, de forma que se pueda definir el escalado de colores similares a seleccionar.



**Pasos:** Selección de la varita mágica. Valor de Tolerancia: 10. Deselección de contiguo. Copiar y pegar en un nuevo documento. Programa de ejecución: Photoshop CS3.

La herramienta de **selección irregular (lazo)** es más adecuada si queremos seleccionar una imagen que contiene distintos colores. Por ejemplo, seleccionar una persona de una foto para separarla del fondo.

Los programas de gráficos suelen ofrecernos distintas opciones de lazo. El lazo por defecto nos permite seleccionar a mano alzada. Normalmente disponemos también de un **lazo poligonal** para seleccionar formas rectilíneas. Vamos clicando punto a punto y va trazando líneas rectas entre estos, dando como resultado una selección poligonal. Otra variante del lazo es el **lazo magnético**, a veces diferenciado como otra herramienta en forma de tijeras como en el GIMP. El programa reconoce los cambios de color en la imagen y magnetiza el trazado de la selección con ellos; ésto nos ayuda a seguir los contornos.



**Pasos:** Selección de la herramienta lazo magnético. Copiar y pegar en nuevo documento. Reescalar la selección copiada. Programa de ejecución: Photoshop CS3.

Una herramienta parecida al lazo es la que nos permite crear **trazados usando curvas de Bézier**. Estos trazados luego pueden convertirse en selección de la imagen. La ventaja para una persona acostumbrada a trabajar con objetos Bézier, es que puede cambiar de posición los puntos (nodos) o usar "manejadores" para definir las curvas. La utilización de la herramienta de selección mediante la pluma de Bézier es la que nos da un mayor nivel de agudeza y control en el resultado de las selecciones. Aunque al principio puede resultar algo difícil acostumbrarse a ella, es muy recomendable practicar para adquirir destreza en su uso.

Asociadas a las herramientas de selección disponemos de varias **opciones de menú** en la barra superior, también accesibles a través de un menú contextual clicando el botón derecho del ratón. Entre las más habituales encontramos:

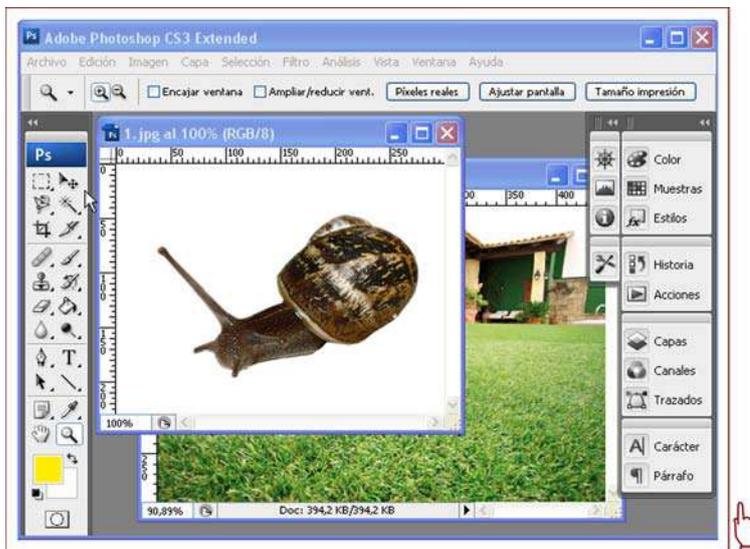
- **Seleccionar todo:** selecciona toda la imagen.
- **Deseleccionar:** elimina la selección respetando la imagen.

#### Ved también

En el apartado "Gráficos vectoriales" podéis ampliar información sobre los objetos Bézier.

- **Calar/Difuminar:** difumina los límites de la selección de la misma forma que lo haría el parámetro de herramienta equivalente. La ventaja es que podemos aplicarlo a una selección ya creada. Hay que determinar el número de píxeles de "calado". En algunos programas, como GIMP, existe además la opción contraria, "enfocar".
- **Extender/crecer:** permite ampliar la selección un número determinado de píxeles a partir de su contorno actual. En algunos programas, como GIMP y Photoshop, existe además la opción contraria, "encoger".
- **Seleccionar similar:** selecciona los píxeles del resto de la imagen que son del mismo color que los seleccionados.
- **Invertir selección:** selecciona los píxeles no seleccionados y deselectiona los seleccionados. Una especie de "negativo" de la selección actual.

Esta última opción es extremadamente útil. Por ejemplo, si tenemos una figura compleja a nivel de colores sobre un fondo de color homogéneo, la mejor opción será seleccionar el fondo usando la varita mágica (con cierto grado de tolerancia que abarque las variaciones del fondo) y luego "invertir la selección", obteniendo de ese modo una selección precisa de la figura.



**Pasos:** Selección de la varita mágica. Valor de tolerancia 10, selección del blanco. Invertir selección. Copiar y pegar en nuevo documento. Programa de ejecución: Photoshop CS3.

Otra opción muy útil es la posibilidad de **guardar la selección** en la que los datos de la misma se almacenan en un canal adicional de la imagen (**canal alfa**) que permite recuperarla. Cuando se está trabajando con una imagen con varias selecciones a las que necesitamos volver continuamente esta posibilidad es fundamental para aplicarles efectos, para copiarlas y engancharlas en otro sitio, etc.

### 2.5.3. Filtros algorítmicos

Los **filtros** son quizás lo más específico del medio digital en lo que se refiere a gráficos de mapa de bits. Se trata de algoritmos que se aplican a una imagen (o a una parte de ella) variando el color de los píxeles de modo que se consiga un determinado efecto.

Los filtros son pequeños programas que, en la mayoría de los casos, nos permiten ajustar sus parámetros y previsualizar los resultados de su aplicación. Los programas de gráficos tratan los filtros como extensiones modulares del programa (*plug-in*) y, por lo tanto, se pueden añadir nuevos. El software comercial permite que "terceras partes", empresas o personas distintas a la creadora del software, desarrollen filtros que el usuario puede incorporar previa compra o descarga. Al haberse mantenido Photoshop como líder del software de gráficos de mapa de bits durante mucho tiempo la mayoría de programas admite la importación de sus filtros (o sea de los filtros que cumplen las especificaciones dadas por Adobe para *plug-in* de Photoshop).



Filtros aplicados a la imagen: Artístico-Fresco, Bosquejar-Fotocopia, Desenfocar-Desenfoque radial, Distorsionar-Onda, Distorsionar-Esferizar, Estilizar-Difusión, Interpretar-Destello, Pixelar-Grabado, Textura-Grieta, Trazo de pincel-Salpicaduras, Estilizar-Solarizar de la aplicación Photoshop CS4.

### 2.5.4. Ajustes de imagen

Si las herramientas de dibujo usan metáforas y símiles del dibujo y la pintura convencionales sobre papel y tela, las opciones de **ajuste de imagen** tienen como referente la fotografía. Son opciones que permiten "ajustar" parámetros de la imagen como el color o el contraste y que se asemejan al trabajo de laboratorio que se realiza con la fotografía analógica, con un proceso químico de revelado.

Los ajustes más importantes son:

- **Niveles:** permiten corregir y/o equilibrar el número de píxeles en luces (o "resaltados"), medios tonos y sombras usando un histograma que muestra la cantidad de píxeles de la imagen en cada nivel de brillo (en una escala de 0 a 256 niveles posibles).
- **Curvas:** permite corregir el rango tonal, igual que "niveles", pero también se puede modificar color por color (de los tres primarios: rojo, verde y azul).
- **Brillo/contraste:** permite modificar el brillo de todos los píxeles de una imagen, luces, medios tonos y sombras, todos a la vez, y también corregir el contraste entre ellos, acentuándolo o mitigándolo.
- **Balance o equilibrio de color:** permite modificar el balance del color, ajustando las dominancias de color que puede haber en una imagen.
- **Tono/saturación o matiz/saturación:** permite modificar los tonos de color de una imagen, aumentar o bajar su saturación y también aumentar o bajar su luminosidad. Afecta a toda la imagen. Usando este ajuste con la opción de colorear activada se pueden realizar virados de un mismo tono en toda la imagen.

En cualquier caso la mayoría de ellos pueden entenderse como una actuación sobre el color de la imagen.

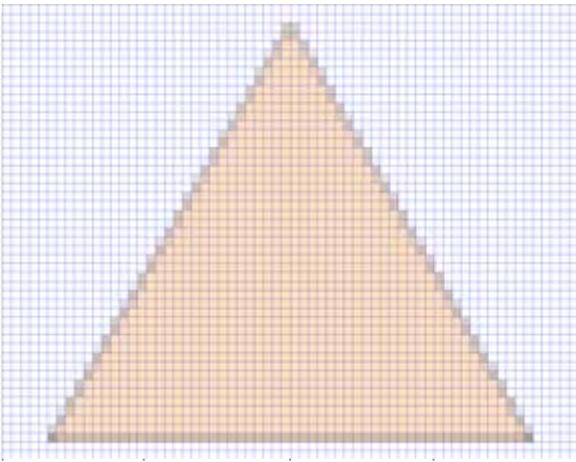
#### Ved también

En el apartado "Color y diseño gráfico" del módulo "Conceptos básicos de diseño gráfico", se tratan más a fondo algunas cuestiones relacionadas con estos ajustes.

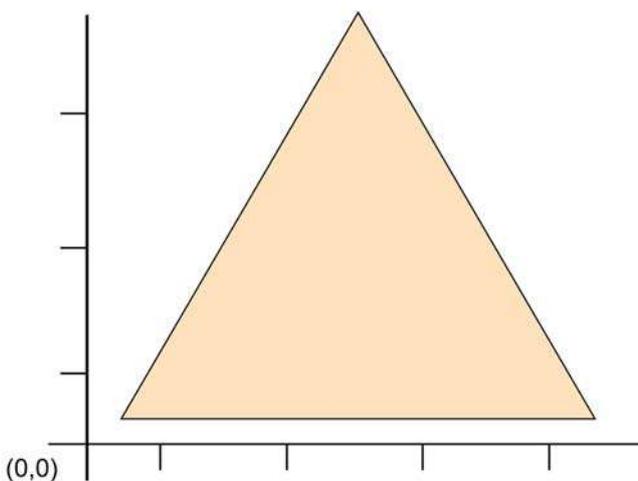
### 3. Gráficos vectoriales

#### 3.1. Dibujo basado en objetos

En los gráficos vectoriales la información de la imagen no se almacena sobre la base de una matriz de píxeles, como en los gráficos de mapa de bits. Lo que codifica el gráfico es la **información numérica de la geometría** del mismo. Por lo tanto, tenemos objetos gráficos independientes entre sí que se definen por puntos de coordenadas y vectores que unen dichos puntos.



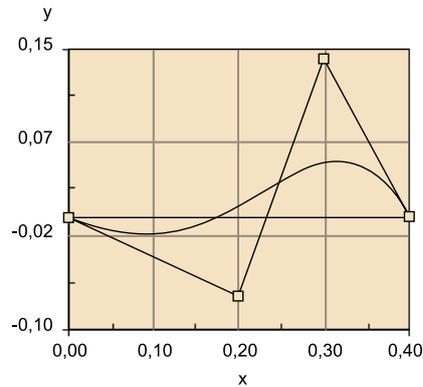
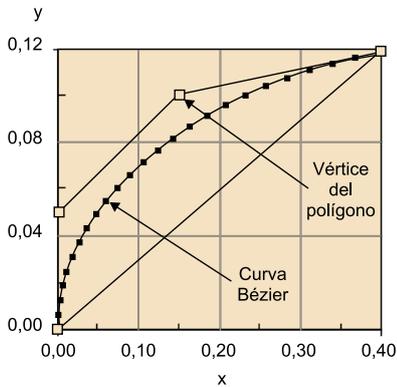
Ejemplo de un triángulo en un gráfico de mapa de bits (a una resolución baja). Tenemos información del color de cada punto de la imagen (representados aquí por las casillas). No tenemos información del triángulo en sí, éste se muestra a partir de la información de los puntos. El gráfico está guardado en un fichero GIF.



Ejemplo de un triángulo en un gráfico vectorial. El triángulo se define a partir de las coordenadas de sus vértices y de la unión de éstos a partir de líneas rectas. Tenemos pues la información geométrica del objeto. El gráfico está guardado en un fichero SWF.

Hay varias formas de codificar una imagen a partir de vectores. La más extendida entre los programas y los formatos de fichero de gráficos vectoriales es la que se basa en las llamadas **curvas de Bézier** o por extensión **objetos Bézier**.

### 3.1.1. Objetos Bézier



Los **objetos Bézier** son segmentos de línea conectados entre sí por **nodos**. Cada segmento lo entendemos como un vector con un punto inicial y un punto final que definen la línea. A estos se añaden **dos puntos de control** que definen la curvatura de la misma. Los puntos de control parten de las tangentes de cada uno de los puntos extremos o nodos. Cuando los cuatro puntos están alineados tenemos una recta. Cuando los puntos de control se separan tenemos algún tipo de línea curva. Esta curva está siempre contenida dentro de un polígono cuadrilátero cuyos vértices son dichos cuatro puntos. La curva se calcula a partir de una interpolación creada por una secuencia de funciones que se basa en las coordenadas de los puntos. Esto hace que sea escalable y se vea bien a cualquier nivel de ampliación.

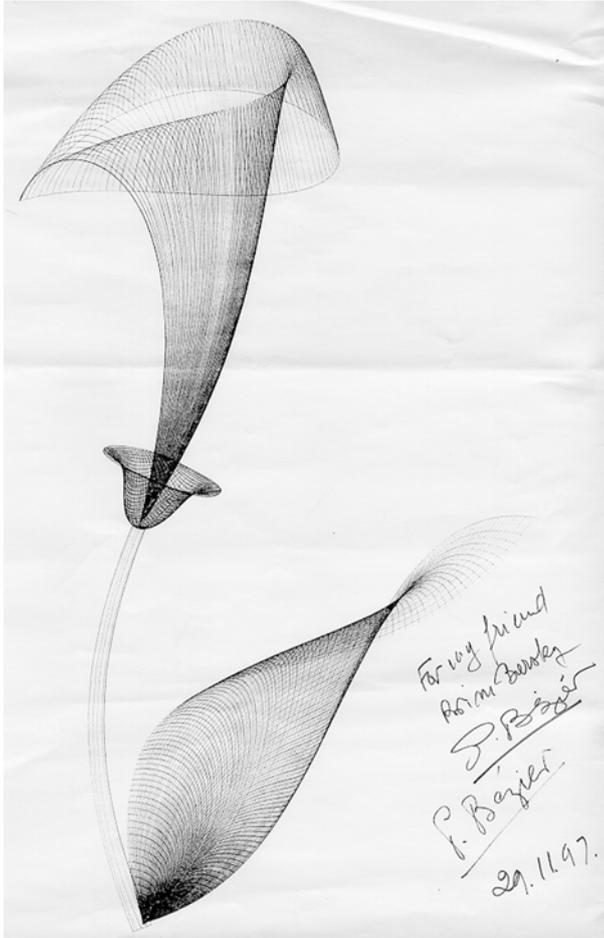
#### Pierre Etienne Bézier

Las llamadas **curvas de Bézier** deben su nombre y su origen a Pierre Etienne Bézier (1910-1999), que fue ingeniero de Renault durante 42 años (de 1931 a 1975). Como responsable del Departamento *des méthodes mécaniques*, a principios de la década de los sesenta, se empieza a preocupar por la utilización de herramientas informáticas en el diseño de carrocerías. Las herramientas de CAD (*Computer Aided Manufacturing*) basadas en polígonos no representaban bien las superficies curvas. Bézier logró un método fácil de usar y a la vez exacto para describir curvas a partir de cuatro puntos. El sistema se lanzó en 1968 y en 1975 ya estaba en pleno uso. Las **curvas de Bézier** han sido un elemento clave para el posterior desarrollo de la informática gráfica vectorial, tanto en 3D como en 2D.

Antes que Bézier dos matemáticos que trabajaban para Citroën (James Ferguson y Paul de Casteljaou) llegaron a los mismos resultados. Pero sus hallazgos no fueron conocidos hasta mucho más tarde porque se guardaron bajo secreto industrial.



Fotografía de Pierre Etienne Bézier.



© Descendentes Pierre Bézier. Esta imagen se reproduce acogiéndose al derecho de cita o reseña (art. 32 LPI), y está excluida de la licencia por defecto de estos materiales.  
Obra de Pierre Bézier realizada usando curvas de Bézier.

### 3.1.2. Bézier en el software gráfico

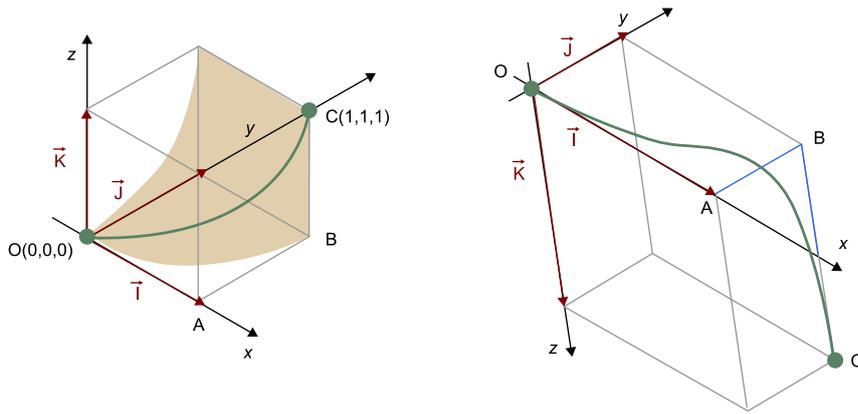
El software de gráficos vectoriales encuentra en las curvas de Bézier una metodología fácil de usar y de computar. La forma de trabajar se parece pero no es exactamente la misma que se ha visto en nuestra demostración anterior. La práctica del dibujo vectorial ha permitido depurar una interfaz más eficaz y "amable".

#### Curvas de Bézier en gráficos 3D

Aquí estamos viendo el uso de curvas de Bézier en software de gráficos de dos dimensiones. Pero Pierre Bézier quería las curvas para diseñar carrocerías de coche. Por lo tanto, quería curvar superficies, no sólo líneas. El procedimiento es el mismo. Para entenderlo podemos empezar describiendo una superficie curva dentro de un cubo y luego deformar este cubo (cambiando las posiciones de sus vértices) modificando así la curva inscrita en él.

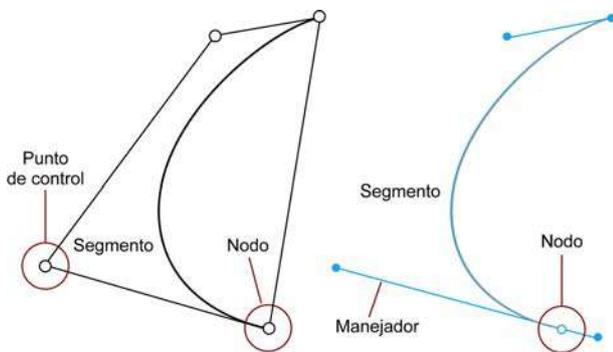
#### Uso de las curvas de Bézier en la geometría de las fuentes tipográficas

En tipografía digital es necesario un sistema de codificación de la información geométrica que permita que los tipos se impriman perfectamente a cualquier tamaño. Para ello también se usa el sistema de Bézier que fue adoptado como sistema estándar de codificación de curvas para el **lenguaje Postscript**. Es pues el sistema usado en las fuentes True Type (con curvas de segundo orden) y Postscript Tipo 1 (con curvas de tercer orden).



Un conjunto de curvas de este tipo conectadas entre sí nos permitirían definir cualquier superficie.

El software de gráficos 3D (que escapa de los objetivos de este material) usa también las curvas de Bézier para definir la forma de las superficies.



Para crear una línea se van creando los **nodos**, conectados entre sí por líneas rectas. Si se quiere curvar uno de éstos segmentos se extrae de "dentro" de uno de los nodos un **manejador**, que en realidad es un punto de control o vértice del polígono. Regulando el manejador correspondiente a cada nodo se define la **forma de la curva**.

En los programas de gráficos es habitual llamar **pluma** a la herramienta que permite crear curvas de Bézier de esta forma. También se suele representar por el icono de una pluma estilográfica.



Aquí puede observarse la secuencia de creación de una curva con programario vectorial. Se han realizado capturas de pantalla en 4 programas distintos entre los más utilizados (Adobe Illustrator, Freehand, Corel y Adobe Flash). Como puede observarse la metodología es muy similar y las diferencias en la interfaz son mínimas. Clicka el botón para avanzar o retroceder en el proceso de creación de la curva.

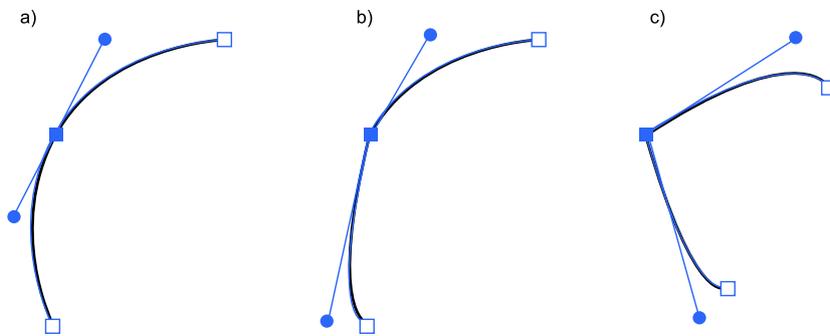
### 3.1.3. Trabajo de precisión con las curvas Bézier

Dibujando objetos Bézier es fácil crear elementos con demasiados nodos o con curvas que no conservan continuidad en todo el trazado. Existen una serie de normas que pueden ayudarnos a trabajar con mayor precisión. En la tabla siguiente presentamos las más importantes.

Normas	Incorrecto	Correcto
<p><b>La línea 1/3:</b> un manejador (distancia entre nodo y punto de control) no debe medir más de 1/3 del segmento al que modifica.</p>		
<p><b>Línea huella:</b> deben utilizarse el menor número de nodos posibles, así en un arco no deberían utilizarse más de dos nodos.</p>		
<p><b>Línea tangente:</b> el manejador debe ser tangente al segmento, pero nunca cruzarlo.</p>		

Aunque al formularlas como "normas" usamos un tono categórico, no hay que perder de vista que el diseño gráfico tiene un carácter experimental y cualquier norma puede ser cuestionada y subvertida. Estas normas son el resultado de la experiencia práctica y ahí reside su valor.

Otro aspecto importante a tener en cuenta es la diferencia entre **puntos de vértice** y **puntos de curva**. Cuando conectamos dos segmentos curvos, si queremos que la curva tenga una buena continuidad, las dos tangentes del nodo que comparten los segmentos deberán estar alineadas. En cambio si el nodo hace la función de vértice las dos tangentes no estarán alineadas.



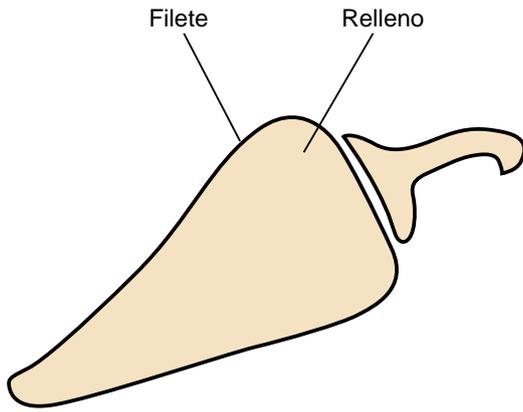
a) Aquí tenemos dos segmentos curvos bien conectados entre sí. Observad que las tangentes del nodo central están alineadas. El resultado es una curva que fluye sin "roturas". b) En esta línea las tangentes del nodo central no están alineadas y la curva no tiene buena continuidad. c) Aquí el nodo central es un vértice, deliberadamente los dos segmentos no forman una línea continua. Las tangentes no están alineadas y forman un ángulo.

El software de gráficos nos facilita aquí la tarea permitiendo que los dos manejadores se muevan conjuntamente o sean independientes entre sí. La mayor parte de programas de dibujo vectorial permiten que un punto sea de **tipo vértice** o de **tipo curva** y pasar de un tipo a otro cambiando las propiedades del punto.

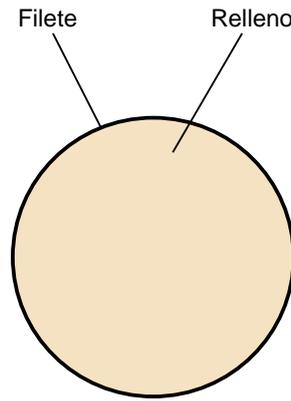
### 3.2. Propiedades de los objetos vectoriales

Las propiedades principales que componen un objeto vectorial son las siguientes:

- **Color interior o relleno:** determina el relleno del objeto. Este puede ser un color plano, una textura o un degradado. Para que se pueda rellenar un objeto, éste debe estar cerrado (el último nodo se une con el primero).
- **Filete o línea:** representa la línea externa que envuelve al objeto. Se puede variar el color, el tipo (líneas discontinuas, puntos, etc.) y el grosor.
- **Posición:** determina cuál es la situación del objeto en el documento de trabajo. Los objetos vectoriales se distribuyen en un sistema de coordenadas de dos dimensiones (x, y), lo que equivale a un plano.
- **Tamaño:** determina qué área ocupa el objeto en el documento de trabajo.



Tamaño: Ancho = 275; Alto = 155  
 Posición: X = 150; Y = 0



Tamaño: Ancho = 53; Alto = 53  
 Posición: X = 150; Y = 0

Propiedades del objeto vectorial

### 3.2.1. Propiedades de la línea o filete

Las propiedades del filete pueden variar de un programa a otro. Básicamente podemos diferenciar las propiedades que ofrecen los programas de "primera generación" destinados a impresión para artes gráficas (Freehand, Illustrator, Corel) y las que ofrecen los de "segunda generación" pensados para la publicación en web y, por lo tanto, para la visualización en pantalla (Fireworks, Flash).

Los programas originariamente destinados a crear diseños, de los que se sacarán fotolitos y luego se imprimirán en una imprenta offset, dan mucha importancia a la forma de la línea. En ellos se pueden controlar propiedades como las que se muestran en la tabla siguiente:

Propiedades de la línea	
<p>6pt </p> <p>12pt </p> <p>24pt </p>	<p><b>Grosor:</b> se reparte (a partes iguales) entre la parte inferior y superior de la línea imaginaria marcada por los dos nodos. El ancho se suele medir por puntos de pica (unidad de medida usada por los tipógrafos). En el software orientado a web, las medidas son por píxeles.</p>
	<p><b>Remate:</b> la terminación de la línea puede acabar a la misma altura del último nodo o bien ser tan gruesa como el ancho de la misma. También puede tener una terminación redonda o, en algunos programas, en forma de punta, de flecha o de círculo.</p>
<p>Agudo</p> <p>Redondo</p> <p>Biselado</p>	<p><b>Vértice:</b> la forma de los vértices (nodos no terminales de la línea) también puede definirse. Normalmente, se ofrecen tres opciones: agudo, redondo y biselado.</p>

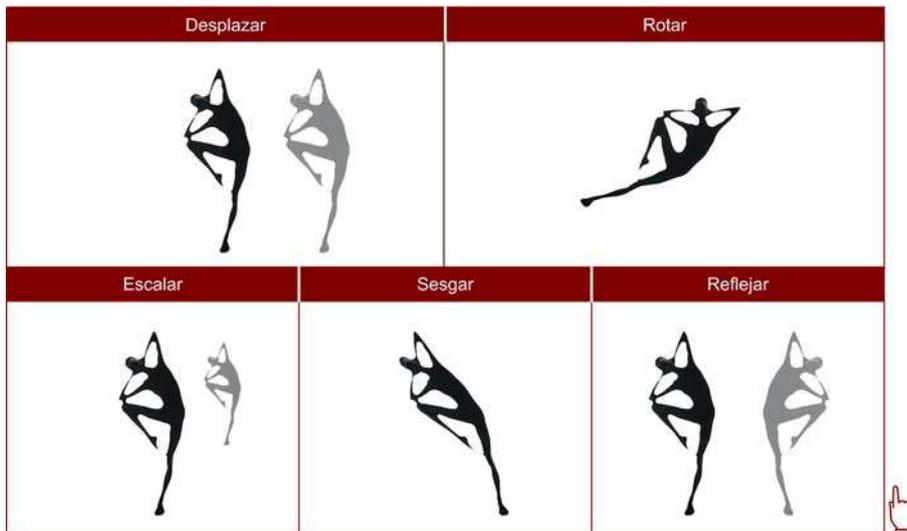
Los programas pensados para crear gráficos vectoriales que se publicarán en Internet hacen mayor hincapié en la visualización en pantalla. Por ello, introducen elementos como el suavizado (o *antialiasing*) para que la línea no se vea dentada. También se introduce la simulación de trazos de herramienta (caligráfico, carbón, aerógrafo, etc.) y texturas que puedan dar carácter a la línea, aproximándose a los efectos del software de mapa de bits.

Ejemplo	Descripción
	Línea básica "dura" (cuando la línea es inclinada vemos el dentado), de 10 píxeles de ancho, sin borde y sin textura.
	Línea básica "suave" ( <i>antialiasing</i> ), ancho de 10 píxeles, con borde difuminado 50 y sin textura.
	Línea básica con remate redondeado "suave" ( <i>antialiasing</i> ), ancho de 30 píxeles, difuminado de borde de 6 y sin textura.
	Línea básica con remate redondeado "suave" ( <i>antialiasing</i> ), ancho de 30 píxeles, difuminado de borde de 90 y con una textura ("Arena") a un 40%.
	Línea básica con remate redondeado "suave" ( <i>antialiasing</i> ), ancho de 40 píxeles, difuminado de borde de 90 y con una textura ("Arena") a un 95%.
	Línea "de aerógrafo", ancho de 30 píxeles, difuminado de borde de 60, sin textura.

### 3.2.2. Transformaciones de los objetos vectoriales

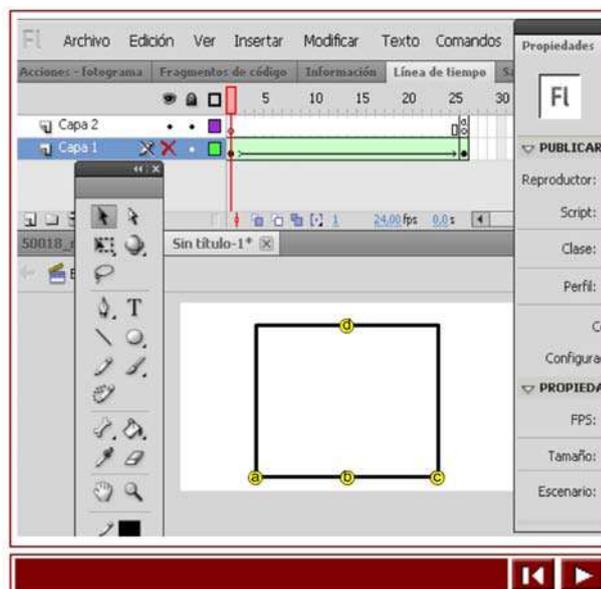
El software vectorial es como un laboratorio de geometría. Cualquier transformación geométrica tiene una herramienta o procedimiento para aplicarla a los objetos vectoriales.

Las transformaciones más habituales se muestran en la tabla siguiente:

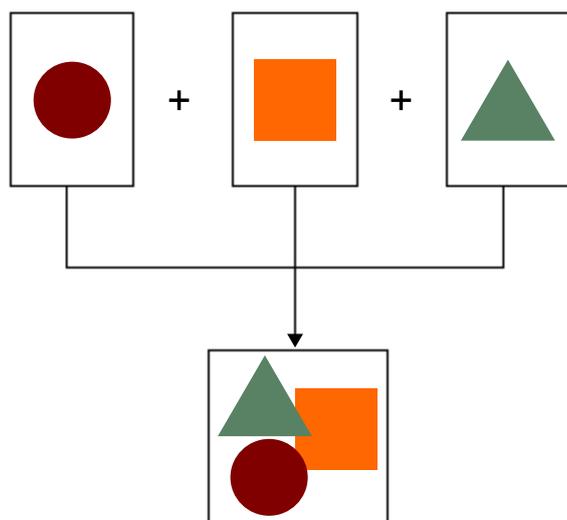


Todas estas transformaciones pueden ser controladas numéricamente con precisión o bien realizarse de forma intuitiva sobre el propio objeto.

Otra posibilidad disponible es la **fusión de un objeto con otro** distinto creando objetos intermedios que forman una secuencia de metamorfosis. Se puede elegir el número de pasos y en algunos programas (como Flash) es posible insertar una serie de puntos (llamados **consejos de forma**) que permiten controlar donde irá a parar un punto del objeto inicial en el objeto final.



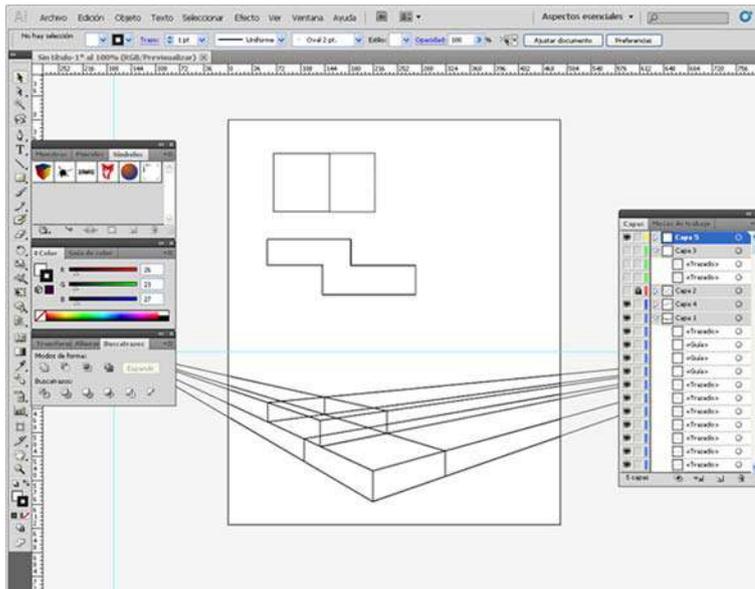
### 3.2.3. Composición de gráficos vectoriales



Composición en planos o capas de los objetos vectoriales.

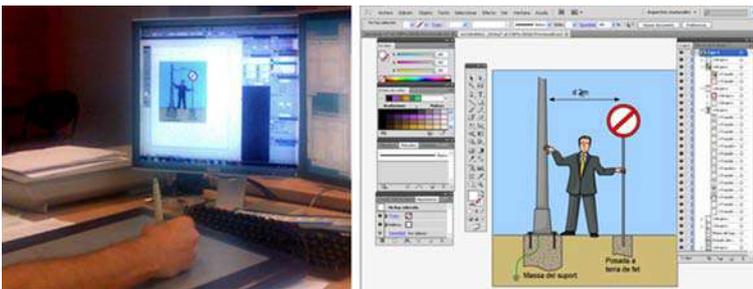
Las imágenes vectoriales se componen de diferentes objetos independientes, lo cual permite la manipulación de cada uno de ellos por separado. Estos objetos no se encuentran en el mismo plano geométrico. Están superpuestos entre sí y existe la posibilidad de pasarlos hacia delante o hacia atrás. Esto es así para la mayoría de programas vectoriales (Illustrator, Freehand, Fireworks, CorelDraw, Inkscape, Xara Xtreme) y quizás Flash sea la excepción. En Flash los objetos comparten un mismo plano geométrico y al superponerlos se suman automáticamente (a no ser que estén agrupados).

Además la mayoría de programas ofrecen la posibilidad de trabajar con **capas**. La metáfora de las capas supone que estamos trabajando con distintas hojas de papel de cebolla o acetato transparente. Cada capa es independiente y se superpone a la anterior. Trabajar con capas permite organizar mejor el trabajo. Las capas pueden bloquearse u ocultarse, bloqueando u ocultando todos los objetos que contienen. En cada capa hay también varios planos en función del número de objetos que tengamos en ella.



Trabajo con capas en Fireworks

### 3.3. Mano alzada: dibujar vectores sin nodos ni manejadores



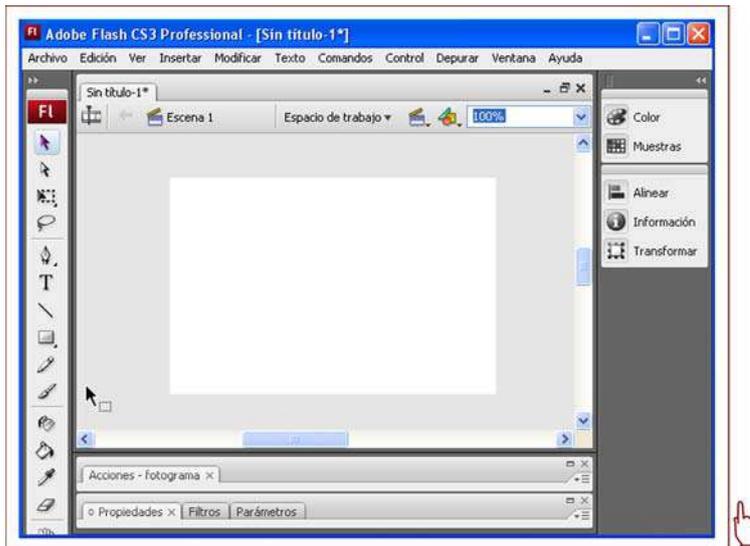
Existe una forma de dibujar vectores sin usar nodos ni manejadores. En los programas de dibujo vectorial esta función corresponde a la llamada **herramienta de mano alzada** o de **pincel**, llamadas así por contraste con la herramienta "de pluma" para trazar nodos. Internamente el resultado sigue siendo objetos Bézier pero la forma de dibujar es muy distinta. Podríamos decir más "libre" o más "natural". Los trazos se generan a partir de movimientos del ratón sin tener que ir marcando la línea punto a punto. En realidad lo que está generando el programa no son líneas sino áreas cerradas o rellenos.

Cuando se saca mayor partido a la mano alzada es usando una tableta gráfica con lápiz que detecte la presión. Los programas que son "sensibles a presión" pueden cambiar el grueso de la línea generada (en realidad relleno) en función de ésta. Ello representa una ventaja considerable para dar carácter gráfico al trazo, ya que las líneas convencionales tienen un grueso constante sin posibilidad de modulación. Esta forma de trabajar está especialmente indicada para crear ilustraciones.

Aunque la herramienta de mano alzada está disponible desde hace tiempo en el software vectorial orientado al diseño gráfico, su mayor popularización va pareja con la aparición de Flash que la llevará a sus máximas posibilidades.

La interfaz de dibujo de Flash equipara prácticamente el dibujo vectorial a la forma en que se dibuja en un programa de mapa de bits. Aunque los objetos gráficos se basan internamente en Bézier es posible borrarlos con una herramienta de goma y al dibujar con un mismo color los objetos se unen automáticamente entre sí.

Quienes pueden sacar mayor partido de este modo de trabajo son los ilustradores, incluso aquellos acostumbrados a trabajar con herramientas no-digitales, que encuentran aquí una forma similar de dibujo basada en el gesto de la mano.



Clicad para ver una simulación animada de trabajo con Flash.

Primero se usa el pincel y la goma como si se tratara de un programa de mapa de bits. Pero luego puede observarse como al seleccionar estos objetos están definidos por curvas de Bézier y es posible modificar y editar sus nodos.

### 3.4. La resolución de las imágenes vectoriales

Los objetos que componen una imagen vectorial presentan sus propias características, así podemos tener objetos con colores, tamaños y formas determinadas, y variando una de esas características modificaremos el objeto. Ello significa que al aumentar una imagen no estamos distribuyendo una serie de píxeles, ni aumentando su número por superficie, lo único que hacemos es variar los parámetros del algoritmo que calcula el objeto. Podemos ampliar lo que deseemos y su calidad no se verá afectada, siempre será la máxima que ofrezca el dispositivo de salida (pantalla, impresora, etc.).

En consecuencia, el objeto vectorial no depende de la resolución y en la mayoría de los casos su tamaño de almacenamiento es muy inferior al que podría tener una imagen de mapa de bits.

### 3.5. Cuándo usar gráficos vectoriales

Trabajaremos con **imágenes vectoriales** cuando deseemos hacer un tipo de imagen que se pueda modificar o retocar fácilmente. Bien porque se necesite una aplicación de dicha imagen a **varios tamaños** o bien porque interese obtener un archivo de **reducido tamaño** de almacenamiento.

La flexibilidad de trabajo que permiten deriva de la independencia de sus objetos y la posibilidad de poderlas reproducir a cualquier tamaño sin pérdida de calidad.

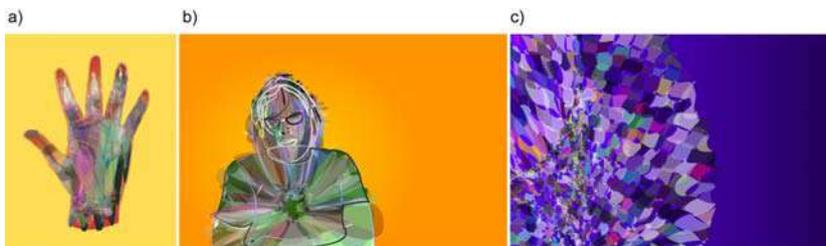
Normalmente, se suelen utilizar para trabajos que deban imprimirse con calidad profesional en resoluciones altas. Este tipo de trabajos pueden ser publicaciones de libros, revistas, logotipos, trípticos, etc.

También se usan en aplicaciones interactivas y multimedia en Internet. La baja memoria de almacenamiento de los gráficos vectoriales los convierte en el tipo de imagen ideal para un medio que está muy condicionado por el tamaño de los documentos. Si además se trabaja con formatos que permiten añadir programación (como SVG o SWF) es posible manipular dinámicamente la geometría de los objetos.

Las imágenes vectoriales también condicionan un determinado resultado gráfico. Por su propia naturaleza son adecuados para dibujos de línea o de colores planos. Pueden contener degradados pero no en la línea de las imágenes de tono continuo como las fotográficas. No es imposible, pero resulta muy difícil simular una imagen fotográfica mediante el uso de vectores y, de hacerlo, el número de nodos necesarios hará que el espacio de almacenamiento sea muy alto. Hay varias líneas gráficas que exploran las posibilidades de una estética de los gráficos vectoriales aprovechando sus propias características. En esta línea también son usados para creaciones audiovisuales, desde narrativas o de tipo artístico hasta publicitarias.



En este trabajo Luciano Lourenço ha usado las herramientas de vectorización y retoque de Inkscape para crear una imagen de tipo fotorrealista muy convincente.  
© Luciano Lourenço. *Busto de Mulher* 2009. Free Art License.

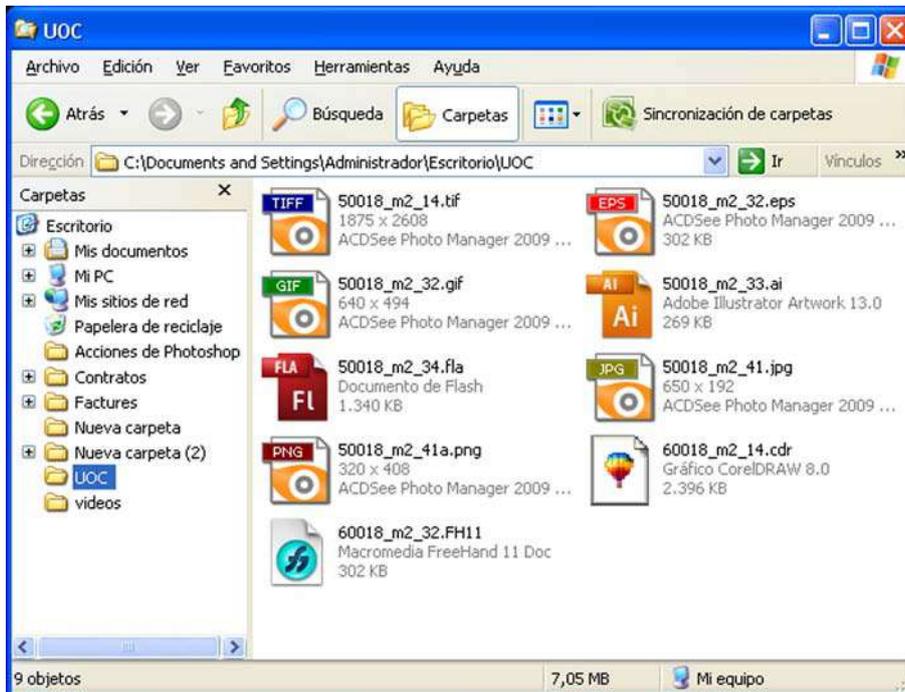


a) *Another hand*. b) *A bit of a mess*. c) *Fading tiled friend*.  
© Antonio Roberts 2009 - [www.helloatfood.com](http://www.helloatfood.com) - Creative Commons - Reconocimiento Compartir Igual - 3.0 - uk.

En estos trabajos Antonio Roberts explora las posibilidades de una estética de gráficos vectoriales, aprovechando sus propias características y posibilidades como vía de experimentación estética. Usa la mancha, la línea de mano alzada y los efectos de transparencia en composiciones donde la combinación de múltiples colores comparte protagonismo con las formas. El trabajo ha sido realizado con Inkscape.

## 4. Formatos gráficos

### 4.1. Codificar gráficos



En etapas anteriores se ha visto que hay dos formas de codificar una imagen digital:

- **mapa de bits:** se guarda la información de color de cada píxel de una retícula.
- **vectorial:** se guarda información geométrica de cada objeto.

En consecuencia ¿existen sólo dos formatos de archivo para codificar gráficos, uno de mapa de bits y otro vectorial? Pues no. Existe una enorme cantidad de formatos gráficos, cada uno de ellos con sus características propias. Esto en parte es así porque el desarrollo de formatos no es un proceso planificado. Surgen gran cantidad de iniciativas de las universidades, de la industria del software y de programadores independientes, tanto de forma individual como coordinada. En función de factores muy diversos algunos de estos formatos tienen éxito y son aceptados por todo el mundo y otros no.

Pero existe otra razón para la gran diversidad de formatos: responden a distintas necesidades. Hay formatos adecuados para una finalidad y formatos adecuados para otra.

Como creadores gráficos, es importante saber distinguir los tipos de formato y sobre todo saber cuál escoger en función de su aplicación final.

#### 4.1.1. Tipos de formato

Antes de hacer un repaso de los formatos gráficos más significativos veamos esquemáticamente cuáles son los aspectos que nos permitirán distinguir unos de otros.

Algunas características tienen que ver con el **almacenamiento de datos**:

- **La forma de almacenarlos:** puede ser en píxeles (mapa de bits), vectorial o un metafichero (formatos que admiten tanto gráficos vectoriales como de mapa de bits).
- **Posibilidad de características extra:** admitir canales alfa y/o transparencia, descarga progresiva (para gráficos web), especificaciones para la separación de colores (para gráficos a imprimir), etc.

Otras tienen que ver con su **finalidad y condiciones de uso**:

- **Finalidad del gráfico:** dependiendo si el gráfico está destinado a una página web o a impresión será más adecuado un formato u otro.
- **Condiciones de uso del formato:** si es multiplataforma (para distintos sistemas operativos), si es un formato de libre uso o tiene alguna restricción, si cumple los estándares o no, qué empresas o organizaciones lo apoyan, etc.

#### Ved también

En el subapartado siguiente "Formatos para gráficos vectoriales y metaficheros" se comentarán estos aspectos para formatos concretos, tanto de mapa de bits como vectoriales.

#### 4.2. Formatos para gráficos vectoriales y metaficheros

Podemos agrupar los diferentes formatos gráficos en dos grandes grupos que corresponden a dos generaciones. La primera, formada por **formatos para artes gráficas y arquitectura**. La segunda generación compuesta por **formatos para Internet y la web**.

Algunos de los formatos que llamamos **vectoriales** son en realidad **metaficheros**, ya que admiten gráficos de mapa de bits incrustados y también texto. Por el tratamiento de los elementos gráficos como objetos independientes, es habitual que los formatos vectoriales incorporen otro tipo de elementos y sean, en mayor o menor medida, metaficheros. Aquí los trataremos conjuntamente pero indicando en cada caso si se trata de un formato vectorial estricto o bien de un metafichero.

### 4.2.1. Formatos para artes gráficas

Durante años el formato vectorial más aceptado para artes gráficas fue el **EPS** (*Encapsulated PostScript*) cumpliendo el papel que más tarde cumpliría el **PDF**. EPS se apoya en Postscript, un lenguaje de descripción de objetos vectoriales que se desarrolló durante la segunda mitad de la década de los setenta y cuya implementación final estuvo a cargo de John Warnock, que poco después fundaría Adobe Systems Inc.

El **lenguaje Postscript** es un lenguaje de programación completo y usa curvas de Bézier cúbicas, con dos manejadores, para describir los objetos. Pero EPS además de basarse en Postscript es un formato pensado para transmitir información a una impresora que interprete dicho lenguaje independientemente de la plataforma y el dispositivo. Además de la información geométrica de los elementos gráficos el EPS guarda información de su posición dentro de un marco.

El EPS admite gráficos de mapa de bits incrustados y es usado para exportar imágenes de mapa de bits junto con un trazado que las recorta del fondo, lo que permite tener una silueta con "fondo transparente".

Otros formatos extendidos son los de Microsoft **WMF** (*Windows Meta File*) y su "sucesor" **EMF** (*Enhanced Windows Metafile*). Son formatos compatibles con cualquier software Microsoft y que interpretan bien los datos de otros formatos vectoriales, como por ejemplo, los formatos de arquitectura o de ingeniería (**dwg**, **dxf**, etc.). Aunque no son formatos muy usados en artes gráficas y su uso suele quedar circunscrito a contextos donde se trabaja con varios programas de Microsoft entre los que cumplen la función de formatos de intercambio.

El formato **PDF** (*Portable Document File*), creado por Adobe como el EPS, se presenta como una evolución de éste que cumple la doble función de servir para las artes gráficas y para Internet. Incorpora a las características del EPS el hecho de no necesitar las fuentes tipográficas para representar bien el texto. Guarda la geometría de las mismas y si no las encuentra en el sistema las dibuja a partir de los datos que tiene. Se puede considerar un metafichero que puede incorporar:

- Texto guardado como tal; con información sobre las fuentes tipográficas para visualizarlo.
- Gráficos vectoriales para ilustraciones y otros elementos; almacenados a partir de sus datos geométricos.
- Mapas de bits incrustados para fotografías y otras imágenes.

La primera versión del formato PDF se lanzó en 1993 y sus sucesivas versiones han ido siempre acompañadas del lanzamiento del software de visualización y edición del mismo, **Acrobat**, por parte de Adobe. En julio del 2008 es aceptado por la International Organization for Standardization (ISO) como estándar (ISO 32000-1:2008 PDF) y su especificación está disponible para que cualquier desarrollador pueda implementar herramientas de soporte. Adobe mantiene la propiedad de diversas patentes sobre el formato pero permite su uso sin contrapartidas mientras se cumpla con esa especificación.

#### Ved también

En el apartado siguiente "Formatos para Internet" se comenta el uso de PDF como formato para la publicación en Internet.

### 4.2.2. Formatos para Internet

Durante mucho tiempo uno de los problemas de los formatos vectoriales fue que no eran compatibles con el lenguaje web HTML, a diferencia de los archivos de mapa de bits, que en seguida fueron incorporados a la web y fueron reconocidos por los navegadores. La solución a esta carencia ya está disponible. Se trata del formato **SVG** (*Scalable Vector Graphics*), un formato basado en el lenguaje XML que tiene el apoyo del W3 Consortium (W3C), el organismo independiente que define los estándares de desarrollo de la www. Ya está soportado por varios navegadores, mientras que en otros, que aún no soportan plenamente el lenguaje XML, es necesario un módulo extra para poder visualizar archivos SVG en ellos.

La ventaja de SVG es que además de dibujar geometría vectorial, permite crear animaciones y programar interactividad a partir de lenguajes de script. Algunos programas de gráficos vectoriales (como Illustrator o Inkscape) exportan a dicho formato.

Los demás formatos vectoriales disponibles para Internet no son compatibles con lenguajes estándares como XML o no se incorporan de forma integrada con el resto de elementos de página en HTML. En estos momentos los más relevantes son: *Shockwave Flash* (**swf**) y *Portable Document File* (**PDF**).

El **PDF** funciona como formato para la edición electrónica además de cumplir la función de formato destinado a la impresión en sustitución de EPS, como ya se ha dicho. Permite que la maquetación de elementos de página (gráficos y texto) se mantenga estable entre plataformas, permite crear hipervínculos a direcciones web externas al documento y añadir otras características de navegación e interactividad dentro del mismo. También permite incorporar información relativa al *copyright*.

El PDF requiere de un visualizador propio. Adobe desarrolla **Acrobat Reader** que se distribuye como programa independiente pero también como extensión (*plug-in*) del navegador. De esta forma es posible ver un documento PDF dentro de la ventana del navegador abriendo el módulo Acrobat. Lo que no es posible es incrustar un gráfico en PDF dentro de una página HTML. Hay otros

visualizadores de PDF (como EVince o KPDF) y varios programas de edición que permiten editarlo, como el propio **Acrobat Professional** que Adobe desarrolla en paralelo a la evolución del formato.

El PDF es un formato habitual de distribución de documentos muy extendido en las administraciones públicas y entre empresas. Se usa también muy a menudo como versión imprimible de las páginas web y, desde que se publicó su especificación, esta opción se genera a menudo de forma automatizada a partir de los datos en XHTML.

El formato **SWF**, siglas de *small web format*, es un formato desarrollado inicialmente por FutureWave Software y Macromedia (luego absorbida por Adobe), que permite crear gráficos vectoriales con animaciones e interactividad. También permite vídeo integrando en su interior archivos en formato FLV (*Flash Vídeo*). Fue inicialmente el formato nativo de publicación de Flash. A partir del 2008 la nueva propietaria Adobe empezó un proceso de apertura de la especificación para que otros desarrolladores puedan desarrollar herramientas y para que los buscadores (como Google o Yahoo) puedan indexar el contenido de sus ficheros.

A diferencia de los documentos PDF, los archivos SWF sí que pueden incrustarse dentro de una página HTML, pero para visualizarse requieren de una extensión (*plug-in*) del navegador. Mientras los navegadores no tenían integrada la interpretación de ficheros SVG el SWF se convirtió en estándar *de facto* para ficheros vectoriales y de animación en la web. La diferencia con SVG es que SWF no es un estándar abierto impulsado por un organismo de consenso como el W3C. La política de Adobe es que cada nueva versión de su programa Flash implica una nueva versión del formato, por lo que el usuario deberá estar siempre actualizando a las nuevas versiones del *plug-in*.

Además del propio Flash hay otros programas de gráficos vectoriales que generan archivos SWF como Fireworks o Illustrator.

### 4.3. Formatos para mapa de bits

Hay una gran cantidad de formatos de archivo para gráficos de mapa de bits.

En los formatos de mapa de bits un factor importante a tener en cuenta es el **sistema de compresión** que utilizan, ya que suelen ser archivos que ocupan mucha memoria, pues tienen que guardar información para cada píxel. Por eso algunos son más adecuados para impresión, donde la memoria de almacenamiento no es tan importante, y otros para la web, donde el tamaño del archivo puede ser importante en función del ancho de banda.

Otro factor importante es el relativo a la **profundidad de color** que admiten.

Vamos a ver estas cuestiones para formatos concretos a lo largo de este apartado.

### 4.3.1. TIFF, el formato de impresión

El formato habitual para imprimir gráficos de mapa de bits es **TIFF** (*Tagged Image File Format*). Es un formato desarrollado conjuntamente por Microsoft y la desaparecida Aldus, actualmente propiedad de Adobe.

El formato TIFF es multiplataforma y compatible con múltiples programas y dispositivos (escáneres, impresoras). Está pensado para garantizar la calidad de impresión y por ello los archivos TIFF suelen ocupar bastante memoria en disco al no aplicarse a las imágenes una compresión con pérdida de datos. Admite distintos modos de color, entre ellos: escala de grises, CMYK, RGB y CIE L\*a\*b.

Aunque se suele usar para imágenes de mapa de bits, en realidad es un **meta-fichero** que permite incorporar ciertos datos vectoriales (como un perfil de recorte) e información de páginas o capas.

El formato TIFF admite opcionalmente un sistema de compresión sin pérdida de calidad, el conocido como **LZW** (*Lempel-Ziv-Welch*). Los archivos comprimidos ocupan menos memoria pero tardan más en abrirse.

### 4.3.2. Formatos para la web: GIF, JPEG y PNG

Los formatos de mapa de bits más habituales para la web son GIF, PNG y JPEG. Cada uno tiene características distintas que lo hacen más adecuado para un tipo de gráfico.

**GIF** (*Graphics Interchange Format*) es un formato desarrollado en 1987 por CompuServe Inc. para el intercambio de gráficos en Internet. Desde entonces, es uno de los formatos más usados para los gráficos en la web. GIF codifica los gráficos hasta 8 bits por lo que admite paletas de 256 colores o menos. Es adecuado para gráficos con colores planos y/o líneas finas.

En 1989 CompuServe sacó una nueva versión llamada **GIF89a** que incorpora la posibilidad de designar uno o varios colores de la paleta como transparente, conocido como **transparencia de índice**. Ello es útil cuando se quiere tener un gráfico silueteado sobre el fondo de una página web.

Otra variante es el **GIF animado** que se compone de una secuencia de diferentes fotogramas en formato GIF. La sucesión de estos hará que veamos una secuencia animada. Para realizar este tipo de GIF se puede recurrir a programas

#### Ved también

Los modos y sistemas de compresión del color (escala de grises, CMYK, RGB, CIE L\*a\*b) están explicados en el apartado "Color y diseño gráfico" del módulo "Bases de diseño gráfico".

Ampliar información sobre el sistema de compresión LZW en los apartados "Sistemas de compresión" y "La polémica LZW".

creados con este fin, o a otros que incorporan un módulo de animación GIF como Fireworks. El formato GIF usa un LZW como algoritmo de compresión (sin pérdida de calidad).

El GIF incorpora junto a la imagen la paleta de colores que se ha usado. Es una paleta indexada, lo que significa que cada color tiene asignado un número mediante el cual se indica en la imagen el color con el que se representará cada píxel. Si una imagen tiene muchas variedades de un color es posible incorporar una paleta con todas ellas al precio de reducir el resto de colores. Podríamos, por ejemplo, definir una paleta con 256 tonalidades de rojo; pero la imagen sólo podría usar esas tonalidades y no otros colores.

Los GIF, excepto la variante animada, permiten ser almacenados como **entrelazados**. Esta opción es útil para la web, ya que permite que una imagen se vaya viendo a medida que se descarga, de esta manera nos podemos hacer una idea de la imagen que aparecerá en ese lugar.

Como respuesta a las restricciones en el uso del algoritmo de compresión LZW (polémica que se trata más adelante) la comunidad de software libre emprendió el desarrollo de un formato gráfico para la web que sustituyera a GIF. El resultado es un formato mucho mejor y con nuevas características: **PNG** (*Portable Network Graphics*).

Existen 3 versiones en función del número de bits usados para la codificación del gráfico: **PNG8**, **PNG24** y **PNG32**. Es un formato adecuado tanto para colores planos como fotografías de tono continuo. Admite el **modo indexado** incorporando una paleta de colores (PNG8) y también el **modo RGB** (PNG24 y PNG32) generando variedad de colores a partir de sus tres componentes (rojo, verde y azul) almacenados en canales.

También admite **carga entrelazada** basada en el método Adam7 (más rápido que el de GIF). En la versión PNG8 admite transparencia de índice, en la versión PNG 24 no admite transparencia, pero en la versión PNG32 puede guardarse un **canal Alfa** por lo que permite diferentes niveles de transparencia o una silueta con los contornos suavizados respecto al fondo. Tiene un sistema de compresión libre sin pérdida de calidad conocido como Deflate/Inflate, derivado del LZ77, que es de la misma "familia" que el LZW.

### ¿PNG no se ve en navegadores antiguos?

PNG es más reciente que GIF y no es soportado por navegadores antiguos.

Concretamente no empieza a ser soportado hasta la versión 4 de los navegadores Microsoft Explorer o Netscape Navigator y no es soportado plenamente (incluyendo transparencia Alfa) hasta versiones superiores a la 6 de ambos navegadores.

Sí es soportado por las versiones recientes de estos navegadores y por las nuevas generaciones de navegadores como Firefox, Safari o Chrome.

#### Ved también

Sobre los canales alfa ver también el subapartado "Herramientas de selección" del apartado "Gráficos de mapa de bits" de este mismo módulo.

Otro formato a considerar es el **JPEG** (*Joint Photographic Experts Group*), que adquiere el nombre de un comité creado para buscar un algoritmo eficaz para compresión de **imágenes fotográficas o de tono continuo**; tanto de escala de grises como en color. El sistema de compresión de JPEG obtiene una ratio de compresión mayor que LZW o el que usa PNG, aunque implica **pérdida de datos**. A mayor compresión, menor peso, mayor pérdida y menor calidad. Existen diferentes formas de especificar las diferentes calidades, siendo la más común la que especifica 13 niveles, del 0 al 12, siendo 12 la de mayor calidad y 0 la de menor.

Las imágenes JPEG admiten **carga progresiva**, parecida a la carga entrelazada. Este tipo de carga permite que cuando empiezan a llegar los datos de la imagen del servidor se dibuje una versión de baja definición (con aspecto muy pixelado) y conforme se va descargando la imagen se van haciendo barridos progresivos que aumentan su resolución hasta llegar a la versión final.

Este formato codifica el **color en RGB** y **no permite trabajar con paletas indexadas**. **No soporta ningún tipo de transparencia**. Es un buen formato para representar degradados, pero no contornos finos, detalles o áreas de color plano. Por lo que resulta adecuado para imágenes fotográficas, para lo que fue creado, pero no para imágenes con líneas o textos. Aunque hay estrategias de compresión que resuelven este problema que serán tratadas en secciones posteriores.

### 4.3.3. Sistemas de compresión

Los **sistemas de compresión** reducen la memoria que ocupa un archivo. **Comprimir** puede ser necesario para que ocupe menos espacio al almacenarlo en el disco pero se hace imprescindible cuando este archivo debe transferirse o publicarse en Internet. A menor tamaño de archivo, mayor rapidez en la transferencia de los datos. Una vez se tienen los datos el archivo se descomprime y se muestra.

La descompresión también representa una pérdida de tiempo, pero en general, para los sistemas de compresión/descompresión y las capacidades de procesamiento de los dispositivos electrónicos actuales, el tiempo perdido en la descompresión es menor al ganado por la bajada en el tamaño del archivo.

De algún modo lo que hacen los sistemas de compresión es eliminar redundancia y "resumir" la información a almacenar. Esto puede hacerse de dos formas:

- **Con pérdida de datos:** equiparar información parecida a un solo valor (por ejemplo, varios tonos de azul a un solo azul). Se resume la información, perdiendo detalles. La información destruida no se recupera en la

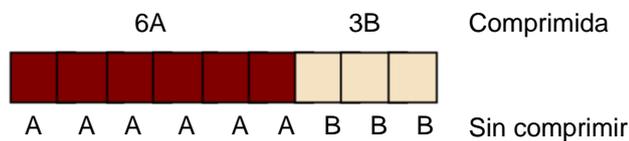
#### Ved también

Sobre las estrategias para evitar pérdida de calidad aparente en archivos JPEG que incluyan texto o líneas finas, ver el subapartado "Optimización de archivos de mapa de bits en formato PNG y GIF" del apartado "Optimización de gráficos para web" del módulo "Producción de gráficos".

descompresión. Este tipo de compresión es la usada, por ejemplo, en el formato JPEG.

- **Sin pérdida de datos:** se buscan formas de optimizar el registro de los datos eliminando la información redundante y simplificando la forma de expresarla. Toda la información "reducida" se vuelve a recuperar en la descompresión. Este tipo de compresión es la usada, por ejemplo, en los formatos TIFF, GIF o PNG.

**LZW** (*Lempel-Ziv-Welch*) es el algoritmo que usa el formato GIF y, opcionalmente, el formato TIFF para comprimir los gráficos (también la utiliza pdf). Se trata de un sistema de compresión sin pérdida de datos. A grandes rasgos se puede decir que dicho algoritmo realiza una codificación de secuencias consecutivas de valores de color, reduciendo al máximo la redundancia en la repetición de secuencia.



Un sistema de compresión sin pérdida de datos es el *Run-length encoding* (RLE) en el que secuencias de datos consecutivas con el mismo valor son almacenadas de forma "resumida" como un único valor más su recuento.

Observa de forma esquemática lo que podría hacer un sistema de compresión RLE con esta fila de 9 píxeles para resumir la información.

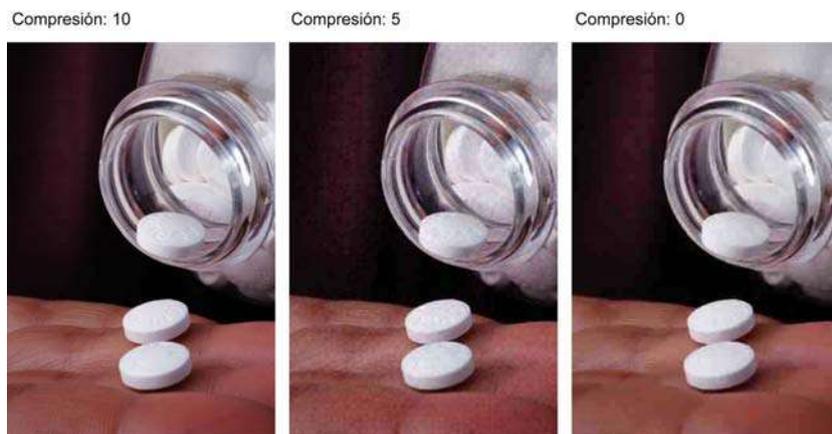
Realiza un proceso de reconocimiento de secuencias consecutivas de color, es decir, recorre uno a uno todos los píxeles. Si hay más de un píxel consecutivo del mismo color, almacena los datos de dicho píxel y la cantidad de ellos que hay en la secuencia. El resultado: una información que requería 9 dígitos (AAAAAABBB) se puede expresar con sólo 4 dígitos (6A3B).

Los sistemas derivados de LZ77 (como LZW) parten de esta compresión como base y la mejoran teniendo en cuenta la repetición en secuencias de dígitos (como podría ser en ABAABAAB).

El sistema de compresión **Deflate/Inflate** que usa PNG es un derivado de LZ77 (el algoritmo usado en zip, gzip, pkzip, etc.). El sistema sería similar al que hemos descrito en la ilustración del apartado anterior pero optimizado, pues introduce varios filtros que preprocesan la información de color para lograr una codificación más eficiente. Es también un **sistema de compresión sin pérdidas**.

El sistema usado por JPEG es muy bueno comprimiendo imágenes fotográficas, pero no con imágenes de colores planos o con líneas finas muy contrastadas. Básicamente, lo que hace este algoritmo es guardar la imagen separando la información de brillo de los matices de color y elimina las diferencias de color sutiles que no podemos apreciar, por eso es un formato de compresión que genera un proceso de pérdidas. Por muy poco que queramos comprimir,

siempre se pierde información. Como ya se ha dicho, se puede controlar el nivel de compresión, y cuanto mayor sea la compresión aplicada, menor será la calidad de la imagen y el volumen del archivo.



#### 4.3.4. La polémica LZW

Ya se ha mencionado anteriormente cómo las restricciones de uso de LZW impulsaron el desarrollo de PNG. Hagamos un breve repaso de este proceso.

El sistema de compresión **LZW** (*Lempel-Ziv-Welch*) fue desarrollado en el año 1983 por Terry Welch mejorando los algoritmos LZ77 y LZ78 desarrollados en 1977 y 1978 respectivamente por los investigadores israelíes Jacob Ziv y Abraham Lempel. El sistema es patentado en Estados Unidos en el año 1985, esta patente pertenece actualmente a la empresa **Unisys**.

En 1987 **CompuServe** crea su formato para intercambio de gráficos en Internet **GIF** (*Graphics Interchange Format*) que incorpora el sistema de compresión LZW, igual que su sucesor GIF89a de 1989. Durante estos años GIF se convierte en el estándar de intercambio de imágenes en Internet (no existía JPEG aún) y luego es también el estándar para imágenes sobre páginas web que todos los navegadores interpretan. Hasta finales de 1994 Unisys no reclama nada pero a partir de ese momento, reforzado por un acuerdo con CompuServe, reclama el pago de *royalties* a los desarrolladores de software que soportan GIF. Esto es incompatible con los programas de código abierto.

Ante esta situación algunos programadores de la comunidad de software libre se ponen en marcha para crear un formato alternativo. Durante 1995 el desarrollo avanza rápidamente con múltiples aportaciones. Se usa LZ77, que sí estaba libre de patentes, y la codificación de Huffman. En 1996 PNG ya está disponible como formato libre de patentes y se convierte en recomendación del W3C. CompuServe anuncia que su próxima versión de GIF (GIF24) abandonará el método LSW y se basará en la especificación PNG. En junio del 2003

la patente de LSW expira en Estados Unidos pero Unisys advierte que sigue vigente en otros países. Según *The Software Freedom Law Center* a partir de octubre de 2006 ya no quedan patentes relevantes en vigor sobre LSW.

El resultado de dicho proceso es enormemente positivo. Actualmente disponemos de un nuevo formato (PNG) mucho mejor que GIF y libre de restricciones para su desarrollo y uso. El único inconveniente para la implantación total del uso de PNG en la web era que los navegadores antiguos no lo interpretaban. Pero el problema ha desaparecido con las nuevas versiones.

Concretamente, Microsoft Internet Explorer no lo soportó hasta su versión 4 y no interpretó sus características de forma aceptable (como transparencia alfa, etc.) hasta las versiones 6 y 7. Netscape Navigator también empezó a soportarlo en la versión 4 pero no es hasta la versión 6 que soporta las nuevas características del formato. Firefox nació ya soportando plenamente este formato. La ventaja en este proceso es que el sistema de codificación de PNG implementa cada nueva característica en partes separadas, de forma que un navegador que no las interprete no deja de mostrar la imagen.

Como vemos, no todos los formatos de archivo son libres de restricciones. Y esto puede tener relación, como en este caso, con un sistema de compresión de datos que utiliza el formato en cuestión. El caso de PNG es en parte una excepción, pero también lo es el de GIF. En general los propietarios de los formatos no suelen restringir su uso ni cobrar por él, aunque en muchos casos no permiten su desarrollo a otras personas u organizaciones.

#### 4.4. Software y formatos

Además de los formatos que hemos visto hasta aquí, hay otros formatos de gráficos que se usan cuando trabajamos con ellos en un programa de edición. Son los **formatos nativos** o **formatos fuente** del programa.

Los programas de edición y tratamiento de gráficos necesitan un formato para guardar la información mientras no se ha terminado el trabajo. Generalmente se trata de información adicional propia del programa que permite guardar un archivo para seguir trabajando con él, a modo de proyecto no terminado que todavía contiene información que después no será visible en la imagen final (por ejemplo, la información de capas). También, para no perder esta información adicional, la mayoría de programas permiten guardar o exportar en varios formatos distintos al formato nativo. De esta forma podemos publicar un gráfico en formato PNG o JPEG en una web mientras mantenemos una versión en formato nativo de nuestro editor de gráficos donde la imagen tiene varias capas y el texto se mantiene editable, por si tenemos que hacer modificaciones.

Algunos de estos formatos se acaban convirtiendo en estándares porque mejoran sensiblemente la forma de guardar información, porque introducen características nuevas o porque el programa es un líder en su sector. Es el caso, no como formato de publicación pero sí como formato de intercambio entre programas, del formato nativo de Photoshop (que tiene la extensión "\*.psd"). Programas como Gimp o Fireworks abren los archivos de este formato y recuperan la estructura de capas y otros datos de trabajo almacenados.

El problema con los formatos nativos es que suelen mantenerse como propiedad de la empresa desarrolladora lo que implica llegar a acuerdos con ella o bien hacer ingeniería inversa para generar dicho formato desde software de la competencia. Además suelen aparecer nuevas versiones con cada nueva versión del programa y no siempre son compatibles con las anteriores.

El caso contrario es que un nuevo software adquiera como nativo un formato ya existente. Es, por ejemplo, el caso de Fireworks con PNG. Como PNG es un formato libre, Macromedia, que entonces era la empresa desarrolladora, pudo incorporarle características propias como la aceptación de gráficos vectoriales que luego pueden ser recuperados desde el propio programa o desde Flash. Aunque el uso de la misma extensión puede confundir a los usuarios Puesto que no es lo mismo un formato PNG nativo de Fireworks que el estándar PNG, y ambos usan la misma extensión "\*.png". Este es un caso, y no el único, en el que la extensión no nos informa suficientemente bien sobre el contenido de un fichero.

En el caso de GIMP su formato nativo (*XCF*, *eXperimental Computing Facility*), al ser libre, se ha difundido como formato de exportación o importación para otros programas de edición gráfica también libres.

Hay formatos de publicación que también han nacido vinculados a un programa. No son formatos fuente (no guardan información de trabajo) pero sí son en cierta forma formatos nativos, ya que están vinculados al tipo de información que el programa genera. Es el caso de formatos que hemos citado como SWF y PDF que siguen vinculados al desarrollo y actualización de Flash y Acrobat, respectivamente. Y ese vínculo sigue siendo importante aunque el formato se convierta en un estándar oficial (como en el caso de PDF).

#### **4.5. Información rápida sobre formatos**

En la tabla siguiente se recoge de forma esquemática la información básica de algunos de los formatos más usados.

Formato	Características
<b>JPEG</b> <i>Joint Picture Expert Group</i>	Extensión *.jpeg *.jpg. Mapa de bits. Modo RGB. 24 bits. Método propio de compresión con pérdida de datos. No admite transparencias. Admite carga progresiva. Adecuado para imágenes de tono continuo como las fotográficas. Es capaz de reducir mucho el tamaño de los archivos. Adecuado para la web.
<b>GIF</b> <i>Graphical Interchange Format</i>	Extensión *.gif. Versiones: GIF87, GIF89a y GIF animado. Mapa de bits. Modo indexado. 8 bits. Método de compresión LZW sin pérdida de datos. Admite transparencia de índice (GIF89a). Admite carga entrelazada (GIF87 y GIF89a). Admite secuencia de fotogramas (GIF animado). Desarrollado y propiedad de Compuserve. Método de compresión LZW patentado por Unisys. Adecuado para imágenes con pocos colores y/o colores planos. Adecuado para la web.
<b>PNG</b> <i>Portable Network Graphics</i>	Extensión *.png. Versiones: PNG-8, PNG-24 y PNG-32. Mapa de bits. Modo indexado (PNG-8) o RGB (PNG-24 y PNG-32). Versiones del formato para 8, 24 y 32 bits. Método de compresión deflate/inflate sin pérdida de datos. Admite transparencia de índice (PNG-8) y transparencia Alfa (PNG-32). Admite cargas entrelazadas (Adam7). Desarrollado por programadores independientes del movimiento de software libre. Adecuado tanto para colores planos como para tono continuo. Adecuado para la web. No se visualiza o se visualiza mal en navegadores antiguos.
<b>TIFF</b> <i>Tagged Image File Format</i>	Extensión *.tiff *.tif. Mapa de bits. Modos RGB, CMYK, CIE L*a*b y escala de grises. 32 bits. Método de compresión LZW opcional. Desarrollado por Aldus y Microsoft. Propiedad de Adobe. Adecuado para impresión. Requiere bastante memoria de almacenamiento.
<b>BMP</b> <i>Windows Bitmap Format</i>	Extensión *.bmp. Mapa de bits. Modo indexado y RGB. 24 bits o menos. Desarrollado y propiedad de Microsoft Corp. Comprime poco pero se abre rápidamente. Adecuado para gráficos a visualizar en pantalla. No se visualiza en los navegadores web.
<b>PICT</b> <i>Macintosh Picture</i>	Extensión *.pic *.pict. Meta fichero. Modo indexado y RGB. 24 bits o menos. Desarrollado y propiedad de Apple Computer. Aunque es un meta fichero se suele usar para mapas de bits. Rápido en descompresión. Adecuado para gráficos a visualizar en pantalla y también como formato de intercambio entre programas.

Formato	Características
<b>Adobe Photoshop</b> (formato nativo)	Extensión *.psd. Mapa de bits. Modos CIE L*a*b, RGB, CMYK, indexado, duotono, escala de grises y pluma. 32 bits. Admite canales alfa y capas, además de otras informaciones del programa. Desarrollado y propiedad de Adobe Systems. Adecuado como formato de trabajo en Photoshop o para transmitir archivos de este programa a otro que lo interprete.
<b>WMF</b> <i>Windows Meta File</i>	Extensión *.wmf. Meta fichero. Desarrollado y propiedad de Microsoft Corp. Aunque es un meta fichero se usa sobre todo para gráficos vectoriales. Adecuado como formato de intercambio entre programas.
<b>EMF</b> <i>Enhanced Windows Metafile</i>	Extensión *.emf. Meta fichero. Versión mejorada de WMF. Desarrollado y propiedad de Microsoft Corp. Aunque es un meta fichero se usa sobre todo para gráficos vectoriales. Adecuado como formato de intercambio entre programas.
<b>EPS</b> <i>Encapsulated PostScript</i>	Extensión *.eps. Vectorial, también admite mapas de bits con trazados. Utiliza el lenguaje Postscript de descripción de vectores. Desarrollado y propiedad de Adobe Systems. Adecuado para impresión.
<b>PDF</b> <i>Portable Document Format</i>	Extensión *.pdf. Vectorial, admite mapas de bits incrustados. Evolución del EPS. Conserva la información de las tipografías. Tiene la posibilidad de añadir hipervínculos e interactividad. Requiere de un programa de visualización o un <i>plug-in</i> de navegador para visualizarse. En la web se visualiza independientemente de la página HTML. Desarrollado y propiedad de Adobe Systems. Es adecuado para impresión y también para la distribución de documentos electrónicos imprimibles por Internet. En 2008 fue reconocido como estándar ISO.
<b>SWF</b> <i>Shockwave Flash</i>	Extensión *.swf. Admite gráficos vectoriales, archivos de mapa de bits incrustados, animaciones y elementos interactivos. Requiere un <i>plug-in</i> para visualizarse en la web y Flash Player para ejecutarse en disco. Se visualiza integrado ( <i>embed</i> ) en la página HTML. Desarrollado por Future Wave Software, Macromedia y Adobe. Propiedad de Adobe. Es adecuado para gráficos vectoriales, películas animadas y aplicaciones interactivas, tanto para la web como para reproducción directa de disco.
<b>SVG</b> <i>Scalable Vector Graphics</i>	Extensión *.svg. Vectorial, admite animaciones y programación. Basado en XML y por lo tanto compatible con HTML. No requiere ningún tipo de <i>plug-in</i> para visualizarse sobre HTML, aunque a la práctica lo necesita, ya que los principales navegadores (MS-Explorer y Netscape) aún no lo soportan. Homologado e impulsado por el W3 Consortium. Desarrollado por un consorcio de empresas y organizaciones. Formato libre. Adecuado para gráficos vectoriales y sistemas interactivos en la web. Es el estándar abierto para gráficos vectoriales en Internet. Es una "recomendación" (equivalente a estándar) del W3C desde 2001.

## 5. Software de creación y tratamiento gráfico

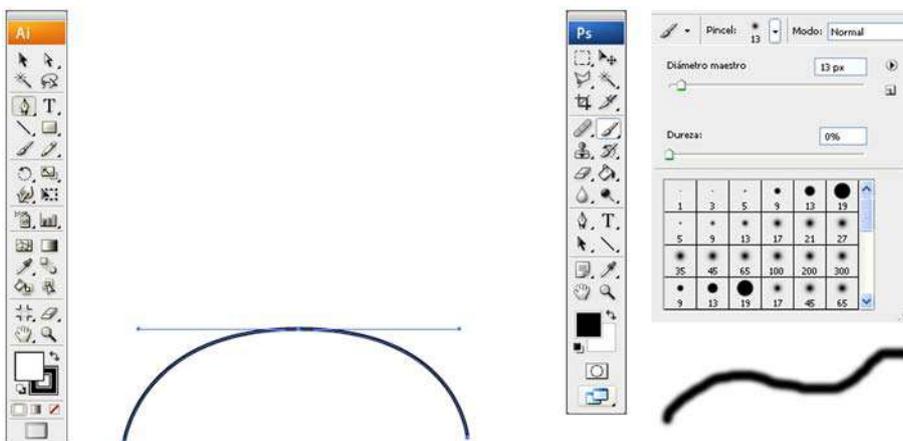
Afortunadamente la oferta de programas de creación y tratamiento de gráficos es amplia y variada. En esta sección se ofrece una panorámica de esta oferta, proporcionando conocimientos y orientaciones para elegir la herramienta en función del tipo de tarea a realizar y las condiciones de trabajo.

Para presentar el software de gráficos se revisan sus características a partir de tres binomios:

- software bitmap y vectorial
- software orientado a impresión y orientado a la web
- software propietario y *open-source*

Cada programa puede adscribirse a varias de estas categorías. Pero su estudio por separado nos puede proporcionar una mayor comprensión.

### 5.1. Software bitmap y vectorial



La pluma "Bézier" y el pincel "de píxeles" son el paradigma de dos formas de trabajar distintas.

Anteriormente nos hemos referido a la importancia de la distinción bitmap/vectorial. Esta distinción también se da en el campo de los programas de gráficos. Hay programas de gráficos vectoriales que trabajan con objetos Bézier y programas de gráficos de mapa de bits que trabajan con imágenes compuestas por píxeles. Esto se traduce en la configuración de los entornos de trabajo.

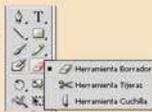
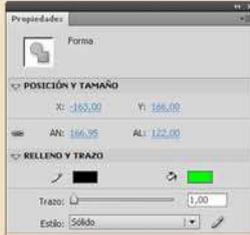
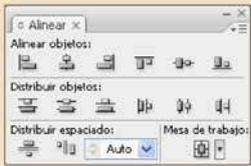
Los **entornos de trabajo vectoriales** están más orientados al dibujo geométrico; en cambio los **entornos de mapa de bits** tienen un planteamiento más orientado al dibujo "gestual". De hecho una distinción que se hacía en los primeros programas de gráficos en la década de los ochenta era entre programas "de dibujo" (vectoriales) y programas "de pintura" (mapa de bits). Otra referen-

cia para los programas de mapa de bits es la fotografía y el retoque fotográfico, mientras que los vectoriales pueden tomar como referente el diseño gráfico, la ilustración y el *collage*.

Aunque, como veremos, estas formas de trabajar acaban hibridándose; el imaginario de referencia marcó los primeros desarrollos de entornos de trabajo y nos ha dejado un legado de metáforas que aún se mantiene.

### 5.1.1. Entornos de trabajo vectorial

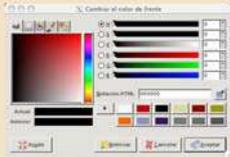
En la tabla siguiente se muestran los principales elementos de trabajo en un programa de gráficos vectoriales.

Principales elementos de trabajo en un programa de gráficos vectoriales	
Herramientas para objetos Bézier	<p>Para crear objetos Bézier disponemos de una herramienta que crea los nodos (<b>pluma</b>), herramientas de selección (<b>flechas</b>) para moverlos o extraer y controlar los manejadores y de una <b>cuchilla</b> para cortar las líneas en objetos independientes. También disponemos de una herramienta para crear líneas rectas y de otra para crear directamente figuras geométricas como rectángulos, elipses y polígonos.</p> 
Herramientas de transformación	<p>Los objetos creados pueden ser transformados geoméricamente. Disponemos de herramientas y/o procedimientos para <b>rotar, desplazar, escalar, sesgar o reflejar los objetos</b>. También disponemos de opciones de menú para combinar entre sí los objetos. La forma de hacerlo difiere de un programa a otro.</p> 
Control de propiedades	<p>Los objetos son independientes entre sí y tienen una serie de propiedades que son parametrizables. De una forma u otra los programas nos ofrecen una forma de cambiar dichas propiedades seleccionando el objeto en cuestión y accediendo a ellas a través de una ventana. Las propiedades se refieren al ancho y tipo de línea, al <b>relleno del objeto</b>, a su <b>tamaño y posición</b>, etc.</p> 
Alineación	<p>La alineación de los objetos respecto a los otros también se puede controlar. Disponemos de un sistema de control por el que seleccionando varios objetos podemos <b>alinearlos centrados, a la izquierda, a la derecha o distribuirlos de forma que el espacio entre ellos sea igual</b>.</p> 
Capas	<p>Los programas vectoriales pueden trabajar con capas. En cada capa hay un conjunto de objetos. Las capas pueden <b>ocultarse o bloquearse</b> para no mover algún objeto por error.</p> 
Color	<p>El color en un programa vectorial es una propiedad del objeto. Podemos seleccionar un objeto y <b>asignarle un color</b>. También se pueden <b>crear nuevos colores</b> usando los selectores de color. Pero atención: a cada color le corresponde una codificación (CMYK, RGB, color de librería PANTONE,...). Cada objeto podría tener una codificación distinta y esto sería problemático, por ejemplo, si hay que imprimir la imagen. Es importante tener claro con que codificación de color se va a trabajar.</p> 

Con algunas variaciones, este es el entorno de trabajo en programas como Illustrator, Corel Draw, Inkscape, Xara Xtreme, Fireworks o Freehand. De hecho, en Fireworks cambian algunas cosas y más aún en Flash, como veremos más adelante.

### 5.1.2. Entornos de trabajo bitmap

En la tabla siguiente se muestran los principales elementos de trabajo en un programa de gráficos de mapa de bits.

Principales elementos de trabajo en un programa de gráficos de mapa de bits			
<p><b>Herramientas para dibujo y pintura</b></p> <p>Para dibujar píxeles las principales herramientas son el <b>lápiz</b>, la <b>brocha</b>, la <b>goma de borrar</b>, el <b>aerógrafo</b> y el <b>bote de pintura</b> que rellena áreas. Además disponemos de control de parámetros para el lápiz y la brocha (forma, dureza...). La herramienta de <b>cuantagotas</b> permite escoger un color de la imagen.</p> 	<p><b>Canales</b></p> <p>Los programas de mapa de bits representan el color a través de canales. En cada canal hay un componente de color. Por ejemplo, en el modo RGB tenemos tres canales: rojo, verde y azul. Es posible visualizar sólo un canal y trabajar únicamente con ese componente en escala de grises.</p> 		
<p><b>Herramientas de selección</b></p> <p>Para seleccionar áreas de píxeles están las herramientas de selección, como la <b>selección regular</b> (rectángulo o elipse), el <b>lazo</b> para selecciones irregulares y la <b>varita mágica</b> para seleccionar píxeles similares contiguos o de toda la imagen. Las herramientas de selección se complementan con una serie de opciones de menú relacionadas con la selección.</p> 	<p><b>Ajustar</b></p> <p>Los programas de mapa de bits tienen un referente en el retoque fotográfico de laboratorio. Por eso nos brindan herramientas precisas para <b>modificar y ajustar</b> la imagen.</p> 		
<p><b>Selector de color</b></p> <p>El <b>selector de color</b> sirve para escoger el color con el que pintar. Se ofrecen varios sistemas para definir el color (HSV, RGB, CMYK, L*a*b, etc.).</p> 	<p><b>Filtros algorítmicos</b></p> <p>La disposición de los píxeles en la retícula puede ser modificada aplicándoles un algoritmo, pues al fin y al cabo, se trata de datos numéricos. Hay una gran diversidad de filtros algorítmicos que crean efectos determinados en la imagen interviniendo en el color de los píxeles. Los programas de gráficos incorporan la posibilidad de aplicar dichos filtros que muchas veces son pequeños programas externos que se utilizan desde el programa.</p> 		

### 5.1.3. Confluencias

Como se ha visto anteriormente, el tipo de codificación de la imagen (vectorial o mapa de bits) suele condicionar el tipo de interfaz del entorno de trabajo y una determinada forma de trabajar.

Ahora bien, con los años se ha visto que una interfaz pensada para el dibujo vectorial podía ser útil en un programa de mapa de bits y viceversa. Ha empezado una convergencia de procedimientos de trabajo entre los dos tipos de programas.

Uno de los ejemplos más claros es el uso de las **capas** en **programas de mapa de bits**, que no tenían inicialmente y que adoptaron bajo la influencia de la experiencia de trabajo con programas vectoriales. Con un sistema de capas una parte de una imagen puede mantenerse independiente del resto (como si fuera un objeto vectorial) y aplicarle ajustes o filtros de forma autónoma.

Otro elemento que los programas de mapa de bits han adoptado es un buen **control tipográfico**, que al principio se reservaba a los programas vectoriales.



Ventana de capas en Photoshop. Distintas partes de la imagen están en capas diferentes. Las capas se pueden ocultar y también enlazar entre sí.

Una influencia en dirección contraria es la del **pincel** y la **goma** por parte de programas vectoriales. El caso más claro de ello es Flash. Con estas herramientas se consigue que el usuario pueda dibujar de una forma mucho más intuitiva sin apenas darse cuenta de que está generando objetos Bézier. Las "líneas" del pincel no son tales, en realidad se están generando áreas, objetos cerrados.

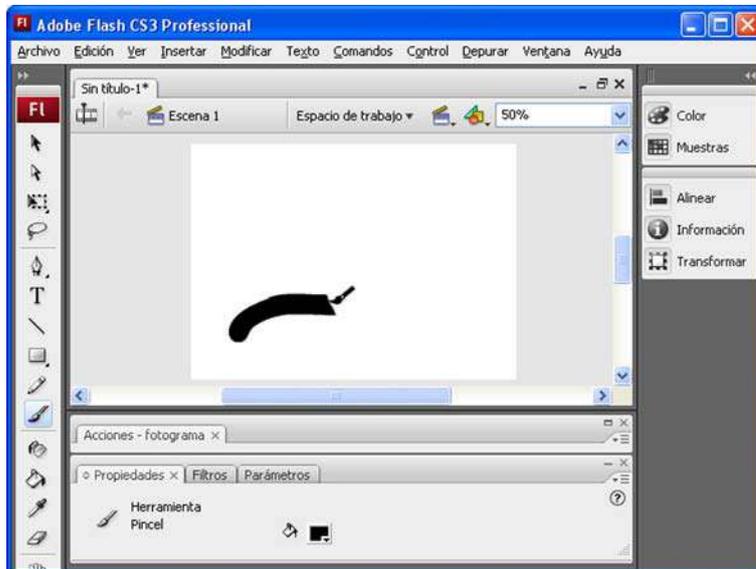


Ilustración realizada con el pincel de Macromedia Flash.

Un caso paradigmático de dicha confluencia es el programa de gráficos **Fireworks**. En principio se trata de un programa de dibujo vectorial que, como la mayoría de estos, admite la importación de gráficos de mapa de bits. Pero no es sólo eso, Fireworks permite crear directamente gráficos de mapa de bits y modificarlos. De hecho, Fireworks integra todas las características propias de un programa de mapa de bits y de un programa de dibujo vectorial. El mejor ejemplo es su ventana de herramientas con un grupo para la edición en mapa de bits y un grupo para la edición vectorial. En su entorno de trabajo conviven elementos vectoriales con elementos de mapa de bits. Su ventana de propiedades es contextual y cambia en función del elemento elegido.

La elección sobre la codificación definitiva de la imagen se toma al final. Fireworks trabaja en una versión de PNG nativa pero para grabar cualquier otro formato tiene que exportar. Una ventana de optimización permite configurar las características cuando se quiere grabar en un formato de mapa de bits.

## 5.2. Software orientado a impresión y orientado a la web

La finalidad última de los gráficos a generar condiciona las características del programa a elegir para su edición.

El proceso de producción puede ser significativamente distinto si el destino final es un producto impreso de cuando es un producto que se distribuirá en entornos digitales y de visualización en pantalla. Y esto determina las herramientas. Dependiendo de la aplicación final de la imagen las características del software editor serán distintas.

En el caso del gráfico que tiene que imprimirse, nos preocupa ver en pantalla una "simulación" lo más aproximada posible a lo que será el resultado en papel. El software tiene que simular los efectos de la superposición de tintas, de la cuatricromía, etc. En cambio el gráfico que tiene que verse en pantalla tiene que optimizarse para reducir el tamaño del fichero (peso del archivo) y tiene que comprimirse sin perder calidad. El programa de gráficos tiene que tener herramientas para dicha optimización.

Muchos editores de gráficos están pensados para satisfacer las dos necesidades, no sólo para abarcar un abanico más amplio de funcionalidades, también porque muchos gráficos tienen que tener las dos aplicaciones (impresa y en pantalla). Pero aun en estos casos suele haber un punto, en el que el proceso de producción se bifurca generando dos versiones del gráfico y usando recursos distintos del programa.



Herramientas de Macromedia Fireworks

En general el destino final condiciona el entorno de trabajo y los productos de referencia que lo configuran suelen ser la **publicación impresa** y la **publicación en web**.

### 5.2.1. Software orientado a impresión



Interfaz de trabajo de Illustrator con los iconos de cambio de página, las reglas y las guías. El símbolo es la mesa del diseñador gráfico, su espacio de trabajo.

Los programas orientados a impresión utilizan la metáfora de la página para el entorno de trabajo. Se trabaja con medidas físicas (centímetros, pulgadas, etc.), se dispone de reglas de página, de guías y retículas que ayudan en la maquetación (disposición de las columnas de texto, gráficos, titulares, etc.). Se puede definir un área de "sangrado" fuera de la página en la que extendemos los elementos que quedarán en el límite de corte de página para que no quede una línea blanca.

Se dispone de un buen control tipográfico que incluye, además de la elección de la fuente y el tamaño, definición de interletrado (o *kerning*), interlineado, parámetros de párrafo, alineación, línea de base, etc.

Otro factor importante es determinar la densidad de trama con la que se van a imprimir las imágenes de tono continuo (fotografías, determinadas ilustraciones, etc.). En cuanto a los colores elegidos permite escoger entre cuatricromía o tintas planas y definir aspectos como la posibilidad de que una tinta se superponga a otra o bien se cree una "reserva".

#### Cuatricromía y tintas planas

La **cuatricromía** es un sistema de representación del color que a partir de un tramado de cuatro tintas consigue simular una amplia variedad de colores. Los sistemas de cuatricromía usan 3 colores básicos (cian, magenta, amarillo) más el negro, y usan el blanco del papel como quinto color. Con la combinación de tramas de las cuatro tintas se consigue reproducir fotografías y cualquier tipo de imágenes.



Propiedades de texto en Illustrator

En artes gráficas se habla de **tintas planas** o **tintas directas** cuando en vez de cuatricromía se usan tintas de colores específicos, normalmente para elementos gráficos con colores sólidos, aunque también se puede usar en tramados. Las tintas directas son mezcladas por el fabricante o las mezcla el impresor siguiendo las indicaciones de un catálogo estandarizado de tintas. Hay unos pocos fabricantes internacionales de catálogos de tintas. El más usado es PANTONE. Otros catálogos son: Letraset, Toyo, DIC o Trumatch.

#### Ved también

Amplia información sobre la mezcla partitiva en la que se basa el sistema de cuatricromía en los subapartados "Mezclas: aditiva, sustractiva y partitiva" y "Codificación digital del color" del apartado "Color y diseño gráfico" del módulo "Conceptos básicos de diseño gráfico".

Los programas orientados a impresión suelen ser programas vectoriales que admiten gráficos de mapa de bits incrustados y que tienen diversas herramientas para la maquetación y, en algunos casos, para la edición. Son programas como: Illustrator, Corel Draw, Xara Xtreme, In Design, QuarkXPress, Scribus o algunos discontinuados en su desarrollo pero que en su momento fueron relevantes, como Freehand o PageMaker. Estos programas tienen además controles extra de impresión como la posibilidad de ordenación de páginas, el control de gráficos vinculados y fuentes tipográficas usadas, la inserción de marcas de corte y de registro, el control de tintas y separación de colores, etc.

Existen también programas de mapa de bits orientados a crear imágenes que deberán insertarse en uno de dichos programas o bien imprimirse directamente. Estos programas deberán permitir el modo de trabajo en CMYK o tintas planas, controlar la densidad de trama y guardar con un formato orientado a impresión como TIFF. Photoshop o GIMP son programas que pueden cumplir estos objetivos. Fireworks es un programa que sobre una base vectorial permite trabajar en mapa de bits y ser usado para una finalidad similar a la que hemos descrito.

### 5.2.2. Software orientado a web

En la creación de gráficos para la web o, en general, para ser visualizados en pantalla, las necesidades son muy distintas a las descritas para imágenes cuyo destino es un producto impreso.

Las unidades de medida son generalmente los píxeles. Ya no tiene sentido trabajar en términos de "página", y en un sistema de medida física, el tamaño del área de trabajo tiene que ver con el tamaño final de la imagen en píxeles en el dispositivo de salida.

Las características del entorno de trabajo también son distintas si vamos a generar gráficos vectoriales o gráficos de mapa de bits. En la generación de gráficos de mapa de bits para la web es muy importante la optimización de los mismos. Los programas de gráficos introducen controles de optimización que permiten controlar la calidad y el número de colores, establecer transparencias, etc.; asimismo, pueden simular, en tiempo real, el efecto de dicha optimización sobre la imagen.

Los programas orientados a web también pueden tener opciones para crear un GIF animado y para determinar si la imagen debe cargarse en entrelazado o no. Ejemplos de programas que realizan estas funciones son Fireworks o el módulo de Photoshop llamado Image Ready.

Otro elemento importante de estos programas es el tratamiento que hacen del color. Se trabaja en modo RGB pero se permite seleccionar el color a través del código hexadecimal que usa HTML. Además se incluye la posibilidad de trabajar con la paleta web, una paleta de 216 colores consensuada por todo el sector, y en el selector de color puede darse la posibilidad de mostrar sólo los colores de esta paleta.

Si se tiene que crear una imagen vectorial para la web (en formato SWF o SVG) las necesidades también son distintas. Los programas están provistos de todas las herramientas de creación vectorial pero además permiten suavizados que mejoran la visualización en pantalla.

Los formatos vectoriales para Internet incluyen posibilidades de animación y programación. Por ello, programas como Flash, Silverlight o Gnash tienen un entorno de trabajo con línea de tiempo, capas y fotogramas pensados para la animación, así como sistemas de programación para añadir interactividad. También los programas inicialmente desarrollados para impresión, como Illustrator, han añadido la posibilidad de generar gráficos para la web y exportar a los formatos citados. Existe también el caso ya mencionado de Fireworks que permite crear gráficos para web, tanto vectoriales como de mapa de bits, con un entorno de trabajo muy orientado a facilitar la optimización de estos últimos.

### 5.3. Software propietario y *open-source*

#### 5.3.1. El software

Los ordenadores sólo procesan **código binario**, instrucciones en ceros y unos. Pero los programadores no escriben directamente en este lenguaje-máquina. Escriben en **lenguajes de alto nivel**, lenguajes lógicos, con rígidas reglas sintácticas y algunos parecidos con el lenguaje escrito (lenguajes como C, Pascal, Java, etc.). Estos lenguajes se traducen a **lenguaje binario de bajo nivel** para que el ordenador lo ejecute. A lo que el programador escribe se le llama **código fuente**, a lo que el ordenador procesa **código binario**.

#### Ved también

Sobre la optimización de gráficos para la web ver la sección dedicada a ello del módulo "Producción de gráficos".

El **software** es un conjunto de instrucciones programadas que permiten hacer algo. El software puede distribuirse sólo con el código binario para ser utilizado o con el código fuente incluido para saber como está programado y/o modificarlo.

### 5.3.2. La aparición del software propietario

En las primeras décadas de la informática el software se desarrollaba en las universidades y centros de investigación. El código fuente estaba disponible para cualquier desarrollador que quisiera mejorarlo.

Pero a principios de los ochenta aparece el ordenador personal que se extiende por oficinas y hogares. Los programas empiezan a venderse como producto independiente del ordenador y sin el código fuente. Crece una industria del software que esconde el código fuente para evitar ser copiado por la competencia. El software se convierte en propietario y se protege a través de la legislación de *copyright* y las patentes.

En el campo del diseño y las artes gráficas, junto a la aparición del Macintosh en 1984, empresas como Aldus y Adobe se introducen en los estudios de diseño con programas como Page Maker, Freehand, Illustrator y Photoshop. El sistema operativo del Macintosh es propiedad de la empresa Apple. Otra empresa, Microsoft Corp., propietaria del sistema MS-DOS extendido en la mayoría de PC compatibles introduce un sistema parecido, el Windows, aún dominante, sentando las bases de un monopolio. Serán también actores relevantes en el campo del software de edición gráfica Quark Inc., desarrolladora de QuarkX-Press, y Macromedia, desarrolladora de Flash, Dreamweaver y Fireworks entre otros. En 1994 Adobe adquiere a su competidora Aldus y en 2005 hará lo mismo con Macromedia.

Ante esta situación el usuario se ve comprometido y obligado a no distribuir el software comprado bajo férreas licencias. Además depende de la empresa propietaria para que se corrijan errores o se implementen nuevas características, ya que él no dispone del código fuente.

### 5.3.3. El proyecto GNU

A mediados de los ochenta, un grupo de programadores insatisfechos con esta situación de monopolios de software propietario empiezan un proyecto para, en sus palabras, "liberar" el software. Richard M. Stallman, que había trabajado en el MIT en el desarrollo del sistema operativo UNIX y herramientas para este, se siente engañado por su privatización e inicia el proyecto GNU; acrónimo recursivo que significa *Gnu's Not Unix* (Gnu No es Unix).

Quieren crear un nuevo sistema operativo equivalente a Unix pero libre. Su tarea es ardua porque las herramientas que necesitan (programas de texto, compiladores, etc.) también son propietarias. Por lo tanto empiezan por ahí, por crear esas herramientas. Crean también la *Free Software Foundation* (FSF) para velar por el proceso.

#### 5.3.4. La licencia GPL

Uno de los mayores aciertos del proyecto GNU es el haber sabido poner de su parte las leyes de *copyright* creando la *General Public Licence* (GPL). Esta licencia se basa en los propios derechos de autor y de reproducción para definir los términos bajo los que se distribuirá un programa.

Un software con licencia GPL debe ser distribuido junto al código fuente, además cualquier persona puede distribuirlo libremente y modificarlo si quiere. La GPL establece que cualquier modificación del software deberá estar obligatoriamente bajo dicha licencia. Por lo que se ha dicho que la GPL contamina todo lo que toca. Cualquier desarrollo basado en software libre seguirá siendo libre para no infringir la licencia con la que se distribuye.

#### 5.3.5. El movimiento del software libre

Actualmente el movimiento de programadores que desarrollan software libre de código abierto (*open-source*) es un sistema auto-organizado eficaz, que dispone de un buen sistema operativo (Linux) que funciona en la mayoría de servidores de Internet, además se va incorporando cada vez más a hogares y oficinas, tiene múltiples herramientas de software, así como un sistema eficiente de desarrollo y corrección de fallos.

Algunas importantes empresas (como IBM, Sun o Novell, por citar algunas) contribuyen a este proceso. Y un gran número de pequeñas empresas, programadores independientes y equipos informales también lo hacen. Compañías como Google participan también en la financiación de proyectos (con el *Google Summer of Code* desde 2005).

En el campo del software de gráficos es especialmente relevante para el impulso de los proyectos y el intercambio de soluciones el *Libre Graphics Meeting*, una convención que desde 2006 reúne anualmente a los desarrolladores de programas de gráficos libres. Juegan también un papel relevante la compañía Xara, desarrolladora del software vectorial Xara Xtreme, y la Blender Foundation, que coordina el desarrollo del software de animación y gráficos 3D Blender; además de los equipos de desarrollo en torno a programas como Gimp, Inkscape o Scribus.

### 5.3.6. Software libre para gráficos

En este momento, además de **Linux** como sistema operativo, hay buenas herramientas de software libre para servidores, para programación, para edición de textos y software de oficina, navegadores de Internet, etc. ¿Qué pasa con el software de edición de gráficos?

Existe una buena herramienta de gráficos de mapa de bits: **The GIMP** (*The GNU Image Manipulation Program*). Se trata de un programa con funcionalidades básicas equivalentes a Photoshop disponible para las plataformas Linux, Windows y Macintosh. En cuanto a gráficos vectoriales, existen diversos programas entre los que cabe destacar **Inkscape** y **Xara Xtreme** (orientados al dibujo) y **Scribus** (orientado a la maquetación editorial).



Los defensores del software propietario argumentan que un desarrollo de herramientas como el que se ha producido en el mundo del diseño en las últimas décadas sólo es posible con una fuerte industria del software, financiada por la venta de licencias de programas, que responda a las necesidades del sector. Pero también se argumenta que dicha industria muchas veces no invierte sus esfuerzos en corregir los errores, sino en añadir nuevas funcionalidades innecesarias que justifiquen la aparición de una nueva versión del programa que los usuarios deberán actualizar mediante pago.

En la situación en que quedó el mercado tras la adquisición de Macromedia por parte de Adobe en 2005, esta compañía se convirtió en el actor principal en el desarrollo de software propietario en el campo del diseño, con programas como Photoshop, Fireworks, Flash, Illustrator, In Design y Dreamweaver, entre otros. Adobe mantiene la venta de licencias de sus principales programas mientras, especialmente a partir de 2008 con el anuncio de su *Open Screen Project*, empieza una política de apertura de las especificaciones de sus formatos como SWF, FLV o PDF.

Compañías de Internet como Google se están introduciendo en el desarrollo de herramientas de gráficos discretamente, empezando por herramientas orientadas al usuario doméstico como **Picasa**. Es un programa propietario que se ofrece gratuitamente para organizar y editar fotografías.

El desenlace y/o evolución de esta situación depende de las actuaciones de los diversos agentes implicados y, en buena medida, de la actitud que tomen los usuarios.

## **Bibliografía**

### **Bibliografía básica**

**Gordon, B.; Gordon, M.** (2007). *Manual de diseño gráfico digital*. Barcelona: Gustavo Gili.

**Fuenmayor, E.** (1996). *Ratón, ratón*. Barcelona: Gustavo Gili.

**Blasco, L.** (2011). *Sobreimpresión de la pantalla al papel y viceversa*. Barcelona: Index Books.

### **Bibliografía recomendada**

**Simons, J.** (2007). *Manual del diseñador*. Barcelona: Index Books.

