

# Gràfics digitals

Jordi Alberich  
Albert Corral  
David Gómez Fontarills  
Alba Ferrer Franquesa

PID\_00156474



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



*Llicència Creative Commons, versió 2.5, modalitat BY-SA (attribution - share alike), que permet modificar l'obra, reproduir-la, distribuir-la o comunicar-la públicament sempre que se'n reconegui l'autoria i sempre que l'obra derivada quedi subjecta a la mateixa llicència que el material original.*

# Índex

<b>1. Introducció als gràfics digitals</b> .....	5
1.1. Imatge digital enfront d'imatge analògica .....	5
1.1.1. Introducció: gràfics codificats .....	5
1.1.2. Canvis per als grafistes .....	7
1.1.3. Dispositius d'entrada i sortida de dades .....	7
1.2. Digitalització .....	8
1.2.1. Escàner .....	8
1.2.2. Càmeres digitals .....	9
1.3. Dispositius d'entrada per a creació de gràfics .....	11
1.4. Dispositius de sortida de gràfics .....	12
1.4.1. Impressora .....	12
1.4.2. Pantalla del monitor .....	13
<b>2. Gràfics de mapa de bits</b> .....	15
2.1. Una retícula de píxels .....	15
2.2. Resolució de la imatge .....	15
2.3. Resolució de pantalla .....	17
2.4. Profunditat de color .....	19
2.5. Eines d'edició de gràfics de mapa de bits .....	20
2.5.1. Eines de dibuix .....	21
2.5.2. Eines de selecció .....	23
2.5.3. Filtres algorítmics .....	27
2.5.4. Ajustaments d'imatge .....	27
<b>3. Gràfics vectorials</b> .....	29
3.1. Dibuix basat en objectes .....	29
3.1.1. Objectes Bézier .....	29
3.1.2. Bézier en el programari gràfic .....	31
3.1.3. Treball de precisió amb les corbes de Bézier .....	32
3.2. Propietats dels objectes vectorials .....	34
3.2.1. Propietats de la línia o filet .....	34
3.2.2. Transformacions dels objectes vectorials .....	36
3.2.3. Composició de gràfics vectorials .....	38
3.3. Mà alçada: dibuixar vectors sense nodes ni agafadors .....	39
3.4. La resolució de les imatges vectorials .....	40
3.5. Quan podem utilitzar gràfics vectorials .....	41
<b>4. Formats gràfics</b> .....	43
4.1. Codificar gràfics .....	43
4.1.1. Tipus de format .....	44
4.2. Formats per a gràfics vectorials i metafítxers .....	44
4.2.1. Formats per a arts gràfiques .....	45

4.2.2.	Formats per a Internet .....	46
4.3.	Formats per a mapa de bits .....	47
4.3.1.	TIFF, el format d'impressió .....	48
4.3.2.	Formats per al web: GIF, JPEG i PNG .....	48
4.3.3.	Sistemes de compressió .....	50
4.3.4.	La polèmica LZW .....	52
4.4.	Programari i formats .....	53
4.5.	Informació ràpida sobre formats .....	54
<b>5.</b>	<b>Programari de creació i tractament gràfic.....</b>	<b>57</b>
5.1.	Programari <i>bitmap</i> i vectorial .....	57
5.1.1.	Entorns de treball vectorial .....	58
5.1.2.	Entorns de treball <i>bitmap</i> .....	59
5.1.3.	Confluències .....	61
5.2.	Programari orientat a impressió i orientat al web .....	62
5.2.1.	Programari orientat a impressió .....	63
5.2.2.	Programari orientat al Web .....	64
5.3.	Programari propietari i <i>open-source</i> .....	65
5.3.1.	El programari .....	65
5.3.2.	L'aparició del programari de propietat .....	66
5.3.3.	El projecte GNU .....	66
5.3.4.	La llicència GPL .....	67
5.3.5.	El moviment del programari lliure .....	67
5.3.6.	Programari lliure per a gràfics .....	67

# 1. Introducció als gràfics digitals

## 1.1. Imatge digital enfront d'imatge analògica

### 1.1.1. Introducció: gràfics codificats

La digitalització de dades introdueix un canvi significatiu en les possibilitats de manipulació i tractament. També treballar amb gràfics digitals és significativament diferent de fer-ho amb gràfics analògics. Una imatge analògica és, per exemple, la que podem traçar amb un llapis o un pinzell sobre un paper. L'anomenem analògica des de la perspectiva del digital. És analògica perquè no ha estat generada per un procés codificat ni s'emmagatzema de manera codificada.

#### Sketchpad d'Ivan Sutherland

Els sistemes de disseny assistit per ordinador, també coneguts per les sigles angleses CAD, són sistemes de programari i, de vegades, maquinari molt avançats. Però fa molt temps que rondem pel cap dels programadors, no són res de nou. Vet aquí a l'avi dels CAD actuals, el sistema Sketchpad.

L'Sketchpad va ser desenvolupat l'any 1963 per Ivan Sutherland com a part de la seva tesi doctoral. Introduïa alguns avenços que trigarien a desenvolupar-se com a eines d'ús comú, ja que ja hi havia una interfície gràfica d'usuari, GUI, moltes dècades abans que s'estenguessin per tot el planeta informàtic.

El sistema Sketchpad es va crear al Laboratori Lincoln del MIT sobre un ordinador TX-2, una de les millors màquines d'aleshores, ja que tenia 320 Kb de memòria base i 8 Mb de memòria externa en forma de cintes magnètiques. Quant a dispositius gràfics, estava dotat amb un monitor de set polzades  $1024 \times 1024$ , a més d'un punter òptic i una imitació de ratolí.



Fotògraf desconegut. Aquesta imatge es reproduïx acollint-se al dret de citació o ressenya (art. 32 LPI), i està exclosa de la llicència per defecte d'aquests materials.

Sketchpad, un programari de dibuix de gràfics d'enginyeria per a l'ordinador TX-2. Creat el 1963 per Ivan Sutherland com a part del seu doctorat en el Massachusetts Institute of Technology, va ser la primera interfície gràfica d'usuari per a ordinadors.

Durant anys, segles, mil·lennis, els gràfics han estat principalment analògics. Una imatge digital és una imatge codificada per mitjà de dígit (símbols arbitraris que no guarden una analogia amb el referent). Abans de la informàtica hi ha hagut imatges codificades d'aquesta manera com, per exemple, el teixit de jacquard, que usava targetes perforades per als telers (i que són l'antecedent directe de les targetes perforades dels primers ordinadors), o el brodat de punt de creu, que usa retícules i símbols per a codificar els colors.

Però quan es parla de gràfics digitals se sol fer referència a la tecnologia informàtica. Aquesta codifica la informació per mitjà d'un sistema numèric **binari** basat només en zeros i uns. La seva unitat bàsica és el **bit** (de l'anglès *binary digit*), que pot tenir un dels dos valors. Agrupant els bits en paquets de vuit (**bytes**) i assignant a cada un un valor numèric, es comença una cadena de sofisticació creixent amb la qual és possible emmagatzemar qualsevol informació. En conseqüència, el que tenim és un sistema de codificació de la informació que és independent del suport (es pot emmagatzemar en qualsevol suport que pugui donar zeros i uns, tant si és magnètic com òptic o de qualsevol altre tipus), i la informació es pot copiar amb total fidelitat. A diferència de la imatge analògica, en la imatge digital la còpia és igual que l'original perquè es copien nombres i no hi ha "soroll" que interfereixi en el procés.

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

En informàtica, *bit* és la contracció de l'expressió *binari digit* (**dígit binari**). Però, alhora, en anglès significa 'trosset' o 'porció'. El bit representa una variable binària en què cada dígit pren el valor de 0 o el valor d'1. Per tant, dins de l'ordinador, constitueix la unitat més petita en què es pot codificar la informació (paraula, nombre, instrucció). En conseqüència, qualsevol informació es mesura pel nombre de bits necessaris per a la seva representació en un sistema binari.

El sistema binari treballa sobre la base de 2, és a dir, únicament utilitza dues xifres per a expressar qualsevol quantitat. Aquestes xifres són els dos valors que pot tenir un bit: 0 i 1. El sistema en el qual treballen els ordinadors és binari. A nivell electrònic la màquina considera:

- **valor 0** = transmissió d'un senyal elèctric de tensió entre 0 volts i 0,0 volts.
- **Valor 1** = transmissió d'un senyal elèctric de tensió entre 0,8 volts i 4,5 volts.

Del bit deriven la resta d'unitats de mesura. Així l'element byte, conjunt de 8 bits, seguit de seus múltiples, el kilobyte, el megabyte, el gigabyte, el terabyte. L'augment de l'escala està determinat per la potència de 2 més pròxima a 1.000 que expressa la fórmula:

$$2^{10} = 1.024$$

Cada unitat manté aquesta relació amb l'anterior i la següent:

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. Canvis per als grafistes

A mitjan la dècada dels vuitanta, els ordinadors personals i el programari d'autoedició van entrar en els estudis de disseny i van transformar per complet la manera de treballar dels professionals de les arts gràfiques.

Els programes d'autoedició –o publicació d'escriptori, coneguda en anglès com a *desktop publishing* (DTP)– eren programes orientats a la maquetació de llibres i revistes o al disseny de cartells i fullets. El valor afegit procedia principalment del fet que el que es veia en pantalla s'aproximava força al resultat final imprès i permetia modificar un disseny amb facilitat.

La dimensió conceptual del canvi l'expressa molt bé Elena Fuenmayor de la manera següent:

"...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.

Durant la dècada dels noranta la publicació en format digital va cobrar força. El suport final ja no és solament el paper, el gràfic es veurà en pantalla (de l'ordinador, del mòbil, etc.). Especialment amb l'expansió d'Internet, l'ordinador, a més d'instrument de creació, passa a ser el dispositiu on es visualitza el publicat en la Xarxa.

Dissenyar alguna cosa perquè es vegi en pantalla i per un dispositiu informàtic és diferent, tècnicament i conceptualment, de fer-ho per a ser imprès en paper. En pocs anys, en la feina del grafista hi ha hagut dues grans transformacions que li han aportat flexibilitat i noves possibilitats.

### 1.1.3. Dispositius d'entrada i sortida de dades

A grans trets un ordinador sol dividir-se en tres parts:

1) **Unitat central de procés** (CPU en les sigles angleses), que inclou el processador i la resta de circuits i xips que el complementen.

2) **Memòria d'accés directe** (RAM) i **d'emmagatzemament** (disc dur i diversitat de discos extraïbles).

3) **Dispositius d'entrada i sortida de dades** (a més de les targetes controladores).

És obvi que aquests últims són els que permeten a una persona treballar (interactuar) amb l'ordinador. Uns són la via d'entrada d'informació i instruccions, i els altres la manera de verificar la feina feta i visualitzar resultats.

Respecte al treball amb gràfics digitals, els **dispositius d'entrada** són els que permeten digitalitzar gràfics analògics (escàner, càmeres digitals, targetes digitalitzadores A/V) i els que permeten generar imatges en el mateix ordinador (ratolí, teclat, tauleta gràfica). Els **dispositius de sortida** de dades principals són la impressora i el monitor.



## 1.2. Digitalització

### 1.2.1. Escàner

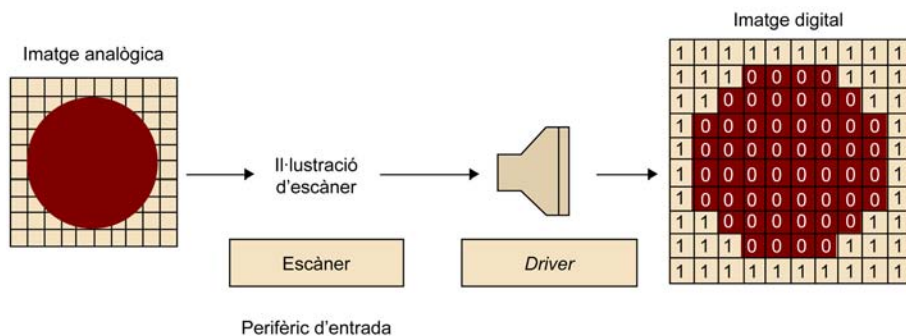
**Digitalitzar** implica transcriure estructures analògiques a noves estructures basades en zeros i uns.

L'**escàner** és un dels dispositius més utilitzats per a aquest procés i serà un exemple útil per a entendre com es pot fer aquesta transcripció.



Suposem que es vol digitalitzar un dibuix senzill en blanc i negre. L'escàner farà una escombratge amb llum de la imatge punt per punt. Una matriu interpretarà les diferències de llum en impulsos elèctrics que després passaran a codi binari. Les zones fosques s'emmagatzemaran com a zeros i les zones clares com a uns. Les dades de l'escàner passen a l'ordinador però aquest necessita tenir d'un programa "conductor" (*driver*) que envii les instruccions al dispositiu i rebí la informació. A més serà necessària una aplicació amb què l'usuari pugui controlar els paràmetres d'escan<sup>1</sup>.

<sup>(1)</sup>En aquest material s'usa el terme *escan* (de l'anglès *scanning*) per a referir-se al procés d'exploració d'una imatge per a la digitalització mitjançant un aparell d'escàner (de l'anglès *scanner*).



Si la imatge original té diferents valors de gris que es volen codificar, en comptes d'1 bit d'informació per a cada punt se'n faran servir vuit, agrupats en 1 byte (el qual ens donarà fins a 256 tons diferents). La matriu transformarà les diferències lluminoses en diferències de voltatge, a les quals després s'assignarà un valor numèric. En la digitalització d'imatges en color, l'escàner farà tres escombratges (un per a cada un dels colors primaris) amb filtres que s'encarreguen de deixar passar només l'espectre de llum que correspongui. En cada un s'usaran 8 bits i es codificaran les diferències d'intensitat. La suma dels tres colors compon la imatge en color en què cada punt té 24 bits (8 + 8 + 8) d'informació.

Així, doncs, cada punt es pot codificar amb un sol bit (per a la imatge en blanc i negre), amb vuit (per a la imatge amb diferents valors de gris, escala de grisos) o amb vint-i-quatre (per a la imatge en color, color real). Cada un d'aquests punts, la unitat bàsica de la imatge digital obtinguda, l'anomenem **píxel** (contracció de *picture element*).

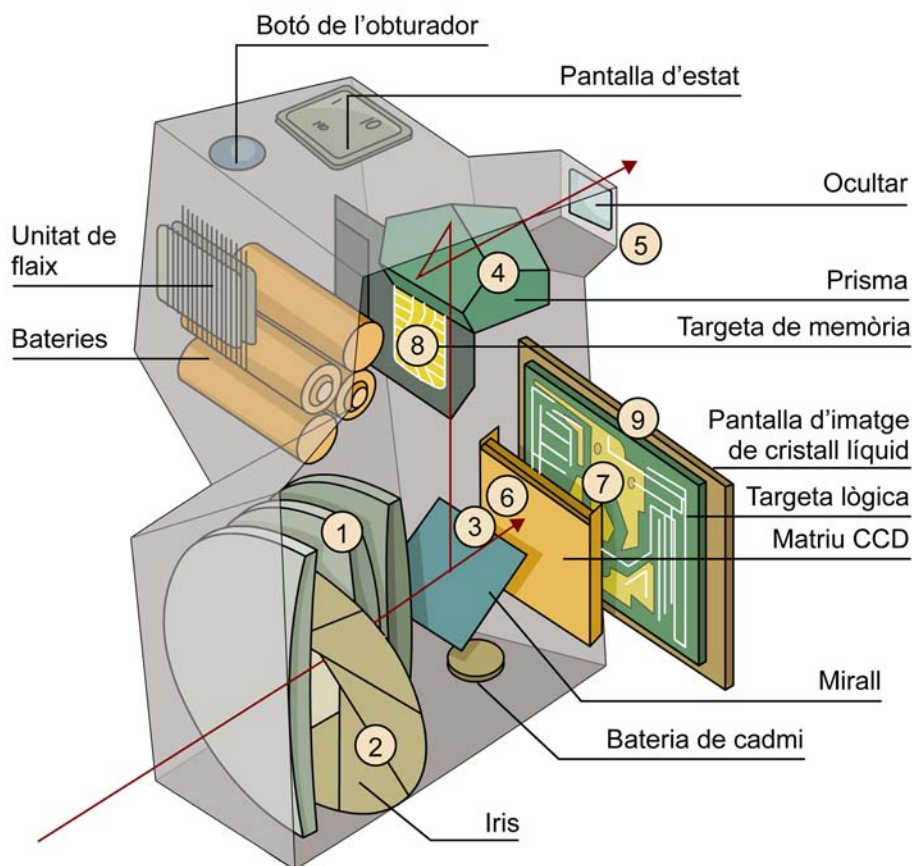
### 1.2.2. Càmeres digitals

Les **càmeres digitals** són un altre dispositiu de digitalització molt comú. Amb elles fotografem o gravem imatges del món real i després les copiem en el disc dur de l'ordinador. En la fotografia química tradicional la llum que travessa la lent queda impressionada en una pel·lícula (amb tres capes superposades de productes químics que reaccionen a diferents longituds d'ona). A les càmeres digitals, la llum que entra per la lent és captada per una retícula de punts de fòsfor que converteixen la llum rebuda en corrent elèctric. Aquests captors

fotosensibles donen més càrrega elèctrica com més intensa és la llum o més prolongada l'exposició. Aquestes càrregues es transfereixen a una unitat de memòria i la imatge es registra digitalment.

Per a obtenir una imatge en color s'utilitzen filtres, de manera que cada punt de fòsfor rebí una de les tres longituds d'ona. En funció del fabricant (Sony, Nikon, Casio, Minolta, Panasonic, Epson, HP, etc.) i de si són per a imatge en moviment (vídeo) o imatge estàtica (fotografia), s'utilitza una cinta de vídeo digital o alguna altra unitat d'emmagatzemament com discos durs o targetes de memòria que varien segons el fabricant. Per a l'entrada de les dades a l'ordinador s'utilitzen connectors com el Firewire (nom comercial d'IEEE-1394, també conegut com a i.Link) o USB. També s'utilitzen lectors específics per a cada tipus de targeta (lectors de targetes).

L'esquema següent explica com s'obté una imatge amb una càmera rèflex digital.



Prement parcialment el botó de l'obturador es dispara el focus automàtic i els mecanismes d'exposició, s'ajusta l'objectiu (1) i l'obertura de l'iris (2). En una càmera rèflex, com la que es mostra en la il·lustració, la llum que entra en l'objectiu és desviada per un mirall (3), a través d'un prisma (4), a una petita pantalla de visor acoblada a un ocular (5). Quan es prem completament el botó de l'obturador el mirall gira, deixa lliure el camí de la llum i s'activa la matriu CCD (6). La lectura de la matriu CCD és processada per una placa lògica (7), on es comprimeix. La imatge processada s'envia a la targeta de memòria (8) per al seu emmagatzemament i a la pantalla de la imatge de cristall líquid (9). Després d'uns segons la càmera ja està preparada per a prendre una altra imatge.

### 1.3. Dispositius d'entrada per a creació de gràfics

Si es crea la imatge directament en l'ordinador (**imatge sintètica**) s'han de fer servir programes específics de creació de gràfics. Però per a entrar dades en aquests programes també són necessaris dispositius d'entrada.

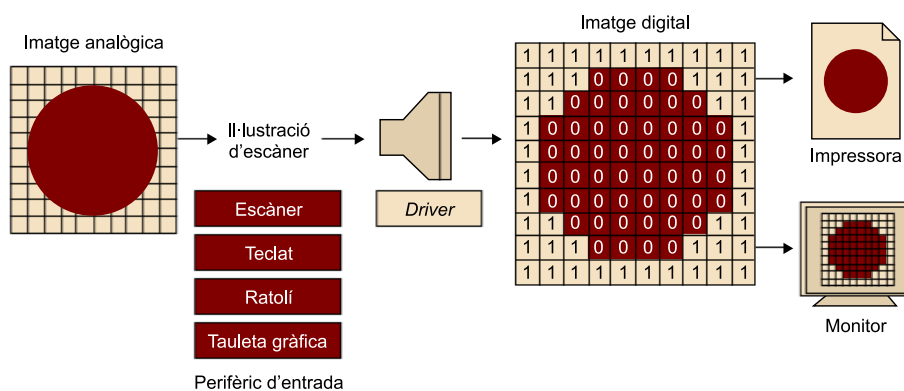


Teclat, ratolí i tauleta gràfica

Un dels primers dispositius que es van afegir a l'ordinador és el **teclat**. El seu propòsit inicial era introduir instruccions alfanumèriques (text i nombres) encara que el sistema operatiu i el programari en treuen el màxim partit, i són font d'entrada de tot tipus d'instruccions.

Des de la dècada dels vuitanta els ordinadors personals (i especialment els destinats al disseny gràfic) incorporen el *mouse* –el **ratolí**. Es tracta d'un dispositiu que tradueix els moviments d'una bola sobre una superfície en moviment d'un cursor en pantalla. És un element clau per a la interacció amb la interfície gràfica d'usuari (assenyalar, seleccionar, clicar, arrossegar i deixar anar) i del maneig de programes de gràfics. Posteriorment la bola s'ha substituït per tecnologia òptica.

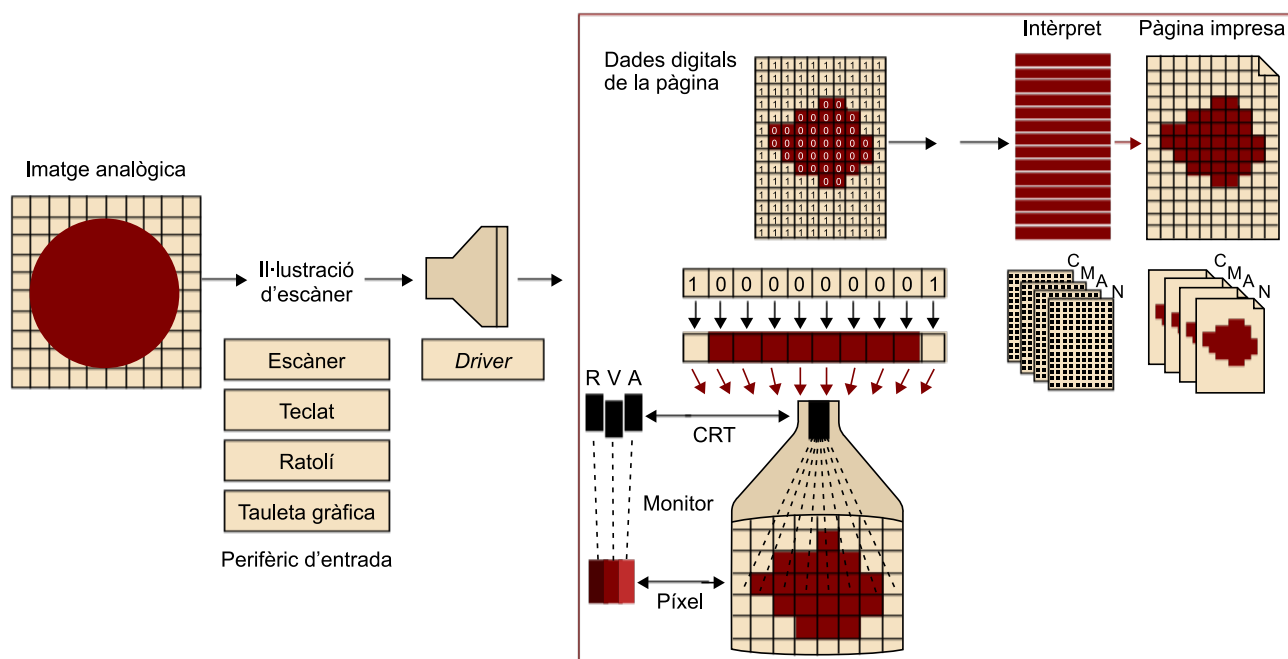
Per a treballs d'il·lustració les **tauletes gràfiques** milloren sensiblement la manera de dibuixar respecte al ratolí. Consten d'un "llapis" que es desplaça sobre una superfície que recull els punts de pressió exercits sobre ella. La majoria de tauletes són sensibles a les diferències de pressió i permeten que el programari de gràfics les tradueixi en diferències de gruix de línia.



## 1.4. Dispositius de sortida de gràfics

### 1.4.1. Impressora

Respecte als gràfics, els dispositius principals de sortida de dades són, com s'ha dit, la **impressora** i la **pantalla del monitor**. En aquest cas el procés és l'invers a l'entrada de dades. La informació digital s'ha de transcriure en informació analògica: tinta sobre paper mitjançant les impressores, i ones lluminoses en el cas del monitor.

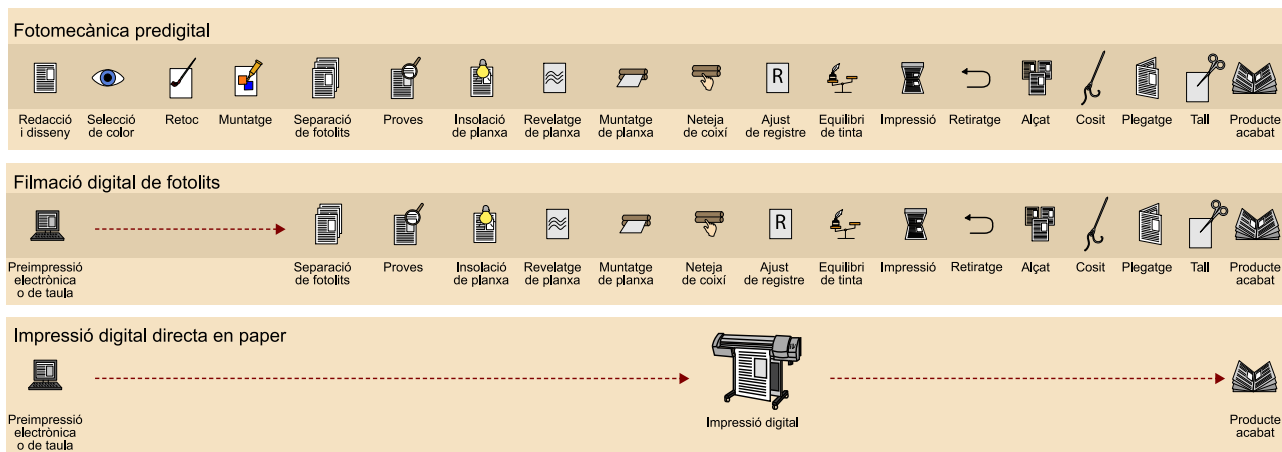


La **impressora** és un dels primers dispositius de sortida que es van incorporar als ordinadors. Els primers ordinadors que rebien les dades en targetes perforades en la dècada dels cinquanta del segle XX ja imprimien els seus resultats en paper. Des de la dècada dels vuitanta tenim "impressores d'escriptori" connectades directament a l'ordinador, per mitjà d'un dels ports d'entrada o d'una xarxa, que ens permeten obtenir ràpidament una imatge impresa del treball realitzat. Les tecnologies d'impressora (matricials, injecció de tinta, làser, etc.) evolucionen constantment i avui en dia la qualitat de la majoria és bona, variant en rapidesa i prestacions.

Les impressores a color usen quatre colors bàsics (cian, magenta, groc i negre) per a compondre la imatge per mitjà d'algun tipus de trama. També han evolucionat els plòters o impressores de gran format, convergint en part amb les tecnologies d'escriptori, millorant en qualitat i admetent tot tipus de suports d'impressió: paper, plàstics, tèxtil. Un altre dispositiu d'impressió és la filmadora de fotolits que tradueix la informació digital en impressió química sobre acetat. Aquest va ser un dels elements clau del canvi en la manera de treballar en el sector de les arts gràfiques, ja que el fotolit obtingut directament

de l'ordinador serveix per a insolar la planxa d'impressió (òfset) o la pantalla d'estampació (serigrafia), i evita el pas per la fotomecànica convencional. Actualment ja hi ha dispositius que insolen directament sobre la planxa (*direct to plate*) sense necessitat de fotolits. Destinades a substituir la impremta tradicional òfset en tiratges curts (200, 500, 1.000), hi ha impressores basades en les tecnologies d'escriptori capaces d'imprimir múltiples exemplars amb qualitat i rapidesa.

#### Evolució de la impressió



© UOC Creative Commons Reconocimiento Compartirlqual 3.0-es  
Imatge d'elaboració pròpia basada en un esquema de Roberto García Belchin de la *Guía de reproducción digital del color*

#### Evolució dels processos d'impressió

L'esquema mostra com s'han reduït o integrat els processos per a obtenir un producte acabat imprès. S'ha simplificat aquesta evolució en tres fases, des de la reprografia prèvia a la digitalització, passant per un procés basat en la filmació de fotolits des de l'ordinador i finalment un procés d'impressió directa de múltiples exemplars sobre paper.

#### 1.4.2. Pantalla del monitor

Un altre dispositiu de visualització és la **pantalla del monitor**. Va ser un dispositiu clau en l'aparició de l'ordinador personal, en la millora de la interacció persona/màquina i, evidentment, en la possibilitat de creació i tractament de gràfics per ordinador.

Perquè la informació digital es visualitzi en pantalla, s'ha de generar un senyal lluminós. Per això, independentment de la tecnologia, els monitors tenen una retícula de punts, equivalents al píxel, que poden emetre llum. El conjunt de píxels il·luminats compon la imatge. Les tecnologies més freqüents són el tub de rajos catòdics (CRT en les sigles angleses) i les pantalles de matriu activa (TFT, de *thin film transistor*).

Els **monitors CRT** tenen a l'interior un tub de rajos catòdics amb diversos canons que disparen electrons contra el material fosforescent de la pantalla. Aquest està format per partícules de fòsfor que en rebre l'impacte de l'electró emeten radiació visible. La diferent composició del fòsfor de la pantalla determinarà la longitud d'ona i en conseqüència el color percebut. Cada punt (píxel) de la pantalla té partícules de tres tipus que emetran llum corresponent al

blau, vermell o verd. Per a encendre-les s'utilitzen simultàniament tres feixos d'electrons. La intensitat de cada feix determina la intensitat (amplitud) de l'ona emesa. Com que les partícules només emeten llum durant un període curt, els feixos d'electrons escombren la pantalla diverses vegades per segon encenent-les. L'alta freqüència del refrescament de la pantalla permet que es percebi una imatge estable i contínua. Els desavantatges principals d'aquest tipus de monitor són que ocupa molt espai a la taula, a causa de l'espai necessari per al tub de rajos catòdics, que consumeix molta energia i que emet radiacions perjudicials per a l'usuari. Per això s'han buscat tecnologies alternatives que en bona part les han substituït.

En els **monitors TFT** no s'escombra una superfície fosforescent. Cada punt de la pantalla és una petita cel·la fixa de cristall líquid que deixa passar part de la radiació visible generada per làmpades de llum blanca en la part posterior. Cada cel·la correspon a un píxel i rep una càrrega variable d'un elèctrode alimentat per un transistor. Hi ha tres transistors per a cada cel·la (un per a cada color) i en funció de la càrrega rebuda les molècules del cristall líquid varien l'ondulació girant els rajos de llum. Un filtre polaritzat impedeix que passi la llum que no està alineada amb aquest.

## 2. Gràfics de mapa de bits

### 2.1. Una retícula de píxels

En termes generals hi ha dues maneres de codificar digitalment la informació gràfica, que donen lloc a dos tipus de gràfics: els anomenats **mapa de bits** i els **gràfics vectorials**.

En els gràfics de mapa de bits la informació s'emmagatzema punt per punt, i divideix la imatge en caselles quadrades formant una mena de retícula. Cada casella és un píxel, del qual es coneix la posició i s'hi assigna un valor corresponent al seu color. La forma de la imatge no es registra geomètricament, aquesta es percebrà quan es mostri cada píxel a la retícula. Es tracta del mateix sistema que hem explicat per a la digitalització del gràfic per mitjà dels escàners i les càmeres digitals. El nombre de bits usats per a la codificació del gràfic determina el nombre màxim de colors que pot tenir cada píxel. La mida de la retícula determina la qualitat (resolució) de la imatge.

#### Mapa de bits, també anomenats...

De vegades, la terminologia en el camp dels gràfics digitals és diversa per a un mateix concepte. Altres maneres de referir-se a les imatges de mapa de bits són:

- **Imatge tramada:** en anglès *raster*, referit a la reixeta rectangular formada per píxels.
- **Imatge matricial:** perquè està formada per una matriu de píxels.
- **Mapa de píxels o *pixmap*:** per contracció de l'expressió anglesa *pixel map*.

### 2.2. Resolució de la imatge

La resolució d'una imatge és el nombre d'unitats gràfiques per unitat de superfície.

En una imatge impresa és el nombre de punts per unitat de superfície. En una imatge digital és el nombre de píxels per unitat de superfície.

Com s'ha explicat, un **píxel** (abreviatura de *picture element*) és la unitat bàsica d'una imatge digital de mapa de bits. Distingeix l'element mínim sobre el qual podem definir el color. No té res a veure amb alguna mesura del món físic. Podem dir que el píxel és una unitat de divisió de la imatge sense una mida real concreta.

Amb **unitat de superfície** sí que ens referim a mesures del món físic. Podem usar els centímetres però, per influència de la cultura anglosaxona, solem usar les polzades. Una polzada és, doncs, una unitat de mesura física del sistema imperial britànic de mesures que correspon a 2,54 centímetres (o 25,4 mil·límetres) del sistema mètric decimal.

La resolució d'una imatge digital se sol expressar en píxels per polzada (abreujat com a ppp) –*pixels per inch* (ppi) en anglès. Quants píxels hi ha en aquest requadre d'una polzada? Depèn de la resolució d'entrada: escanejada, càmera digital, imatge creada o modificada en el programa de gràfics. Com més píxels hi hagi en una polzada més petits seran i millor serà la qualitat i la definició (detall) de la imatge resultant (més resolució).

### Resolució de la imatge analògica

És possible parlar de resolució d'una imatge analògica utilitzant com a unitat gràfica el punt. Així podem parlar de punts per centímetre o, més habitualment, punts per polzada –en anglès, *dots per inch* (dpi)– quan ens referim a la trama en una imatge impresa. Quants punts de trama caben en una polzada (en un quadrat de 2,54 × 2,54 centímetres)? Això depèn de la resolució del dispositiu de sortida (impressora, filmadora, etc.). En funció del nombre de punts per polzada tindrem més o menys definició de la imatge impresa resultant, com més nombre de punts més definició d'imatge.

En arts gràfiques se sol diferenciar entre la resolució de la imatge tractada en l'ordinador i la resolució de la imatge resultant impresa usant les expressions *resolució d'entrada* i *resolució de sortida*:

- **Resolució d'entrada:** el nombre de píxels per polzada d'una imatge analògica captada per un dispositiu de digitalització –escàner, càmera– o modificada en un programa de gràfics.
- **Resolució de sortida:** el nombre de punts per polzada del dispositiu que imprimeix la imatge (passem la imatge digital a un mitjà analògic).

L'abreviació **ppp** en català indueix a confusió perquè no sabem si la primera *p* es refereix a píxels o punts. Per això se solen usar les abreviacions angleses **ppi** (*pixels per inch*) i **dpi** (*dots per inch*) per a distingir entre la resolució d'entrada (imatge digital) i la resolució de sortida (imatge analògica).

Quan en aquests materials usem *ppp* sempre ens referim a píxels per polzada, si cal referir-se a la resolució d'una imatge analògica usem l'abreviació anglesa *dpi* o s'omet l'abreviació.

### Píxels per polzada

Els píxels per polzada es refereixen a la **polzada lineal**. Per exemple:

Si tenim un quadrat amb les dimensions d'1 × 1 polzada amb una resolució de 120 ppp tindrem: 120 × 120 píxels (120 × 120 = 14.400 píxels en tota l'àrea).

Dit d'una altra manera: si tenim una imatge de 120 ppp de resolució això significa que prenent un fragment d'1 × 1 polzada d'aquesta imatge hi podrem comptar 14.400 píxels.

Si la imatge és de 300 ppp de resolució en un fragment d'1 × 1 polzada tenim 300 × 300 píxels, que són 90.000 píxels en aquesta àrea.

La **qualitat** d'una imatge digital està directament relacionada amb la resolució, ja que com més gran sigui la quantitat de píxels per polzada que presenti, més alta serà la qualitat, però també serà més gran l'espai que requereix per a ser emmagatzemada, ja que presentarà més bits d'informació.



Més quantitat de ppi = més qualitat = més gran

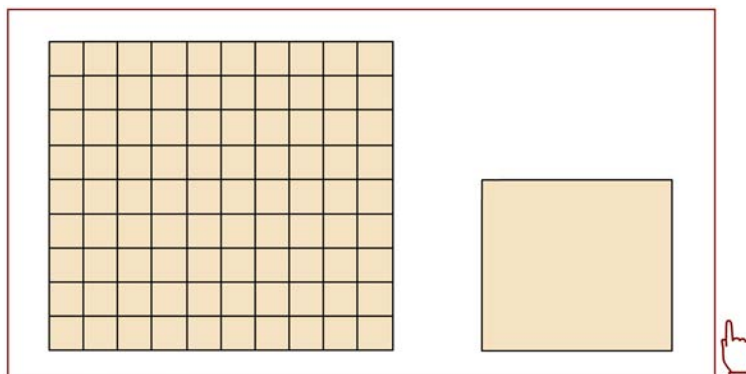
En l'àmbit de producció, les resolucions establertes més comunes per a treballar són les que s'exposen en la taula siguiente.

Resolucions més comunes en l'àmbit de producció

Resolució (ppi)	Producte	Perifèric de sortida
72	Pàgina web, CD/DVD	Monitor
200	Publicació impresa de qualitat mínima	Impressora convencional
600	Publicació impresa de qualitat òptima	Impremta
1.200-2.400	Filmació de fotolits	Òfset, serigrafia

### Activitat

A continuació disposeu d'una demostració interactiva que us pot ajudar a entendre el concepte de **resolució**. Els quadrats de les retícules emulen la funció d'un píxel. L'objectiu és experimentar de manera pràctica el treball a diferents resolucions i veure les limitacions de les imatges de baixa resolució.



A partir de les quadrícules següents, intenteu representar el model original. Per a això polseu sobre cada un dels quadrats perquè canviïn de color. Copieu píxel a píxel.

### 2.3. Resolució de pantalla

La **resolució de pantalla** determina la qualitat de les imatges que es representen en el monitor.

Les pantalles vectorials van ser les estàndard durant les dècades dels anys seixanta i setanta. Representaven la imatge per mitjà d'un oscil·loscopi que movia un feix d'electrons i dibuixava la imatge en pantalla. Al principi dels vuitanta es van rebutjar en favor de les pantalles d'escombratge basades en una retícula de píxels que, a partir de la dècada dels noranta, se substituirien progressivament per pantalles planes de plasma i cristall líquid basades també en una retícula de píxels. Per aquest motiu, tant les imatges de mapa de bits com les vectorials, han de representar-se finalment sobre una pantalla basada en mapes de bits.

La resolució d'una pantalla està determinada per la quantitat de píxels que pot representar en una unitat de superfície. Així, si prenguéssim un requadre d'1 × 1 polzada d'un monitor i comptéssim els píxels que hi ha, podríem determinar la resolució.

Durant molts anys la resolució dels monitors va ser limitada. Els ordinadors Macintosh d'Apple anaven acompanyats de pantalles que representaven 72 píxels per polzada. La resta d'ordinadors personals –clònics compatibles amb el PC d'IBM– usaven pantalles que representaven 96 píxels per polzada. Així la mida física d'una imatge digital en pantalla podia ser relativament estable, i l'anomenada *resolució de pantalla* a 72 ppp o 96 ppp es considerava establerta, i encara es pren com a referència, per inèrcia, en l'àmbit del disseny.

La millora tecnològica de la definició dels monitors ha fet que això deixi de ser així. Els monitors –especialment per l'empenta dels dispositius de pantalla petita com mòbils, *netbooks* o PDA– han deixat enrere les resolucions de 72 ppp i 96 ppp. Actualment la resolució de pantalla és un valor variable que depèn del monitor que utilitzem. Hi ha un límit en la mida física del píxel en pantalla que depèn de la mida dels punts de fòsfor (en pantalles de rajos catòdics) o de la cel·la (en pantalles de cristall líquid), però aquest límit depèn de la definició del monitor.

En conseqüència no hi ha una relació estable entre la resolució de la imatge digital i la mida física en la qual aquesta imatge es mostrarà en pantalla.

Una imatge, posem per exemple de 400 × 300 píxels, veurem que ocuparà la mateixa memòria i es mostrarà a la mateixa mida en un navegador web tant si s'ha definit una resolució de 72 ppp com si s'ha definit de 300 ppp. Per què?

Doncs perquè l'important en pantalla no és la relació (ppp) píxels/polzada definida en la mateixa imatge, sinó les mateixes dimensions que s'han donat a la imatge (píxels d'amplada × píxels d'altura) i la resolució que hem escollit en el nostre sistema operatiu, com també la mida física del monitor on visualitzem aquesta imatge.

### **Exemple de resolució de pantalla**

Una imatge de 800×600 píxels ocuparà tota la pantalla si hem escollit 800×600 com a resolució en les preferències del sistema operatiu. Tant en una pantalla de 14" com en una de 17" com en una de 19". La diferència és que la imatge es veurà físicament més gran en la pantalla de 19", ja que cada píxel serà més gran (ocuparà més espai en la superfície de la pantalla). Si canviem la resolució de 800×600 a 1.024×768 píxels o més, aquesta imatge de 800×600 ja no ocuparà tota la pantalla, només ocuparà una part.

Per tant, l'important en les imatges que es mostren en pantalla no és la resolució (ppp) sinó la mida (píxels d'altura per píxels d'amplada) que en determina la qualitat, ja que amb més píxels es pot representar millor un element gràfic, cosa que queda clara si pensem en una fotografia. Tot i així, el dissenyador no pot controlar la mida física en la qual es mostrarà aquesta mida en píxels. És l'usuari qui, per mitjà de la configuració de l'equip, té el control final de la seva experiència.

## 2.4. Profunditat de color

La **qualitat del color** d'una imatge de mapa de bits la determina la quantitat d'informació utilitzada per a representar cada píxel. Ens referim a la informació que es guarda per a cada píxel, com **profunditat de color del píxel** o també **profunditat del píxel**.

Les possibilitats de codificar informació de color depenen del nombre de bits que dediquem per a cada píxel. Així, a una imatge que disposi d'un nombre de bits més alt tindrà més possibilitats cromàtiques, però això també farà que ocupi més memòria d'emmagatzemament en l'ordinador: serà un arxiu amb més informació i més pes. Es diu que una imatge té una profunditat de color d'1 bit o de 8 bits en funció dels bits assignats a la codificació del color per a cada píxel. D'aquesta manera podem determinar l'escala següent:

Profunditat de color		
<b><math>2^1 = 2</math> tons</b>	<b>1 bit</b>	Imatges en blanc i negre.
<b><math>2^8 = 256</math> tons</b>	<b>8 bits</b>	Es poden crear imatges amb una escala tonal de qualitat. Utilitzat per a escala de grisos. O una imatge amb una representació del color limitada a una paleta de 256 tons com a màxim (color indexat).
<b><math>2^{24} = 16.777.216</math> tons</b>	<b>24 bits o color vertader</b>	Es disposa de 8 bits per canal de llum primària (vermell, verd i blau). El resultat és una gamma cromàtica aparentment il·limitada per a l'ull humà. Les imatges de qualitat han de tenir, com a mínim, aquesta quantitat de tons. També es coneix com a <i>color real (true color)</i> .
<b><math>2^{32} = 4.294.967.296</math> tons</b>	<b>32 bits o color vertader</b>	Difereix de l'anterior perquè disposa d'un canal de 8 bits per a funcions gràfiques avançades, com ara l'ús de manipulacions especials. També utilitzat en el processament d'imatges en quadricromia que tenen quatre canals (cian, magenta, groc i negre) i també necessiten 8 bits per canal.

En funció del context de treball haurem d'arribar a un compromís entre la qualitat que volem per a la imatge i la quantitat de memòria que necessitarà l'arxiu. Això és especialment important en la creació de gràfics per a Internet on, a causa de l'amplada de banda, reduir la memòria que ocupen els gràfics pot ser important.

### Vegeu també

Àmplia informació sobre la profunditat de color en relació amb els sistemes de codificació i als modes de color en el subapartat "Codificació digital del color" de l'apartat "Color i disseny gràfic" del mòdul 4, "Bases de disseny gràfic".



N'hi ha prou amb 8 bits per a representar aquesta icona que identifica el repositori d'imatges i arxius lliures del projecte "Commons" de la Fundació Wikimedia en què s'utilitzen colors plans.

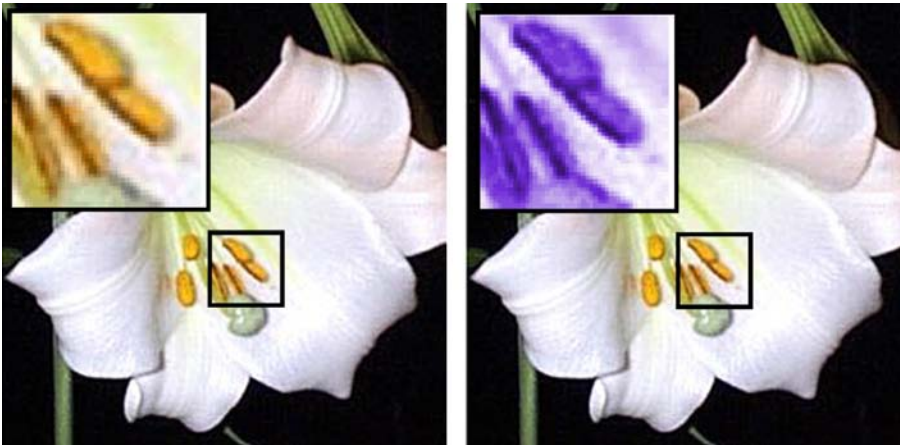


N'hi ha prou amb 8 bits per a aquesta fotografia.



N'hi ha prou amb 24 bits per a representar aquesta fotografia amb bona qualitat.

## 2.5. Eines d'edició de gràfics de mapa de bits



En el primer gràfic ampliem un fragment per veure com la imatge està constituïda per píxels. En el segon gràfic mostrem el tractament aplicat només sobre aquest fragment. S'han modificat els valors de color dels píxels que el constitueixen. En aquest exemple, s'ha fet un tint en blau que només afecta els píxels seleccionats. El programa de gràfics no pot reconèixer una flor, per a ell són grans quantitats de píxels.

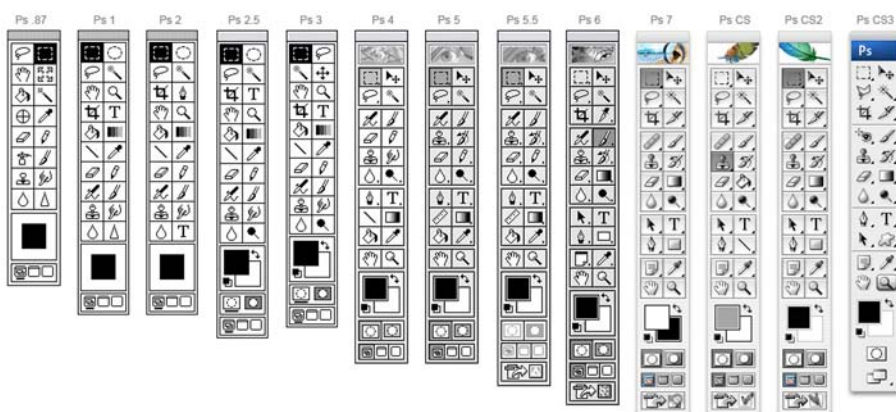
La majoria de programes de gràfics usen la convenció d'una caixa d'eines, disponible com a finestra del sistema, en la qual l'usuari pot escollir diferents maneres de crear o manipular els gràfics.

En un gràfic de mapa de bits qualsevol actuació implica la modificació dels colors dels píxels, és a dir, es modifiquen els píxels la suma dels quals constitueix la imatge. La manera més simple d'editar seria, per exemple, utilitzant el cursor com un llapis que va canviant a negre els píxels per on passa, dibuixar a mà alçada. Però evidentment les eines actuals s'han sofisticat molt més i permeten aconseguir efectes molt diferents i de més concreció. A més, les eines són parametrizables. Això permet que, en la majoria de programes (és així

tant en el GIMP com en el Photoshop i Fireworks, per esmentar-ne alguns), un doble clic a la icona de l'eina obri una finestra en la qual és possible ajustar els paràmetres per a aconseguir l'efecte desitjat.



L'aspecte visual dels elements de l'entorn de treball d'un programa com el GIMP es poden adaptar integrant-se en el sistema operatiu –Windows, Linux, MacOS– i en el "tema" visual escollit. El joc d'eines es manté estable i és molt similar al del Photoshop.



#### Eines del Photoshop

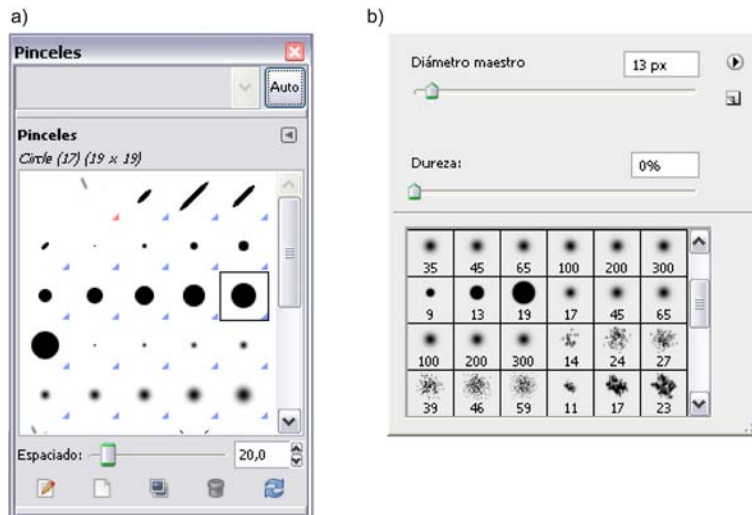
Evolució de la caixa d'eines del Photoshop des de les primeres versions. Com es pot veure, des de la primera versió dels vuitanta tampoc no ha variat tant. El conjunt principal d'eines continua essent el mateix en aquest i en altres editors de gràfics, però són més parametritzables.

© Adobe. Aquesta imatge es reproduïx acollint-se al dret de cita o ressenya (art. 32 LPI), i està exclosa de la llicència per defecte d'aquests materials.

### 2.5.1. Eines de dibuix

Els programes de gràfics han evolucionat per intentar obtenir resultats similars als que es poden aconseguir amb qualsevol eina de dibuix. S'han desenvolupat algoritmes que simulen el comportament, per exemple, de llapis, carbó, pinzells, etc.

Les eines més habituals són el **llapis** i el **pinzell o brotxa**. Són dues eines molt similars que se solen diferenciar perquè el **llapis** genera un traç d'un sol color, la qual cosa en baixa resolució produeix un dentat en la línia, i el **pinzell** genera traç suavitzat en les vores mitjançant un escalat de color entre la línia i el fons. De tota manera, aquest sol ser un paràmetre ajustable que també pot incloure l'eina del pinzell.



a) Selecció de pinzell en el GIMP 2.6.8. b) Selecció de brotxa/pinzell en el Photoshop CS4.

Per al llapis i el pinzell es poden escollir diferents gruixos i formes que determinaran el traç. També és possible crear el nostre propi pinzell; en aquest cas s'haurà de determinar en píxels el seu **diàmetre**, la **duresa** (que determina el nombre de píxels del suavitzat en les vores), la **forma** (pot ser circular, el·líptica, quadrada o de qualsevol altra forma) i l'**angle d'inclinació**. Altres paràmetres ajustables són l'**opacitat**. Per sota del 100% el traç augmenta en transparència i el color resultant és una barreja del color escollit i el del fons. I, finalment, allò que anomenen **mode** determina de quina manera s'aplica el traç en relació amb el que ja hi ha en la imatge. Per exemple, tenim el mode Enfosquir, que fa que el pinzell només pinti sobre els colors més clars que l'escollit.

El pinzell també pot tenir un altre paràmetre anomenat **esvaïment**, que fa que a mig traç ens quedem sense "pintura", com si es tractés d'un pinzell que hem de tornar a sucuar al pot. Per a simular aquest efecte, la pèrdua de color és progressiva.

Les eines de llapis i pinzell solen estar preparades per a rebre paràmetres de canvi de pressió. Si disposem d'una paleta amb llapis òptic sensible a la pressió, el traç pot canviar de gruix en funció d'aquesta i es pot obtenir una línia modulada.

En la taula següent es mostren diferents efectes en funció dels paràmetres d'eina utilitzats:

Pinzell	Paràmetres	Exemple d'ús
	Radi: 1 px. Duresa: 100% Opacitat: 100% Fluxe: 100%	

Altres eines de dibuix importants en l'edició gràfica són:

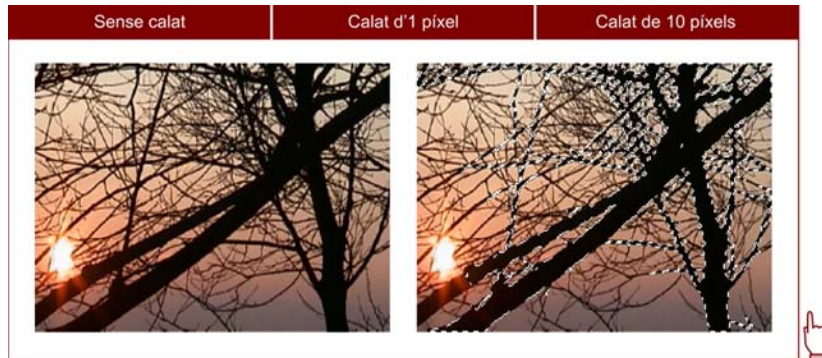
- La **goma**. Que esborra el dibuix, i amb el gruix i opacitat parametrizables. La goma pot esborrar el color escollit com a color de fons o canviar-lo en transparent si hem escollit aquesta opció com a fons del gràfic.
- L'**aerògraf**. Que simula aquesta eina amb un traç difuminat que se superposa al color existent.
- El **pot de pintura**. Que pinta del mateix color els píxels contigus al punt en què s'aplica.

### 2.5.2. Eines de selecció

Les eines de selecció també són un element clau en el programari de mapa de bits. Permeten seleccionar una zona de píxels per a aplicar-los un efecte, desplaçar-los o copiar-los per a després enganxar-los en un altre arxiu. Hi ha tres eines bàsiques de selecció:

- La selecció regular en forma de rectangle o el·lipse.
- El llaç o selecció irregular.
- La vareta màgica que selecciona els píxels amb un valor de color similar.

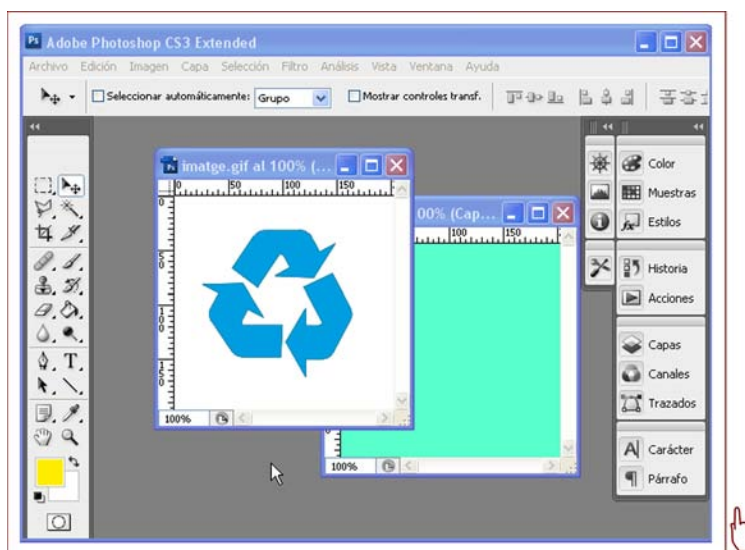
Les eines de selecció també són parametrizables. En general, els tres tipus d'eines permeten determinar un calat o suavitzat de la selecció de manera que, si apliquem color, aquest s'integrarà amb el fons en els límits de la selecció.



Selecció sobre la imatge original. Es pot aplicar un calat en la selecció i després copiar el contingut per a enganxar-lo sobre un altre fons.

L'eina de **selecció regular** permet seleccionar lliurement o de manera restringida. Es pot restringir l'ús de l'eina a una mida fixa o bé usar una proporció restringida. Les dues opcions són molt útils. Si, per exemple, hem de retallar diverses fotos a una mateixa mida farem servir la mida fixa per a enquadrar-les i després retallar la imatge. Si la resolució de partida és diferent podem utilitzar la proporció restringida per a seleccionar el que vulguem i després adaptar la mida de la imatge.

La **vareta màgica** s'acostuma a utilitzar per a seleccionar àrees del dibuix que tenen un mateix color. Per defecte se seleccionen els píxels iguals que són contigus. Però, com a paràmetre o opció del menú, també hi ha la possibilitat d'escollir tots els píxels de la imatge que tinguin el mateix color que el píxel sobre el qual s'ha clicat. Un altre paràmetre editable és l'anomenada **tolerància** (en el Photoshop i en el Fireworks) o **llindar** (a The GIMP). Permet seleccionar, no solament els píxels del mateix color, sinó els que són d'un color similar. El llindar de tolerància és ajustable, de manera que es pot definir l'escalat de colors similars a seleccionar.



**Passos:** selecció de la vareta màgica. Valor de tolerància: 10. Deselecció de contigu. Copiar i enganxar en un nou document. Programa d'execució: Photoshop CS3.



L'eina de **selecció irregular (llaç)** és més adequada si volem seleccionar una imatge que conté diferents colors. Per exemple, si volem seleccionar una persona d'una foto per a separar-la del fons.

Els programes de gràfics ens solen oferir diferents opcions de llaç. El llaç per defecte ens permet seleccionar a mà alçada. Normalment també disposem d'un **llaç poligonal** per a seleccionar formes rectilínies. Cliquem punt a punt i va traçant línies rectes entre aquests, i dóna com a resultat una selecció poligonal. Una altra variant del llaç és el **llaç magnètic**, de vegades diferenciat com una altra eina en forma de tisores com en el GIMP. El programa reconeix els canvis de color en la imatge i magnetitza el traçat de la selecció amb aquests; això ens ajuda a seguir els contorns.



**Passos:** selecció de l'eina llaç magnètic. Copiar i enganjar en un document nou. Reescalar la selecció copiada. Programa d'execució: Photoshop CS3.

Una eina semblant al llaç és la que ens permet crear **traçats utilitzant corbes de Bézier**. Aquests traçats després es poden convertir en selecció de la imatge. L'avantatge per a una persona acostumada a treballar amb objectes Bézier és que pot canviar de posició els punts (nodes) o utilitzar agafadors per a definir les corbes. La utilització de l'eina de selecció mitjançant la ploma de Bézier és la que ens dóna un nivell d'agudesa i control més alt en el resultat de les seleccions. Encara que al principi acostumar-s'hi pot ser una mica difícil, és molt recomanable practicar per a utilitzar-lo amb facilitat.

Associades a les eines de selecció disposem de diverses **opcions de menú** en la barra superior, també accessibles des d'un menú contextual clicant al botó dret del ratolí. Entre les més habituals hi ha:

- **Seleccionar tot.** Selecciona tota la imatge.
- **Desseleccionar.** Elimina la selecció respectant la imatge.
- **Calar/difuminar.** Difumina els límits de la selecció de la mateixa manera que ho faria el paràmetre d'eina equivalent. L'avantatge és que la podem

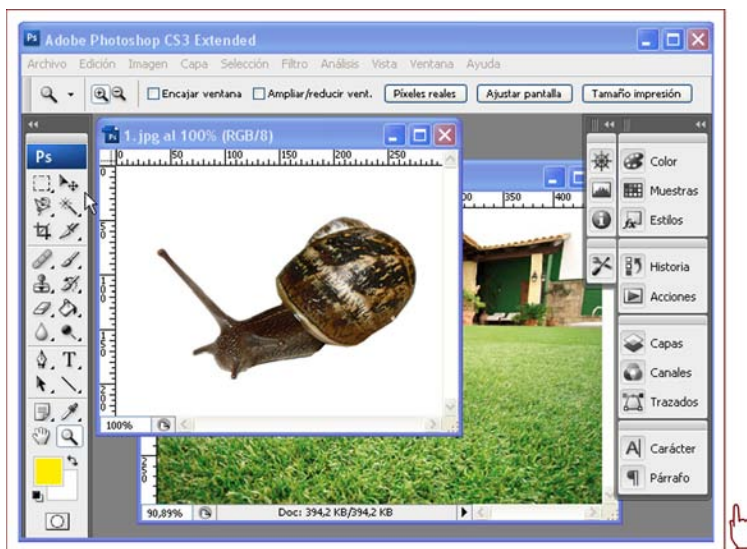
#### Vegeu també

En l'apartat "Gràfics vectorials" podeu ampliar informació sobre els objectes Bézier.

aplicar a una selecció ja creada. Cal determinar el nombre de píxels de calat. En alguns programes, com el GIMP, hi ha a més l'opció contrària, enfocar.

- **Estendre/créixer.** Permet ampliar la selecció un nombre determinat de píxels a partir del contorn actual. En alguns programes, com el GIMP i Photoshop, hi ha a més l'opció contrària, encongir.
- **Seleccionar similar.** Selecciona els píxels de la resta de la imatge que són del mateix color que els seleccionats.
- **Invertir selecció.** Selecciona els píxels no seleccionats i desselecciona els seleccionats. Una mena de "negatiu" de la selecció actual.

Aquesta última opció és extremadament útil. Per exemple, si tenim una figura complexa pel que fa als colors sobre un fons de color homogeni, la millor opció és seleccionar el fons usant la vareta màgica (amb cert grau de tolerància que inclogui les variacions del fons) i després invertir la selecció. D'aquesta manera obtenim una selecció precisa de la figura.



Passos: selecció de la vareta màgica. Valor de tolerància 10, selecció del blanc. Invertir selecció. Copiar i enganjar en un document nou. Programa d'execució: Photoshop CS3.

Una altra opció molt útil és la possibilitat de **desar la selecció**, amb la qual les dades d'aquesta s'emmagatzemen en un canal addicional de la imatge (**canal alfa**) que permet recuperar-la. Quan es treballa amb una imatge amb diverses seleccions a les quals necessitem tornar contínuament, aquesta possibilitat és fonamental per a aplicar-hi efectes, per copiar-les i enganjar-les en un altre lloc, etc.

### 2.5.3. Filtres algorítmics

Els **filtres** són potser el més específic del mitjà digital referent a gràfics de mapa de bits. Es tracta d'algoritmes que s'apliquen a una imatge (o a una part) i varien el color dels píxels de manera que s'aconsegueixi un efecte determinat.

Els filtres són petits programes que, en la majoria casos, ens permeten ajustar els paràmetres i previsualitzar els resultats de la seva aplicació. Els programes de gràfics tracten els filtres com extensions modulares del programa (*plug-in's*) i, per tant, se'n poden afegir de nous. El programari comercial permet que "terceres parts", empreses o persones diferents de la creadora del programari, desenvolupin filtres que l'usuari pot incorporar després de la compra o descàrrega. En haver-se mantingut Photoshop com a líder del programari de gràfics de mapa de bits durant molt temps, la majoria de programes admet la importació dels seus filtres (és a dir, dels filtres que compleixen les especificacions d'Adobe per a *plug-in's* del Photoshop).



Fotògraf desconegut. Aquesta imatge es reproduceix acollint-se al dret de citació o ressenya (art. 32 LPI), i està exclosa de la llicència per defecte d'aquests materials.

### 2.5.4. Ajustaments d'imatge

Si les eines de dibuix usen metàfores i símil del dibuix i la pintura convencionals sobre paper i tela, les opcions d'**ajustament d'imatge** tenen com a referent la fotografia. Són opcions que permeten ajustar paràmetres de la imatge com el color o el contrast i que s'assemblen al treball de laboratori que es fa amb la fotografia analògica, amb un procés químic de revelatge.

Els ajustaments més importants són:

- **Nivells.** Permeten corregir i/o equilibrar el nombre de píxels en llums (o ressaltats), tons mitjans i ombres utilitzant un histograma que mostra la quantitat de píxels de la imatge en cada nivell de brillantor (en una escala de 0 a 256 nivells possibles).
- **Corbes.** Permeten corregir el rang tonal, igual que els nivells, però també es pot modificar color per color (dels tres primaris: vermell, verd i blau).
- **Brillantor/contrast.** Permet modificar la brillantor de tots els píxels d'una imatge, llums, tons mitjans i ombres, tots alhora, i també corregir el contrast entre ells, accentuant-lo o mitigant-lo.
- **Balanç o equilibri de color.** Permet modificar el balanç del color, ajustant les dominàncies de color que hi pot haver en una imatge.
- **To/saturació o matís/saturació.** Permet modificar els tons de color d'una imatge, augmentar-ne o baixar-ne la saturació i també augmentar-ne o baixar-ne la lluminositat. Afecta tota la imatge. Utilitzant aquest ajustament amb l'opció d'acolorir activada es poden realitzar virats d'un mateix to en tota la imatge.

En tot cas, la majoria es poden entendre com una actuació sobre el color de la imatge.

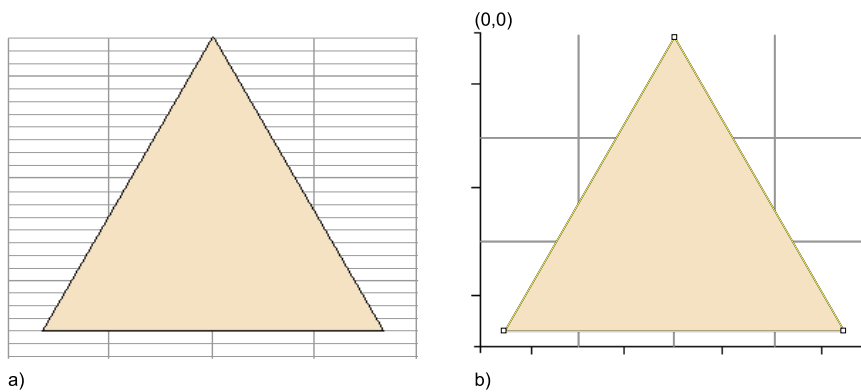
#### Vegeu també

En l'apartat "Color i disseny gràfic" del mòdul 4, "Bases del disseny gràfic", es tracten més a fons algunes qüestions relacionades amb aquests ajustaments.

### 3. Gràfics vectorials

#### 3.1. Dibuix basat en objectes

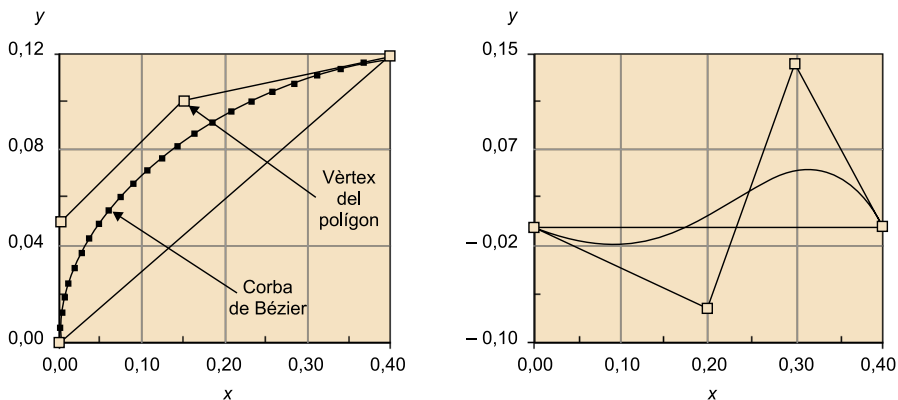
En els gràfics vectorials la informació de la imatge no s'emmagatzema sobre la base d'una matriu de píxels, com en els gràfics de mapa de bits. El que codifica el gràfic és la **informació numèrica de la seva geometria**. Per tant, tenim objectes gràfics independents entre ells que es defineixen per punts de coordenades i vectors que uneixen aquests punts.



a) Exemple d'un triangle en un gràfic de mapa de bits (a una resolució baixa). Tenim informació del color de cada punt de la imatge (representats aquí per les caselles). No tenim informació del triangle per si mateix, aquest es mostra a partir de la informació dels punts. El gràfic està desat en un fitxer PNG.  
 b) Exemple d'un triangle en un gràfic vectorial. El triangle es defineix a partir de les coordenades dels vèrtexs i de la unió d'aquests a partir de línies rectes. Tenim, doncs, la informació geomètrica de l'objecte. El gràfic està desat en un fitxer SWF.

Hi ha diverses maneres de codificar una imatge a partir de vectors. La més estesa entre els programes i els formats de fitxer de gràfics vectorials és la que es basa en les anomenades **corbes de Bézier** o, per extensió, **objectes Bézier**.

##### 3.1.1. Objectes Bézier



Els **objectes Bézier** són segments de línia connectats entre ells per **nodes**. Cada segment l'entendem com un vector amb un punt inicial i un punt final que defineixen la línia. A aquests s'afegeixen **dos punts de control** que en defineixen

xen la curvatura. Els punts de control parteixen de les tangents de cada punt extrem o node. Quan els quatre punts estan alineats tenim una recta. Quan els punts de control se separen tenim algun tipus de línia corba. Aquesta corba sempre és dins d'un polígon quadrilàter els vèrtexs del qual són aquests quatre punts. La corba es calcula a partir d'una interpolació creada per una seqüència de funcions que es basa en les coordenades dels punts. Això fa que sigui escalable i es vegi bé en qualsevol nivell d'ampliació.

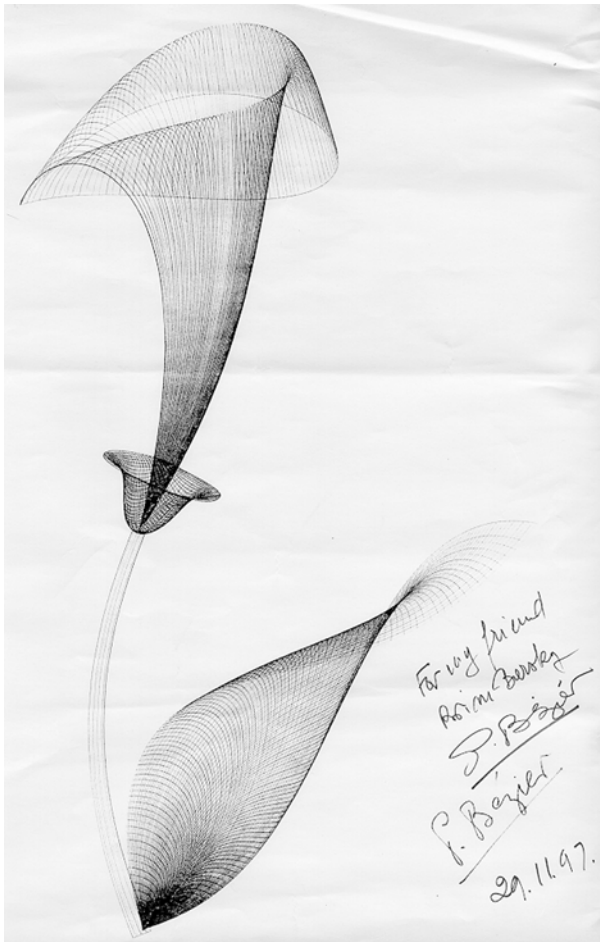
### Pierre Etienne Bézier

Les anomenades **corbes de Bézier** deuen el nom i l'origen a Pierre Etienne Bézier (1910-1999), que va ser enginyer de Renault durant quaranta-dos anys (de 1931 a 1975). Com a responsable del departament *des méthodes mécaniques*, al començament de la dècada dels seixanta, es comença a preocupar per la utilització d'eines informàtiques en el disseny de carrosseries. Les eines de CAD (*computer aided manufacturing*) basades en polígons no representaven bé les superfícies corbes. Bézier va aconseguir un mètode fàcil d'usar i alhora exacte per a descriure corbes a partir de quatre punts. El sistema es va llançar el 1968, i el 1975 ja estava en ple ús. Les **corbes de Bézier** han estat un element clau per al desenvolupament posterior de la informàtica gràfica vectorial, tant en 3D com en 2D.

Abans que Bézier, dos matemàtics que treballaven per a Citroën (James Ferguson i Paul de Casteljaou) van arribar als mateixos resultats. Però les seves troballes no es van conèixer fins a molt més tard perquè es van guardar sota secret industrial.



Fotografia de Pierre Etienne Bézier



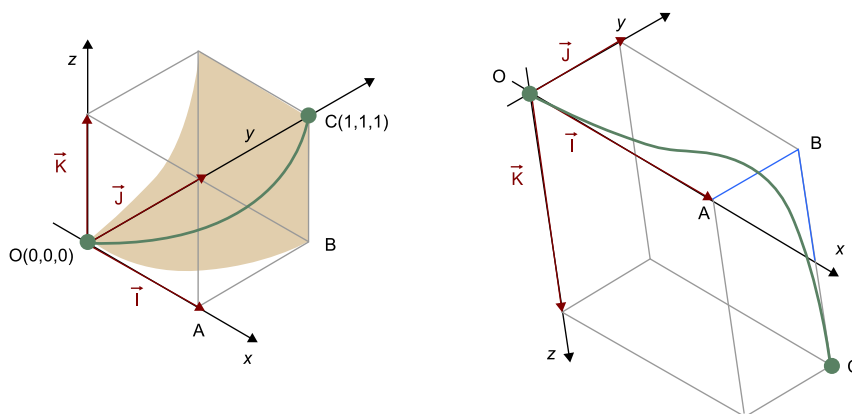
© Descendants Pierre Bézier. Aquesta imatge es reproduïx acollint-se al dret de citació o ressenya (art. 32 LPI), i està exclòsa de la llicència per defecte d'aquests materials.  
Obra de Pierre Bézier feta utilitzant corbes de Bézier.

### 3.1.2. Bézier en el programari gràfic

El programari de gràfics vectorials troba en les corbes de Bézier una metodologia fàcil d'usar i de computar. La manera de treballar s'assembla però no és exactament la mateixa que s'ha vist en la demostració anterior. La pràctica del dibuix vectorial ha permès depurar una interfície més eficaç i "amable".

#### Corbes de Bézier en gràfics 3D

Aquí estem veient l'ús de corbes de Bézier en programari de gràfics de dues dimensions. Però Pierre Bézier volia les corbes per a dissenyar carrosseries de cotxe. Per tant, volia corbar superfícies, no solament línies. El procediment és el mateix. Per a entendre-ho podem començar descrivint una superfície corba dins d'un cub i després deformar aquest cub (canviant les posicions dels vèrtexs) i modificar-ne així la corba inscrita.

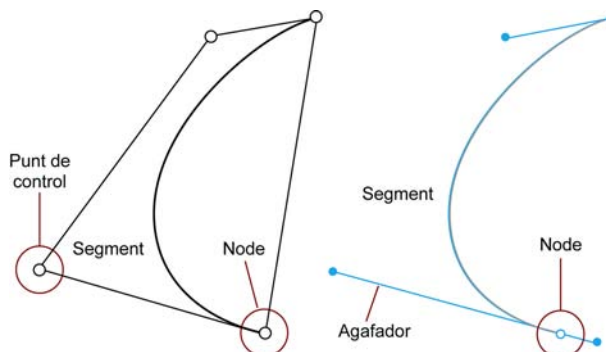


Un conjunt de corbes d'aquest tipus connectades entre elles ens permetrien definir qualsevol superfície.

El programari de gràfics 3D (que escapa dels objectius d'aquest material) també fa servir les corbes de Bézier per a definir la forma de les superfícies.

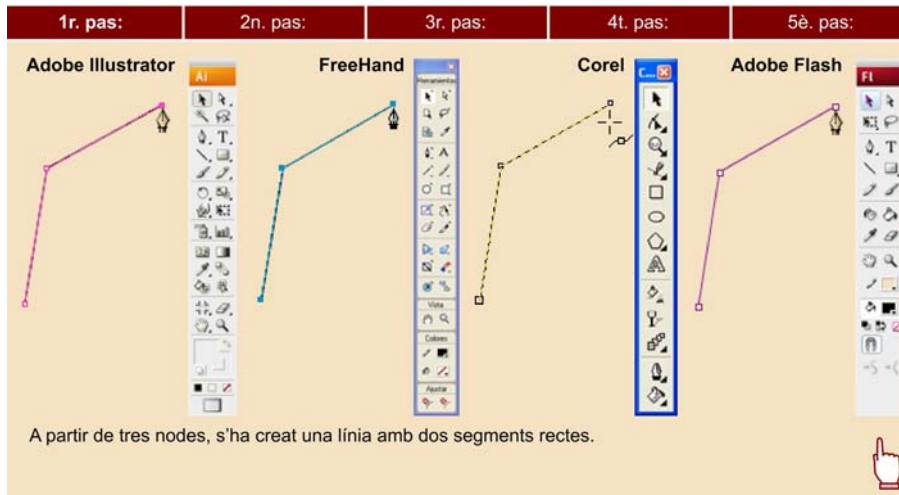
#### Ús de les corbes de Bézier en la geometria de les fonts tipogràfiques

En tipografia digital és necessari un sistema de codificació de la informació geomètrica que permeti que els tipus s'imprimeixin perfectament a qualsevol mida. Per a això també s'usa el sistema de Bézier, que es va adoptar com a sistema estàndard de codificació de corbes per al llenguatge **PostScript**. És, doncs, el sistema usat en les fonts True Type (amb corbes de segon ordre) i PostScript Tipus 1 (amb corbes de tercer ordre).



Per a crear una línia es van creant els **nodes**, connectats entre ells per línies rectes. Si es vol corbar un d'aquests segments, de "dins" d'un dels nodes s'extreu un **agafador**, que en realitat és un punt de control o vèrtex del polígon. Regulant l'agafador corresponent a cada node es defineix la **forma de la corba**.

En els programes de gràfics és habitual anomenar **ploma** l'eina que permet crear corbes de Bézier d'aquesta manera. També se sol representar per la icona d'una ploma estilogràfica.

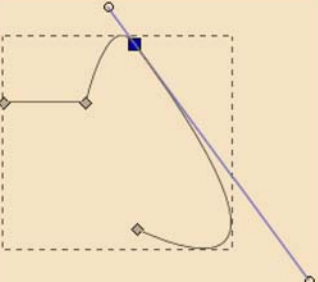
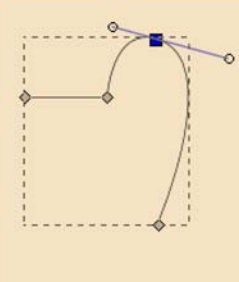


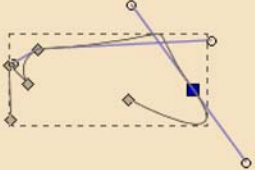



Aquí es pot veure la seqüència de creació d'una corba amb programari vectorial. S'han fet captures de pantalla en quatre programes diferents entre els més utilitzats (Adobe Illustrator, Freehand, Corel y Adobe Falsh). Com es pot veure, la metodologia és molt similar i les diferències en la interfície són mínimes. Cliqueu en el botó per avançar o retrocedir en el procés de creació de la corba.

### 3.1.3. Treball de precisió amb les corbes de Bézier

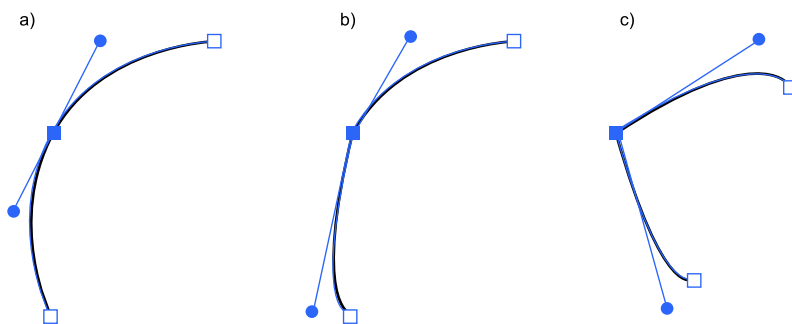
Dibuixant objectes Bézier és fàcil crear elements amb massa nodes o amb corbes que no conserven continuïtat en tot el traçat. Hi ha una sèrie de normes que ens poden ajudar a treballar amb més precisió. En la taula següent presentem les més importants.



Normes	Incorrecte	Correcte
<p><b>La línia 1/3:</b> un agafador (distància entre node i punt de control) no ha de fer més d'1/3 del segment al qual modifica.</p>		
<p><b>Línia empremta:</b> s'ha d'utilitzar el nombre de nodes més petit possible; per tant, en un arc no s'haurien d'utilitzar més de dos nodes.</p>		
<p><b>Línia tangent:</b> l'agafador ha de ser tangent al segment, però mai creuar-lo.</p>		

Encara que en formular-les com a "normes" usem un to categòric, no s'ha de perdre de vista que el disseny gràfic té un caràcter experimental i qualsevol norma es pot qüestionar i subvertir. Aquestes normes són el resultat de l'experiència pràctica i aquí rau el seu valor.

Un altre aspecte important a tenir en compte és la diferència entre **punts de vèrtex** i **punts de corba**. Quan connectem dos segments corbs, si volem que la corba tingui una bona continuïtat, les dues tangents del node que comparteixen els segments han d'estar alineades. En canvi, si el node fa la funció de vèrtex, les dues tangents no estaran alineades.



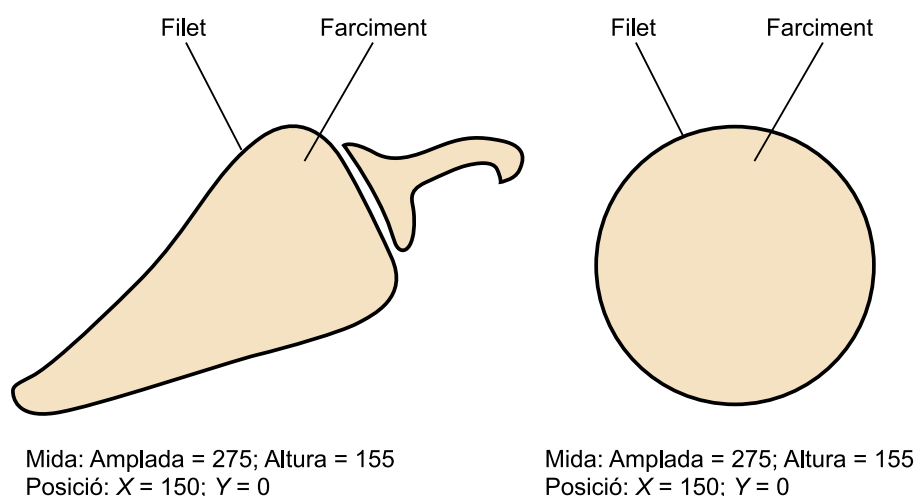
- a) Aquí tenim dos segments corbs ben connectats entre ells. Fixeu-vos que les tangents del node central estan alineades. El resultat és una corba que flueix sense "ruptures". b) En aquesta línia les tangents del node central no estan alineades i la corba no té bona continuïtat. c) Aquí el node central és un vèrtex, i deliberadament els dos segments no formen una línia contínua. Les tangents no estan alineades i formen un angle.

Aquí el programari de gràfics ens facilita la tasca permetent que els dos agafadors es moguin conjuntament o siguin independents. La major part de programes de dibuix vectorial permet que un punt sigui de **tipus vèrtex** o de **tipus corba** i passar d'un tipus a un altre canviant les propietats del punt.

### 3.2. Propietats dels objectes vectorials

Les propietats principals que componen un objecte vectorial són les següents:

- **Color interior o farciment.** Determina el farciment de l'objecte. Aquest pot ser un color pla, una textura o un degradat. Perquè un objecte es pugui omplir ha d'estar tancat (l'últim node s'uneix amb el primer).
- **Filet o línia.** Representa la línia externa que envolta l'objecte. Es pot variar el color, el tipus (línies discontinües, punts, etc.) i el gruix.
- **Posició.** Determina quina és la situació de l'objecte en el document de treball. Els objectes vectorials es distribueixen en un sistema de coordenades de dues dimensions ( $x, y$ ), la qual cosa equival a un pla.
- **Mida.** Determina quina àrea ocupa l'objecte en el document de treball.





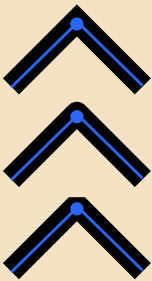


Propietats de l'objecte vectorial






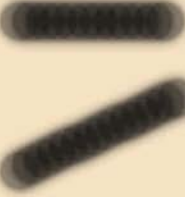
#### 3.2.1. Propietats de la línia o filet

Les propietats del modificable del filet poden variar d'un programa a l'altre. Bàsicament podem diferenciar les propietats que ofereixen els programes de "primera generació" destinats a impressió per a arts gràfiques (Freehand, Illustrator, Corel) i les que ofereixen els de "segona generació" pensats per a la publicació en web i, per tant, per a la visualització en pantalla (Fireworks, Flash).

Els programes originàriament destinats a crear dissenys, dels quals es trauran fotolits i que després s'imprimiran en una impremta òfset, donen molta importància a la forma de la línia. En ells es poden controlar propietats com les que es mostren en la taula següent:

Propietats de la línia	
<p>6pt </p> <p>12pt </p> <p>24pt </p>	<p><b>Grossària:</b> es reparteix (a parts iguals) entre la part inferior i superior de la línia imaginària marcada pels dos nodes. L'amplada se sol mesurar per punts de pica (unitat de mesura usada pels tipògrafs). En el programari orientat a web, les mides són en píxels.</p>
	<p><b>Acabament:</b> la línia pot acabar a la mateixa altura de l'últim node o bé pot tenir la mateixa grossària que l'amplada. També pot tenir un acabament rodó o, en alguns programes, en forma de punta, de fletxa o de cercle.</p>
 <p>Agut</p> <p>Rodó</p> <p>Biselat</p>	<p><b>Vèrtex:</b> la forma dels vèrtexs (nodes no terminals de la línia) també es pot definir. Normalment, s'ofereixen tres opcions: agut, rodó i biselat.</p>

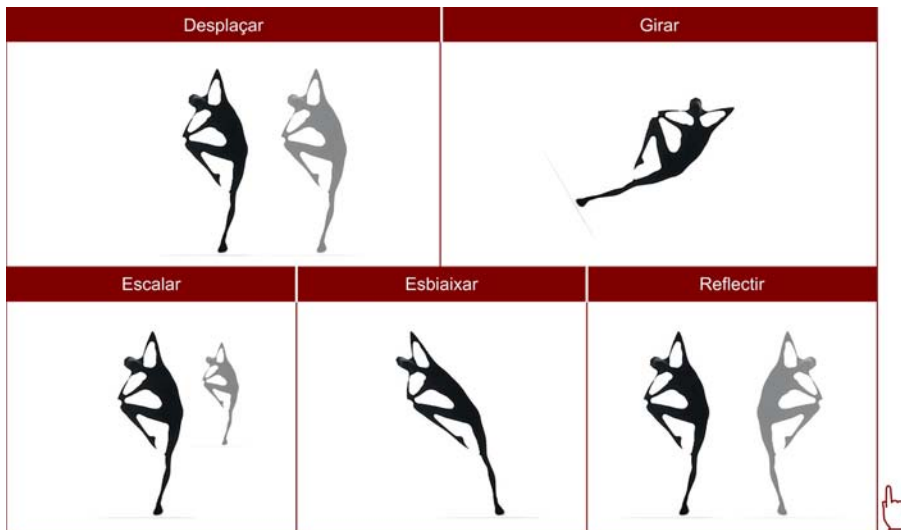
Els programes pensats per a crear gràfics vectorials que es publicaran a Internet insisteixen més en la visualització en pantalla. Per això introdueixen elements com el suavitzat (o antialiàsing) perquè la línia no es vegi dentada. També s'introdueix la simulació de traços d'eina (cal-ligràfic, carbó, aerògraf, etc.) i textures que puguin donar caràcter a la línia, aproximant-se als efectes del programari de mapa de bits.

Exemple	Descripció
	Línia bàsica "dura" (quan la línia és inclinada en veiem el dentat), de 10 píxels d'amplada, sense vora i sense textura.
	Línia bàsica "suau" (antialiàsing), amplada de 10 píxels, amb vora difuminada 50 i sense textura.
	Línia bàsica amb acabament arrodonit "suau" (antialiàsing), amplada de 30 píxels, difuminat de vora de 6 i sense textura.
	Línia bàsica amb acabament arrodonit "suau" (antialiàsing), amplada de 30 píxels, difuminat de vora de 90 i amb una textura ("Sorra") a un 40%.
	Línia bàsica amb acabament arrodonit "suau" (antialiàsing), amplada de 40 píxels, difuminat de vora de 90 i amb una textura ("Sorra") a un 95%.
	Línia "d'aerògraf", amplada de 30 píxels, difuminat de vora de 60, sense textura.

### 3.2.2. Transformacions dels objectes vectorials

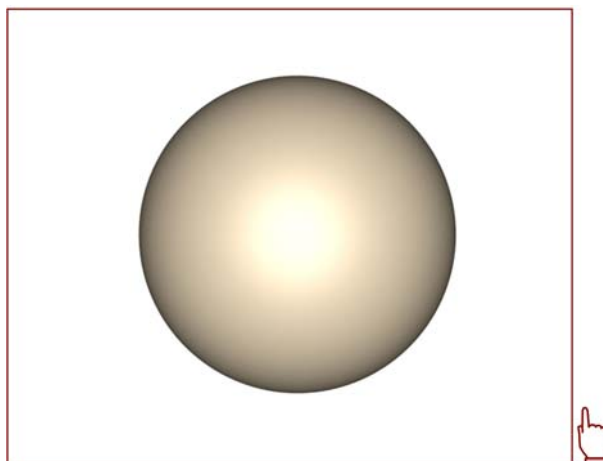
El programari vectorial és com un laboratori de geometria. Qualsevol transformació geomètrica té una eina o procediment per a aplicar-la als objectes vectorials.

Les transformacions més habituals es mostren en la taula següent:

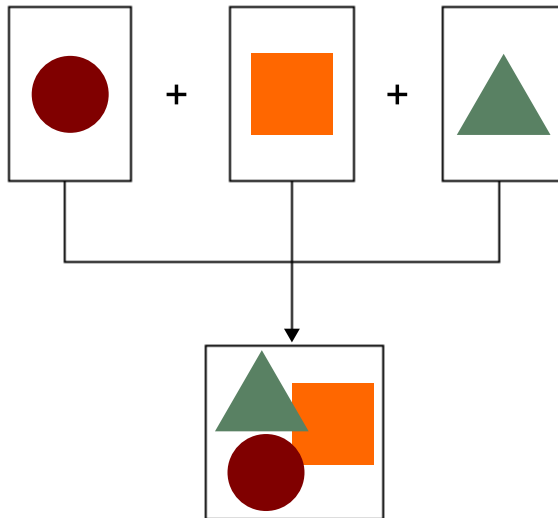


Totes aquestes transformacions es poden controlar numèricament amb precisió o bé realitzar-se de manera intuïtiva sobre el propi objecte.

Una altra possibilitat és la **fusió d'un objecte amb un altre** de diferent, de manera que es creen objectes intermedis que formen una seqüència de metamorfosi. Es pot escollir el nombre de passos, i en alguns programes (com el Flash) és possible inserir una sèrie de punts (anomenats **consells de forma**) que permeten controlar on anirà a parar un punt de l'objecte inicial en l'objecte final.



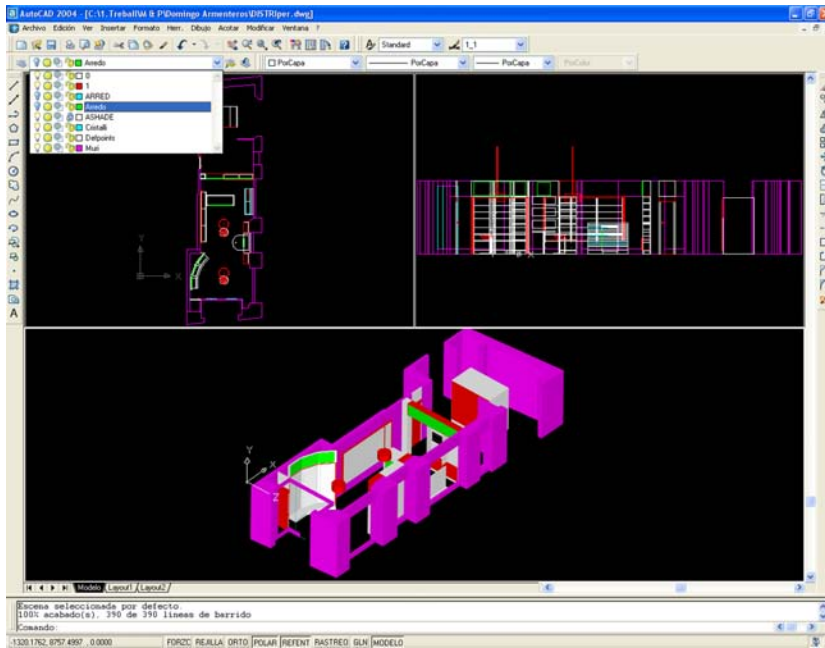
### 3.2.3. Composició de gràfics vectorials



Composició en plans o capes dels objectes vectorials

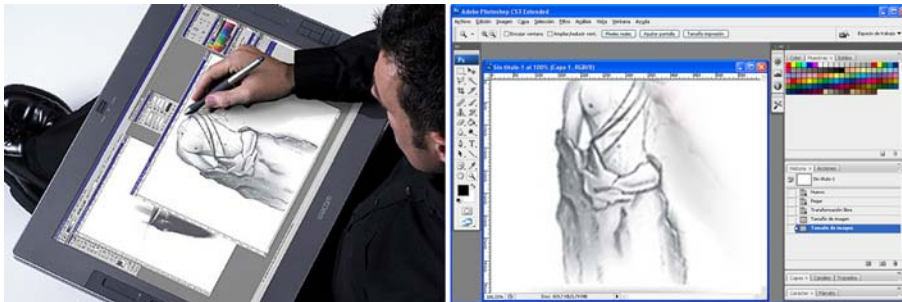
Les imatges vectorials consten de diferents objectes independents, la qual cosa permet la manipulació de cada un separatament. Aquests objectes no es troben en el mateix pla geomètric. Estan superposats i hi ha la possibilitat de passar-los cap a endavant o cap a enrere. Això és així per a la majoria de programes vectorials (Illustrator, Freehand, Fireworks, CorelDraw, Inkscape, Xara Xtreme) i potser el Flash sigui l'excepció. En el Flash els objectes comparteixen un mateix pla geomètric, i en superposar-los se sumen automàticament (tret que estiguin agrupats).

A més, la majoria de programes ofereixen la possibilitat de treballar amb **capes**. La metàfora de les capes suposa que estem treballant amb diferents fulls de paper de ceba o acetat transparent. Cada capa és independent i se superposa a l'anterior. Treballar amb capes permet organitzar millor el treball. Les capes es poden bloquejar o ocultar, de manera que es bloquegin o ocultin tots els objectes que contenen. En cada capa també hi ha diversos plans en funció del nombre d'objectes que hi tinguem.



Treball amb capes en Fireworks

### 3.3. Mà alçada: dibuixar vectors sense nodes ni agafadors

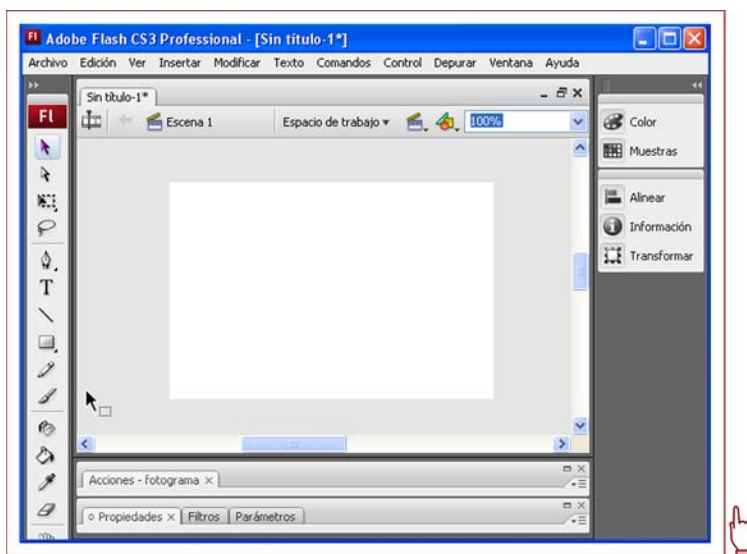


Hi ha una manera de dibuixar vectors sense usar nodes ni agafadors. En els programes de dibuix vectorial, aquesta funció correspon a l'anomenada **eina de mà alçada** o de **pinzell**, anomenada així per contrast amb l'eina de ploma per a traçar nodes. Internament el resultat continua essent objectes Bézier, però la manera de dibuixar és molt diferent. Podríem dir que més "lliure" o més "natural". Els traços es generen a partir de moviments del ratolí sense haver de marcar la línia punt a punt. En realitat, el que genera el programa no són línies sinó àrees tancades o farciments.

Com es treu més partit a la mà alçada és utilitzant una tauleta gràfica amb llapis que detecti la pressió. Els programes que són "sensibles a pressió" poden canviar el gruix de la línia generada (en realitat, farciment) en funció d'aquesta. Això representa un avantatge considerable per a donar caràcter gràfic al traç, ja que les línies convencionals tenen un gruix constant sense possibilitat de modulació. Aquesta manera de treballar està especialment indicada per a crear il·lustracions.

Encara que l'eina de mà alçada està disponible des de fa temps en el programari vectorial orientat al disseny gràfic, la seva major popularització va paral·lela a l'aparició del Flash, que la portarà a les seves màximes possibilitats. La interfície de dibuix del Flash equipara pràcticament el dibuix vectorial a la manera en què es dibuixa en un programa de mapa de bits. Encara que els objectes gràfics es basen internament en Bézier, és possible esborrar-los amb una eina de goma i en dibuixar amb un mateix color els objectes s'uneixen automàticament entre ells.

Qui pot treure més partit d'aquesta manera de treballar són els il·lustradors, fins i tot els acostumats a treballar amb eines no digitals, que troben aquí una forma similar de dibuix basada en el gest de la mà.



Cliqueu per veure una simulació animada de treball amb Flash.

Primer s'utilitza el pinzell i la goma com si es tractés d'un programa de mapa de bits; però després es pot veure com en seleccionar aquests objectes estan definits per corbes de Bézier i és possible modificar i editar els seus nodes.

### 3.4. La resolució de les imatges vectorials

Els objectes que componen una imatge vectorial presenten les seves pròpies característiques, així podem tenir objectes amb colors, mides i formes determinades, i variant una d'aquestes característiques modifiquem l'objecte. Això significa que en augmentar una imatge no distribuïm una sèrie de píxels, ni augmentem el seu nombre per superfície, l'únic que fem és variar els paràmetres de l'algorisme que calcula l'objecte. Podem ampliar el que volem i la seva qualitat no en quedarà afectada, sempre serà la màxima que ofereixi el dispositiu de sortida (pantalla, impressora, etc.).



En conseqüència, l'objecte vectorial no depèn de la resolució i, en la majoria de casos, la seva mida d'emmagatzemament és molt inferior a la que podria tenir una imatge de mapa de bits.

### 3.5. Quan podem utilitzar gràfics vectorials

Treballarem amb **imatges vectorials** quan vulguem fer un tipus d'imatge que es pugui modificar o retocar fàcilment, bé perquè es necessiti una aplicació d'aquesta imatge en **diverses mides** o bé perquè interressi obtenir un arxiu de **mida reduïda** d'emmagatzemament.

La flexibilitat de treball que permeten deriva de la independència dels seus objectes i la possibilitat de reproduir-les a qualsevol mida sense perdre qualitat.

Normalment, s'utilitzen per a treballs que s'hagin d'imprimir amb qualitat professional en resolucions altes. Aquest tipus de treballs poden ser publicacions de llibres, revistes, logotips, tríptics, etc.

També s'utilitzen en aplicacions interactives i multimèdia a Internet. La memòria d'emmagatzemament baixa dels gràfics vectorials els converteixen en el tipus d'imatge ideal per a un mitjà que està molt condicionat per la mida dels documents. Si a més es treballa amb formats que permeten afegir programació (com l'SVG o SWF) és possible manipular dinàmicament la geometria dels objectes.

Les imatges vectorials també condicionen un determinat resultat gràfic. Per la seva pròpia naturalesa són adequats per a dibuixos de línia o de colors plans. Poden contenir degradats però no en la línia de les imatges de to continu com les fotogràfiques. No és impossible, però resulta molt difícil, simular una imatge fotogràfica utilitzant vectors i, si es fa, el nombre de nodes necessaris fa que l'espai d'emmagatzemament sigui molt alt. Hi ha diverses línies gràfiques que exploren les possibilitats d'una estètica dels gràfics vectorials aprofitant les seves pròpies característiques. En aquesta línia també es fan servir per a creacions audiovisuals, des de narratives o de tipus artístic fins a publicitàries.



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



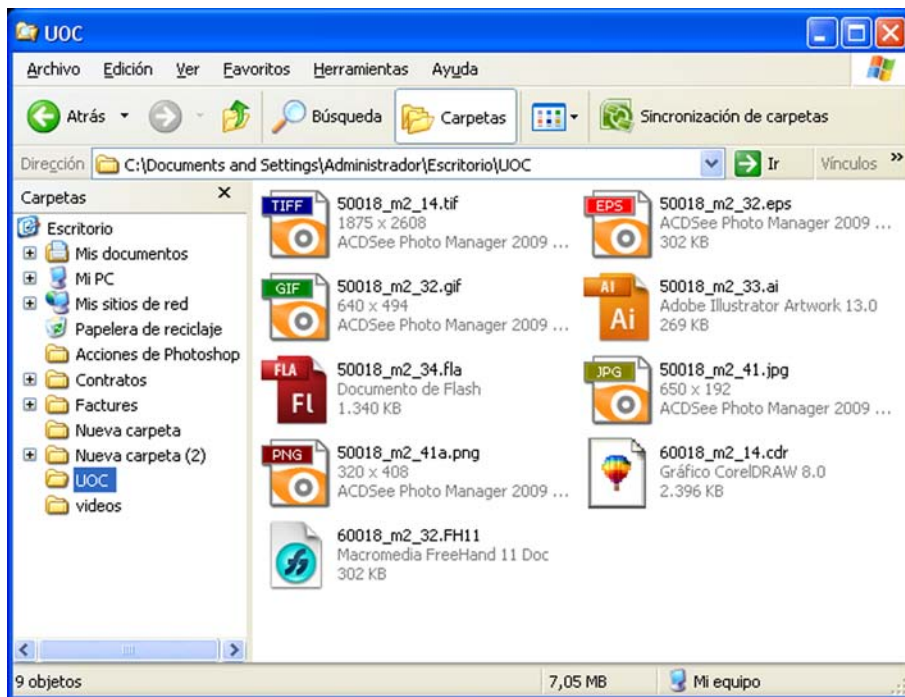
En aquest treball Luciano Lourenço ha fet servir les eines de vectorització i retoc d'Inkscape per a crear una imatge de tipus fotorealista molt convincent.  
© Luciano Lourenço (2009). *Bust de Mulher.*  
Free Art License.

En aquests treballs Antonio Roberts explora les possibilitats d'una estètica de gràfics vectorials, aprofitant les seves pròpies característiques i possibilitats com a via d'experimentació estètica. Usa la taca, la línia de mà alçada i els efectes de transparència en composicions en què la combinació de múltiples colors comparteix protagonisme amb les formes.

El treball s'ha realitzat amb Inkscape.

## 4. Formats gràfics

### 4.1. Codificar gràfics



En etapes anteriors s'ha vist que hi ha dues maneres de codificar una imatge digital:

- **Mapa de bits:** es desa la informació de color de cada píxel d'una retícula.
- **Vectorial:** es desa informació geomètrica de cada objecte.

En conseqüència, ¿només hi ha dos formats de fitxer per a codificar gràfics, un de mapa de bits i un altre de vectorial? Doncs, no. Hi ha una quantitat enorme de formats gràfics, cada un amb les seves característiques pròpies. Això en part és així perquè el desenvolupament de formats no és un procés planificat. Sorgeixen gran quantitat d'iniciatives de les universitats, de la indústria del programari i de programadors independents, tant de manera individual com coordinada. En funció de factors molt diversos, alguns d'aquests formats tenen èxit i són acceptats per tothom i d'altres no.

Però hi ha una altra raó per a la gran diversitat de formats: responen a diferents necessitats. Hi ha formats adequats per a una finalitat i formats adequats per a una altra.

Com a creadors gràfics, és important saber distingir els tipus de format i sobretot saber quin escollir en funció de la seva aplicació final.

#### 4.1.1. Tipus de format

Abans de fer un repàs dels formats gràfics més significatius vegem esquemàticament quins aspectes ens permetran distingir els uns dels altres.

Algunes característiques estan relacionades amb l'emmagatzemament de dades:

- **La manera d'emmagatzemar-los:** pot ser en píxels (mapa de bits), vectorial o amb un metafitxer (formats que admeten tant gràfics vectorials com de mapa de bits).
- **Possibilitat de característiques extra:** admetre canals alfa i/o transparència, descàrrega progressiva (per a gràfics web), especificacions per a la separació de colors (per a gràfics a imprimir), etc.

D'altres estan relacionades amb la seva **finalitat i condicions d'ús**:

- **Finalitat del gràfic:** depenent de si el gràfic està destinat a una pàgina web o a impressió serà més adequat un format o un altre.
- **Condicions d'ús del format:** si és multiplataforma (per a diferents sistemes operatius), si és un format d'ús lliure o té alguna restricció, si compleix els estàndards o no, quines empreses o organitzacions hi donen suport, etc.

#### 4.2. Formats per a gràfics vectorials i metafitxers

Podem agrupar els diferents formats gràfics en dos grans grups que corresponen a dues generacions: la primera, formada per **formats per a arts gràfiques i arquitectura**, i la segona generació, composta per **formats per a Internet i el web**.

Alguns dels formats que anomenem **vectorials** són en realitat **metafitxers**, ja que admeten gràfics de mapa de bits incrustats i també text. Per al tractament dels elements gràfics com a objectes independents, és habitual que els formats vectorials incorporin un altre tipus d'elements i siguin, en més o menys mesura, metafitxers. Aquí en parlarem conjuntament però indicant en cada cas si es tracta d'un format vectorial estricte o bé d'un metafitxer.

#### Vegeu també

En el subapartat següent, "Formats per a gràfics vectorials i metafitxers", es comentaran aquests aspectes per a formats concrets, tant de mapa de bits com vectorials.

### 4.2.1. Formats per a arts gràfiques

Durant molts anys el format vectorial més acceptat per a arts gràfiques va ser l'EPS (*encapsulated postscript*), complint el paper que més tard compliria el PDF. L'EPS recolza en el PostScript, un llenguatge de descripció d'objectes vectorials que es va desenvolupar durant la segona meitat de la dècada dels setanta i la implementació final del qual va anar a càrrec de John Warnock, que poc després fundaria Adobe Systems Inc.

El llenguatge **PostScript** és un llenguatge de programació complet i usa corbes de Bézier cúbiques, amb dos agafadors, per a descriure els objectes. Però l'EPS, a més de basar-se en el PostScript, és un format pensat per a transmetre informació a una impressora que interpreti aquest llenguatge independentment de la plataforma i el dispositiu. A més de la informació geomètrica dels elements gràfics, l'EPS desa informació de la seva posició dins d'un marc.

L'EPS admet gràfics de mapa de bits incrustats i es fa servir per a exportar imatges de mapa de bits juntament amb un traçat que les retalla del fons, la qual cosa permet tenir una silueta amb "fons transparent".

Altres formats estesos són els de Microsoft **WMF** (*windows meta file*) i el seu "successor" **EMF** (*enhanced windows metafile*). Són formats compatibles amb qualsevol programari Microsoft i que interpreten bé les dades d'altres formats vectorials com, per exemple, els formats d'arquitectura o d'enginyeria (**dwg**, **dxf**, etc.). Encara que no són formats gaire utilitzats en arts gràfiques i el seu ús sol quedar circumscrit a contextos en què es treballa amb diversos programes de Microsoft entre els que compleixen la funció de formats d'intercanvi.

El format **PDF** (*portable document file*), creat per Adobe com l'EPS, es presenta com una evolució d'aquest que compleix la doble funció de servir per a les arts gràfiques i per a Internet. Incorpora a les característiques de l'EPS el fet de no necessitar les fonts tipogràfiques per a representar bé el text. En desa la geometria i si no les troba en el sistema les dibuixa a partir de les dades que té. Es pot considerar un metafitxer que pot incorporar:

- text desat com a tal; amb informació sobre les fonts tipogràfiques per a visualitzar-lo
- gràfics vectorials per a il·lustracions i altres elements; emmagatzemats a partir de les seves dades geomètriques
- mapes de bits incrustats per a fotografies i altres imatges

La primera versió del format PDF es va llançar en 1993, i les versions successives sempre han anat acompanyades del llançament del programari de visualització i edició d'aquest, **Acrobat**, per part d'Adobe. El juliol de 2008 és acceptat per la International Organization for Standardization (ISO) com a estàndard (ISO 32000-1:2008 PDF) i la seva especificació està disponible perquè qualsevol

#### Vegeu també

En l'apartat següent, "Formats per a Internet", es comenta l'ús del PDF com a format per a la publicació a Internet.

desenvolupador pugui implementar eines de suport. Adobe manté la propietat de diverses patents sobre el format però en permet l'ús sense contrapartides mentre es compleixi amb aquesta especificació.

#### 4.2.2. Formats per a Internet

Durant molt temps un dels problemes dels formats vectorials va ser que no eren compatibles amb el llenguatge web HTML, a diferència dels arxius de mapa de bits, que de seguida es van incorporar al web i van ser reconeguts pels navegadors. La solució a aquesta mancança ja està disponible. Es tracta del format **SVG** (*scalable vector graphics*), un format basat en el llenguatge XML que té el suport del W3 Consortium (W3C), l'organisme independent que defineix els estàndards de desenvolupament del WWW. Ja té el suport de diversos navegadors, mentre que en altres, que encara no donen suport plenament al llenguatge XML, és necessari un mòdul extra per a poder-hi visualitzar arxius SVG.

L'avantatge de l'SVG és que, a més de dibuixar geometria vectorial, permet crear animacions i programar interactivitat a partir de llenguatges de *script*. Alguns programes de gràfics vectorials (com l'Illustrator o Inkscape) exporten a aquest format.

Els altres formats vectorials disponibles per a Internet no són compatibles amb llenguatges estàndards com l'XML o no s'incorporen de manera integrada amb la resta d'elements de pàgina en HTML. En aquests moments els més rellevants són *shockwave flash* (**SWF**) i *portable document file* (**PDF**).

El **PDF** funciona com a format per a l'edició electrònica i, a més, compleix la funció de format destinat a la impressió en substitució d'EPS, com ja s'ha dit. Permet que la maquetació d'elements de pàgina (gràfics i text) es mantingui estable entre plataformes, permet crear hipervincles a adreces web externes al document i afegir altres característiques de navegació i interactivitat dins d'aquest. També permet incorporar informació relativa al *copyright*.

El PDF requereix un visualitzador propi. L'Adobe desenvolupa **Acrobat Reader**, que es distribueix com a programa independent però també com a extensió (*plug-in*) del navegador. D'aquesta manera és possible veure un document PDF dins de la finestra del navegador obrint el mòdul Acrobat. El que no és possible és incrustar un gràfic en PDF dins d'una pàgina HTML. Hi ha altres visualitzadors de PDF (com l'EVince o KPDF) i diversos programes d'edició que permeten editar-lo, com el mateix **Acrobat Professional** que Adobe desenvolupa en paral·lel a l'evolució del format.

El PDF és un format habitual de distribució de documents molt estès en les administracions públiques i entre empreses. També s'usa molt sovint com a versió imprimible de les pàgines web i, des que es va publicar la seva especificació, aquesta opció es genera sovint de manera automatitzada a partir de les dades en XHTML.

El format **SWF**, sigles de *small web format*, és un format desenvolupat inicialment per FutureWave Software i Macromedia (després absorbida per Adobe) que permet crear gràfics vectorials amb animacions i interactivitat. També accepta vídeo que integri a l'interior arxius en format FLV (*flash vídeo*). Inicialment va ser el format natiu de publicació del Flash. A partir de 2008 la nova propietària Adobe va començar un procés d'obertura de l'especificació perquè altres desenvolupadors poguessin desenvolupar eines i perquè els cercadors (com Google o Yahoo) poguessin indexar el contingut dels seus fitxers.

A diferència dels documents PDF, els arxius SWF sí que es poden incrustar dins d'una pàgina HTML, però per a visualitzar-se requereixen una extensió (*plug-in*) del navegador. Mentre els navegadors no tenien integrada la interpretació de fitxers SVG, el SWF es va convertir en estàndard *de facto* per a fitxers vectorials i d'animació en el web. La diferència amb l'SVG és que l'SWF no és un estàndard obert impulsat per un organisme de consens com el W3C. La política d'Adobe és que cada nova versió del seu programa Flash implica una nova versió del format, per la qual cosa l'usuari sempre ha d'estar actualitzant a les noves versions del *plug-in*.

A més del mateix Flash hi ha altres programes de gràfics vectorials que generen arxius SWF com el Fireworks o Illustrator.

### 4.3. Formats per a mapa de bits

Hi ha una gran quantitat de formats de fitxer per a gràfics de mapa de bits.

En els formats de mapa de bits un factor important a tenir en compte és el **sistema de compressió** que utilitzen, ja que solen ser arxius que ocupen molta memòria perquè han de guardar informació per a cada píxel. És per això que alguns són més adequats per a impressió, en què la memòria d'emmagatzemament no és tan important, i d'altres per al web, en què la mida de l'arxiu pot ser important en funció de l'amplada de banda.

Un altre factor important és el relatiu a la **profunditat de color** que admeten.

Veurem aquestes qüestions per a formats concrets al llarg d'aquest apartat.

### 4.3.1. TIFF, el format d'impressió

El format habitual per a imprimir gràfics de mapa de bits és el **TIFF** (*tagged image file format*). És un format desenvolupat conjuntament per Microsoft i la desapareguda Aldus, actualment propietat d'Adobe.

El format TIFF és multiplataforma i compatible amb múltiples programes i dispositius (escàners, impressores). Està pensat per a garantir la qualitat d'impressió, i per això els arxius TIFF solen ocupar força memòria en disc en no aplicar-se a les imatges una compressió amb pèrdua de dades. Admet diferents modes de color, com ara escala de grisos, CMYK, RGB i CIE L\*a\*b.

Encara que se sol usar per a imatges de mapa de bits, en realitat és un **meta-fitxer** que permet incorporar certes dades vectorials (com un perfil de retall) i informació de pàgines o capes.

El format TIFF admet opcionalment un sistema de compressió sense pèrdua de qualitat, el conegut com a **LZW** (Lempel-Ziv-Welch). Els arxius comprimits ocupen menys memòria però triguen més a obrir-se.

### 4.3.2. Formats per al web: GIF, JPEG i PNG

Els formats de mapa de bits més habituals per al web són GIF, PNG i JPEG. Cada un té característiques diferents que el fan més adequat per a un tipus de gràfic.

**GIF** (*graphics interchange format*) és un format desenvolupat el 1987 per CompuServe Inc. per a l'intercanvi de gràfics a Internet. Des de llavors, és un dels formats més usats per als gràfics en el web. El GIF codifica els gràfics fins a 8 bits, per la qual cosa admet paletes de 256 colors o menys. És adequat per a gràfics amb colors plans i/o línies fines.

El 1989 CompuServe va treure una nova versió anomenada **GIF89a** que incorpora la possibilitat de designar un o diversos colors de la paleta com a transparents, conegut com a **transparència d'índex**. Això és útil quan es vol tenir un gràfic siluetat sobre el fons d'una pàgina web.

Una altra variant és el **GIF animat** que consta d'una seqüència de diferents fotogrames en format GIF. La successió d'aquests fa que vegem una seqüència animada. Per a realitzar aquest tipus de GIF es pot recórrer a programes creats amb aquesta finalitat, o a altres que incorporen un mòdul d'animació GIF com el Fireworks. El format GIF utilitza un **LZW** com a algoritme de compressió (sense pèrdua de qualitat).

#### Vegeu també

Les maneres i sistemes de compressió del color (escala de grisos, CMYK, RGB, CIE L\*a\*b) estan explicats en l'apartat "Color i disseny gràfic", del mòdul "Bases de disseny gràfic".

Amplieu informació sobre el sistema de compressió LZW en els apartats "Sistemes de compressió" i "La polèmica LZW".



Al costat de la imatge, el GIF incorpora la paleta de colors que s'ha fet servir. És una paleta indexada, que significa que cada color té assignat un nombre amb el qual s'indica en la imatge el color amb què es representarà cada píxel. Si una imatge té moltes varietats d'un color és possible incorporar una paleta on hi siguin totes a canvi de reduir la resta de colors. Per exemple, podríem definir una paleta amb 256 tonalitats de vermell, però la imatge només podria usar aquestes tonalitats i no altres colors.

Excepte la variant animada, els GIF es poden emmagatzemar com a **entrellaçats**. Aquesta opció és útil per al web, ja que permet que una imatge es vegi a mesura que es descarrega, i d'aquesta manera ens podem fer una idea de la imatge que apareixerà en aquest lloc.

Com a resposta a les restriccions en l'ús de l'algoritme de compressió LZW (polèmica que s'explica més endavant), la comunitat de programari lliure va emprendre el desenvolupament d'un format gràfic per al web que substituís el GIF. El resultat és un format molt millor i amb noves característiques: el **PNG** (*portable network graphics*).

Hi ha tres versions en funció del nombre de bits usats per a la codificació del gràfic: **PNG8**, **PNG24** i **PNG32**. És un format adequat tant per a colors plans com per a fotografies de to continu. Admet el **mode indexat** incorporant una paleta de colors (PNG8) i també la **mode RGB** (PNG24 i PNG32) generant varietat de colors a partir dels seus tres components (vermell, verd i blau) emmagatzemats en canals.

També admet **càrrega entrellaçada** basada en el mètode Adam7 (més ràpid que el del GIF). En la versió PNG8 admet transparència d'índex, en la versió PNG 24 no admet transparència, però en la versió PNG32 es pot guardar un **canal alfa** per la qual cosa permet diferents nivells de transparència o una silueta amb els contorns suavitzats respecte al fons. Té un sistema de compressió lliure sense pèrdua de qualitat conegut com a *deflate/inflate*, derivat de l'LZ77, que és de la mateixa "família" que l'LZW.

### El PNG no es veu en navegadors antics?

El PNG és més recent que el GIF i no és suportat per navegadors antics.

Concretament, no comença a ser suportat fins a la versió 4 dels navegadors Microsoft Explorer o Netscape Navigator i no és suportat plenament (incloent-hi transparència alfa) fins a versions superiors a la 6 de tots dos navegadors.

Sí que és suportat per les versions recents d'aquests navegadors i per les noves generacions de navegadors com el Firefox, Safari o Chrome.

Un altre format a considerar és el **JPEG** (*joint photographic experts group*), que adquireix el nom d'un comitè creat per a buscar un algoritme eficaç per a compressió d'**imatges fotogràfiques o de to continu**, tant d'escala de grisos com en color. El sistema de compressió del JPEG obté una ràtio de compressió superior que l'LZW o la que fa servir el PNG, encara que implica **pèrdua de dades**. Com més compressió, menys pes, més pèrdua i menys qualitat. Hi ha

#### Vegeu també

Sobre els canals alfa vegeu també el subapartat "Eines de selecció" de l'apartat "Gràfics de mapa de bits" d'aquest mateix mòdul.

diferents maneres d'especificar les diferents qualitats, i la més comuna és la que especifica tretze nivells, del 0 al 12, essent 12 la de més qualitat i 0 la de menys.

Les imatges JPEG admeten **càrrega progressiva**, semblant a la càrrega entrelaçada. Aquest tipus de càrrega permet que quan comencen a arribar les dades de la imatge del servidor es dibuixi una versió de baixa definició (amb aspecte molt pixelat), i a mesura que es descarrega la imatge es fan escombratges progressius que augmenten la seva resolució fins a arribar a la versió final.

Aquest format codifica el **color en RGB** i **no permet treballar amb paletes indexades**. **No suporta cap tipus de transparència**. És un bon format per a representar degradats, però no contorns fins, detalls o àrees de color pla. Per aquest motiu resulta adequat per a imatges fotogràfiques, per a les quals es va crear, però no per a imatges amb línies o textos, encara que hi ha estratègies de compressió que resolen aquest problema que es veuran en seccions posteriors.

#### 4.3.3. Sistemes de compressió

Els **sistemes de compressió** redueixen la memòria que ocupa un arxiu. **Comprimir** pot ser necessari perquè ocupi menys espai en emmagatzemar-lo en el disc, però es fa imprescindible quan aquest arxiu s'ha de transferir o publicar a Internet. Com més petit és l'arxiu, més ràpida és la transferència de les dades. Una vegada es tenen les dades, l'arxiu es descomprimeix i es mostra.

La descompressió també representa una pèrdua de temps, però en general, per als sistemes de compressió/descompressió i les capacitats de processament dels dispositius electrònics actuals, el temps perdut en la descompressió és inferior al guanyat per la disminució de la mida de l'arxiu.

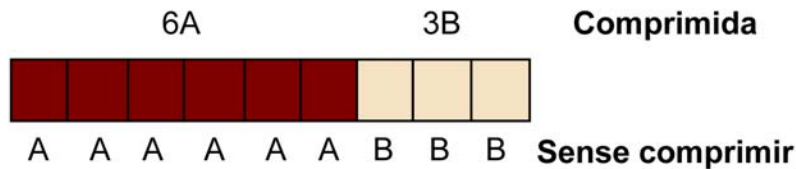
D'alguna manera, el que fan els sistemes de compressió és eliminar redundància i "resumir" la informació a emmagatzemar. Això es pot fer de dues maneres:

- **Amb pèrdua de dades.** Equiparar informació semblant a un sol valor (per exemple, diversos tons de blau a un sol blau). Es resumeix la informació i perden detalls. La informació destruïda no es recupera en la descompressió. Aquest tipus de compressió és el que s'utilitza, per exemple, en el format JPEG.
- **Sense pèrdua de dades.** Es busquen maneres d'optimitzar el registre de les dades eliminant la informació redundant i simplificant la manera d'expressar-la. Tota la informació "reduïda" es torna a recuperar en la descompressió. Aquest tipus de compressió és el que s'utilitza, per exemple, en els formats TIFF, GIF o PNG.

#### Vegeu també

Sobre les estratègies per a evitar pèrdua de qualitat aparent en arxius JPEG que incloguin text o línies fines vegeu el subapartat "Optimització d'arxius de mapa de bits en format PNG i GIF" de l'apartat "Optimització de gràfics per a web" del mòdul "Producció de gràfics".

L'**LZW** (Lempel-Ziv-Welch) és l'algoritme que usa el format GIF i, opcionalment, el format TIFF per a comprimir els gràfics (també l'utilitza el PDF). Es tracta d'un sistema de compressió sense pèrdua de dades. A grans trets, es pot dir que aquest algoritme codifica seqüències consecutives de valors de color, i redueix al màxim la redundància en la repetició de seqüència.



Un sistema de compressió sense pèrdua de dades és el *run-length encoding* (RLE), en el qual seqüències de dades consecutives amb el mateix valor s'emmagatzemen de manera "resumida" com un únic valor més el seu recompte.

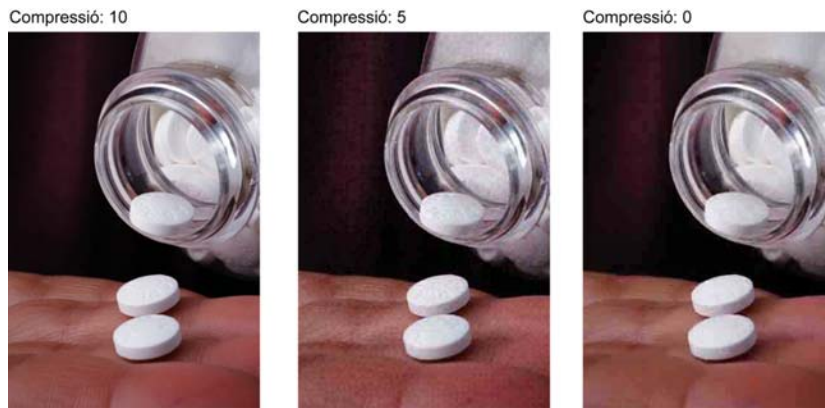
Observa de manera esquemàtica el que podria fer un sistema de compressió RLE amb aquesta fila de 9 píxels per a resumir la informació.

Realitza un procés de reconeixement de seqüències consecutives de color, és a dir, recorre un a un tots els píxels. Si hi ha més d'un píxel consecutiu del mateix color, emmagatzema les dades d'aquest píxel i la quantitat d'aquestes que hi ha en la seqüència. El resultat: una informació que requeria nou dígit (AAAAA BBBB) es pot expressar amb només quatre dígit (6A3B).

Els sistemes derivats d'LZ77 (com l'LZW) parteixen d'aquesta compressió com a base i la milloren tenint en compte la repetició en seqüències de dígit (com podria ser en AABAABAAB).

El sistema de compressió **deflate/inflate** que utilitza el PNG és un derivat de l'LZ77 (l'algoritme utilitzat en zip, gzip, pkzip, etc.). El sistema seria similar al que hem descrit en la il·lustració de l'apartat anterior però optimitzat, ja que introdueix diversos filtres que preprocessen la informació de color per a aconseguir una codificació més eficient. També és un **sistema de compressió sense pèrdues**.

El sistema utilitzat pel JPEG és molt bo comprimint imatges fotogràfiques, però no amb imatges de colors plans o amb línies fines molt contrastades. Bàsicament, el que fa aquest algoritme és guardar la imatge separant la informació de brillantor dels matisos de color i elimina les diferències de color subtils que no podem apreciar; per això és un format de compressió que genera un procés de pèrdues. Per molt poc que vulguem comprimir, sempre es perd informació. Com ja s'ha dit, es pot controlar el nivell de compressió, i com major sigui la compressió aplicada, menor serà la qualitat de la imatge i el volum de l'arxiu.



#### 4.3.4. La polèmica LZW

Ja s'ha esmentat anteriorment com les restriccions d'ús de l'LZW van impulsar el desenvolupament del PNG. Fem un breu repàs d'aquest procés.

El sistema de compressió LZW (Lempel-Ziv-Welch) va ser desenvolupat l'any 1983 per Terry Welch millorant els algoritmes LZ77 i LZ78 desenvolupats el 1977 i el 1978 respectivament pels investigadors israelites Jacob Ziv i Abraham Lempel. El sistema es patentà als Estats Units l'any 1985, i actualment aquesta patent pertany a l'empresa Unisys.

El 1987 CompuServe crea el seu format per a intercanvi de gràfics a Internet GIF (*graphics interchange format*), que incorpora el sistema de compressió LZW, igual que el seu successor GIF89a de 1989. Durant aquests anys, el GIF es converteix en l'estàndard d'intercanvi d'imatges a Internet (encara no existia el JPEG) i després també és l'estàndard per a imatges sobre pàgines web que tots els navegadors interpreten. Fins al final de 1994 Unisys no reclama res, però a partir d'aquell moment, reforçat per un acord amb CompuServe, reclama el pagament de regalies o *royalties* als desenvolupadors de programari que suporten GIF. Això és incompatible amb els programes de codi obert.

Davant d'aquesta situació alguns programadors de la comunitat de programari lliure es posen en marxa per crear un format alternatiu. Durant el 1995 el desenvolupament avança ràpidament amb múltiples aportacions. S'utilitza l'LZ77, que sí era lliure de patents, i la codificació de Huffman. El 1996 el PNG ja està disponible com a format lliure de patents i es converteix en recomanació del W3C. CompuServe anuncia que la seva pròxima versió del GIF (GIF24) abandonarà el mètode LZW i es basarà en l'especificació PNG. El juny de 2003 la patent de l'LSW expira als Estats Units, però Unisys adverteix que continua vigent en altres països. Segons The Software Freedom Law Center, a partir d'octubre de 2006 ja no queden patents rellevants en vigor sobre l'LSW.

El resultat d'aquest procés és enormement positiu. Actualment disposem d'un nou format (PNG) molt millor que el GIF i lliure de restriccions per al seu desenvolupament i ús. L'únic inconvenient per a la implantació total de l'ús del PNG en el web era que els navegadors antics no l'interpretaven. Però el problema ha desaparegut amb les noves versions.

Concretament, Microsoft Internet Explorer no el va suportar fins a la versió 4 i no va interpretar les seves característiques de manera acceptable (com a transparència alfa, etc.) fins a les versions 6 i 7. Netscape Navigator també va començar a suportar-lo en la versió 4, però no és fins a la versió 6 que suporta les noves característiques del format. El Firefox ja va néixer suportant plenament aquest format. L'avantatge en aquest procés és que el sistema de codificació del PNG implementa cada nova característica en parts separades, de manera que un navegador que no les interpreti no deixa de mostrar la imatge.

Com veiem, no tots els formats de fitxer estan lliures de restriccions. I això pot tenir relació, com en aquest cas, amb un sistema de compressió de dades que utilitza el format en qüestió. El cas del PNG és en part una excepció, però també ho és el del GIF. En general, els propietaris dels formats no en restringeixen l'ús ni cobren per ell, encara que en molts casos no permeten que altres persones o organitzacions el desenvolupin.

#### 4.4. Programari i formats

A més dels formats que hem vist fins aquí, hi ha altres formats de gràfics que s'usen quan hi treballem en un programa d'edició. Són els **formats nadius** o **formats font** del programa.

Els programes d'edició i tractament de gràfics necessiten un format per a desar la informació mentre no s'ha acabat el treball. Generalment es tracta d'informació addicional pròpia del programa que permet desar un arxiu per a continuar treballant-hi, com un projecte no acabat que encara conté informació que després no es veurà en la imatge final (per exemple, la informació de capes). També, per a no perdre aquesta informació addicional, la majoria de programes permeten desar o exportar en formats diferents del format natiu. D'aquesta manera podem publicar un gràfic en format PNG o JPEG en un web mentre mantenim una versió en format natiu del nostre editor de gràfics, en què la imatge té diverses capes i el text es manté editable, per si hem de fer modificacions.

Alguns d'aquests formats s'acaben convertint en estàndards perquè milloren sensiblement la manera de desar informació, perquè introdueixen característiques noves o perquè el programa és un líder en el sector. És el cas, no com a format de publicació però sí com a format d'intercanvi entre programes, del

format natiu del Photoshop (que té l'extensió \*.psd). Programes com Gimp o Fireworks obren els arxius d'aquest format i recuperen l'estructura de capes i altres dades de treball emmagatzemades.

El problema amb els formats nadius és que se solen mantenir com a propietat de l'empresa desenvolupadora, la qual cosa implica arribar a acords amb ella o bé fer enginyeria inversa per a generar el format esmentat des de programari de la competència. A més solen aparèixer noves versions amb cada nova versió del programa i no sempre són compatibles amb les anteriors.

El cas contrari és que un nou programari adquireixi com a nadiu un format ja existent. És, per exemple, el cas del Fireworks amb el PNG. Com que el PNG és un format lliure, Macromedia, que aleshores era l'empresa desenvolupadora, hi va poder incorporar característiques pròpies com l'acceptació de gràfics vectorials que després es poden recuperar des del mateix programa o des del Flash. Encara que l'ús de la mateixa extensió pot confondre els usuaris, ja que no és el mateix un format PNG natiu del Fireworks que l'estàndard PNG, i tots dos usen la mateixa extensió \*.png. Aquest és un cas, i no l'únic, en què l'extensió no ens informa prou bé sobre el contingut d'un fitxer.

En el cas del GIMP, el seu format natiu (XCF, *experimental computing facility*), en ser lliure, s'ha difós com a format d'exportació o importació per a altres programes d'edició gràfica també lliures.

Hi ha formats de publicació que també han nascut vinculats a un programa. No són formats font (no guarden informació de treball), però sí que en certa manera són formats nadius, ja que estan vinculats al tipus d'informació que el programa genera. És el cas de formats que hem esmentat, com el SWF i el PDF, que continuen vinculats al desenvolupament i actualització del Flash i l'Acrobat respectivament. I aquest vincle continua essent important encara que el format es converteixi en un estàndard oficial (com en el cas del PDF).

#### 4.5. Informació ràpida sobre formats

En la taula següent es recull de manera esquemàtica la informació bàsica d'alguns dels formats més usats.

Format	Característiques
<b>JPEG</b> <i>joint picture expert group</i>	Extensió *.jpeg *.jpg. Mapa de bits. Mode RGB. 24 bits. Mètode propi de compressió amb pèrdua de dades. No admet transparències. Admet càrrega progressiva. Adequat per a imatges de to continu com les fotogràfiques. És capaç de reduir molt la mida dels arxius. Adequat per al Web.

<b>Format</b>	<b>Característiques</b>
<b>GIF</b> <i>graphical interchange format</i>	<p>Extensió *.gif.            Versions: GIF87, GIF89a i GIF animat.            Mapa de bits.            Mode indexat.            8 bits.            Mètode de compressió LZW sense pèrdua de dades.            Admet transparència d'índex (GIF89a).            Admet càrrega entrelaçada (GIF87 i GIF89a).            Admet seqüència de fotogrames (GIF animat).            Desenvolupat per Compuserve i propietat d'aquest. Mètode de compressió LZW patentat per Unisys.            Adequat per a imatges amb pocs colors i/o colors plans.            Adequat per al web.</p>
<b>PNG</b> <i>portable network graphics</i>	<p>Extensió *.png.            Versions: PNG-8, PNG-24 i PNG-32.            Mapa de bits.            Mode indexat (PNG-8) o RGB (PNG-24 i PNG-32).            Versions del format per a 8, 24 bits i 32 bits.            Mètode de compressió <i>deflate/inflate</i> sense pèrdua de dades.            Admet transparència d'índex (PNG-8) i transparència alfa (PNG-32).            Admet càrregues entrelaçades (Adam7).            Desenvolupat per programadors independents del moviment de programari lliure.            Adequat tant per a colors plans com per a to continu.            Adequat per al Web. No es visualitza o es visualitza malament en navegadors antics.</p>
<b>TIFF</b> <i>tagged image file format</i>	<p>Extensió *.tiff *.tif.            Mapa de bits.            Modes RGB, CMYK, CIE L*a*b i escala de grisos.            32 bits.            Mètode de compressió LZW opcional.            Desenvolupat per Aldus i Microsoft. Propietat d'Adobe.            Adequat per a impressió. Requereix força memòria d'emmagatzemament.</p>
<b>BMP</b> <i>windows bitmap format</i>	<p>Extensió *.bmp.            Mapa de bits.            Modes indexat i RGB.            24 bits o menys.            Desenvolupat per Microsoft Corp i propietat d'aquest.            Comprimeix poc però s'obre ràpidament. Adequat per a gràfics a visualitzar en pantalla.            No es visualitza en els navegadors web.</p>
<b>PICT</b> <i>Macintosh picture</i>	<p>Extensió *.pic *.pict.            metafitxer.            Mode indexat i RGB.            24 bits o menys.            Desenvolupat per Apple Computer i propietat d'aquest.            Encara que és un metafitxer se sol usar per a mapes de bits. Ràpid en descompressió. Adequat per a gràfics a visualitzar en pantalla i també com a format d'intercanvi entre programes.</p>
<b>Adobe Photoshop</b> (format natiu)	<p>Extensió *.psd.            Mapa de bits.            Modes CIE L*a*b, RGB, CMYK, indexat, bitò, escala de grisos i ploma.            32 bits.            Admet canals alfa i capes, a més d'altres informacions del programa.            Desenvolupat per Adobe Systems i propietat d'aquest.            Adequat com a format de treball en Photoshop o per a transmetre arxius d'aquest programa a un altre que l'interpreti.</p>
<b>WMF</b> <i>windows meta file</i>	<p>Extensió *.wmf.            Metafitxer.            Desenvolupat per Microsoft Corp i propietat d'aquest.            Encara que és un metafitxer s'usa sobretot per a gràfics vectorials. Adequat com a format d'intercanvi entre programes.</p>

<b>Format</b>	<b>Característiques</b>
<b>EMF</b> <i>enhanced windows metafile</i>	Extensió *.emf. Metafitxer. Versió millorada del WMF. Desenvolupat per Microsoft Corp i propietat d'aquest. Encara que és un metafitxer s'usa sobretot per a gràfics vectorials. Adequat com a format d'intercanvi entre programes.
<b>EPS</b> <i>encapsuled postscript</i>	Extensió *.eps. Vectorial, també admet mapes de bits amb traçats. Utilitza el llenguatge PostScript de descripció de vectors. Desenvolupat per Adobe Systems i propietat d'aquest. Adequat per a impressió.
<b>PDF</b> <i>portable document format</i>	Extensió *.pdf. Vectorial, admet mapes de bits incrustats. Evolució de l'EPS. Conserva la informació de les tipografies. Té la possibilitat d'afegir hiperenllaços i interactivitat. Requereix un programa de visualització o un <i>plug-in</i> de navegador per a visualitzar-se. En el web es visualitza independentment de la pàgina HTML. Desenvolupat per Adobe Systems i propietat d'aquest. És adequat per a impressió i també per a la distribució de documents electrònics imprimibles per Internet. El 2008 va ser reconegut com a estàndard ISO.
<b>SWF</b> <i>shockwave Flash</i>	Extensió *.swf. Admet gràfics vectorials, arxius de mapa de bits incrustats, animacions i elements interactius. Requereix un connector o <i>plug-in</i> per a visualitzar-se en el web, i Flash Player per a executar-se en disc. Es visualitza integrat ( <i>embed</i> ) en la pàgina HTML. Desenvolupat per Future Wave Software, Macromedia i Adobe. Propietat d'Adobe. És adequat per a gràfics vectorials, pel·lícules animades i aplicacions interactives, tant per al web com per a reproducció directa de disc.
<b>SVG</b> <i>scalable vector graphics</i>	Extensió *.svg. Vectorial, admet animacions i programació. Basat en XML i per tant compatible amb HTML. No requereix cap tipus de connector o <i>plug-in</i> per a visualitzar-se sobre HTML, encara que a la pràctica en necessita, ja que els navegadors principals (MS-Explorer i Netscape) encara no el suporten. Homologat i impulsat pel W3 Consortium. Desenvolupat per un consorci d'empreses i organitzacions. Format lliure. Adequat per a gràfics vectorials i sistemes interactius en el web. És l'estàndard obert per a gràfics vectorials a Internet. És una "recomanació" (equivalent a estàndard) del W3C des de 2001.



## 5. Programari de creació i tractament gràfic

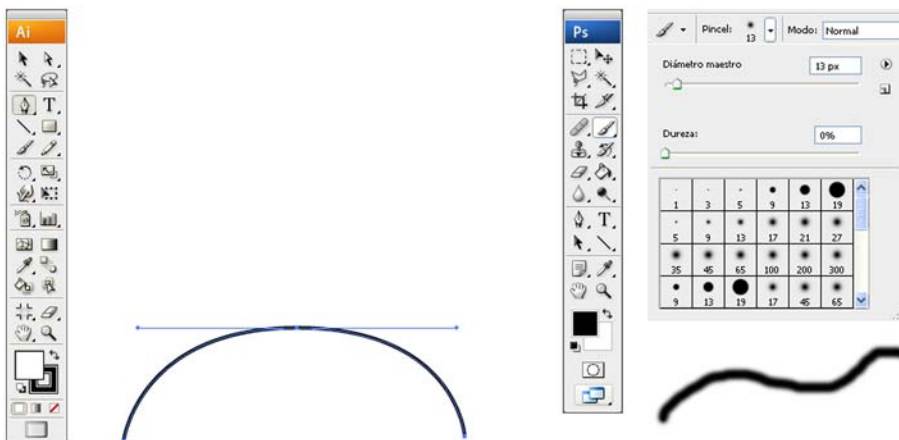
Afortunadament l'oferta de programes de creació i tractament de gràfics és àmplia i variada. En aquesta secció s'exposa una panoràmica d'aquesta oferta, proporcionant coneixements i orientacions per a escollir l'eina en funció del tipus de tasca a realitzar i les condicions de treball.

Per a presentar el programari de gràfics es revisen les seves característiques a partir de tres binomis:

- programari de mapa de bits (*bitmap*) i vectorial
- programari orientat a impressió i orientat al Web
- programari de propietat i *open-source*

Cada programa es pot adscriure a diverses d'aquestes categories. Però el seu estudi per separat ens en pot facilitar la comprensió.

### 5.1. Programari *bitmap* i vectorial



La ploma Bézier i el pinzell de píxels són el paradigma de dues maneres de treballar diferents.

Anteriorment ens hem referit a la importància de la distinció mapa de bits / vectorial. Aquesta distinció també es dona en el camp dels programes de gràfics. Hi ha programes de gràfics vectorials que treballen amb objectes Bézier i programes de gràfics de mapa de bits que treballen amb imatges compostes per píxels. Això es tradueix en la configuració dels entorns de treball.







Els **entorns de treball vectorials** estan més orientats al dibuix geomètric; en canvi els **entorns de mapa de bits** tenen una orientació més orientada al dibuix "gestual". De fet, una distinció que es feia en els primers programes de gràfics en la dècada dels vuitanta era entre programes de dibuix (vectorials) i

programes de pintura (mapa de bits). Una altra referència per als programes de mapa de bits és la fotografia i el retoc fotogràfic; mentre que els vectorials poden prendre com a referent el disseny gràfic, la il·lustració i el *collage*.

Encara que, com veurem, aquestes maneres de treballar s'acaben hibridant, l'imaginari de referència va marcar els primers desenvolupaments d'entorns de treball i ens ha deixat un llegat de metàfores que encara es manté.

### 5.1.1. Entorns de treball vectorial





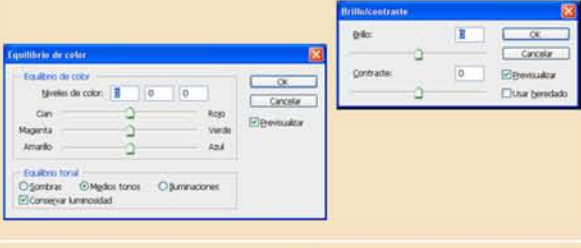

En la taula següent es mostren els elements principals de treball en un programa de gràfics vectorials.

Principals elements de treball en un programa de gràfics vectorials	
Eines per a objectes Bèzier	<p>Per a crear objectes Bèzier disposem d'una eina que crea els nodes (<b>ploma</b>), eines de selecció (<b>fletxes</b>) per a moure'ls o extraure i controlar els agafadors i d'una <b>fulla</b> per a tallar les línies en objectes independents. També disposem d'una eina per a crear línies rectes i d'una altra per a crear directament figures geomètriques, com rectangles, el·lipsis i polígons.</p> 
Eines de transformació	<p>Els objectes creats es poden transformar geomètricament. Disposem d'eines i/o procediments per a <b>girar, desplaçar, escalar, esbiaixar o reflectir els objectes</b>. També disposem d'opcions de menú per a combinar entre ells els objectes. La manera de fer-ho difereix d'un programa a un altre.</p> 
Control de propietats	<p>Els objectes són independents entre ells i tenen una sèrie de propietats que són parametrizables. D'una manera o una altra els programes ens ofereixen una forma de canviar aquestes propietats seleccionant l'objecte en qüestió i accedint-hi per mitjà d'una finestra. Les propietats es refereixen a l'amplada i al tipus de línia, al <b>farciment de l'objecte</b>, a la <b>grandària i posició</b>, etc.</p> 
Alineació	<p>L'alineació dels objectes respecte als altres també es pot controlar. Disposem d'un sistema de control pel qual seleccionant diversos objectes ens podem <b>alinejar centrats, a l'esquerra, a la dreta o distribuir-los de manera que l'espai entre ells sigui igual</b>.</p> 
Capes	<p>Els programes vectorials poden treballar amb capes. En cada capa hi ha un conjunt d'objectes, les capes es poden <b>ocultar o bloquejar</b> per a no moure algun objecte per error.</p> 
Color	<p>El color en un programa vectorial és una propietat de l'objecte. Podem seleccionar un objecte i <b>assignar-li un color</b>. També es poden <b>crear colors nous</b> usant els selectors de color. Però atenció: a cada color li correspon una codificació (CMYK, RGB, color de biblioteca Pantone, etc.). Cada objecte podria tenir una codificació diferent i això seria problemàtic, per exemple, si s'ha d'imprimir la imatge. És important tenir clar amb quina codificació de color es treballarà.</p> 

Amb algunes variacions, aquest és l'entorn de treball en programes com l'Illustrator, Inkscape, Corel Draw, Inkscape, Xara Xtreme, Fireworks o Free-hand. De fet, a Fireworks canvien algunes coses i encara més en el Flash, com veurem més endavant.

### **5.1.2. Entorns de treball *bitmap***

En la taula següent es mostren els elements principals de treball en un programa de gràfics de mapa de bits.

Principals elements de treball en un programa de gràfics de mapa de bits	
<p>Eines per a dibuix i pintura</p>	<p>Les principals eines per a dibuixar píxels són el <b>llapis</b>, la <b>brotxa</b>, la <b>goma d'esborrar</b>, l'<b>aerògraf</b> i el <b>pot de pintura</b> que farceix àrees. A més a més, disposem d'un control de paràmetres per al llapis i la brotxa (forma, duresa, etc.). L'eina de <b>comptagotes</b> permet escollir un color de la imatge.</p> 
<p>Eines de selecció</p>	<p>Per a seleccionar àrees de píxels hi ha les eines de selecció, com la <b>selecció regular</b> (rectangle o el·lipse), el <b>llaç</b> per a seleccions irregulars i la <b>vareta màgica</b> per a seleccionar píxels similars contigus o de tota la imatge. Les eines de selecció es complementen amb una sèrie d'opcions de menú relacionades amb la selecció.</p> 
<p>Selector de color</p>	<p>El <b>selector de color</b> serveix per a escollir el color amb què pintar. S'ofereixen diversos sistemes per a definir el color (HSV, RGB, CIE L*a*b, etc.).</p> 
<p>Canals</p>	<p>Els programes de mapa de bits representen el color a través de canals. En cada canal hi ha un component de color. Per exemple, en el mode RGB tenim tres canals: vermell, verd i blau. És possible visualitzar solament un canal i treballar solament amb aquest component en escala de grisos.</p> 
<p>Ajustar</p>	<p>Els programes de mapa de bits tenen un referent en el retoc fotogràfic de laboratori. Per això ens ofereixen eines precises per <b>modificar</b> i <b>ajustar</b> la imatge.</p> 
<p>Filtres algorítmics</p>	<p>La disposició dels píxels en la retícula es pot modificar aplicant-los un algoritme, ja que al cap i a la fi es tracta de dades numèriques. Hi ha una gran diversitat de filtres algorítmics que creen efectes determinats en la imatge intervenint en el color dels píxels. Els programes de gràfics incorporen la possibilitat d'aplicar aquests filtres que moltes vegades són petits programes externs que s'utilitzen des del programa.</p> 

### 5.1.3. Confluències

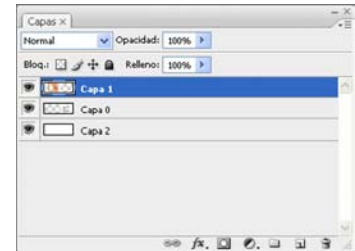
Com s'ha vist anteriorment, el tipus de codificació de la imatge (vectorial o mapa de bits) sol condicionar el tipus d'interfície de l'entorn de treball i una determinada manera de treballar.

Ara bé, amb els anys s'ha vist que una interfície pensada per al dibuix vectorial podia ser útil en un programa de mapa de bits i viceversa. Ha començat una convergència de procediments de treball entre els dos tipus de programes.

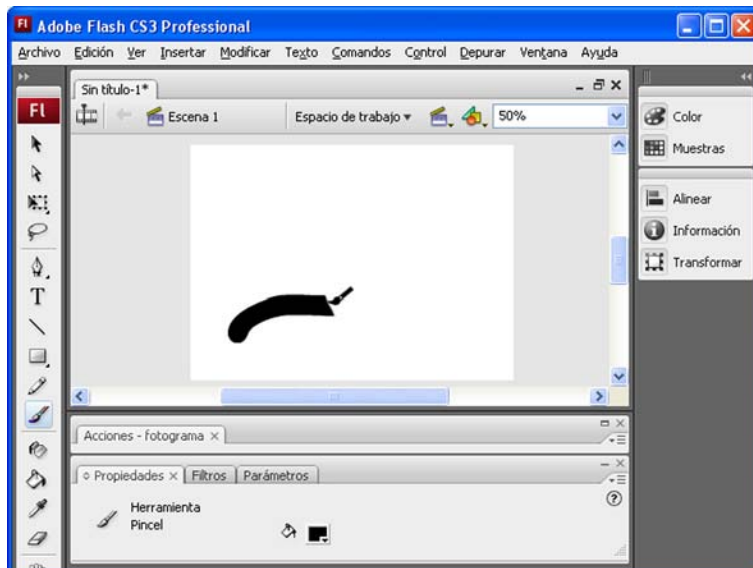
Un dels exemples més clars és l'ús de les **capes** en **programes de mapa de bits**, que inicialment no en tenien i en van adoptar sota la influència de l'experiència de treball amb programes vectorials. Amb un sistema de capes, una part d'una imatge pot mantenir-se independent de la resta (com si fos un objecte vectorial) i s'hi poden aplicar ajustaments o filtres de manera autònoma.

Un altre element que els programes de mapa de bits han adoptat és un bon **control tipogràfic**, que al principi es reservava als programes vectorials.

Una influència en direcció contrària és la del **pinzell** i la **goma** per part de programes vectorials. El cas més clar és el Flash. Amb aquestes eines s'aconsegueix que l'usuari pugui dibuixar d'una manera molt més intuïtiva sense gairebé adonar-se que està generant objectes Bézier. Les "línies" del pinzell no són tals, en realitat s'estan generant àrees, objectes tancats.



Finestra de capes a Photoshop. Diferents parts de la imatge són en capes diferents. Les capes es poden ocultar i també enllaçar entre elles.



Il·lustració realitzada amb el pinzell de Macromedia Flash

Un cas paradigmàtic d'aquesta confluència és el programa de gràfics **Fireworks**. En principi es tracta d'un programa de dibuix vectorial que, com la majoria, admet la importació de gràfics de mapa de bits. Però no solament és això, Fireworks permet crear directament gràfics de mapa de bits i modificar-los. De fet, Fireworks integra totes les característiques pròpies d'un programa de mapa de bits i d'un programa de dibuix vectorial. El millor exemple és la seva finestra d'eines amb un grup per a l'edició en mapa de bits i un grup per a l'edició vectorial. En el seu entorn de treball conviuen elements vectorials amb elements de mapa de bits. La seva finestra de propietats és contextual i canvia en funció de l'element escollit.

La codificació definitiva de la imatge s'escull al final. El Fireworks treballa en una versió de PNG nativa, però per a gravar qualsevol altre format ha d'exportar. Una finestra d'optimització permet configurar les característiques quan es vol gravar en un format de mapa de bits.

## 5.2. Programari orientat a impressió i orientat al web

La finalitat última dels gràfics a generar condiona les característiques del programa que s'ha de triar per a la seva edició.

El procés de producció pot ser significativament diferent segons si la destinació final és un producte imprès o si és un producte que es distribuirà en entorns digitals i de visualització en pantalla, i això determina les eines. Depenent de l'aplicació final de la imatge, les característiques del programari editor seran diferents.

En el cas del gràfic que s'ha d'imprimir ens preocupa veure en pantalla una "simulació" el més aproximada possible al resultat en paper. El programari ha de simular els efectes de la superposició de tintes, de la quadricromia, etc. En canvi, el gràfic que s'ha de veure en pantalla s'ha d'optimitzar per a reduir la mida del fitxer (pes de l'arxiu) i s'ha de comprimir sense perdre qualitat. El programa de gràfics ha de tenir eines per a aquesta optimització.

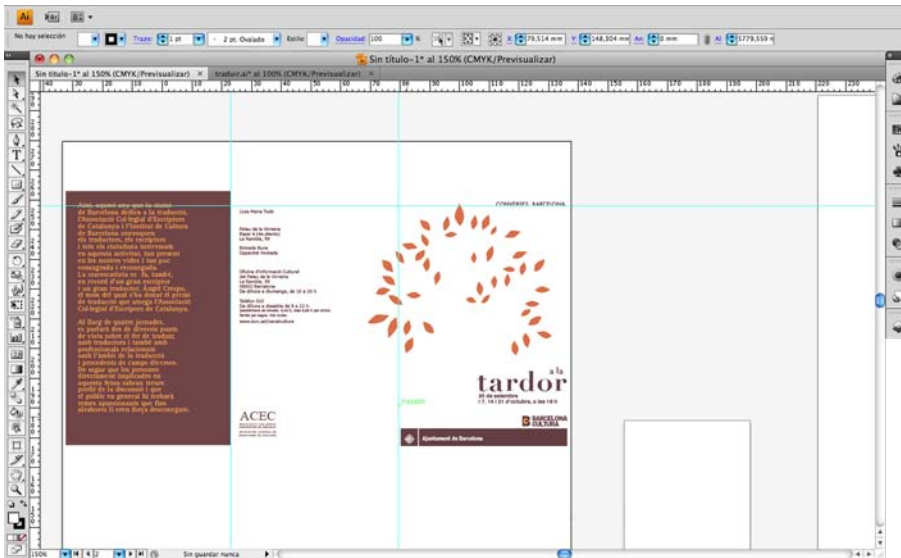
Molts editors de gràfics estan pensats per a satisfer les dues necessitats, no solament per a incloure un ventall més ampli de funcionalitats, sinó també perquè molts gràfics han de tenir les dues aplicacions (impresa i en pantalla). Però encara en aquests casos hi sol haver un punt en el qual el procés de producció es bifurca, i genera dues versions del gràfic i utilitza recursos diferents del programa.



Eines de  
Macromedia  
Fireworks

En general, la destinació final condiona l'entorn de treball, i els productes de referència que el configuren solen ser la **publicació impresa** i la **publicació al web**.

### 5.2.1. Programari orientat a impressió



Interfície de treball de l'Illustrator amb les icones de canvi de pàgina, les regles i les guies. El símbol és la taula del dissenyador gràfic, el seu espai de treball.

Els programes orientats a impressió utilitzen la metàfora de la pàgina per a l'entorn de treball. Es treballa amb mesures físiques (centímetres, polzades, etc.), es disposa de regles de pàgina, de guies i retícules que ajuden en la maquetació (disposició de les columnes de text, gràfics, titulars, etc.). Es pot definir una àrea de sagnia fora de la pàgina en la qual estenem els elements que quedaran en el límit de tall de pàgina perquè no quedi una línia blanca.

Es disposa d'un bon control tipogràfic que inclou, a més de l'elecció de la font i la mida, definició d'interlletratge (o *kerning*), interlineat, paràmetres de paràgraf, alineació, línia de base, etc.

Un altre factor important és determinar la densitat de trama amb què s'imprimiran les imatges de to continu (fotografies, determinades il·lustracions, etc.). Quant als colors escollits permet escollir entre quadricromia o tintes planes i definir aspectes com la possibilitat que una tinta se superposi a una altra o bé es creï una "reserva".

#### Quadricromia i tintes planes

La **quadricromia** és un sistema de representació del color que a partir d'un tram de quatre tintes aconsegueix simular una àmplia varietat de colors. Els sistemes de quadricromia usen tres colors bàsics (cian, magenta i groc) més el negre, i fan servir el blanc del paper com a cinquè color. Amb la combinació de trames de les quatre tintes s'aconsegueix reproduir fotografies i qualsevol tipus d'imatges.

En arts gràfiques es parla de **tintes planes** o **tintes directes** quan en comptes de quadricromia s'utilitzen tintes de colors específics, normalment per a elements gràfics amb colors sòlids, encara que també es pot usar en tramats. Les tintes directes són barrejades



Propietats de text en l'Illustrator

pel fabricant o les barreja l'impressor seguint les indicacions d'un catàleg estandarditzat de tintes. Hi ha uns quants fabricants internacionals de catàlegs de tintes. El més usat és Pantone. Altres catàlegs són Letraset, Toyo, DIC o Trumatch.

#### **Vegeu també**

Àmplia informació sobre la barreja partitiva en què es basa el sistema de quadricromia en els subapartats "Barreges: additiva, substractiva i partitiva" i "Codificació digital del color", de l'apartat "Color i disseny gràfic" del mòdul "Bases de disseny gràfic".

Els programes orientats a impressió solen ser programes vectorials que admeten gràfics de mapa de bits incrustats i que tenen diverses eines per a la maquetació i, en alguns casos, per a l'edició. Són programes com l'Illustrator, Corel Draw, Xara Xtreme, In Design, QuarkXPress, Scribus o alguns de discontinus en el desenvolupament però que en el seu moment van ser rellevants, com el Freehand o PageMaker. Aquests programes tenen, a més, controls extra d'impressió com la possibilitat d'ordenació de pàgines, el control de gràfics vinculats i fonts tipogràfiques usades, la inserció de marques de tall i de registre, el control de tintes i separació de colors, etc.

També hi ha programes de mapa de bits orientats a crear imatges que s'hauran d'inserir en un dels programes esmentats o bé imprimir-se directament. Aquests programes hauran de permetre el mode de treball en CMYK o tintes planes, controlar la densitat de trama i desar amb un format orientat a impressió com el TIFF. El Photoshop o el GIMP són programes que poden complir aquests objectius. El Fireworks és un programa que sobre una base vectorial permet treballar en mapa de bits i ser utilitzat per a una finalitat similar a la que hem descrit.

### **5.2.2. Programari orientat al Web**

En la creació de gràfics per al Web o, en general, per a ser visualitzats en pantalla, les necessitats són molt diferents de les descrites per a imatges la destinació de les quals és un producte imprès.

Les unitats de mesura són generalment els píxels. Ja no té sentit treballar en termes de pàgina i, en un sistema de mesura física, la mida de l'àrea de treball està relacionada amb la mida final de la imatge en píxels en el dispositiu de sortida.

Les característiques de l'entorn de treball també són diferents si generem gràfics vectorials o gràfics de mapa de bits. En la generació de gràfics de mapa de bits per al web és molt important optimitzar-los. Els programes de gràfics introdueixen controls d'optimització que permeten controlar la qualitat i el nombre de colors, establir transparències, etc.; també poden simular, en temps real, l'efecte d'aquesta optimització en la imatge.



Els programes orientats a web també poden tenir opcions per a crear un GIF animat i per a determinar si la imatge s'ha de carregar en entrellaçat o no. Exemples de programes que fan aquestes funcions són el Fireworks o el mòdul del Photoshop anomenat Image Ready.

Un altre element important d'aquests programes és el tractament que fan del color. Es treballa en mode RGB però es permet seleccionar el color mitjançant el codi hexadecimal que usa HTML. A més, s'inclou la possibilitat de treballar amb la paleta web, una paleta de 216 colors consensuada per tot el sector, i en el selector de color es pot donar la possibilitat de mostrar només els colors d'aquesta paleta.

Si s'ha de crear una imatge vectorial per al web (en format SWF o SVG) les necessitats també són diferents. Els programes estan proveïts de totes les eines de creació vectorial, però a més permeten suavitzats que milloren la visualització en pantalla.

Els formats vectorials per a Internet inclouen possibilitats d'animació i programació. Per això, programes com el Flash, Silverlight o Gnash tenen un entorn de treball amb línia de temps, capes i fotogrames pensat per a l'animació, i també sistemes de programació per a afegir interactivitat. També els programes desenvolupats inicialment per a impressió, com l'Illustrator, han afegit la possibilitat de generar gràfics per al web i exportar els formats esmentats. També hi ha el cas ja esmentat del Fireworks, que permet crear gràfics per a web, tant vectorials com de mapa de bits, amb un entorn de treball molt orientat a facilitar l'optimització d'aquests últims.

### 5.3. Programari propietari i *open-source*

#### 5.3.1. El programari

Els ordinadors només processen **codi binari**, instruccions en zeros i uns. Però els programadors no escriuen directament en aquest llenguatge de màquina. Escriuen en **llenguatges d'alt nivell**, llenguatges lògics, amb regles sintàctiques rígides i algunes semblances amb el llenguatge escrit (llenguatges com el C, Pascal, Java, etc.). Aquests llenguatges es tradueixen a **llenguatge binari de baix nivell** perquè l'ordinador l'executi. El que el programador escriu s'anomena **codi font**, el que l'ordinador processa **codi binari**.

El **programari** és un conjunt d'instruccions programades que permeten fer alguna cosa. El programari només es pot distribuir amb el codi binari per a ser utilitzat o amb el codi font inclòs per a saber com està programat o modificar-lo.

#### Vegeu també

Sobre l'optimització de gràfics per al web, vegeu la secció dedicada a aquest tema del mòdul "Producció de gràfics".

### 5.3.2. L'aparició del programari de propietat

En les primeres dècades de la informàtica el programari es desenvolupava a les universitats i centres de recerca. El codi font estava disponible per a qualsevol desenvolupador que volgués millorar-lo.

Però al començament dels vuitanta apareix l'ordinador personal, que s'estén per oficines i llars. Els programes es comencen a vendre com a producte independent de l'ordinador i sense el codi font. Creix una indústria del programari que amaga el codi font per evitar ser copiat per la competència. El programari es converteix en propietari i es protegeix mitjançant la legislació de *copyright* i les patents.

En el camp del disseny i les arts gràfiques, al costat de l'aparició del Macintosh el 1984, empreses com l'Aldus i l'Adobe s'introdueixen en els estudis de disseny amb programes com el Page Maker, Freehand, Illustrator i Photoshop. El sistema operatiu del Macintosh és propietat de l'empresa Apple. Una altra empresa, Microsoft Corp., propietària del sistema MS-DOS està en la majoria de PC compatibles, introdueix un sistema semblant, el Windows, encara dominant, i estableix les bases d'un monopoli. També seran actors rellevants en el camp del programari d'edició gràfica Quark Inc., desenvolupadora de QuarkX-Press, i Macromedia, desenvolupadora de Flash, Dreamweaver i Fireworks entre d'altres. El 1994 l'Adobe adquireix la seva competidora Aldus, i el 2005 farà el mateix amb Macromedia.

Davant d'aquesta situació l'usuari es veu compromès i obligat a no distribuir el programari comprat sota fèrries llicències. A més depèn de l'empresa propietària que es corregeixin errors o s'implementin noves característiques, ja que ell no disposa del codi font.

### 5.3.3. El projecte GNU

A mitjan els vuitanta un grup de programadors insatisfets amb aquesta situació de monopolis de programari propietari comencen un projecte per a "alliberar", en les seves paraules, el programari. Richard M. Stallman, que havia treballat en el MIT en el desenvolupament del sistema operatiu UNIX i eines per a aquest, se sent enganyat per la seva privatització i inicia el projecte GNU, acrònim recursiu que significa *GNU's not UNIX* (GNU no és UNIX).

Volen crear un nou sistema operatiu equivalent a l'UNIX però lliure. La tasca és àrdua perquè les eines que necessiten (programes de text, compiladors, etc.) també són propietàries. Per tant, comencen per allà, per crear aquestes eines. També creen la Free Software Foundation (FSF) per vetllar pel procés.

### 5.3.4. La llicència GPL

Un dels millors encerts del projecte GNU és haver sabut posar de part seva les lleis de *copyright* creant la General Public Licence (GPL). Aquesta llicència es basa en els mateixos drets d'autor i de reproducció per a definir els termes sota els quals es distribuirà un programa.

Un programari amb llicència GPL s'ha de distribuir juntament amb el codi font, a més qualsevol persona pot distribuir-lo lliurement i modificar-lo si vol. La GPL estableix que qualsevol modificació del programari haurà d'estar obligatòriament sota aquesta llicència, per la qual cosa s'ha dit que la GPL contamina tot el que toca. Qualsevol desenvolupament basat en programari lliure continuarà essent lliure per no infringir la llicència amb què es distribueix.

### 5.3.5. El moviment del programari lliure

Actualment el moviment de programadors que desenvolupen programari lliure de codi obert (*open-source*) és un sistema autoorganitzat eficaç, que disposa d'un bon sistema operatiu (Linux) que funciona en la majoria de servidors d'Internet, i a més es va incorporant cada vegada més a llars i oficines, té múltiples eines de programari, i també un sistema eficient de desenvolupament i correcció de fallades.

Algunes empreses importants (com IBM, Sun o Novell, per esmentar-ne algunes) contribueixen a aquest procés. I un gran nombre de petites empreses, programadors independents i equips informals també ho fan. Companyies com Google també participen en el finançament de projectes (amb el *Google Summer of Code* des de 2005).

En el camp del programari de gràfics és especialment rellevant per a l'impuls dels projectes i l'intercanvi de solucions el Libre Graphics Meeting, una convenció que des de 2006 reuneix anualment els desenvolupadors de programes de gràfics lliures. També tenen un paper rellevant la companyia Xara, desenvolupadora del programari vectorial Xara Xtreme, i la Blender Foundation, que coordina el desenvolupament del programari d'animació i gràfics 3D Blender; a més dels equips de desenvolupament entorn de programes com Gimp, Inkscape o Scribus.

### 5.3.6. Programari lliure per a gràfics

En aquest moment, a més del Linux com sistema operatiu, hi ha bones eines de programari lliure per a servidors, per a programació, per a edició de textos i programari d'oficina, navegadors d'Internet, etc. Què passa amb el programari d'edició de gràfics?

Hi ha una bona eina de gràfics de mapa de bits: **The GIMP** (*The GNU Image Manipulation Program*). Es tracta d'un programa amb funcionalitats bàsiques equivalents al Photoshop disponible per a les plataformes Linux, Windows i Macintosh. Quant a gràfics vectorials, hi ha diversos programes entre els quals es pot destacar **Inkscape** i **Xara Xtreme** (orientats al dibuix), i **Scribus** (orientat a la maquetació editorial).



Els defensors del programari propietari argumenten que un desenvolupament d'eines com el que s'ha produït en el món del disseny en les últimes dècades solament és possible amb una indústria forta del programari, finançada per la venda de llicències de programes, que respongui a les necessitats del sector. Però també s'argumenta que moltes vegades aquesta indústria no inverteix els seus esforços a corregir els errors, sinó a afegir noves funcionalitats innecessàries que justifiquin l'aparició d'una nova versió del programa que els usuaris hauran d'actualitzar mitjançant pagament.

En la situació en què va quedar el mercat després de l'adquisició de Macromedia per part d'Adobe el 2005, aquesta companyia es va convertir en l'actor principal en el desenvolupament de programari propietari en el camp del disseny, amb programes com Photoshop, Fireworks, Flash, Illustrator, In Design i Dreamweaver, entre altres. Adobe manté la venda de llicències dels seus programes principals mentre, especialment a partir de 2008 amb l'anunci del seu *Open Screen Project*, comença una política d'obertura de les especificacions dels seus formats com l'SWF, FLV o PDF.

Companyies d'Internet com Google s'estan introduint en el desenvolupament d'eines de gràfics discretament, començant per eines orientades a l'usuari domèstic com **Picasa**. És un programa propietari que s'ofereix gratuïtament per a organitzar i editar fotografies.

El desenllaç i evolució d'aquesta situació depèn de les actuacions dels diversos agents implicats i, en bona mesura, de l'actitud que prenguin els usuaris.