DigitAll

Formació en
Competències
Digitals

3

Creació de continguts digitals

DigitAll

Formació en
Competències
Digitals

Creació de continguts digitals

Nivell A1

ÍNDEX

DigitAll

Creació de
continguts digitals

3.1
DESENVOLUPAMENT DE CONTINGUTS

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

Ús de formats digitals

Ús de formats digitals

La creació de continguts digitals és fonamental per conèixer un producte digital. A l'hora de crear un contingut i de publicar-lo a alguna plataforma digital (xarxes socials, webs, etc.), el receptor pot trobar-lo quan se cerqui informació relacionada. Per garantir la visibilitat dels nostres continguts, independentment dels motors de recerca, haurem d'assegurar-nos de complir amb uns estàndards de qualitat i seleccionar adequadament entre els formats d'aquests. A més, la difusió dels continguts serà fonamental.

La classificació més general dels formats per a la creació de continguts digitals inclou: **text, imatge, àudio** i **vídeo**.

El text ha estat un dels formats estrella de la comunicació humana, atès que pocs invents han impactat en el desenvolupament de la nostra cultura com ho va fer la imprenta. No obstant això, encara que el text sembla sempre el mateix, vam poder trobar textos publicats en diferents mitjans digitals, com els blocs o pàgines col·laboratives com a Viquipèdia, així com diferents tipus de formats digitals amb què podem treballar. Hem de tenir en compte que la compatibilitat entre els formats no és total, per la qual cosa podem perdre informació per transformar un tipus d'arxiu en un altre. Entre els formats més comuns hi ha:

Format	TXT	RTF	ODT	DOC(X)	PDF
Text enriquit	No	Sí	Sí	Sí	Sí
Pàgines	No	Sí	Sí	Sí	Sí
Inclou imatges, taules	No	Sí	Sí	Sí	Sí
S'obre habitualment amb	Qualsevol processador de text	Qualsevol processador de text	Open Office	Microsoft Word	Adobe Acrobat Reader
Situació d'ús recomanable	Només text no enriquit. Fàcil d'obrir en qualsevol situació	Format simple, però admet text enriquit. Treball en diverses plataformes	Format lliure i obert. Editable des de qualsevol mitjà o plataforma	Més comú. DOCX ocupa menys que DOC	No es permet edició. Estàndard per distribuir documentació

En una **imatge** trobarem fotografies o imatges com a continguts més utilitzats. Aquestes ajuden a fer un contingut digital molt més interessant i suggerent, i milloren l'experiència del receptor. És un dels recursos més utilitzats a les xarxes socials i a les pàgines web, ja que ofereixen un suport al text en publicacions i és un reclam d'atenció. Podem trobar fotografies, il·lustracions, infografies, mems i bàners a qualsevol racó de la xarxa. Els formats més destacats són:

Format	JPG	GIF	PNG	TIFF	RAW
Qualitat	Alta	Mitjana	Alta	Molt alta	Molt alta
Grandària	Alt	Mitjana	Mitjana	Molt alt	Molt alt
Compressió permesa	Molt alta	Mitjana	Alta	Alta	Alta
Pèrdues en comprimir	Sí	No	No	No	No
Nombre de colors	24 bits color, 8 bits blanc i negre	Fins a 256 colores	24 bits de color	1 a 64 bits	48 bits
Admet fons transparent	No	Sí	Sí	Sí	No
Permet animacions	No	Sí	No	No	No
Situació d'ús recomanable	Càmeres digitals, imatges, impressió, intercanvi d'imatges	Internet, imatges de mida reduïda, logos	Internet, gràfics, iconografia, programari	Imatges d'alta qualitat, càmeres digitals, escàners, impressió	Càmeres digitals

L'**àudio** ofereix un tipus de contingut que es pot utilitzar en el moment que el receptor consideri més oportú, cosa que aporta un valor afegit a l'hora de comunicar. Entre les diferents opcions d'àudio tenim el pòdcast, les entrevistes i els audiollibres. Els formats més utilitzats són:

Format	FLAC	MP3	WAV	OGG	WMA
Qualitat	Alta	Mitjana-baixa	Alta	Mitjana-baixa	Alta-mitjana
Grandària	Alta	Mitjana	Alta	Baixa	Mitjana
Compressió permesa	No	Sí	No	Sí	Sí
Admet metadades	Sí	Sí	Sí	Sí	Sí
Situació d'ús recomanable	Arxius musicals d'alta qualitat	Àudios per a webs o dispositius portàtils	Àudios per a webs o dispositius portàtils	Àudios per a webs o dispositius portàtils	Àudios per a webs o dispositius portàtils

Per acabar, el **vídeo** resulta un format que unifica la imatge i l'àudio, ja que constitueix un dels continguts que funciona millor en xarxes socials i pàgines web, i ajuda a il·lustrar un projecte o a mostrar-ne el costat més humà. A la xarxa podem trobar revisions de productes, tutorials, transmissions en directe, espots o vídeos musicals. Els formats principals de vídeo es recullen a la taula següent:

Format	AVI	MKV	MP4	OGG	MPEG
Qualitat	Variable	Alta	Mitjana	Alta	Mitjana
Grandària	Variable	Alta	Mitjana	Baixa	Mitjana
Compressió permesa	Sí	Poca	Sí	Sí	Sí
Situació d'ús recomanable	Emmagatzematge de vídeos capturats amb càmera	Emmagatzematge de vídeos d'alta qualitat	Vídeos per a webs o dispositius portàtils	Ideal per a vídeos a internet per la seva qualitat-pes	Format estàndard per a compressió i ús a Internet

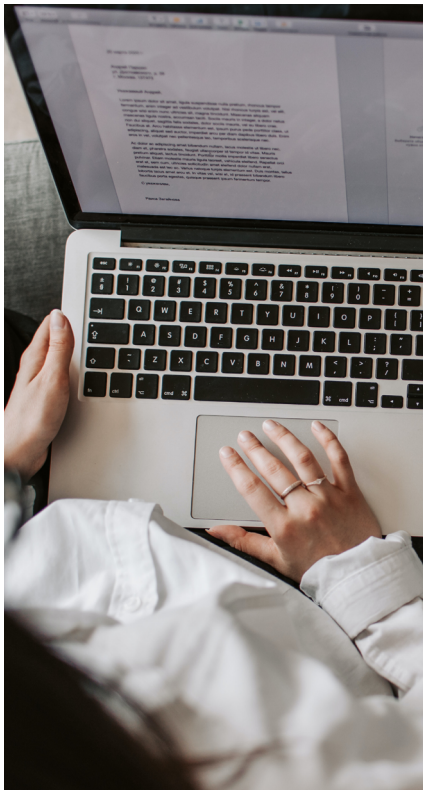
Els continguts digitals representen eines molt poderoses a l'hora de transmetre informació, amb finalitats docents, comercials o de qualsevol mena. L'elecció de formats adequats serà fonamental per aconseguir impulsar el nostre projecte i assolir així la major quantitat de receptors possible.

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

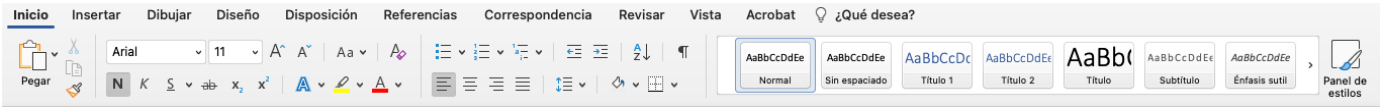
Processament de text i aplicació de plantilles

Processament de text i aplicació de plantilles



En aquest document anirem descrivint quines són les funcionalitats bàsiques que ens ofereix un processador de textos. Ressaltarem l'ús de plantilles de documents proporcionades pel processador o la possibilitat de crear plantilles pròpies que s'hi emmagatzemin.

Les eines que descriurem en aquest document es troben en programari de fàcil accés, com ara Word, de Microsoft, encara que les podríem trobar en qualsevol altre processador de text. Un dels avantatges més grans és la capacitat de canviar el format del text i document en qualsevol moment. La barra conté botons i llistes que es despleguen per a totes les característiques que es canvien més sovint sobre l'aspecte del text. A la següent imatge veiem la barra d'eines de Word:



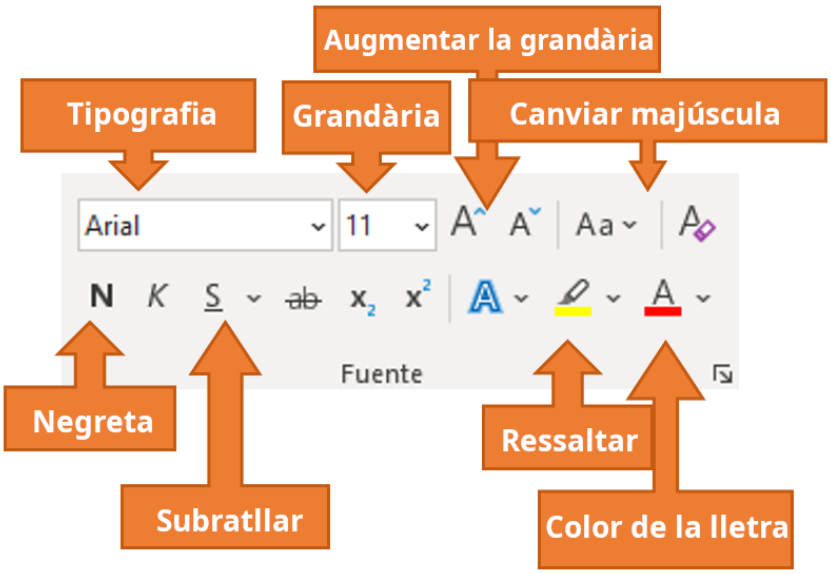
En el quadre següent veiem resumides les funcions que posteriorment anirem descrivint:

Bloc	Conté	Permet
Porta-retalls	Retallar, copiar, enganxar, enganxar des de	Mostrar tots els elements emmagatzemats
Font	Seleccionar tipografia, mida de lletra, negreta, cursiva, subratllat, ratllat, subíndex, superíndex, ombra, subratllat en colors, canviar entre majúscules o minúscules	Personalitzar amb opcions avançades de fonts i caràcters. Afegir efectes visuals, estils i colors
Paràgraf	Llista amb vinyeta, numeració, llista multinivell, sagnia, alinear el text a l'esquerra, al centre, a la dreta o justificar, interlineat, ombreig del paràgraf, afegir vores, ordenar una selecció alfabèticament o numèricament, mostrar les marques del paràgraf per edició avançada	Ajustar amb precisió el disseny, espaiat, sagnia i altres
Estils	Aplicar estils predeterminats, crear-los o eliminar-los	Administrar i personalitzar l'aparença d'un text

I ara anirem veient amb una mica més de detall per a què es fa servir cadascuna d'aquestes eines seguint l'ordre de la imatge d'esquerra a dreta.

- **Porta-retalls:** emmagatzema text i gràfics que copia o talla des de qualsevol lloc i permet enganxar els elements emmagatzemats a qualsevol altre fitxer Office. Primer cal seleccionar amb el ratolí allò que es vulgui copiar.

- **Font:** podem modificar l'aparença de les fonts usant les eines d'aquesta graella i fent-hi clic. Primer haurem de seleccionar el text que es vol modificar i després, marcant les eines, modificarem la tipografia, mida, gruix o color.



- **Paràgraf:** igual que es pot canviar el format de les fonts, es pot modificar l'aspecte dels paràgrafs. En aquest cas es pot: sagnar text, ajustar els marges, crear taules, afegir vinyetes i números a llistes.

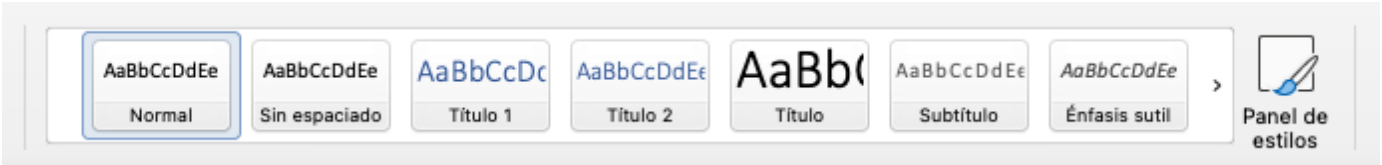
  Per afegir una sagnia a la primera línia es podria utilitzar la tecla Tabulador del teclat. No obstant això, si el paràgraf consta de diverses línies, és pertinent fer servir l'eina que defineix els sagnats, augmentant o disminuint la sagnia.

Quan apareixen llistes que es volen numerar o posar una vinyeta, es poden fer servir aquestes eines. Apareixerà un número, o una vinyeta, davant de cadascun dels paràgrafs seleccionats, amb els sagnats corresponents per separar el text dels números. Quan el paràgraf té una extensió superior a una línia, les línies a partir de la segona s'alineen amb la primera que conté el número.

Amb la funció de l'interlineat es pot afegir espai vertical entre línies o paràgrafs. També s'hi poden afegir espais abans i després dels paràgrafs.

Amb aquests botons es fixa l'alineació en diferents posicions pel que fa als marges de la pàgina.

- **Estils:** teniu una col·lecció d'estils i altres característiques per al document. Cada document té una plantilla amb estils per defecte a la llista desplegada.



Es pot modificar un estil actualitzant-lo perquè coincideixi amb el format del document. Al grup **Estils** de la pestanya **Inici**, feu clic amb el botó dret a l'estil que voleu canviar i marqueu Actualitza [nom d'estil] perquè coincideixi amb la selecció.

- **Ús de plantilles predefinides:** en iniciar un document al Word es pot donar format amb la barra d'eines al text o bé utilitzar plantilles predeterminades. Aquestes plantilles contenen informació i elements de disseny predefinits i podrem utilitzar directament qualsevol plantilla que es descarregui o se'n pot personalitzar una al gust i guardar-la com a plantilla per al futur. En qualsevol cas, aquestes plantilles tindran una estètica agradable predefinida, per la qual cosa nosaltres només ens hauríem d'ocupar de treballar el contingut del nostre document. A més de documents, tenim plantilles per a calendaris, targetes de presentació i molt més.

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

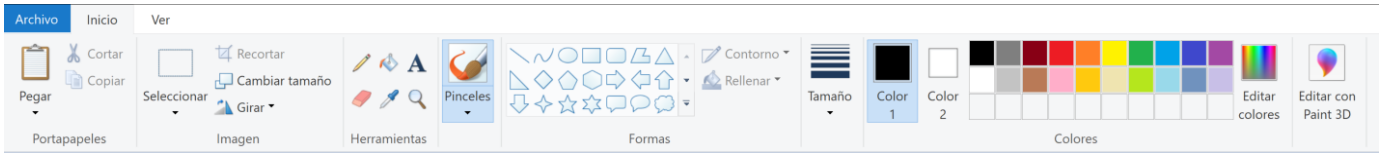
Ús d'eines bàsiques de disseny d'imatges

Ús d'eines bàsiques de disseny d'imatges

En aquest document anirem descrivint quines són les eines més bàsiques que ens trobarem quan emprem programes d'edició d'imatges. Aquest tipus d'eines són senzilles d'usar i es poden fer servir en multitud de situacions quotidianes. No obstant això, moltes vegades no les fem servir perquè bé ignoram les seves funcions o bé no sabem ni tan sols que existeixen.

Les eines que descriurem en aquest document es troben en programari de fàcil accés, per exemple, **Paint** si estem usant un PC o bé l'eina d'edició d'imatges a què podem accedir des de la galeria del nostre telèfon intel·ligent. Qualsevol d'aquestes dues opcions resultarà molt intuïtiva d'usar i molt accessible per tenir un primer contacte al camp de l'edició/disseny d'imatges.

A la següent imatge veiem la barra d'eines de Paint. Mitjançant l'ús d'aquestes eines podrem anar dissenyant una imatge al nostre **llenç**, que és l'espai en blanc sobre el qual podem treballar:



Al següent quadre veiem resumides les eines que posteriorment anirem descrivint:

Bloc	Conté	Permet
Porta-retalls	Retallar, copiar, enganxar, enganxar des de	Emmagatzemar informació
Selecció d'imatge	Seleccionar-ho tot, selecció rectangular, selecció lliure, invertir selecció, selecció transparent, eliminar, retallar, canviar mida, girar	Seleccionar i treballar amb una imatge
Edició d'imatge	Llapis, pinzell, farcit amb color, text, esborrany, comptagotes, lupa	Introduir dibuixos a mà alçada, text i pintar
Formes	Base de dades de formes, contorn, interior	Introduir formes predefinides
Mides	Base de dades de mides	Seleccionar la mida del llapis i el pinzell
Colors	Base de dades de colors, color 1, color 2	Seleccionar el color de treball

Tingues en compte que aquestes són les eines més bàsiques, treballarem amb eines més avançades en un altre programari a nivells posteriors. I ara anirem veient amb una mica més de detall per què valen cadascuna d'aquestes eines seguint l'ordre de la imatge d'esquerra a dreta:

- **Porta-retalls:** com el nom indica, és un magatzem de text i imatge que ens permetrà retallar o copiar una imatge o part d'una imatge i emmagatzemar-la en aquest espai. Després podrem utilitzar aquesta informació enganxant-la on vulguem. També podrem fer servir l'opció Enganxar des de per triar un arxiu d'imatge del nostre equip i enganxar-lo al llenç.
- **Tria d'imatge:** són el grup d'eines que ens permetran anar seleccionant parts de la nostra imatge. Al punt anterior parlem d'"una part d'una imatge", ara utilitzarem aquestes eines de selecció per indicar amb quina part de la imatge volem treballar. Així podrem seleccionar-ho tot o bé només una part usant l'eina de selecció rectangular (lògicament, té forma de rectangle) o la de selecció lliure, que ens permetrà seleccionar una àrea de qualsevol forma. Després podrem invertir la selecció per seleccionar tot allò que no hem marcat amb les eines anteriors, eliminar la part del llenç que hem seleccionat, o fer-ne una selecció transparent, és a dir, indicarem que el fons de la nostra

Saber-ne més

Copiar i retallar són dues accions molt semblants, però en tallar una imatge o part d'una imatge eliminam del nostre llenç, mentre que en copiar conservam la informació original.

selecció és transparent. La resta de les eines d'aquest apartat ens proporcionaran també recursos útils com retallar, per fer que la nostra selecció es converteixi en el total de la imatge, canviar-ne la mida, per proporcionar noves dimensions a la nostra imatge, o girar, per inclinar o voltejar la imatge.

- **Eines per editar la imatge:** són el grup d'eines que ens permetran editar una imatge a voluntat. Entre aquestes hi ha eines que ens permetran dibuixar a mà alçada, com són el llapis o el pinzell. L'avantatge del pinzell és que ens permetrà seleccionar diversos formats de sortida canviant la forma i la mida del pinzell per adequar-la al traç que volem fer. Podrem pintar més ràpidament fent servir el farcit amb color que, perquè ens fem una idea, la icona és una galleda de pintura i ens permetrà emplenar el nostre dibuix d'un color amb un sol clic del ratolí. També podrem afegir text a la nostra imatge triant entre una àmplia gamma de fonts, mides i formats. Si ens hem equivocat, podrem corregir usant l'eina Esborrany. I, finalment, tenim dues eines més que encara que es trobin en aquest bloc no són d'edició exactament: es tracta de l'eina comptagotes, que ens permetrà seleccionar un color de la nostra imatge (per poder, per exemple, seguir pintant amb el mateix color) i la lupa, que ens permetrà anar augmentant o allunyant la imatge segons les nostres necessitats.
- **Formes:** és un paquet d'eines molt útil perquè ens permetrà recórrer a un munt de formes predissenyades que no haurem de dibuixar a mà alçada i només les haurem de seleccionar per afegir-les a la nostra imatge. Així tenim des de formes geomètriques senzilles fins a dibuixos de polígons lliures. A més, podrem triar el contorn de la nostra forma i l'interior entre una sèrie d'opcions predefinides.
- **Selecció de mides:** aquesta eina ens permetrà definir com de gruixat o d'estret serà el traç que farem amb el nostre llapis o el nostre pinzell.
- **Paleta de colors:** finalment arribarem a la zona de selecció del color de treball. Aquí haurem d'anar seleccionant amb

⚠ ATENCIÓ

Recorda que aquestes formes que introdueixis al dibuix podràs pintar-les després molt fàcilment amb l'eina farcit amb color.

quin color volem, per exemple, pintar amb el pinzell. Cal tenir en compte que sempre podrem tornar a la paleta de colors a triar un nou color per a alguna cosa que ja haguem pintat abans i tornar-ho a pintar de nou (tantes vegades com vulguem). Aquí tenim diverses opcions: el més obvi és que podem seleccionar qualsevol color de la paleta de colors o bé anar a l'apartat d'Editar colors per tenir-ne una selecció molt més àmplia per triar. A més, si hi ha un parell de colors que farem servir molt, podem seleccionar-los com a Color 1 i Color 2, de manera que la nostra selecció es quedarà guardat i no haurem d'estar contínuament buscant-los a la paleta.

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

Disseny estructurat d'imatges basat en capes i modificacions a través de màscares

Disseny estructurat d'imatges basat en capes i modificacions a través de màscares

En parlar de **màscares de capa** hem de tenir en compte que parlem de recursos que ens permetran editar una imatge de manera no destructiva, és a dir, no alterarem la informació original de la nostra imatge, sempre podríem després eliminar una màscara de capa i recuperar la nostra imatge original. Això mateix és aplicable, com comentarem més endavant per a les màscares vectorials.

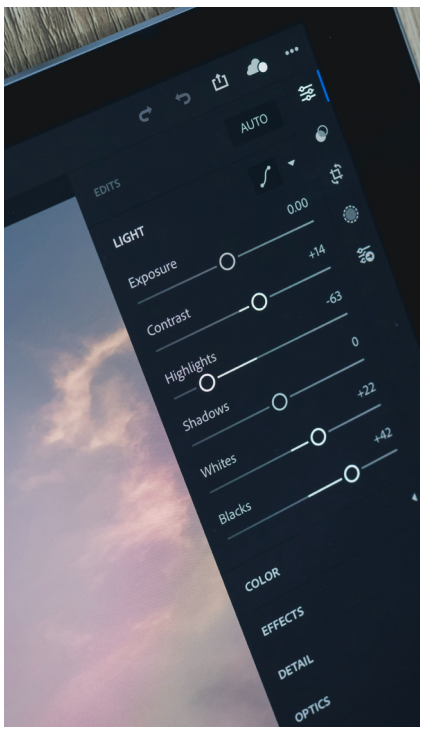
Aquest tipus d'edicions es poden dur a terme a qualsevol programa d'edició d'imatges i consisteixen a afegir capes addicionals sobre la nostra imatge que, en la versió més senzilla, ens permetran **mostrar** o **amagar** parts d'una imatge (per exemple, un fons). Per altra banda, les màscares de capa es poden aplicar sobre la totalitat d'una imatge o sobre una part molt concreta.

D'aquesta manera, indicarem quines parts de la nostra imatge volem mostrar o amagar pintant sobre la màscara de capa, tenint en compte que les màscares de capa treballen en **escala de grisos**, és a dir, només admeten colors entre el blanc i el negre. Així, podrem pintar la màscara de blanc, i aleshores la màscara ens mostrarà la totalitat de la imatge (100%) o de negre, i la màscara ens ocultarà tota la imatge (0%). Però també ens podem quedar en un punt intermedi (gris) en què la màscara ens mostrarà la imatge semitransparent, és a dir, ens amagarà una certa part de la informació en funció del to de gris que hàgim triat. Dit amb altres paraules, la màscara de capa ens permetrà regular la seva **opacitat**.

Normalment, els programes d'edició ens permetran regular les propietats d'una màscara i controlar així la seva opacitat (de vegades també s'anomena densitat) mitjançant eines de tipus lliscant passant d'alts nivells d'opacitat/densitat (100% de la capa en negre, tot ocult) a nivells baixos d'opacitat/densitat (0% de la capa en negre, es mostra tot).

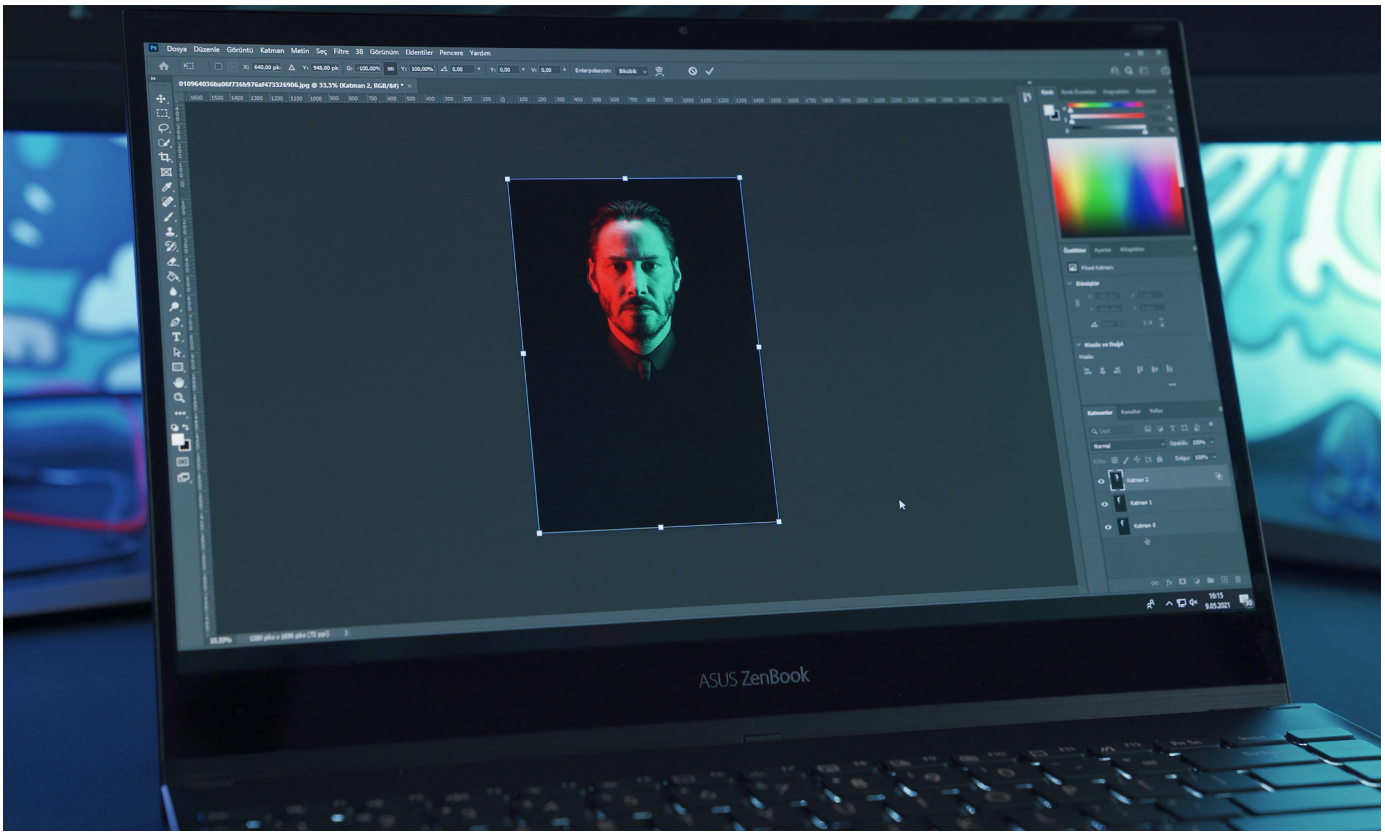
Saber-ne més

Encara que segurament *Photoshop* sigui el programari d'edició d'imatges més conegut, hi ha alternatives al camp del programari lliure, com ara ***GIMP*** (*gimp.org.es*), que permeten treballar pràcticament amb les mateixes eines d'edició que els programes de pagament professionals.



Això no obstant, aquestes eines tenen importants diferències amb altres que ens proporcionen el mateix resultat. Per exemple, si fem servir l'eina Esborrany sobre una imatge, obtindrem el mateix efecte que si afegim una màscara de capa i ocultam (pintant de negre) la part que volem esborrar, amb la diferència que, després d'aplicar l'Esborrany**, la informació de la imatge s'elimina i la perdem per a sempre** mentre que en el cas de la màscara de capa podem simplement descartar la capa i recuperar la informació que havíem amagat a la vista.

Finalment, les màscares de capa són eines d'edició molt potents que, a més de permetre'ns mostrar o amagar parts d'una imatge, també ens permetran fer altres operacions d'edició com l'aplicació d'efectes concrets sobre part de la nostra imatge, crear fotomuntatges o composicions o simplement la realització d'ajustos de color sobre tota la nostra imatge o part (per exemple, el cel) i, com hem comentat anteriorment, aquests canvis es produeixen sense alterar la imatge original, per la qual cosa podem descartar-los i recuperem la nostra imatge sense editar.



Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

Imatges vectorials vs. imatges rasteritzades

Imatges vectorials vs. imatges rasteritzades

Les **imatges rasteritzades** són aquelles que es componen de petites peces que formen un mosaic. Aquestes peces s'anomenen **píxels**. A més quantitat de píxels per unitat d'àrea, més resolució tindrà la imatge, i n'augmentarà la **definició**. També es coneixen com a **mapes de bits**.

Si una imatge té una resolució de 800x1000 píxels, vol dir que conté 800 píxels horitzontals i 1000 verticals. Si observam aquesta imatge en una pantalla o una impressió, el més probable és que l'ull humà no detecti els píxels. No obstant això, si n'augmentam la mida o la imprimim ampliada, el mosaic es farà evident. Tot i això, a diferència de les imatges rasteritzades, els **gràfics vectorials** no es componen per píxels, sinó per **punts de control** que determinen angles i línies que formen la figura. Podem observar les diferències entre tots dos tipus d'imatges a la figura 1.

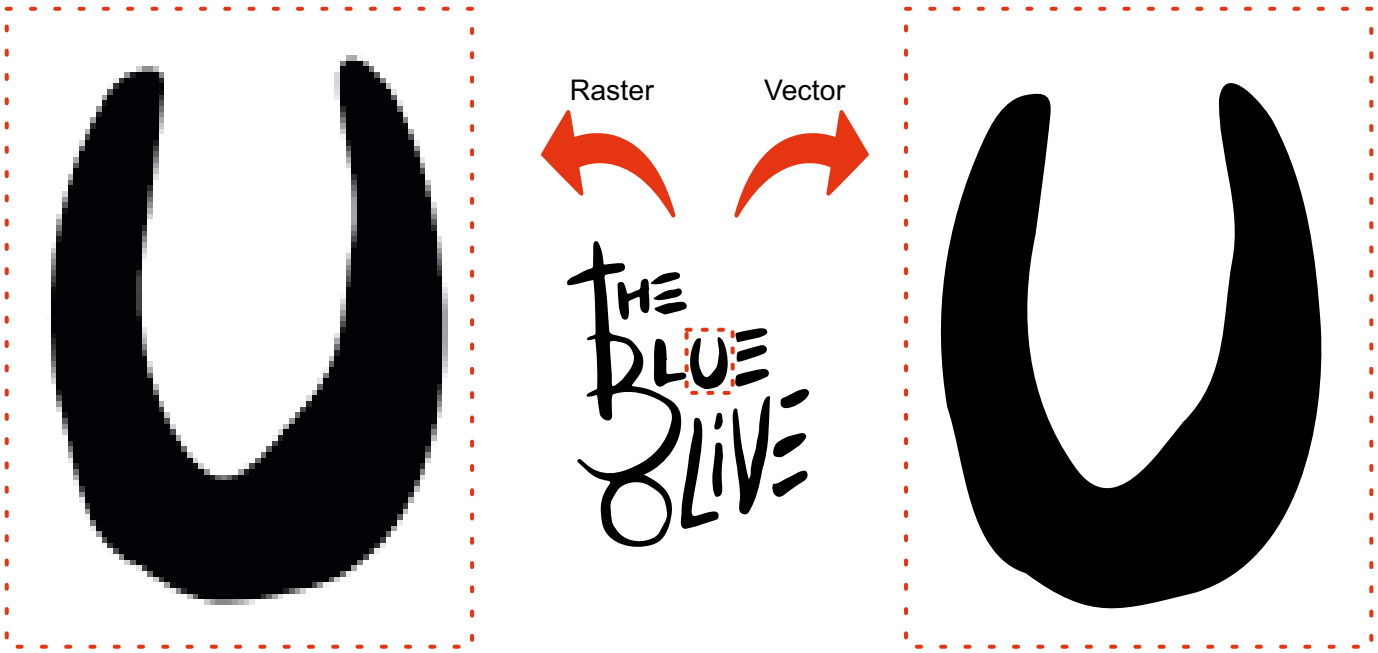


Figura 1. Diferències entre ràster i vector.

Les imatges rasteritzades són precises (qualitat fotogràfica) i ofereixen moltes possibilitats en contextos de creació gràfica i artística, però solen ocupar molt d'espai i no és recomanable modificar-ne la mida. Sobretot augmentar-lo, ja que suposa una pèrdua de qualitat notable. Les eines de programari més conegudes per al seu maneig són *Adobe Photoshop*, *Gimp* i *Corel Photo-Paint* i els seus formats *png*, *jpg*, *gif* o *tif*. D'altra banda, les imatges vectorials no perden qualitat si són ampliades o reduïdes, perquè escalen automàticament la mida. Una mateixa imatge vectorial ens pot servir per imprimir una targeta de visita o un cartell publicitari enorme i la seva qualitat serà òptima. A més, els seus fitxers no solen pesar molt. Per contra, a l'hora de fer-ne dissenys, trobarem algunes limitacions respecte dels ràsters, i el seu aspecte no s'acostarà a la qualitat fotogràfica. Els programes més utilitzats en vectors són *Adobe Ilustrator*, *Inkscape* i *CorelDraw*, i els seus formats *eps*, *ai*, *pdf*, *svg* o *sketch*.

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

Compressió d'imatges

Compressió d'imatges

Algunes vegades les imatges digitals de grans dimensions dificulten l'ús de correus electrònics o llocs web, ja que requereixen molt de temps per ser carregades i l'usuari ha d'esperar molt de temps per obtenir la imatge/informació. Això és a causa que les imatges utilitzades contenen moltes dades o, dit altrament, són de grans dimensions, per la qual cosa ocupen molt d'espai d'emmagatzematge i són difícils de transferir o processar.

La **compressió d'imatges** és l'aplicació de procediments per reduir la imatge original a uns bytes. L'objectiu d'aquest procés és **reduir** la redundància de la imatge i emmagatzemar i transferir les dades de manera eficient. És a dir, comprimir una imatge és reduir les dades redundants i irrellevants de la imatge, amb la **pèrdua més petita**, per permetre'n l'emmagatzematge o la transmissió de manera eficient. Les dades que estan duplicades són prescindibles i es guarden en un format reduït o són eliminades, i disminueixen la mida del fitxer on es desa la informació digital. El que se cerca amb aquest procés és reduir els bytes d'informació sense degradar la imatge o perseguint una qualitat d'imatge prou acceptable.

El procés de compressió pot ser "amb pèrdues" o "sense pèrdues". La c**ompressió sense pèrdues** redueix una imatge i la qualitat coincideix amb l'original, virtualment no té cap informació perduda. Tot i que sembla el mode de compressió ideal, moltes vegades no resol el problema de la mida i la imatge continua sent massa gran. Aquest tipus de reducció d'informació és generalment utilitzat en situacions en què la qualitat de la imatge és més important que l'espai d'emmagatzematge o la velocitat de càrrega d'una web.

A la **compressió amb pèrdues**, la imatge comprimida no té la mateixa qualitat, mida o informació que l'original. És a dir, la imatge es degrada. El pitjor d'això és que la compressió és irreversible i una vegada s'ha aplicat a una imatge no es pot tornar a l'estat original. Si una imatge es comprimeix repetides vegades, es va incrementant la distorsió successivament.

Hi ha diferents formats de compressió i cada un utilitza un algorisme diferent per produir les imatges comprimides. Els més comuns es descriuen a continuació:

- **BMP** (de l'anglès "*Bit Map Picture*"), format de compressió sense pèrdues de Microsoft. Podeu guardar 16,7 milions de colors i 256 tons de gris.

- **PNG** (de l'anglès "*Portable Network Graphics*"), és un format que redueix la grandària mitjançant identificació de patrons i comprimeix aquells que són comuns junts. La compressió PNG produeix arxius més grans que JPEG, però és usada majoritàriament quan la qualitat de la imatge és el més rellevant, com en webs amb logos, icones, imatges amb text, dibuixos i esquemes.

- **RAW** (de l'anglès traduït com a "cru"), format que conté totes les dades obtingudes pel sensor de la càmera fotogràfica, tal com ha estat presa la imatge. Per a una fotografia professional, on es vol obtenir molta informació (per tant, una mida d'imatge gran) és l'ideal. Es necessiten programes específics per al seu maneig. És un format de compressió de dades sense pèrdua d'informació. La dificultat més gran és que cada fabricant de càmeres utilitza una versió del format pel qual falta estandardització. Aquest fet en redueix la utilitat i augmenta les incompatibilitats.

- **TIFF** (de l'anglès "*Tagged Image File Format*"), és un format de compressió que es pot considerar del grup sense pèrdues. Es fa servir en casos d'imatges amb alta resolució i qualitat. No és adequat per a webs. S'usa a la indústria gràfica.

- **WEBP**, format de compressió de Google desenvolupat només per al seu ús a la web. És un format molt versàtil, ja que permet seleccionar entre compressió amb pèrdues o sense. Pretén ser un sistema de compressió que competeixi amb JPEG a la compressió d'imatges per a webs.

- **GIF** (del inglés "*Graphic Exchange Format*"), format de compressió que es pot considerar del grup de compressió amb pèrdues o sense pèrdues. Aquesta classificació ambigua depèn de la imatge que es vol reduir. GIF té un límit de 256 colors. Si l'original té 256 o menys colors, la compressió serà sense pèrdues. El format GIF es classifica

al grup de pèrdues si l'original tingués més de 256 colors. És un format per a vídeos i animacions simples.

- **JPEG** (de l'anglès "*Joint Group of Photographic Experts*"), és un format d'ús extensiu a webs i fotografia digital, produeix compressió amb pèrdues, però és àmpliament utilitzat en nombroses eines i aplicacions. Permet escollir el grau de compressió (i, per tant, el grau de pèrdua).

Format	BMP	PNG	RAW	TIFF	WEBP	GIF	JPEG
Tipus de compressió	Sense pèrdua	Sense pèrdua	Sense pèrdua	Sense pèrdua/ opcional amb pèrdua	Amb/sense pérdida	Amb/sense pérdida	Amb pérdida
Grandària	Molt gran	Gran	Molt gran	Molt gran	Molt gran	Molt petita	Petita
Colors	Molt bo	Molt bo	Molt bo	Molt bo	Molt bo	Suficient	Molt bo
Recomanar per a	Fotografies	Imatges amb fons transparent	Fotografies	Fotografies	Web	Gràfics, il·lustracions, animacions	Fotografies

Els consells per donar format a una imatge fotogràfica són:

1 | Si és possible, pren la imatge en format RAW, potser has de comprar targetes de dades per poder emmagatzemar més informació al dispositiu/càmera.

2 | Desa la feina o la imatge final a TIFF.

3 | Usa JPEG per compartir la imatge original amb altres.

4 | No emmagatzemis la imatge a JPEG.

5 | Desa la feina a WEBP per fer servir les imatges a pàgines web.

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

Creació de vídeo

Creació de vídeo

Tots vam fer de dibuixos petits en pàgines d'un quadern que en passar-les ràpidament el nostre dibuix s'animava i adquiria moviment. Això, que es diu filoscopi, representa el que és en realitat el vídeo, ja que podem entendre una seqüència de vídeo com una **successió d'imatges** que se superposen prou ràpid perquè el nostre cervell apreciï un moviment on no n'hi ha.

Digitalment, podrem obtenir vídeo per diverses vies: bé usant aquesta manera clàssica d'animació, és a dir, mitjançant imatges que un programa ens transformarà en vídeo (per exemple, ***Blender*** (***blender.org***) o bé gravant-les nosaltres mateixos.

Per realitzar els nostres enregistraments de vídeo cal que disposem d'un **equip d'enregistrament**. Aquests sistemes poden variar entre si enormement en funció del resultat final que cercam, però tots tenen en comú que no només són capaços de gravar vídeo, sinó que, a més, també ens registraran i integraran el so. És a dir, estaran compostos, almenys, d'una **càmera** i un **micròfon**.

Lògicament, aquests equips variaran enormement entre si i en depèn la qualitat del vídeo resultant. En el cas de la imatge, parlarem de la **resolució**. Així, ja són habituals els telèfons mòbils amb què podem gravar vídeo, però aquests sistemes estan limitats, per una banda, per les dimensions del mòbil i, per altra banda, per la capacitat d'emmagatzematge. És a dir, estan preparats per gravar vídeos de baixa qualitat (o baixa resolució), ja que, perquè els telèfons continuïn sent petits, no poden portar sistemes d'enregistrament potents, però, si els portessin, la qualitat del vídeo superaria la seva capacitat d'emmagatzematge.

D'altra banda, sistemes d'enregistrament més professionals, com les càmeres de vídeo o les càmeres rèflex, ens permetran gravar vídeo d'elevada qualitat/resolució, ja que disposen de millors sistemes per capturar vídeo i àudio, però a més poden presentar funcionalitats addicionals com, per exemple, estabilitzadors d'imatge per evitar tremolors o sistemes d'infraroig per gravar a la nit.

Saber-ne més

La pel·lícula de culte *Malson abans de Nadal* (1993) va ser creada mitjançant una tècnica coneguda com a stop-motion, que no és més que anar prenent imatges seqüencials d'objectes estàtics per aparentar moviment.

Després de gravar un vídeo, podem recórrer a diversos programes que ens permetran **editar** el resultat obtingut digitalment i obtenir així un vídeo més d'acord amb el que cercam. Com passava durant l'enregistrament, també trobarem una gran variabilitat en els programes d'edició de vídeo. En aquest punt, hem de tenir en compte que un dels factors limitants a l'edició de vídeo és la resolució a la qual s'ha gravat, per tant, si un vídeo s'ha enregistrat a alta resolució, sempre en podrem reduir la qualitat, però no en podrem augmentar la resolució si el vídeo va ser gravat a baixa resolució.

Així, els editors més senzills, els que podem trobar a telèfons mòbils o als mateixos equips portàtils d'enregistrament, no ens donaran gaires possibilitats. Per exemple, és comú que aquests editors incorporin un sistema que permeti, almenys, retallar les parts del vídeo que no ens interessen.

No obstant això, si volem fer una edició una mica més completa, com ara variar l'angle d'enregistrament, ajustar els colors de la imatge, fer servir filtres o inserir text, l'haurem de fer des d'un PC, on podrem recórrer a programes d'edició (com ***Kdenlive*** (***kdenlive.org/es***) que ens permetran pràcticament modificar el vídeo a voluntat.

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

Compressió de vídeo

Compressió de vídeo

A l'hora de parlar de vídeo, és convenient diferenciar entre qualitat, resolució i mida, tres conceptes que estan molt relacionats entre si.

La **resolució d'un vídeo** determina la quantitat de detalls que té, així com la nitidesa. Es mesura en funció del nombre de píxels continguts en la relació d'aspecte estàndard de 16:9, la més habitual en pantalles, monitors i televisors. La resolució determina la **qualitat** del vídeo, ja que com més gran sigui la relació de píxels, millor es veurà el vídeo. En els darrers anys, la resolució ha anat augmentant des dels 240p (baixa qualitat) fins als 4K, format de vídeo d'alta definició. Independentment del dispositiu on es reprodueixi el vídeo, com més gran sigui la seva resolució, més nítid es veurà. Sempre es pot fer la imatge més petita sense perdre qualitat, però fer les imatges més grans es traduirà en pèrdues de qualitat. La relació entre les diferents resolucions es pot observar a la figura següent:

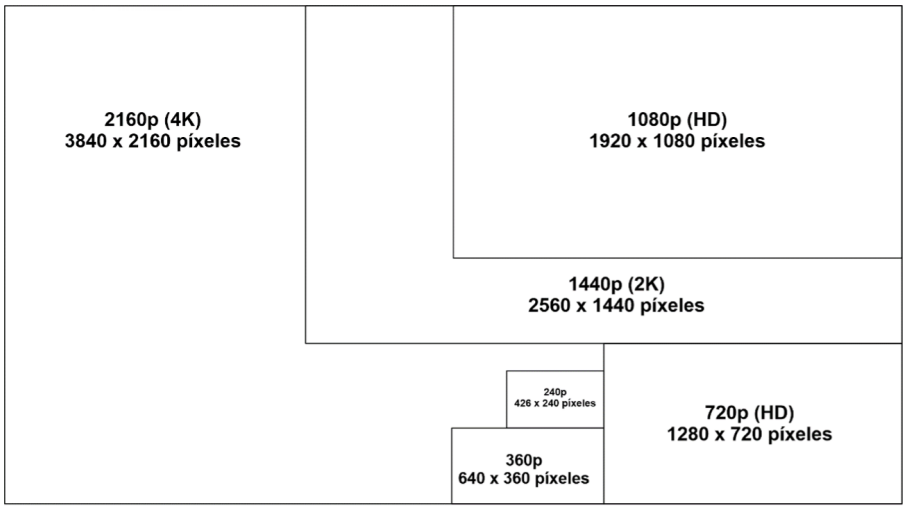


Figura 1. Diferents resolucions de vídeo.

En parlar de **mida**, podem referir-nos a la quantitat de píxels continguts en aquest (resolució) oa la mida en bits que l'arxiu de vídeo ocupa en desar-lo al disc dur de l'ordinador. Entra en joc així el concepte de **compressió**, per aconseguir optimitzar els fitxers de vídeo ocupant la mínima informació possible. Però comprimir comporta una sèrie de riscos pel que fa a qualitat de l'arxiu, per la qual cosa caldrà assolir un compromís.

En augmentar la compressió, la qualitat de la imatge disminueix. Una compressió elevada significa que píxels de colors similars es converteixen en un sol color, de manera que la nitidesa es perd i la imatge queda menys detallada. A més, la compressió de les imatges ha d'anar compensada al moviment temporal, per mantenir una taxa de bits (flux d'imatge) que compleixi els requisits desitjats. La compressió de vídeo permetrà també millorar la transmissió de dades en streaming, ja que la quantitat de dades implicades en un vídeo sol excedir les capacitats de processament de maquinari actuals. Així s'aconsegueix una transmissió més eficient.

Per poder garantir que el vídeo sigui accessible per a l'emissor, caldrà triar el format de fitxer de vídeo correcte. I això no és tan senzill com definir el tipus de format a l'ordinador, sinó que caldrà valorar els tipus de codificació (còdec), contenidors d'arxiu de vídeo i tipus de format de vídeo. El **format del fitxer** jugarà un paper determinant, ja que un fitxer de vídeo no només inclou imatges sinó també àudio i és possible que metadades (com subtítols, estructures de menús o informació del fitxer). Per tant, en el format de vídeo es considera, per una banda, el **còdec**, que comprimeix i descomprimeix el vídeo, i un **contenidor** que agrupa tot i ho fa funcionar a qualsevol programari de reproducció.

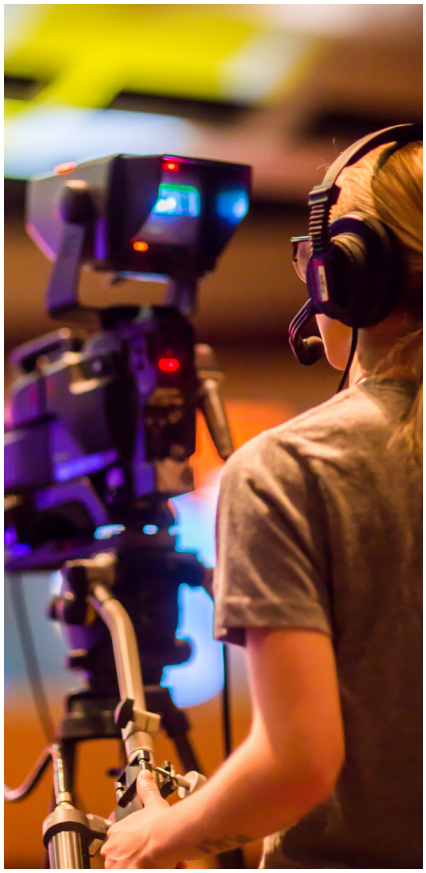
El **còdec de vídeo** és un protocol per codificar i descodificar vídeo. Defineix l'ordre que s'utilitza per dissenyar les dades d'un fitxer d'àudio o de vídeo, de manera que es pugui reproduir i editar. Organitza les dades dels mitjans que es mantenen al contenidor. Hi ha diversos còdecs d'àudio i vídeo, cadascun amb els seus avantatges i inconvenients. Els més comuns són el **H.264**, el **MPEG-4** i el **DivX**. Són còdecs eficients amb una bona capacitat de preservar-ne la qualitat i reduir-ne la mida. Els formats contenidors més utilitzats són **AVI**, **MP4** i **MOV**. Es poden utilitzar amb diferents còdecs en funció dels dispositius i programaris on es reprodueixin.

La funció última del còdec és la de comprimir el vídeo (perquè ocupi menys) i descomprimir-lo en reproduir-lo (perquè es vegi bé). I aquesta compressió pot ser amb pèrdua o sense. La **compressió amb pèrdua** permet obtenir fitxers més petits, i omet dades que donen lloc a fitxers de pitjor qualitat de vídeo. Això es fa molt evident en compressions acumulatives (quan

es comprimeix alguna cosa ja comprimida prèviament). **La compressió sense pèrdua** manté les dades del fitxer original, oferint millor qualitat de vídeo i evitant la degradació del fitxer. Com a desavantatge, ocupen una mida més gran

Alguns dels formats de vídeo més utilitzats són:

- **MP4** és especialment útil per compartir contingut en línia. És el més recomanat per pujar vídeos a YouTube amb bona qualitat. Sol combinar-se amb el còdec H.264 o H.265. Solen tenir mides relativament petites i conservar una bona qualitat.
- **AVI** és un format estàndard força antic i universalment acceptat. Compatible amb una enorme varietat de còdecs, permet moltes configuracions de fitxer diferents. Reproduïbles a la majoria dels reproductors, les mides de fitxer solen ser grans, menys recomanables per a la seva transmissió o descàrrega.
- **MKV** permet emmagatzemar diversos canals o pistes d'àudio i subtítols, amb molt bona qualitat i en molt poc espai. A més, és un format obert pel qual no cal pagar drets.
- **FLV** és un format que es va fer molt comú a causa de la seva mida de fitxer petit i la seva compatibilitat amb la majoria dels exploradors i reproductors Flash. El seu ús ha disminuït actualment.
- **MOV** és un format de vídeo per a Apple, desenvolupat per al reproductor Quicktime. Tenen una qualitat molt alta i ocupen mides de fitxer considerables. Presenta una menor compatibilitat que els anteriors.
- **WMV** és el format de vídeo de Windows Media. Amb mides petites, són una bona opció per a enviaments, encara que no destaca per cap característica especial. Per canviar entre formats hi ha diversos convertidors. Alguns són MiniTool Movie Maker, iMovie (Mac), Video Converter (Android), VLC Media Player i Online Videoconverter.



Per canviar entre formats hi ha diversos convertidors. Alguns són *MiniTool Movie Maker*, *iMovie* (Mac), *Video Converter* (Android), *VLC Media Player* i *Online Videoconverter*.

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

Àudio i compressió

Àudio i compressió

En el procés de creació d'un contingut audiovisual no hi ha dubte que la música, o l'àudio en general, presenta una rellevància que la majoria de vegades supera la de la mateixa imatge. El procés de narrativa de marca, o l'art de com explicar, desenvolupar i adaptar una història en aquest format, requereix l'ús de les eines d'àudio per aconseguir construir un missatge robust que impacti el receptor. La música, els efectes de so o la narració són elements fonamentals per apel·lar a les emocions del públic. Utilitzar aquests recursos de manera eficaç facilitaran que l'oient mantingui l'atenció en el nostre contingut.

Quan parlam de compressió d'àudio, hem d'especificar que ens referim a reduir la mida de l'arxiu, reduir la taxa de bits d'un senyal digital d'àudio tornant-lo més petit, conservant gairebé tota la informació original. En funció del grau de compressió, la pèrdua de qualitat podrà ser imperceptible o molt notòria.

Serà important controlar la mida de l'àudio, perquè ocupi menys espai als dispositius i sigui, per tant, més senzill de transmetre a menys ample de banda.

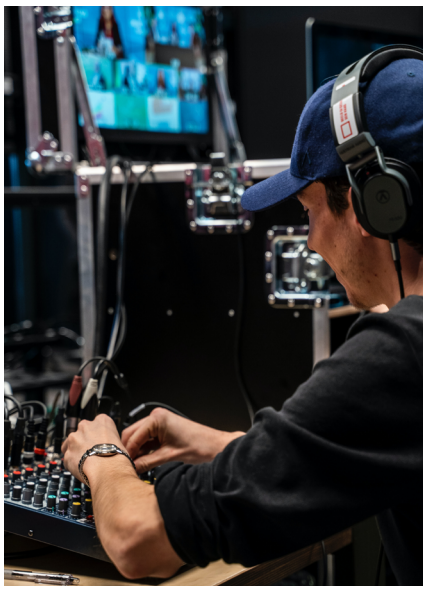
NOTA

En l'àmbit de la producció musical, quan es parla de compressió d'àudio, no es fa referència a la reducció de la mida del fitxer, sinó al processament que permet controlar el rang dinàmic d'un senyal d'àudio. En el context musical, la dinàmica és la diferència entre les parts fortes i les suaus d'una cançó. En el context de l'àudio, en lloc de parts fortes i suaus, hi ha pics individuals i depressions del senyal. El rang dinàmic serà alt si hi ha molta diferència entre aquests pics i valls i baix si la diferència és petita. I aquests nivells es poden regular amb un compressor.

Hi ha diversos tipus de formats en funció de la compressió de l'àudio. Per parlar-ne, distingirem entre formats d'àudio sense comprimir i formats d'àudio comprimits (amb pèrdua i sense).

NOTA

Podeu provar de veure un vídeo que us hagi impactat, sense àudio. Un exemple clàssic és l'escena de la dutxa de *Psicosi* (*e.digitall.org.es/psicosis*)(Alfred Hitchcock, 1960) amb o sense música.



Formats d'àudio sense comprimir

També coneguts com a fitxers Hi-Res o d'alta resolució, són aquells que preserven tota la informació de l'àudio original, processada i emmagatzemada de manera digital. Aquests fitxers proporcionen la més alta qualitat i fidelitat d'àudio, ocupant grans quantitats d'espai d'emmagatzematge. Entre els més habituals hi ha:

- **WAV** (*Waveform Audio Format*) és el format sense comprimir més habitual i utilitzat, tant per la indústria professional com en l'àmbit d'usuari. Sol contenir àudio sense comprimir en format PCM, i és un format adequat per a Windows (encara que Mac també ho reprodueix).
- **AIFF** (*Audio Interchange File Format*) és un altre format sense compressió d'Apple que també es pot reproduir a PC. La majoria dels fitxers contenen àudio sense comprimir en format PCM.

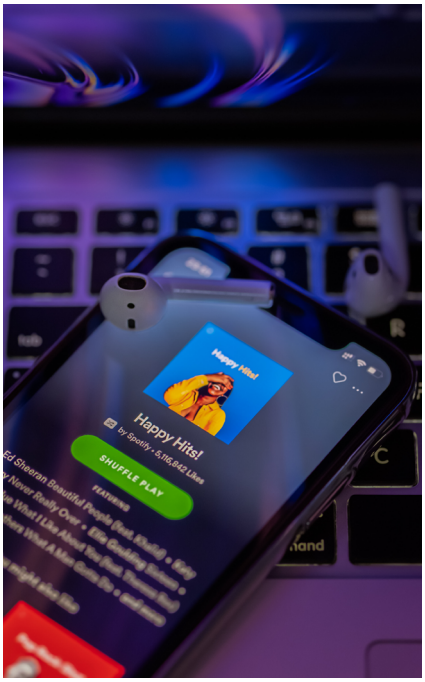
NOTA

El format PCM és la codificació més utilitzada per convertir una ona analògica en una ona digital (seqüència de bits) reproduïble en mitjans digitals.

Formats d'àudio comprimit sense pèrdues

Aconsegueixen comprimir la mida dels fitxers sense que es perdi qualitat general d'àudio. El nivell de compressió o estalvi d'espai digital no és tan gran com el que ens ofereix la compressió amb pèrdues, però sí que s'observa una reducció en comparació amb els formats sense compressió. Els més habituals són:

- **FLAC** (*Free Lossless Audio Codec*), format de compressió sense pèrdua més comú a l'àmbit musical. És de codi obert i inclou metadades incrustades (informació de l'àlbum, etc.). Ocupen aproximadament la meitat del fitxer original.
- **ALAC** (*Apple Lossness Audio Codec*) és molt similar a FLAC, però desenvolupat per Apple. També és de codi obert, tot i ser d'Apple.



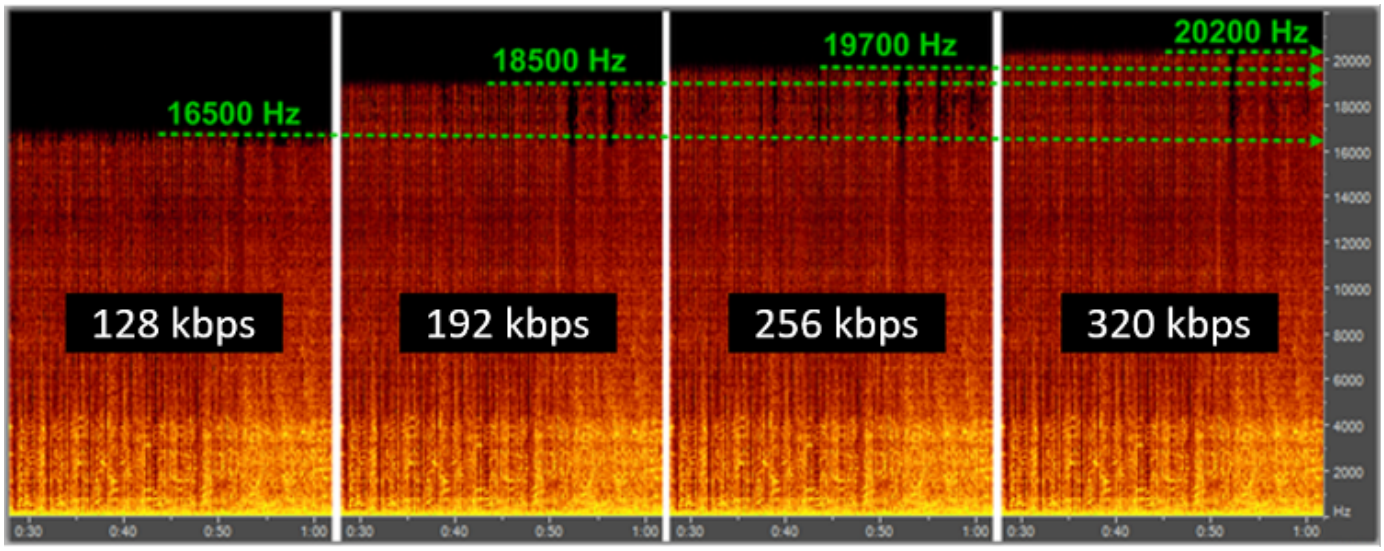
Formats d'àudio comprimit amb pèrdues

Aquests formats sacrifiquen la qualitat per minimitzar la mida. Pesen poc i són reproduïbles per qualsevol dispositiu, però generen un so pobre i sense brillantor. No es fan servir en àmbits professionals, ja que es perd la fidelitat del so. Entre els més habituals podem trobar:

- **MP3** (*Moving Picture Experts Group*) és el més popular dels formats amb pèrdua. El seu algorisme elimina parts o freqüències que són fàcilment audibles per l'orella humana. La seva compressió pot realitzar-se a diferents nivells de kbps (quilobits per segon), i és el de més qualitat el de 320 kbps.
- **AAC** (*Advanced Audio Coding*), també conegut com a MPEG-4. És el format comprimit amb pèrdues utilitzat per iTunes en descàrregues i per YouTube per gestionar l'estríming de l'àudio.
- **OGG Vorbis** és un format comprimit amb pèrdues de codi obert molt utilitzat en plataformes de estríming com Spotify per estalviar amplada de banda (com a referència, la versió gratuïta de Spotify reprodueix a 128 kbps, mentre que la versió prèmium ho fa a 320 kbps).
- **WMA** (*Windows Media Audio*) format creat per Microsoft. Igual que els dos anteriors, pretén abordar algunes de les fallades del mètode de compressió de l'MP3 amb un enfocament similar. Tot i que en termes objectius WMA presenta una compressió de més qualitat que MP3, és un format no admès per molts dispositius. Tampoc no ofereix cap benefici sobre ACC o OGG.

Quin format d'àudio pot ser el més adequat?

Això dependrà de per què necessitem el nostre àudio. En qualsevol cas, si capturam i editam àudio sense processar, la millor opció serà que facem servir un format sense comprimir. Així ens assegurarem que treballam amb la millor qualitat d'àudio possible. En acabar, sempre podrem comprimir (però això no ho podem fer al revés). Si ens interessa conservar una representació d'àudio fidel, serà convenient fer servir compressió d'àudio sense pèrdues. Aquesta és la raó per la qual els melòmans prefereixen el format FLAC per sobre de l'MP3, tot i que requereix disposar de més espai d'emmagatzematge. Finalment, si el fitxer no conté música, cal estalviar espai i la qualitat és un paràmetre sacrificable, es pot recórrer a la compressió d'àudio amb pèrdua. La realitat és que majoria de les persones no aconsegueix distingir entre compressió amb pèrdua i sense. És extremadament difícil percebre diferències, escoltant amb equips o dispositius de qualitat, ja que les compressions solen estar al límit de l'audició humana. Tot i això, amb equips d'alta gamma o auriculars professionals i amb una oïda una mica educada, es poden discernir diferències en matisos de l'àudio. I, sens dubte, mesurant amb un espectròmetre les longituds d'ona, observarem diferències notables.



Exemple de compressió d'àudio. S'observa com, d'esquerra a dreta, en un arxiu en MP3, l'extrem de freqüències agudes (mesurat en Hz) és més gran com més baix és el nivell de compressió (major quantitat de kbps).

Creació de
continguts digitals

Nivell A1 3.1 Desenvolupament de continguts

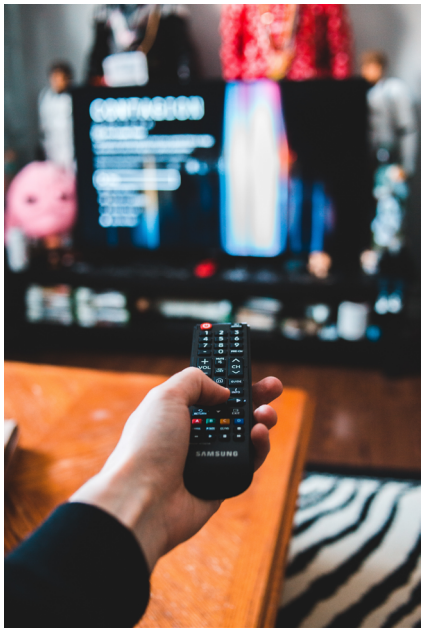
Continguts digitals a Internet

Continguts digitals a Internet

Els **continguts digitals** són fonamentals a l'hora de desenvolupar qualsevol estratègia de comunicació en els temps moderns, bé l'enfocament sigui docent, de màrqueting o qualsevol altre tipus d'objectiu. Sense continguts digitals, Internet seria un desert sense valor ni interès per a ningú. Generar continguts de qualitat en diferents formats serà clau per desenvolupar qualsevol projecte, ja que serà la manera d'atreure el receptor, fidelitzar-lo i aportar-hi alguna cosa. Qualsevol material informatiu que es pugui incloure en un mitjà digital constitueix un contingut digital. Aquests poden estar formats per text, imatge, vídeo, àudio, aplicació, programari o qualsevol altre mitjà d'interacció que puguem imaginar. Entre els més coneguts, podem trobar els següents:

- **Blogs personals:** consisteixen en una plataforma web que permet expressar amb personalitat el missatge de l'emissor. És una manera de contactar directament amb el receptor generant interès sobre els continguts a transmetre de manera creativa, incloent-hi material educatiu, informatiu o rellevant per al públic objectiu. Habitualment els blocs disposen d'una secció de comentaris per a cada entrada on el receptor pot interactuar amb l'emissor i la resta dels receptors.
- **Xarxes socials:** són estructures formades per persones i organitzacions connectades a partir d'interès o valors comuns. A través d'aquestes xarxes es construeixen relacions entre individus o empreses horitzontalment, superant les limitacions físiques. Independentment del seu ús personal o professional, permeten compartir continguts de manera senzilla, ràpida i gratuïta. Serà important definir els continguts de cada perfil perquè el receptor tingui clar que troba allò que busca. Les principals xarxes socials actualment són Facebook, Youtube, Instagram, TikTok, WhatsApp, LinkedIn, X, Pinterest, Telegram, Tumblr, etc.
- **Plataformes de vídeo:** els vídeos són eines útils per il·lustrar un concepte o resoldre dubtes que poden ajudar a mostrar el costat més creatiu o humà d'un projecte. Suposen una manera molt propera de transmetre un missatge. A l'hora d'incloure'ls en un correu electrònic, poden suposar un problema a causa de la grandària que aquests

solen ocupar. Serà recomanable utilitzar plataformes de streaming per pujar aquests continguts, i incloure-hi un enllaç al correu electrònic. Es pot fer servir Youtube o Vimeo, dues de les plataformes gratuïtes més conegudes que permeten compartir aquest tipus de material.



- **Plataformes d'àudio o pòdcast:** suposa una manera còmoda i interessant de desenvolupar la presentació i la discussió d'un tema o esdeveniment particular per mitjà d'un fitxer d'àudio, disponible en un fitxer per descarregar o en estríming. És un format de baixa demanda, ja que el receptor pot escoltar-ho quan vulgui en qualsevol dispositiu. De la mateixa manera que amb el vídeo, pot suposar un problema incloure un fitxer d'àudio com un adjunt en un correu electrònic. Per resoldre aquesta situació, es pot recórrer a SoundCloud o Spreaker, plataformes que permeten penjar fitxers d'àudio en diferents formats i compartir-los en funció de les necessitats. Algunes de les plataformes de gestió de pòdcast més conegudes són iVox, SoundCloud, Anchor, Spotify for Podcasters, Google Podcasts, Spreaker, MixCloud i Apple Podcasts, entre moltes altres

Aprofitar aquests formats permet escurçar distàncies, conèixer nous continguts i apostar per espais de reflexió i creixement personal. El temps i l'esforç de preparació dels continguts haurà de ser optimitzat per aconseguir crear material de qualitat. Amb els continguts digitals s'aconsegueix atreure més públic objectiu, millorar el posicionament i la visibilitat d'un projecte, i humanitzar-lo, dotant-lo d'autoritat.

DigitAll

Creació de
continguts digitals

3.2
INTEGRACIÓ I REELABORACIÓ DE CONTINGUT DIGITAL

Creació de
continguts digitals

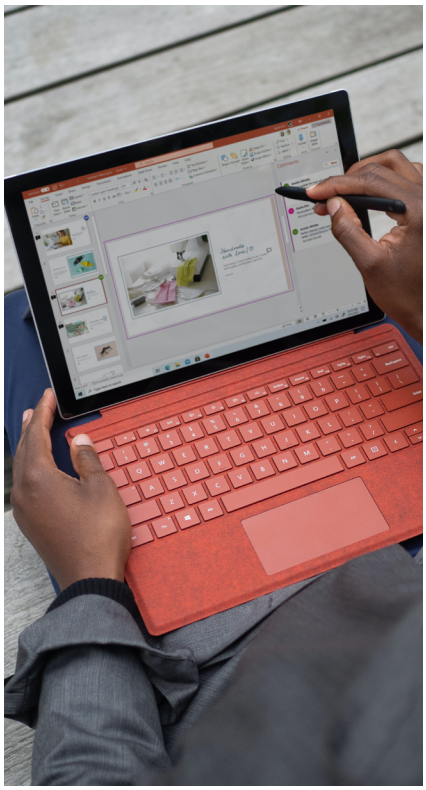
Nivell A1 3.2 Integració i reelaboració de contingut digital

Integració de text, imatges, àudio i vídeo a presentacions

Integració de text, imatges, àudio i vídeo a presentacions

Les eines de creació i edició de presentacions disposen d'una sèrie d'elements comuns que l'usuari utilitzarà segons una sèrie de criteris de disseny i guies d'estil per crear les vostres presentacions. Els més importants són:

- **Diapositiva (*slide*):** unitat bàsica de presentació de la informació. Una presentació constarà d'un conjunt de diapositives, cadascuna de les quals pot incloure informació textual (taules, enllaços a llocs web), gràfic (imatges i gràfics) i, fins i tot, multimèdia (àudio, imatges animades, vídeos).
- **Animacions:** són efectes que poden o no incloure's en els elements que conté una diapositiva. Així doncs, no tots els elements dins d'una diapositiva han de contenir una animació, sinó que serà l'usuari qui estableixi per a cada element les animacions que consideri més adequades. Les animacions us permetran mostrar, emfatitzar o ocultar elements d'una diapositiva per ajudar el presentador a transmetre el missatge.
- **Transició:** espai entre una diapositiva i la següent a la presentació. Aquest és un element opcional en una presentació de diapositives, ja que l'usuari pot optar perquè no hi hagi cap efecte de transició entre dues diapositives. Triar una transició adequada per a una presentació no és una tasca fàcil ni tampoc un aspecte trivial, atès que l'elecció de la transició afectarà el ritme de la presentació.



Utilitzant aquests tres elements principals, qualsevol usuari podrà crear una presentació amb diapositives per a un propòsit concret. Sigui quina sigui l'eina de creació i edició de diapositives utilitzada, aquesta permetrà a l'usuari incloure qualsevol tipus de contingut a les seves diapositives i fer una presentació davant l'audiència, ja que totes incorporen un mode de presentació o passi de diapositives a aquest efecte.

El mode de presentació o passi de diapositives és el que ofereix qualsevol eina de creació i edició de diapositives per impartir la presentació un cop feta. A través d'això, el presentador pot

compartir amb l'audiència les diapositives utilitzades (ocultant els aspectes de disseny) i interactuar-hi a través de les animacions i transicions definides perquè la presentació sigui tan efectiva com sigui possible.

A continuació, es mostren algunes de les eines més populars actualment:

Eina	Logo	Descripció
PowerPoint (privativa)		PowerPoint és el programari de creació i edició de presentacions de ***Microsoft*** (*e.digitall.org.es/powerpoint*) que permet als usuaris crear presentacions atractives que consten de pàgines individuals, o diapositives. És una eina molt completa i el seu ús està àmpliament estès. És compatible amb Windows i macOS. Això no obstant, és una eina de pagament i s'ha d'adquirir una llicència o pagar una subscripció mensual o anual.
Keynote (privativa)		És una eina gratuïta integrada a la suite ofimàtica d'***Apple*** (*apple.com/keynote*) permet crear presentacions efectives d'una manera senzilla. En aquest programari és molt fàcil crear animacions i seqüenciar-les si cal, ja que disposa de multitud d'efectes d'animació molt originals. Keynote permet treballar sense problemes entre dispositius macOS i iOS, a més de poder editar presentacions realitzades amb PowerPoint.
Presentaciones de Google (privativa)		És una eina gratuïta de Google. ***Presentaciones de Google*** (*e.digitall.org.es/slides*) està disponible com a aplicació web, aplicació mòbil per a Android, iOS, Windows, BlackBerry i com a aplicació d'escriptori a ChromeOS de Google. Els usuaris poden accedir a les presentacions a través del lloc web de Google Drive. Les presentacions poden ser editades i compartides per múltiples usuaris simultàniament i els usuaris poden veure els canvis diapositiva per diapositiva a mesura que altres col·laboradors fan edicions. També és compatible amb els formats de fitxer de PowerPoint.
LibreOffice Impress (lliure)	*Galdam Jitsu (Geeko), CC BY-SA 4.0 e.digitall.org.es/cc-licenses via Wikimedia Commons*	És una eina gratuïta i amb llicència de programari lliure desenvolupada per The Document Foundation per a la suite ofimàtica de ***LibreOffice*** (*e.digitall.org.es/impress*). El seu ús està molt estès principalment dins de sistemes operatius lliures, com ara els basats en GNU/Linux. La seva interfície és semblant a PowerPoint i, a més, és compatible amb aquest format.
Beamer (lliure)		Beamer és un complement emprat per la comunitat d'usuaris de l'editor de text LaTeX. Aquest permet la reutilització de documents a ***LaTeX*** (*latex-project.org*) per crear presentacions d'alta qualitat tipogràfica. No és pròpiament una eina de presentació, cosa que limita les seves possibilitats. A més a més, el seu ús és impossible per a usuaris que no coneixen LaTex. Està disponible per als sistemes operatius Windows, macOS, Unix, GNU/Linux, etc.
Prezi (privativa)	Prezi	És una eina de ***Prezi*** (*prezi.com*). Aquestes presentacions parteixen d'una imatge o grafisme que mostra la totalitat de la presentació per, utilitzant la metàfora d'una lupa, anar fent zoom sobre diferents àrees per mostrar el contingut. Les presentacions de Prezi són molt efectistes i dinàmiques, i inclouen animacions que permeten a l'audiència veure, entendre i recordar idees. Aquest tipus de ferramentes està destinada a elaborar presentacions en un context més informal o familiar.
Canva (privativa)	Canva	***Canva*** (*canva.com*) ofereix serveis no només per a la creació de presentacions, sinó també d'infografies i altres tipus de material publicitari. La seva interfície amb disseny drag & drop (arrossegar i deixar anar) fa que sigui molt fàcil per a l'usuari dissenyar i crear una presentació o qualsevol altre tipus de material. A més, Canva ofereix una biblioteca enorme de plantilles amb milers d'elements incorporats i que l'usuari pot utilitzar segons el seu criteri. Aquesta eina està disponible com a aplicació web, aplicació mòbil per a Android i iOS i d'escriptori per a Windows i macOS.

Creació de
continguts digitals

Nivell A1 3.2 Integració i reelaboració de contingut digital

Fulls de càlcul: representació i càlcul amb dades

Fulls de càlcul: representació i càlcul amb dades

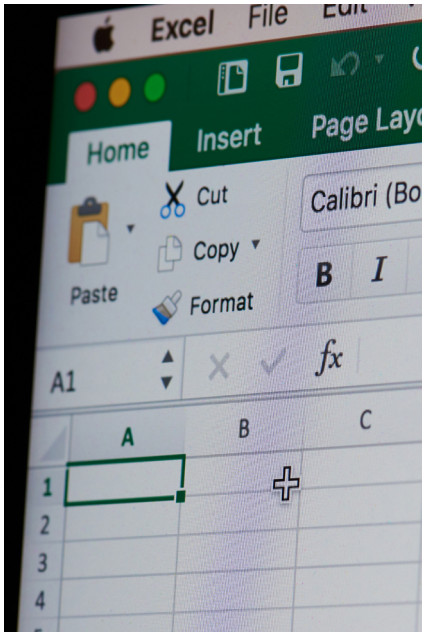
Un full de càlcul és una eina informàtica que permet treballar amb dades numèriques i alfanumèriques. Aquestes eines permeten realitzar operacions matemàtiques amb les dades i obtenir-ne representacions gràfiques. Aquestes figures poden ser reutilitzades en presentacions o informes. Per exemple, un professor pot emprar un full de càlcul per gestionar les qualificacions dels estudiants. Aquesta eina us permetria calcular les qualificacions mitjanes, fer estadístiques o crear llistats d'alumnes que incloguessin gràfics estadístics.

Les eines de fulls de càlcul compten amb una sèrie d'elements comuns, i hi destaquen:

- **Cel·la:** la unitat bàsica de presentació de la informació és la cel·la. S'hi emmagatzema una dada (numèrica o alfanumèrica).
- **Full:** el conjunt de cel·les d'una taula s'anomena full. El conjunt de cel·les horitzontals d'una taula és una fila i el conjunt de cel·les verticals s'anomenen columna. Les files s'identifiquen amb números i les columnes amb lletres. Per tant, cada cel·la queda definida per la fila i la columna que ocupa dins la taula. Per exemple, A3 fa referència a la cel·la de la primera fila i la tercera columna.
- **Rang:** són totes les cel·les contingudes en un rectangle definit per la cel·la superior esquerra i per la cel·la inferior dreta. Per exemple, A1:B2 són les cel·les A1, A2, B1 i B2.
- **Llibre:** és un fitxer d'un full de càlcul i està format per diversos fulls. Cada full té un nom per poder ser referenciat les dades contingudes.
- **Plantilla:** llibre que utilitza com a base per crear altres llibres similars. Es poden crear plantilles per a llibres i fulls.
- **Operacions i funcions:** els fulls de càlcul permeten realitzar operacions aritmètiques (sumes, restes, multiplicacions i divisions) emprant files o columnes. També hi ha càlculs més complexos que es recullen en les funcions esmentades. Les funcions permeten fer càlculs sobre un conjunt de dades situades en un o en diversos fulls. Per

exemple, podeu calcular la mitjana d'una columna de dades, sumar tots els valors d'una fila o comptar el nombre de compareixences d'un determinat número o text en una regió seleccionada. Podeu fer que el contingut d'una cel·la sigui el resultat d'aplicar una funció determinada.

- **Gràfics:** l'usuari pot seleccionar una regió del full i incrustar un tipus de gràfic elaborat a partir d'un conjunt de dades seleccionades. Aquestes eines disposen d'una àmplia varietat de tipus de gràfics predefinits com els de línies, àrees, columnes, barres, dispersió, circulars, histogrames o organigrames entre d'altres.



ATENCIÓ

Els usos més habituals dels fulls de càlcul són:

- **Organitzar dades.** La possibilitat de disposar de múltiples fulls permet classificar i organitzar les dades, a més de realitzar diferents tipus de cerques.
- **Compartir dades.** Els llibres es poden compartir entre usuaris, permetent de manera senzilla i àgil l'accés a les dades a diferents usuaris.
- **Depurar i filtrar dades.** Els fulls contenen totes les dades, però disposen de filtres que permeten visualitzar-ne versions simplificades que permeten l'anàlisi i la depuració d'aquestes.
- **Graficar dades.** Permet fer diversos tipus de gràfics i diagrames amb l'objectiu de facilitar la comprensió de la informació, analitzar patrons, etc. Aquests gràfics es poden incloure posteriorment en presentacions o en informes.
- **Manipulació matemàtica de les dades.** Es poden fer senzillament càlculs matemàtics amb les dades d'un llibre. Aquests càlculs poden anar d'operacions simples fins a problemes matemàtics complexos.

A continuació, es mostren algunes de les eines més populars actualment:

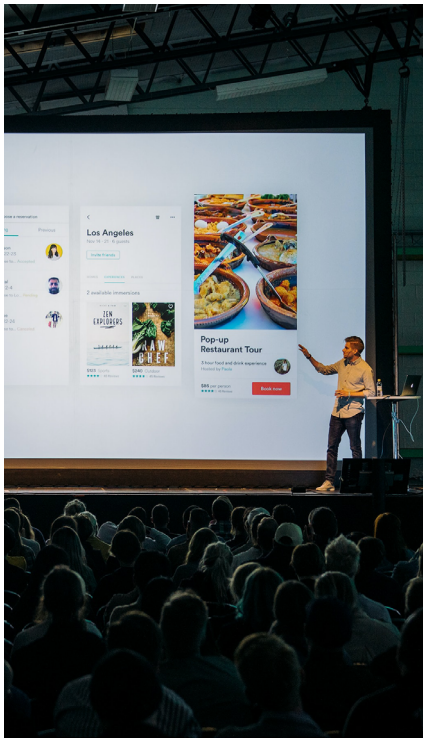
Eina	Logo	Descripció
Excel (privativa)		Excel és el programari de fulls de càlcul de ***Microsoft*** (e.digitall.org.es/excel) que permet als usuaris treballar amb múltiples funcions. És una eina molt completa i el seu ús està àmpliament estès. És compatible amb Windows i macOS. Això no obstant, és una eina de pagament i s'ha d'adquirir una llicència o pagar una subscripció mensual o anual.
Numbers (privativa)		És una eina gratuïta integrada a la suite ofimàtica d'***Apple*** (apple.com/numbers) que permet crear fulls de càlcul d'una manera senzilla. Destaca per la seva simplicitat i elegància. Numbers permet treballar sense problemes entre dispositius macOS i iOS, a més de poder afegir diagrames utilitzant un Apple Pencil a iPad. A més, incorpora una gran varietat de plantilles predefinides per a una gran quantitat d'escenaris d'ús.
Fulls de càlcul de Google (privativa)		És una eina gratuïta de Google. ***Fulls de càlcul de Google*** (e.digitall.org.es/sheets) està disponible com a aplicació web, cosa que permet als usuaris treballar amb diferents sistemes operatius. Els usuaris poden accedir als fulls de càlcul mitjançant el lloc web de Google Drive. Els fulls de càlcul es guarden al núvol i poden ser usats per múltiples usuaris simultàniament. Els usuaris poden veure els canvis a mesura que altres col·laboradors fan edicions. També és compatible amb altres formats externs, inclòs el fitxer d'Excel.
LibreOffice Calc (lliure)	*Galdam Jitsu (Geeko), CC BY-SA 4.0* e.digitall.org.es/cc-licenses via Wikimedia Commons	És una eina gratuïta i amb llicència de programari lliure desenvolupada per The Document Foundation per a la suite ofimàtica de ***LibreOffice*** (e.digitall.org.es/calc). El seu ús està molt estès principalment dins de sistemes operatius lliures, com ara els basats en GNU/Linux. La seva interfície és semblant a Excel i a més és compatible amb aquest format.
Zoho Sheet (privativa)		***Zoho Sheet*** (zoho.com/sheet) és una aplicació que forma part de la suite ofimàtica d'aquest desenvolupador. Compte amb les funcions habituals, taules dinàmiques i gràfics. Realitza el treball al núvol i, per tant, permet el treball col·laboratiu i accedir als projectes des de qualsevol ordinador amb connexió a internet. És gratuït fins a 25 usuaris, per la qual cosa és una alternativa per a petites empreses i usuaris finals.

Creació de
continguts digitals

Nivell A1 3.2 Integració i reelaboració de contingut digital

Composició de continguts digitals existents

Composició de continguts digitals existents



Quan preparem o elaborem contingut digital per a qualsevol propòsit (presentar un producte, anunciar un servei, elaborar contingut sobre un esdeveniment o viatge personal), sovint disposem d'elements com ara imatges, àudios o vídeos que volem **combinar** per generar el nou contingut. En aquests casos, el nostre objectiu és generar un contingut digital tipus ***slideshow***, és a dir, una composició d'imatges, vídeos i sons amb un propòsit específic.

Actualment, hi ha multitud d'**eines de programari gratuïtes i intuïtives** perquè qualsevol usuari sense un coneixement expert pugui generar el seu contingut digital. Algunes d'aquestes eines fins i tot són accessibles directament des del **web**, sense necessitat d'instal·lar cap programari específic. Totes tenen un sistema d'edició comú, encara que hi hagi certes diferències i particularitats als menús i botons de cadascuna. Per això, utilitzen un sistema de tipus ***drag and drop***, que permet arrossegar i deixar anar en una línia temporal els elements que volem que formin part del nou contingut. Aquests elements es mostraran tal com els hàgim ordenat a la línia de temps.

A l'hora d'utilitzar **imatges**, hem de saber que és possible fer múltiples transformacions i processaments per aconseguir l'efecte desitjat. Així, és possible utilitzar programes i eines web que permeten **incloure text** en una foto (per crear, per exemple, un mem), **combinar diverses fotos existents** (creant muntatges on, per exemple, el cap d'una persona apareix al cos d'una altra, es fusionen dues escenes o dos paisatges...), **crear animacions** a partir de diverses imatges, fer-ne efectes per aconseguir un resultat original i **crear vídeos** a partir d'un conjunt d'imatges.

Per part seva, l'**àudio** és un element molt útil per donar **ritme** a la presentació i generar ambient. Per això, és possible introduir peces d'àudio que acompanyin una imatge per aportar més acció (per exemple, un àudio de bullici sobre la imatge d'un embús) o inserir àudio de fons al llarg de diferents imatges amb l'objectiu de crear ambient i generar **emocions** (per exemple, incloure una melodia d'una cançó de bressol sobre les fotos d'un nadó). En un vídeo, també és possible **substituir** tot o part d'àudio que conté el vídeo per un altre de la nostra elecció.

Pel que fa al **vídeo**, és l'element multimèdia amb més capacitat de transmissió de continguts. És especialment important que tant les imatges com els àudios i vídeos que s'incloguin en el nou contingut siguin de **bona qualitat**. A més, en el cas de l'àudio i el vídeo és especialment important assegurar-se que els fitxers d'aquest tipus que es volen incloure són **compatibles** amb el programari que s'està utilitzant per generar el contingut digital. En cas de no ser-ho, hi ha **eines gratuïtes de conversió d'arxius** que es poden utilitzar, tant com a aplicació d'escriptori com en versió web. Quan el contingut ja ha estat elaborat, el més comú és exportar-lo en format de vídeo. Aquest procés es coneix amb el nom de **renderització**. El que està renderitzat portarà més o menys temps i generarà un fitxer de menys o més pes en funció de la quantitat d'elements introduïts, la durada del vídeo i la resolució desitjada.

Per **combinar** tots els elements anteriors també és possible **utilitzar transicions i animacions o efectes especials**. Les transicions permeten combinar diferents elements mitjançant efectes que permeten donar continuïtat al contingut. Les animacions o efectes especials, alhora, aporten dinamisme al contingut generat per millorar-ne l'impacte.

De totes aquestes maneres és possible crear continguts digitals a partir d'altres existents. Aquests nous continguts van des d'un **mem**, una **infografia** o una **presentació** fins a continguts més complexos de tipus ***slideshow*** que combinin tots els elements multimèdia disponibles.

Saber-ne més

Eines per a conversió de fitxers:

• Convertidor en línia gratuït d'àudio que suporta fins a 300 formats.
online-audio-converter.com/es

• Eina que permet convertir qualsevol format de text, imatge, vídeo o àudio, a un altre del mateix tipus.
online-convert.com

DigitAll

Creació de
continguts digitals

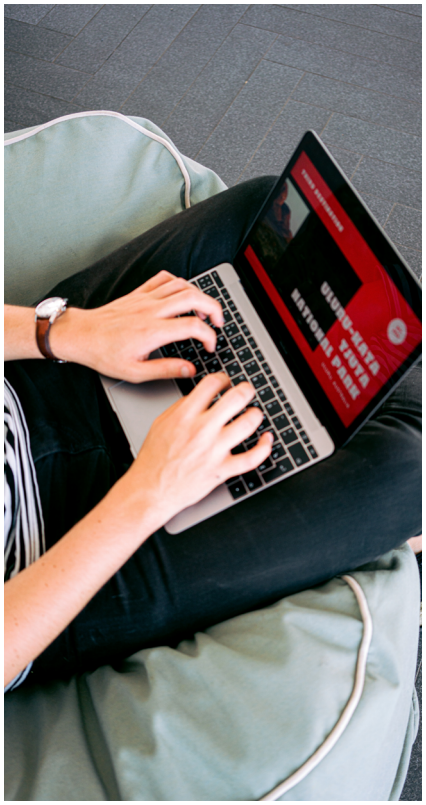
3.3
DRETS D'AUTOR I LLICÈNCIES DE PROPIETAT INTEL·LECTUAL

Creació de
continguts digitals

Nivell A1 3.3 Drets d'autor i llicències de propietat intel·lectual

Drets d'autor i llicències de propietat intel·lectual

Drets d'autor i llicències de propietat intel·lectual



Propietat intel·lectual: conceptes fonamentals

Són objecte de propietat intel·lectual totes les creacions originals literàries, artístiques o científiques expressades per qualsevol mitjà o suport, tangible o intangible, actualment conegut o que s'inventi en el futur, entre les quals es comprenen:

- Els programes d'ordenat.
- Les col·leccions d'obres alienes, com les bases de dades que per la selecció o la disposició dels seus continguts constitueixin creacions intel·lectuals. La protecció reconeguda a aquestes col·leccions es refereix únicament a la seva estructura com a forma d'expressió de la selecció o disposició dels seus continguts, i no és extensiva a aquests.

Drets d'autor en creacions intel·lectuals

La propietat intel·lectual està integrada per drets de caràcter personal i patrimonial, que atribueixen a l'autor la plena disposició i el dret exclusiu a l'explotació de l'obra, sense més limitacions que les establertes a la llei.

Es considera autor a la persona natural que crea alguna obra literària, artística o científica. Es presumirà autor, llevat de prova en contra, a qui aparegui com a tal a l'obra, mitjançant el seu nom, signatura o signe que ho identifiqui. La condició d'autor té un caràcter irrenunciable; no es pot transmetre "inter vivos" ni "mortis causa", no s'extingeix amb el transcurs del temps així com tampoc entra en el domini públic ni és susceptible de prescripció.

La propietat intel·lectual d'una obra literària, artística o científica correspon a l'autor pel fet de la seva creació. Inclou el conjunt de drets que corresponen als autors i altres titulars (artistes, productors, organismes de radiodifusió...) respecte de les obres i prestacions fruit de la seva creació.

En el sistema anglosajón, a diferencia del continental, el derecho de autor se conoce como "copyright".

- Propietat intel·lectual = Drets d'autor = Sistema continental
- Copyright = Drets d'explotació = Sistema anglosaxó

Definició de copyright

El símbol del Copyright © informa al públic que una obra és original i que el seu ús, reproducció, transformació, publicació, etc. està subjecte a drets d'autor. El titular o cessionari en exclusiva d'un dret d'explotació sobre una obra o producció protegides per la llei pot anteposar al seu nom el símbol © amb precisió del lloc i l'any de la divulgació d'aquelles. No cal sol·licitar la inscripció dels drets de propietat intel·lectual de l'obra en qüestió per incloure el símbol ©.

Els símbols i les referències esmentats s'han de fer constar en mode i col·locació de tal manera que mostrin clarament que els drets d'explotació estan reservats.

Public Domain (Domini Públic)

Els drets d'explotació d'una obra, en qualsevol forma i, en especial, els drets de reproducció, distribució, comunicació pública i transformació duraran tota la vida de l'autor i 70 anys després de la seva mort o declaració de mort. L'extinció dels drets d'explotació de les obres en determinarà el pas al domini públic. Les obres de domini públic podran ser utilitzades per qualsevol, sempre que es respecti l'autoria i la integritat de l'obra i impedint qualsevol deformació, modificació, alteració o atemptat que suposi perjudici als seus legítims interessos o menyscabament a la seva reputació.

Les obres de domini públic poden estar regulades per Creative Commons amb dues marques:

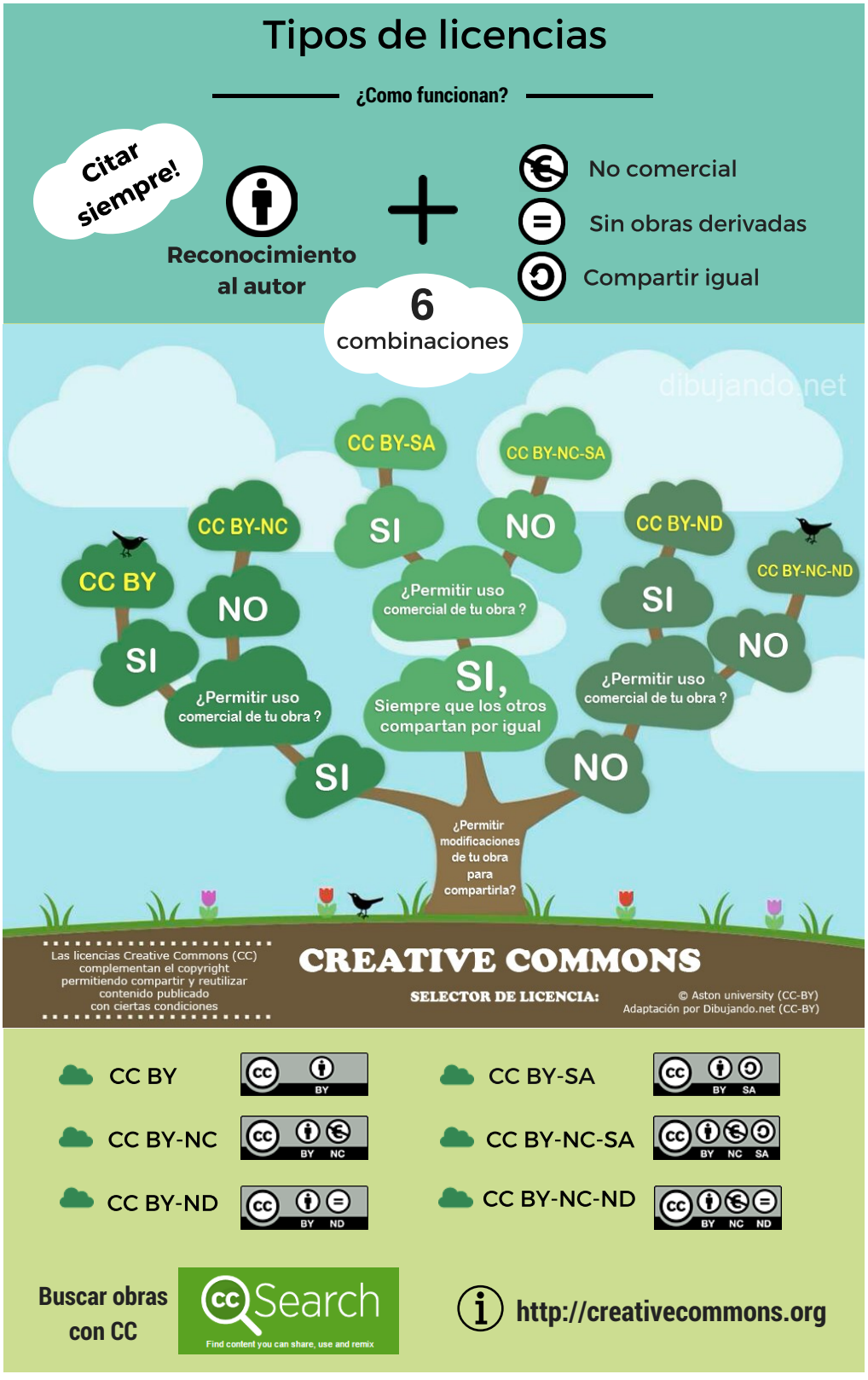
- **© Public Domain:** indica les obres que no estan protegides per drets d'autor (d'explotació) per haver expirat el termini de protecció i que, per tant, pertanyen al Domini Públic.
- **0 Public Domain:** és una marca que els autors de noves creacions poden atorgar a les seves obres i que indica que renuncien a qualsevol dret sobre aquesta, per la qual cosa té el mateix tractament que si es tractés d'una obra al Domini Públic.

Drets d'autor en creacions intel·lectuals

Les llicències CC permeten als autors, o altres titulars de drets, autoritzar i cedir, sota certes condicions, alguns dels drets sobre les seves obres i mantenir-ne una altra part. Les CC es poden aplicar a qualsevol tipus de continguts creatius, incloses les bases de dades.

Es gestionen a través de ***Creative Commons*** (***creativecommons.org***), organització no governamental i sense ànim de lucre que ofereix autors i creadors, llicències i eines lliures.

Cada llicència CC detalla quins drets cedeix l'autor, sota quines condicions i què poden fer els usuaris amb l'obra, sobretot en el cas que vulguin tornar a publicar-la o modificar-la. Hi ha 4 condicions bàsiques que es combinen a les llicències CC: Reconeixement (obligatori i imprescindible), Ús No comercial, Sense Obres Derivades, i Compartir Igual. La seva combinació genera 6 llicències CC diferents, tal com s'esquematitza al següent esquema:



Iconos creados por Freepik desde www.flaticon.com (CC BY 3.0)

crue Universidades Españolas Red de Bibliotecas REBIUN

Font: ***Creative commonESP*** (*e.digitall.org.es/rebiun*)

Creació de
continguts digitals

Nivell A1 3.3 Drets d'autor i llicències de propietat intel·lectual

Plagi

Plagi

Com ja hem explicat en unitats anteriors, tots els drets respecte de les obres literàries, artístiques i científiques pertanyen als autors, siguin aquests artistes, escriptors, productors o programadors informàtics. I ells són els que han d'autoritzar l'ús i l'explotació de les seves creacions, ja que la propietat intel·lectual empara tots els seus drets.

En els darrers 20 anys s'està produint un increment exponencial dels plagis, a causa, en bona part, de la irrupció d'Internet i de la massificació del seu ús.

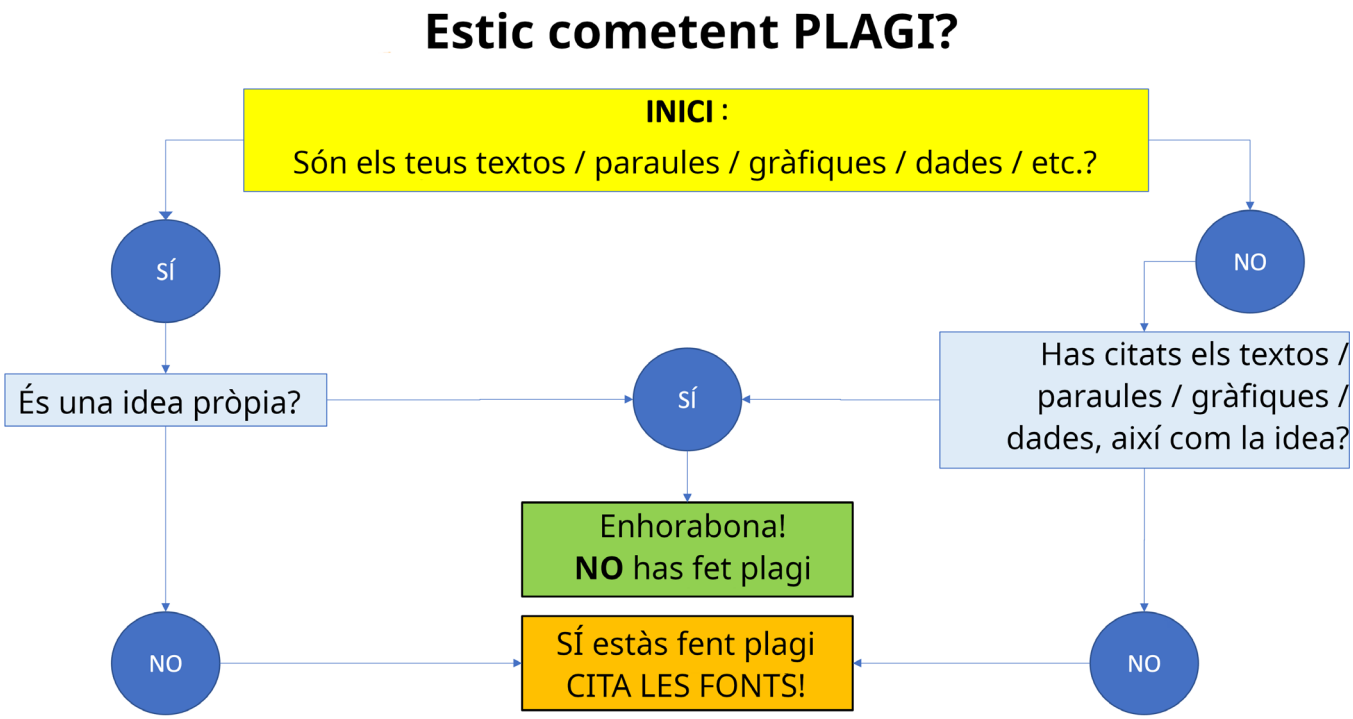
Plagiar consisteix a copiar en allò **substancial** obres alienes, presentant-les com a pròpies. Això no obstant, és important assenyalar que no tota coincidència entre dues o més creacions suposa un plagi. Només quan parlem de **coincidències estructurals bàsiques i fonamentals**, i no quan són accessòries o afegides, és a dir, no transcendentals.

Moltes persones, de manera voluntària o per desconeixement, descarreguen de la xarxa textos, fotografies, programes informàtics, articles acadèmics o científics, etc. **Aquesta informació no es pot fer servir lliurement i com si fos pròpia**, ja que estan protegits mitjançant la Propietat Intel·lectual, excepte en els casos en què l'autor ho autoritzi expressament. Igual que la resta d'informació, un bloc, una pàgina web o un document electrònic són obres d'algun autor i estan subjectes a les mateixes condicions d'ús i cita, atès que estan emparades pels drets d'autor.

No has de copiar i enganxar sense més ni més. **Sempre cal citar la font de procedència**. Depenent de la gravetat de la còpia, es pot cometre una infracció greu regulada al Codi Penal (**articles 270 a 272** (*e.digitall.org.es/legislacion*)) i que comporta greus conseqüències per a l'infractor, com són penes de presó de 6 mesos a 4 anys. L'article 270.2 és el responsable que, durant els darrers anys, hagin hagut de tancar webs com RojaDirecta, Series. ly o EliteTorrent, de gran fama entre els internautes. Tot i que només mostraven enllaços a continguts carregats a la web pels seus usuaris, els Tribunals els va fer responsables de delictes contra la Propietat Intel·lectual.

A l'entorn universitari, més del 60% dels estudiants ha incorregut en alguna forma de **plagi acadèmic** per fer els seus treballs. Per evitar-ho, l'estudiant no ha de copiar el text, les gràfiques, les dades, etc., tal com apareixen als articles originals, llevat que se'n citi l'autor i l'obra d'on procedeixen i es transcriguin entre cometes.

El diagrama següent et permetrà determinar si estàs o no plagiant:



DigitAll

Creació de
continguts digitals

3.4
PROGRAMACIÓ

Creació de
continguts digitals

Nivell A1 3.4 Programació

Característiques dels algorismes i resolució de problemes

Característiques dels algorismes i resolució de problemes

Hi ha multitud de problemes que es poden resoldre mitjançant un ordinador, però tots tenen en comú el que parteixen d'unes dades d'entrada i es processen mitjançant una sèrie d'instruccions per proporcionar uns resultats; és a dir, segueixen l'esquema que es mostra a la Figura 1. Els programes són línies de codi escrites per persones, anomenades programadors, en algun llenguatge de programació que entén l'ordinador i que resolen un problema. Podria pensar-se que per solucionar un problema només cal aprendre un llenguatge de programació i seure directament a escriure codi. Tot i això, això seria equivalent a construir un cotxe sense abans haver-lo dissenyat o a rodar una pel·lícula sense un guió previ. En el context de la programació, els algorismes constitueixen una de les eines bàsiques per al disseny dels programes.

Figura 1. Parts d'un problema resoluble amb un ordinador.

Un algorisme és la descripció de la solució d'un problema mitjançant la **seqüència ordenada de passos** que el resolen. Es caracteritza per ser:

- **Precís,** és a dir, no ha de posseir ambigüitats per ser interpretat sempre de la mateixa manera.
- **Robust,** és a dir, no ha de contenir errors.
- **Definit,** responent sempre de la mateixa manera davant de les mateixes circumstàncies.
- **Ordenat,** deixant clar quina és la seqüència d'instruccions que cal fer.
- **Finito,** cosa que significa que acaba en algun moment.
- **Llegible,** és a dir, comprensible per a qualsevol persona que el llegeixi.

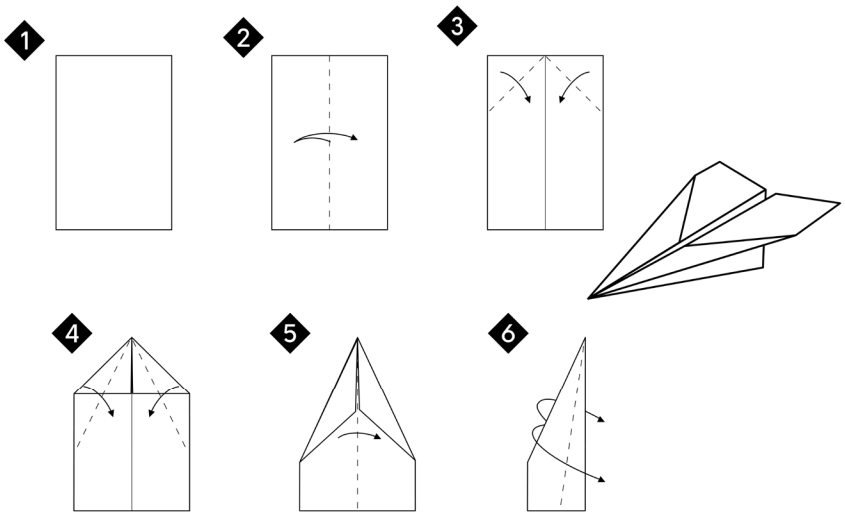


Figura 2. Instruccions per fer un avió de paper. (e.digitall.org.es/aviones-papel)

La Figura 2 il·lustra un algorisme per solucionar el problema consistent en fer un avió de paper. En aquest exemple, l'entrada és el full de paper; la sortida, l'avió construït; i les instruccions estan representades per les imatges de la 1 a la 6. Fixa't que l'algorisme està descrit gràficament i que compleix totes les propietats: precís, robust, definit, ordenat, finit i llegible.

En el context de la programació, els algorismes es poden escriure de moltes maneres, encara que se sol utilitzar el pseudocodi per ser un llenguatge semblant al que fem servir per comunicar-nos, però sense les ambigüitats i imprecisions pròpies del llenguatge natural. Ho aprendràs tot sobre el pseudocodi en un altre tema posterior. Ara el més important és que assimilis el procés previ relacionat amb la resolució de problemes. Perquè per dissenyar l'algorisme que descriu la solució d'un problema, abans és imprescindible **analitzar el problema** de manera detallada, per tal d'identificar cadascuna de les tres parts (Figura 1): dades d'entrada, instruccions per processar-les i dades de sortida. Per fer-ho, pots intentar donar resposta a les preguntes següents:

- Què vull obtenir? (*Sortida*)
- Quines dades tenc? (*Entrada*)
- Quins instrumentos tenc? i,
- Quines instruccions cal seguir i en quin ordre?

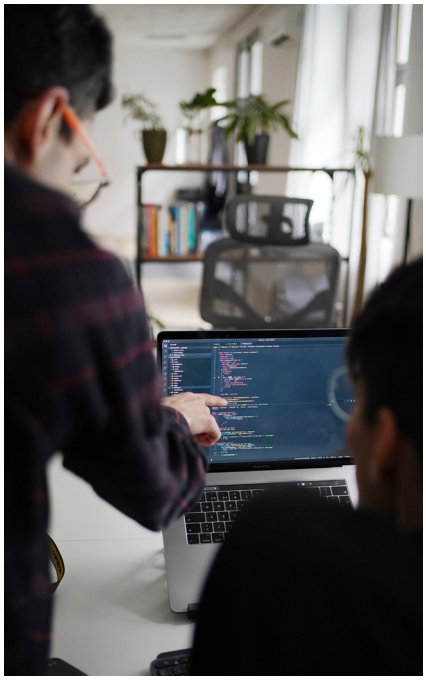
} Processament

Ho entendràs tot d'una amb un exemple:

Imagina que et demanen que escriguis un programa que calculi l'àrea d'un cercle, de ràdio un nombre real positiu. Per abordar-ho, has de respondre les preguntes anteriors:

- **Què vull obtenir?**
  Un número que representi l'àrea del cercle.
- **Quines dades tenc?**
  Amb el nombre que representa el radi.
- **Quins instruments tenc?**
  L'operació producte la proporcionen tots els llenguatges de programació.
- **Quines instruccions cal seguir i en quin ordre?**
  Vegem-ho:

  P1. Anomenam ràdio, per exemple, el número que representa el radi del cercle
  P2. Cal obtenir el valor del radi (suposem que se li dona el valor 2)
  P3. Anomenam PI, per exemple, el valor 3,14
  P4. Anomenam àrea, per exemple, el nombre que representa el resultat de l'expressió PI*ràdio*ràdio (és a dir, de 3,14*2*2)
  P5. El valor d'àrea es proporciona com a resultat (el número 12,56)



Observa que no es pot mostrar el valor d'àrea si encara no se n'ha calculat el valor; és a dir, el pas P5 sempre ha d'anar després del pas P4. O no es pot fer un producte de dos números si no se'n coneix el valor; és a dir, que el pas P4 sempre ha d'anar després del pas P2. Tot i això, el pas P3 pot anar abans o després del pas P1, i abans o després del pas P2.

⚠ ATENCIÓ

L'ordre de les instruccions és tan important com les instruccions mateixes.

A més, el nom dels números i les expressions podria haver estat un altre. Per exemple, al nombre que representa el radi se li podria haver posat r, R, radius, X, o qualsevol altre que ens agradi. El mateix per al nom del valor 3,14159 i per al valor de l'expressió que calcula l'àrea. Per tant, es poden crear diferents algorismes igual de vàlids que el que hem escri

⚠ ATENCIÓ

L'algoritme que descriu la solució d'un problema no ha de ser únic.

Perquè et vagis familiaritzant amb el pseudocodi, una possible descripció de l'algorisme de l'exemple seria el següent:

```
Algorisme AreaCercle
Dades
  radi, àrea, PI: Números
Instruccions
  Obtenir el valor de radi
  PI ← 3,14
  àrea ← PI * radi * radi
  Proporcionar el valor d'àrea
Fi AreaCercle
```

Finalment, podeu comprovar que l'exemple compleix totes les propietats dels algorismes: cal, robust, definit, ordenat, finit i llegible. A més, l'algorisme és correcte: fa allò que ha de fer. Per tant, és el moment de traduir-lo a un llenguatge de programació, cosa que es coneix com a 'implementació'. La Figura 3 conté diverses implementacions de l'algorisme de l'exemple a Pascal, Python i Java. Veureu que totes són molt semblants a l'algorisme en pseudocodi i, a més, entre elles, diferenciant-se únicament en detalls específics de sintaxi.

```
program AreaCirculo;
var radio, PI, area:Real;
begin
  radio:=2;
  PI:=3.14;
  area:=PI*radio*radio;
  writeln (area);
end.
```

```
radio=2
PI=3.14
area=PI*radio*radio
print(area)
```

```
public class AreaCirculo {
        public static void main(String[] args) {
                double radio=2;
                double PI=3.14;
                double area=PI*radio*radio;
                System.out.println(area);
        }
}
```

Figura 3. Possibles implementacions de l'algorisme de l'exemple a Pascal, Python i Java.

En definitiva, un cop definit l'algorisme, la implementació és gairebé immediata. Per tant, per programar la solució d'un problema, el procés que cal seguir, de manera resumida, és el següent:

1 | Anàlisi del problema
Per identificar les dades d'entrada, els passos que cal seguir per al processament i la sortida que heu de proporcionar.

2 | Disseny de l'algorisme
Cal comprovar que compleix totes les propietats i que és correcte. S'escriurà en pseudocodi.

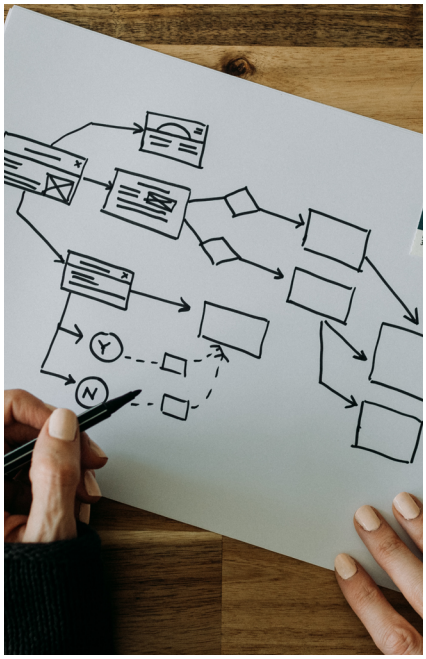
3 | Implementació
Traducció de l'algorisme al llenguatge de programació escollit.

Creació de
continguts digitals

Nivell A1 3.4 Programació

Diagrames de flux

Diagrames de flux



Tal com s'ha comentat anteriorment, els diagrames de flux permeten representar visualment el flux de qualsevol algorisme, facilitant-ne així la comprensió. És una eina àmpliament utilitzada en l'àmbit de la programació de computadors, especialment entre equips de desenvolupament programari per a la documentació de projectes, i l'intercanvi d'idees i conceptes.

Vegem-ne un exemple senzill d'ús. Per això, hem de suposar el problema següent, en què es vol dissenyar un algorisme per controlar la venda d'entrades a un concert, limitada a majors d'edat. L'algorisme ha de determinar si la venda està permesa o no. L'algorisme que resoldria la problemàtica esmentada anteriorment podria ser el següent:

1 | Inici del programa

2 | Sol·licitar a l'usuari l'any de naixement

3 | Sol·licitar a l'usuari l'any actual

4 | Calcular l'edat d'acord amb les dues dades anteriors com la diferència entre l'any actual i l'any de naixement

5 | Consultar l'edat i comproveu si és igual o superior als 18 anys

- En cas afirmatiu, és a dir, que l'edat sigui igual o superior als 18 anys, el programa ha de mostrar per pantalla un missatge informant que la venda és permesa.
- En cas negatiu, és a dir, l'edat és inferior als 18 anys, cal informar per pantalla que la venda no està permesa.

6 | Fi del programa

Tot i que el càlcul de l'edat requereix una precisió més gran tenint en compte el dia de naixement i el mes, per a aquest primer exemple s'ha decidit simplificar únicament a l'any de naixement. L'algorisme anterior té la representació equivalent en un diagrama de flux, tal com mostra la figura següent.

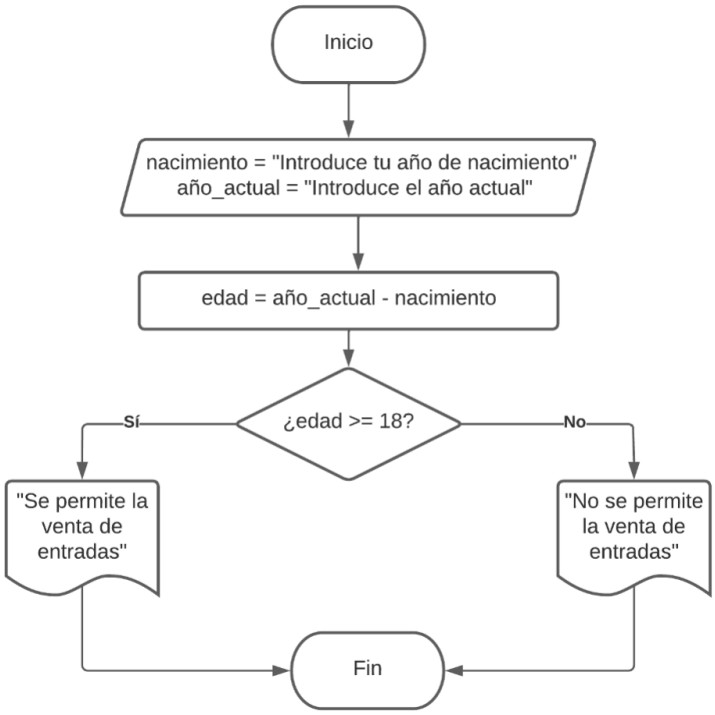
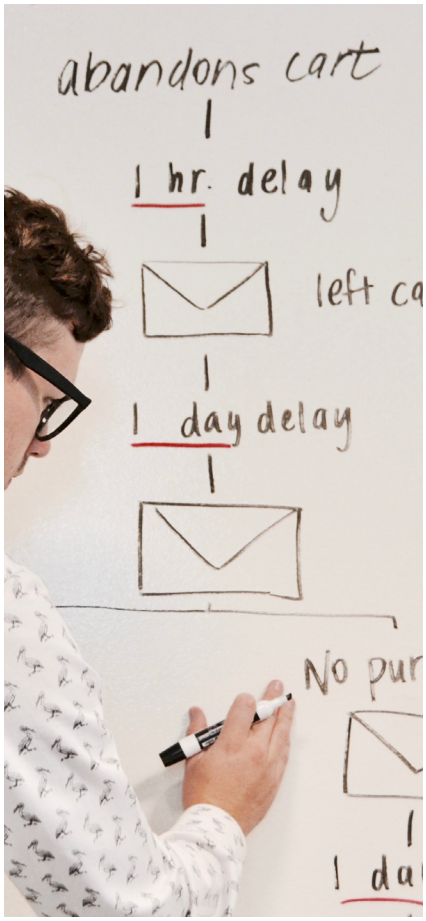


Figura 1. Diagrama de flux associat a un algorisme per al control de venda d'entrades segons l'edat.

Inicialment es marca l'inici del programa. Les dues primeres instruccions que heu d'executar el programa corresponen a la sol·licitud de dades d'entrada, representades mitjançant un paral·lelogram. Perquè l'usuari sigui conscient que el programa sol·licita dades i està en espera d'entrada, cal mostrar missatges per pantalla. En primer lloc, es mostra per pantalla la frase "Introdueix el teu any de naixement", i la dada introduïda per teclat s'emmagatzema a una variable anomenada naixement. En segon lloc, es mostra per pantalla la frase "Introdueix l'any actual" i la dada introduïda s'emmagatzema a la variable any_actual. Posteriorment, es procedeix al càlcul de l'edat restant el contingut emmagatzemat a les dues variables anteriors. Com que es tracta d'un procediment, es representa mitjançant un rectangle. Una vegada calculada l'edat és possible consultar-ne el valor (representat mitjançant un rombe). En funció de si l'avaluació deriva en un valor veritable o fals, el flux de l'algorisme continua per un camí diferent. Camí de l'esquerra en cas que l'edat sigui més gran o igual a divuit, i el camí de la dreta en cas contrari. En tots dos casos es mostra un missatge a l'usuari informant de la decisió presa. Finalment es marca la finalització del programa.



Creació de
continguts digitals

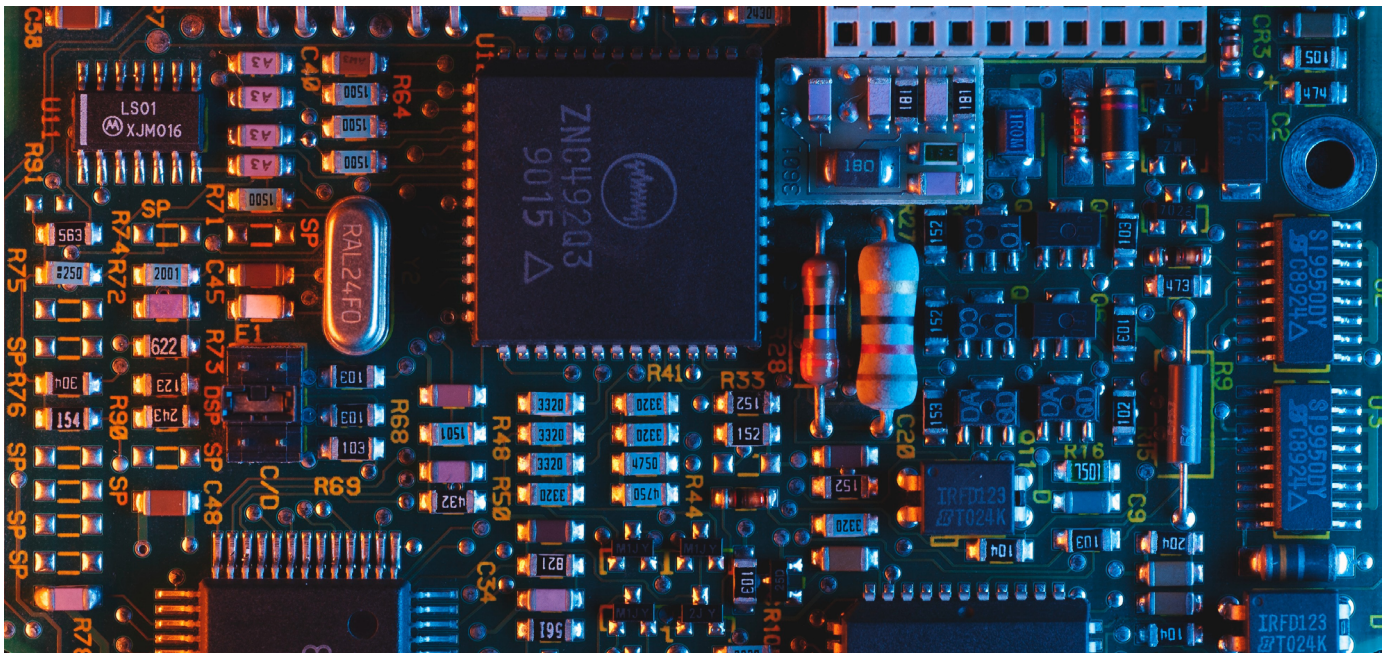
Nivell A1 3.4 Programació

Màquines programables. El concepte de programa

Màquines programables. El concepte de programa

Els mòbils, tauletes, portàtils i ordinadors o ordinadors personals que fem servir en el nostre dia a dia, i als quals ens podem referir amb caràcter general com a **màquines**, no deixen de ser un conjunt de circuits integrats amb multitud de xips connectats entre si, que ens permeten fer una gran varietat de tasques a les nostres activitats quotidianes. A l'àmbit laboral i escolar són un instrument de treball i en el personal són un instrument de relació, desenvolupament personal i d'entreteniment. Sens dubte, aquestes màquines o dispositius informàtics són, avui dia, unes eines fonamentals a la nostra vida diària.

Podem reflexionar sobre un exercici molt senzill i pensar en la quantitat de dispositius electrònics que integren algun tipus de dispositiu de processament. Les televisions, els equips de cinema a casa, les consoles de vídeo, els assistents personals com Alexa, els dispensadors automàtics de menjar per a mascotes, els frigorífics, les rentadores, les calderes o els equips d'aire condicionat són alguns exemples d'elements que ens envolten i que estan basats en tecnologia, materialitzada per una gran varietat de petites màquines.



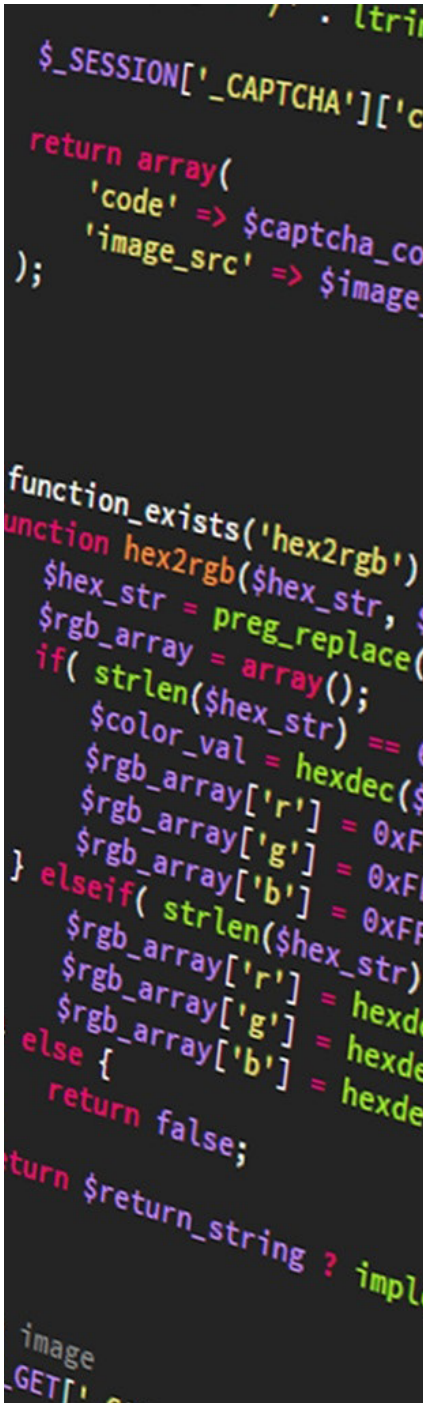
La placa base és un dels components fonamentals de qualsevol ordinador.

La generalitat d'ús dels dispositius informàtics o màquines, per altra banda, és necessària si es considera la diversitat de problemes o àrees en els quals són aplicables, atès que implica un treball addicional quan volem que resolguin problemes particulars. Aquest treball consisteix a dir a la màquina o al dispositiu què és el que ha de fer. En altres paraules, ens hi hem de comunicar, per ordenar-los què és el que han de fer. Malauradament, i malgrat els recents avenços en tècniques de processament de llenguatge natural, encara no parlem el mateix idioma, especialment pel que fa a indicar a una màquina el problema que ha de resoldre. En aquest punt, no parlem d'indicar a un assistent virtual que ens reservi entrades per estrenar una pel·lícula al nostre cinema favorit. Ens referim a transmetre les instruccions que una màquina ha d'executar per fer una determinada tasca.

Així, **programar** es pot entendre com l'acte de convertir la funcionalitat que té una aplicació en instruccions o ordres per a la màquina que, eventualment, les executarà. Per mitjà de la programació ordenem a la màquina com cal comportar.

Un cop tenim clar què és programar una màquina, és convenient definir el concepte de programa, entès com el conjunt d'instruccions i ordres que contenen la funcionalitat desitjada per a l'aplicació que s'està desenvolupant. Cada aplicació que executam a la nostra màquina és un programa amb instruccions que ha d'executar. Instruccions que seran executades de manera seqüencial, amb canvis en el flux d'execució causats per les instruccions o per esdeveniments que es produeixin de manera externa, com per exemple la interacció de l'usuari a través del teclat o el ratolí.

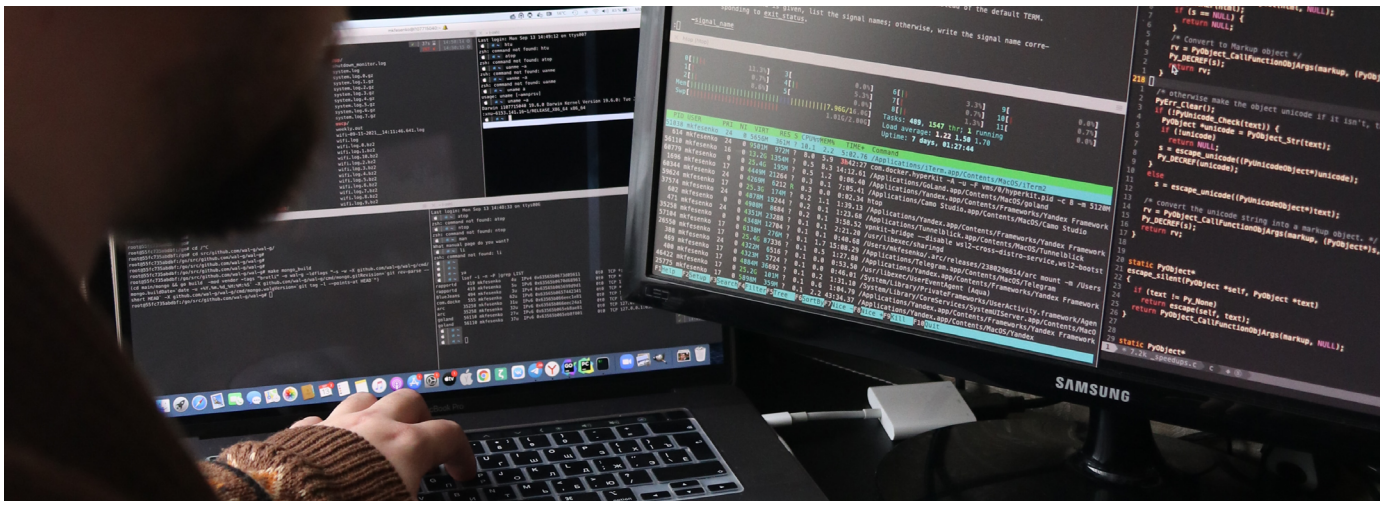
Cal que recordeu que la màquina o ordinador està composta per components físics. Alguns dels més rellevants són la placa base, que és el dispositiu que connecta a tots els elements, processador o CPU, responsable d'executar les instruccions dels programes, memòria RAM, que permet emmagatzemar informació per proporcionar-hi un ràpid accés, disc dur, que ofereix un mecanisme d'emmagatzematge d'informació més permanent, targeta gràfica, que optimitza el tractament d'imatges i vídeo, o la targeta de so, responsable de gestionar els elements sonors. Tot i això, el programa és un conjunt



Fragment de codi font.

d'instruccions que algú haurà escrit i emmagatzemat en un arxiu o fitxer. La pregunta que ens podem fer en aquest punt és com ens podem comunicar o donar instruccions a aquest conjunt d'elements físics electrònics que conformen la màquina? És evident la necessitat d'establir un nexe entre aquests dos mons: 1) el món físic, representat pels components de la màquina, i 2) el món lògic, representat pels programes.

Finalment, resulta interessant que coneguis que la programació s'engloba en allò que anomenem formalment construcció de programari, definit com un procés complex i sofisticat que va més enllà del que comunament coneixem com a programació. En aquest sentit, la construcció de programari es pot dividir en diverses etapes generals: 1) definició del problema, per definir clarament, i de manera precisa, el problema que es vol resoldre, 2) especificació o definició de requisits, que permet generar-ne una descripció detallada sobre allò que el sistema programari ha de fer, 3) planificació del desenvolupament, que té com a objectiu detallar la planificació de la resta de fases de la construcció de programari, 4) disseny, que persegueix organitzar com s'estructurarà el codi del programa, 5) programació o codificació, entesa com l'escriptura d'un conjunt de sentències que persegueixen la generació d'un resultat, 6) proves, que posa el focus a comprovar que el programa creat funciona com cal i, finalment, 7) manteniment, fase relacionada amb l'actualització del codi.



Creació de
continguts digitals

Nivell A1 3.4 Programació

Llenguatges de programació. Definició i evolució

Llenguatges de programació. Definició i evolució

Els circuits que componen els ordinadors treballen amb dos nivells de tensió. Són circuits digitals, i aquests dos nivells s'associen amb els números 0 i 1. Et resulta familiar ara el terme sistema binari? Les accions que pot fer un dispositiu informàtic es representaran a través d'un conjunt de seqüències de 0 i 1.

Si abans hem definit un programa com el conjunt d'instruccions que contenen la funcionalitat desitjada per a l'aplicació que s'està desenvolupant, la recreació d'aquest programa a la memòria principal d'un ordinador consistirà en la traducció d'aquest conjunt d'instruccions en seqüències de 0 i 1. Aquest sistema de codis, directament interpretable per la màquina, s'anomena **llenguatge màquina**.

Cada màquina té el seu llenguatge màquina amb què es pot programar. Aquest llenguatge és específic per a l'arquitectura interna de la màquina. En altres paraules, una màquina només pot reconèixer aquestes instruccions o codis d'operacions programada. Tots els llenguatges màquina disposen d'un conjunt d'instruccions similars: operacions simples, per exemple, per copiar informació, operacions aritmètiques, operacions lògiques, que manegen valors booleans (veritable i fals), i operacions d'entrada/sortida, associades a les connexions de la màquina pel que fa a l'exterior.

Els programes en llenguatge màquina s'executen molt eficientment, ja que es redacten específicament pels circuits que els han d'interpretar i executar. Aquest codi és directament interpretable per la CPU i no requereix transformacions prèvies per ser executat. No obstant això, la programació en llenguatge màquina és un treball difícil i extremadament molest per al programador, i cal que aquest conegui l'arquitectura física de la màquina amb un gran nivell de detall.

Saber-ne més

La programació en llenguatge màquina, en paraules de John Backus, era un art fosc, una matèria arcana, només a l'abast d'uns quants que van començar a considerar membres d'una classe sacerdotal guardiana de certes habilitats i misteris massa complexos per als mortals normals, que s'oposaven a cap canvi revolucionari que pogués fer la programació tan simple que qualsevol pogués fer-la.

Precisament, aquesta dificultat per programar va ser l'objectiu de "democratitzar la tasca de programar els ordinadors", ja que va reduir el període de formació dels programadors i facilitant la tasca de programar màquines. Es pretenia així incrementar la distància entre els programes i el fred acer de la màquina per, alhora, reduir-la entre el codi i els programadors. Per això seria imprescindible, com discutirem més endavant, establir un nivell d'independència entre el programa i l'arquitectura en què s'executarà aquest.

Un dels primers avenços significatius en aquesta direcció va ser l'ús d'una notació simbòlica o mnemònica, emprada per representar cada instrucció o codi d'operació de la màquina. Aquestes claus mnemotècniques eren més fàcils de recordar que els codis numèrics, però, però, requerien que, una vegada establerta la seqüència d'instruccions en mnemònic que solucionaven el problema, fossin traduïdes a llenguatge màquina. A aquest llenguatge se'l va denominar **llenguatge assemblador**.

El següent llistat de codi mostra un exemple de suma de dos números en codi assemblador per a una arquitectura 8086, emmagatzemats a les adreces de memòria 4000 i 4002. El resultat s'emmagatzema a les adreces 5000 i 5002. L'objectiu de mostrar aquest exemple és il·lustrar el salt que hi ha entre dur a terme la programació en llenguatge màquina, directament a través de zeros i uns, i la programació mitjançant una notació amb un nivell semàntic més elevat.

```
2000    MOV    CX, 0000        # Carrega 0000 en el registre CX
2003    MOV    AX, [4000]      # Carrega el número 3000 en l'acumulador AX
2007    MOV    BX, [4002]      # Carrega 3002 en el registre BX
200B    ADD    AX, BX [AX]     # Suma el contingut d'AX i BX
200D    JNC    2010            # Salta a direcció 2010 si la suma no s'neduu una
200F    INC    CX              # Incrementa el contingut de l'acumulador AX
2010    MOV    [5000], AX      # Mou el contingut de AX a la posició 3004
2014    MOV    [5002], CX      # Mou el contingut de CX a la posició 3004
2018    HLT                    # Atura el programa
```

No obstant això, i en existir una correspondència estreta, generalment un per un, entre les claus del llenguatge assemblador i els codis de les operacions de les màquines, la programació continua estant propera a la màquina i continua sent un procés minuciós i complicat. En aquest context, cal crear un llenguatge el més proper possible al programador que permeti expressar les diferents accions que ha de realitzar la màquina. Aquí hi ha el concepte de llenguatge de programació d'alt nivell.

Els llenguatges de programació d'alt nivell se solen associar a la noció d'abstracció. En altres paraules, quan fas servir un llenguatge d'aquest tipus et pots centrar en crear un programa sense preocupar-te dels detalls de la màquina que l'executarà. El principal avantatge d'aquest enfocament és que la programació se simplifica i es fa més entenedora, en part perquè les sentències de programació s'assemblen més al llenguatge humà del que passaria amb un llenguatge de programació de baix nivell, o fins i tot pel que fa al llenguatge assemblador. Com abordarem més endavant, lús dun llenguatge dalt nivell implica la necessitat deines que tradueixin les seves sentències a **llenguatge màquina**.

Finalment, i com podeu imaginar, actualment no existeix un únic llenguatge de programació d'alt nivell. Per contra, hi ha tota una varietat de llenguatges que varien en funció del seu objectiu de disseny i de les seves característiques. Alguns dels més populars en els darrers anys, o fins i tot dècades, són C, Java, Python o JavaScript. La figura 1 mostra els llenguatges de programació d'alt nivell més populars des del 2002.

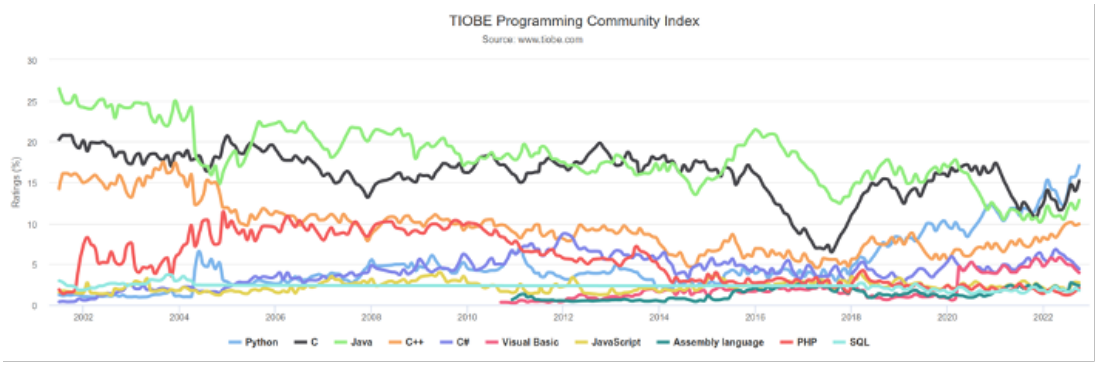


Figura 1. Índex TIOBE, on es mostra l'evolució dels llenguatges d'alt nivell més populars. Data d'octubre del 2022.

Creació de
continguts digitals

Nivell A1 3.4 Programació

Intèrprets vs. compiladors

Intèrprets vs. compiladors

Per facilitar la programació de les màquines es fan servir els llenguatges de programació d'alt nivell, com ara C++ o Python. Aquests llenguatges permeten al programador indicar què és el que volen que faci en un llenguatge proper per a ell, però llunyà del que entén la màquina, que són els llenguatges màquina.

D'aquesta manera sorgeix la necessitat d'algun tipus de traductor o processador de llenguatges capaços de convertir les instruccions donades pel programador en instruccions executables per la màquina.

Sobre aquesta base és possible construir uns programes especials, dissenyats per transformar els programes escrits en llenguatges de programació d'alt nivell, entenedors pels humans, en llenguatge màquina o codi binari, entenedors per les màquines. Aquests programes són els **compiladors i els intèrprets**. Per exemple, l'existència d'un compilador per al llenguatge de programació C++ fa possible que un programa escrit a C++ es tradueixi en llenguatge màquina associat a una arquitectura maquinari determinada. De la mateixa manera, l'existència d'un intèrpret per al llenguatge de programació Python fa possible que un programa escrit a Python es pugui executar sobre la mateixa màquina o arquitectura física.

Si bé els compiladors i els intèrprets persegueixen el mateix objectiu, és a dir, traduir codi escrit en un llenguatge d'alt nivell en llenguatge màquina, hi ha **diferències** substancials entre tots dos. Tres de les més rellevants són les següents:

- Els compiladors transformen el codi d'un programa en codi binari abans d'executar-lo en una màquina determinada. En altres paraules, abans d'executar el programa cal compilar-lo. Aquest procés assegura, abans de fer servir el programa, que el codi ha estat escrit d'acord amb l'especificació del llenguatge. No obstant això, els intèrprets no generen codi màquina, ja que directament les entenen i les converteixen en instruccions executables a la màquina.
- El codi compilat s'executa de manera més ràpida que el codi interpretat. Això implica que hi hagi àmbits on sigui preferible fer servir un llenguatge compilat en lloc utilitzar un llenguatge interpretat. Un exemple seria el món dels videojocs, on el programa, és a dir, el videojoc, ha de

funcionar amb una taxa d'imatges per segon elevada per fer aquesta sensació d'animació, alhora que s'encarrega d'interactuar amb el jugador.

- Un compilador analitzarà tot el codi font d'un programa per trobar errors abans d'executar-lo. Per contra, l'intèrpret ho farà línia a línia.

En aquest context de compiladors i intèrprets, se solen utilitzar els termes llenguatges de *programació compilats i llenguatges de programació interpretats*, depenent de si la generació de codi màquina es realitza a través de compiladors i intèrprets. A l'hora de triar un llenguatge de programació per crear un programa, cal tenir en compte diferents variables. Algunes de les més rellevants són l'experiència de l'equip de treball amb un llenguatge de programació determinat, el rendiment requerit pel programa (els llenguatges interpretats estan associats a uns temps d'execució més alts) o la capacitat del llenguatge perquè el programador creï programes de manera àgil (els llenguatges interpretats solen tenir avantatge respecte als compilats).

Creació de
continguts digitals

Nivell A1 3.4 Programació

Expressions i assignació

Expressions i assignació

Les **expressions** són el mecanisme fonamental de còmput dels programes, i es tracta de la instrucció més bàsica que podem fer servir quan desenvolupem un programa. Les expressions estan formades per **valors** (p. ex., 4 o 546) i **operadors** (p. ex., + o -), i es poden **avaluar** pel computador per obtenir un únic valor o provocar algun efecte desitjat.

Per exemple, podríem construir una expressió que sumi dos números donats utilitzant una expressió com a 2+3. Aquesta expressió estaria formada pels valors 2 i 3 i l'operador de suma (+). L'avaluació d'aquesta expressió resultaria en el valor 5. Així, podem crear expressions de diferents maneres mitjançant la combinació d'operands (números, variables, etc.) i operadors (matemàtics, lògics, etc.).

Vegem a continuació els diferents tipus d'expressions que hi ha a la majoria dels llenguatges de programació.

Expressions d'assignació de variables

Com ja hem vist anteriorment, les variables permeten al programa emmagatzemar valors per treballar-hi posteriorment. En aquest sentit, les variables són ideals per guardar el valor resultant d'avaluar una expressió i així poder encadenar expressions al llarg del programa.

Utilitzant aquesta tècnica podríem aconseguir emmagatzemar, per exemple, el valor 23 en una variable anomenada número utilitzant la següent expressió:

```
nombre = 23
```

Aquest tipus d'expressions s'anomenen **expressions d'assignació de variables** i estan formades per l'operador d'assignació (=) que divideix l'expressió d'assignació de variable a una part esquerra i una part dreta. A la part dreta de l'operador es trobarà l'expressió l'avaluació de la qual volem desar, mentre que a la part esquerra es trobarà la variable que servirà per emmagatzemar el resultat d'aquesta avaluació.

Com apunt addicional, emmagatzemar un valor en una variable per primera vegada abans que s'utilitzi al programa s'anomena Inicialitzar una variable. Això és important perquè una variable pot tenir un valor inesperat si no s'inicialitza, cosa que pot causar errors al programa. A més, inicialitzar una variable assegura que sempre tingui un valor vàlid i conegut, cosa que pot facilitar la correcció d'errors i el manteniment del codi.

Vegem alguns exemples d'aquest tipus d'expressions i el resultat de les vostres avaluacions:

```
1 | nombre_imparell = 23
2 | nombre_parell = 4
3 | resultat = nombre_imparell + nombre_parell
4 | nombre_imparell = resultat
```

A la línia 1 s'inicialitza una variable amb el resultat d'avaluar l'expressió a la dreta de l'operador d'assignació (=), guardant així el valor 23 a la variable nombre_imparell. De la mateixa manera, a la línia 2, s'inicialitza una altra variable (nombre_parell) amb el valor 4. Després, a la línia 3 s'executa una expressió en què s'avalua la suma de les variables creades amb el valor 27, assignant el resultat a la variable resultat. Finalment, a la línia 4 s'assigna el valor de la variable resultat (27) a la variable nombre_imparell, substituint el valor d'aquesta última (23) pel de la primera (27).

⚠ ATENCIÓ

Com s'acaba d'esmentar al text, una variable es pot inicialitzar (*crear*) la primera vegada que se li assigna un valor. Després d'això, aquesta variable pot ser utilitzada en qualsevol expressió posterior amb altres variables i valors. Les assignacions següents que pateixi aquesta variable simplement substituiran el valor que tingués anteriorment pel resultat obtingut després de la nova avaluació, però sense crear una nova variable amb el mateix nom.

A més d'utilitzar expressions d'assignació de variables a partir d'altres expressions definides al codi del programa, hi ha expressions que tenen com a avaluació obtenir dades d'altres fonts de dades com, per exemple, altres dispositius d'entrada. En aquest cas, estaríem parlant d'executar una expressió d'assignació de variables avaluant-ne alguna altra expressió

que sigui capaç de llegir els valors des del mateix teclat del computador, per exemple:

```
1 | salutacio = llegir_teclat()
```

A la línia 1, l'expressió de la dreta de l'operador d'assignació aturarà l'execució del programa fins que l'usuari introdueixi algun text utilitzant el teclat i premeu la tecla [ENTER]. En acabar l'avaluació de l'expressió, el text introduït per teclat serà el resultat que se us assignarà a la variable salutació.

Expressions que provoquen efectes col·laterals

Les expressions que hem vist fins ara poden ser utilitzades per compondre expressions d'assignació de variables, ja que el resultat de la vostra avaluació pot ser emmagatzemat en una variable. No obstant això, hi ha altres expressions l'avaluació de les quals no torna cap resultat, sinó que **provoca un efecte col·lateral** en l'estat de l'ecosistema d'execució (p. ex., intèrpret, sistema operatiu, maquinari,...). Per tant, no podem fer servir aquestes expressions en expressions d'assignació.

Per exemple, una expressió que mostri text per la pantalla de l'ordinador no tornaria cap valor, sinó que utilitzaria aquesta pantalla per mostrar el resultat d'avaluar aquesta expressió:

```
1 | mostrar_pantalla('¡Hola, mon!')
```

En aquest exemple, l'avaluació de l'expressió faria aparèixer per pantalla el text "Hola, món!". Aquest seria el comportament esperat quan s'executa aquesta expressió. No obstant això, podeu veure com el resultat de l'expressió no s'assigna a cap variable, sinó que simplement avalua i provoca l'efecte col·lateral de mostrar text en pantalla.

Expressions aritmètiques

Les expressions aritmètiques ens permeten fer operacions matemàtiques sobre valors numèrics. Els llenguatges de programació solen proporcionar un conjunt d'operadors aritmètics comú que podem fer servir per construir expressions més complexes:

Operador	Operació	Expressió d'exemple	Resultat de l'avaluació d'exemple
**	Exponent	3 ** 3	27
%	Resta de la divisió	55 % 7	6
//	Divisió entera (truncada)	55 // 7	7
/	Divisió amb decimals	55 / 7	7.857...
*	Multiplicació	4 * 5	20
-	Resta	6 - 2	4
+	Suma	1 + 2	3

Per exemple, utilitzant la taula anterior, podríem construir expressions com les següents:

```
1 |  2 + 2 * 4
2 | 15 + 3 * 24 ** 2 % 3
3 | ((15 + 3) * 24) ** (2 % 3)
```

El resultat d'avaluar l'expressió de la línia 2 pot ser confús, per la qual cosa cal conèixer l'ordre en què s'avaluen els operadors aritmètics a les expressions. Normalment, l'ordre de precedència dels operadors és molt similar al dels operadors en l'àmbit de les matemàtiques: primer s'avalua l'operador d'exponent (**); després els operadors de multiplicació (*), divisió sencera (//), divisió amb decimals (/) i resta de la divisió (%), respectant l'ordre d'aparició d'esquerra a dreta; finalment, els operadors de suma (+) i resta (-). En qualsevol cas, és convenient fer servir parèntesis per facilitar la llegibilitat d'expressions més complexes (línia 3).

Expressions alfanumèriques

Les **expressions alfanumèriques** són aquelles compostes per valors textuals i numèrics. En els llenguatges de programació, hi ha algunes operacions que podem realitzar entre cadenes de text i valors numèrics.

D'una banda, podem **concatenar cadenes de text** utilitzant l'operador de suma (+):

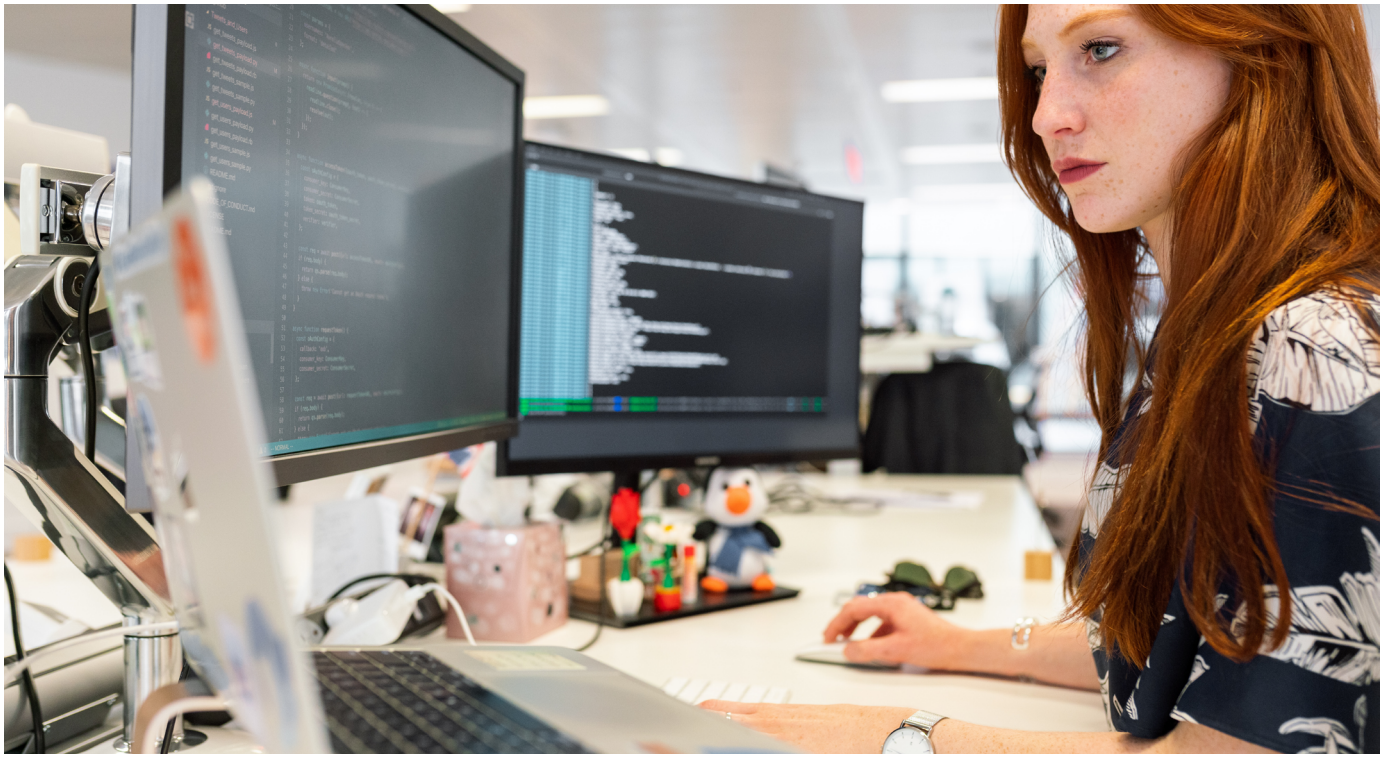
```
1 | 'Hola' + 'Adeu'
```

En aquest exemple, el resultat d'avaluar aquesta expressió seria el text HolaAdeu.

D'altra banda, podem replicar **cadenes de text** tantes vegades com desitgem utilitzant l'operador de multiplicació (*):

```
1 | 'Hola' * 3
```

Això donaria com a resultat el text HolaHolaHola, ja que s'ha replicat el text original 3 vegades.



Expressions relacionals i lògiques

Les **expressions relacionals** produeixen resultats de veritable o fals a partir de la seva avaluació. Aquest tipus d'expressions s'utilitzen per construir sentències condicionals, permetre als programes prendre decisions i alterar-ne el flux d'execució en conseqüència.

Els llenguatges de programació proporcionen un conjunt d'operadors relacionals que podem utilitzar per construir aquest tipus d'expressions, normalment:

Operador	Operació
==	Igual a
!=	No igual a
<	Menor que
>	Major que
<=	Menor que o igual a
>=	Major que o igual a

A continuació, es mostren alguns exemples simples d'expressions relacionals:

```
1 | 'Hola' == 'Hola'
2 | 'Hola' == 'Adeu'
3 | 'Hola' != 'Adeu'
4 | 7 < 8
5 | 7 < 7
6 | 7 <= 7
```

Aquestes expressions s'avaluarien a valors de True (línies 1, 3, 4 i 6) i False (línies 2 i 5).

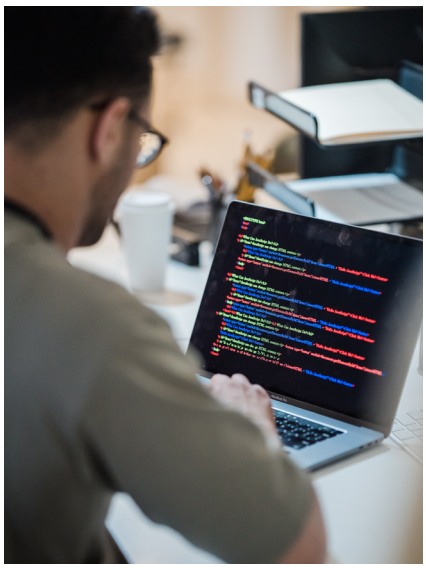
D'altra banda, hi ha una sèrie d'operadors lògics (and, or i not) que podem utilitzar per combinar diverses expressions relacionals i construir així les anomenades **expressions lògiques**:

Expressió	Resultat de l'evaluació
True and True	True
True and False	False
False and True	False
False and False	False
True or True	True
True or False	True
False or True	True
False or False	False
not True	False
not False	True

Per exemple, a continuació es mostren algunes expressions d'aquest tipus:

```
1 | (4 < 5) and (5 < 6)
2 | (4 < 5) and (9 < 6)
3 | (1 == 2) or (2 == 2)
```

A la línia 1 primer s'avaluarien les expressions relacionals entre parèntesis, obtenint una expressió intermèdia tal que True and True. Després, es tornaria a avaluar aquesta expressió obtenint com a resultat final el valor True. L'avaluació de la resta d'expressions de l'exemple es faria de manera semblant, obtenint els resultats False (línia 2) i True (línia 3).

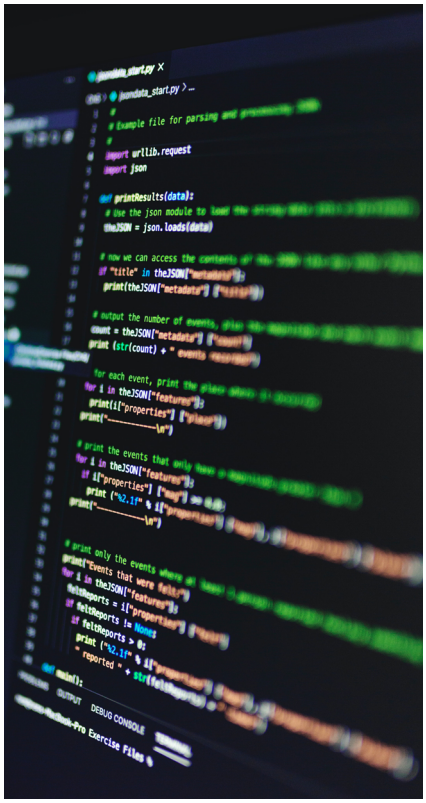


Creació de
continguts digitals

Nivell A1 3.4 Programació

Control de flux d'execució

Control del flux d'execució



El terme *flux d'execució d'un programa* involucra tres conceptes que es relacionen entre si: 1) flux, 2) execució i 3) programa. Tot i que els definirem a continuació, heu de saber que hi ha una relació molt propera, a escala semàntica, entre el flux d'execució d'un programa i la idea de diagrama de flux, prèviament abordat des d'una perspectiva visual a l'hora d'introduir el concepte d'algorisme.

El concepte de *programa*, prèviament discutit, es pot definir com una seqüència d'instruccions que l'ordinador és capaç d'entendre i que tenen un objectiu ben definit. En segon lloc, la paraula *execució* fa referència, precisament, a aquesta capacitat de la màquina per fer que el programa faci el que ha de fer, emprant per això els recursos físics o maquinari disponibles. Per exemple, un d'aquests recursos seria un registre del processador de l'ordinador que s'utilitza per emmagatzemar o guardar el valor d'una variable. Finalment, el concepte *flux* reflecteix aquesta noció de dinamisme de què gaudeixen els programes en execució, i que estableix el camí que el programa recorrerà des que s'arrenca fins que s'acaba. Cada pas d'aquest camí estarà representat per una instrucció de codi. En aquest punt, és important destacar que, encara que un programa estigui compost d'una seqüència d'instruccions, no ha d'executar-les totes i cadascuna d'elles en un ordre determinat. Per contra, el programa executarà un subconjunt de les instruccions esmentades en funció d'una sèrie de condicions.

Aquestes condicions les estableix el *programador*, de manera directa, mitjançant instruccions que modifiquen o controlen el flux d'execució del programa. D'altra banda, el flux d'execució del programa es pot veure alterat depenent del valor puntual de les variables que es fan servir al programa. Com veurem en exemples posteriors, les instruccions que es fan servir per controlar el flux d'un programa solen combinar-se amb l'ús de variables.

Però abans de discutir com es pot controlar el flux d'un programa mitjançant sentències més complexes, vegem un exemple més senzill. La llista següent mostra l'estructura general d'un programa, compost per set instruccions. D'acord amb aquesta estructura, un ordinador que executés aquest programa processaria primer sentència 1. Després, processaria sentència 2. A continuació, processaria la instrucció goto etiqueta. En aquest punt, el lector podria pensar que el flux d'execució del programa continuaria executant, en ordre seqüencial, sentència 4. Tot i això, la instrucció goto, que representa l'exemple més senzill de control de flux en un programa, provocaria que aquest "saltés" a la instrucció etiqueta:. Precisament, la traducció literal de goto podria ser anar a, en el context d'anar a un altre punt del codi del programa. En altres paraules, la instrucció goto provoca un canvi o salt incondicional al flux d'execució del programa, de manera que la següent instrucció a executar, sentència 5, sigui la que està escrita just després de la instrucció etiqueta:

```
sentencia 1
sentencia 2
goto etiqueta
sentencia 3
sentencia 4
etiqueta:
sentencia 5
```

Si bé els salts incondicionals no requereixen cap condició per executar-se, les sentències de control condicional, d'altra banda, es basen en la idea d'avaluar una condició o expressió per disparar possibles canvis en el flux d'execució d'un programa. En aquest sentit, recorda que les sentències condicionals són les estructures més bàsiques que permeten a un computador actuar prenent decisions. A l'exemple de sentència condicional que ja coneixes, "si el gos borda, llavors encendre la il·luminació del jardí", la paraula clau si introdueix una condició a manera de pregunta (lladra el gos?). Si la resposta a aquesta pregunta és afirmativa, la part immediatament posterior a la paraula clau llavors serà l'acció que s'executarà.

⚠ ATENCIÓ

El codi que abusa de l'ús de sentències de salt incondicional pot convertir-se en l'anomenat codi espagueti, ja que aquestes sentències compliquen l'estructura del flux d'execució d'un programa. Aquest terme deriva de la comparació amb un plat d'espaguetis, on el codi mateix es visualitza com un conjunt de fils enredats. L'espagueti nuat representaria la metàfora del flux d'execució del programa. Així, i amb caràcter general, es desaconsella l'ús de sentències de salt incondicional.

La condició a avaluar en una sentència condicional sol estar relacionada amb la consulta del valor d'una variable determinada, o fins i tot de diverses. El llistat de codi següent mostra un fragment escrit en el llenguatge de programació Python que exemplifica, de manera senzilla, aquesta situació. A les línies 1 i 2 es declaren les variables a i b, a les quals se li assignen, respectivament, els valors 13 i 7. A la línia 4 apareix una sentència condicional, representada a Python per la paraula clau if. A continuació, apareix la condició que cal avaluar, a > b. Així, si el valor d'a és més gran que el valor de b, llavors s'executarà la sentència de la línia 5. Si no ho és, és a dir, si el valor d'a és menor o igual al valor de b, llavors s'executarà la sentència de la línia 7. Fixa't com a la línia 6 s'utilitza una altra paraula clau a Python, else, per agrupar les sentències que s'executaran quan l'avaluació de la condició equival a un valor fals.

```
a = 13
b = 7

if a > b:
   print('El valor d'a és més gran que el valor de b')
else:
   print('El valor d'a és menor o igual que el valor de b')
```

Com podeu imaginar, si executéssiu aquest codi el resultat seria el següent:

```
El valor d'a és menor o igual que el valor de b
```

Les sentències condicionals es poden complicar, encadenant les unes amb les altres per així cobrir més situacions.

⚠ ATENCIÓ

El terme utilitzat per agrupar sentències condicionals s'anomena *niadament*. Així, podeu niar o encadenar diverses sentències condicionals per controlar diferents situacions o condicions, associant diferents conjunts d'accions en funció del resultat de l'avaluació de la condició que governa la sentència.

En aquest sentit, el fragment de codi anterior es pot estendre tal com es mostra a continuació. Concentra't en la sentència elif de la línia 6. En essència, el que li estaríem indicant al programa és que si l'avaluació de la condició anterior, és a dir, la de la línia 4, va ser falsa, llavors prova aquesta nova condició, definida a la línia 6. Finalment, i igual que a l'exemple anterior, la sentència else de la línia 8 s'utilitza per indicar l'acció que s'ha d'executar (línia 9) en cas que totes les condicions anteriors s'hagin avaluat a fals.

```
a = 13
b = 7

if a > b:
   print('El valor d'a és més gran que el valor de b')
elif a == b:
   print('El valor d'a és igual al valor de b')
else:
   print('El valor d'a és menor que el valor de b')
```

Hi ha altres llenguatges de programació que agrupen diverses sentències del tipus *if-then-else* en una única sentència. Per exemple, el llenguatge de programació Java proporciona la sentència *switch,* tal com es mostra al següent exemple de codi.

```
int dia = 3;
switch (dia) {
  case 1:
    System.out.println("Dilluns");
    break;
  case 2:
    System.out.println("Dimarts");
    break;
  case 3:
    System.out.println("Dimecres");
    break;
  /* Resta de dies aquí: Dijous, Divendres, Dissabte */
  case 7:
    System.out.println("Diumenge");
    break;
}
```

L'expressió de la sentència *switch* de la línia 2 només s'avalua una vegada. Posteriorment, el valor d'aquesta expressió es compara amb cadascun dels valors de la sentència *switch*, vinculats a les diferents sentències case que la componen. En aquest exemple, es compara el valor de la variable dia amb cadascun dels valors de les sentències case. Així, el valor de dia es comparará primer amb el valor 1, després amb el valor 2, etc. Quan hi hagi una coincidència, és a dir, quan la comparació sigui veritable, llavors s'executa el codi que hi ha a continuació de la sentència case corresponent. En aquest exemple, s'executarà el codi de les línies 10 i 11, de manera que el resultat d'executar aquest programa seria dimecres.

D'altra banda, com ja coneixes, els llenguatges de programació ofereixen sentències per repetir fragments de codi. Normalment, aquesta repetició es durà a terme mentre una determinada condició sigui veritable. Per exemple, i tornant al llenguatge de programació Python, la sentència while permet l'execució d'un conjunt de sentències mentre la condició que la governa sigui veritable. El llistat de codi següent mostra un cas concret.

```
i = 1;

while i <= 5:
    print(i)
    i = i + 1
```

Com pots observar, és relativament fàcil identificar que el codi contingut al bucle while, és a dir, el codi de les línies 4 i 5, s'executarà 5 vegades. Això és equivalent a afirmar que el bucle while farà 5 iteracions. La condició que governa el bucle equival a preguntar-se a cada iteració si el valor de la variable i és menor o igual que 5. La primera vegada que es realitza aquesta comparació, el valor de la variable i és 1. Com 1 és menor o igual que 5, la condició s'avalua a veritable i, per tant, el codi del bucle (línies 4 i 5) s'executa. Aquest codi mostra el valor de la variable i per pantalla i, aquí ve la part important, suma una unitat al contingut de la variable i. Posteriorment, el flux de control del programa torna a la línia 3, avaluant, de

nou, la condició. Ara, el valor de la variable i és 2, que es torna a comparar amb 5. Aquest procés es repetirà fins que la variable i contingui el valor 6. En aquest punt, el bucle while no executarà més iteracions.

Finalment, la immensa majoria dels llenguatges de programació inclouen sentències que permeten escriure bucles amb més flexibilitat. Una de les més clàssiques és la sentència *for*. Típicament, aquest tipus de sentències permetrà incloure més funcionalitat a l'hora de declarar el bucle, com ara l'assignació de valor a variables o fins i tot la mateixa actualització de la variable que governa l'execució del bucle. Així, el programador té més flexibilitat per escriure codi. Discutim un altre exemple a Python.

```
for i in range(1, 6):
    print(i)
```

Aquest fragment de codi produeix exactament el mateix resultat que el fragment anterior, en què es va utilitzar la sentència *while*. En altres paraules, el codi de la línia 2 s'executarà cinc cops (el bucle farà cinc iteracions). A diferència del fragment anterior, la variable i es declara a la mateixa sentència del bucle *for* de la línia 1. A més, i mitjançant la sentència *range*, estem indicant que el bucle iterarà mentre el valor de la variable i es trobi al rang d'1 a 6, és a dir, mentre que el valor de la variable i sigui 1, 2, 3, 4 o 5. Com pots apreciar, hem estat capaços de recrear, amb només 2 línies de codi, la mateixa funcionalitat que al llistat anterior, que contenia 4 línies de codi (ometent la línia en blanc).

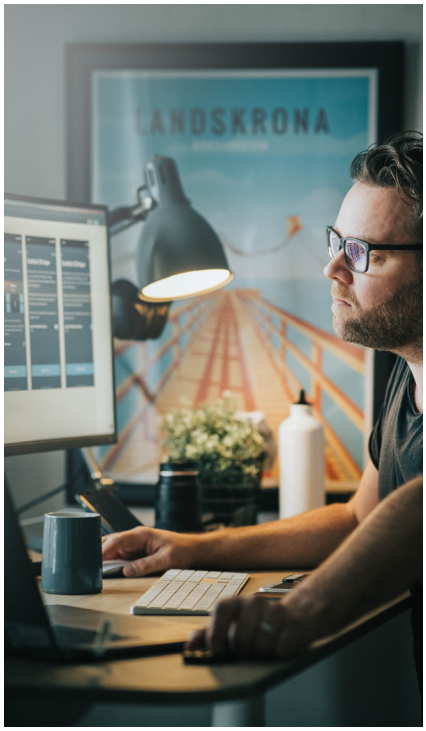
```
var nodeRequire = typeof require === 'function' && require;

function newRequire(name, jumped) {
  if (!cache[name]) {
    if (!modules[name]) {
      // if we cannot find the module within our internal map or
      // cache jump to the current global require ie. the last bundle
      // that was added to the page.
      var currentRequire = typeof parcelRequire === 'function' && parcelRequi
      if (!jumped && currentRequire) {
        return currentRequire(name, true);
```

Creació de
continguts digitals

Nivell A1 3.4 Programació

Guies d'estil

Guies d'estil



Les **guies d'estil** són documents que defineixen un conjunt de convencions de programació, és a dir, un conjunt de bones pràctiques, mètodes de programació i altres regles que garanteixen el desenvolupament de programes de manera consistent. Per exemple, les guies d'estil s'encarreguen de documentar aspectes dels programes relatius a com hauríem de nomenar les nostres variables, o si hauríem de fer servir espais o tabulacions per indentar el codi, entre d'altres.

Tot i que les guies d'estil solen utilitzar-se principalment durant el desenvolupament de projectes de programari en equip, la seva efectivitat en projectes individuals està més que assegurada; és molt més senzill llegir i entendre el codi d'un programa que segueix una guia d'estil de manera consistent que el daltre que no ho fa.

Hi ha multitud de guies d'estils, tantes com llenguatges de programació n'hi ha o fins i tot més d'una per llenguatge. Com a programadors, podem triar la guia d'estil que preferim en funció del suport existent, la documentació i la facilitat d'adopció. L'important és que siguem consistents amb la guia d'estil que triem per al desenvolupament del nostre programa i no barregem diferents guies d'estil per a un mateix projecte.

Entre els beneficis que ens aporten les guies d'estil, podem destacar els següents:

- Milloren la llegibilitat del codi, facilitant-ne la lectura i comprensió.

- Facilita que altres desenvolupadors puguin fer contribucions als nostres programes. De la mateixa manera, ens permeten implicar-nos ràpidament en el desenvolupament d'altres programes que usin guies d'estil amb què ja estem familiaritzats.

- Ens ajuden a centrar-nos en allò que realment importa, és a dir, desenvolupar els nostres programes.

Un cas de guia d'estil seria la **PEP 8** o **Guia d'Estil per a Codi Python**, creada pel mateix autor del llenguatge Python, Guido van Rossum. Aquesta guia d'estil proporciona indicacions sobre l'organització del codi, el nom de variables i altres elements del llenguatge, l'ús de comes o la cometes de cadenes de text, entre d'altres.

Per exemple, PEP 8 aconsella anomenar les variables i funcions en minúscules i separar les diferents paraules amb guions baixos. Alguns exemples vàlids de noms de variables que s'ajusten al que indica aquesta regla serien nombre_d_aprovats o comptador, mentre que alguns exemples incorrectes serien nombreDAprovats o COMPTADOR.

A més de les guies d'estil, també podem millorar la llegibilitat dels nostres programes i la seva sostenibilitat futura documentant aquelles parts del codi més complexes o en què consideram que val la pena posar èmfasi. Afegint comentaris al codi podem fer aquesta documentació directament sobre el mateix codi del programa, de manera que qualsevol programador pugui revisar aquests comentaris durant l'etapa de desenvolupament.

En aquest context, la guia d'estil PEP 8 també dona algunes indicacions sobre l'ús de comentaris al codi com, per exemple, afegir comentaris que realment aportin alguna cosa al lector i no exposin obvietats, o documentar utilitzant comentaris en anglès, entre d'altres:

```
# Improves the accuracy by 5% to reduce possible detection errors
x = x * 1.05
```

DigitAll

Formació en
Competències
Digitals

https://www.digitall.org.es/

DigitAll

Coordinación General

Universidad de Castilla-La Mancha
Carlos González Morcillo
Francisco Parreño Torres

Coordinadores de área

Área 1. Búsqueda y gestión de información y datos

Universidad de Zaragoza
Francisco Javier Fabra Caro

Área 2. Comunicación y colaboración

Universidad de Sevilla
Francisco Javier Fabra Caro
Francisco de Asís Gómez Rodríguez
José Mariano González Romano
Juan Ramón Lacalle Remigio
Julio Cabero Almenara
María Ángeles Borrueco Rosa

Área 3. Creación de contenidos digitales

Universidad de Castilla-La Mancha
David Vallejo Fernández
Javier Alonso Albusac Jiménez
José Jesús Castro Sánchez

Área 4. Seguridad

Universidade da Coruña
Ana M. Peña Cabanas
José Antonio García Naya
Manuel García Torre

Área 5. Resolución de problemas

UNED
Jesús González Boticario

Coordinadores de nivel

Nivel A1

Universidad de Zaragoza
Ana Lucía Esteban Sánchez
Francisco Javier Fabra Caro

Nivel A2

Universidad de Córdoba
Juan Antonio Romero del Castillo
Sebastián Rubio García

Nivel B1

Universidad de Sevilla
Francisco de Asís Gómez Rodríguez
José Mariano González Romano
Juan Ramón Lacalle Remigio
Montserrat Argandoña Bertran

Nivel B2

Universidad de Castilla-La Mancha
María del Carmen Carrión Espinosa
Rafael Casado González
Víctor Manuel Ruiz Penichet

Nivel C1

UNED
Antonio Galisteo del Valle

Nivel C2

UNED
Antonio Galisteo del Valle

Maquetación

Universidad de Salamanca
Fernando De la Prieta Pintado
Pilar Vega Pérez
Sara Alejandra Labrador Martín

Creadores de contenido

Área 1. Búsqueda y gestión de información y datos

1.1 Navegar, buscar y filtrar datos, información y contenidos digitales

Universidad de Huelva
Ana Duarte Hueros (coord.)
Arantxa Vizcaíno Verdú
Carmen González Castillo
Dieter R. Fuentes Cancell
Elisabetta Brandi
José Antonio Alfonso Sánchez
José Ignacio Aguaded
Mónica Bonilla del Río
Odiel Estrada Molina
Tomás de J. Mateo Sanguino (coord.)

1.2 Evaluar datos, información y contenidos digitales

Universidad de Zaragoza
Ana Belén Martínez Martínez
Ana María López Torres
Francisco Javier Fabra Caro
José Antonio Simón Lázaro
Laura Bordonaba Plou
María Sol Arqued Ribes
Raquel Trillo Lado

1.3 Gestión de datos, información y contenidos digitales

Universidad de Zaragoza
Ana Belén Martínez Martínez
Francisco Javier Fabra Caro
Gregorio de Miguel Casado
Sergio Ilarri Artigas

Área 2. Comunicación y colaboración

2.1 Interactuar a través de tecnología digitales

Iseazy

2.2 Compartir a través de tecnologías digitales

Universidad de Sevilla
Alién García Hernández
Daniel Agüera García
Jonatan Castaño Muñoz
José Candón Mena
José Luis Guisado Lizar

2.3 Participación ciudadana a través de las tecnologías digitales

Universidad de Sevilla
Ana Mancera Rueda
Félix Biscarri Triviño
Francisco de Asís Gómez Rodríguez
Jorge Ruiz Morales
José Manuel Sánchez García
Juan Pablo Mora Gutiérrez
Manuel Ortigueira Sánchez
Raúl Gómez Bizcocho

2.4 Colaboración a través de las tecnologías digitales

Universidad de Sevilla
Belén Vega Márquez
David Vila Viñas
Francisco de Asís Gómez Rodríguez
Julio Barroso Osuna
María Puig Gutiérrez
Miguel Ángel Olivero González
Óscar Manuel Gallego Pérez
Paula Marcelo Martínez

2.5 Comportamiento en la red

Universidad de Sevilla
Ana Mancera Rueda
Eva Mateos Núñez
Juan Pablo Mora Gutiérrez
Óscar Manuel Gallego Pérez

2.6 Gestión de la identidad digital

Iseazy

Área 3. Creación de contenidos digitales

3.1 Desarrollo de contenidos

Universidad de Castilla-La Mancha
Carlos Alberto Castillo Sarmiento
Diego Cordero Contreras
Inmaculada Ballesteros Yáñez
José Ramón Rodríguez Rodríguez
Rubén Grande Muñoz

3.2 Integración y reelaboración de contenido digital

Universidad de Castilla-La Mancha
José Ángel Martín Baos
Julio Alberto López Gómez
Ricardo García Ródenas

3.3 Derechos de autor (copyright) y licencias de propiedad intelectual

Universidad de Castilla-La Mancha
Gabriela Raquel Gallicchio Platino
Gerardo Alain Marquet García

3.4 Programación

Universidad de Castilla-La Mancha
Carmen Lacave Rodero
David Vallejo Fernández
Javier Alonso Albusac Jiménez
Jesús Serrano Guerrero
Santiago Sánchez Sobrino
Vanesa Herrera Tirado

Área 4. Seguridad

4.1 Protección de dispositivos

Universidade da Coruña
Antonio Daniel López Rivas
José Manuel Vázquez Naya
Martiño Rivera Dourado
Rubén Pérez Jove

4.2 Protección de datos personales y privacidad

Universidad de Córdoba
Aida Gema de Haro García
Ezequiel Herruzo Gómez
Francisco José Madrid Cuevas
José Manuel Palomares Muñoz
Juan Antonio Romero del Castillo
Manuel Izquierdo Carrasco

4.3 Protección de la salud y del bienestar

Universidade da Coruña
Javier Pereira Loureiro
Laura Nieto Riveiro
Laura Rodríguez Gesto
Manuel Lagos Rodríguez
María Betania Groba González
María del Carmen Miranda Duro
Nereida María Canosa Domínguez
Patricia Concheiro Moscoso
Thais Pousada García

4.4 Protección medioambiental

Universidad de Córdoba
Alberto Membrillo del Pozo
Alicia Jurado López
Luis Sánchez Vázquez
María Victoria Gil Cerezo

Área 5. Resolución de problemas

5.1 Resolución de problemas técnicos

Iseazy

5.2 Identificación de necesidades y respuestas tecnológicas

Iseazy

5.3 Uso creativo de la tecnología digital

Iseazy

5.4 Identificar lagunas en las competencias digitales

Iseazy

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia