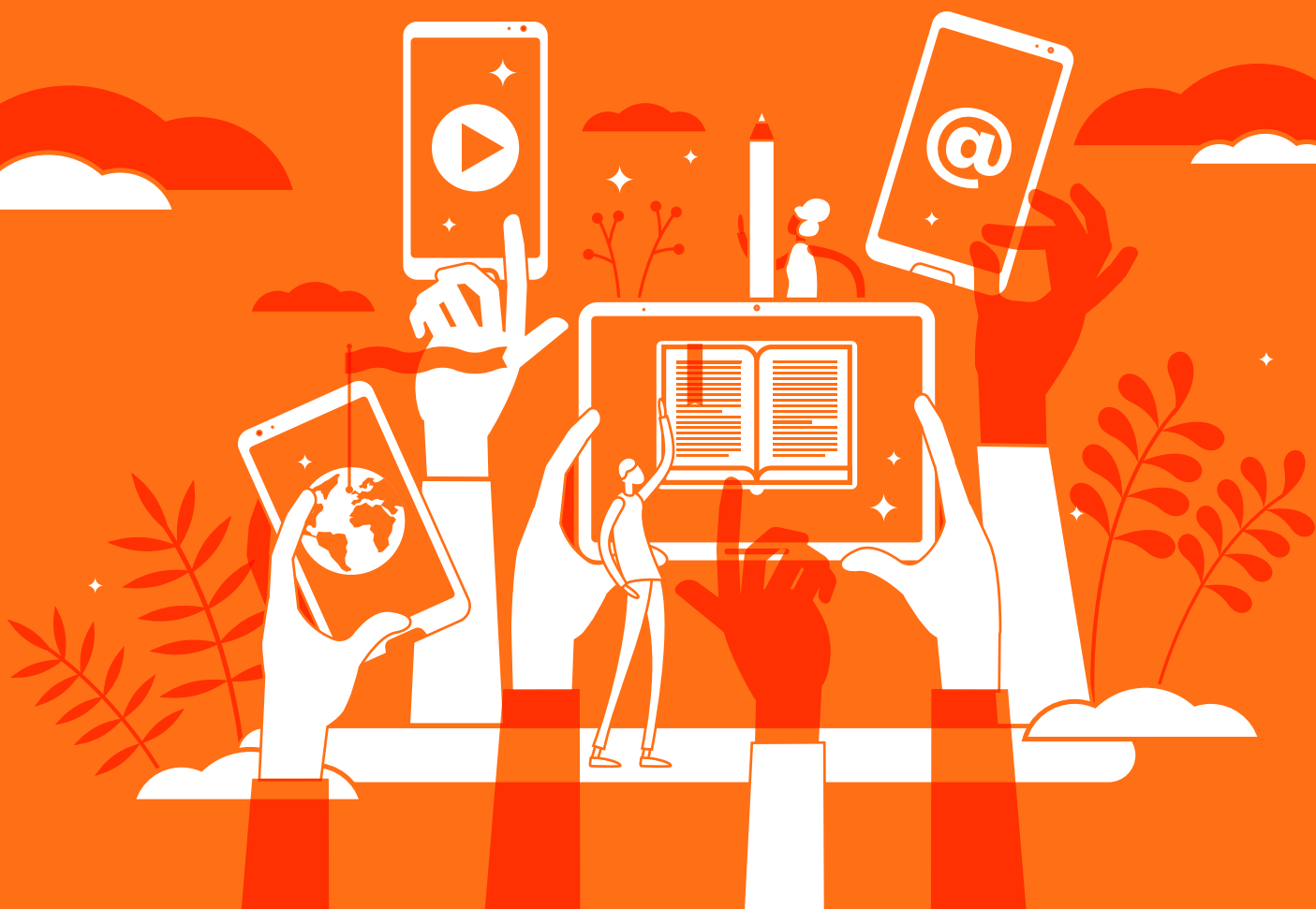




Formació en
Competències
Digitals

3

Creació de continguts digitals



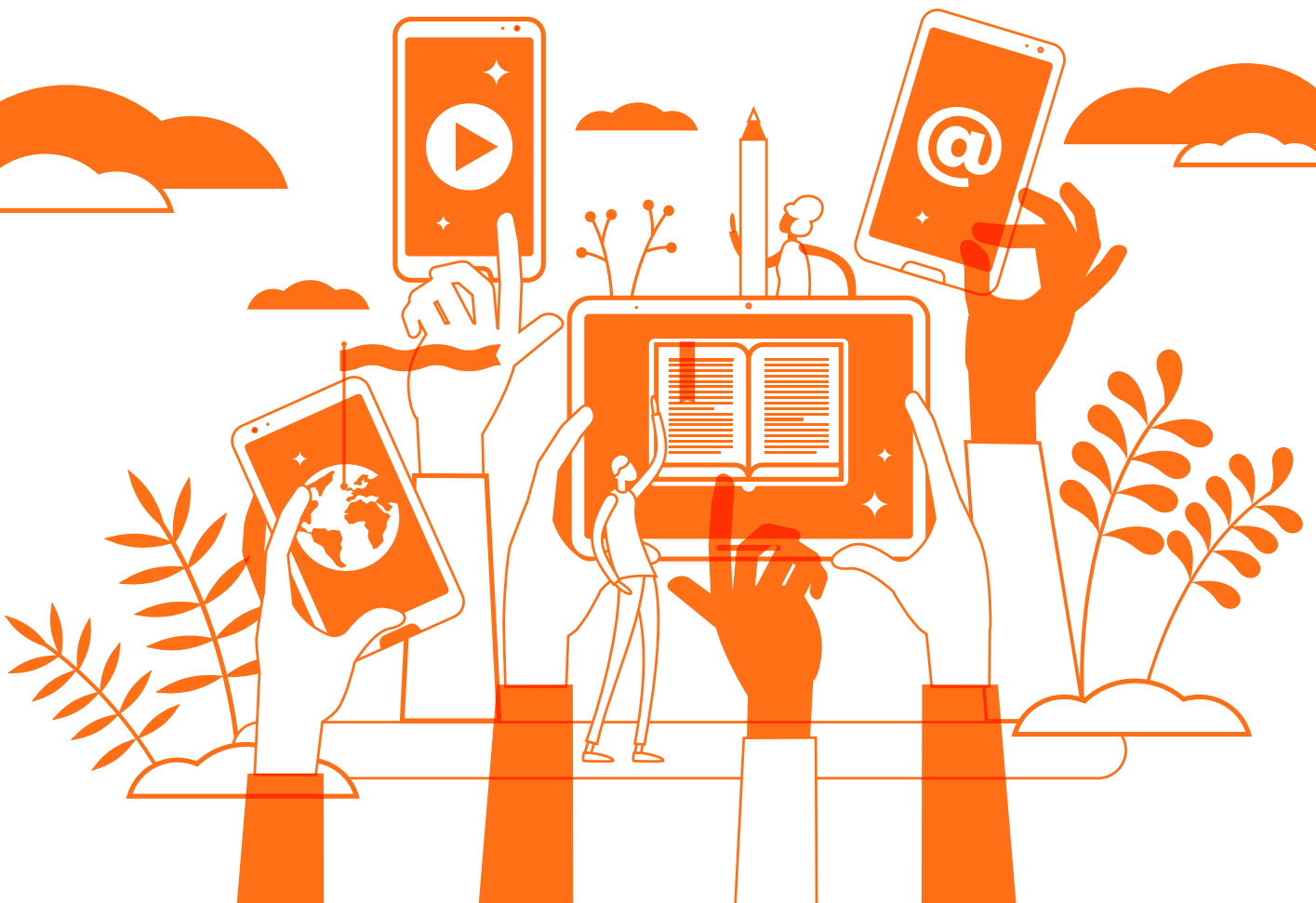


Formació en
competències
digitals



Creació de
contingut
digital

Nivell B2





Creació de continguts digitals

ÍNDEX

3.1. DESENVOLUPAMENT DE CONTINGUT

- [Ús avançat dels processadors de text: connexió amb altres eines ofimàtiques](#)
- [Ús d'eines habituals en processadors de text lliures i en línia](#)
- [Efectes aplicats sobre text](#)
- [Creació d'imatges animades](#)
- [Eines de disseny gràfic lliures i al núvol](#)
- [Compressió d'imatges i vídeo per lots](#)
- [Alternatives per a la creació de llocs web](#)
- [Nivells d'adequació de WCAG](#)

3.2. INTEGRACIÓ I REELABORACIÓ DE CONTINGUT DIGITAL

- [Transicions i animacions avançades en presentacions](#)
- [Remasterització de vídeo i àudio](#)

3.3. DRETS D'AUTOR I LICÈNCIES DE PROPIETAT INTEL·LECTUAL

- [Dotant de copyright una obra](#)
- [Programari lliure, freeware, shareware, adware](#)

3.4. PROGRAMACIÓ

- [Aspectes generals de Python com a llenguatge](#)
- [Tipus de dades a Python](#)
- [Aspectes generals de variables i constants a Python](#)
- [Funciones y paso de parámetros a Python](#)
- [Diccionaris: definició i usos principals](#)
- [Maneig bàsic d'arxius a Python](#)
- [Què és el programari 2.0?](#)





DigitAll

Creació de
continguts digitals

3.1

DESENVOLUPAMENT DE CONTINGUTS





Creació de continguts digitals

Nivell B2 3.1 Desenvolupament de continguts

Ús avançat dels processadors de text: connexió amb altres eines ofimàtiques





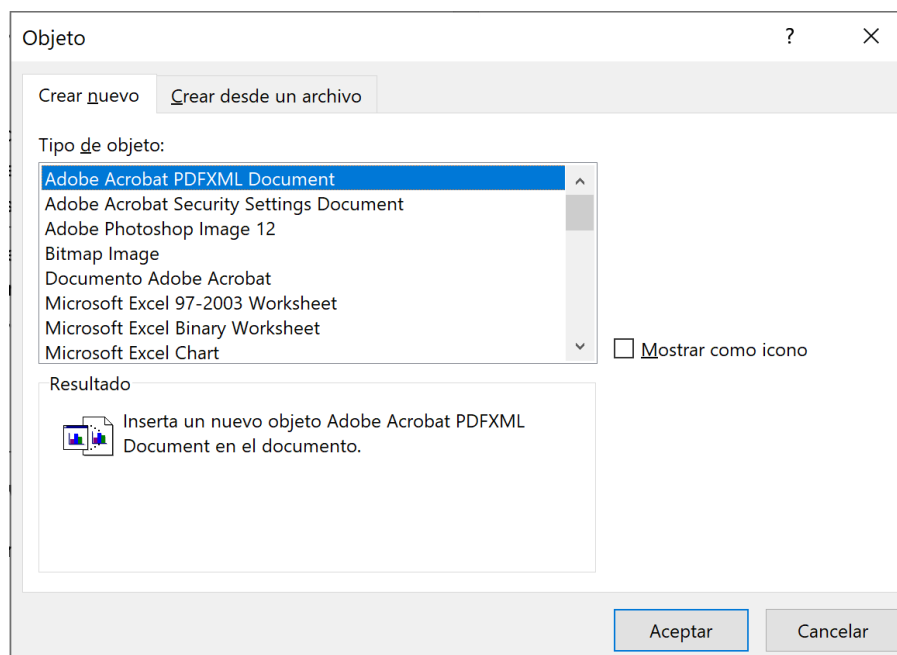
Ús avançat dels processadors de text: connexió amb altres eines ofimàtiques

Eines ofimàtiques connectables

Els processadors de text (nosaltres emprarem Word de Microsoft com a exemple) tenen la capacitat de connectar-se amb altres eines d'ofimàtica per millorar la productivitat i eficiència a la feina. Aquí et presentam algunes alternatives com a fulls de càlcul, presentacions o arxius PDF.

Passos per vincular altres eines a Word

Connectar els arxius és molt senzill. Només s'ha de seleccionar "Objecte" en la pestanya "Inserir".



En el desplegable que s'obre podrem seleccionar eines de diferents tipus. Podrem afegir-les a Word creant-los de nou ("crear nou") o creant-los a partir d'un arxiu ja existent. Una vegada creat el vincle amb l'objecte i ja inserit en el document de text podrem accedir a totes les utilitats de l'eina que hagi estat inserida fent doble clic sobre l'objecte.



Full de càlcul en processador de text

Per connectar un full de càlcul de Microsoft Excel o Google Sheets a Word, també es poden “vincular” les dades d’aquesta manera:

- 1| Obre el document de Word i el full de càlcul d’Excel que vulguis vincular.
- 2| Selecciona al full d’Excel la cel·la o rang de cel·les que vulguis vincular.
- 3| Fes clic dret en la selecció i selecciona l’opció “Copiar”.
- 4| Torna al document de Word i posiciona el cursor en el lloc on vulguis inserir les dades vinculades.
- 5| Fes clic dret i selecciona “Aferrament especial”.
- 6| A la finestra emergent, selecciona “Aferrar vincle” i fes clic a “Full de càlcul”.

Enllestit! Les dades del full d’Excel s’han vinculat al document de Word.

Això permet que es puguin actualitzar les dades del full de càlcul directament des del document de Word i inserir taules, gràfics i altres elements que requereixen càlculs o anàlisis.

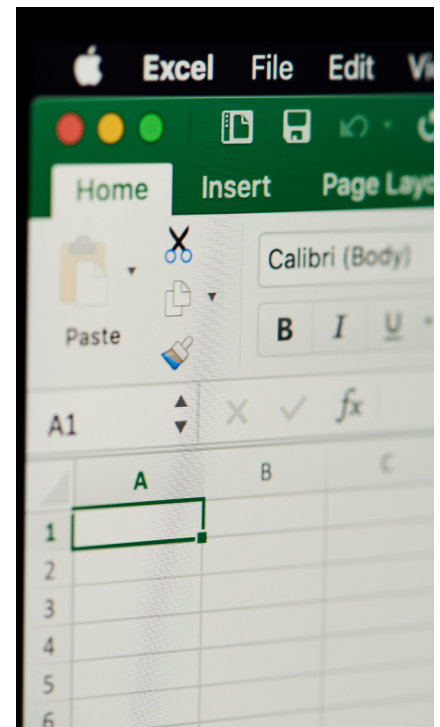
Utilitat: Si ets un professor o estudiant i necessites fer una presentació en Word que inclogui dades numèriques, podries vincular aquestes dades a un full de càlcul d’Excel perquè s’actualitzin automàticament en el document de Word cada vegada que es facin canvis al full d’Excel.

Arxiu PDF en processador de text

Els passos a seguir són els que apareixen en el punt 1.1 d’aquest document.

La utilitat de vincular un PDF en un document de text queda exemplificada en els següents exemples:

- 1| Imagina que estàs escrivint un informe i necessites incloure alguns gràfics o imatges que es troben en un arxiu PDF. En lloc d’haver de copiar i aferrar cada imatge, pots simplement inserir l’arxiu PDF en el teu document de Word i tenir accés a totes les imatges en un sol lloc.





2 En inserir un arxiu PDF en el teu document de Word, pots estar segur que el format es mantindrà igual, ja que l'arxiu PDF és una imatge fixa i s'inclou la totalitat de l'arxiu.

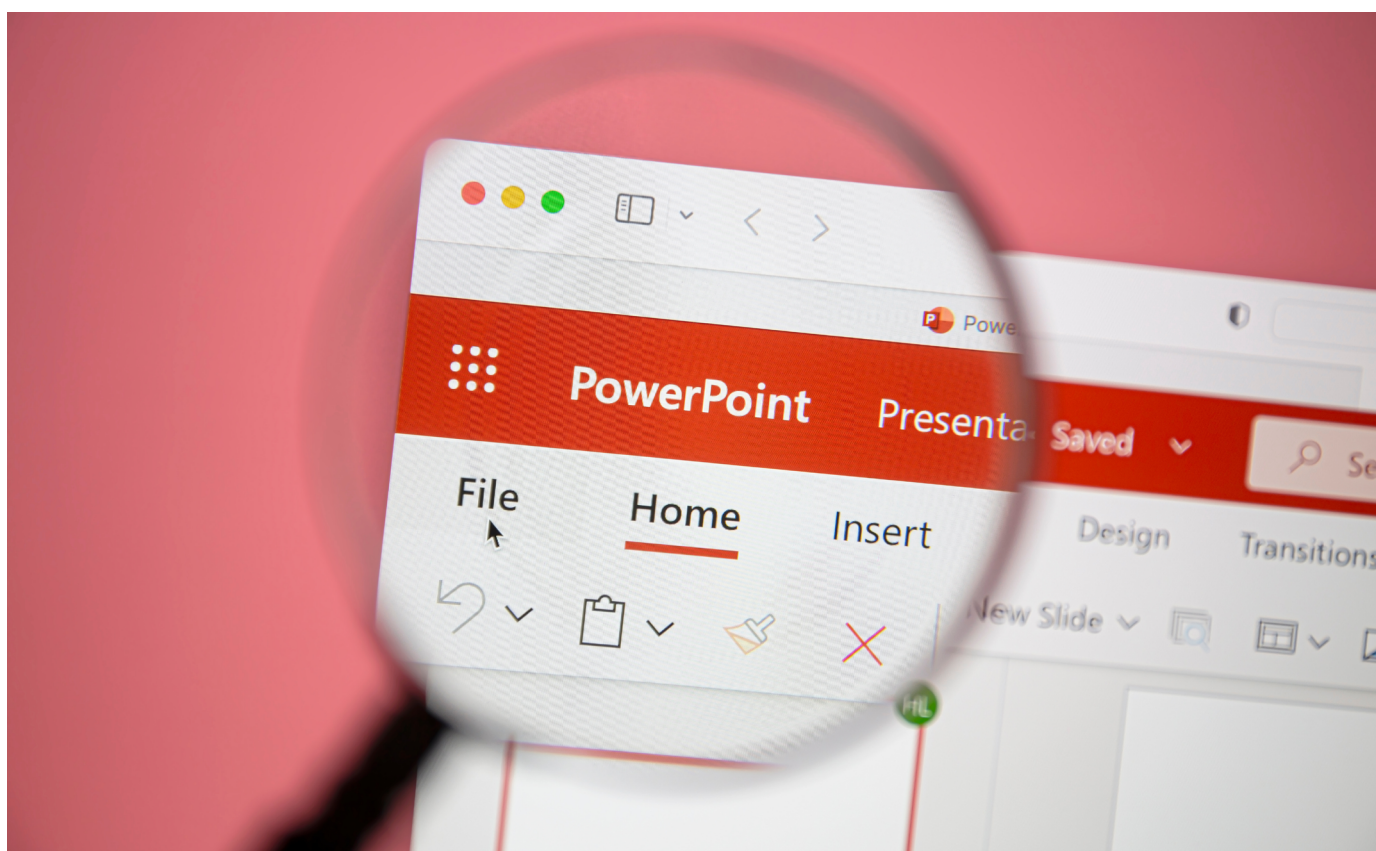
Afegir presentacions al processador de text

Finalment, les eines per elaborar presentacions com Microsoft PowerPoint també poden connectar-se amb processadors de text com Word mitjançant l'opció d'"Inserir objecte", on es pot inserir una presentació completa en un document de Word.

Saber-ne més

Connectar Word amb altres eines, com a fulls de càlcul a Excel o presentacions a PowerPoint, facilita la creació d'informes complets i professionals. Els usuaris poden importar dades i gràfics directament des d'altres aplicacions, la qual cosa agilitza el procés de creació i millora la precisió de la informació presentada. Això estalvia temps i redueix errors. Per a més informació pots visitar el següent web:

e.digitall.org.es/objetos-vinculados





Creació de
continguts digitals

Nivell B2 3.1 Desenvolupament
de continguts

Ús d'eines
habituals
als processadors
de text lliures
i en línia





Ús d'eines habituals als processadors de text lliures i en línia

Moltes de les eines habituals en els processadors de text també estan disponibles en els processadors de text lliures i en línia, cosa que significa que no és necessari comprar programari costós per tenir accés a aquestes funcions. Aquí es presenten alguns exemples d'eines comunes que es poden trobar en els processadors de text en línia i lliures.

Eines comunes

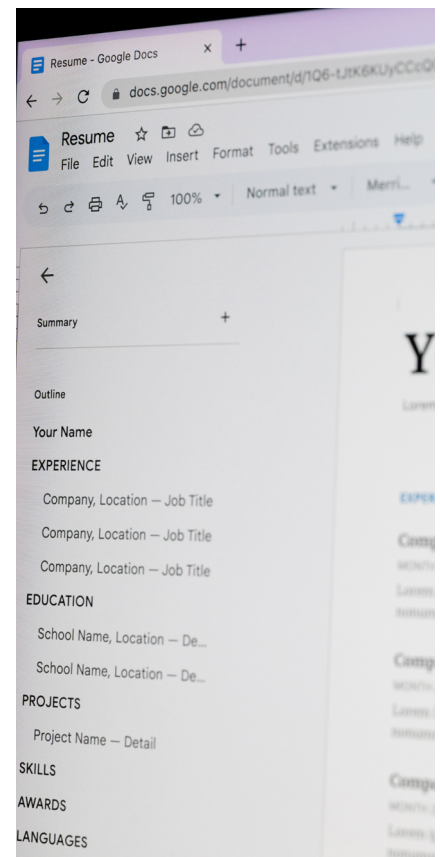
1 | Edició de text: permeten crear, editar i donar format a documents de text. Es poden canviar fonts, grandàries, estils, colors, alineació del text, i també es pot aplicar negreta, cursiva, subratllat i altres efectes de text.

2 | Eines de format: tant els processadors de text lliures com els processadors de text en línia tenen eines de format que permeten ajustar el disseny del document, com a marges, espaiat entre línies i paràgrafs, sagnies, entre altres.

3 | Inserir imatges i altres elements multimèdia: permeten inserir imatges, gràfics, taules i altres elements multimèdia en els documents. Aquests elements es poden editar, ajustar, redimensionar i moure dins del document segons sigui necessari.

4 | Revisió de text: tant els processadors de text lliures com els processadors de text en línia tenen eines de revisió de text que permeten corregir l'ortografia i gramàtica, així com fer seguiment de canvis i comentaris en el document.

5 | Desar i compartir documents: també permeten desar documents en diferents formats, com DOC, DOCX, PDF, RTF, entre altres. A més, es poden compartir documents amb altres usuaris per col·laborar en temps real o per a revisió i comentaris.





6 | Plantilles i estils: tant els processadors de text lliures com els processadors de text en línia ofereixen plantilles i estils predefinits que permeten donar un aspecte professional i consistent als documents. Aquestes plantilles i estils es poden personalitzar segons les necessitats de l'usuari.

7 | Eines de cerca i reemplaçament: ambdós tipus de processadors de text tenen eines de cerca i reemplaçament que permeten trobar i reemplaçar paraules o frases en el document, la qual cosa facilita l'edició i correcció de documents llargs.

8 | Autodesat i sincronització al núvol: els processadors de text en línia solen tenir l'avantatge de la sincronització en el núvol, la qual cosa permet que els documents es desin automàticament en línia i es puguin accedir des de diferents dispositius. Això facilita el treball col·laboratiu i l'accés als documents des de qualsevol lloc i en qualsevol moment.

Processadors lliures i en línia

Alguns exemples de processadors de text lliures són LibreOffice Writer i Apache OpenOffice Writer, que són part de les suites de codi obert i es poden descarregar i emprar de manera gratuïta en computadores locals. Alguns exemples de processadors de text en línia són Google Docs, Microsoft Office En línia, que són aplicacions web que es poden accedir a través d'un navegador web i ofereixen eines de processament de text en el núvol.

A la següent taula es troben alguns programes lliures d'edició de text i la pàgina web on localitzar-los.

Alternatives de processadors de text

Programa	Compatibilitat amb Office	Web
OpenOffice Writer	visualitzar	openoffice.org
LibreOffice Writer	visualitzar / modificar documents	es.libreoffice.org
Google Docs	visualitzar / modificar documents	docs.google.com
WPS Office Writer	visualitzar / modificar documents	wps.com/office



i *Saber-ne més*

Explora totes les capacitats de les eines de processament de text en línia. No només permeten l'edició i el format de text, sinó també l'emmagatzematge al núvol, la col·laboració en temps real, i la integració amb altres aplicacions útils.



En aquesta pàgina web pots trobar 12 alternatives gratuïtes a Microsoft Word.

e.digitall.org.es/freeword-processors



Creació de
continguts digitals

Nivell B2 3.1 Desenvolupament
de continguts

Efectes aplicats sobre text





Efectes aplicats sobre text

El text en disseny gràfic no és simplement una qüestió d'escriure paraules en un llenç, pot ser una eina poderosa i dinàmica que transforma el treball i el fa més atractiu.

Les eines de disseny gràfic, com Adobe Photoshop, GIMP o Illustrator, ens permeten treballar amb efectes de text sorprenents, que van més enllà del que un processador de text estàndard, com Microsoft Word, pot fer. Encara que Word ofereix una gamma bàsica d'efectes de text, les eines de disseny gràfic ofereixen una major profunditat i flexibilitat.

A continuació, es descriuen algunes de les opcions de disseny de text i les seves principals utilitats.

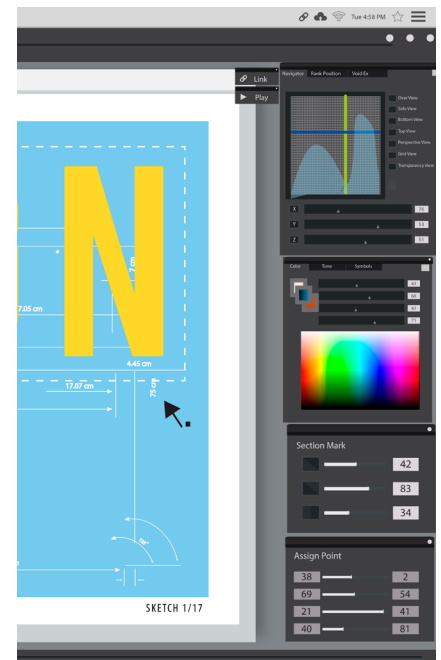
Definicions d'alguns efectes comuns

1 | Bisell i relleu

El bisell i relleu són efectes que li donen al text una aparença 3D. Aquests efectes poden fer que el text sembli que està sobresortint o enfonsant-se en la pàgina. Els ajustaments permeten controlar la profunditat del bisell i la direcció de la llum que crea l'ombra i el relleu. El bisell crea una transició entre el text i el fons, generalment fent que les vores del text semblin aixecar-se per sobre del fons. El relleu, d'altra banda, és similar al bisell, però crea una aparença de profunditat des de dins del text cap a fora, la qual cosa pot fer l'efecte que el text està tallat o pressionat en el fons.

2 | Ombra paral·lela

També coneguda com a ombra de caiguda o llançada, l'ombra paral·lela afegeix una ombra darrere del text que fa l'efecte que el text està surant damunt del fons. Es pot ajustar la distància, l'angle i la difuminació de l'ombra per crear diferents efectes.





3 | Il·luminació interior i exterior (resplendor)

Aquests dos efectes són essencials per a donar al text una sensació de textura i dimensionalitat. Manipulen la il·luminació del text per fer la sensació que hi ha una font de llum en l'escena. La il·luminació interior fa l'efecte que la llum ve des del centre del text, mentre que la il·luminació exterior crea l'aparença que la llum ve des de fora del text. Aquests efectes són útils per a crear text metàl·lic o de neó.

4 | Degradat (superposició de degradat)

Aquest efecte s'utilitza per crear una transició suau d'un color a un altre dins del text. Pot ser utilitzat per agregar profunditat, per a crear un efecte de llum i ombra, o simplement per fer que el text sigui més interessant i atractiu visualment.

Exemple a Adobe Photoshop i GIMP

A continuació, posarem un text pla en un llenç i modificarem cadascuna de les eines explicades a dalt usant el desplegable d'Estil de capa d'Adobe Photoshop (Imatge 1 - a la dreta).

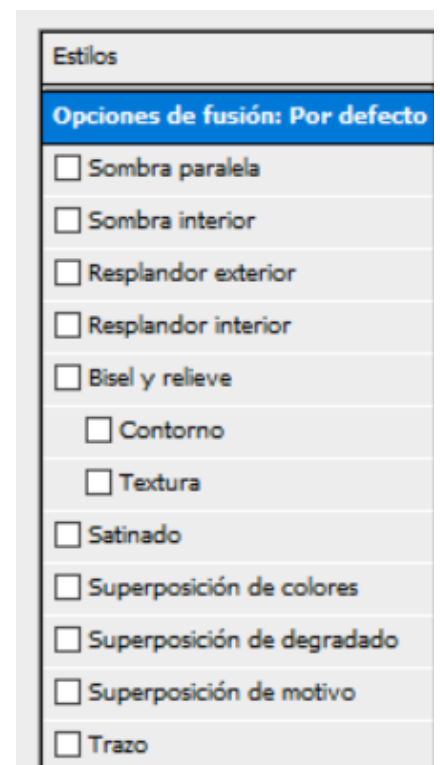
D'aquesta manera passarem de la Imatge 2 a la 3 (a baix) emprant: ombra paral·lela normal amb una opacitat de 77% i angle de 150°, lluentor interior amb mode de fusió trama i opacitat del 75%, un bisell amb biaix interior i profunditat de 300% i una superposició de degradat del 42%.

Lorem ipsum

Imatge 2. Text inicial.

Lorem ipsum

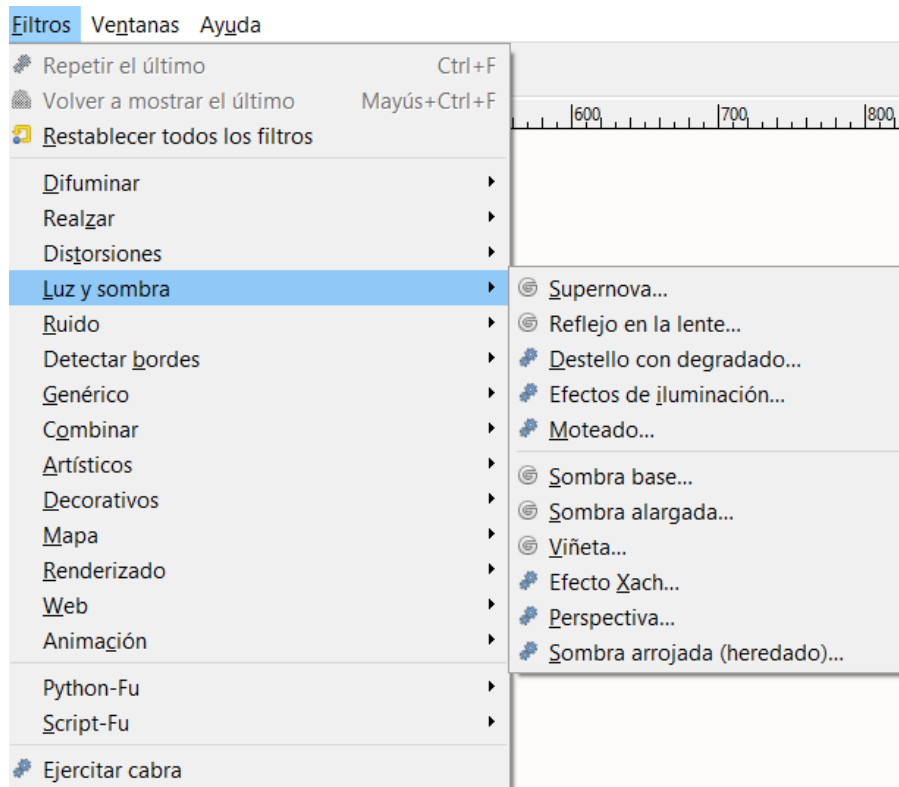
Imatge 3. Text editat.



Imatge 1. Estils de capes.



Per fer el mateix en GIMP emprarem la pestanya de Filtres, amb la qual podrem afegir diferents característiques al text com són la il·luminació, ombra o textures.



Imatge 4. Pestanya filtres.

Saber-ne més

L'experimentació és clau. Les eines de disseny gràfic solen tenir opcions preestablertes per a aquests efectes, però també permeten un gran grau de personalització. És important recordar que a vegades menys significa més. Els efectes de text han d'emprar-se per millorar el missatge que s'està transmetent i NO com a distracció.

És recomanable visitar la pàgina oficial d'**Adobe** (adobe.com), o en **GIMP** (docs.gimp.org) on es poden trobar tutorials i guies pas a pas molt útils.



Creació de
continguts digitals

Nivell B2 3.1 Desenvolupament
de continguts

Creació d'imatges animades





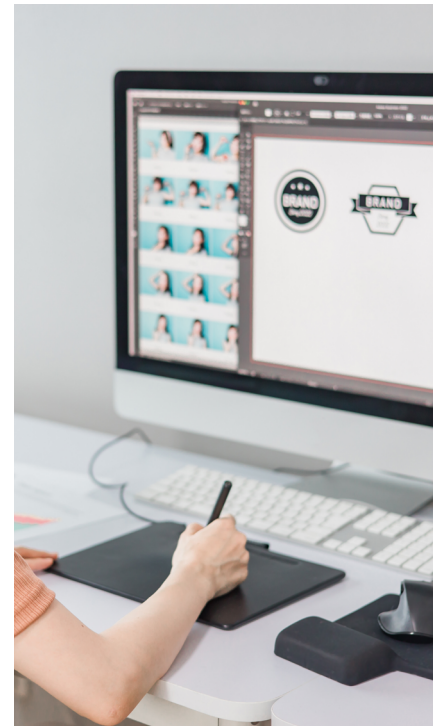
Creació d'imatges animades

La creació d'imatges animades és una disciplina que requereix pràctica i paciència. No obstant això, amb les eines adequades i el coneixement correcte, pots crear animacions impressionants que captaran l'atenció del teu públic.

A continuació, proposem un pas a pas de com fer aquestes imatges.

Elegir el programari adequat

Una de les funcions més versàtils i útils de GIMP és la seva capacitat per generar imatges en format GIF o PNG animat. Aquests formats són àmpliament utilitzats en la web i altres mitjans digitals a causa de la seva versatilitat i la capacitat de suportar animació. En aquest informe, descriurem pas a pas com generar aquest tipus d'imatges utilitzant GIMP.



Procediment

Creació de capes per a l'animació

Cada quadre de la teva animació correspondrà a una capa en GIMP. Per crear una nova capa, ves al menú 'Capa', després a 'Nova Capa'. S'obrirà una finestra on podràs assignar-li un nom a la capa (per exemple, "Quadre1"), definir la grandària (si és diferent del de la imatge original) i el color de fons.

Repeteix aquest procés per a cada quadre de la teva animació. Recorda que l'ordre de les capes en el panell de capes (generalment a la dreta del llenç) determinarà l'ordre dels quadres en l'animació. La capa superior correspon al primer quadre, i la capa inferior a l'últim quadre. Cada quadre de la teva animació serà una capa separada en GIMP.

Fes clic a 'Capa' > 'Nova capa' per començar a crear els teus quadres.

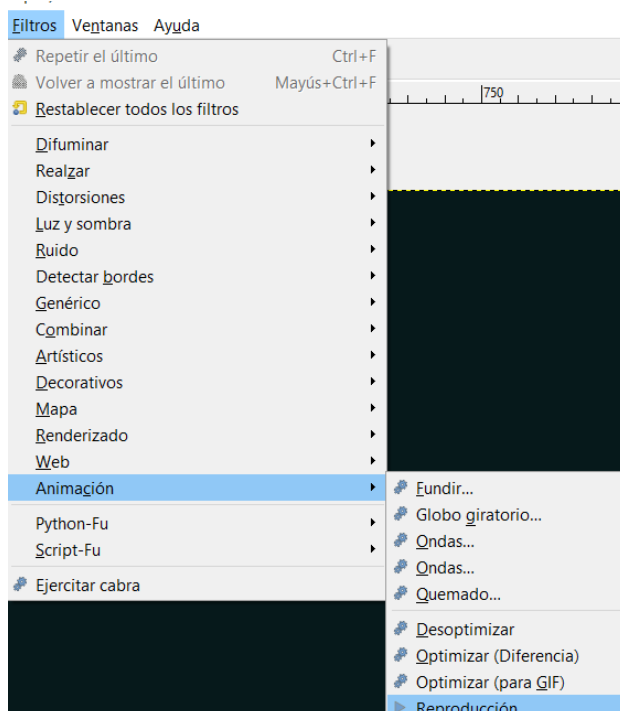


Creació de capes per a l'animació

Amb les capes creades, podeu començar a dissenyar cada quadre de la vostra animació. Selecciona la capa que vols editar al panell de capes i utilitza les eines de GIMP per crear el teu disseny. Per exemple, si voleu dibuixar una línia al primer quadre, has de seleccionar la capa "Quadre1", triar l'eina 'Llapis' del panell d'eines, triar el color i dibuixar la teva línia. Pots incloure eines de dibuix, farciment, text i més.

Previsualitzar l'animació

Una vegada que hagi acabat de dissenyar totes les teves capes, pots previsualitzar la teva animació. Ves al menú 'Filtros', després a 'Animació' i selecciona 'Reproduir'. S'obrirà una finestra de previsualització on podràs veure com es veu la teva animació. Si necessites fer algun ajustament, tanca la finestra de previsualització i torna a les teves capes per fer les modificacions necessàries.





Exportar com GIF o PNG animat

Quan estiguis satisfet amb la teva animació, és hora d'exportar-la. Ves al menú 'Arxiu' i selecciona 'Exportar com'. En la finestra de diàleg que s'obre, tria el format que desitges (GIF o PNG), assigna un nom al teu arxiu i fes clic a 'Exportar'. En el següent quadre de diàleg, assegura't de marcar la casella 'Com a animació'. També pots ajustar altres paràmetres com el temps de retard entre els quadres i si l'animació es repetirà indefinidament. Una vegada que hakis acabat, fes clic a 'Exportar'.



Saber-ne més

L'animació és un art, i com tot art, hi ha uns certs principis que s'han de seguir per crear treballs efectius i atractius. Els 12 principis de l'animació, establerts pels animadors de Disney, Ollie Johnston i Frank Thomas, són un excel·lent punt de partida i són considerats la "Bíblia" de l'animació.

Si vols més detalls et recomanem veure els tutorials del mateix programa.

gimp.org/es/tutoriales/animaciones



Creació de
continguts digitals

Nivell B2 3.1 Desenvolupament
de continguts

Eines de disseny gràfic lliures i al núvol





Eines de disseny gràfic lliures i al núvol

Introducció al disseny d'imatges

El disseny d'una imatge és un tema central en moltes situacions quotidianes o laborals que ens permetrà millorar des de les nostres mateixes fotos, si som aficionats a la fotografia, fins a aconseguir, a través del disseny gràfic, un currículum més atractiu que ens pugui diferenciar d'altres candidats per a accedir a un lloc de treball.

Encara que per fer un bon disseny gràfic sempre podrem recórrer a un professional, existeixen en l'actualitat una sèrie d'eines que ens facilitaran bastant la tasca a l'hora de dissenyar imatges. Aquestes eines, en general, tenen en comú que són capaces de proporcionar a l'usuari una sèrie d'idees de disseny predefinides, en forma de plantilla, i permeten editar-les de manera senzilla, la qual cosa facilita molt el disseny de la nostra imatge fins i tot si només tenim nocions bàsiques de disseny gràfic.

En aquest document dividirem aquestes eines en dos grans blocs, les lliures que podem instal·lar en el nostre equip i les que estan disponibles al web.

Eines lliures per al disseny d'imatges 2D

Les eines lliures per al disseny d'imatges 2D ofereixen una alternativa assequible, personalitzable i d'alta qualitat a les eines de programari propietari, la qual cosa les fa una opció popular per a dissenyadors i artistes.

Entre els grans avantatges d'aquestes eines ens trobam que són gratuïtes per descarregar i utilitzar, i que són de codi obert, la qual cosa permet que els usuaris treballin en comunitat proposant millores o noves funcionalitats.

En aquest camp, ja hem treballat amb les dues segurament més emprades:

- **GIMP:** un editor d'imatges amb una àmplia gamma d'eines per a l'edició de fotos, la creació de gràfics i la il·lustració digital.





- **Inkscape:** un programari de gràfics vectorials que s'utilitza per crear il·lustracions, gràfics, logotips i dissenys complexos.

Les seves alternatives més emprades entre el programari privatiu són Adobe Photoshop i Adobe Illustrator, respectivament.

Eines per treballar al núvol

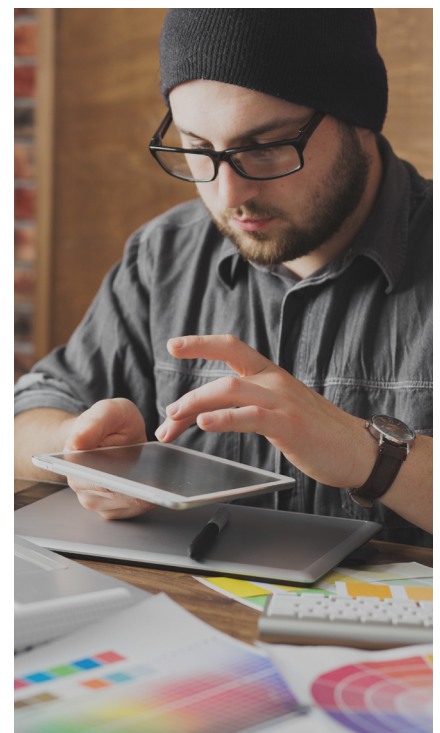
Al núvol disposam de multitud d'eines per treballar el disseny d'imatges. Encara que cadascuna té les seves particularitats, podem descriure algunes característiques generals a continuació:

- **Plantilles:** compten amb una àmplia varietat de plantilles predissenyades per crear dissenys en diferents formats, com a publicacions en xarxes socials, presentacions, targetes de visita, entre altres.
- **Biblioteca de recursos:** ofereixen una àmplia biblioteca de recursos, com a imatges, icones i fonts, per personalitzar i millorar els dissenys.
- **Editor fàcil d'emprar:** són intuïtius i fàcils d'usar, amb eines simples per afegir i editar elements, ajustar colors, canviar fonts i més.
- **Col·laboració:** permeten que els usuaris comparteixin i col·laborin en projectes, la qual cosa és ideal per a equips de disseny o per a treballs en grup.
- **Emmagatzematge al núvol:** emmagatzemen els dissenys al núvol, la qual cosa permet l'accés i l'edició des de qualsevol dispositiu amb una connexió a Internet.
- **Multitud de formats de sortida:** permeten exportar els nostres dissenys a multitud de formats d'imatge. Alguns permeten exportar a altres formats com PDF o generar fins i tot presentacions o vídeos curts.

De totes les disponibles, segurament Canva és una de les més emprades, així que descriurem les seves característiques principals breument.

NOTA

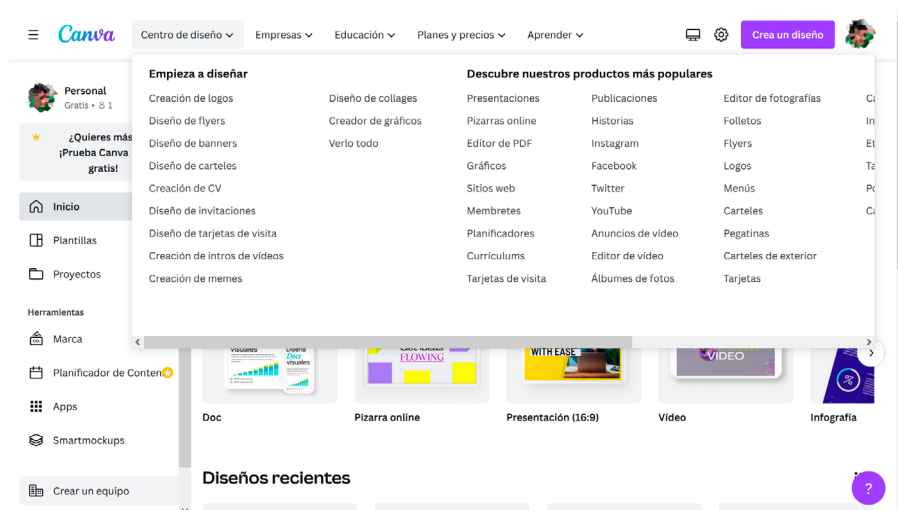
Recorda que, a més d'aquestes eines, tens disponibles bancs d'imatges lliures, com Pixabay, que posen a la teva disposició multitud d'imatges de qualitat.





Disseny d'imatges amb Canva

Es tracta d'una de les eines de disseny més populars. No només ens permetrà dissenyar imatges de manera senzilla, sinó que, a més, com veiem en la imatge, té una àmplia biblioteca amb plantilles que ens ajudarà a crear la nostra imatge des de zero. Així, Canva ens ofereix multitud de plantilles agrupades en temàtiques (per exemple, xarxes socials) que al seu torn se subdivideixen en altres més concretes (per exemple, històries d'Instagram). Finalment, també ens facilita, en comptes de seleccionar una temàtica, seleccionar el tipus de document que volem crear (per exemple, una infografia) d'un ampli ventall i, posteriorment, ens ofereix plantilles per a editar de manera senzilla aquest document proporcionant-nos, per tant, una idea amb una estètica agradable. A més, disposa de diversos plans de pagament que milloren les característiques del compte gratuït ampliant les funcionalitats ja existents que poden resultar interessants en l'àmbit professional. Per exemple, el compte gratuït proporciona 5 GB d'emmagatzematge, mentre que la professional incrementa l'espai disponible fins a 1 TB.



Detall de l'eina Canva en la qual es mostren les plantilles ordenades per blocs dins del Centre de disseny.



Conclusions

En resum, aquestes eines per al disseny d'imatges són populars a causa de la seva facilitat d'ús, flexibilitat i accessibilitat.

Particularment, les eines en línia per al disseny d'imatges en el núvol tenen l'avantatge afegit que els dissenyadors poden treballar en els seus dissenys des de qualsevol lloc amb una connexió a Internet i també poden compartir i col·laborar en projectes amb altres dissenyadors.

Encara que pugui semblar poc rellevant, la col·laboració en línia és una qualitat importantíssima d'aquestes eines, ja que permet als membres d'un equip que treballen des de diferents localitzacions col·laborar en un projecte a temps real, la qual cosa millora l'eficiència, ja que els membres de l'equip poden participar des de qualsevol lloc amb una connexió a Internet, i la qualitat del treball, pel fet que els membres de l'equip poden aportar diferents habilitats i perspectives

Saber-ne més

Pots animar-te a fer els teus primers dissenys al núvol. Nosaltres et recomanem que empris Canva perquè compleix amb totes les característiques que li permeten ser una eina d'èxit en aquest camp i, en ser molt emprada, és senzill trobar informació i idees noves per a provar. Però et deixem per aquí cinc alternatives que també són molt interessants:

- **Adobe Express:** adobe.com/es/express
- **Desygner:** desygner.com/es
- **Piktochart:** piktochart.com
- **Visme:** visme.co/es
- **Vista Create:** create.vista.com/es



Creació de
continguts digitals

Nivell B2 3.1 Desenvolupament
de continguts

Compressió d'imatges i vídeo per lots



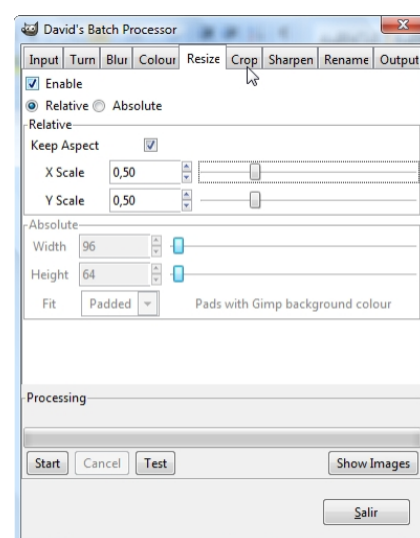


Compressió d'imatges i vídeo per lots

Compressió d'imatges i vídeo per lots

En cas de voler comprimir diverses imatges, és convenient fer aquesta tasca de manera simultània per optimitzar el temps i la feina. Una manera de fer-ho és comprimint per lots. Aquesta tasca es pot fer mitjançant els següents passos:

- 1 | Organitzar les imatges:** crear una carpeta o directori en l'ordinador i col·locar totes les imatges que s'han de comprimir en aquest directori. Cal assegurar-se que les imatges estiguin en un format compatible amb la compressió, com JPEG o PNG.
- 2 | Triar una eina de compressió per lots:** existeixen diverses eines disponibles que permeten comprimir imatges en lots de manera ràpida i senzilla. Algunes opcions populars són:
 - **Adobe Photoshop:** si es disposa d'accés a Adobe Photoshop, es pot utilitzar la seva funció "Processador d'imatges" per comprimir imatges en lots. Simplement creant una acció que faci la compressió i després aplicant-la a totes les imatges en el directori.
 - **GIMP:** ja s'ha parlat i treballat en aquest curs amb aquesta alternativa de programari lliure d'edició d'imatges gratuït i de codi obert. Es pot utilitzar la funció "Processador de lots" per comprimir les imatges desitjades. És necessari configurar el processador per aplicar la compressió a totes les imatges en el directori seleccionat.
 - **Utilitats en línia:** existeixen nombroses eines en línia que permeten comprimir imatges per lots de manera gratuïta. Algunes opcions comunament utilitzades inclouen **TinyPNG** (tinypng.com), **Compressor.io** (compressor.io) i **Optimizilla** (imagecompressor.com/es). Aquestes eines permeten carregar i comprimir múltiples imatges al mateix temps.



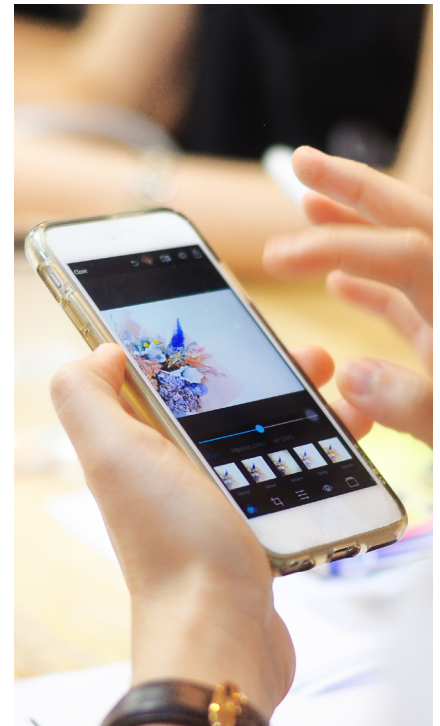
Processament per lots a GIMP. Permet redimensionar i comprimir un conjunt d'imatges i exportar-la en una sola acció.



3 | Configurar la compressió: una vegada seleccionada l'eina a utilitzar, és probable que calgui configurar els ajustaments de compressió. Es pot triar el nivell de compressió desitjat, tal que, com ja s'ha explicat en aquest curs, més nivell de compressió redueix més la grandària de l'arxiu, però també pot afectar la qualitat de la imatge. És necessari trobar un equilibri entre la qualitat i la grandària d'arxiu que sigui adequat per a les necessitats.

4 | Iniciar el procés de compressió: una vegada configurats els ajustaments de compressió, cal iniciar el procés de compressió per lots. Això pot implicar carregar les imatges en l'eina en línia o executar el processador de lots en el programari d'edició d'imatges. L'eina processarà totes les imatges en el directori seleccionat i les comprimirà segons els ajustament.

5 | Desar les imatges comprimides: una vegada que es completi el procés de compressió, l'eina proporcionarà les imatges comprimides. Serà necessari desar-les en una ubicació específica perquè no es confonguin amb les imatges originals. Per això és recomanable crear un nou directori o afegir un sufix als noms d'arxiu per distingir les imatges comprimides.



És recomanable fer una còpia de seguretat de les imatges originals abans de fer qualsevol compressió per evitar pèrdua de dades en cas que ocorri algun problema durant el procés.

És necessari trobar un equilibri entre la qualitat i la grandària d'arxiu que sigui adequat per a les necessitats.



Compressió de vídeo per lots

Comprimir diversos vídeos en lots pot ser una tasca útil per reduir la mida dels fitxers i facilitar-ne la distribució. Els passos bàsics per comprimir vídeos en lots seran els mateixos que amb les imatges:

1 | Organitza els vídeos: igual que amb les imatges, serà convenient crear una carpeta o directori a l'ordinador i col·locar tots els vídeos que es vulguin comprimir en aquest directori. És recomanable assegurar-se que els vídeos estiguin en formats compatibles amb la compressió, com MP4, AVI o MKV.

2 | Triar una eina de compressió per lots: hi ha diverses eines disponibles que permeten comprimir vídeos en lots de manera eficient. Algunes opcions són:

- **HandBrake:** és una eina gratuïta i de codi obert que permet comprimir vídeos en lots. Es poden configurar els ajustaments de compressió, com la mida, la qualitat i el format de sortida, i després aplicar-los als vídeos en el directori corresponent.
- **FFmpeg:** és una solució de línia de comandaments molt potent i versàtil per processar vídeos. Es pot escriure un script de lot que utilitzi FFmpeg per a comprimir tots els vídeos en el directori desitjat. El script pot contenir els comandos necessaris per ajustar els paràmetres de compressió, com la taxa de bits, la resolució i el format de sortida.
- **Programes d'edició de vídeo:** alguns programes d'edició de vídeo, com Adobe Premiere Pro o Final Cut Pro, també tenen la capacitat de comprimir vídeos en lots. Es poden importar tots els vídeos en el programa, configurar els ajustaments de compressió i després exportar-los com un lot.

3 | Configurar la configuració de compressió: una vegada seleccionada l'eina de compressió, s'han de configurar els ajustaments de compressió segons les necessitats. Es poden ajustar paràmetres com la resolució, la taxa de bits, el còdec de compressió i altres aspectes que afectin la qualitat i la grandària de l'arxiu resultant.



4 | Dur a terme el procés de compressió en lots: després de configurar els ajustaments de compressió, cal iniciar el procés de compressió en lots utilitzant l'eina seleccionada. Això pot implicar carregar els vídeos en l'eina i aplicar-hi els ajustaments de compressió en cadascun, o executar un script de lot si s'està utilitzant alguna eina de línia de comandaments.

5 | Desar els vídeos comprimits: una vegada que es completi el procés de compressió, l'eina proporcionarà els vídeos comprimits. Una vegada més, és necessari desar-los en una ubicació específica perquè no es barregin amb els vídeos originals.

La compressió de vídeo implica una pèrdua de qualitat, per la qual cosa és recomanable fer proves prèviament amb un únic vídeo i ajustar els paràmetres de compressió per a trobar l'equilibri adequat entre qualitat i grandària d'arxiu segons les necessitats.

Saber-ne més

Si vols saber-ne més, pots consultar el llibre "*Compressió d'àudio i vídeo*" de Cliff Wootton (Editorial Anaya).





Creació de
continguts digitals

Nivell B2 3.1 Desenvolupament
de continguts

Alternatives per a la creació de llocs web





Alternatives per a la creació de llocs web

Introducció

Abans de l'existència de les eines que descriurem, la limitació més grossa que existia a l'hora de crear una pàgina web era que calia tenir coneixements de programació. Escriure codi pot ser una habilitat complexa d'aprendre, i moltes persones i empreses no podien permetre's la contractació de desenvolupadors web per crear el seu lloc web.

No obstant això, amb el sorgiment de plataformes accessibles i fàcils d'usar per a la creació de llocs web, aquesta limitació va disminuir significativament. Ara, qualsevol persona amb un ordinador (o fins i tot un mòbil) i una connexió a Internet pot crear una pàgina web d'aspecte professional i personalitzada sense haver d'aprendre a codificar des de zero.

Possibilitats actuals

A continuació, et presentam algunes de les opcions més populars per crear pàgines web sense haver d'escriure codi des de zero. Cadascuna té els seus aspectes positius i negatius, i l'opció òptima dependrà de les necessitats de cada usuari. En general, les podem agrupar en dues categories:

- **Plataformes en línia de construcció de llocs web:** són eines allotjades en el núvol que normalment es basen en sistemes d'"agafar i deixar anar" (de l'anglès, *drag and drop*), és a dir, que els usuaris poden afegir i personalitzar elements en el seu lloc web simplement arrossegant-los i col·locant-los on desitgin que apareguin. A més, solen presentar plantilles predissenyades que els usuaris poden personalitzar. Aquí trobem eines com Wix, Weebly, Squarespace, entre altres.
- **Sistemes de Gestió de Continguts (CMS, per les sigles en anglès):** a diferència dels anteriors, els CMS requereixen





una mica més de coneixement tècnic i, fins i tot, habilitats de codificació per explotar tot el seu potencial. WordPress és el més popular i àmpliament utilitzat, ofereix una gran quantitat de temes i connectors, és a dir, una àmplia gamma d'opcions, que ens permetran personalitzar el nostre lloc web proporcionant-li un aspecte professional. Per exemple, WordPress ofereix el connector "Elementor" que ens permet editar de manera visual (mitjançant drag and drop) sense necessitat d'escriure codi. WordPress és una plataforma de creació de llocs web més adequada per a llocs web més grans i complexos que requereixen més flexibilitat i escalabilitat.

Com podem suposar, aquestes eines han permès tant a particulars com a empreses tenir una presència en línia i expandir el seu abast sense haver d'invertir en habilitats tècniques costoses o recórrer a desenvolupadors web.

Saber-ne més

Adreces d'interès:

- **Wix:** es.wix.com
- **Weebly:** weebly.com/es
- **Squarespace:** es.squarespace.com
- **Wordpress:** wordpress.com
- **Visual Studio:** visualstudio.microsoft.com/es
- **Eclipse:** eclipse.org/ide
- **NetBeans:** netbeans.apache.org
- **IntelliJ IDEA:** jetbrains.com/idea

Entorns professionals

Els desenvolupadors web professionals solen treballar en **entorns de desenvolupament** (IDE, per les seves sigles en anglès), que són aplicacions que proporcionen als desenvolupadors eines per escriure, provar i depurar codi. Bàsicament, un IDE és un programa que permet als desenvolupadors crear aplicacions, llocs web, programari i altres programes. Els IDE solen incloure un editor de text, eines de depuració, un compilador, un depurador, un generador de codi i altres eines que ajuden els desenvolupadors a treballar de manera més eficient. Encara que presenten menys



limitacions que els CMS, es necessiten tenir coneixements de programació per poder aprofitar al màxim les seves funcionalitats. Alguns exemples d'entorns de desenvolupament populars són Visual Studio de Microsoft, Eclipse, NetBeans i IntelliJ IDEA.

Conclusions

Avui dia existeixen diverses alternatives per a la creació de llocs web que van més enllà de la creació de codi des de zero. Els gestors de continguts, com WordPress, ofereixen una manera més accessible i fàcil per a la creació de llocs web sense necessitat de tenir coneixements avançats en programació. A més, existeixen altres entorns de desenvolupament més avançats per a aquells que tinguin coneixements tècnics i vulguin desenvolupar llocs web més complexos i personalitzats. En definitiva, aquestes alternatives permeten més flexibilitat i versatilitat en la creació de llocs web, la qual cosa resulta en una major satisfacció i experiència d'usuari.



NOTA

Un **tema** a WordPress és un conjunt d'arxius que determinen el disseny i l'aparença d'un lloc web. Els temes es descarreguen i instal·len en la plataforma de WordPress i es poden personalitzar fàcilment per adaptar-se a les necessitats i preferències individuals d'un lloc web.

Els **connectors** són petites peces de programari que es poden descarregar i instal·lar en la plataforma de WordPress per afegir funcionalitats i característiques addicionals a un lloc web, com a formularis de contacte, galeries d'imatges, integració de xarxes socials, etc.

Saber-ne més

Per entendre millor el sistema de WordPress, pots accedir a la documentació que proporciona la plataforma tant per resoldre els primers dubtes com per a qüestions més avançades.



es.wordpress.org/support



Creació de
continguts digitals

Nivell B2 3.1 Desenvolupament
de continguts

Nivells d'adequació de WCAG



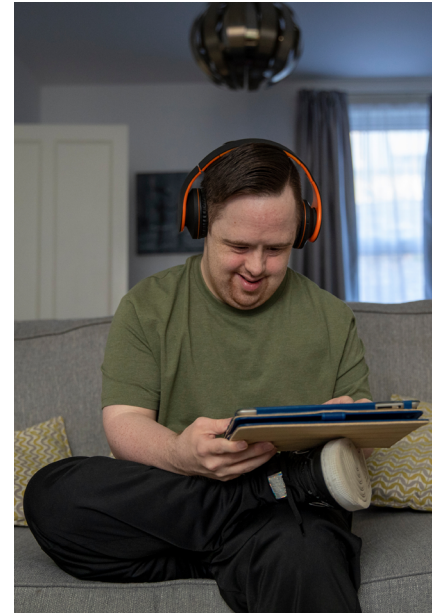


Nivells d'adequació de WCAG

Els nivells de conformitat del **WCAG (Web Content Accessibility Guidelines)** (e.digitall.org.es/niveles-WCAG) són una manera de classificar i avaluar l'accessibilitat d'un lloc web. Es van introduir a la versió 2.0 de les pautes de la WCAG, que van ser publicades pel World Wide Web Consortium (W3C) el desembre de 2008. Aquesta versió actualitzada va reemplaçar l'enfocament de prioritats de la versió 1.0 amb un sistema de nivells d'adequació.

Aquestes pautes proporcionen recomanacions detallades i tècniques per fer que el contingut web sigui més accessible. A la versió 2.0 del WCAG, es van establir un total de 12 pautes principals, agrupades en quatre principis fonamentals. Aquests principis són:

- 1 | Perceptible:** el contingut web ha de presentar-se d'una manera que sigui perceptible per a tots els usuaris, inclosos aquells amb discapacitats visuals o auditives. Les pautes d'aquest principi aborden aspectes com el contrast de color, l'ús d'alternatives textuais per a imatges i mitjans, i la capacitat de controlar la presentació del contingut.
- 2 | Operable:** el contingut web ha de ser operable per a tots els usuaris, inclosos aquells que tenen dificultats motores o de navegació. Les pautes d'aquest principi se centren en la navegació i la interacció, incloent-hi elements com l'accessibilitat del teclat, la capacitat d'ometre contingut repetitiu i el temps suficient per a llegir i utilitzar el contingut.
- 3 | Comprensible:** el contingut web ha de ser comprensible per a tots els usuaris, inclosos aquells amb discapacitats cognitives o de comprensió. Les pautes d'aquest principi aborden la claredat del llenguatge, l'estructura i l'organització del contingut, i la capacitat de proporcionar ajuda i orientació clares.
- 4 | Robust:** el contingut web ha de ser robust i compatible amb diferents tecnologies i navegadors, per a garantir que pugui ser accedit per una àmplia gamma d'usuaris. Les pautes d'aquest principi se centren en l'ús d'estàndards web, la compatibilitat amb tecnologies d'assistència i la creació de contingut que sigui resistent a errors i canvis en la tecnologia.





Cadascuna d'aquestes pautes es desglossa en criteris de conformitat específics, que indiquen els requisits tècnics i funcionals que han de complir-se per a aconseguir els nivells d'adequació (A, AA i AAA). Cal tenir en compte que les pautes i els criteris del WCAG continuen evolucionant i s'han llançat versions posteriors, com la versió 2.2 publicada el maig de 2022 i la versió 3.0 disponible i en contínua evolució. Cada versió introdueix noves pautes i criteris per abordar els desafiaments d'accessibilitat emergents i millorar l'experiència dels usuaris amb discapacitats al web.

Els criteris de conformitat específics, que indiquen els requisits tècnics i funcionals que han de complir-se per a aconseguir els nivells d'adequació, són els nivells A, AA i AAA.

Nivells de conformitat

L'objectiu principal d'introduir els nivells d'adequació va ser proporcionar una estructura més clara i flexible per avaluar i millorar l'accessibilitat web. Els nivells d'adequació permeten un enfocament progressiu, on els desenvolupadors i dissenyadors poden treballar cap a la millora contínua de l'accessibilitat en diferents etapes. Aquests nivells estan organitzats en tres categories: **A, AA i AAA**. Cada nivell té una sèrie de criteris de satisfacció que s'han de complir per a aconseguir aquest nivell de conformitat. A continuació, s'expliquen detalladament cada nivell i els seus criteris de satisfacció:

Nivell de conformitat A

El Nivell A estableix els criteris mínims d'accessibilitat que s'han de complir. Aquests criteris són els requisits bàsics per a aconseguir una certa mesura d'accessibilitat i abordar les barreres més comunes que enfronten les persones amb discapacitats. Complir amb el nivell A implica prendre mesures essencials per millorar l'accessibilitat i garantir que el contingut sigui accessible per a algunes persones amb discapacitats.





Alguns criteris de satisfacció són:

- Proporcionar alternatives textuais per a les imatges.
- Utilitzar una estructura d'encapçalats lògica per organitzar el contingut.
- Garantir que els formularis siguin accessibles i puguin ser completats per teclat.

Nivell de conformitat AA

El nivell AAA és el nivell més alt de conformitat i estableix els criteris més estrictes i complets per a l'accessibilitat. Complir amb els criteris de nivell AAA porta l'accessibilitat a un nivell superior i garanteix una experiència inclusiva per a una àmplia gamma de discapacitats. Aconseguir el nivell AAA pot requerir canvis significatius en el disseny, l'estructura i la funcionalitat del contingut web, i alguns dels criteris poden ser difícils d'aconseguir en uns certs contextos. Alguns criteris de satisfacció són:

Alguns criteris de satisfacció són:

- Proporcionar una navegació clara i consistent a tot el lloc web.
- Permetre als usuaris ajustar el temps d'espera abans que el contingut canviï automàticament.
- Proporcionar una versió simplificada del contingut per facilitar la seva comprensió.

Nivell de conformitat AAA

El nivell AAA és el nivell més alt de conformitat i estableix els criteris més estrictes i complets per a l'accessibilitat. Complir els criteris de nivell AAA porta l'accessibilitat a un nivell superior i garanteix una experiència inclusiva per a una àmplia gamma de discapacitats. Assolir el nivell AAA pot requerir canvis significatius en el disseny, estructura i funcionalitat del contingut web, i alguns dels criteris poden ser difícils d'aconseguir en certs contextos.

Alguns criteris de satisfacció són:

- Proporcionar una navegació clara i consistent a tot el lloc web.
- Permetre als usuaris ajustar el temps d'espera abans que el contingut canviï automàticament.
- Proporcionar una versió simplificada del contingut per a facilitar la seva comprensió.





Conclusions

El nivell AA és generalment considerat com l'estàndard recomanat i és àmpliament adoptat. Complir amb els criteris de satisfacció del nivell AA proporciona una accessibilitat sòlida i millora l'experiència per a la majoria dels usuaris amb discapacitats. El nivell AAA, encara que més rigorós, pot ser més difícil d'aconseguir i alguns dels seus criteris poden ser difícils de complir en uns certs contextos. Els nivells d'adequació no són obligatoris per llei en general (sí que ho són en portals d'administració pública), però complir-los és molt recomanable i pot ser requerit en uns certs contextos legals o governamentals. Especialment en docència, el compliment dels nivells d'adequació ajuda a garantir una experiència inclusiva per a una àmplia gamma d'usuaris, ja que dona exemple i proporciona beneficis tant per a les persones amb discapacitats com per a altres usuaris.

Saber-ne més

Si vols conèixer detalladament els continguts de la WCAG, pots revisar-los en la seva pàgina web.

[w3.org/TR/WCAG22](https://www.w3.org/TR/WCAG22)





DigitAll

3.2

INTEGRACIÓ I REELABORACIÓ DE CONTINGUT DIGITAL





Creació de
continguts digitals

Nivell B2 3.2 Integració i reelaboració
de contingut digital

Transicions i animacions avançades en presentacions





Transicions i animacions avançades en presentacions

Perquè una presentació sigui efectiva, aquesta ha d'incorporar **elements que imprimeixin estructura i ritme a les seves diapositives**. En aquest sentit, les eines de creació de presentacions posen a la disposició dels seus usuaris tres elements principals: transicions, animacions i botons d'acció.

Transicions

A l'hora de canviar de diapositiva quan es mostra una presentació, és comuna visualitzar efectes de moviment entre una diapositiva i la següent. Aquests efectes permeten donar continuïtat a la presentació i al seu contingut.

La **transició** de diapositives és l'**efecte visual** que s'aplica sobre l'espai entre una diapositiva i la següent.

Afegir una transició a una diapositiva en una presentació PowerPoint és molt senzill. Per això, l'usuari haurà de fer els següents passos:

- 1 | Seleccionar** la diapositiva sobre la qual s'aplicarà la transició.
- 2 | Fer clic** a la pestanya **Transicions**. Triar la transició desitjada entre les disponibles. La transició "Cap" permet eliminar la transició d'una diapositiva.
- 3 | Triar** la direcció de la transició en el botó "**Opcions d'efectes**".
- 4 | Al grup Intervals**, ajustar més opcions sobre la transició seleccionada: durada de la transició, aplicar la transició en fer clic o en transcórrer un temps determinat, agregar un so a la transició o aplicar la transició a totes les diapositives de la presentació.





Animacions

Les animacions dins d'una diapositiva van ser introduïdes en el document **A3C32B1D02 "Animacions en presentacions"**. En una diapositiva, és possible que un objecte, com un text, imatge, gràfic o forma, contengui més d'una animació. Quan en una diapositiva hi ha múltiples animacions, és usual utilitzar el panell d'animació.



ANIMACIONS EN PRESENTACIONS

Document referenciat:
A3C32B1D02

El panell d'animació és l'espai que permet administrar les propietats de les animacions aplicades a una diapositiva.

Per treballar amb el panell d'animació a Microsoft PowerPoint, s'han de fer els següents passos (Figura 1):

- 1 | Selecció de l'objecte** que es pretén animar dins de la diapositiva.
- 2 | En la pestanya Animacions**, seleccionar l'opció **Panell d'Animació**.
- 3 | Cada animació aplicada sobre un objecte s'afegirà al Panell d'Animació.**
- 4 | Per administrar les propietats d'una animació**, és possible fer clic a la fletxa que apareix a la dreta de l'animació en qüestió.
 - a. En fer clic:** reproduïx l'animació quan es fa clic amb el ratolí.
 - b. Iniciar amb anterior:** reproduïx l'animació al mateix temps que l'efecte anterior.
 - c. Iniciar després d'anterior:** reproduïx l'animació després de reproduir l'efecte anterior.
 - d. Intervals:** permet ajustar la durada de l'animació i altres propietats.

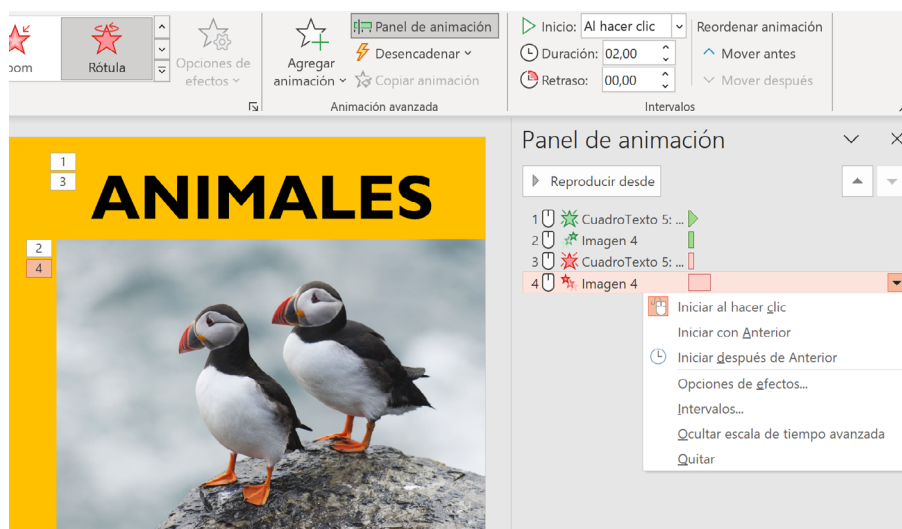


Figura 1. Principals opcions del panell d'animació.

Botons d'acció

En moltes ocasions és necessari que el presentador interactuï amb les diapositives durant la impartició de la presentació. Això pot ocórrer, per exemple, per obrir un arxiu local del presentador, accedir a una pàgina web a través d'un enllaç, reproduir una pista d'àudio del dispositiu o reproduir un vídeo extern. Per facilitar aquesta comunicació i interacció entre el presentador i la diapositiva, són molt útils els **botons d'acció**.

Un **botó d'acció** és un objecte, concretament una forma, que s'insereix en una diapositiva i que fa una acció quan es passa el ratolí o s'hi fa clic.

Per inserir botons d'acció dins d'una diapositiva PowerPoint, han de seguir-se els següents passos:

- 1** | **Selecció de la diapositiva** on inserir el botó d'acció.
- 2** | Fer clic en la **pestanya Insertar** i, després, en el botó Formes.
- 3** | Selecció del botó desitjat dins del grup **Botons d'acció**. Dibuixar la forma.
- 4** | S'obre un **quadre de diàleg**. Selecció de la pestanya "**Fer clic amb el ratolí**" o "**Passar el ratolí per damunt**" segons com es vol que s'executi l'acció associada al botó.



5 | Configurar l'acció del botó dins del llistat d'accions.

Les més importants són: crear un enllaç a qualsevol diapositiva, arxiu extern o adreça web; executar un programa o reproduir un so.

Saber-ne més

Transicions de diapositives en PowerPoint.

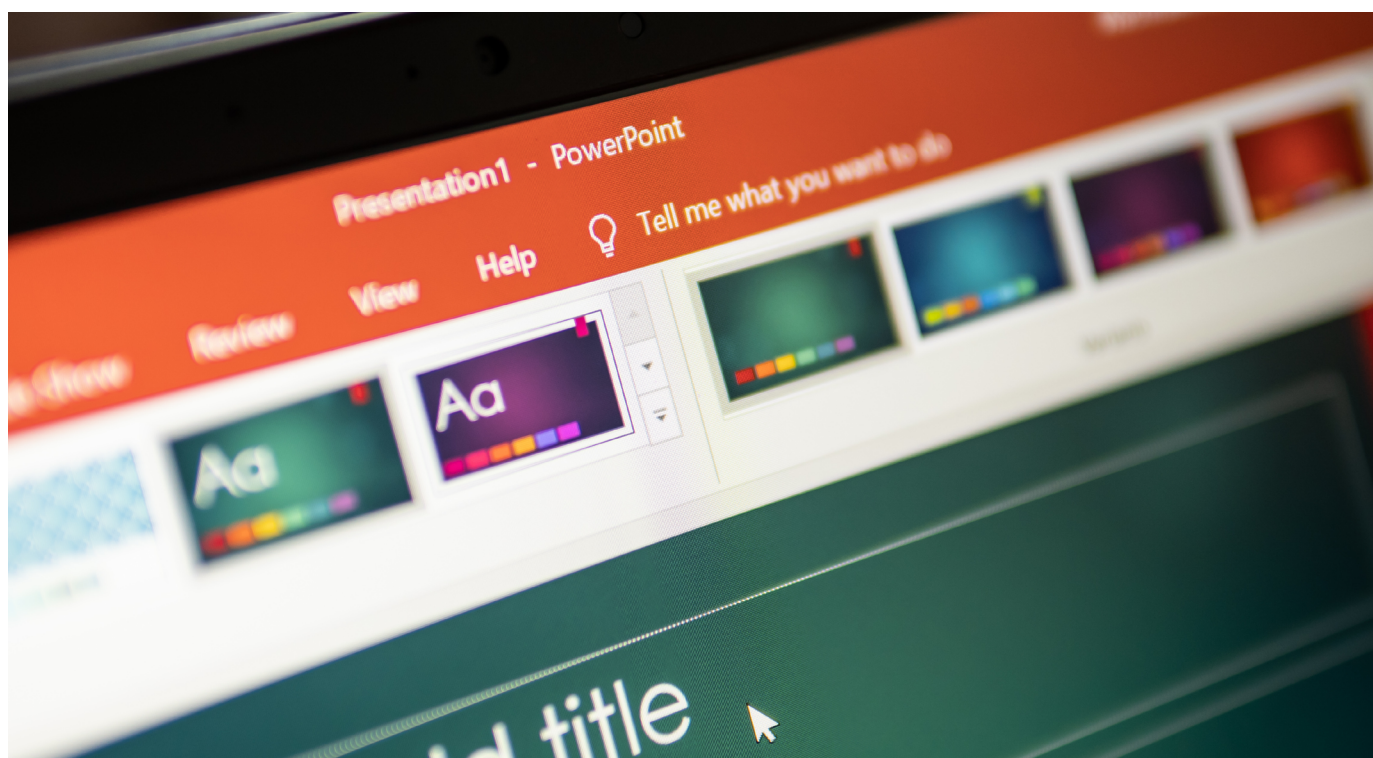
e.digitall.org.es/transiciones-diapositivas

El panell d'animació en PowerPoint.

e.digitall.org.es/efectos-animaciones

Botons d'acció en PowerPoint.

e.digitall.org.es/botones-accion





Creació de
continguts digitals

Nivell B2 3.2 Integració i reelaboració
de contingut digital

Remasterització de vídeo i àudio





Remasterització de vídeo i àudio

Com s'ha estudiat en els nivells anteriors, la qualitat dels components d'àudio i vídeo és molt important per generar una bona sensació en els consumidors de continguts digitals. No obstant això, com s'ha vist en el vídeo **A3C32B2V02 "Anàlisi de qualitat per a la millora de continguts digitals"**, els continguts poden tenir diverses deficiències que redueixen la qualitat d'aquest, com per exemple un vídeo amb **baixa resolució**, un **enquadrament inadequat**, que el **so sigui difícil de percebre**, etc. A continuació, es mostra com les tècniques de remasterització poden ajudar a solucionar aquestes problemàtiques.

La remasterització és el procés mitjançant el qual es modifica una obra sonora o audiovisual prèviament gravada per millorar la qualitat de l'enregistrament original.

⚠️ ATENCIÓ

No s'ha de confondre la **remasterització** amb la **reelaboració** de contingut digital, ja que en el procés de remasterització el contingut no es modifica per obtenir una versió diferent. La finalitat principal sempre serà la de millorar la qualitat del so o imatge de la peça audiovisual.



ANÀLISI DE QUALITAT
PER A LA MILLORA
DE CONTINGUTS
DIGITALS

e.digitall.org.es/A3C32B2V02

Remasterització d'àudio

Antigament, els estudis utilitzaven cintes per emmagatzemar les gravacions realitzades. Aquestes cintes eren **posteriorment processades i remasteritzades** per generar el denominat arxiu màster que posteriorment s'utilitzava per generar les còpies que eren distribuïdes. El problema amb els enregistraments fets en cinta és que aquestes posseeixen unes certes limitacions quant a la qualitat de so que permeten emmagatzemar, així com el soroll que es genera quan s'enregistra en aquest format. Per això, en l'actualitat, quan s'emmagatzemen aquestes gravacions antigues en digital, aquestes són sotmeses a un **procés de remasterització** en el qual s'elimina el soroll de fons de les gravacions, es redueixen els defectes produïts per deterioracions en les cintes, i es millora el so dels





En l'actualitat, els àlbums d'estudi són enregistrats utilitzant pistes separades per a cadascun dels instruments, la qual cosa dona lloc a peces amb més de 48 pistes en molts casos. A continuació, els tècnics de l'estudi barregen cadascuna d'aquestes pistes de la manera desitjada per obtenir un fitxer de dues úniques pistes (corresponents a altaveus situats a esquerra i dreta) en un format denominat estèreo. En tenir diferents pistes per a cada instrument, es pot remasteritzar la gravació per **accentuar diferents instruments, modificar la percepció dels instruments en l'espai**, o modificar les freqüències del senyal per a equalitzar el so. Una eina en línia molt útil és **Audioalter** (audioalter.com).

Finalment, un altre dels usos principals de la remasterització d'àudio consisteix a **recuperar la definició d'àudio perduda durant la compressió** d'aquest per al seu emmagatzematge de manera eficient. Aquesta tècnica pot ser molt interessant per a ús domèstic, ja que els fitxers d'àudio comercials en la primera dècada dels 2000 utilitzaven formats amb una alta taxa de compressió per reduir al màxim la mida del fitxer (sacrificant la definició de so). El motiu d'aquesta compressió era la reduïda capacitat d'emmagatzematge dels dispositius de reproducció de l'època. L'eina **Podcastle.ai** (e.digitall.org.es/magic-dust) també permet augmentar la qualitat dels enregistraments.

Remasterització de vídeo

La remasterització de vídeo és **àmpliament utilitzada en l'actualitat**, principalment en la **indústria cinematogràfica**. Els seus usos són molt variats, ja que permet **modificar una gran quantitat d'aspectes sobre l'enregistrament original**. Per exemple, es poden modificar els plans de les preses per aconseguir diferents enquadraments o corregir una càmera mal alineada respecte a l'horitzó. També poden utilitzar-se per modificar la saturació del color, el contrast i la lluentor, corregint així les preses massa fosques o sobreexposades.



**EINES BASEDES
EN INTEL·LIGÈNCIA
ARTIFICIAL PER
A MODIFICAR
CONTINGUT DIGITAL**

e.digitall.org.es/A3C32B1V04

Saber-ne més

Una imatge sobreexposada s'obté quan la quantitat de llum que aconseguix el sensor de la càmera és excessiva, fet que impedeix apreciar amb claredat els detalls de les parts més il·luminades.



Un altre ús molt interessant de la remasterització de vídeo és l'aplicació d'**eines d'acoloriment de gravacions antigues en blanc i negre**. Mitjançant tècniques basades en intel·ligència artificial s'aconsegueix "**endevinar**" quin és el color de cadascuna de les parts de la imatge en cada fotograma del vídeo. Aquestes tècniques resulten molt cridaneres, ja que aconseguen dur al present escenes de pel·lícules molt antigues. Destaca, per exemple, l'eina **DeOldify** (deoldify.ai) que utilitza xarxes neuronals profundes per acolorir imatges i vídeos. La Figura 1 mostra un exemple de l'ús d'aquesta tècnica.



Figura 1. "Toffs and Toughs" by Jimmy Sime (1937). Imatge obtinguda usant DeOldify.

També és comú aplicar tècniques per augmentar la resolució dels vídeos, adaptant els enregistraments amb baixa resolució a resolucions FullHD o 4K. Això permetria projectar el vídeo original en pantalles amb una resolució molt major evitant que el resultat es vegi pixelat. Dins d'aquestes tècniques, es poden usar eines com **Gigapixel AI** (topazlabs.com/gigapixel-ai) de Topaz Labs, la qual empra **xarxes neuronals profundes** per augmentar la resolució dels vídeos. Aquesta aplicació és usada per empreses com Netflix, Disney, Apple, Warner Bros, entre altres.



Altres usos de tècniques de remasterització de vídeo inclouen l'interpolat de fotogrames, la qual cosa permet donar suavitat als vídeos en passar aquests de 24 o 30 fotogrames per segon a 60 fotogrames per segon. Destacam l'eina **Smoother AI** (e.digitall.org.es/smoothier-ai) que permet aplicar aquesta tècnica fàcilment. També és comú aplicar tècniques de reducció de soroll per evitar el soroll generat en emmagatzemar vídeos antics en sistemes d'emmagatzematge que puguin tenir pèrdues o estar danyats. Hi ha multitud d'eines per aconseguir aquesta reducció de soroll, com per exemple **Neat Video** (neatvideo.com).

Com pot veure's, les possibilitats en aquest camp són molt àmplies. A més, l'auge de les tècniques d'intel·ligència artificial en aquests últims anys ha multiplicat les possibilitats i el nombre d'eines disponibles per remasteritzar àudio i vídeo. Animam al lector a investigar sobre aquestes eines amb els seus mateixos vídeos, aconseguint així materials de major qualitat i suplint les deficiències comentades en el vídeo **A3C32B2V02 "Anàlisi de qualitat per a la millora de continguts digitals"**.



ANÀLISI DE QUALITAT
PER A LA MILLORA
DE CONTINGUTS
DIGITALS

e.digitall.org.es/A3C32B2V02

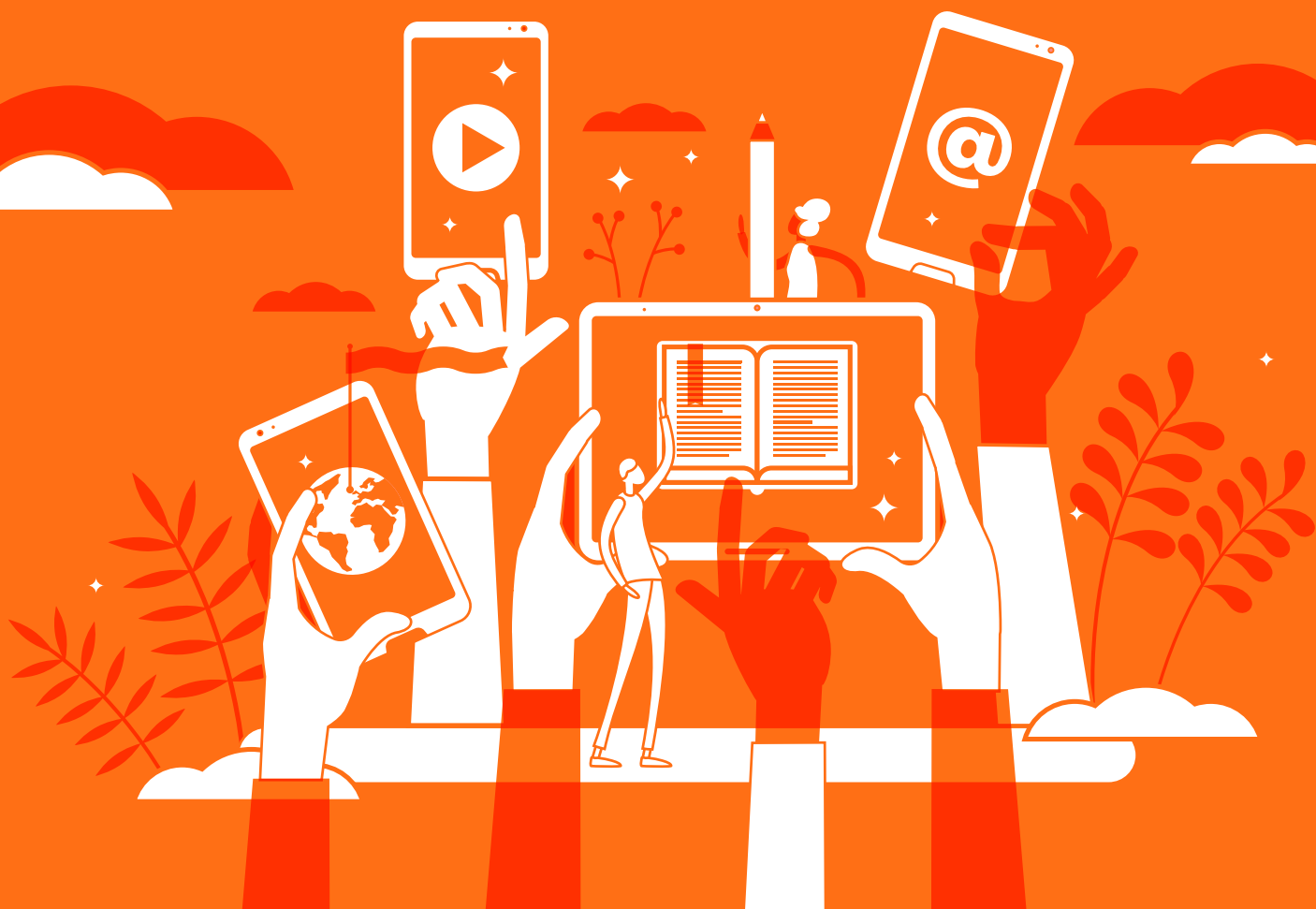




DigitAll

3.3

DRETS D'AUTOR I LICÈNCIES DE PROPIETAT INTEL·LECTUAL





Creació de continguts digitals

Nivell B2 3.3 Drets d'autor i llicències de propietat intel·lectual

**Dotant
de copyright
una obra**





Dotant de copyright una obra

El contingut d'aquest document brinda coneixements sobre el mecanisme de registre de la propietat intel·lectual.

El Registre de la Propietat Intel·lectual és únic en tot el territori nacional. Les seves competències estan distribuïdes entre l'Estat i les comunitats autònomes. Per aquesta raó, les inscripcions fetes a qualsevol dels Registres tenen la mateixa validesa en tot el territori espanyol.

El Registre Central forma part de l'Administració General de l'Estat i depèn del Ministeri de Cultura i Esport. a totes les capitals de província de les comunitats autònomes de Canàries, Illes Balears, Cantàbria, Castella i Lleó, Castella-la Manxa i Navarra, així com a les Ciutats Autònomes de Ceuta i Melilla existeix una Oficina del Registre Central.

A les comunitats autònomes d'Andalusia, Aragó, Astúries, Catalunya, Comunitat Valenciana, Extremadura, Galícia, La Rioja, Madrid, Múrcia i País Basc i a les ciutats de Ceuta i Melilla existeixen Registres Territorials que depenen de cadascuna d'aquestes Comunitats o Ciutats Autònomes.

- ◇ Les sol·licituds d'inscripció que es presentin en el Registre Central i a través de les Oficines Provincials que en depenen, es poden fer:

- **Presencialment**, aportant:

- Els impresos oficials de sol·licitud d'inscripció.
- L'exemplar de l'obra, actuació o producció, en la forma i mode indicats en els impresos oficials.
- La documentació habitual, que, segons els casos, es requereixi en virtut de la legislació és:
 - ◇ **Dades del sol·licitant i autors:** fotocòpia del DNI o document acreditatiu de la identitat si fos estranger. Si fos menor o incapacitat s'haurà de presentar fotocòpia del llibre de família o document que acrediti la pàtria potestat, tutela o curatela.
 - ◇ **Dades del representant, si existís:** el Poder de Representació.
- Justificant de l'abonament de la taxa corresponent. El pagament s'efectua prèvia liquidació que efectua el Registre en l'imprès oficial.

⚠ ATENCIÓ

Cada Registre Territorial té les seves pròpies instruccions de presentació de sol·licituds, és per això que, per presentar una sol·licitud en un Registre Territorial, s'haurà de contactar prèviament amb ell per informar-se sobre la manera de fer-ho.



- **Telemàticament** mitjançant l'aplicació telemàtica **RePI** (e.digital.org.es/repi) disponible en la Seu Electrònica del Ministeri de Cultura i Esport. El sol·licitant haurà de disposar de signatura electrònica per fer el procediment de registre via telemàtica. Una vegada que hagi gestionat el procés d'identificació i autenticació s'hauran de seguir els següents passos:
 - **Emplenar la sol·licitud.** Per això s'han de completar les dades del sol·licitant i del representat, en cas que existís. També s'han d'introduir les dades de l'obra que es vol registrar, com per exemple: tipus d'obra, títol de l'obra i data de publicació o divulgació, en cas que es tracti d'una obra divulgada o publicada. A més, s'han d'incloure les dades referides a l'autor o autors, si l'obra correspon a un, o diversos autors. En aquest últim cas haurà d'indicar-se expressament. Finalment, també es podrà adjuntar documents addicionals a la sol·licitud, quan així ho indiqui expressament la mateixa sol·licitud o l'interessat el cregui necessari
 - **Pagament de les taxes.** La sol·licitud de primera inscripció requereix efectuar el pagament d'una taxa abans de signar i enviar la sol·licitud. El pagament es podrà efectuar de manera telemàtica o en el banc.
 - **Firma i enviament de la sol·licitud.** Una vegada introduïdes les dades necessàries en l'aplicació, s'haurà d'"Enviar" i a continuació "Signar", emprant la signatura electrònica.

Si tot ha anat bé en la tramitació telemàtica, s'indicarà que la sol·licitud ha estat registrada amb èxit i es mostraran totes les dades introduïdes en la sol·licitud, en què s'incluen també el número de l'expedient que s'ha creat, el número de registre i la data.

A més, caldrà aportar una còpia electrònica de l'exemplar identificatiu de l'obra. Per això, una vegada que s'ha registrat una obra, el sistema oferirà una adreça web per procedir a l'aportació de la còpia de l'exemplar identificatiu.

⚠ ATENCIÓ

Quan es fa un registre cal indicar clarament el tipus d'OBRA. S'ha de seleccionar el tipus i subtipus d'obra. Per a alguns tipus és possible introduir els detalls específics de l'obra. Aquests detalls no cal incloure'ls per als següents tipus d'obra: coreografia i pantomima, bases de dades, programes d'ordinador, pàgina web o multimèdia, mera fotografia, dret "sui generis" sobre una base de dades.

**NOTA**

Quan existeixin diversos autors, a l'hora d'indicar els "Autors" s'haurà d'indicar quin dels tipus d'autoria assenyalats en aquest apartat s'atribueix a cadascun.

Saber-ne més

Oficines dependents dels Registres Territorials e.digitall.org.es/registro-territorial

Oficines dependents del Registre Central: e.digitall.org.es/registro-central

Legislació estatal:

- Reial decret legislatiu 1/1996, de 12 d'abril, pel qual s'aprova el text refós de la Llei de Propietat Intel·lectual, regularitzant, aclarint i harmonitzant les disposicions legals vigents sobre la matèria (e.digitall.org.es/boe-8930).
- Reial decret 281/2003, de 7 de març, pel qual s'aprova el Reglament del Registre General de la Propietat Intel·lectual (e.digitall.org.es/boe-6247).
- **Legislació autonòmica:** e.digitall.org.es/legislacion-autonomica





Creació de
continguts digitals

Nivell B2 3.3 Drets d'autor i llicències
de propietat intel·lectual

**Programari
lliure, freeware,
shareware,
adware**





Programari lliure, freeware, shareware, adware

El programari també és un **producte intel·lectual que pot i ha de protegir-se**. Un programador que ha desenvolupat un valuós i útil programari es pot plantejar què fer amb el seu programari, amb el seu programa informàtic, perquè pugui ser utilitzada per tercers. Aquest permís d'ús a una altra persona, incloses també la còpia, la modificació, la distribució o la venda, amb els límits i condicions que tu vulguis establir, es formalitza mitjançant la llicència del programari. Aquesta **llicència** és un **contracte sota el qual un titular d'un element protegit per propietat intel·lectual** (per exemple, el nostre programari) concedeix la seva autorització d'ús a una altra persona, a un tercer, amb les condicions establertes en les clàusules. Encara que si el que vol és vendre la titularitat del seu programa haurà de signar-se un contracte de cessió de programari. El titular o autor del programari és el llicenciador i l'usuari és el llicenciatari.

Quin tipus de llicència es pot aplicar al programari? En funció de les condicions o permisos o autoritzacions que es vulguin imposar als llicenciataris, i depenent de la plataforma en la qual voler situar-ho, es distingeixen els següents tipus de llicències:

- **Llicència GPL (GNU LGPL)**. És programari lliure i permet que s'enllaci amb mòduls no lliures. Ha d'estar disponible i accessible per a còpies il·limitades i a qualsevol persona que ho sol·liciti. En la seva primera versió es deia 'Llicència Pública General per a Biblioteques de GNU.
- **Llicència BSD**. És una llicència permissiva, perquè gairebé no imposa condicions sobre el que un usuari pot fer: el programari pot ser venut i no és obligatori incloure el codi font.
- **Llicències MPL**. Són les de Mozilla Firefox, Mozilla Thunderbird i la major part d'altre programari de Mozilla, una mica menys permissives que les anteriors.





- **Llicència Debian** (*Debian Free Software Guidelines-DFSG*). Hi deriva la llicència d'*Open Source Initiative*. Estableix l'exigència de publicació del codi font, la redistribució lliure (també del treball derivat sota la mateixa llicència de l'original) i que no pot discriminar cap persona o grup de persones, ni el programari ni cap forma de la seva utilització.
- **Llicència Open Source Initiative**. Una de les llicències més conegudes i utilitzades, anomenada també de fonts obertes o codi obert, que permet la lliure distribució (també del codi font), la modificació d'aquest, els desenvolupaments derivats i la seva redistribució en les mateixes condicions que el programari original. A més, Open Source estableix la integritat del codi font de l'autor, que no ha de ser discriminatòria de cap persona o grups de persones, ni restringir la utilització del programari a camps de domini o activitat i ha de ser neutral en relació amb la tecnologia.
- **Llicència de X.org** (X Windows System). Distribuïda pel Consorci X, com a organització que controla el seu desenvolupament, és usada en els sistemes operatius Linux i UNIX nascuts com a alternativa als sistemes gràfics com Microsoft. És programari lliure, encara que sense adherir-se al copyleft.
- **Llicència Freeware**. Autoritza l'ús del programari de manera lliure i gratuïta, però en general a particulars, i no per a empreses o organismes oficials. La llicència Donationware és similar, però demana a canvi un donatiu, encara que no sigui obligatori. La llicència Postcardware és similar, però sol·licita l'enviament d'una postal com a confirmació de la seva utilització, en lloc del donatiu.
- **Llicències de prova o Shareware**. Sol·liciten un registre previ i autoritza l'ús d'un programa perquè l'usuari l'avalui i posteriorment el compri. La llicència de Demo és la sessió d'un programa per a la seva avaluació. Són molt populars en el cas de proves de videojocs, per exemple.

Finalment, si no volem que l'usuari del programari no el modifiqui, ni el comparteixi, ni el distribueixi sense l'autorització del creador, la llicència seria de tipus privatiu, com són, per exemple, les llicències del sistema operatiu macOS d'Apple o Windows.

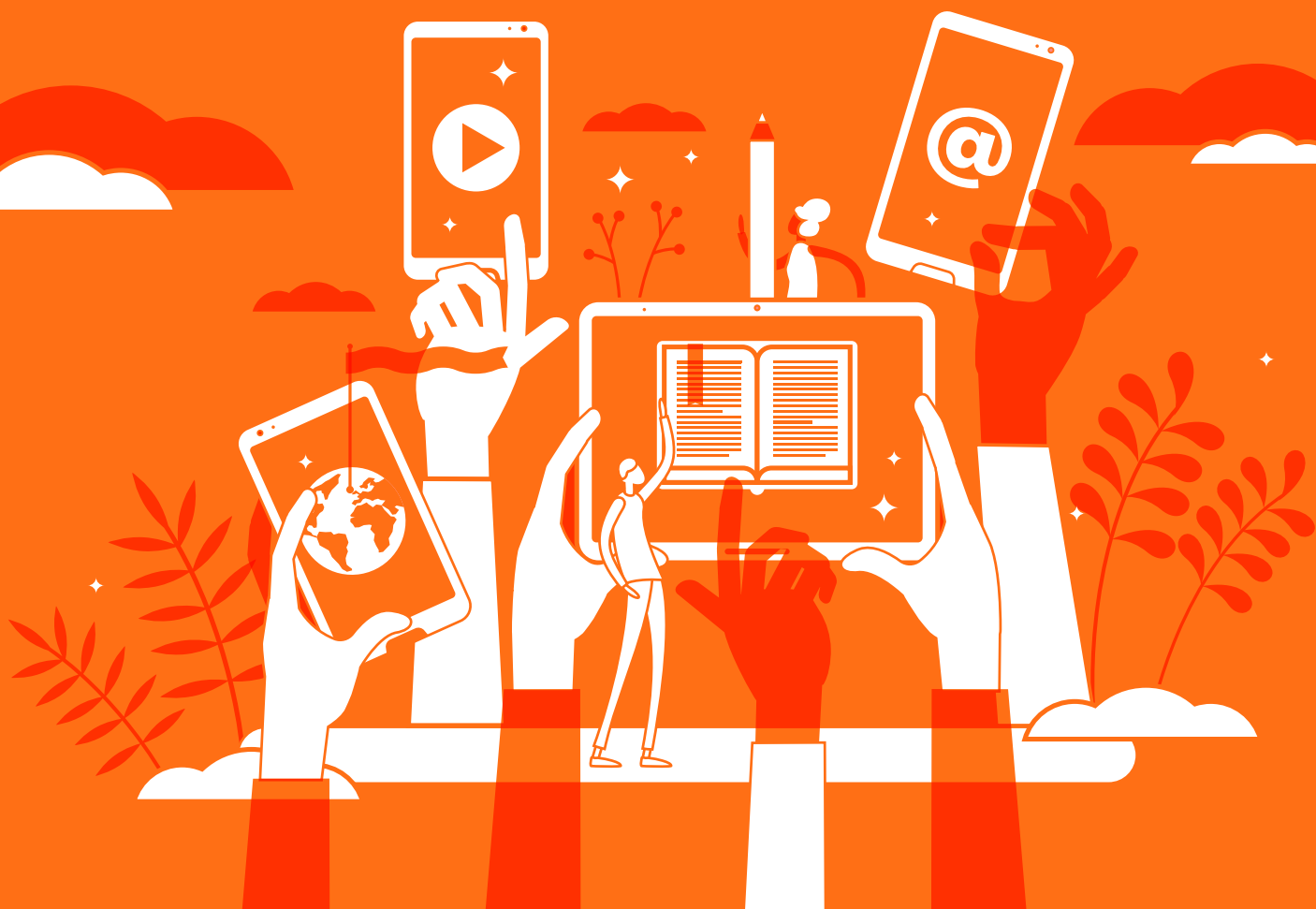




DigitAll

3.4

PROGRAMACIÓ





Creació de
continguts digitals

Nivell B2 3.4 Programació

Aspectes generals de Python com a llenguatge





Aspectes generals de Python com a llenguatge

Introducció

Quan un inicia el seu camí d'aprenentatge en el món de la programació, una de les decisions a prendre és "quin llenguatge de programació emprar". Python pot ser una bona elecció per diverses raons: es tracta d'un llenguatge versàtil, flexible, amb una corba d'aprenentatge relativament suau en comparació amb altres llenguatges, multiplataforma i totalment gratuït.

En aquesta secció s'introduiran els aspectes lèxics, sintàctics i semàntics de Python. Es presentarà l'estructura general d'un programa en Python, introduint i familiaritzant a l'estudiant amb els diferents nivells conceptuals que solen veure's en codi Python. Finalment, es discutirà el concepte d'indentació en Python i l'absència de claus/sentències per delimitar l'abast d'instruccions o estructures de control.



Aspectes lèxics, sintàctics i semàntics de Python

Quan s'utilitza un llenguatge per expressar-nos, es fa ús de les eines que proporciona com l'alfabet, el vocabulari amb el conjunt de paraules que el formen i el seu significat, i la gramàtica que representa el conjunt de regles que determinen l'ordre en el qual han d'aparèixer les paraules. En el cas d'un llenguatge de programació, els elements disponibles són similars.

El **lèxic** d'un llenguatge de programació, com Python, determina els símbols i paraules que podem emprar per a construir les sentències que s'inclouen en els programes. L'**anàlisi lèxica** de Python desglossa un arxiu (amb el programa que hem escrit) en tokens segons regles bàsiques que governen el mode d'escriure programes en aquest llenguatge.

A continuació, l'**analitzador sintàctic** de Python s'encarrega d'analitzar l'**estructura sintàctica** del programa. Utilitza la sortida de l'analitzador lèxic per a construir un arbre de sintaxi abstracte (ANSA) que representa l'estructura sintàctica del



programa d'entrada. Amb aquesta estructura, es poden fer diverses tasques, la detecció d'errors sintàctics o la generació de codi objecte. Finalment, l'**analitzador semàntic** fa ús de la sortida generada per l'analitzador sintàctic per a detectar possibles errors que afecten la **semàntica del llenguatge**, com la verificació de tipus, accés a variables i mètodes, rang de valors, assignacions vàlides, variables no definides o operacions il·legals, entre altres.

Perquè un programa en Python pugui ser executat, aquest ha d'estar lliure d'errors lèxics, sintàctics i semàntics.

Estructura general d'un programa a Python

Un programa en Python sol tenir la següent estructura general:

1 | Comentaris: són línies de text que expliquen el propòsit del programa i com funciona. Es tracta d'un element opcional, a criteri del programador, i és ignorat per l'interpret de Python. Per tant, no produeix cap mena d'error.

```
# Aquest és un comentari en una línia
"""
Aquest és un comentari en diverses línies
utilitzant cometes triples
"""
```

2 | Declaració d'importacions: són línies de codi que permeten importar mòduls o biblioteques que contenen funcions i classes útils per al programa.

```
import math
print(math.pi) # Imprime el valor de pi (3.141592653589793)
```

3 | Definició de funcions: són blocs de codi que fan una tasca específica i poden ser anomenades diverses vegades en el programa. Les funcions es defineixen utilitzant la paraula clau "def" seguida del nom de la funció i els paràmetres entre parèntesis.

```
def suma(a, b):
    resultat = a + b
    return resultat
```



4 | Declaració de variables: són línies de codi que declaren i assignen valors a variables que s'utilitzaran en el programa. Aquestes variables poden ser senzilles (emmagatzemen un únic valor) o estructures de dades més complexes, com matrius, vectors, llistes, o diccionaris, entre altres.

```
# Assignar valors a variables
número = 10
nom = "Josep Antoni"
és_actiu = True
```

5 | Bloc principal: és el cos principal del programa on s'efectuen les operacions i es criden a les funcions definides anteriorment. Tant en el bloc principal com en les funcions, se solen emprar estructures de control de flux, com les condicionals i els bucles.

Indentació a Python

La indentació a Python és l'ús d'espais o tabulacions al començament d'una línia de codi per indicar la jerarquia i estructura del programa. En Python, les indentacions són utilitzades per a indicar l'abast de les estructures de control de flux (com a condicionals i bucles) i de les funcions.

L'absència de claus o sentències per delimitar l'abast d'instruccions o estructures de control en Python és una **característica distintiva del llenguatge**. En lloc d'utilitzar claus per a delimitar l'abast, tal com es fa en molts altres llenguatges de programació, s'utilitzen indentacions. A continuació, es mostra un exemple en el qual es representa, en primer lloc, un senzill fragment de codi amb l'ús de claus per delimitar el bloc, i el seu equivalent en Python emprant les indentacions:

```
if x > 0: {
    print("x és major que zero")
}
```

Codi a Python:

```
if x > 0:
    print("x és major que zero")
```

⚠ ATENCIÓ

Les indentacions a Python no són un aspecte purament estètic i, en el cas de no respectar-les, produiran errors de sintaxi. Es tracta d'una característica particular de Python que fa que el codi sigui més llegible i fàcil de comprendre, facilitant, de manera clara, la visualització de la jerarquia i l'estructura dels programes.



Creació de
continguts digitals

Nivell B2 3.4 Programació

Tipus de dades a Python





Tipus de Dades a Python

Introducció

Com en qualsevol altre llenguatge de programació, Python admet l'emmagatzematge de dades en variables, i cada variable pot tenir un tipus de dada associada. En la següent secció s'oferirà una visió general dels tipus de dades bàsiques que ofereix Python i un exemple d'ús de cadascun.

Tipus de dades bàsiques a Python

1 | Boolean Type (bool): a Python, el tipus de dades 'bool' representa un valor lògic, és a dir veritable (*True*) o fals (*False*). Aquest tipus de dades sol ser utilitzat habitualment en expressions condicionals i operacions lògiques. Un exemple senzill podria ser el següent:

```
x = True
y = False
if x:
    print("x és vertader")
else:
    print("x és fals")
```

2 | Text Type (str): el tipus de dada 'str' en Python representa una seqüència de caràcters (text) i pot ser definit entre cometes simples o dobles. A continuació, es mostra un exemple senzill en el qual es crea una cadena de text i es mostra en pantalla:

```
missatge = "Hola, món"
print(missatge)
```

3 | Numeric types (int, float): Python ofereix la possibilitat de representar tipus de dades numèriques. Els més habituals són 'int', que permet representar nombres sencers i 'float', que representa números amb decimals. Es tracta d'una mena de dada que permet la realització d'operacions matemàtiques com a suma, resta, multiplicació o divisió, entre altres. Alguns exemples senzills d'ús d'aquests dos tipus de dades són els següents:





```
num1 = 5
num2 = 10
resultado = num1 + num2
print(resultado)
```

```
num1 = 5.5
num2 = 10.1
resultado = num1 + num2
print(resultado)
```

Si s'efectués una operació entre un sencer i un número en punt flotant, el resultat seria un número en punt flotant.

4 | Sequence Types (list, tuple, range): els tipus de dades de seqüència Python, tal com el seu mateix nom indica, representen seqüències d'elements. Aquests tipus inclouen *list*, *tuple* i *range*.

List és un tipus de seqüència que permet emmagatzemar la seqüència ordenada d'elements de qualsevol mena. Les *l·listes* es defineixen entre claudàtors `[]` i poden ser modificades després de la seva creació.

```
fruites = ["pomes", "banana", "pera"]
print(fruitess)
```

Tuple és similar a *list*, tret que una tupla no pot ser modificada una vegada creada. Les *tuples* es defineixen entre parèntesis `()`.

```
dies_setmana = ("dilluns", "dimarts", "dimecres")
print(dies_setmana)
```

- **Range** és un tipus de dada que permet crear una seqüència de nombres sencers i se sol utilitzar amb freqüència en estructures de control de flux com els bucles *for*. El següent exemple generaria i imprimiria la seqüència de nombres sencers de l'1 al 10.

```
números = range(10)
print(list(números))
```

A la darrera línia de codi, el rang "*números*" és convertit a una llista per imprimir els elements generats. També podrien ser emmagatzemats per a la seva manipulació, si així es desitges. A continuació, es mostra un senzill exemple:

```
rang = range(1, 6)
l·lista = list(rang)
```



5 | Mapping Type (dict): a Python, el *mapping* es refereix a una mena d'objecte que emmagatzema una col·lecció d'elements en parells clau-valor, on cada clau està associada amb un valor. En altres llenguatges de programació, aquest tipus d'objecte a vegades es coneix com un "diccionari" o "taula hash". Les claus són úniques (no podem usar claus iguals per a parells distints) i immutables (no es poden modificar), i cada clau està associada a un valor. Precisament, aquestes claus s'utilitzen per accedir als valors als quals estan associades. Per exemple, podríem crear un diccionari en el qual les claus siguin noms de països i els valors associats les capitals d'aquests països. D'aquesta manera, podríem recuperar sempre de manera senzilla el nom de la capital a partir del nom del país:

```
capitals = {'Espanya': 'Madrid', 'França': 'Paris', 'Alemanya': 'Berlin', 'Itàlia': 'Roma'}
```

S'ha de recalcar que com a valor podem emprar diferents classes d'elements, com a llistes, nombres sencers, o qualsevol altre tipus d'objecte mutable

Una manera senzilla de mostrar en pantalla tant les claus com els valors associats és la següent:

```
print(list(capitals.keys()))  
print(list(capitals.values()))
```

6 | Set Types (set): a Python, set és un tipus de dada que permet emmagatzemar una col·lecció d'elements únics i sense ordre. És similar a una llista o un diccionari, però sense claus i amb elements únics. Els sets es creen usant claus `{}` o la funció `set()`. Per exemple, podríem definir d'aquesta manera el conjunt de colors els valors dels quals no procedeix que es modifiquin, en la major part d'ocasions:

```
colors = {'vermell', 'verd', 'blau'}  
colors = set(['vermell', 'verd', 'blau'])
```

Aquest tipus de dada suporta operacions matemàtiques-lògiques com la unió, la intersecció o la diferència, entre altres.

Saber-ne més

Per saber més sobre els tipus de dades a Python pots consultar la documentació oficial en castellà a docs.python.org/es.



Creació de
continguts digitals

Nivell B2 3.4 Programació

Aspectes generals de variables i constants a Python





Aspectes generals de variables i constants a Python

Les variables, per conveni, se solen nomenar amb lletres minúscules per facilitar la seva identificació. A més, els seus noms o identificadors han de començar per una lletra o per un guió baix (`_`) i poden seguir amb més lletres, números o guions baixos. Els identificadors no poden incloure espais en blanc, ni es poden utilitzar noms de funcions ja existents a Python. Si un identificador està format per diverses paraules, sol separar-se amb guions per conveni.

Una variable es considera definida una vegada que se li ha assignat un valor i, per tant, està inicialitzada. Un exemple de variables ben definides seria el següent:

```
preu_hora = 5
dies = 10
quilòmetres = 60
```

Les constants no existeixen com a tal a Python. Per això, els programadors simplement declaren una variable, li assignen un valor inicial i esperen que ningú no modifiqui aquest valor. Existeix una convenció entre programadors per al nomenat de les constants, per així identificar-les fàcilment i recordar que no han de ser modificades. Al contrari que les variables, i atès el conveni que se sol adoptar, el seu nom s'ha d'escriure completament en majúscules.

Un exemple de constants ben definides seria el següent:

```
DIES_ANY = 365
MINUTS_HORA = 60
SEGON_HORA = 60
```



Assignació múltiple

Per definir una variable cal inicialitzar-la amb l'operador "=", com s'ha vist anteriorment. A més, existeixen maneres d'inicialitzar variables conjuntament. Per exemple, un valor pot ser assignat a diverses variables alhora:

```
a = b = c = 1 # Comença a, b i c amb el valor 1
```

En cas que es vulguin inicialitzar diverses variables alhora, però amb diferents valors, la sintaxi seria com segueix:

```
a, b, c = 10, 50, 75 # Comença a amb 10, b amb 50 i c amb 75
```





Creació de
continguts digitals

Nivell B2 3.4 Programació

Funcions i pas de paràmetres a Python





Funcions i pas de paràmetres a Python

Python és un llenguatge de programació versàtil que posa l'accent en la llegibilitat del codi i la sintaxi simple. Un element fonamental en qualsevol llenguatge de programació, incloent-hi Python, és la definició de funcions. A continuació, es discutiran les funcions de Python i es proporcionaran detalls sobre com es creen i s'utilitzen. A més, s'examinarà la diferència entre arguments i paràmetres i s'introduirà el concepte de paràmetres per defecte.

Creació i ús de funcions

En Python, una funció és un bloc de codi reutilitzable que du a terme una acció específica. Les funcions proporcionen una manera d'organitzar el codi en blocs lògics, la qual cosa permet més claredat i facilitat de manteniment. Les funcions es creen utilitzant la paraula clau **def**, seguida del nom de la funció i els parèntesis. Per exemple:

```
def la_meva_funció():  
    print("¡Hola, món!")
```

Una vegada definida, la funció es pot cridar o invocar, per executar-la, utilitzant el seu nom seguit de parèntesi:

```
la_meva_funció()
```

Executar aquesta funció mostraria per pantalla el text «Hola, món!». Com es pot observar, una funció pot estar formada per moltes altres expressions que seran executades en ordre quan s'invoqui a aquesta funció. En aquest cas, la funció que s'ha definit executaria una única expressió formada per la invocació a una altra funció anomenada **print** per mostrar text en pantalla.





Arguments i paràmetres

Els termes argument i paràmetre sovint s'utilitzen indistintament en programació, però hi ha una diferència clau. Un paràmetre és una variable en la definició de la funció, mentre que un argument és el valor que es passa a la funció quan es diu.

Per exemple, en la següent funció, **nom** és un paràmetre:

```
def saludar(nom):  
    print(f"iHola, {nom}!")
```

I quan s'invoca a la funció, el valor que li passa, **Pere**, és un argument:

```
saludar("Pere")
```

Però, què succeiria si s'invocés a la funció de l'exemple sense cap argument? Per exemple, talment així:

```
saludar()
```

En aquest cas, s'ha definit una funció que espera un argument en el seu paràmetre **nombre**, per la qual cosa, si s'invocés a la funció sense passar-li aquest argument, l'interpret de Python respondria amb la següent sortida d'error per pantalla:

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: saludar() missing 1 required positional argument: 'nombre'
```

Com indica l'error, a la invocació de la funció **saludar()** li falta un argument, tal com aquesta espera que sigui invocada.

NOTA

Python permet inserir variables directament sobre en les cadenes de text, tal com es mostra en la invocació a la funció **print()** de l'exemple anterior. Per a això, només cal prefixar la cadena amb el caràcter **f** i embolicar entre claus **{ }** les variables o expressions que es desitgin incloure en la cadena de text.



Paràmetres per defecte

Per acabar, Python permet definir paràmetres amb valors per defecte en les funcions. Això significa que, si en invocar la funció no es proporciona un valor per a aquest paràmetre, s'utilitzarà el valor per defecte que s'hagi indicat en definir la funció. Per exemple:

```
def saludar(nom="Món"):
    print(f"¡Hola, {nom}!")
```

En aquest cas, si s'invoca a la funció sense cap argument mitjançant `saludar()`, es mostrarà el text «Hola, Món!» per pantalla, ja que el paràmetre `nom` té com a valor per defecte assignat la cadena de text «Món». Però si li passa un argument, com a `saludar("Pere")`, es mostrarà «Hola, Pere!» en el seu lloc. Aquesta tècnica és molt útil per a donar a les funcions un comportament per defecte i evitar així errors com el mostrat en la secció anterior. Per exemple, s'indiquen a continuació alguns casos d'ús:

- **Dotar a les funcions d'un comportament flexible:** els paràmetres per defecte permeten que les funcions es comportin de manera diferent depenent de quanta informació es proporcioni. Per exemple, una funció `potencia(base, exponent=2)` podria calcular el quadrat d'un número si només es proporciona un argument, però calcularia la potència especificada si es proporcionen dos arguments.
- **Simplificar les invocacions de la funció:** en funcions que requereixen molts paràmetres, proporcionar valors per defecte pot simplificar la invocació a la funció, ja que només es necessitaran arguments per a uns pocs paràmetres crítics o obligatoris.



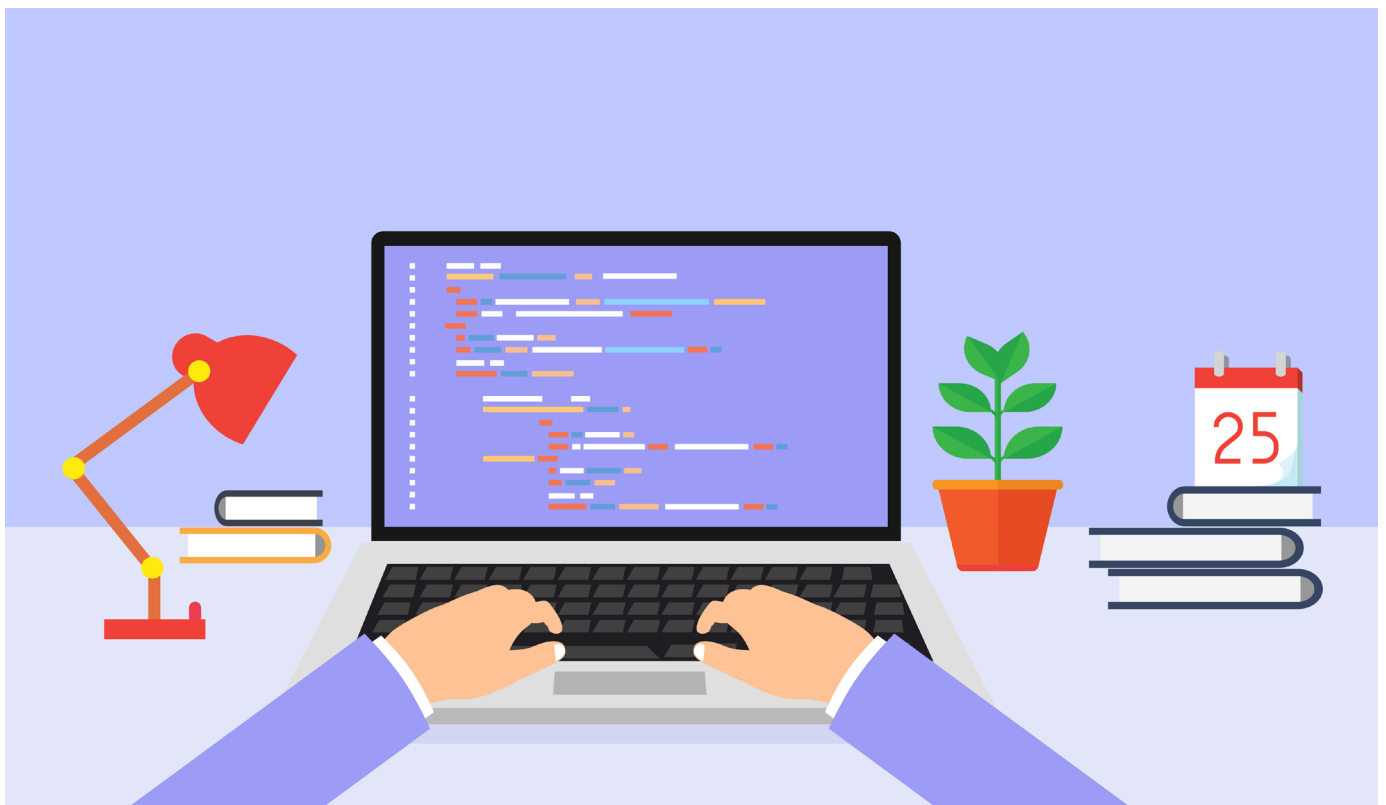
Conclusió

Els beneficis d'utilitzar funcions a Python són múltiples. En primer lloc, permeten la reutilització del codi, la qual cosa millora l'eficiència i la coherència; en lloc d'haver de repetir el mateix codi en diferents parts del programa, es pot definir una funció una vegada i després cridar-la sempre que calgui efectuar aquesta operació específica.

A més, les funcions a Python contribueixen a l'organització i la llegibilitat del codi; en agrupar segments de codi relacionats en funcions, fem que els programes siguin més fàcils d'entendre i de mantenir. Aquest benefici s'amplifica encara més per les capacitats de Python per definir paràmetres per defecte i proporcionar arguments a les funcions pel seu nom, la qual cosa millora la flexibilitat i la claredat de les interaccions amb les funcions.

Saber-ne més

Per saber més sobre les funcions a Python, pots consultar la documentació oficial en castellà: e.digitall.org.es/definir-funciones.





**Creació de
continguts digitals**

Nivell B2 3.4 Programació

**Diccionaris:
definició i usos
principals**





Diccionaris: definició i usos principals

Els diccionaris en Python representen una altra estructura de dades molt convenient per a uns certs tipus de problemes. Al contrari del que succeeix amb una llista, accessibles a través de la posició que ocupa un determinat element dins d'aquesta, un diccionari és accessible mitjançant una clau. En aquest punt, és important remarcar que les claus d'un diccionari en Python han de ser d'un tipus immutable (int, float, decimal, bool, string, tuple, i range). Típicament, el tipus de les claus d'un diccionari serà int o string.

A l'hora de considerar l'ús de diccionaris en Python, aquests es poden entendre com a conjunts de parells clau-valor. Així, les claus d'un diccionari han de ser úniques. Per exemple, si s'utilitza un diccionari per a manejar informació d'un conjunt de persones de manera que la clau sigui el DNI, llavors no és possible emmagatzemar dues persones en el diccionari amb el mateix DNI.

Com el lector pot imaginar en aquest punt, les operacions que més sovint s'utilitzen a l'hora de manejar un diccionari són dues: i) afegir un nou valor amb la corresponent clau i ii) extreure un valor del diccionari, emprant per a això la clau que permet indexar-lo. Evidentment, també és possible eliminar elements d'un diccionari.

El següent fragment de codi mostra un exemple d'ús bàsic d'un diccionari des de l'interpret interactiu de Python:

```
>>> d = {}
>>> d['David'] = 39
>>> d['Albert'] = 34
>>> d
{'David': 39, 'Albert': 34}
>>> d['David']
39
>>> del d['David']
>>> d
{'Albert': 34}
```





És possible iterar sobre els elements d'un diccionari, sent fins i tot possible recuperar, de manera simultània, la clau i el valor de cada element del diccionari. El següent exemple mostra aquesta característica, alhora que exposa una altra manera diferent, i més sintetitzada, de crear un diccionari a Python:

```
>>> d= {'David': 39, 'Albert': 34, 'Xavier': 33, 'Cristina': 30, 'Elena': 23}
>>> for k, v in d.items():
...     print(k, v)
...
David 39
Albert 34
Xavier 33
Cristina 30
Elena 23
```

Com s'ha introduït prèviament, els diccionaris són estructures de dades ideals per emmagatzemar dades on existeix una relació clara entre una referència o propietat única de cada element de l'estructura i el seu valor corresponent.

⚠️ ATENCIÓ

A l'hora de triar una estructura de dades en Python per desar elements que pertanyen a un conjunt, és possible dubtar entre utilitzar una llista o un diccionari. Amb caràcter general, és important recordar que el diccionari és més ràpid que la llista en Python. A més, tant la cerca d'elements com el seu recorregut és més eficient en un diccionari. No obstant això, la llista manté una relació d'ordre entre els seus elements, mentre que això no succeeix en un diccionari.

Exemple d'ús del tipus diccionari a Python

En aquesta secció es mostra un senzill exemple d'ús de diccionaris a Python. En essència, es fa ús d'un diccionari per emmagatzemar i manipular informació relativa a capitals del món.





```
>>> capitals = {'Espanya': 'Madrid', 'Portugal': 'Lisboa', 'França': 'Paris',
'Itàlia': 'Roma'}
>>> print(len(capitals))
4
>>> if 'Espanya' in capitals:
...     print(capitals ['Espanya'])
...
Madrid
>>> print(capitals ['Regne Unit'])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'Regne Unit'
>>> 'Grècia' in capitals
False
>>> capitals.pop('Itàlia')
'Roma'
>>> print(capitals)
{'Espanya': 'Madrid', 'Portugal': 'Lisboa', 'França': 'Paris'}
>>> capitals ['Portugal'] = 'Oporto'
>>> print(capitals)
{'Espanya': 'Madrid', 'Portugal': 'Oporto', 'França': 'Paris'}
```

Quan es tracta d'accedir a un diccionari mitjançant una clau que no es troba en aquest, com passa en l'exemple anterior quan `capitals` s'indexa mitjançant Grècia, l'interpret llança una excepció indicant que existeix un error amb aquesta clau. És possible comprovar si una clau està continguda en un diccionari mitjançant l'operador `in`.

D'altra banda, en l'exemple anterior també s'il·lustra l'ús de l'operació `pop()`, emprada per eliminar una entrada del diccionari a partir d'una clau.

També és possible actualitzar una entrada del diccionari. Per això, simplement ha d'indexar-se a través de la clau, ja existent en aquest, indicant el nou valor. Aquest cas concret s'il·lustra, en l'exemple, amb l'actualització de *la capital de Portugal*.

Finalment, és possible manejar operacions relatives a l'ordenació d'elements en un diccionari. Per exemple, es pot aplicar la funció `sorted()` per obtenir, en ordre ascendent, una llista que contingui les claus d'un diccionari:

```
>>> sorted(capitals)
['Espanya', 'França', 'Portugal']
```



Creació de
continguts digitals

Nivell B2 3.4 Programació

Maneig bàsic d'arxius a Python





Maneig bàsic d'arxius a Python

Python inclou diverses característiques integrades per al maneig d'arxius. Aquestes funcions permeten als programadors crear, llegir, escriure, i manipular arxius de text d'una manera senzilla i eficient. A continuació, s'explorarà el procés de crear i gestionar arxius en Python, amb èmfasis en operacions bàsiques com l'obertura, lectura, escriptura i tancament d'arxius.

Crear un arxiu de text amb diverses línies de contingut

Python utilitza la funció `open()` per a obrir un arxiu. Si l'arxiu especificat no existeix, Python el crearà automàticament. Per escriure en un arxiu, s'utilitza el mètode `write()`. Per exemple, així es crearia un arxiu de text amb diverses línies:

```
# Crear un fitxer anomenat 'exemple.txt'
archivo = open('exemple.txt', 'w')

# Escriure diverses línies al fitxer
arxiu.write('Aquesta és la primera línia.\n')
arxiu.write('I aquesta és la segona línia.\n')
arxiu.write('Finalment, aquesta és la tercera línia.\n')

# Tancar l'arxiu
arxiu.close()
```

Si s'obris ara l'arxiu *exemple.txt* amb un editor de text, es mostraria el següent contingut:

```
Aquesta és la primera línia.
I aquesta és la segona línia.
Finalment, aquesta és la tercera línia.
```

L'argument `'w'` en la funció `open()` indica que l'arxiu s'obre per a escriptura (*write*). Si es tornés a escriure en aquest arxiu emprant aquest enfocament, l'arxiu se sobreescriuria amb el nou contingut, eliminant qualsevol contingut existent. Per evitar aquest comportament, es pot configurar l'obertura de l'arxiu per afegir contingut a un arxiu existent usant l'argument `'a'` (*append*). Per exemple:



```
# Obrir l'arxiu anomenat 'exemple.txt'  
arxiu = open('exemple.txt', 'a')  
  
# Afegir dues línies al fitxer  
arxiu.write('Això és una línia nova.\n')  
arxiu.write('I aquesta és una altra línia afegida a l'arxiu.\n')  
  
# Tancar l'arxiu  
arxiu.close()
```

Ara, en obrir l'arxiu amb un editor de text, es mostraria el següent contingut:

```
Aquesta és la primera línia.  
I aquesta és la segona línia.  
Finalment, aquesta és la tercera línia.  
Això és una línia nova.  
I aquesta és una altra línia afegida al fitxer.
```

Operacions bàsiques: obertura, lectura, escriptura i tancament

Obertura

Com es va esmentar anteriorment, s'utilitza la funció `open()` per obrir un arxiu. Aquesta funció té dos arguments, el nom de l'arxiu i el mode:

```
# Obrir un fitxer per a lectura  
arxiu = open('exemple.txt', 'r')
```

Cal destacar, que el segon argument de la funció `open()` és opcional i que, si no s'especifica, prendrà per defecte el valor `'r'`.

Lectura

Després d'obrir un arxiu, es poden emprar diversos mètodes per llegir el seu contingut. El mètode `read()` llegeix tot el contingut de l'arxiu i el desa en una variable que indicam:

```
# Llegir tot el contingut del fitxer  
contingut = arxiu.read()  
print(contingut)
```



Escriptura

Per a escriure en un arxiu, primer s'ha d'obrir en mode d'escriptura ('w') o d'agregació ('a'). Després, s'utilitza el mètode `write()`:

```
# Obrir un fitxer per a escriptura
arxiu = open('exemple.txt', 'w')

# Escriure a l'arxiu
arxiu.write('Hola, món!')
```

Tancament

Després de treballar amb un arxiu, és important tancar-lo per alliberar recursos del sistema i assegurar-se que l'arxiu es desa correctament en el sistema. Per això, s'utilitza el mètode `close()`:

```
# Tancar l'arxiu
arxiu.close()
```

Per no oblidar tancar l'arxiu quan es finalitzi el treball amb ell, es recomana utilitzar la sentència `with`. Aquesta sentència permet obrir un arxiu i assegurar-se que es tanqui correctament una vegada s'ha acabat de treballar amb ell, fins i tot si ocorren errors durant el procés. Per exemple:

```
# Exemple d'obertura d'un fitxer amb with
with open("arxiu.txt", "r") as arxiu:
    contingut = arxiu.read()
    print(contingut)
```

Quan acabi l'execució del bloc indentat per la sentència `with`, l'arxiu es tancarà automàticament sense necessitat d'haver d'invocar al mètode `close()`.

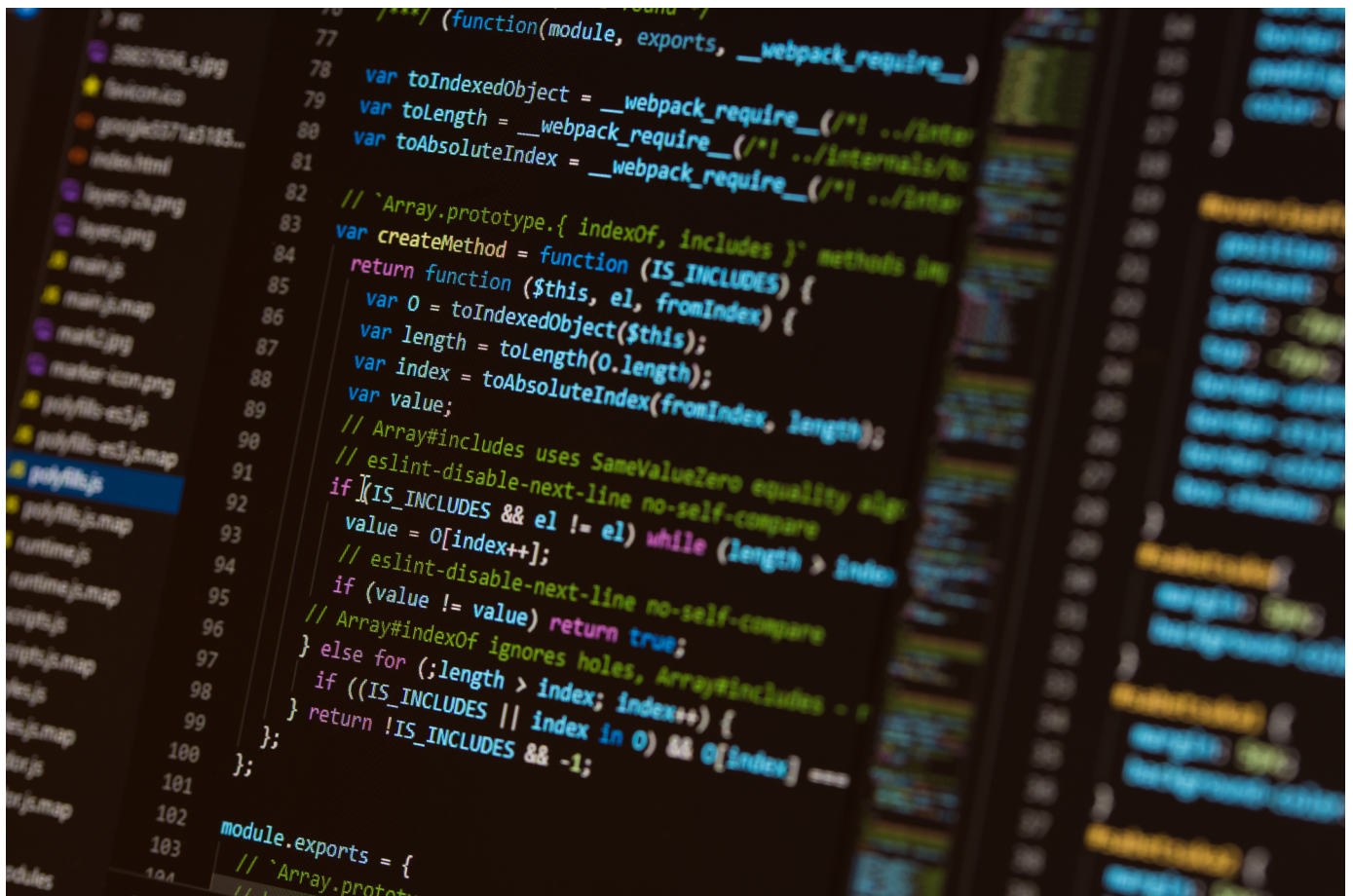


Conclusió

El maneig d'arxius és una tasca essencial i comuna en la programació. Python ofereix funcions i mètodes integrats que faciliten la lectura, escriptura, obertura i tancament d'arxius. Encara que els aspectes discutits anteriorment només cobreixen les operacions bàsiques, Python ofereix moltes més funcions avançades per a treballar amb arxius. Assegura't d'explorar més a fons la documentació oficial de Python per obtenir una comprensió més completa d'aquestes funcions.

Saber-ne més

Per saber més sobre el maneig d'arxius en Python, pots consultar la documentació oficial en castellà: e.digitall.org.es/archivos-phyton.





Creació de
continguts digitals

Nivell B2 3.4 Programació

Què és el programari 2.0?





Què és el programari 2.0?

L'enginyeria programari, o software, és una disciplina que consta d'una sèrie de mètodes, tècniques i eines que permeten el desenvolupament de programari. Fins al moment, l'únic protagonista d'aquest desenvolupament de programari ha estat el programador. La qualitat i eficiència dels programes desenvolupats sempre han depès de l'art i bon fer dels programadors. Aquesta concepció del desenvolupament de programari, centrada en experts programadors, és el que es coneix com a **programari 1.0**. No obstant això, donada la revolució dels últims anys que ha sofert la disciplina de la intel·ligència artificial, especialment àrees concretes com el **Deep learning** o **aprenentatge profund**, aquest enfocament de desenvolupament de programari està canviant, donant pas a nous actors que col·laboraran en el desenvolupament de programes informàtics. Aquests actors són màquines capaces de desenvolupar programari de manera autònoma, o almenys col·laborar en algunes de les tasques de desenvolupament, facilitant així la labor del programador clàssic.

El terme **programari 2.0** va ser encunyat per Adrej Karpathy, responsable de l'àrea d'intel·ligència artificial de l'empresa **Tesla**. La manera de desenvolupament de programari tradicional és lenta i laboriosa; per això, són molts els errors que cometen els desenvolupadors. No obstant això, les **tècniques d'aprenentatge automàtic** dins de la intel·ligència artificial permeten automatitzar part d'aquest desenvolupament. Per això, els ordinadors aprenen a programar mitjançant exemples, és a dir, a partir de programes ja existents ben implementats són capaços d'entendre la seva estructura i replicar-la. Aquest aprenentatge permet l'automatització de tasques de programació, accelerant el procés de desenvolupament i reduint considerablement els errors habitualment comesos.

Resumidament, la fonamentació del **programari 1.0** es basa en l'anàlisi del problema, el disseny dels algorismes i components necessaris i, la implementació i proves dels mateixos per part del desenvolupador. D'altra banda, el **programari 2.0** es basa en la recopilació d'exemples de tasques de desenvolupament que serveixen perquè un ordinador, mitjançant tècniques d'aprenentatge automàtic, aprengui a replicar-les.



Exemple

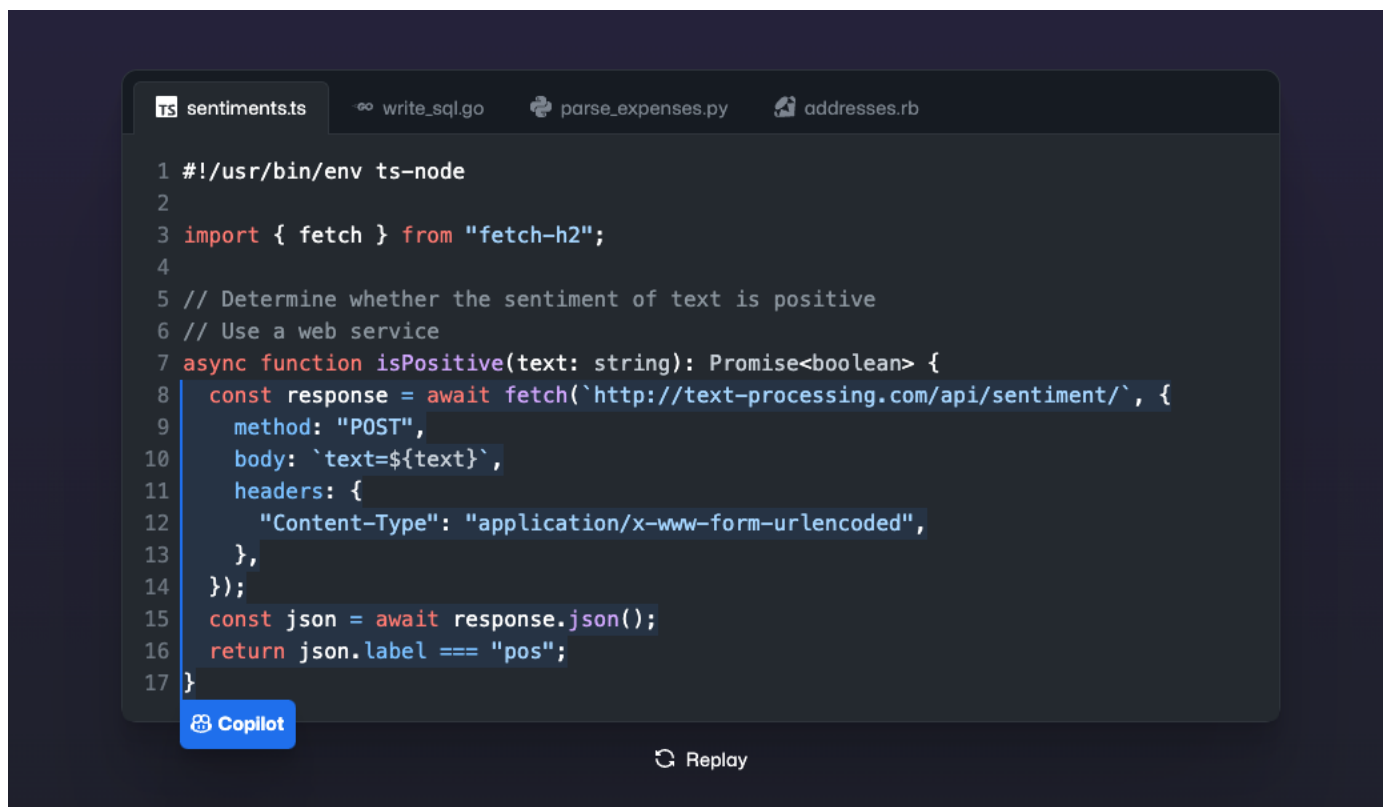
Són moltes les aplicacions que en els pròxims anys es trobaran associades al **programari 2.0** en àrees com el reconeixement visual i de veu, traducció automàtica, jocs, bases de dades, entre altres.

Un exemple és **Copilot** (github.com/features/copilot), una eina que permet a l'usuari descriure en llenguatge natural una funcionalitat que necessita implementar, i aquesta tradueix a codi font la funcionalitat indicada en el llenguatge de programació que se li demani.

Saber-ne més

Aprenentatge Profund. A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," in *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.

Definició Software 2.0: e.digitall.org.es/software-2



```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

Captura de pantalla de Copilot. Font: github.com/features/copilot.



DigitAll

Formació en
Competències
Digitals



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
Odriel 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
Martíño 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



El material del proyecto DigitAll se distribuye bajo licencia CC BY-NC-SA 4.0. Puede obtener los detalles de la licencia completa en: <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>