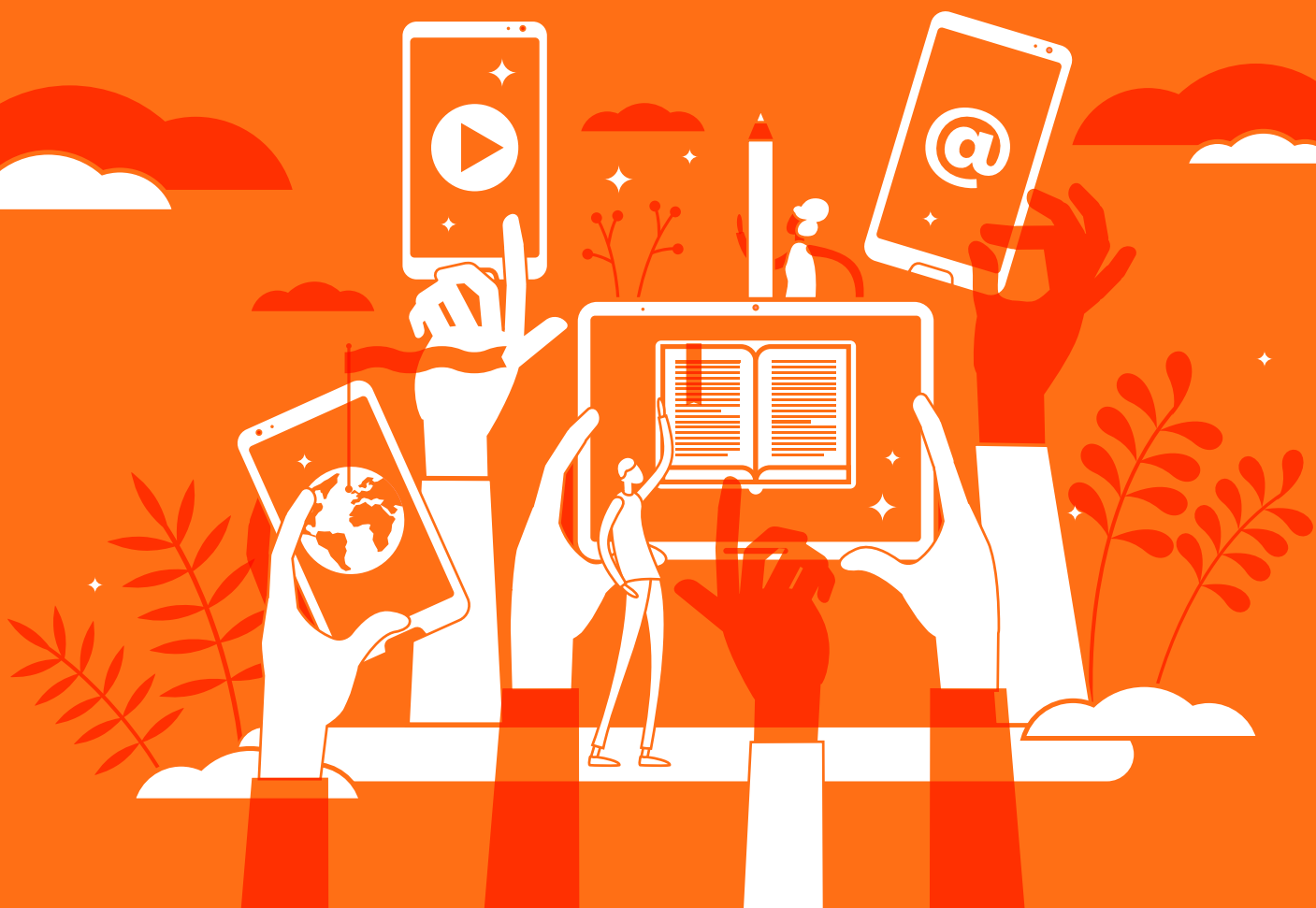




Gaitasun
digitaletan
prestakuntza

3

Eduki digitalak sortzea





Eduki digitalak
sortzea

A1 maila





Eduki digitalak sortzea

AURKIBIDEA

3.1 EDUKIAK GARATZEA

- *Formatu digitalak erabiltzea*
- *Testua prozesatzea eta txantiloiak aplikatzea*
- *Irudiak diseinatzeko oinarrizko tresnak erabiltzea*
- *Geruzetan oinarritutako irudien diseinu egituratua eta maskara bidezko aldaketak*
- *Irudi bektorialak vs irudi rasterizatuak*
- *Irudiak konprimatzea*
- *Bideoa sortzea*
- *Bideoa konprimatzea*
- *Audioa eta konpresioa*
- *Eduki digitalak Interneten*

3.2 EDUKI DIGITALAK INTEGRATZEA ETA BIRLANTZEA

- *Aurkezpenetan testua, irudiak, audioa eta bideoa integratzea*
- *Kalkulu-orriak: adierazpidea eta kalkulua datuekin*
- *Dauden eduki digitalen konposizioa*

3.3 EGILE-ESKUBIDEAK ETA JABETZA INTELEKTUALEKO LIZENTZIAK

- *Egile-eskubideak eta jabetza intelektualeko lizentziak*
- *Plagioa*

3.4 PROGRAMAZIOA

- *Algoritmo baten ezaugarriak eta problemak ebaztea*
- *Fluxu-diagramak*
- *Makina programagarriak. Programaren kontzeptua*
- *Programazio-lengoaiak. Definizioa eta bilakaera*
- *Interpretatzaileak VS konpiladoreak*
- *Adierazpenak eta esleipena*
- *Exekuzio-fluxuaren kontrola*
- *Estilo-gidaliburuak*



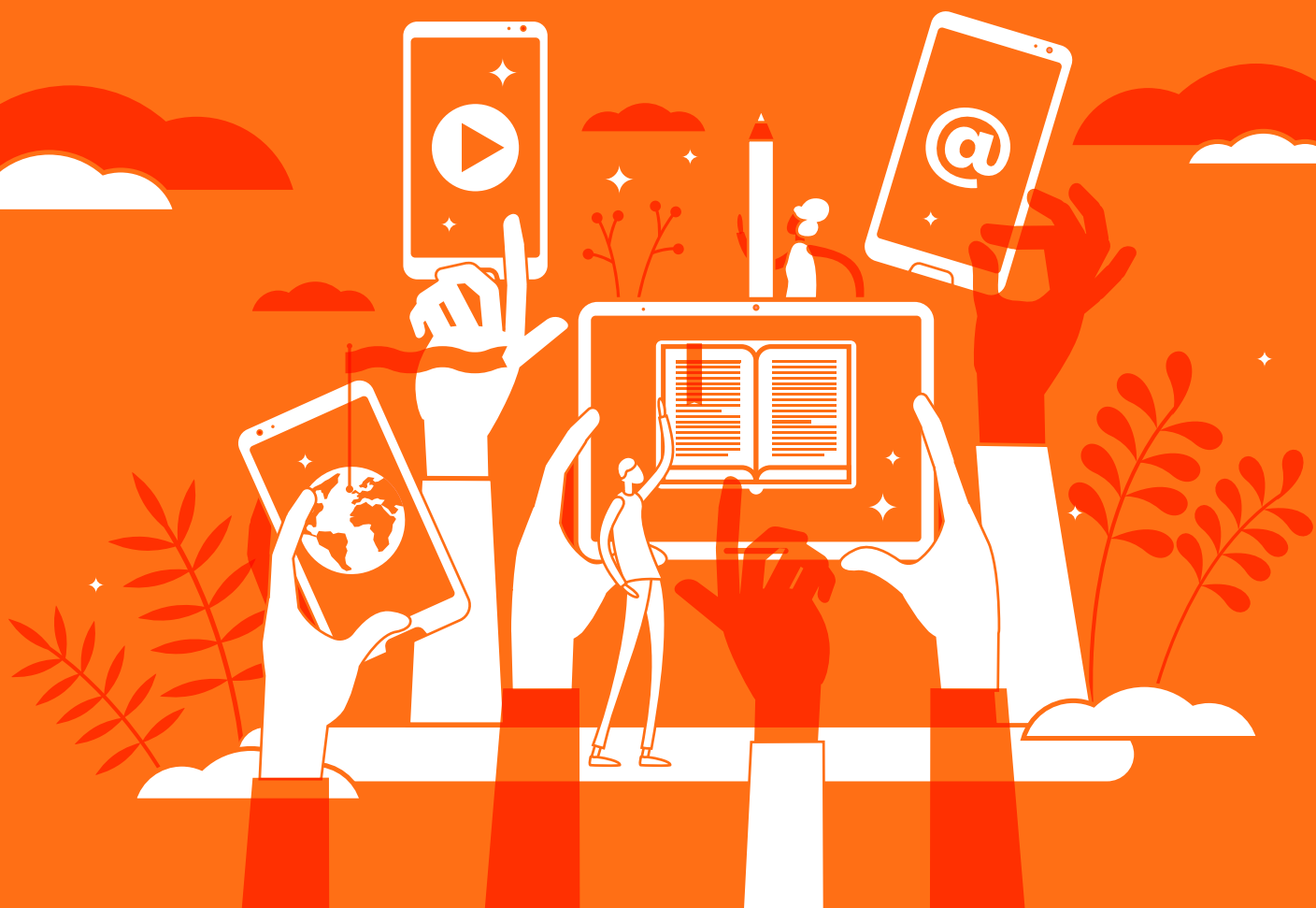


DigitAll

Eduki digitalak
sortzea

3.1

EDUKIAK SORTZEA





Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Formatu digitalak erabiltzea





Formatu digitalak erabiltzea

Eduki digitalak sortzea funtsezkoa da produktu digital bat ezagutarazteko. Eduki bat sortu eta plataforma digitalen batean argitaratzen denean (sare sozialak, webguneak, etab.), harekin lotutako informazioa bilatzen ari denean aurkituko du hartzaileak. Bilaketa-tresnak edozein direla ere, gure edukien ikusgaitasuna bermatuko badugu, kalitate-estandar batzuk bete beharko ditugu, eta behar bezala hautatu beharko dugu haien formatuen artean. Gainera, edukiak zabaltzea funtsezkoa izango da.

Eduki digitalak sortzeko formatuen sailkapen orokorrenak honako hauek hartzen ditu barne: **testua, irudia, audioa eta bideoa**.

Testua giza komunikazioaren formatu nagusietako bat izan da, asmakizun gutxi izan dute, gure kulturaren garapenean, inprentak izaniko adinako eraginik. Hala ere, testuak beti berdina dirudien arren, askotariko hedabide digitaletan argitaratutako testuak aurkituko ditugu –hala nola blogetan edo orrialde kolaboratiboetan (Wikipedia, adibidez) argitaratutakoak–, eta hainbat formatu digital mota ere lantzeko. Kontuan izan behar dugu formatuen arteko bateragarritasuna ez dela erabatekoa; beraz, informazioa gal dezakegu fitxategi mota bat beste bat bihurtzean. Hauek dira formatu ohikoenak:



Formatua	TXT	RTF	ODT	DOC(X)	PDF
Testu aberastua	Ez	Bai	Bai	Bai	Bai
Orrialdeak	Ez	Bai	Bai	Bai	Bai
Irudiak, taulak	Ez	Bai	Bai	Bai	Bai
Oro har, honela irekitzen da	Edozein testu-prozesadore	Edozein testu-prozesadore	Open Office	Microsoft Word	Adobe Acrobat Reader
Erabilera-egoera gomendagarria	Aberastu gabeko testua bakarrik. Edozein egoeratan irekitzeko erraza.	Formatu sinplea, baina testu aberastua onartzen du. Zenbait plataformatan lan egitea.	Formatu librea eta irekia. Edozein bitarteko edo plataformaren bidez edita daiteke	Ohikoena. DOCxek DOCek baino gutxiago okupatzen du.	Ezin da editatu. Dokumentazioa banatzeko estandarra



Irudian, argazkiak edo irudiak aurkitzen ditugu gehien erabiltzen den eduki gisa. Horiek eduki digital askoz interesgarriagoa eta iradokitzaileagoa egiten laguntzen dute, eta hartzailearen esperientzia hobetzen dute. Sare sozialetan eta web-orrietan gehien erabiltzen den baliabideetako bat da; testuari laguntza ematen dio argitalpenetan, eta arreta erakartzen du. Argazkiak, ilustrazioak, infografiak, memeak eta *bannerrak* aurkituko ditugu sareko edozein txokotan. Hona hemen formatu nabarmenenak:

Formatua	JPG	GIF	PNG	TIFF	RAW
Kalitatea	Handia	Ertaina	Handia	Oso handia	Oso handia
Tamaina	Handia	Ertaina	Ertaina	Oso handia	Oso handia
Konpresioa onartzen du	Oso handia	Ertaina	Handia	Handia	Handia
Konpresio-galerak	Bai	Ez	Ez	Ez	Ez
Kolore kopurua	Koloretako 24 bit, zuri-beltzeko 8 bit	256 kolore arte	Koloretako 24 bit	1-64 bit artean	48 bit
Hondo gardena onartzen du	Ez	Bai	Bai	Bai	Ez
Animazioak onartzen ditu	Ez	Bai	Ez	Ez	Ez
Erabilera-egoera gomendagarria	Kamera digitalak, irudiak, inprimaketa, irudi-trukea	Internet, tamaina txikiko irudiak, logoak	Internet, grafikoak, ikonografia, softwarea	Kalitate handiko irudiak, kamera digitalak, eskanerrak, inprimaketa	Kamera digitalak





Audioak eduki mota bat eskaintzen du, hartzaileak egokitzat jotzen duen unean erabil daitekeena, eta horrek balio erantsia ematen du komunikatzeko orduan. Audio-aukeren artean podcasta, elkarrizketak eta audioliburuak daude. Honakoak dira erabilienak:

Formatua	FLAC	MP3	WAV	OGG	WMA
Kalitatea	Handia	Ertaina/txikia	Handia	Ertaina/txikia	Handia/ertaina
Tamaina	Handia	Ertaina	Handia	Txikia	Ertaina
Konpresioa onartzen du	Ez	Bai	Ez	Bai	Bai
Metadatuak onartzen ditu	Bai	Bai	Bai	Bai	Bai
Erabilera-egoera gomendagarria	Kalitate handiko musika-artxiboak	Web edo gailu eramangarrietarako audioak	Web edo gailu eramangarrietarako audioak	Web edo gailu eramangarrietarako audioak	Web edo gailu eramangarrietarako audioak

Amaitzeko, **bideoa** irudia eta audioa bateratzen dituen formatua da, eta sare sozialetan eta web-orrietan hoberen funtzionatzen duen edukietako bat da, proiektu bat ilustratzen edo haren alde gizatiarrena erakusten laguntzen baitu. Sarean honako hauek aurkituko ditugu: produktuen berrikuspenak, tutorialak, zuzeneko transmisioak, iragarkiak eta musika-bideoak. Bideo-formatu nagusiak honako taula honetan jasotzen dira:

Formatua	AVI	MKV	MP4	OGG	MPEG
Kalitatea	Aldagaia	Handia	Ertaina	Handia	Ertaina
Tamaina	Aldagaia	Handia	Ertaina	Txikia	Ertaina
Konpresioa onartzen du	Bai	Gutxi	Bai	Bai	Bai
Erabilera-egoera gomendagarria	Kamera bidez eginiko jatorrizko bideoak biltegitratzea	Kalitate handiko bideoak biltegitratzea	Webguneetarako edo gailu eramangarrietarako bideoak	Ezin hobea Interneteko bideoetarako, duen kalitateagatik/pisuagatik	Interneten konprimatzeko eta erabiltzeko formatu estandarra

Eduki digitalak oso tresna indartsuak dira informazioa transmititzeko orduan, irakaskuntza- eta merkataritza-helburuekin edo beste edozein helbururekin. Formatu egokiak hautatzea funtsezkoa izango da gure proiektua bultzatzeko eta, horrela, ahalik eta hartzaile gehien lortzeko.



Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Testua prozesatzea eta txantiloiak aplikatzea





Testua prozesatzea eta txantiloiak aplikatzea

Dokumentu honetan, testu-prozesadore batek eskaintzen dizkigun oinarriko funtzionalitateak deskribatuko ditugu. Prozesadoreak emandako dokumentu-txantiloien erabilera eta biltegitratzen diren berezko txantiloiak sortzeko aukera nabarmenduko ditugu.

Dokumentu honetan deskribatuko ditugun tresnak sarbide errazeko softwarean daude, hala nola Microsoft enpresaren **Worden**, nahiz eta beste edozein testu-prozesadoretan ere aurkituko ditugun. Abantailarik handienetako bat testuaren eta dokumentuaren formatua edozein unetan aldatzeko gaitasuna da. Barrak botoiak eta zerrendak ditu, testuaren itxurari buruz sarrien aldatzen diren ezaugarri guztietarako zabaltzen direnak.

Irudi honetan, Wordeko tresna-barra ikusten da:



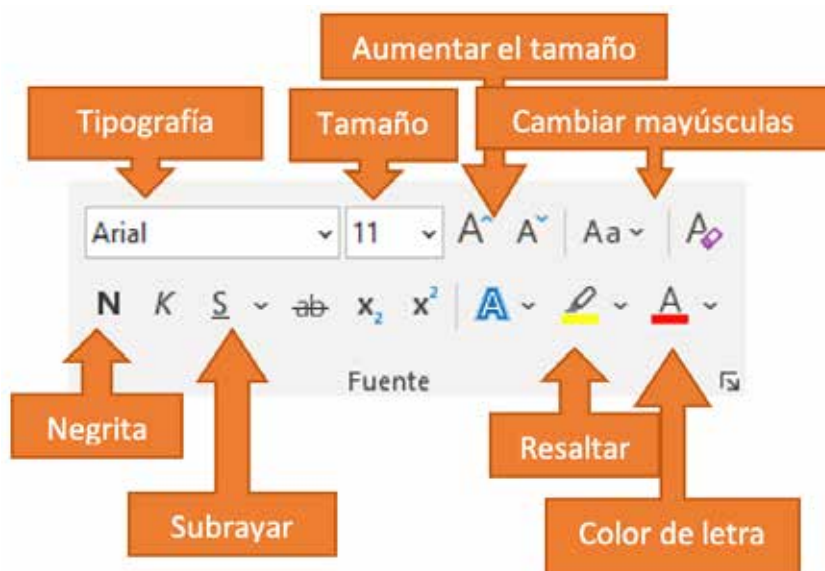
Taula honetan, ondoren deskribatuko ditugun funtzioak laburbiltzen dira:

Blokea	Badu	Onartzen du
Arbela	Ebakitzea, kopiatzea, itsastea, hemendik itsastea.	Biltegitratutako elementu guztiak erakustea.
Iturria	Hautatzea tipografia, letraren tamaina, letra lodia, letra etzana, azpimarratua, marratua, azpiindizea, goi-indizea, itzala, koloretan azpimarratua, letra larrien edo xeheen artean aldatzea.	Iturrien eta karaktereen aukera aurreratuekin pertsonalizatzea. Efektu bisualak, estiloak eta koloreak gehitzea.
Paragrafoa	Zerrenda binetarekin; zenbakikuntza; maila anitzeko zerrenda; koska; testua ezkerrera, erdira edo eskuinera lerrokatzea edo justifikatzea; lerroarte; paragrafoa iluntzea; ertzak gehitzea; hautaketa alfabetikoki edo zenbakiz ordenatzea; paragrafoaren markak erakustea edizio aurreraturako.	Diseinua, tartea, koska eta abar zehaztasunez doitzea.
Estiloak	Estilo lehenetsiak aplikatzea, sortzea edo ezabatzea.	Testu baten itxura administratzea eta pertsonalizatzea.



Eta orain xehetasun pixka bat gehiagorekin ikusiko dugu tresna horietako bakoitza zertarako erabiltzen den, ezkerretik eskuinera doan irudiaren ordenari jarraituz.

- **Arbela:** edozein lekutatik kopiatzen edo ebakitzen diren testuak eta grafikoak gordetzen ditu, eta gordetako elementuak Officeko beste edozein fitxategitan itsasteko aukera ematen du. Lehendabizi, kopiatu nahi dena hautatu behar da saguarekin.
- **Iturria:** iturrien itxura alda dezakegu parrilla honen tresnak erabiliz eta haien gainean klik eginez. Lehenik eta behin, aldatu nahi dugun testua hautatu beharko dugu, eta, gero, tresnak markatuta, tipografia, tamaina, lodiera eta kolorea aldatuko ditugu.



- **Paragrafoa:** iturrien formatua alda daitekeen bezala, paragrafoen itxura alda daiteke. Kasu horretan, hau egin daiteke: testua koskatu, marjinak doitu, taulak sortu, zerrendetan binetak eta zenbakiak gehitu.

Lehen lerroan koska bat gehitzeko, teklatuko Tabuladorea tekla erabiltzen da. Hala ere, paragrafoak lerro bat baino gehiago baditu, koska-markak definitzen dituen tresna erabili behar da, eta koska handitu edo murriztu.





Zenbaitu edo bineta bat jarri nahi duten zerrendak agertzen direnean, tresna hauek erabil daitezke. Zenbaki bat edo bineta bat agertuko da hautatutako paragrafo bakoitzaren aurrean, testua zenbakietatik bereizteko koskekin. Paragrafoak lerro bat baino gehiagoko luzera duenean, bigarrenetik aurrerako lerroak zenbakia duen lehenengoarekin lerrokatzen dira.



Lerroartearen funtzioarekin lerroen edo paragrafoen arteko tarte bertikala gehitzen da. Paragrafoen aurretik eta ondoren espazioak ere gehi daitezke.



Botoi horiekin, lerrokatzea posizio desberdinetan finkatzen da orrialdearen marjinekiko.



- **Estiloak:** estilo-bilduma bat eta beste ezaugarri batzuk ditu dokumenturako. Dokumentu bakoitzak estilo lehenetsiak dituen txantilo bat du zerrenda zabalduan.



Estilo bat alda daiteke, eguneratuz, dokumentuaren formatuarekin bat etor dadin. **Hasiera** fitxako **Estiloak** taldean, eskuineko botoiarekin egin behar da klik aldatu nahi den estiloan, eta Eguneratu [estilo-izena] markatuko da, hautaketarekin bat etor dadin.



- **Txantilo aurredefinituak erabiltzea:** dokumentu bat Worden hastean, formatua eman dakioke testuari tresna-barrak, edo txantilo lehenetsiak erabil daitezke. Txantilo horiek aurrez definitutako informazioa eta diseinu-elementuak dituzte, eta zuzenean erabili ahal izango dugu deskargatzen den edozein txantilo edo nahi dugun bezala pertsonaliza dezakegu eta etorkizunerako txantilo gisa gorde. Nolanahi ere, txantilo horiek estetika atsegina izango dute aurrez definitua, eta, beraz, guk gure dokumentuaren edukia baino ez dugu landu beharko. Dokumentuetarako ez ezik, egutegietarako, aurkezpen-txarteletarako eta askoz gehiagotarako txantiloak ere baditugu.





Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Irudiak diseinatzeko oinarrizko tresnak erabiltzea



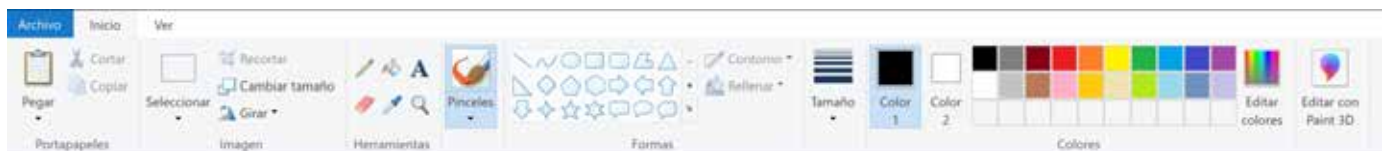


Irudiak diseinatzeko oinarrizko tresnak erabiltzea

Dokumentu honetan, irudiak editatzeko programak erabiltzen ditugunean aurkituko ditugun tresnarik oinarrizkoenak deskribatuko ditugu. Tresna horiek erabilerrazak dira eta eguneroko egoera askotan erabil daitezke. Hala ere, askotan ez ditugu erabiltzen, ez dakigulako zer funtzio dituzten, edo ez dakigulako existitzen diren.

Dokumentu honetan deskribatuko ditugun tresnak erraz eskuratzeko moduko softwarean daude; adibidez, **Paint**, PC bat erabiltzen ari bagara, edo gure smartphonearen galeriatik eskura dezakegun irudiak editatzeko tresna. Bi aukera horietako edozein oso intuitiboa izango da, eta oso eskuragarria irudiak editatzeko/diseinatzeko alorrean lehen kontaktua izateko.

Hurrengo irudian, Painten tresna-barra ikusiko dugu. Tresna horiek erabiliz, gure **mihisean** irudi bat diseinatu ahal izango dugu, hau da, landu dezakegun espazio zurian:





Ondorengo taulan laburtuta ikusiko ditugu ondoren deskribatuko ditugun tresnak:

Blokea	Badu	Onartzen du
Arbela	Ebakitzea, kopiatzea, itsastea, hemendik itsastea.	Informazioa gordetzea.
Irudia hautatzea	Dena hautatzea, hautaketa laukizuzena, hautaketa librea, hautaketa alderantzikatzea, hautaketa gardena, ezabatzea, ebakitzea, tamaina aldatzea, biratzea.	Irudi baten zati bat hautatzea eta harekin lan egitea.
Irudia editatzea	Arkatza, pintzela, kolorezko betegarria, testua, zirriborroa, tanta-kontagailua, lupa.	Esku hutsezko marrazkiak, testua eta koloreztatzea.
Formak	Formen datu-basea, ingerada eta betegarria.	Aurrez definitutako formak sartzea.
Tamainak	Tamainen datu-baseak.	Arkatzaren eta pintzelaren tamaina hautatzea.
Koloreak	Koloreen datu-basea, 1 kolorea, 2 kolorea.	Lan-kolorea hautatzea.

Kontuan izan horiek direla tresnarik oinarrizkoenak, eta tresna aurreratuagoekin lan egingo dugu beste software batean, geroagoko mailetan. Eta orain xehetasun gehixeagorekin ikusiko dugu tresna horietako bakoitzak zertarako balio duen, ezkerretik eskuinera doan irudiaren ordenari jarraituz:

- **Arbela:** izenak adierazten duen bezala, irudi bat edo irudi baten zati bat *moztu* edo *kopiatzeko* eta espazio horretan gordetzeko aukera emango digun testu- eta irudi-biltegia da. Gero, nahi dugun lekuan itsatsi ahal izango dugu informazio hori. *Hemendik itsatsi* aukera ere erabil dezakegu, gure ekipoaren irudi-fitxategi bat aukeratzeko eta mihisean itsasteko.
- **Irudia hautatzea:** gure irudiaren zatiak hautatzeko aukera emango diguten tresnen multzoa dira. Aurreko puntuan «irudi baten zati» bati buruz hitz egingo dugu; orain, hautaketa-tresna horiek erabiliko ditugu irudiaren zer zatirekin lan egin nahi dugun adierazteko. Horrela, dena edo zati bat bakarrik *hautatu* ahal izango dugu, hautaketa laukizuzeneko tresna erabiliz (logikoa denez, laukizuzen forma du) edo *hautaketa librekoa*, eremu bat edozein modutan hautatzeko aukera emango diguna. Ondoren, *hautaketa alderantzikatu* ahal izango dugu aurreko tresnekin markatu ez dugun guztia hautatzeko, hautatu dugun horma-atalaren zatia *ezabatze*ko, edo *hautaketa*

Informazio gehiago

Kopiatzea eta ebakitzea oso antzekoak dira, baina irudi bat edo irudi baten zati bat ebakitzean gure mihisetik ezabatzen ari gara, eta, kopiatzean, jatorrizko informazioa gordetzen dugu.



gardena egiteko, hau da, gure hautaketaren hondoa *gardena dela* adierazteko. Atal honetako gainerako tresnek ere baliabide erabilgarriak emango dizkigute, hala nola *ebakitzea*, gure hautaketa irudi osoa izan dadin, *tamaina aldatzea*, gure irudiari dimentsio berriak emateko, edo *biratzea*, gure irudia inklinatzeko edo iraultzeko.

- **Irudia editatzeko tresnak:** irudi bat nahi bezala editatzeko aukera emango diguten tresnen multzoa dira. Horien artean, esku hutsez marrazteko aukera emango diguten tresnak daude, hala nola *arkatza* edo *pintzela*. Pintzelaren abantaila da zenbait irteera-formatu hautatzeko aukera emango digula, pintzelaren forma eta tamaina aldatuta, egin nahi dugun trazura egokitzeko. Azkarrago margotu ahal izango dugu *kolorezko betegarria* erabiliz; izan ere, ideia bat egiteko, ikonoa pintura-ontzi bat da, eta gure marrazkia kolore batez betetzeko aukera emango digu saguaren klik bakarrarekin. Era berean, gure irudiari *testua* gehitu ahal izango diogu, hainbat iturri, tamaina eta formaturen artean aukeratuta. Huts egin badugu, *zirriborroa* tresna erabiliz zuzendu ahal izango dugu. Eta, azkenik, beste bi tresna ditugu, bloke honetan egon arren edizioakoak berez ez direnak: *tanta-kontagailua*, gure irudiaren kolore bat hautatzeko aukera emango diguna (adibidez, kolore berarekin margotzen jarraitzeko), eta *lupa*, irudia gure beharren arabera handitzen edo urruntzen lagunduko diguna.
- **Formak:** tresna-pakete oso erabilgarria da, aukera emango baitigu aurrez diseinatutako forma pila batera jotzeko, eta ez dugu esku hutsez marraztu beharko, eta hautatu baino ez ditugu egin beharko gure irudiari gehitzeko. Horrela, forma geometriko sinpleetatik poligono librean marrazkietaraino. Gainera, gure formaren *ingerada* eta *betegarria* aukeratu ahal izango ditugu aurrez definitutako aukera batzuen artean.
- **Tamainak hautatzea:** tresna honek aukera emango digu zehazteko nolakoa izango den gure arkatzarekin edo gure pintzelarekin egingo dugun marra lodia edo estua.
- **Kolore-paleta:** azkenean, lan-kolorea hautatzeko eremura iritsiko gara. Hemen, zer kolorerekin margotu nahi dugun aukeratu beharko dugu, adibidez, gure pintzelarekin.



⚠ ADI

Gogoratu marrazkian sartzen dituzun forma horiek oso erraz margotu ahal izango dituzula gero, kolorezko betegarria izeneko tresnaren bidez.



Kontuan izan behar da beti kolore-paletara itzuli ahal izango dugula kolore berri bat aukeratzera, aurretik margotu dugun zerbaitetarako, eta berriro koloreztatu ahal izango dugula (nahi beste aldiz). Hemen aukera batzuk ditugu: begien bistakoa da kolore-paletako edozein kolore hauta dezakegula, edo, bestela, *koloreak editatzeko* atalera joan gaitzkeela, aukera askoz ere zabalagoa izateko. Gainera, kolore pare bat asko erabiltzen badugu, *1. kolorearen* eta *2. kolorearen* artean hauta ditzakegu. Horrela, gure hautaketa gordeta geratuko da, eta ez dugu etengabe bilatu beharko paletan.





Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Irudien diseinu egituratua, geruzetan eta maskara bidezko aldaketetan oinarritua





Geruzetan oinarritutako irudien diseinu egituratua eta maskara bidezko aldaketak

Geruza-maskarez hitz egitean, kontuan izan behar dugu irudi bat modu ez-suntsitzailean editatzeko aukera emango diguten baliabideez ari garela; hau da, ez dugu gure irudiaren jatorrizko informazioa aldatuko, gero geruza-maskara bat ezabatu eta gure jatorrizko irudia berreskura genezake beti. Hori bera aplikatu daiteke, aurrerago azalduko dugun bezala, maskara bektorialetarako.

Horrelako edizioak irudiak editatzeko edozein programatan egin daitezke, eta gure irudiaren gainean geruza gehigarriak gehitzean dautza. Geruza horiek, bertsio errazenean, irudi baten zatiak (adibidez, hondo bat) erakusteko edo ezkutatzeko aukera emango digute. Bestalde, geruza-maskarak irudi oso baten gainean edo zati oso zehatz baten gainean aplikatu daitezke.

Horrela, gure irudiaren zer zati erakutsi edo ezkutatu nahi ditugun adieraziko dugu, geruza-maskararen gainean margotuz, kontuan hartuta geruza-maskarek gris-eskalan lan egiten dutela, hau da, zuriaren eta beltzaren artean koloreak soilik onartzen dituztela. Horrela, maskara zuriz margotu ahal izango dugu, eta orduan maskarak irudi osoa erakutsiko digu (% 100) edo beltza, eta maskarak irudi osoa ezkutatu digu (% 0). Baina tarteko puntu batean ere gera gaitezke (grisean), non maskarak irudi erdi-gardena erakutsiko digun, hau da, informazioaren zati bat ezkutatu digu, hautatu dugun gris-tonuaren arabera. Beste hitz batzuekin esanda, geruza-maskarak **opakutasuna** erregulatzeko aukera emango digu.

Normalean, edizio-programek maskara baten propietateak *erregulatzeko* eta, horrela, haren opakutasuna kontrolatzeko aukera emango digute (batzuetan *dentsitate*a ere esaten zaio), irristagailu motako tresnen bidez, opakutasun-/dentsitate-maila altuetatik (geruza beltzaren % 100, ezkutuan dena) opakutasun-/dentsitate-maila baxuetara igaroz (geruza beltzaren % 0, dena erakusten da).

Informazio gehiago

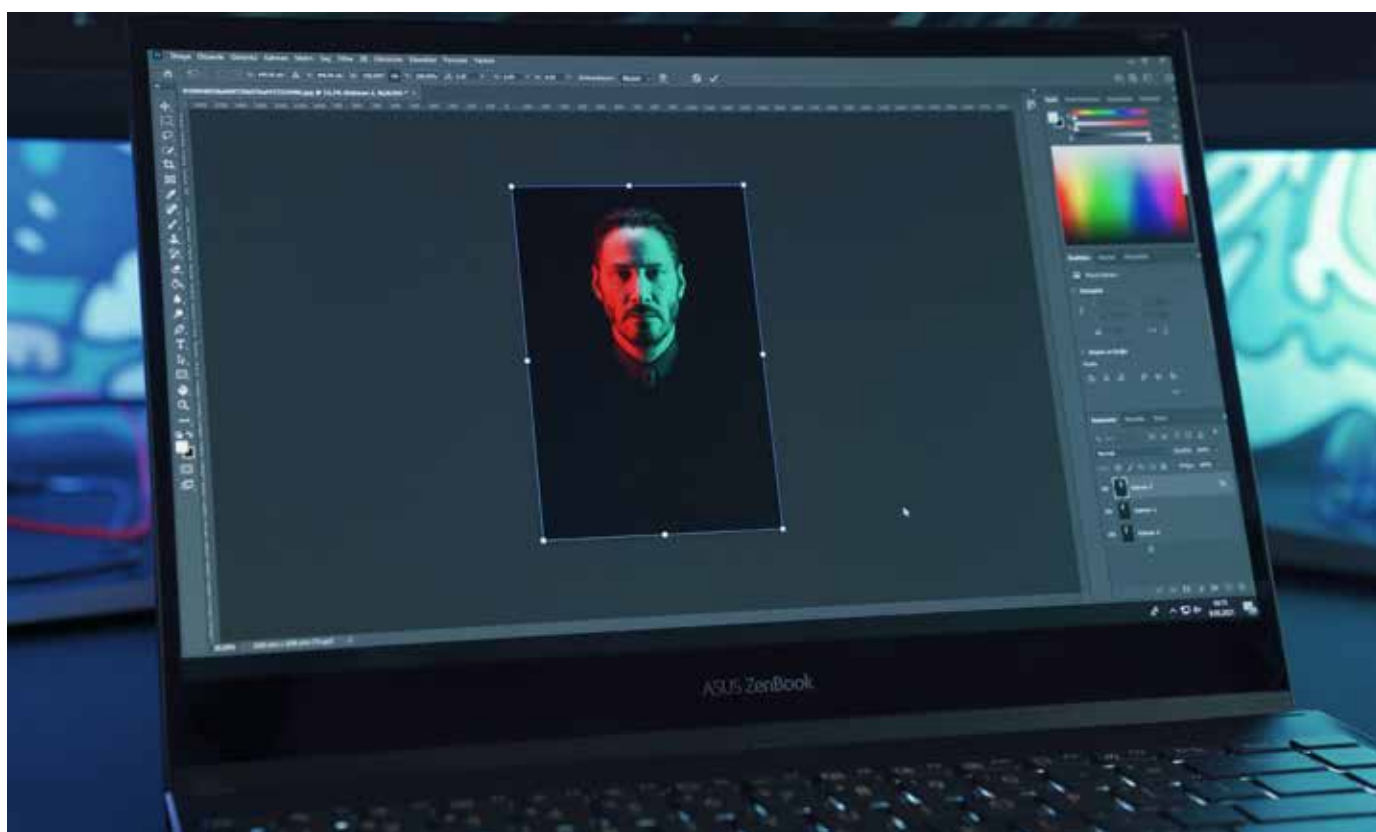
Photoshop, segur aski, edizio-softwarearik ezagunena izango bada ere, bestelako aukerak daude software librearen arloan, hala nola **GIMP** (gimp.org.es), eta aukera ematen dute ordainpeko programa profesionaletan dauden ia edizio-tresna berdinekin lan egiteko.





Hala ere, tresna horiek alde handiak dituzte emaitza bera ematen diguten beste batzuekin alderatuta. Adibidez, irudi baten gainean *zirriborroa* tresna erabiltzen badugu lortuko dugun efektu bera lortuko dugu geruza-maskara bat gehituz eta ezabatu nahi dugun zatia ezkutatuz (beltzez margotuz); baina, zirriborroa aplikatu ondoren, **irudiaren informazioa ezabatu eta betirako galduko dugu**; geruza-maskara erabiliz, berriz, geruza besterik gabe baztertuz bistan ezkutatu dugun informazioa berreskuratuko dugu.

Azkenik, geruza-maskarak edizio-tresna oso indartsuak dira, eta, irudi baten zatiak erakusteko edo ezkutatzeko aukera emateaz gain, edizio-eragiketa batzuk egiteko aukera ere emango digute, hala nola gure irudiaren zati baten gainean *efektu* zehatzak aplikatzea, fotomuntaiak edo konposizioak sortzea edo, besterik gabe, kolore-doikuntzak egitea gure irudi osoan edo haren zati batean (adibidez, zeruan), eta, lehen esan dugun bezala, aldaketa horiek jatorrizko irudia aldatu gabe gertatzen dira, eta, beraz, baztertu eta gure irudia berreskura dezakegu editatu gabe.





Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Irudi bektorialak vs. irudi rasterizatuak

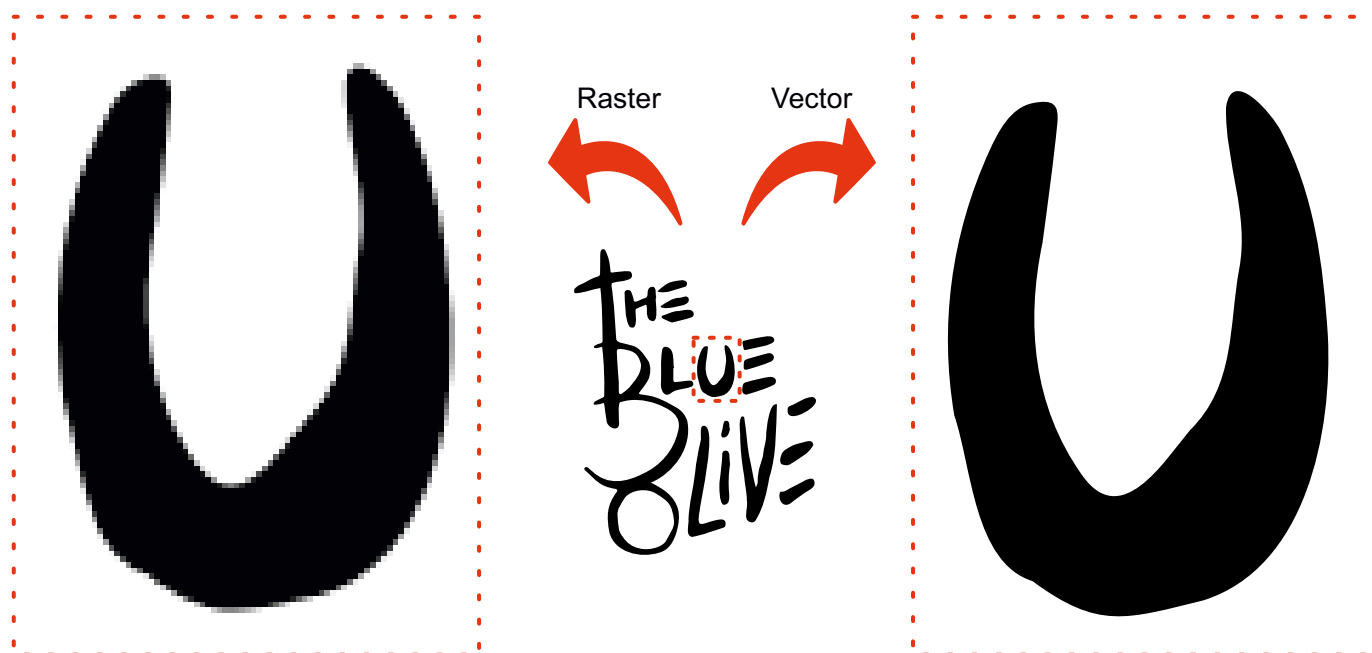




Irudi bektorialak VS. irudi rasterizatuak

Irudi rasterizatuak mosaiko bat osatzen duten pieza txikiz osatutakoak dira. Pieza horiei **pixel** esaten zaie. Zenbat eta pixel gehiago eremu unitate bakoitzeko, orduan eta bereizmen handiagoa izango du irudiak, eta *definizioa* handituko da. **Bit-mapa** ere esaten zaie.

Irudi batek 800x1000 pixeleko bereizmena badu, horrek esan nahi du 800 pixel horizontal eta 1.000 bertikal dituela. Irudi hori pantaila edo inprimaketa batean ikusten badugu, litekeena da giza begiak pixelak ez detektatzea. Hala ere, irudiaren tamaina handitzen badugu edo gehiago inprimatzen badugu, mosaikoa agerikoa izango da. Hala ere, irudi rasterizatuetan ez bezala, **grafiko bektorialak** ez dira pixelez osatzen, baizik eta irudia osatzen duten angeluak eta lerroak zehazten dituzten **kontrol-puntuez**. Bi irudi moten arteko aldeak ikusten dira 1. irudian.



1. irudia. Raster eta bektorearen arteko desberdintasunak.



Irudi rasterizatuak zehatzak dira (argazki-kalitatea) eta aukera asko eskaintzen dituzte sorkuntza grafiko eta artistikoko testuinguruetan, baina espazio asko hartzen dute, eta ez da komeni tamaina aldatzea. Batez ere, handitzea, kalitate-galera nabarmena dakarrelako. Software-tresna ezagunenak hauek dira: Adobe Photoshop, Gimp eta Corel Photo-Paint eta haien png, jpg, gif edo tif formatuak. Bestalde, irudi bektorialeak ez dute kalitatea galtzen handitzen edo murrizten badira, eta automatikoki eskalatzen dute tamaina. Irudi bektorial berak bisita-txartel bat edo publizitate-kartel izugarri bat inprimatzeko balio diezaguke, eta kalitate ezin hobea izango du. Gainera, haien artxiboek ez dute garrantzi handirik izaten. Aitzitik, haiekin diseinuak egiterakoan, zenbait muga aurkituko ditugu rasterrei dagokienez, eta haien itxura ez da argazki-kalitatera hurbilduko. Bektoreetan gehien erabiltzen diren programak Adobe Illustrator, Inkscape eta CorelDraw dira, eta haien formatuak eps, ai, pdf, svg edo sketch dira.





Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Irudiak konprimatzea





Irudiak konprimatzea

Batzuetan, tamaina handiko irudi digitalek zaildu egiten dute mezu elektronikoak eta webgunek erabiltzea; izan ere, denbora asko behar dute kargatzeko, eta erabiltzaileak denbora asko itxaron behar du irudia/informazioa lortzeko. Izan ere, erabilitako irudiek datu asko dituzte, edo, bestela esanda, tamaina handikoak dira, eta, beraz, biltegitratze-espazio handia hartzen dute, eta zaila da transferitzea eta prozesatzea.

Irudiak konprimatzea jatorrizko irudia byte batzuetan murrizteko prozedurak aplikatzea da. Prozesu honen helburua da irudiaren erreduantzia murriztea eta datuak eraginkortasunez biltegitratzea eta transferitzea. Hau da, irudi bat konprimatzea da irudiaren datu erreduantteak eta garrantzirik gabekoak **murriztea, ahalik eta galerarik txikienarekin**, modu eraginkorrean gorde edo transmititzeko. Bikoiztuta dauden datuak ez dira beharrezkoak eta formatu txikian gordez edo ezabatuz informazio digitala gordetzen den fitxategiaren tamaina murrizten da. Prozesu honen bidez, informazio-byteak murriztu nahi dira, irudia degradatu gabe edo irudiaren kalitate nahikoa onargarria lortu gabe.

Konpresio-prozesua «galerekin» edo «galerarik gabe» izango da. Galerarik gabeko konpresioak irudi bat murrizten du, eta haren kalitatea jatorrizkoarekin bat dator; birtualki, ez du inolako informaziorik galduta. Konpresio modu ideala dirudien arren, askotan ez du konpontzen tamainaren arazoa, eta irudiak handiegia da oraindik. Informazioa murrizteko modu hori, oro har, irudiaren kalitatea biltegitratze-espazioa edo web baten karga-abiadura baino garrantzitsuagoa den egoeretan erabiltzen da.

Galerekin konprimatzea, irudi konprimatuak ez du jatorrizkoaren kalitate, tamaina edo informazio bera. Hau da, irudia degradatu egiten da. Okerrena da konpresioa itzulezina dela, eta, irudi bati aplikatu ondoren, ezin dela jatorrizko egoerara itzuli. Irudi bat behin eta berriz konprimatzen bada, distortsioa areagotzen joango da.





Konpresio-formatu desberdinak daude, eta horietako bakoitzak algoritmo desberdina erabiltzen du irudi konprimatuak sortzeko. Ohikoenak jarraian deskribatzen dira:

- **BMP** (ingelesez, *Bit Map Picture*), Microsoften galerarik gabeko konpresio-formatua. 16,7 milioi kolore eta 256 gris-tonu gorde ditzake.
- **PNG** (ingelesez, *Portable Network Graphics*) formatuak tamaina murrizten du, patroiak identifikatuz eta komunak direnak elkarrekin konprimatuz. PNG konpresioak JPEG baino fitxategi handiagoak sortzen ditu, baina batez ere irudiaren kalitatea garrantzitsuena denean erabiltzen da, hala nola logoak, ikonoak, testudun irudiak, marrazkiak eta eskemak dituzten webguneetan.
- **RAW** (ingelesezko hitza, «gordin» itzulita), argazki-kameraren sentsoreak lortutako datu guztiak biltzen dituen formatua, irudia nola hartu den kontuan hartuta. Informazio asko (beraz, irudi-tamaina handia) lortu nahi den argazki profesional batentzat, egokiena da. Horiek erabiltzeko programa espezifikoak behar dira. Informazioa galdu gabe datuak konprimatzeko formatu bat da. Zailtasunik handiena kameraren fabrikatzaile bakoitzak formatuaren bertsio bat erabiltzea da; beraz, estandarizazioa falta da. Horrek erabilera murrizten du eta bateraezintasunak areagotzen ditu.
- **TIFF** (ingelesez, *Tagged Image File Format*) konpresio-formatu bat da, eta galerarik gabeko taldekoa da. Bereizmen eta kalitate handiko irudietan erabiltzen da. Ez da egokia webguneetarako. Industria grafikoan erabiltzen da.
- **WEBP**, Googleren konpresio-formatua, soilik webean erabiltzeko garatua. Formatu oso moldakorra da, galerarekin edo galerarik gabe konprimatzearen artean hautatzeko aukera ematen baitu. Konpresio-sistema honek JPEGrekin lehiatu nahi du webguneetarako irudiak konprimatzeko orduan.
- **GIF** (ingelesez, *Graphic Exchange Format*); konpresio-formatu hau galerak dakartzan edo galerarik gabeko konpresio-multzotzat har daiteke. Sailkapen anbiguo hori murriztu nahi den irudiaren arabera da. GIFek 256 koloreko muga du. Jatorrizkoak 256 kolore edo gutxiago baditu, konpresioa galerarik gabekoa izango da. GIF formatua galera-multzoan sailkatzen da, jatorrizkoak 256 kolore baino gehiago baditu. Bideo eta animazio sinpleetarako formatua da.





- **JPEG** (ingelesez, *Joint Group of Photographic Experts*) webguneetan eta argazki digitaletan modu estentsiboan erabiltzen den formatua da; harekin konprimatzeak galerak dakartza, baina asko erabiltzen da tresna eta aplikazio ugarietan. Konpresio-maila (eta, beraz, galera-maila) hautatzeko aukera ematen du.



Formatua	BMP	PNG	RAW	TIFF	WEBP	GIF	JPEG
Konpresio mota	Galerarik gabe	Galerarik gabe	Galerarik gabe	Galerarik gabe/ aukerakoa galerarekin	Galerarekin/ galerarik gabe	Galerarekin/ galerarik gabe	Galerarekin
Tamaina	Oso handia	Handia	Oso handia	Oso handia	Oso handia	Oso txikia	Txikia
Koloreak	Oso ona	Oso ona	Oso ona	Oso ona	Oso ona	Nahikoa	Oso ona
Honetarako gomendatua	Argazkiak	Atzealde gardena duten irudiak	Argazkiak	Argazkiak	Webgunea	Grafikoak, ilustrazioak, animazioak	Argazkiak

Argazki-irudi bati formatua emateko aholkuak:

- 1** | Ahal bada, hartu irudia RAW formatuan, baliteke datu-txartelak erosi behar izatea gailuan/kameran informazio gehiago gordetzeko.
- 2** | Gorde lana edo amaierako irudia TIFFen.
- 3** | Erabili JPEG jatorrizko irudia beste batzuekin partekatzeko.
- 4** | Ez gorde irudia JPEGn.
- 5** | Gorde lana WEBPen irudiak webguneetan erabiltzeko.



Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Bideoa sortzea





Bideoa sortzea

Txikitan denok egin genituen marrazki txikiak koaderno bateko orrialdeetan, eta orrialdeok azkar pasatuz marrazkia animatu eta mugitu egiten zen. Filoskopia esaten zaio horri, eta bideoa zer den irudikatzen du; izan ere, bideo-sekuentzia bat honela uler dezakegu: **irudien segida** bat, zeinean irudiak gainjartzen baitira gure burmuinak mugimendurik ez dagoen lekuan mugimendua hautemateko behar bezain azkar.

Modu digitalean, zenbait bide daude bideoa lortzeko: animazio-forma klasikoa erabiliz, hau da, programa batek bideo bihurtuko dituen irudien bitartez [adibidez, **Blender** (blender.org)] edo guk geuk grabatuz.

Gure bideo-grabazioak egiteko, **grabazio-ekipo** bat behar dugu. Sistema horiek asko aldatuko dira bilatzen dugun azken emaitzaren arabera, baina guztiek partekatzen dute *bideoa* grabatzeko gai izateaz gain, *soinua* erregistratu eta integratuko dutela. Hau da, gutxienez **kamera** batez eta **mikrofono** batez osatuta egongo dira.

Logikoa denez, ekipo horiek elkarren arteko aldaketa asko izango dituzte, eta horren arabera izango da bideoaren kalitatea. Irudiaren kasuan, **bereizmenari** buruz hitz egingo dugu. Hala, ohikoak dira jada bideoa grabatzeko aukera ematen diguten telefono mugikorrek, baina sistema horiek mugatuta daude, batetik, mugikorraren neurriengatik, eta, bestetik, biltegiratze-dukieragatik. Hau da, kalitate txikiko (edo bereizmen txikiko) bideoak grabatzeko prestatuta daude; izan ere, telefonoak txikiak izan daitezten, ezin dute goi-mailako grabazio-sistematik eduki, baina, edukiko balituzte, bideoaren kalitateak biltegiratze-ahalmena gaindituko luke.

Bestalde, grabazio-sistema profesionalagoek, hala nola bideokamerek edo reflex kamerek, kalitate/bereizmen handiko bideoa grabatzeko aukera emango digute, bideoa eta audioa atzitzeko sistema hobekaituzte, baina, gainera, funtzionalitate gehigarriak dituzte, hala nola *irudi-egonkortzaileak*, dardarak eragozteko, edo infragorri-sistemak gauzez grabatzeko.

Informazio gehiago

Amesgaiztoa Eguberri aurretik (1993) kulturko filma *stop-motion* izeneko teknika baten bidez sortu zuten; hau da, teknika horren bitartez objektu estatikoen irudi sekuentzialak hartzen dira mugimendua itxuratzeko.





Bideo bat grabatu ondoren, programa batzuk erabil ditzakegu, lortutako emaitza modu digitalean **editatzeko**, era horretan, bilatzen dugunarekin bat datorren bideoa lor dezagun. Grabazioan gertatzen zen bezala, askotariko bideo-edizioko programak aurkituko ditugu. Puntu honetan, kontuan izan behar dugu bideo-edizioaren faktore mugatzaileetako bat dela zer bereizmenekin grabatu den. Beraz, bideo bat bereizmen handiarekin grabatu bada, haren kalitatea murriztu ahal izango dugu beti, baina ezin izango dugu bereizmena handitu bideoa bereizmen txikiarekin grabatu bada.

Hala, editore sinpleenek, telefono mugikorretan edo grabatzeko ekipo eramangarrietan aurkituko ditugunek, ez digute aukera handirik emango. Adibidez, ohikoa da editore horiek sistema bat izatea, interesatzen ez zaizkigun bideo-zatiak *ebakitzeko*, gutxienez.

Hala ere, edizio osoago bat egin nahi badugu, hala nola *grabazio-angelua aldatzea*, irudiaren koloreak *doitzea*, *iragazkiak* erabiltzea edo *testua txertatzea*, PC batean egin beharko dugu, eta bertan edizio-programetara jo ahal izango dugu [adibidez, **Kdenlive** (kdenlive.org/es)], eta haien bidez bideoa nahierara aldatzeko aukera izango dugu.





Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Bideoa konprimatzea

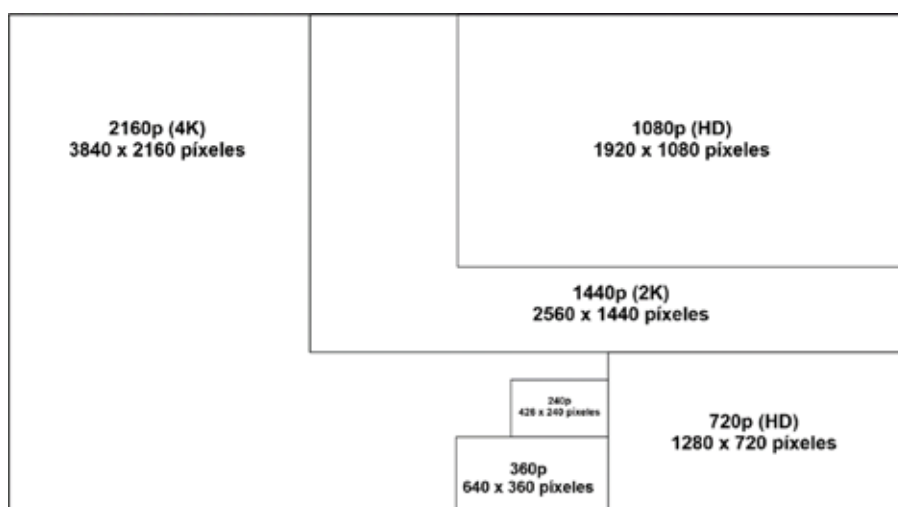




Bideoa konprimatzea

Bideoari buruz hitz egitean, komeni da haren kalitatea, bereizmena eta tamaina bereiztea; hiru kontzeptuok oso lotuta daude elkarren artean.

Bideo baten bereizmenak zehazten du zenbat xehetasun dituen eta zer araztasun duen. 16:9ko itxura estandarreko erlazioko pixel kopuruaren arabera neurtzen da; ohikoena pantaila, monitore eta telebistetan. Bereizmenak bideoaren **kalitatea** zehazten du, pixelen erlazioa zenbat eta handiagoa izan, orduan eta hobeto ikusiko baita bideoa. Azken urteotan, bereizmenak gora egin du 240p-etik (kalitate baxua) eta 4K-ra (bereizmen handiko bideo-formatua). Bideoa zer gailutan erreproduzitzen den alde batera utzita, zenbat eta bereizmen handiagoa izan, orduan eta aratzago ikusiko da. Beti egin daiteke irudi txikiagoa, kalitatea galdu gabe, baina irudi handiagoak egiteak kalitatea galtzea ekarriko du. Irudi honetan ikusten da bereizmenen arteko erlazioa:



1. irudia. Bideo-bereizmenak.

Tamainaz hitz egitean, bertan dauden pixelen kopuruaz (bereizmena) edo bideo-fitxategiak ordenagailuaren disko gogorrean gordetzean hartzen duen biteko tamainaz arituko gara. Horrela, **konpresioaren** kontzeptua erabiltzen da, bideo-fitxategiak ahalik eta informazio gutxien erabiliz optimizatzeke. Baina konprimatzeak zenbait arrisku dakartza artxiboaren kalitateari dagokionez, eta, beraz, konpromiso bat hartu beharko da.



Konpresioa areagotzean, irudiaren kalitatea gutxitzen da. Konpresio handiak esan nahi du antzeko koloretako pixelak kolore bakarra bihurtzen direla; beraz, araztasuna galdu egiten da, eta irudia ez da hain zehatza. Gainera, irudien konpresioa denborazko mugimenduan konpentsatu behar da, nahi ditugun baldintzak betetzen dituen bit-tasa bat mantentzeko (irudi-fluxua). Bideoa konprimatzeak *streaming* bidezko datuen transmisioa hobetzea ere ahalbidetuko du, bideo batean inplikaturako datu kopuruak gaur egungo hardwarea prozesatzeko gaitasunak gainditzen baititu. Horrela, transmisio eraginkorragoa lortzen da.

Bideoa igorlearentzat irisgarria izango dela bermatzeko, bideo-fitxategiaren formatu zuzena aukeratu beharko da. Eta hori ez da ordenagailuan formatu mota definitzea bezain erraza; izan ere, kodetze motak (codec), bideo-fitxategien edukiontzia eta bideo-formatu motak baloratu beharko dira. **Fitxategiaren formatuak** zeregin erabakigarria izango du; izan ere, bideo-fitxategi batek irudiak ez ezik audioa ere hartzen ditu barne, eta baliteke metadatuak izatea (hala nola azpitoluak, menuen egiturak edo fitxategiaren informazioa). Beraz, bideo-formatuan, alde batetik, bideoa konprimatzen eta deskonprimatzen duen **codeca** hartzen da kontuan, eta, bestetik, guztia elkartzen duen eta edozein erreproduktzio-softwaretan funtzionarazten duen **edukiontzi** bat.

Bideo-codeca bideoa kodetzeko eta deskodetzeko protokolo bat da. Audio- edo bideo-fitxategi baten datuak diseinatzeko erabiltzen den ordena definitzen du, erreproduzitu eta/edo editatu ahal izateko. Edukiontzian dauden baliabideen datuak antolatzen ditu. Askotariko audio- eta bideo-codecak daude, bakoitza bere abantaila eta desabantailekin. Ohikoenak **H.264**, **MPEG-4** eta **DivX** dira. Codecak efizienteak dira, eta kalitatea zaintzeko eta tamaina murrizteko gaitasun ona dute. Edukiontzi-formatu erabilienak **AVI**, **MP4** eta **MOV** dira. Hainbat codecekin erabil daitezke, erreproduzitu diren gailuen eta softwareen arabera.

Codecaren azken funtzioa bideoa konprimatzea da (gutxiago okupatzeko) eta erreproduzitzean deskonprimatzea (ondo ikusteko). Eta konpresio hori galerarekin edo galerarik gabe izango da. **Galerarekin konprimatuz** gero, tamaina txikiagoko fitxategiak lortuko dira, datuak galduta, eta ondorioz, bideo-kalitate okerragoa duten fitxategiak izango ditugu.





Hori oso nabarmena da konpresio metagarrietan (aldez aurretik konprimatutako zerbait konprimatzen denean). **Galerarik gabeko konpresioak** jatorrizko artxiboko datuak mantentzen ditu, bideo-kalitate hobea eskaintzen du eta artxiboa degradatzea eragozten du. Desabantaila gisa, tamaina handiagoa hartzen dute.

Gehien erabiltzen diren bideo-formatu batzuk:

- **MP4** bereziki erabilgarria da edukia linean partekatzeko. Gomendagarriena da YouTubera kalitate oneko bideoak igotzeko. H.264 edo H.265 codecarekin konbinatu ohi da. Tamaina txiki samarra izaten dute, eta kalitate ona izaten dute.
- **AVI** formatu estandar nahiko zaharra eta unibertsalki onartua da. Hainbat codecekin bateragarria da, eta fitxategi-konfigurazio ugari ahalbidetzen ditu. Erreproduzitzailerik gehienetan erreproduzitu daitezke. Artxibo-tamainak handiak izaten dira, eta ez dira hain gomendagarriak transmititzeko edo deskargatzeko.
- MKVk audio- eta azpititulu-kanal edo -pista batzuk biltegitratzeko aukera ematen du, oso kalitate onarekin eta oso leku laburrean. Gainera, formatu irekia da, eta ez da eskubiderik ordaindu behar.
- **FLV** oso formatu arrunta da, fitxategi-tamaina txikia duelako eta Flash esploratzaile eta erreproduzitzailerik gehienekin bateragarria delako. Gaur egun gutxiago erabiltzen da.
- **MOV** Appleren bideo-formatu bat da, Quicktime irakurgailurako garatua. Oso kalitate handikoak dira, eta artxibo-tamaina handia izaten dute. Aurrekoek baino bateragarritasun txikiagoa du.
- **WMV** Windows Mediaren bideo-formatua da. Tamaina txikikoak dira, eta aukera ona dira bidaltzeko, baina ez dute ezaugarri berezi nabarmenik.

Formatuak aldatzeko askotariko bihurgailuak daude. Hona hemen horietako batzuk: MiniTool Movie Maker, iMovie (Mac), Video Converter (Android), VLC Media Player eta Online Videoconverter.





Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Audioa eta konpresioa





Audioa eta konpresioa

Ikus-entzunezko edukiak sortzeko prozesuan musikak edo audioak, oro har, norberaren irudiak baino garrantzi handiagoa du, zalantzarik gabe. *Storytelling* prozesuan, edo istorio bat formatu horretan kontatzeko, garatzeko eta egokitzeko artean, audio-tresnak erabili behar dira hartzaileari zirrara eragiten dion mezu sendoa eraikitzea lortzeko. Musika, soinu-efektuak edo narrazioa funtsezko elementuak dira publikoaren emozioetara jotzeko. Baliabide horiek eraginkortasunez erabiltzeak erraztuko du entzulearen arreta gure edukian atxikitzea.

Audio-konpresioaz ari garenean, zehaztu behar dugu fitxategiaren tamaina murrizteaz ari garela, audio-seinale digital baten bit-tasa murrizteaz, txikiago bihurtzeko, jatorrizko informazio ia guztia kontserbatuta. Konpresio-mailaren arabera, kalitate-galera hautemanezina edo oso nabarmena izango da.

Garrantzitsua izango da audioaren tamaina kontrolatzea, gailuetan espazio gutxiago izan dezan eta, beraz, errazagoa izan dadin banda-zabalera txikiagoetara transmititzea.

OHARRA

Musika-ekoizpenaren esparruan, audio-konpresioaz hitz egitean, ez gara fitxategiaren tamaina murrizteaz ari, baizik eta audio-seinale baten lerrun dinamikoa kontrolatzeko aukera ematen duen prozesatzeaz. Musikaren testuinguruan, dinamika abesti baten zati sendo eta leunen arteko desberdintasuna da. Audioaren testuinguruan, zati sendo eta leunen orde, seinalearen tontor indibidualak eta depresioak daude. Barruti dinamikoa handia izango da gailurren eta haranen artean alde handia badago, eta txikia, aldea txikia bada. Eta maila horiek konpresore bat erabiliz erregula daitezke.

Zenbait formatu mota daude audioaren konpresioaren arabera. Horiei buruz hitz egiteko, konprimatu gabeko audio-formatuak eta konprimitutako audio-formatuak (galerarekin eta galerarik gabe) bereiziko ditugu.

OHARRA

Probatu zirrara eragin dizun bideo bat audiorik gabe ikusten. Adibide klasiko bat: *Psikosis filmeko dutxako eszena musikarekin eta musikarik gabe* (e.digitall.org.es/psicosis) (Alfred Hitchcock, 1960).





Konprimatu gabeko audio-formatuak

Hi-Res edo bereizmen handiko fitxategiak ere esaten zaie. Jatorrizko audioaren informazio guztia gordetzen dute, modu digitalean prozesatuta eta biltegitratuta. Fitxategi horiek audioaren kalitate eta fideltasun handiena ematen dute, eta biltegitratze-espazio asko hartzen dituzte. Hona hemen ohikoenak:

- WAV (Waveform Audio Format) konprimatu gabeko formatu ohikoena da, eta industria profesionalak zein erabiltzaileek erabiltzen dute. Konprimatu gabeko audioa izaten du PCM formatuan, eta formatu egokia da Windowserako (Macek ere erreproduzitzen du).
- AIFF (Audio Interchange File Format) Appleren konpresiorik gabeko beste formatu bat da, eta PCan ere erreproduzitu daiteke. Fitxategi gehienek PCM formatuan konprimatu gabeko audioa dute.

OHARRA

PCM formatua da uhin analogiko bat baliabide digitaletan erreproduzitu daitekeen uhin digital (bit-sekuentzia) bihurtzeko gehien erabiltzen den kodeketa.

Galerarik gabe konprimatutako audio-formatuak

Fitxategien tamaina konprimatzea lortzen dute, audioaren kalitate orokorra galdu gabe. Espazio digitalaren konpresio edo aurrezpen maila ez da galerekin eginiko konpresioak ematen diguna bezain handia, baina murrizketarik badago konpresiorik gabeko formatuekin alderatuta. Hauek dira ohikoenak:

- FLAC (Free Lossless Audio Codec), konpresio-formatu arrunta musikaren esparruan, galerarik gabe. Kode irekikoa da eta metadatu txertatuak ditu (albumaren informazioa, etab.). Jatorrizko fitxategiaren erdia hartzen dute, gutxi gorabehera.
- ALAC (Apple Lossness Audio Codec) FLACen oso antzekoa da, baina Applek garatu du. Kode irekikoa ere bada, Applerena izan arren.





Galerekin konprimatutako audio-formatuak

Formatu horiek kalitatea sakrifikatzen dute tamaina minimizatzeko. Pisu txikia dute eta edozein gailutan erreproduzitu daitezke, baina soinu pobrea eta distirarik gabea sortzen dute. Ez dira eremu profesionaletan erabiltzen, soinuaren fideltasuna galtzen baita. Hona hemen ohikoenak:

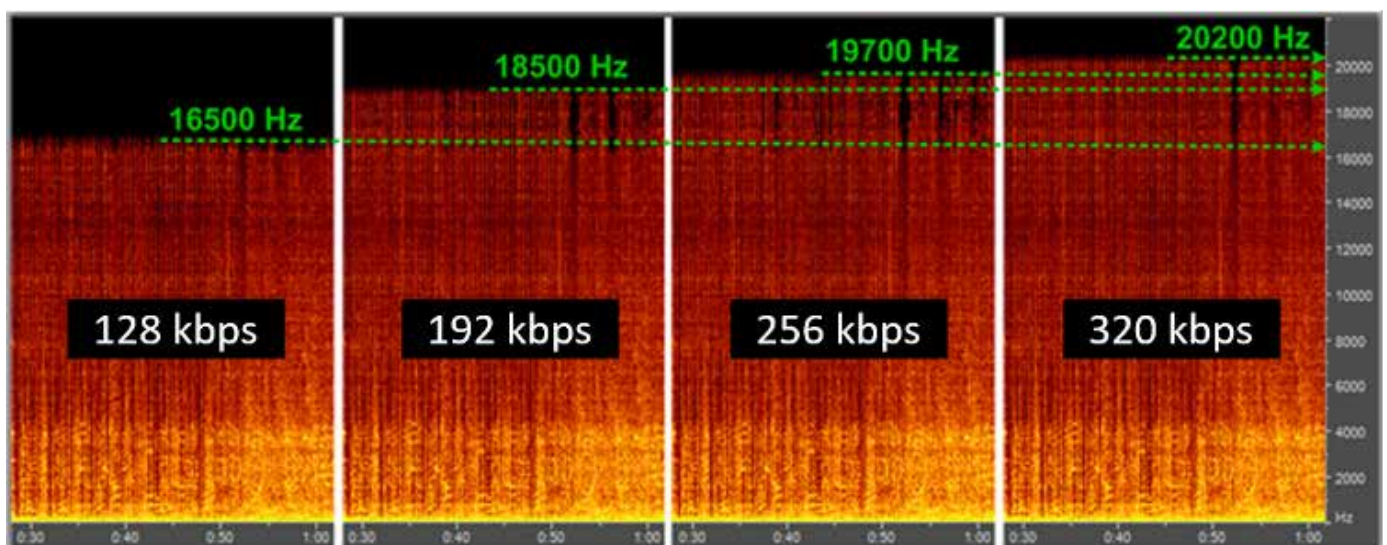
- **MP3** (Moving Picture Experts Group) da ezagunena galera dakarten formatuetan. Bere algoritmoak giza belarran erraz entzun daitezkeen zatiak edo maiztasunak ezabatzen ditu. Konpresioa zenbait kb/s mailatan (kilobit segundoko) egin daiteke, eta kalitate handienekoa 320 kb/s-koa da.
- **AAC** (Advanced Audio Coding), MPEG-4 izenez ere ezaguna. Galeradun formatu konprimatua da, eta iTunesek deskargetan eta YouTube audioaren *streaminga* kudeatzeko erabiltzen dute.
- **OGG Vorbis**: kode irekiko galeradun formatu konprimitua da, eta asko erabiltzen da streaming plataformetan, hala nola Spotifyn, banda-zabalera aurrezteko (ideia bat egiteko, Spotifyren doako bertsioak 128 kb/s-an erreproduzitzen du, eta premium bertsioak, berriz, 320 kb/s-an).
- **WMA** (Windows Media Audio), Microsoftek sortutako formatua. Aurreko biek bezala, MP3 konpresio-metodoaren akats batzuei antzeko ikuspegiarekin heldu nahi die. WMAREN termino objektiboetan MP3k baino kalitate handiagoko konpresioa duen arren, gailu askok ez dute formatu hori onartzen. Era berean, ez du inolako onurarik ematen ZBO edo GEOen aldean.





Zer audio-formatu da egokiena?

Audioa zertarako behar dugun, horren arabera izango da. Edonola ere, prozesatu gabeko audioa atzitzen eta editatzen ari bagara, konprimitu gabeko formatu bat erabiltzea izango da aukerarik onena. Horrela, ahalik eta audio-kalitatearik onenarekin lan egiten ari garela ziurtatuko dugu. Amaitzean, beti konprimatu ahal izango dugu (baina hori ezin dugu alderantziz egin). Audio-irudikapen zehatza gorde nahi badugu, komeni da galerarik gabeko audio-konpresioa erabiltzea. Horregatik, melomanoek nahiago dute FLAC formatua MP3 formatua baino, nahiz eta biltegiratzeko espazio gehiago eduki behar den. Azkenik, artxiobak musikarik ez badu, espazioa aurreztu behar da, eta kalitatea parametro sakrifikagarria da; audioa galerekin konprimitu daiteke. Errealitatean, pertsona gehienek ez dute galeradun eta galerarik gabeko konpresioa bereizten. Oso zaila da desberdintasunak hautematea kalitatezko ekipo edo gailuekin entzunez gero, konpresioak giza entzumenaren mugan baitaude. Hala ere, gama altuko ekipoekin edo entzungailu profesionalekin eta entzumen pixka bat hezia dutenekin, diferentziak bereiz daitezke audioaren ñabarduretan. Eta, jakina, espektrometro batekin uhin-luzerak neurtuta, alde nabarmenak ikusiko ditugu.



Audio-konpresioaren adibidea. Ikusten den bezala, ezkerretik eskuinera, MP3ko fitxategi batean, maiztasun akutuen muturra (Hz-n neurtuta) handiagoa da konpresio-maila zenbat eta txikiagoa izan (kb/s gehiago).



Eduki digitalak
sortzea

A1 maila 3.1 Edukiak
garatzea

Eduki digitalak Interneten





Eduki digitalak Interneten

Eduki digitalak funtsezkoak dira garai modernoetan edozein komunikazio-estrategia garatzeko, dela irakaskuntza-ikuspegia, dela marketin-ikuspegia, dela beste edozein helburu. Eduki digitalik gabe, Internet baliorik eta interesik gabeko alfer-lurra izango litzateke. Kalitatezko edukiak formatu desberdinetan sortzea funtsezkoa izango da edozein proiektu garatzeko, hartzailea erakartzeko, leialtzeko eta hari ekarpenik egiteko. Bitarteko digital batean sartzen den informazio-material oro eduki digitala da. Testua, irudia, bideoa, audioa, aplikazioa, softwarea edo imajina dezakegun beste edozein elkarrekintza-bitarteko izan daitezke. Ezagunen artean, honako hauek nabarmentzen dira:

- **Blog pertsonalak:** igorlearen mezua nortasunez adierazteko aukera ematen duen web-plataforma bat dira. Hartzailearekin zuzenean harremanetan jartzeko modu bat da, eta sormenez transmititu beharreko edukiei buruzko interesa sortzen du, xede den publikoarentzat garrantzitsua den hezkuntza-, informazio- edo informazio-materiala barne. Normalean, blogek iruzkin-atal bat izaten dute sarrera bakoitzerako, eta hartzaileak igorlearekin eta gainerako hartzaileekin elkarreragin dezake.
- Sare sozialak: interes edo balio komunetatik abiatuta konektatutako pertsonak eta erakundeek osatutako egiturak dira. Sare horien bidez, norbanakoen edo enpresen arteko harremanak modu horizontalean eraikitzen dira, muga fisikoak gaindituz. Erabilera pertsonala edo profesionala gorabehera, edukiak erraz, azkar eta doan partekatzeko aukera ematen dute. Garrantzitsua izango da profil bakoitzaren edukiak definitzea, hartzaileak argi izan dezan bilatzen duena aurkitzen duela. Gaur egun, sare sozial nagusiak hauek dira: Facebook, Youtube, Instagram, TikTok, WhatsApp, LinkedIn, Twitter, Pinterest, Telegram, Tumblr eta abar.
- **Bideo-plataformak:** bideoak kontzeptu bat ilustratzeko edo zalantzak argitzeko tresna erabilgarriak dira, eta proiektu baten alde sortzaileena edo gizatiarrena erakusten lagun dezakete. Mezu bat transmititzeko oso modu hurbila dira. Mezu elektronikoko batean sartzean, baliteke arazoa sortzea, hartzen duten tamaina dela eta. Gomendagarria izango da





streaming-plataformak erabiltzea eduki horiek igotzeko, eta mezu elektronikoa haietarako esteka bat sartzea. Youtube eta Vimeo erabil daitezke, material mota hori partekatzea ahalbidetzen duten doako plataformak ezagunenetako bi.

- Audio- edo podcast-plataformak: gai edo ekitaldi partikular baten aurkezpena eta eztabaida audio-fitxategi baten bidez garatzeko modu eroso eta interesgarria da. Fitxategi batean eskuragarri dago deskargatzeko edo streamingean. Eskaera txikiko formatua da, hartzaileak nahi duenean entzun baitezake edozein gailutan. Bideoarekin bezala, arazo bat izan liteke audio-fitxategi bat mezu elektronikoa baten eranskin gisa sartzea. Egoera hori konpontzeko, SoundCloud eta Spreaker plataformetara jo daiteke. Plataformak horiek audio-fitxategiak hainbat formatutan igotzeko eta beharren arabera partekatzeko aukera ematen dute. Podcasta kudeatzeko plataformak ezagunenetako batzuk iVox, SoundCloud, Anchor, Spotify for Podcasters, Google Podcasts, Spreaker, MixCloud eta Apple Podcasts dira, besteak beste.



Formatu horiei esker, distantziak murriztu, eduki berriak ezagutu eta hausnarketarako eta hazkunde pertsonalerako espazioen alde egin daitezke. Edukiak prestatzeko denbora eta ahalegina optimizatu egin behar dira, kalitatezko materiala sortzea lortzeko. Eduki digitalekin, lor daitezke xede-publiko gehiago erakartzea, proiektu baten posizionamendua eta ikusgarritasuna hobetzea eta humanizatzea, eta autoritatea ematea.



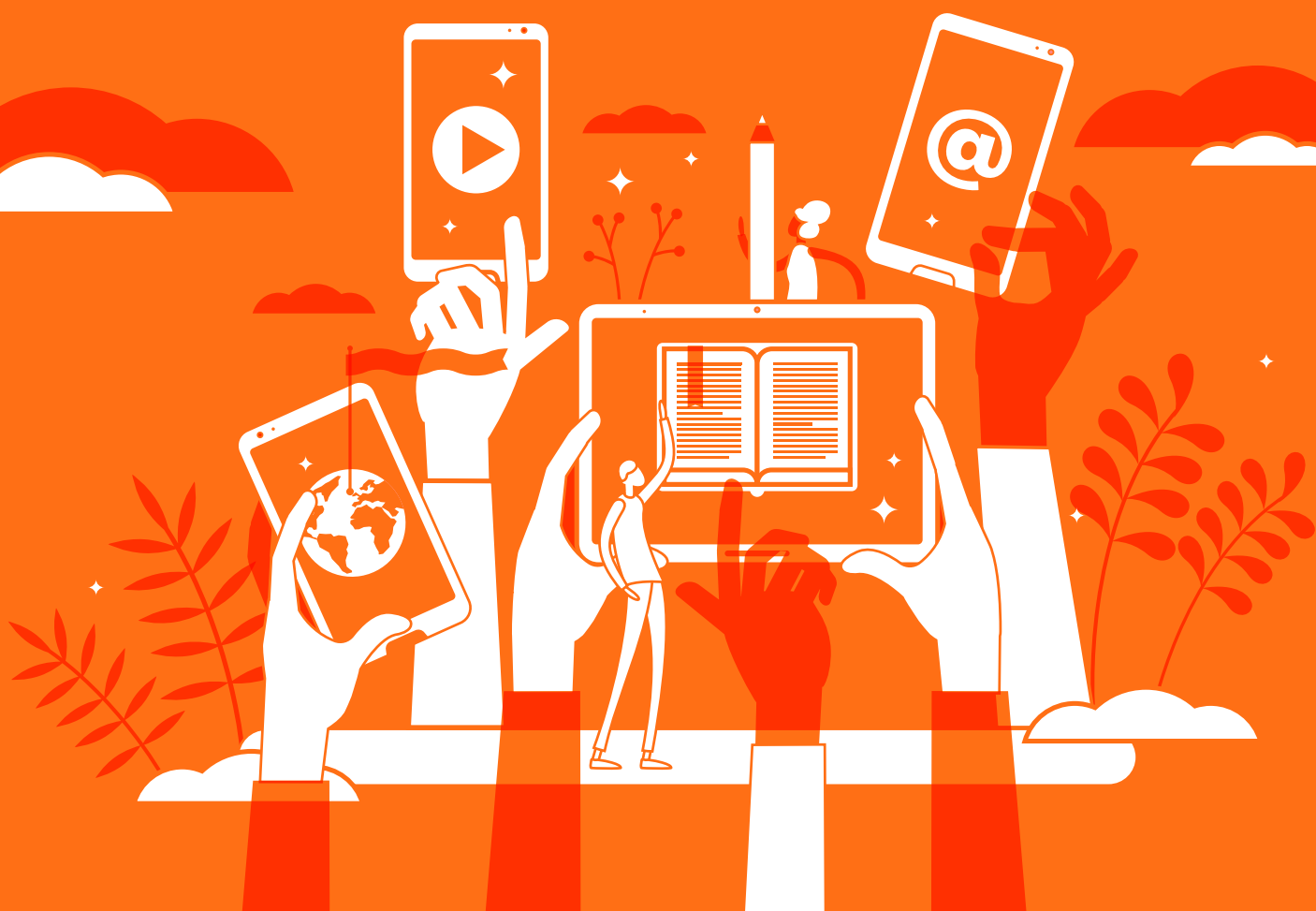


DigitAll

Eduki digitalak
sortzea

3.2

EDUKI DIGITALA INTEGRATZEA ETA BIRLANTZEA





Eduki digitalak
sortzea

A1 maila 3.2 Eduki digitala integratzea
eta birlantzea

Aurkezpenetan testua, irudiak, audioa eta bideoa integratzea





Aurkezpenetan testua, irudiak, audioa eta bideoa integratzea

Aurkezpenak sortzeko eta editatzeko tresnek elementu komun batzuk dituzte, eta erabiltzaileak erabiliko ditu diseinu-irizpide batzuen eta estilo-giden arabera, bere aurkezpenak sortzeko. Honako hauek dira garrantzitsuenak:

- **Diapositiba (slide):** informazioa aurkezteko oinarrizko unitatea. Aurkezpen batek diapositiba-multzo bat izango du, eta horietako bakoitzak informazio testuala (taulak, webguneetarako estekak), grafikoa (irudiak eta grafikoak) eta are multimedia (audioa, irudi animatuak, bideoak) izan ditzake.
- **Animazioa(k):** efektuak dira, eta diapositibaren elementuetan sar daitezke edo ez. Hala bada, diapositiba baten barruko elementu guztiek ez dute animazio bat izan behar; hau da, erabiltzaileak erabakiko du elementu bakoitzerako zer animazio iruditzen zaizkion egokien. Animazioei esker, diapositiba bateko elementuak erakutsi, nabarmendu edo ezkutatu ahal izango dira, aurkezleari mezua transmititzen laguntzeko.
- **Trantsizioa:** diapositiba baten eta hurrengoaren arteko tartea aurkezpenean. Diapositiba-aurkezpen bateko aukerako elementua da; izan ere, erabiltzaileak bi diapositibaren arteko trantsizio-efekturik ez egotea aukeratu dezake. Aurkezpen baterako trantsizio egokia aukeratzea ez da lan erraza, ezta hutsala ere, trantsizioa aukeratzeak aurkezpen-erritmoari eragingo baitio.

Hiru elementu nagusi horiek erabiliz, edozein erabiltzailek diapositiba-aurkezpen bat sortu ahal izango du helburu jakin baterako. Diapositibak sortzeko eta editatzeko erabiltzen den tresna edozein dela ere, tresna horrek aukera emango dio erabiltzaileari bere diapositibetan edozein eduki mota sartzeko eta audientziaren aurrean aurkezpen bat egiteko; izan ere, diapositibak aurkezteko edo pasatzeko modu bat dute guztiek horretarako.

Diapositibak aurkezteko edo pasatzeko modua diapositibak sortzeko eta editatzeko edozein tresnak eskaintzen duena da, aurkezpena egin ondoren emateko. Haren bidez, aurkezleak audientziarekin parteka ditzake erabilitako diapositibak





(diseinuaren alderdiak ezkatatuta) eta haiekin elkarreragin dezake definitutako animazioen eta trantsizioen bidez, aurkezpena ahalik eta eraginkorra izan dadin.

Hona hemen gaur egungo tresnarik ezagunenetako batzuk:

Tresna	Logotipoa	Deskribapena
PowerPoint (pribatiboa)		PowerPoint aurkezpenak sortu eta editatzeko Microsoften softwarea da (e.digitall.org.es/powerpoint), erabiltzaileei aukera ematen die aurkezpen erakargarriak sortzeko, banakako orrialdeez edo diapositibez osatutakoak. Oso tresna osatua da, eta asko erabiltzen da. Bateragarria da Windows eta macOS sistemekin. Hala ere, ordainpeko tresna bat da, eta lizentzia bat erosi edo hileko edo urteko harpidetza ordaindu behar da.
Keynote (pribatiboa)		Doako tresna da, Appleren bulegotikako suitean integratua (apple.com/keynote); aurkezpen eraginkorrak modu errazean sortzeko aukera ematen du. Software honetan oso erraza da animazioak sortzea eta, beharrezkoa izanez gero, sekuentziatzea, animazio-efektu asko eta asko dituelako, oso originalak. Keynotek macOS eta iOS gailuen artean arazorik gabe lan egiteko aukera ematen du, baita PowerPoint-ekin egindako aurkezpenak editatzeko ere.
Googleren aurkezpenak (pribatiboa)		Googleren doako tresna da. Googleren aurkezpenak tresna (e.digitall.org.es/slides) erabilgarri dago web-aplikazio gisa, aplikazio mugikor gisa Android, iOS, Windows eta BlackBerry-rako, eta mahaigaineko aplikazio gisa, Googleren ChromeOS-en. Erabiltzaileak Google Drive webgunearan bidez sar daitezke aurkezpenetara. Aurkezpenak erabiltzaile askok aldi berean editatu eta parteka ditzakete, eta erabiltzaileek aldaketak diapositibaz diapositiba ikusiko dituzte beste kolaboratzaile batzuek edizioak egin ahala. Era berean, bateragarria da PowerPoint-eko fitxategi-formatuekin.
LibreOffice Impress (librea)	 <small>Galdam Jitsu (Geeko), CC BY-SA 4.0 e.digitall.org.es/cc-licenses via Wikimedia Commons</small>	Doako tresna bat da, software libreko lizentzia duena; The Document Foundationek garatu du LibreOfficeren bulegotikako suiterako (e.digitall.org.es/impress). Batez ere sistema eragile libreetan erabiltzen da, hala nola GNU/Linuxen oinarritutakoetan. Bere interfazea PowerPointen antzekoa da, eta, gainera, bateragarria da formatu horrekin.
Beamer (librea)		Beamer LaTeX testu-editorearen erabiltzaileen komunitateak erabiltzen duen osagarria da. Horri esker, LaTeX (latex-project.org) komunitatean dokumentuak berrerabil daitezke kalitate tipografiko handiko aurkezpenak sortzeko. Berez, ez da aurkezpen-tresna bat, eta horrek aukerak mugatzen ditu. Gainera, ezin dute erabili LaTeX ezagutzen ez duten erabiltzaileak. Sistema eragile hauetarako dago eskuragarri: Windows, macOS, Unix, GNU/Linux, etab.
Prezi (pribatiboa)		Prezi ren (prezi.com) tresna bat da. Aurkezpen horiek aurkezpen osoa erakusten duen irudi edo grafismo batetik abiatzen dira, eta, lupa baten metafora erabiliz, zooma egiten da eremu desberdinetan edukia erakusteko. Preziren aurkezpenak oso efektistak eta dinamikoak dira, eta animazioak hartzen dituzte barne ikusleei ideiak ikusi, ulertu eta gogorarazteko. Tresna horien helburua testuinguru informala edo familiarragoan aurkezpenak egitea da.
Canva (pribatiboa)		Canvak (canva.com) zerbitzuak ematen ditu aurkezpenak sortzeko ez ezik, baita infografiak eta bestelako publizitate-materialak sortzeko ere. Drag & drop (arrastatu eta askatu) diseinua duen interfazea dela eta, erabiltzailearentzat oso erraza da aurkezpen bat edo beste edozein material diseinatzea eta sortzea. Gainera, Canvak txantiloil-liburutegi handia eskaintzen du; milaka elementu ditu txertatuak, eta erabiltzaileak bere irizpidearen arabera erabil ditzake. Tresna hori eskuragarri dago web-aplikazio gisa, Android eta iOS-erako aplikazio mugikor gisa eta Windows eta macOS-erako mahaigaineko aplikazio gisa.



Eduki digitalak
sortzea

A1 maila 3.2 Eduki digitala integratzea
eta birlantzea

Kalkulu-orriak: datuak adieraztea eta kalkulatzeko





Kalkulu-orriak: adierazpidea eta kalkulua datuekin

Kalkulu-orri bat zenbakizko datuekin eta datu alfanumerikoekin lan egiteko aukera ematen duen tresna informatiko bat da. Tresna horiek aukera ematen dute datuekin eragiketa matematikoak egiteko eta horien irudikapen grafikoak lortzeko. Irudi horiek aurkezpenetan edo txostenetan berrerabil daitezke. Adibidez, irakasle batek kalkulu-orri bat erabil dezake ikasleen kalifikazioak kudeatzeko. Tresna horrek batez besteko kalifikazioak kalkulatzeko, estatistikak egiteko edo grafiko estatistikoak dituzten ikasleen zerrendak sortzeko ahalbidetuko lioke.

Kalkulu-orrien tresnek elementu komun batzuk dituzte, eta honako hauek nabarmentzen dira:

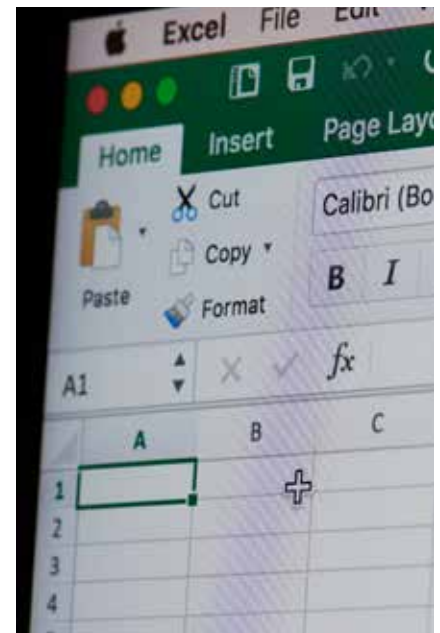
- **Gelaxka:** informazioa aurkezteko oinarriko unitatea gelaxka da. Bertan datu bat gordetzen da (zenbakizkoa edo alfanumerikoa).
- **Orria:** taula bateko gelaxken multzoari orri deitzen zaio. Taula bateko gelaxka horizontalen multzoak errenkada bat osatzen du, eta gelaxka bertikalen multzoari zutabe deitzen zaio. Errenkadak zenbakiekin identifikatzen dira, eta zutabeak, berriz, letrekin. Beraz, gelaxka bakoitza taularen barruko errenkadak eta zutabeak definitzen dute. Adibidez, A3 lehen errenkadako eta hirugarren zutabeko gelaxkari dagokio.
- **Barrutia:** goiko ezkerreko gelaxkak eta beheko eskuineko gelaxkak definitutako laukizuzen batean dauden gelaxka guztiak dira. Adibidez, A1:B2 hauek dira: A1, A2, B1 eta B2 gelaxkak.
- **Liburua:** kalkulu-orri bateko fitxategi bat da, eta zenbait orriz osatuta dago. Orri bakoitzak izen bat du bertan jasotako datuak erreferentziatzeko.
- **Txantiloia:** antzeko beste liburu batzuk sortzeko oinarri gisa erabiltzen den liburu. Liburu eta orrietarako txantiloiak sor daitezke.
- **Eragiketak eta funtzioak:** Kalkulu-orriek aukera ematen dute eragiketa aritmetikoak egiteko (batuketak, kenketak, biderketak eta zatiketak) errenkadak edo zutabeak erabiliz. Kalkulu konplexuagoak ere badaude funtzioak deiturikoetan jasota. Funtzioei esker, kalkuluak egin daitezke





orri batean edo gehiagotan dauden datu-multzo baten gainean. Adibidez, datu-zutabe baten batez bestekoa kalkula daiteke, errenkada baten balio guztiak batu edo hautatutako eskualde bateko zenbaki edo testu baten agerpen kopurua zenbatu. Gelaxka baten edukia funtzio jakin bat aplikatzearen emaitza izan daiteke.

- **Grafikoak:** erabiltzaileak orriaren eskualde bat hauta dezake eta hautatutako datu-multzo batetik abiatuta egiten den grafiko mota bat txertatu. Tresna horiek aurrez definitutako grafiko mota ugari dituzte, hala nola lerroak, eremuak, zutabeak, barrak, sakabanatzea, zirkularrak, histogramak eta organigramak, besteak beste.



⚠ ADI

Hauek dira kalkulu-orrien erabilera ohikoenak:

- **Datuak antolatzea.** Orri bat baino gehiago edukitzeak aukera ematen du datuak sailkatzeko eta antolatzeko, bai eta bilaketa mota desberdinak egiteko ere.
- **Datuak partekatzea.** Erabiltzaileek aukera dute liburuak elkarri partekatzeko, erabiltzaileek datuak erraz eta arin eskuratzeko aukera izan dezaten.
- **Datuak araztea eta iragaztea.** Orriek datu guztiak dituzte, baina haien bertsio sinplifikatuak ikustea ahalbidetzen duten iragazkiak dituzte, datu horiek aztertu eta arazteko aukera ematen dutenak.
- **Datuak grafikatzeko.** Askotariko grafiko eta diagramak egiteko aukera ematen du, informazioa errazago ulertzeko, ereduak aztertze eta abar. Grafiko horiek, ondoren, aurkezpenetan edo txostenetan sar daitezke.
- **Datuen manipulazio matematikoa.** Liburu bateko datuekin kalkulu matematikoa erraz egin daitezke. Kalkulu horiek eragiketa sinpleak eta problema matematiko konplexuak izango dira.



Hona hemen gaur egungo tresnarik ezagunenetako batzuk:

Tresna	Logotipoa	Deskribapena
Excel (privatiboa)		Excel <i>Microsoft</i> en kalkulu-orrien softwarea da (e.digitall.org.es/excel), eta erabiltzaileei aukera ematen die funtzio askorekin lan egiteko. Oso tresna osatua da, eta asko erabiltzen da. Bateragarria da Windows eta macOS sistemekin. Hala ere, ordainpeko tresna bat da, eta lizentzia bat erosi edo hileko edo urteko harpidetza ordaindu behar da.
Numbers (privatiboa)		Doako tresna da, <i>Appleren</i> bulegotikako suitean integratua (apple.com/numbers); kalkulu-orriak erraz sortzeko aukera ematen du. Sinpletasuna eta dotorezia dituelako nabarmentzen da. Numbersek macOS eta iOS gailuen artean arazorik gabe lan egiteko aukera ematen du, baita diagramak gehitzeko aukera ere iPaden Apple Pencil bat erabiliz. Gainera, aurrez definitutako txantilo ugaritu erabilera-egoera askotarako.
Googleko kalkulu-orriak (privatiboa)		Googleren doako tresna da. <i>Googleren kalkulu-orriak</i> (e.digitall.org.es/sheets) eskuragarri dago web-aplikazio gisa, eta horri esker, erabiltzaileek askotariko sistema eragileekin lan egin dezakete. Erabiltzaileak Google Drive webgunearen bidez sar daitezke kalkulu-orrietara. Kalkulu-orriak hodeian gordetzen dira, eta erabiltzaile askok erabil ditzakete aldi berean. Erabiltzaileek aldaketak ikus ditzakete beste kolaboratzaile batzuekin edizioak egin ahala. Halaber, bateragarria da kanpoko beste formatu batzuekin, Excel fitxategia barne.
LibreOffice Calc (librea)	 <small>Galdam Jitsu (Geeko), CC BY-SA 4.0 e.digitall.org.es/cc-licenses Wikimedia Commons bidez</small>	Doako tresna bat da, software libreko lizentzia duena; The Document Foundationek garatu du <i>LibreOfficeren</i> bulegotikako suiterako (e.digitall.org.es/calc). Batez ere sistema eragile libreetan erabiltzen da, hala nola GNU/Linuxen oinarritutakoetan. Bere interfazea Excelen antzekoa da eta, gainera, formatu horrekin bateragarria da.
Zoho Sheet (privatiboa)		<i>Zoho Sheet</i> (zoho.com/sheet) aplikazio bat da, garatzaile honen bulegotikako suitearen barrukoa. Ohiko funtzioak, taula dinamikoak eta grafikoak ditu. Lana hodeian egiten du, eta, beraz, elkarlana ahalbidetzen du, bai eta proiektuetara Interneterako konexioa duen edozein ordenagailutatik sartzeko ere. Doakoa da 25 erabiltzaileraren arte, eta, beraz, enpresa txikientzako eta azken erabiltzaileentzako alternatiba bat da.





Eduki digitalak
sortzea

A1 maila 3.2 Eduki digitala integratzea
eta birlantzea

Dauden eduki digitalen konposizioa





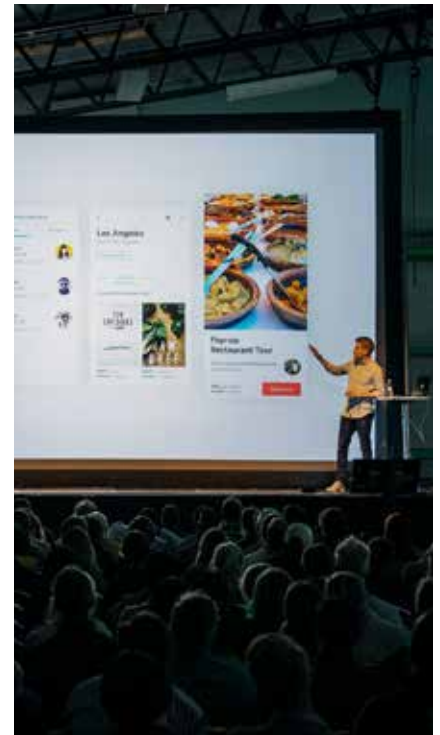
Dauden eduki digitalen konposizioa

Helburua edozein dela (produktu bat aurkeztea, zerbitzu bat iragartzea, gertaera edo bidaiari pertsonal bati buruzko edukia lantzea) eduki digitala prestatzen edo lantzen dugunean, askotan eduki berria sortzeko konbinatu nahi ditugun irudiak, audioak eta bideoak ditugu, eta haiek **konbinatu** nahi ditugu eduki berria sortzeko. Kasu horietan, gure helburua **slideshow** motako eduki digital bat sortzea da, hau da, helburu zehatz bat duen irudi, bideo eta soinuen **konposizio** bat sortzea.

Gaur egun, **software tresna** ugari daude, **doakoak** eta intuitiboak, ezagutza aditua ez duen edozein erabiltzailek eduki digitala sor dezatan. Tresna horietako batzuk **webetik** zuzenean eskuratzen dira, software espezifikorik instalatu beharrik gabe. Guztiek dute edizio-sistema komun bat, nahiz eta desberdintasun eta berezitasun batzuk dauden haietako bakoitzaren menu eta botoietan. Horretarako, **drag and drop** motako sistema bat erabiltzen dute, eduki berriaren parte izatea nahi ditugun elementuak denbora-lerro batean arrastatzeko eta askatzeko aukera ematen duena. Elementu horiek denbora-lerroan ordenatu ditugun bezala erakutsiko dira.

Irudiak erabiltzean, jakin behar dugu eraldaketak eta prozesamenduak egin daitezkeela haien gainean, lortu nahi den efektua lortzeko. Hala, web-programa eta -tresna batzuk erabil daitezke, aukera ematen dutenak argazki batean **testua txertatzeko** (meme bat sortzeko, adibidez), lehendik dauden **zenbait argazki konbinatzeko** (muntaiak sortzeko; adibidez, pertsona baten burua beste baten gorputzean jartzea, bi eszena edo bi paisaia batzea, etab.), zenbait iruditatik abiatuta animazioak sortzeko, irudi horien gainean **efektuak sortzeko**, emaitza original bat lortzeko eta irudi-multzo batetik abiatuta **bideoak sortzeko**.

Bestalde, **audioa** oso elementu erabilgarria da aurkezpenari **erritmoa** emateko eta **giroa** sortzeko. Horretarako, posible da irudi batekin batera doazen audio-piezak sartzea, **akzio** gehiago emateko (adibidez, auto-ilara baten irudiaren gaineko zalaparta-audioa) edo hondoko audioa txertatzea zenbait iruditan, giroa sortzeko eta **emozioak** eragiteko (adibidez, haur baten argazkien gainean sehaska-kanta baten melodia sartzea). Bideo batean audio osoa edo zati bat **ordeztu** ere egin daiteke guk aukeratzen dugun beste batez.





Bideoari dagokionez, edukiak transmititzeko gaitasun handiena duen multimedia-elementua da. Bereziki garrantzitsua da eduki berriak sartzen diren irudiak, audioak eta bideoak kalitate onekoak izatea. Gainera, audioaren eta bideoaren kasuan, bereziki garrantzitsua da ziurtatzea erabili nahi diren mota horretako fitxategiak **bateragarriak** direla eduki digitala sortzeko erabiltzen ari garen softwarearekin. Hala ez bada, **fitxategiak bihurtzeko doako tresnak** daude, bai mahaigaineko aplikazio gisa, bai web-bertsio gisa. Behin edukia eginda, ohikoena da bideo-formatuan esportatzea. Fenomeno hori **renderizatua** izenez da ezaguna. Renderizatuak denbora gehiago edo gutxiago beharko du, eta garrantzi txikiagoko edo handiagoko fitxategia sortuko du, sartutako elementu kopuruaren, bideoaren iraupenaren eta nahi den bereizmenaren arabera.

Aurreko elementu guztiak **konbinatzeko**, trantsizioak eta animazioak edo efektu bereziak ere erabili daitezke. Trantsizioek zenbait elementu konbinatzeko aukera ematen dute, edukiari jarraitutasuna emateko aukera ematen duten efektuen bidez. Animazio edo efektu bereziek, aldi berean, dinamismoa ematen diote sortutako edukiari, haien eragina hobetzeko.

Modu horietan guztietan sor daitezke eduki digitalak lehendik dauden beste batzuetatik abiatuta. Eduki berri horiek izango dira **meme** bat, **infografia** bat eta **aurkezpen** bat, baita **slideshow** motako eduki konplexuagoak ere, eskuragarri dauden multimedia-elementu guztiak konbinatuta.

Informazio gehiago

Fitxategiak bihurtzeko tresnak:

- 300 formatu arteko doako online audio-bihurgailua. online-audio-converter.com/es
- Edozein testu-, irudi-, bideo- edo audio-formatu mota bereko beste bat bihurtzeko aukera ematen duen tresna. online-convert.com





DigitAll

Eduki digitalak
sortzea

3.3

**EGILE-ESKUBIDEAK
ETA JABETZA
INTELEKTUALEKO
LIZENTZIAK**





Eduki digitalak
sortzea

A1 maila 3.3 Egile-eskubideak eta jabetza
intelektualeko lizentziak

Egile-eskubideak eta jabetza intelektualeko lizentziak





Egile-eskubideak eta jabetza intelektualeko lizentziak

Jabetza intelektualak: oinarriko kontzeptuak

Jabetza intelektualaren objektu dira gaur egun ezaguna den edo etorkizunean asmatuko den edozein bitarteko edo euskarri ukigarri edo ukiezin erabiliz adierazitako jatorrizko sorkuntza literario, artistiko edo zientifiko guztiak. Horien artean sartzen dira:

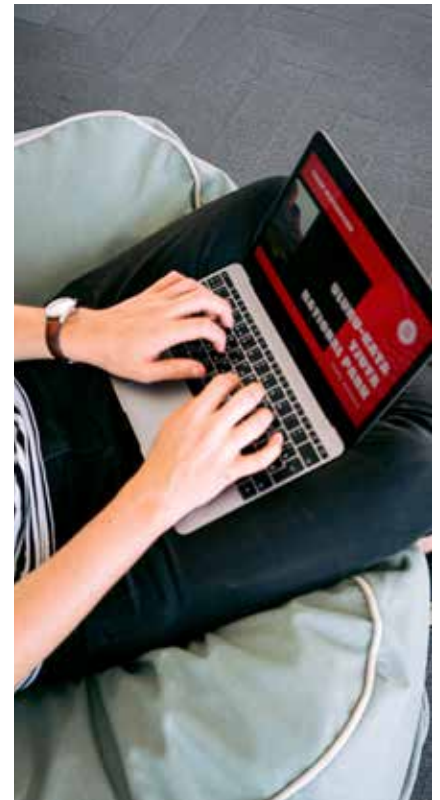
- Ordenatu-programak.
- Besteren lanen bildumak, hala nola edukien aukeraketagatik edo antolamenduagatik sorkuntza intelektualak osatzen dituzten datu-baseak. Bilduma horiei aitortutako babesa haien egiturari dagokio soilik, edukien hautaketa edo antolamendua adierazteko moduari dagokionez, eta ez da edukietara hedatzen.

Egile eskubideak sorkuntza intelektualetan

Jabetza intelektualak eskubide pertsonalek eta ondare-eskubideek osatzen dute. Eskubide horiek egileari esleitzen diote obra erabat xedatzeko eta ustiatzeko eskubide eskusiboa, eta muga bakarrak legean ezarritakoak dira.

Egiletzat hartzen da obra literario, artistiko edo zientifikoren bat sortzen duen pertsona naturala. Kontrakorik frogatu ezean, egiletzat joko da obran egile gisa agertzen dena, haren izenarekin, sinadurarekin edo identifikatzen duen zeinuarekin. Egile-izaerari ezin zaio uko egin; ezin da *inter vivos* edo *mortis causa* transmititu, ez da iraungitzen denborak aurrera egin ahala, eta ez da jabari publikoan sartzen eta ezin da preskribatu.

Obra literario, artistiko edo zientifiko baten jabetza intelektualak egileari dagokio obra hori sortze hutsagatik. Barne hartzen ditu egileei eta beste titular batzuei (artistak, ekoizleak, irati-difusioko erakundeak, etab.) dagozkien eskubideak, hau da, haien sortutako obren eta prestazioen gainekoak. Sistema anglosaxoian, kontinentalean ez bezala, egile-eskubideari copyright esaten zaio.





- Jabetza intelektuala = Egile-eskubideak ➔ Sistema kontinentala eta gaztelaniaduna
- Copyright = Ustiatzeko eskubideak ➔ Sistema anglosaxoia

Copyrightaren definizioa

Copyright© ikurrak publikoari jakinarazten dio obra bat originala dela eta haren erabilera, erreprodukzioa, eraldaketa, argitalpena eta abar egile-eskubideen mende daudela. Legeak babestutako obra edo ekoizpen baten gaineko ustiapen-eskubide baten titularrak edo lagapen-hartzaileak © ikurra jarri ahal izango du bere izenaren aurretik, lan edo ekoizpen horiek jendarteratu diren tokia eta urtea zehaztuz. Ez da beharrezkoa obraren jabetza intelektualeko eskubideak inskribatzeko eskatzea © ikurra gehitzeko.

Aipatutako sinboloak eta erreferentziak jaso eta jarri beharko dira ustiapen-eskubideak erreserbatuta daudela argi eta garbi erakusteko moduan.



Public Domain (jabari publikoa)

Obra bat edozein modutan ustiatzeko eskubideek eta, bereziki, erreproduzitzeko, banatzeko, jendaurrean komunikatzeko eta eraldatzeko eskubideek egilearen bizitza osoan iraungo dute, bai eta hil edo heriotza-deklarazioa egin eta 70 urte bete arte ere. Obrak ustiatzeko eskubideak azkentzeak obrak jabari publikora pasatzea ekarriko du. Edonork erabili ahal izango ditu jabari publikoko obrak, baldin eta obraren egiletza eta osotasuna errespetatzen badira, eta obra desitxuratzea, aldatzea edo haren aurka atentatu egitea eragozten bada, betiere haren interes legitimoren eta ospearen kontrakoa bada.





Jabari publikoko obrak Creative Commons bidez arautu daitezke, bi marka hauekin:

- **CC Public Domain:** egile-eskubideak (ustiapen-eskubideak) babesten ez dituzten obrak adierazten ditu, babes-epa amaitu delako, eta, ondorioz, jabari publikokoak direnak.
- **0 Public Domain:** sorkuntza berrien egileek beren obrei eman diezaieketen marka bat da, haren gaineko edozein eskubideri uko egiten diotela adierazten duena, eta jabari publikoko obra bat balitz bezala tratatuko litzateke.



Egile eskubideak sorkuntza intelektualetan

CC lizentziek egileei edo eskubideen beste titular batzuei baimena eta/edo lagapena ematen diete, baldintza jakin batzuetan, beren obren gaineko eskubideetako batzuk baimendu eta/edo lagatzeko, eta horien beste zati bat mantentzeko. Sormen-edukiak edozein motatako eduki sortzaileei aplika dakizkieke, datu-baseak barne.

Creative Commons bitartez (creativecommons.org) kudeatzen dira; gobernu kanpoko erakunde bat da, eta egileei nahiz sortzaileei lizentzia eta tresna libreak eskaintzen dizkie.

CC lizentzia bakoitzak zehazten du zer eskubide lagatzen dituen egileak, zer baldintzatan eta erabiltzaileek obrarekin zer egin dezaketen, batez ere berriro argitaratu edo aldatu nahi badute. CC lizentzietan oinarritzko 4 baldintza konbinatzen dira: Aitortzea (nahitaezkoa eta ezinbestekoa), Merkataritzakoa ez den erabilera, Lan eratorririk gabea, eta Berdin partekatzea. Konbinatuz gero, 6 CC lizentzia sortzen dira, eskema honetan azaltzen den bezala:





Tipos de licencias

¿Como funcionan?

Citar siempre!



Reconocimiento al autor



No comercial



Sin obras derivadas



Compartir igual

6

combinaciones



CC BY	
CC BY-NC	
CC BY-ND	

CC BY-SA	
CC BY-NC-SA	
CC BY-NC-ND	

Buscar obras con CC



<http://creativecommons.org>

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

crue Universidades Españolas Red de Bibliotecas REBIUN



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



Eduki digitalak
sortzea

A1 maila 3.3 Egile-eskubideak eta jabetza
intelektualeko lizentziak

Plagioa





Plagioa

Aurreko unitateetan azaldu dugun bezala, obra literario, artistiko eta zientifikoekiko eskubide guztiak egileei dagozkie, izan artista, idazle, ekoizle edo programatzaile informatikoak. Eta haien baimendu behar dute beren sorkuntzen erabilera eta ustiapena, Jabetza Intelektualak eskubide guztiak babesten baititu.

Azken 20 urteotan plagioen hazkunde esponentziala gertatzen ari da, neurri handi batean Interneten agerpenagatik eta haren erabilera masifikatuaren ondorioz.

Plagiatzea, funtsean, da besteen lanak kopiatzea eta norberarenak balira bezala aurkeztea. Hala ere, garrantzitsua da adieraztea bi sorkuntzaren edo gehiagoren arteko kointzidentzia oro ez dela plagioa. Soilik egiturazko oinarriko eta funtsezko kointzidentziak daudenean, eta ez, ordea, kointzidentzia osagarriak edo erantsiak direnean, hau da, ez-transzendentalak.

Pertsona askok, borondatez edo ezjakintasunagatik, saretik deskargatzen dituzte testuak, argazkiak, programa informatikoak, artikulak akademikoak eta zientifikoak eta abar.

Informazio hori EZIN da libreki eta norberarena balitz bezala erabili, Jabetza Intelektualaren bidez babestuta baitaude, egileak bereniaz baimentzen duen kasuetan izan ezik. Gainerako informazioa bezala, blog bat, web-orri bat edo dokumentu elektronikoa bat egileren baten lanak dira, eta erabilera- eta aipu-baldintza berak bete behar dituzte, egile-eskubideek babesten baitituzte.

Ez kopiatu eta itsatsi besterik gabe. Jatorrizko iturria aipatu behar dugu beti. Kopiatutakoaren larritasuna nolakoa den, baliteke Zigor Kodean araututako arau-hauste astuna egitea [270. artikulutik 272.era artekoak (e.digital.org.es/legislacion)], eta horrek ondorio larriak dakartzio egileari, hala nola 6 hilabetetik 4 urtera arteko espetxe-zigorak. 270.2 artikulua dela kausa internauten artean ospe handia zuten zenbait webgunek itxi behar izan dute azken urteotan (besteak beste, RojaDirecta, Series.ly eta EliteTorrent). Erabiltzaileek webgunean kargatutako edukietarako estekak soilik erakusten zituzten arren, auzitegiek jabetza intelektualaren aurkako delituen erantzule egin zituzten.

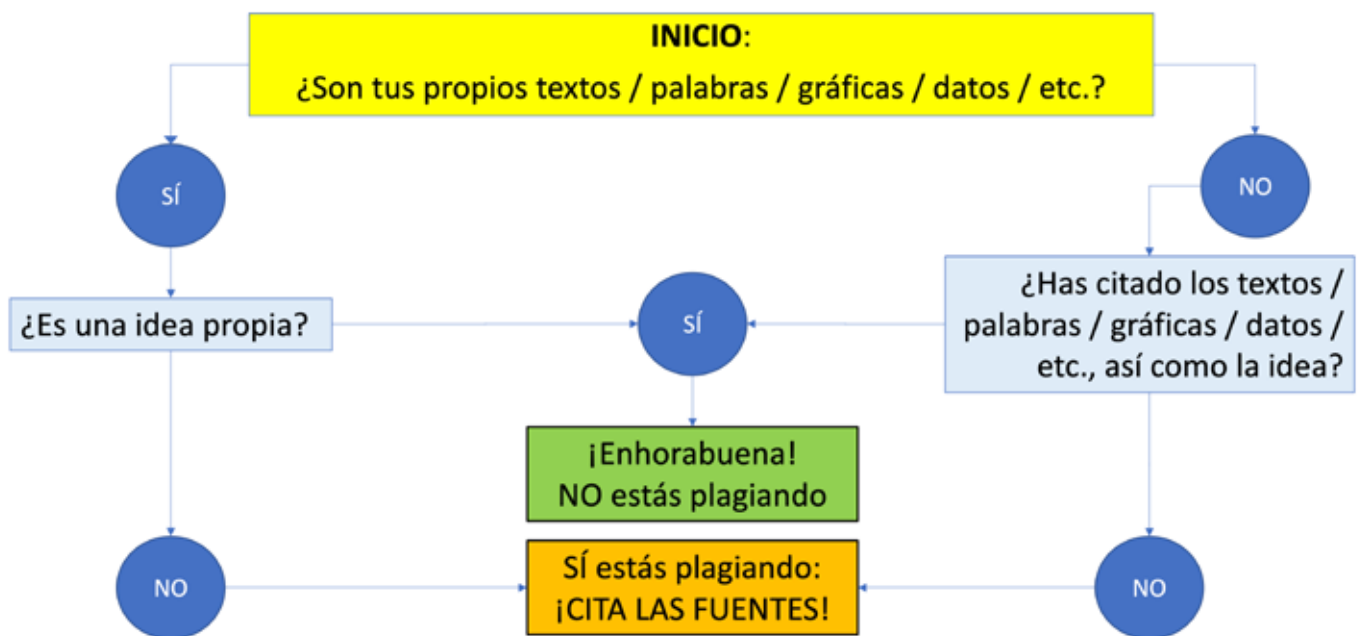




Unibertsitatean, ikasleen % 60k baino gehiagok **plagio akademiko**ren bat egin du lanak egiteko. Hori gerta ez dadin, ikasleak ez ditu testua, grafikoak, datuak eta abar kopiatu behar, jatorrizko artikuluetan agertzen diren bezala, non eta egilea eta jatorrizko obra aipatzen ez diren eta komatxoaren artean transkribatzen ez diren.

Diagrama honek plagiatzen ari zaren edo ez zehazteko aukera emango dizu:

¿Estoy cometiendo PLAGIO?



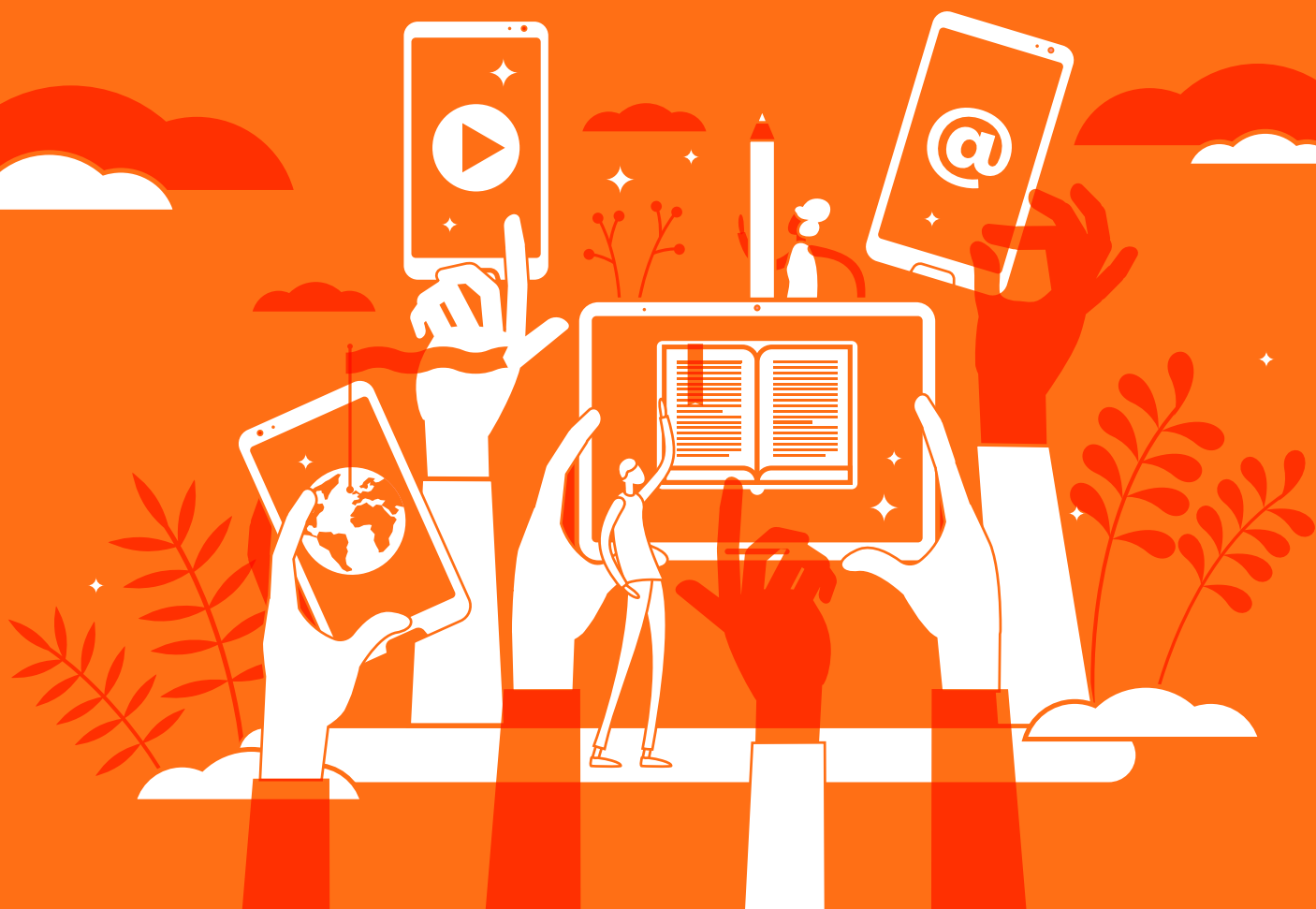


DigitAll

Eduki digitalak
sortzea

3.4

PROGRAMAZIOA





Eduki digitalak
sortzea

A1 maila 3.4 Programazioa

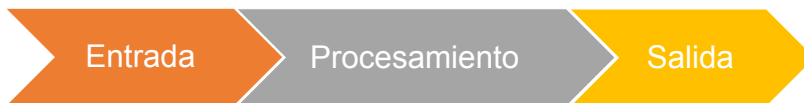
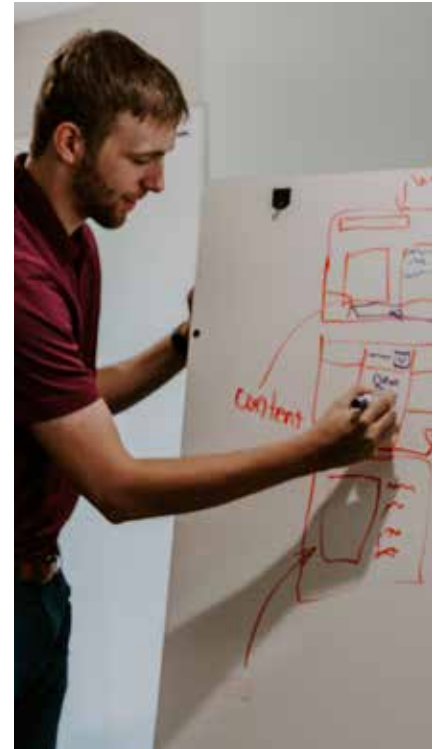
Algoritmoen ezaugarriak eta problemak ebaztea





Algoritmoen ezaugarriak eta problemak ebaztea

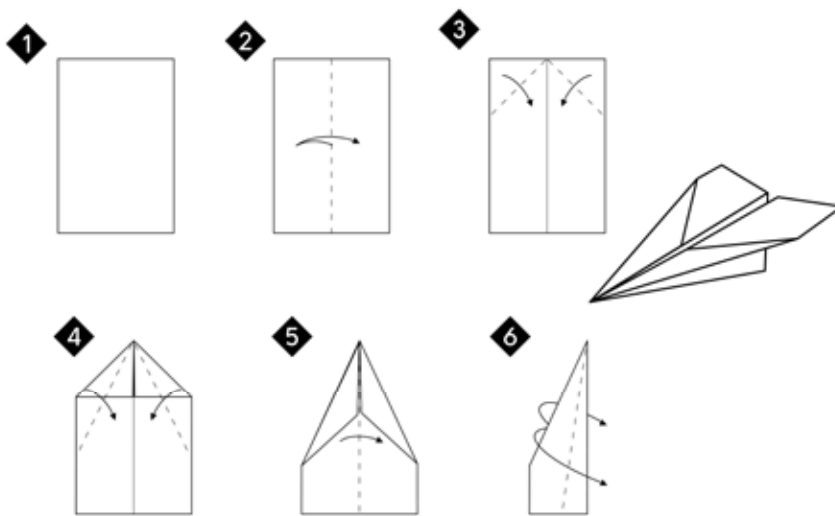
Ordenagailu baten bidez konpon daitezkeen arazo asko daude, baina guztien ezaugarri komuna da sarrera-datu batzuetatik abiatzen direla eta zenbait instrukzioen bidez prozesatzen direla emaitza batzuk emateko; hau da, 1. irudian ageri den eskemari jarraitzen diote. Programak *programatzailer* izeneko pertsonak idatzitako kode-lerroak dira; ordenagailuak ulertzen duen programazio-lengoaia batean idatzita daude, eta problema bat konpontzen dute. Pentsa liteke arazo bat konpontzeko nahikoa dela programazio-lengoaia bat ikastea eta zuzenean esertzea kodea idazten. Hala ere, horren baliokidea izango litzateke auto bat eraikitzea aurretik diseinatu gabe, edo film bat aldeztu aurreko gidoirik gabe filmatzea. Programazioaren testuinguruan, *algoritmoak* programak diseinatzeko oinarritzko tresnetako bat dira.



1. irudia. Ordenagailu batekin soluziona daitezkeen problema baten zatiak.

Algoritmo bat zera da: problema baten soluzioa deskribatzea, hura ebazten duten **urratsen sekuentzia ordenatuaren** bidez. Hauek dira haren ezaugarriak:

- **Zehatza**, hau da, ez du anbiguitasunik izan behar beti modu berean interpreta dadin.
- **Sendoa**, hau da, ez du akatsik izan behar.
- **Definitua**, beti modu berean erantzuten du inguruabar berberen aurrean.
- **Ordenatua**, argi utzita zein den egin beharreko instrukzio-sekuentzia.
- **Finitua**, hau da, uneren batean amaituko dena.
- **Irakurgarria**, hau da, irakurtzen duen edonork ulertzeko modukoa.



2. irudia. Paper-hegazkin bat egiteko instrukzioak. (e.digitall.org.es/aviones-papel)

2. irudiak problema bat soluzionatzeko algoritmo bat erakusten du; hain zuzen, paperezko hegazkin bat egitea. Adibide honetan, sarrera paper-orria da; irteera, eraikitako hegazkina; eta instrukzioak 1etik 6ra bitarteko irudiek irudikatzen dituzte. Ikusiko duzun bezala, algoritmoa grafikoki deskribatuta dago eta propietate guztiak betetzen ditu: zehatza, sendoa, definitua, ordenatua, finitua eta irakurgarria.

Programazioaren testuinguruan, algoritmoak modu askotan idatz daitezke, baina *sasikodea* erabili ohi da, komunikatzeko erabiltzen dugunaren antzeko hizkuntza delako, baina hizkuntza naturalaren berezko anbiguotasunik eta zehaztugabetasunik gabe. Sasikodeari buruzko guztia ikasiko duzu ondorengo beste gai batean. Orain, garrantzitsuena da problemak soluzionatzearekin lotutako aurretiazko prozesua zeureganatzea. Izan ere, arazo baten soluzioa deskribatzen duen algoritmoa diseinatzeko, lehenik eta behin ezinbestekoa da **problema** xehetasunez **aztertzea**, haren hiru zatiak identifikatzeko (1. irudia): sarrera-datuak, horiek prozesatzeko instrukzioak eta irteera-datuak. Horretarako, galdera hauei erantzuten saia zaitezke:

- Zer lortu nahi dut? (*Irteera*)
- Zer datu ditut? (*Sarrera*)
- Zer tresna ditut? eta,
- Zer instrukziori jarraitu behar zaie eta zer (*prozesatzea*)

Berehala ulertuko duzu adibide batekin: }





Imajina ezazu zirkulu baten, irrati-zirkulu baten, benetako zenbaki positibo baten azalera kalkulatu duen programa bat idazteko eskatzen dizutela. Horri heltzeko, aurreko galderari erantzun behar diezu:

- **Zer lortu nahi dut?** Zirkuluaren azalera irudikatzen duen zenbaki bat.
- **Zer datu ditut?** Erradioa adierazten duen zenbakia.
- **Zer tresna ditut?** Emaitzazko eragiketa programazio-lengoaia guztiek ematen dute.
- **Zer instrukziori jarraitu behar zaie eta zer ordenatan?** Ikus dezagun:
 - P1. *Erradioa* esaten diogu, adibidez, zirkuluaren erradioak adierazten duen zenbakiari
 - P2. *Erradioaren balioa* lortu behar da (demagun 2 balioa ematen zaiola)
 - P3. *P1* esaten diogu, adibidez, 3,14 balioari
 - P4. *Azalera* esaten diogu, adibidez, $P1 * erradio * erradio$ (hau da, $3,14 * 2 * 2$) adierazpenaren emaitza adierazten duen zenbakiari
 - P5. Emaitza gisa, *azaleraren balioa* ematen da (12,56 zenbakia)

Ikusiko duzu ezin dela azaleraren balioa erakutsi oraindik haren balioa kalkulatu ez bada; hau da, P5 urratsa beti joan behar da P4 urratsaren ondotik. Edo ezin da bi zenbakiren biderkadura egin horietako baten balioa ezagutzen ez bada; hau da, P4 urratsa beti joan behar da P2 urratsaren ondotik.

Hala ere, P3 urratsa P1 urratsaren aurretik edo ondoren joan daiteke, eta P2 urratsaren aurretik edo ondoren. Gainera, zenbakien eta adierazpenen izena beste bat izan zitekeen. Adibidez, erradioa adierazten duen zenbakiari r , R , *radius*, X edo gustatzen zaigun beste edozein jarri ahal geniokeen. Gauza bera 3,14159 balioaren izenarentzat eta azalera kalkulatu duen adierazpenaren balioarentzat. Horrenbestez, beste algoritmo batzuk sor daitezke, idatzi dugun algoritmoa bezain baliozkoak.

Sasikodearekin ohitzeko, adibideko algoritmoaren deskribapen posible bat hau litzateke:



⚠ ADI

Instrukzioen ordena instrukzioak eurak bezain garrantzitsua da.

⚠ ADI

Problema baten soluzioa deskribatzen duen algoritmoak ez du zertan bakarria izan.

ZirkuluAzalera **Algoritmoa****Datuak**

erradioa, azalera, PI: Zenbakiak

Instrukzioak

Erradioaren balioa lortzea

PI \leftarrow 3,14Azalera \leftarrow PI * erradioa * erradioa

Azalaren balioa ematea

ZirkuluAzalera **amaiera**

Azkenik, adibideak algoritmoen propietate guztiak betetzen dituela egiazta dezakezu: zehatza, sendoa, definitua, ordenatua, finitua eta irakurgarria. Gainera, algoritmoa zuzena da: egin beharrekoa egiten du. Beraz, programazio-lengoaia batera itzultzeko unea da; horri *ezarpena* esaten zaio. 3. irudian adibideko algoritmoaren zenbait ezarpen daude Pascal, Python eta Javan. Ikusiko duzu guztiak sasikodeko algoritmoaren oso antzekoak direla, bai eta, gainera, elkarren artean, eta sintaxiaren xehetasun espezifikoetan soilik bereizten dira.

<pre>program AreaCirculo; var radio, PI, area:Real; begin radio:=2; PI:=3.14; area:=PI*radio*radio; writeln (area); end.</pre>	<pre>radio=2 PI=3.14 area=PI*radio*radio print(area)</pre>	<pre>public class AreaCirculo { public static void main(String[] args) { double radio=2; double PI=3.14; double area=PI*radio*radio; System.out.println(area); } }</pre>
--	--	--

3. irudia. Adibideko algoritmoa Pascal, Python eta Javan ezartzeko aukerak.



Azken batean, algoritmoa definitu ondoren, ia berehala ezartzen da. Beraz, problema baten soluzioa programatzeko, honako prozesu honi jarraitu behar zaio, laburbilduta:

1 | **Problemaren analisia**

Sarrera-datuak, horiek prozesatzeko jarraitu beharreko urratsak eta eman beharreko irteera identifikatzeko.

2 | **Algoritmoaren diseinua**

Propietate guztiak betetzen dituela eta zuzena dela egiaztatu behar da
Sasikodean idatziko da.

3 | **Ezartzea**

Algoritmoa aukeratutako programazio-lengoiara itzultzea.





Eduki digitalak
sortzea

A1 maila 3.4 Programazioa

Fluxu-diagramak

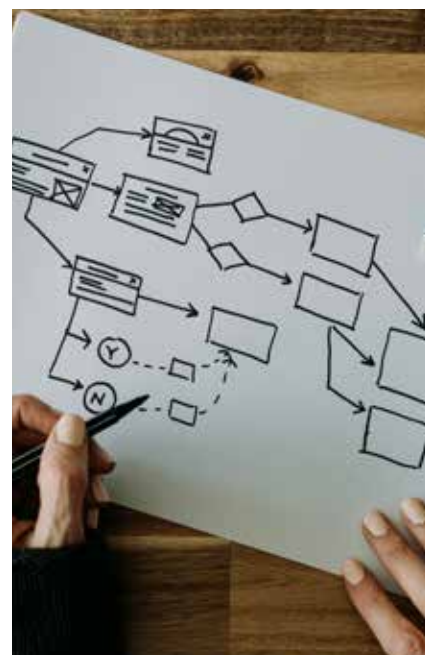




Fluxu-diagramak

Lehen esan bezala, fluxu-diagramek edozein algoritmoren fluxua bisualki irudikatzeko aukera ematen dute, eta, horrela, errazago uler daiteke. Konputagailuen programazioan asko erabiltzen den tresna da, batez ere proiektuak dokumentatzeko eta ideiak eta kontzeptuak trukatzeko softwarea garatzeko ekipoen artean.

Ikus dezagun erabileraren adibide erraz bat. Horretarako, demagun arazo hau: kontzertu baterako sarreren salmenta kontrolatzeko algoritmo bat diseinatu nahi da, adin nagusikoei mugatua. Algoritmoak salmenta baimenduta dagoen edo ez zehaztu behar du. Arestian aipatutako problematika konponduko lukeen algoritmoa honako hau izan liteke:



1 | Programa hastea

2 | Erabiltzaileari jaiotza-urtea eskatzea

3 | Erabiltzaileari oraingo urtea eskatzea

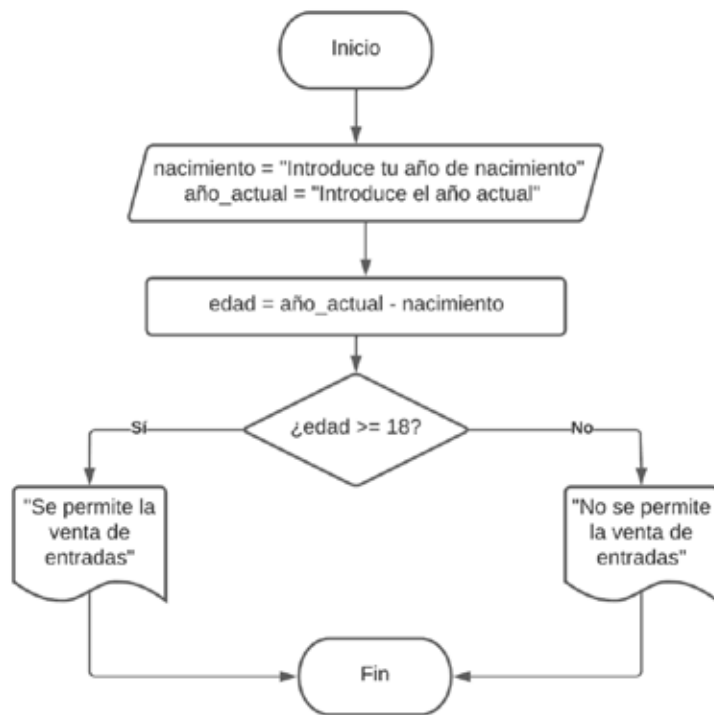
4 | Adina kalkulatzeko aurreko bi datuetan oinarrituta hau da, oraingo urtearen eta jaiotza-urtearen arteko aldea

5 | Adina kontsultatzea eta 18 urtekoa edo hortik gorakoa den egiaztatzea

- Baiezkoa bada, hau da, 18 urte edo gehiago baditu, programak mezu bat erakutsi behar du pantailan, salmenta baimenduta dagoela jakinarazteko.
- Erantzuna ezezkkoa bada, hau da, adina 18 urtetik beherakoa bada, salmenta baimenduta ez dagoela jakinarazi behar da pantaila bidez.

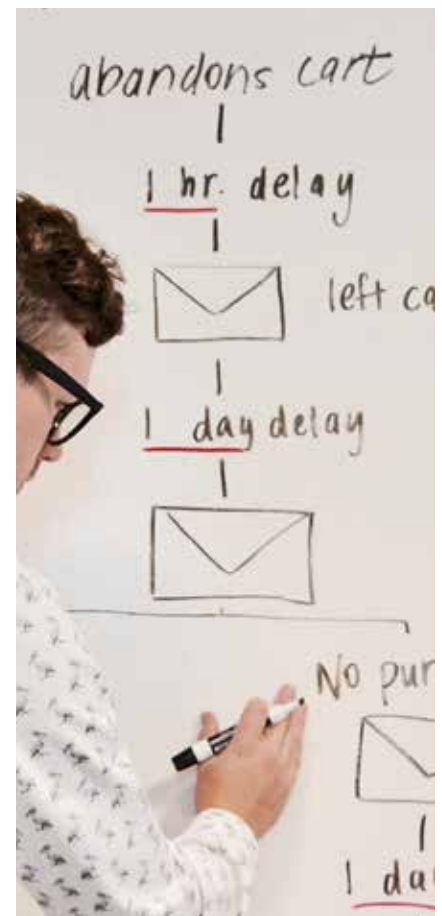
6 | Programa amaitzea

Adina kalkulatzeko zehaztasun handiagoa behar bada ere, jaiotza-eguna eta -hilabetea kontuan hartuta, lehen adibide honetarako jaiotza-urtea soilik sinplifikatzea erabaki da. Aurreko algoritmoak adierazpide baliokidea du fluxu-diagrama batean, hurrengo irudiak erakusten duen bezala.



1. irudia. Sarrera-salmenta adinaren arabera kontrolatzeko algoritmo bati lotutako fluxu-diagrama.

Hasieran, programaren hasiera markatzen da. Programak exekutatu behar dituen lehenengo bi instrukzioak sarrera-datuak eskatzeari dagozkio, paralelogramo baten bidez adierazita. Erabiltzaileak jakin dezan programak datuak eskatzen dituela eta noiz sartuko zain dagoela, beharrezkoa da mezuak pantailan erakustea. Lehenik eta behin, «Sartu jaiotze-urtea» esaldia agertzen da pantailan, eta teklatu bidez sartutako datua *jaiotza* izeneko aldagaian gordetzen da. Bigarrenik, «Sartu oraingo urtea» esaldia agertzen da pantailan, eta sartutako datua *egungo_urtea* aldagaian gordetzen da. Ondoren, *adina* kalkulatzen da, aurreko bi aldagaietan gordetako edukia kenduta. Prozedura bat denez, laukizuzen baten bidez adierazten da. *Adina* kalkulatu ondoren, haren balioa kontsulta daiteke (erronbo baten bidez adierazita). Ebaluazioak benetako balioa edo balio faltsua ematen duen kontuan hartuta, algoritmoaren fluxuak beste bide batetik jarraitzen du. Ezkerreko bidea, hemezortzi urte edo gehiago izanez gero, eta eskuineko bidea, bestela. Bi kasuetan, mezu bat agertuko zaio erabiltzaileari, hartutako erabakiaren berri emateko. Azkenik, programa amaitzea markatzen da.





Creación de
contenidos digitales

A1 maila 3.4 Programazioa

Makina programagarriak. Programaren kontzeptua

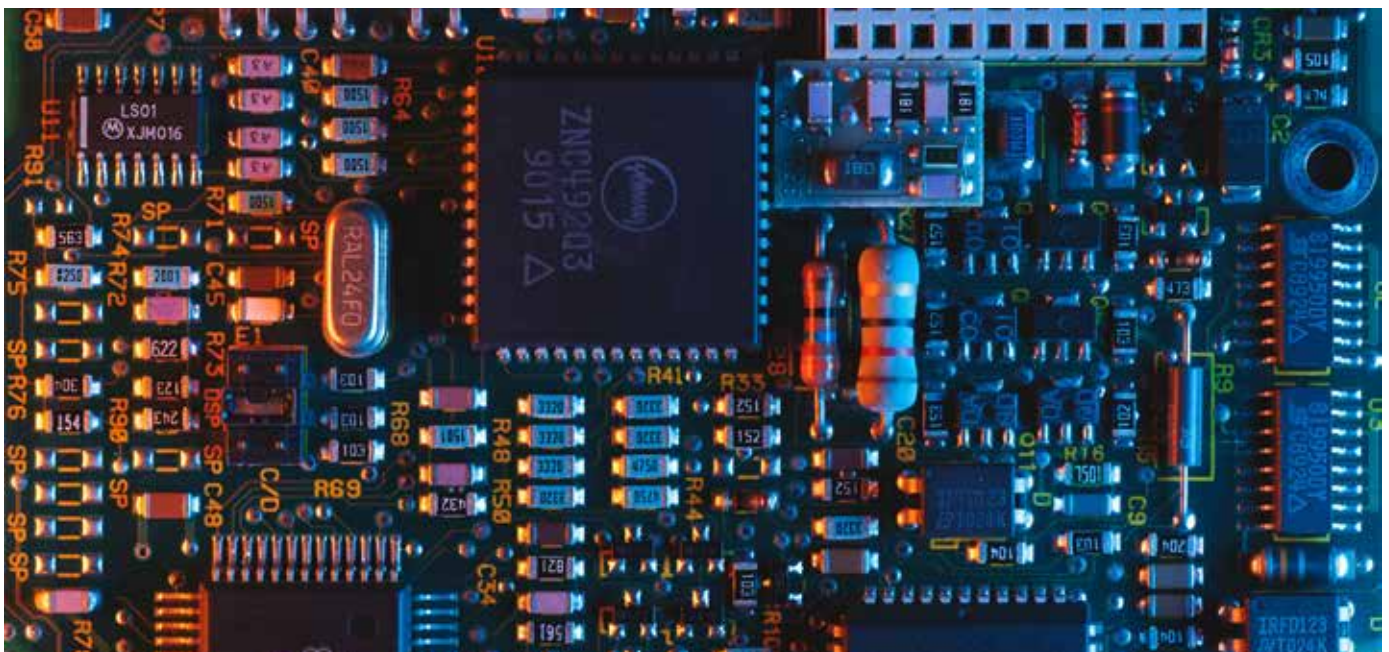




Makina programagarriak. Programaren kontzeptua

Gure egunerokotasunean erabiltzen ditugun mugikorrek, tabletek, eramangarriek eta ordenagailu edo konputagailu pertsonalek (oro har **makinatza** har ditzakegu) elkarren artean konektatutako txip ugari osatutako zirkuitu integratuen multzoa dira, eta gure eguneroko jardueretan zeregin ugari egiteko aukera ematen digute. Lan- eta eskola-eremuan, lanerako tresna dira, eta eremu pertsonalean, berriz, harremanetarako, garapen pertsonalerako eta entretenimendurako tresna. Zalantzarik gabe, makina edo gailu informatiko horiek funtsezkoak dira gure eguneroko bizitzan.

Oso ariketa erraz bati buruz hausnartu dezakegu, eta pentsa dezakegu zenbat gailu elektronikok osatzen duten prozesatze-gailuren bat. Telebistak, etxeko zinema-ekipoak, bideokontsolak, laguntzaile pertsonalak (Alexa, adibidez), maskotentzako janari-banagailu automatikoak, hozkailuak, garbigailuak, galdarak eta aire girotuko ekipoak inguratzen gaituzten eta teknologian oinarrituta dauden elementuen adibide batzuk dira, makina txiki askok gauzatzen dituztenak.



Plaka nagusia edozein ordenagailuren funtsezko osagaietako bat da.



Gailu informatikoen edo makinaren erabilera orokorra beharrezkoa da, bestalde, problemen aniztasuna edo horiek aplikatu daitezkeen arloak kontuan hartzen badira, eta horrek lan gehigarria dakar arazo partikularrak konpontzea nahi denean. Lan hori makinari edo gailuari zer egin behar duen adieraztean datza. Bestela esanda, haiekin komunikatu behar dugu, zer egin behar duten agintzeko. Zoritxarrez, eta hizkuntza naturala prozesatzeko teknikan egin berri diren aurrerapenak gorabehera, oraindik ez dugu hizkuntza bera erabiltzen, batez ere makina bati konpondu beharreko arazoa adierazteari dagokionez. Puntu honetan, ez dugu esan nahi gure zinema gogokoenean film bat estreinatzeko sarrerak erreserbatzen dizkigun laguntzaile birtual bati esan behar diogunik. Makina batek ataza jakin bat egiteko eman behar dituen instrukzioak transmititzeaz ari gara.

Horrela, **programatzea** zera da, aplikazio batek duen funtzionaltasuna makinarentzako instrukzio edo agindu bihurtzeko ekintza, zeinak, batzuetan, horiek exekutatu baititu. Programazioaren bidez, makinari nola jokatu behar duen aginduko diogu.

Makina bat programatzea zer den argi dagoenean, komeni da **programaren** kontzeptua definitzea, hau da, garatzen ari den aplikaziorako nahi den funtzionaltasuna duten instrukzio eta aginduen multzoa definitzea. Gure makinan exekutatu dugun aplikazio bakoitza programa bat da, eta exekutatu behar dituen instrukzioak ditu. Instrukzio horiek modu sekuentzian exekutatu dira, eta exekuzio-fluxuan aldaketak egingo dira, instrukzioek berek edo kanpoko gertaerek eraginda, hala nola erabiltzailearen interakzioak teklatuaren edo saguaren bidez.

Gogoratu makina edo ordenagailua osagai fisiko osatuta dagoela. Hauek dira garrantzitsuenetako batzuk: plaka nagusia, elementu guztiak konektatzen dituen gailua; prozesadorea edo CPU, programen instrukzioak betearazteaz arduratzen dena; RAM memoria, informazioa biltegitratzea ahalbidetzen duena, informazioa azkar eskuratzeko; disko gogorra, informazioa biltegitratzeko mekanismo iraunkorragoa eskaintzen duena; txartel grafikoa, irudien eta bideoen tratamendua optimizatzen duena; edo soinu-txartela, soinu-elementuak kudeatzeaz arduratzen dena. Hala ere, norbaitek idatzi eta fitxategi batean gordeko dituen instrukzioen multzoa da programa. Hona iritsita, galdera hau egin dezakegu: nola komunikatuko gara

```
$_SESSION['_CAPTCHA']['c  
return array(  
    'code' => $captcha_co  
    'image_src' => $image  
);  
  
function_exists('hex2rgb')  
function hex2rgb($hex_str, $  
    $hex_str = preg_replace(  
    $rgb_array = array();  
    if (strlen($hex_str) == 4  
        $color_val = hexdec($  
        $rgb_array['r'] = 0xFF  
        $rgb_array['g'] = 0xFF  
        $rgb_array['b'] = 0xFF  
    } elseif (strlen($hex_str) == 6  
        $rgb_array['r'] = hexde  
        $rgb_array['g'] = hexde  
        $rgb_array['b'] = hexde  
    } else {  
        return false;  
    }  
    return $return_string ? impl  
  
image  
_GET['
```

Iturburu-kodearen zatia.



edo instrukzioak nola emango dizkiegu makina osatzen duten elementu fisiko elektronikorik horiei? Argi dago bi mundu horien artean lotura bat ezarri behar dela: i) mundu fisikoa, makinaren osagaiek adierazten dutena, eta ii) mundu logikoa, programek adierazten dutena.

Azkenik, interesgarria da jakitea programazioa formalki software-erakuntza deitzen dugunaren barruan sartzen dela, hau da, normalean programazio gisa ezagutzen duguna baino haratago doan prozesu konplexu eta sofistikatu baten barruan. Ildo horretan, softwarearen eraikuntza zenbait etapa orokorretan banatzen da: i) arazoaren definizioa, ebatzi nahi den problema argi eta garbi definitzeko; ii) baldintzen zehaztapena edo definizioa, software-sistemak egin behar duenari buruzko deskribapen zehatza egiteko aukera ematen duena; iii) garapenaren plangintza, softwarea eraikitzeko gainerako faseen plangintza zehazteko helburua duena; iv) diseinua, programaren kodea nola egituratuko den antolatzeko; v) programazioa edo kodetzea, emaitza bat sortzea helburu duen sententzia multzo baten idazketa gisa ulertuta; vi) probak, sortutako programak behar bezala funtzionatzen duela egiaztatzen duena; eta, vii) mantentzea, kodea eguneratzearekin lotutako fasea.





Eduki digitalak
sortzea

A1 maila 3.4 Programazioa

Programazio- lengoiak. Definizioa eta bilakaera





Programazio-lengoaiak. Definizioa eta bilakaera

Ordenagailuak osatzen dituzten zirkuituek bi tentsio-mailarekin lan egiten dute. Zirkuitu digitalak dira, eta bi maila horiek 0 eta 1 zenbakiekin lotzen dira. Ezaguna al zaizu orain *sistema bitarra* terminoa? Gailu informatiko batek egiten dituen ekintzak 0 eta 1eko sekuentzia-multzo baten bidez adieraziko dira.

Lehen honela definitu dugu programa bat: «garatzen ari den aplikaziorako nahi den funtzionaltasuna duten instrukzioen multzoa», programa hori berriz sortuko da ordenagailu baten memoria nagusian, eta, horretarako, 0 eta 1eko sekuentzietan *itzuliko* dira instrukzioak. Makinak zuzenean interpreta dezakeen kode-sistema horri **makina-lengoaia** esaten zaio.

Makina bakoitzak bere *makina-lengoaia* du, eta lengoaia horrekin programa daiteke. Lengoaia hori berariazkoa da makina horren barne-arkitekturarako. Bestela esanda, makina batek instrukzio horiek edo berarentzat programatutako eragiketa-kodeak baino ezin ditu ezagutu. Makina-lengoaia guztiek antzeko instrukzioak dituzte: eragiketa sinpleak, adibidez, informazioa kopiatzeko; eragiketa aritmetikoak; eragiketa logikoak, balio boolearrak maneiatzen dituztenak (egia eta gezurra); eta sarrera- eta irteera-eragiketak, makinaren kanpoaldearekiko konexioekin lotutakoak.

Makina-lengoaiako programak oso modu eraginkorrean gauzatzen dira, interpretatu eta gauzatu behar dituzten zirkuituetarako berariaz idazten baitira. CPUk zuzenean interpreta dezake kode hori, eta ez du alde aurreko eraldaketarik behar exekutatzeko. Hala ere, makina-lengoaiaren programazioa lan zaila eta izugarri neketsua da programatzailearentzat, eta beharrezkoa da programatzaileak makinaren arkitektura fisikoa xehetasun handiz ezagutzea.

Informazio gehiago

John Backusen hitzetan, makina-lengoaiaren programazioa zera zen: «arte iluna, materia arkanoa, gutxi batzuen eskura baino ez zegoena; halakoak apaiz klase bateko kideztat hartzen hasi ziren, zeinek pertsona arruntentzat konplexuegiak ziren trebetasun eta misterio batzuk zaintzen zituzten; aurre egiten zioten, orobat, programazioa edonork egiteko bezain sinplea bihur zezakeen aldaketa iraultzaile orori».



Hain zuzen ere, programatzeko zailtasun hark ekarri zuen helburu hau: «ordenagailuak programatzeko lana demokratizatzea», programatzaileen prestakuntza-aldia murriztuz eta makinak programatzeko zeregina erraztuz. Horrela, asmoa zen handitzea programen eta makinaren altzairu hotzaren arteko distantzia, eta gutxitzea, aldi berean, kodearen eta programatzaileen arteko distantzia. Horretarako, ezinbestekoa da, aurrerago aztertuko dugun bezala, independentzia-maila bat ezartzea programaren eta programa gauzatuko den arkitekturaren artean.

Norabide horretan egindako lehen aurrerapen esanguratsuetako bat izan zen notazio sinboliko edo *mnemonikoa* erabiltzea; hain zuzen ere, makinaren instrukzio edo eragiketa-kode bakoitza adierazteko. Gako mnemotekniko horiek gogoratzen errazagoak ziren zenbakizko kodeak baino, baina, hala ere, beharrezkoa zen, behin problema soluzionatzen zuten instrukzio mnemonikoen sekuentzia ezarrita, gakoak makina-lengoaiara itzultzea. Hizkuntza horri **mihiztadura-lengoaia** deitu zitzaion.

Ondorengo kode-zerrendan 8086 arkitektura baterako mihiztadura-lengoaiako bi zenbakiren baturaren adibidea dago, zeinak 4000 eta 4002 memoria-helbideetan gordeta dauden. Emaiza 5000 eta 5002 helbideetan gordetzen da. Adibide honen bidez helburua da erakustea zer jauzi dagoen programazioa makina-lengoaiari, hau da, zuzenean zeroen eta baten bitartez egitetik programazioa maila semantiko handiagoko notazio baten bidez egitera.

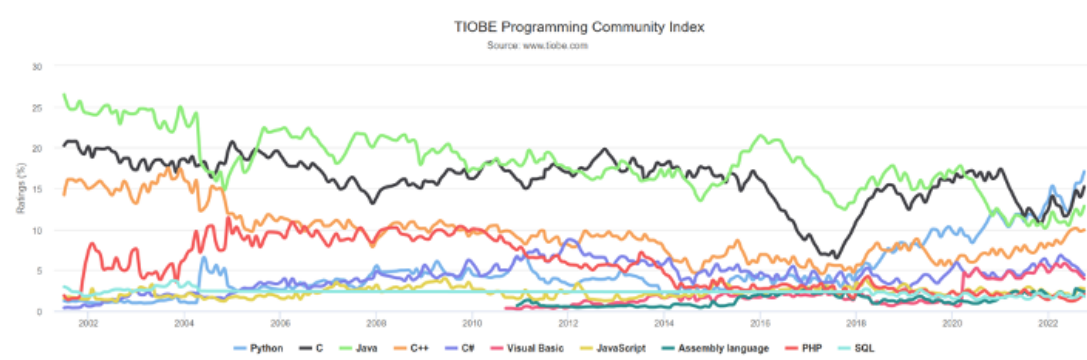
2000	MOV	CX, 0000	# Kargatu 0000 CX erregistroan
2003	MOV	AX, [4000]	# Kargatu 3000 zenbakia AX metagailuan
2007	MOV	BX, [4002]	# Kargatu 3002 BX erregistroan
200B	ADD	AX, BX [AX]	#Batu AX eta BX-en edukia
200D	JNC	2010	# Joan 2010 helbidera, batuketan bururakoa bat ez bada
200F	INC	CX	# Handitu AX metagailuaren edukia
2010	MOV	[5000], AX	#Mugitu AX-en edukia 3004 posiziora
2014	MOV	[5002], CX	#Mugitu CX-en edukia 3004 posiziora
2018	HLT		# Gelditu programa



Hala ere, korrespondentzia estua dago, banakakoa orokorrean, mihiztadura-lengoaiaren eta makinaren eragiketen kodeen artean, horregatik, programazioa makinatik hurbil dago oraindik, eta prozesu zehatza eta konplexua izaten jarraitzen du. Testuinguru horretan, programatzailearengandik ahalik eta hurbilen dagoen lengoia sortu behar da, makinak egin beharreko ekintzak adierazteko. Hor agertzen da **goi-mailako programazio-lengoaiaren** kontzeptua.

Goi-mailako programazio-lengoaiak *abstrakzioaren* nozioarekin lotu ohi dira. Bestela esanda, horrelako lengoia bat erabiltzen duzunean, arreta jar dezakezu programaren sorreran, alde batera utzita programa exekutatu duen makinaren xehetasunak. Ikuspegi honen abantaila nagusia da programazioa sinplifikatzea eta ulergarriagoa egitea, neurri batean programazio-sententziek giza hizkuntzarekin antz handiagoa dutelako behe-mailako programazio-lengoia batekin edo, are, mihiztadura-lengoiairekin izango luketena baino. Aurrerago azalduko dugun moduan, goi-mailako lengoia erabiliz gero, sententziak makina-lengoiaira itzultzen dituzten tresnak behar dira.

Azkenik, eta imajina dezakezun bezala, gaur egun ez dago goi-mailako programazio-lengoia bakar bat. Aitzitik, askotariko lengoaiak daude, eta aldatzen diren zer diseinu-helburu eta ezaugarri dituzten. Azken urteetako edo hamarkadetako ospetsuenetako batzuk C, Java, Python eta JavaScript dira. 1. irudiak 2002. urteaz geroztik gehien erabiltzen diren goi-mailako programazio-lengoaiak erakusten ditu.



1. irudia. TIOBE indizeak goi-mailako lengoia ezagunen bilakaera erakusten du. 2022ko urrikoa.



Eduki digitalak
sortzea

A1 maila 3.4 Programazioa

Interpretatzaileak VS konpiladoreak





Interpretatzaileak VS. Konpiladoreak

Makinen programazioa errazteko, goi-mailako programazio-lengoaiak erabiltzen dira, hala nola C++ eta Python. Lengoaia horiek aukera ematen diote programatzaileari zer egin nahi duen makinari adierazteko; erabilitako lengoaia harengandik hurbil dago baina urruti, ordea, makinak ulertzen duen horretatik, hau da, makina-lengoiatik.

Horrela, lengoaien itzultzaile edo prozesadore baten beharra sortzen da, programatzaileak emandako instrukzioak bihurtzeko daitezkeen makinak exekutatzeko moduko instrukzio.

Oinarri horren gainean, programa berezi batzuk egin daitezke, zeinak diseinatuta baitaude goi-mailako programazio-lengoiarekin idatzitako programak –alegia, gizakiek uler ditzaketenak– hau bihurtzeko: makina-lengoaia edo kode bitar, makinek uler dezaketenak. Programa horiek **konpiladoreak eta interpretatzaileak** dira. Adibidez, C++ programazio-lengoiarako konpiladore bat egoteak aukera ematen du C++ lengoian idatzitako programa bat hardware arkitektura jakin bati lotutako makina-lengoaia bihurtzeko. Era berean, Python programazio-lengoiarako interpretatzaile bat egoteak aukera ematen du Pythonen idatzitako programa bat makina berean edo beste makina edo arkitektura fisiko batean gauzatzeko.

Konpiladoreek eta interpretatzaileek helburu bera dute, hau da, kode idatzia makina-lengoiako goi-mailako lengoaia bihurtzea, baina **alde** handiak daude bien artean. Hauek dira garrantzitsuenetako batzuk:

- Konpiladoreek programa baten kodea kode bitar bihurtzen dute makina jakin batean exekutatu aurretik. Bestela esanda, programa exekutatu aurretik, konpilatu egin behar da. Prozesu horrek ziurtatzen du, programa erabili aurretik, kodea lengoaiaren espezifikazioari jarraikiz idatzi dela. Hala ere, interpretatzaileek ez dute makina-koderik sortzen, zuzenean ulertzen dituztelako eta makinan exekuta daitezkeen instrukzio bihurtzen dituztelako.
- Kode konpilatua interpretatutakoa baino azkarrago exekutatzen da. Horrek esan nahi du eremu batzuetan hobe dela lengoaia konpilatua erabiltzea, lengoaia interpretatu bat erabili ordez. Adibidez, bideojokoen mundua. Bertan, programak, hau da, bideojokoak, bigarren mailako iruditasa altu batekin funtzionatu behar du animazio-sentsazioa emateko, jokalariarekin elkarreragiteaz bat.





- Konpiladore batek programa baten iturburu-kode osoa aztertuko du erroreak aurkitzeko, exekutatu aurretik. Interpretatzaileak, aldiz, lerroz lerro egingo du.

Konpiladoreen eta interpretatzaileen testuinguru horretan, *programazio-lengoaia konpilatuak* eta *programazio-lengoaia interpretatuak* erabiltzen dira, makina-kodea konpiladoreen edo interpretatzaileen bidez sortzen den kontuan hartuta. Programa bat sortzeko programazio-lengoaia aukeratzean, hainbat aldagai hartu beharko dira kontuan. Nabarienetako batzuk honako hauek dira: lantaldeak programazio-lengoaia jakin batekin zer esperientzia duen, programak zer errendimendu behar duen (interpretatutako lengoaiak exekuzio-denbora handiagoekin lotuta daude) eta lengoaiak zer gaitasun duen programatzaileak programak bizkor sor ditzan (interpretatutako lengoaiak abantaila izan ohi dute konpilatuekiko).





Eduki digitalak
sortzea

A1 maila 3.4 Programazioa

Adierazpenak eta esleipena





Adierazpenak eta esleipena

Adierazpenak programen konputatzeko funtsezko mekanismoa dira, eta programa bat garatzen dugunean erabil dezakegun instrukziorik oinarritzkoena da. Adierazpenek **balioak** (adibidez, 4 edo 546) eta **eragileak** (adibidez, + edo -) dituzte, eta konputagailuak **ebalua** ditzake balio bakarra lortzeko edo nahi den efekturen bat eragiteko.

Adibidez, **2+3** adierazpena erabiliz bi zenbaki batzen dituen adierazpena eraiki genezake. Adierazpen hori honako hauer osatuta egongo da: 2 eta 3 balioez eta batuketa-eragileaz (+). Adierazpen horren ebaluazioak **5** balioa dakar. Horrela, zenbait modutako adierazpenak sor ditzakegu eragingaiak (zenbakiak, aldagaiak, etab.) eta eragileak (matematikoak, logikoak, etab.) konbinatuz.

Ikus ditzagun programazio-lengoaia gehienetan dauden adierazpen motak.

Aldagaiak esleitzeko adierazpenak

Lehen ikusi dugun bezala, aldagaiek balioak gordetzeko aukera ematen diote programari, ondoren haiekin lan egiteko. Ildo horretatik, aldagaiak idealak dira adierazpen bat ebaluatzearen ondoriozko balioa gordetzeko eta, horrela, programan zehar adierazpenak kateatzeko.

Teknika hau erabiliz, adibidez, **23** balioa **zenbakia** izeneko aldagaian gordetzea lortuko dugu, honako adierazpen hau erabiliz:

Mota horretako adierazpenei **aldagaiak esleitzeko**

```
zenbakia = 23
```

adierazpenak esaten zaie, eta (=) esleipen-eragileaz osatuta daude, zeinak aldagaia esleitzeko adierazpena ezkerreko eta eskuineko zatietan banatzen baitu. Eragilearen eskuinaldean, gorde nahi dugun ebaluazioa dagokion adierazpena egongo da; ezkerrean, berriz, ebaluazio horren emaitza gordetzeko balioa duen aldagaia. Ohar gehigarri gisa, balio bat programan erabili aurretik aldagai batean lehenengo aldiz gordetzeari «aldagai bat hasieratzea» esaten zaio. Hori





garrantzitsua da, balitekeelako aldagai batek ustekabeko balioa izatea hasieratu ezean, eta horrek programan erroreak eragitea. Gainera, aldagaia hasieratzeak ziurtatzen du beti balio baliozkoa eta ezaguna izatea, eta horrek erroreak zuzentzen eta kodea mantentzen lagunduko du.

Ikus ditzagun horrelako adierazpenen adibide batzuk eta haien ebaluazioen emaitzak:

```
1 | zenbaki_bakoitia = 23
2 | zenbaki_bikoitia = 4
3 | emaitza = zenbaki_bakoitia + zenbaki_bikoitia
4 | zenbaki_bikoitia = emaitza
```

1. lerroan aldagai bat hasieratu da esleipen-eragilearen (=) eskuinaldeko adierazpena ebaluatzeko, eta horrela 23 balioa gordetzen da `zenbaki_bikoitia` aldagaian. Era berean, 2. lerroan, beste aldagai bat (`zenbaki_bikoitia`) hasieratu da, 4 balioarekin. Ondoren, 3. lerroan exekutatuako adierazpenean 27 balioarekin sortutako aldagaien batura ebaluatzen da, eta emaitza esleitzen zaio `emaitza` aldagaiari. Azkenik, 4. lerroan, `emaitza` (27) aldagaiaren balioa esleitu zaio `zenbaki_bakoiti` aldagaiari, eta azken horren (23) balioa lehenengoarekin (27) ordeztu da.

⚠ ADI

Testuan aipatu berri den bezala, aldagai bat hasieratu (*sortu*) daiteke balio bat esleitzen zaion lehen aldian. Horren ondoren, aldagai hori ondorengo edozein adierazpenetan erabil daiteke, beste aldagai eta balio batzuekin. Aldagai horren ondorengo esleipenek ebaluazio berriaren ondoren lortutako emaitzarekin ordeztuko dute aurretik zuen balioa, baina izen bereko beste aldagai bat sortu gabe.

Programaren kodean definitutako beste adierazpen batzuetatik abiatuta aldagaiak esleitzeko adierazpenak erabiltzeaz gain, badira beste datu-iturri batzuetatik datuak lortzea eragiten duten adierazpenak, hala nola beste sarrera-gailu batzuetatik. Kasu honetan, ari gara aldagaiak esleitzeko adierazpen bat exekutatzeaz, balioak konputagailuaren teklatutik bertatik



irakurtzeko gai den beste adierazpenen bat ebaluatuz, adibidez:

```
1 | agurra = irakurri_teklatua()
```

1. lerroan, esleipen-eragilearen eskuineko espresioak programaren exekuzioa geldituko du, harik eta erabiltzaileak teklata erabiliz testuren bat sartu eta [ENTER] tekla sakatu arte. Adierazpenaren ebaluazioa amaitzean, teklatu bidez sartutako testua izango da agurraren aldagaiari esleituko zaion emaitza.

Albo-ondorioak eragiten dituzten adierazpenak

Orain arte ikusi ditugun adierazpenak aldagaiak esleitzeko adierazpenak osatzeko erabil daitezke, haien ebaluazioaren emaitza aldagai batean gorde baitaiteke. Hala ere, beste adierazpen batzuen ebaluazioak ez du emaitzarik ekartzen, eta exekuzio-ekosistemaren egoeran **albo-efektua eragiten du** (adibidez, interpretatzailea, sistema eragilea, hardwarea, etab.). Beraz, ezin ditugu adierazpen horiek erabili esleipen-adierazpenetan.

Adibidez, ordenagailuaren pantailan testua erakusten duen adierazpen batek ez du baliorik itzuliko; aitzitik, pantaila hori erabiliko du adierazpen hori ebaluatzeko emaitza erakusteko:

```
1 | pantaila_erakutsi ('Kaixo, mundua!')
```

Adibide honetan, adierazpenaren ebaluazioak «Kaixo, mundua!» testua agerraraziko du pantailan. Hori da adierazpen hori gauzatzean espero den portaera. Hala ere, kontuan izan adierazpenaren emaitza ez zaiola aldagai bakar bati ere esleitzen; aitzitik, ebaluatu egiten da, eta testua pantailan erakustea du albo-efektu gisa.





Adierazpen aritmetikoak

Adierazpen aritmetikoek balio numerikoen gaineko eragiketa matematikoak egiteko aukera ematen digute. Programazio-lengoaiak eragile aritmetikoen multzo komun bat ematen dute, eta adierazpen konplexuagoak eraikitzeke erabil ditzakegu:

Eragilea	Eragiketa	Adierazpen-adibidea	Ebaluazio-adibidearen emaitza
**	Esponentea	$3^{**} 3$	27
%	Zatiketaren hondarra	$55 \% 7$	6
//	Zatiketa osoa (moztua)	$55 // 7$	7
/	Zatiketa dezimalekin	$55 / 7$	7.857...
*	Biderketa	$4 * 5$	20
-	Kenketa	$6 - 2$	4
+	Batura	$1 + 2$	3

Adibidez, aurreko taula erabiliz, honelako adierazpenak egin genitzake:

```

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

```

2. lerroaren adierazpena ebaluatzearen emaitza baliteke nahasgarria izatea, eta, beraz, beharrezkoa da jakitea eragile aritmetikoak adierazpenetan zer ordenatan ebaluatzen diren. Normalean, eragileen lehentasun-ordena eta matematikaren esparruan operadoreek dutena oso antzekoak dira: lehenik, esponente-eragilea ebaluatzen da (**); ondoren, eragileok: biderketa (*), zatiketa osoa (//), zatiketa dezimalekin (/) eta zatiketaren hondarra (%), ezkerretik eskuinera agertzen diren ordena errespetatuta; azkenik, batuketaren (+) eta kenketaren (-) eragileak. Nolanahi ere, parentesia erabiltzea komeni da, adierazpen konplexuagoak errazago irakurtzeko (3. lerroa).



Adierazpen alfanumerikoak

Adierazpen alfanumerikoak testu- eta zenbaki-balioez osatutakoak dira. Programazio-lengoaletan, eragiketa batzuk testu-kateen eta zenbakizko balioen artean egin ditzakegu.

Alde batetik, **testu-kateak kateatu** ditzakegu (+) batuketa-eragilea erabiliz:

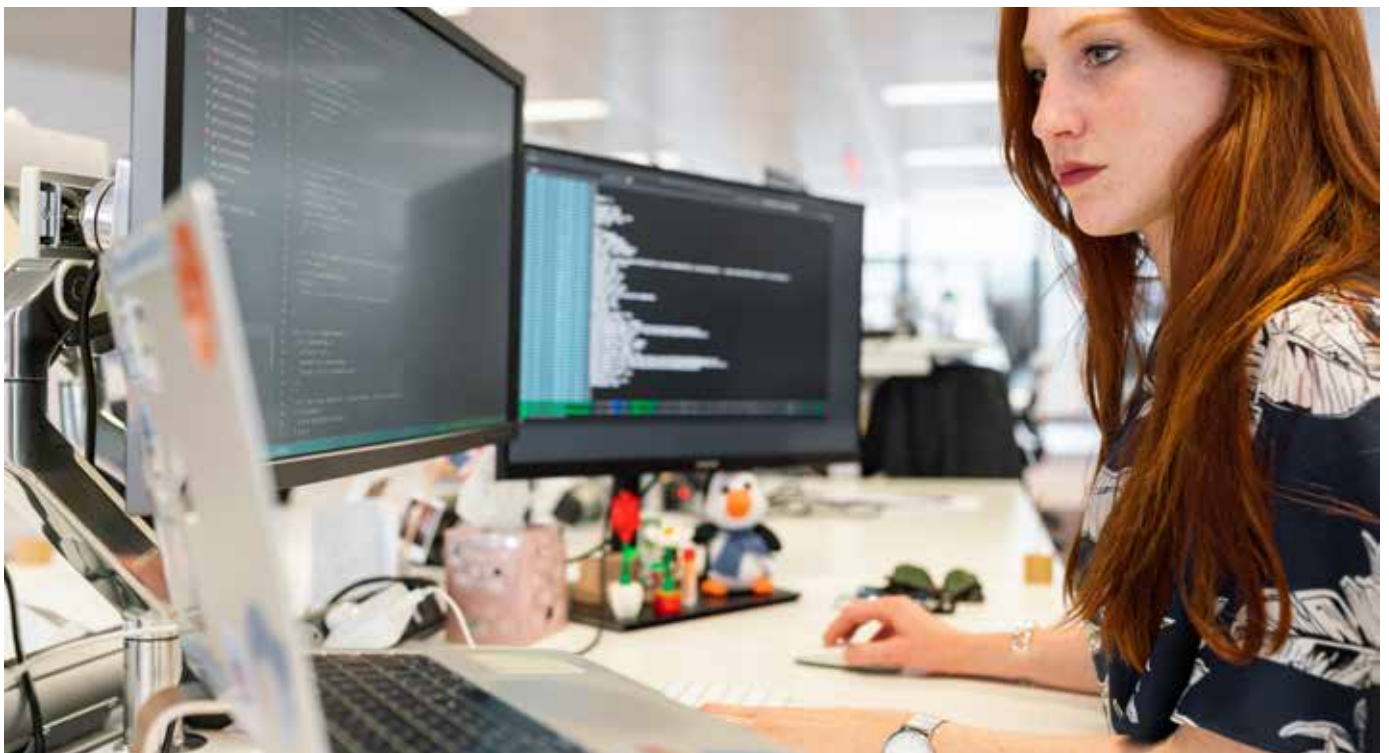
```
1 | 'Kaixo' + 'Agur'
```

Adibide honetan, adierazpen hori ebaluatzearen emaitza «KaixoAgur» testua izango da.

Bestalde, **testu-kateak** nahi adina aldiz erreplika ditzakegu biderketa-eragilea (*) erabiliz:

```
1 | 'Kaixo' * 3
```

Horren emaitza «KaixoKaixoKaixo» testua izango da, jatorrizko testua 3 aldiz erreplikatu baita.





Adierazpen erlazionalak eta logikoak

Adierazpen erlazionalak egiazko edo gezurrezko emaitzak dituzte, haien ebaluaziotik abiatuta. Horrelako adierazpenak baldintzazko sententziak egiteko erabiltzen dira, programei erabakiak hartzen uzteko eta, ondoriozko exekuzio-fluxua aldatzeko.

Programazio-lengoaiek eragile erlazionalen multzo bat ematen dute, eta horrelako adierazpenak egiteko erabil ditzakegu.

Normalean:

Eragilea	Eragiketa
==	Berdin
!=	Desberdin
<	Txikiago
>	Handiago
<=	Txikiago edo berdin
>=	Handiago edo berdin

Hona hemen adierazpen erlazionalen adibide simple batzuk:

```
1 | 'Kaixo' == 'Kaixo'  
2 | 'Kaixo' == 'Agur'  
3 | 'Kaixo' != 'Agur'  
4 | 7 < 8  
5 | 7 < 7  
6 | 7 <= 7
```

Adierazpen horiek **True** (1., 3., 4. eta 6. lerroak) eta **False** (2. eta 5. lerroak) balioetan ebaluatuko lirateke.



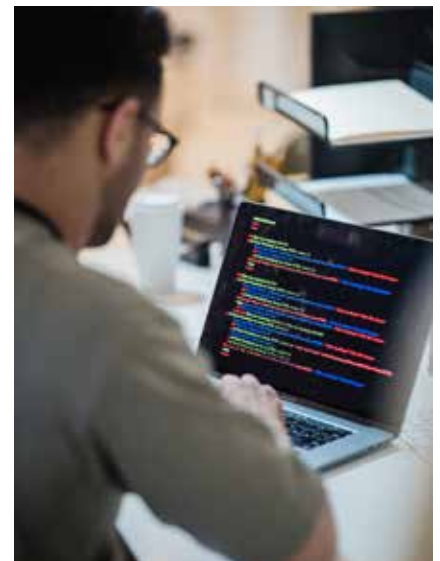
Bestalde, zenbait eragile logiko (**and**, **or** eta **not**) erabil ditzakegu zenbait adierazpen erlazional konbinatzeko eta, horrela, **adierazpen logiko** deritzenak egiteko:

Adierazpena	Ebaluazioaren emaitza
True and True	True
True and False	False
False and True	False
False and False	False
True or True	True
True or False	True
False or True	True
False or False	False
not True	False
not False	True

Hona hemen, adibidez, horrelako adierazpen batzuk:

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

Lehenengo lerroan, parentesi arteko adierazpen erlazionalak ebaluatuko liriateke, eta **True and True** bezalako tarteko adierazpena lortuko da. Ondoren, adierazpen hori berriro ebaluatuko da, eta azken emaitza **True** balioa izango da. Adibideko gainerako adierazpenen ebaluazioa antzera egingo da, eta **False** (2. lerroa) eta **True** (3. lerroa) emaitzak lortuko dira.





Eduki digitalak
sortzea

A1 maila 3.4 Programazioa

Exekuzio- fluxuaren kontrola





Exekuzio-fluxuaren kontrola

Programa baten exekuzio-fluxua terminoak elkarren artean zerikusia duten hiru kontzeptu hartzen ditu barnean: i) fluxua, ii) exekuzioa eta iii) programa. Jarraian definituko ditugun arren, jakin behar duzu oso erlazio hurbila dagoela, maila semantikoa, programa baten exekuzio-fluxuaren eta fluxu-diagramaren ideia artean; azken hori alde aurretik jorratu dugu, ikuspegi bisualetik, algoritmoaren kontzeptuaz aritzean.

Programaren kontzeptua, alde aurretik aztertuta, honela definitzen da: instrukzio-sekuentzia bat, ordenagailuak ulertzeko modukoak eta helburu ondo zehaztua dutenak. Bigarrenik, *exekuzio* hitzak, hain zuzen ere, makinak programak egin behar duena egiteko duen gaitasunari egiten dio erreferentzia, horretarako eskura dauden baliabide fisikoak edo hardwarea erabiliz. Adibidez, baliabide horietako bat aldagai baten balioa biltegitratzeko edo gordetzeko erabiltzen den ordenagailuko prozesadorearen erregistroa da. Azkenik, *fluxuaren* kontzeptuak islatzen du exekutatzen ari diren programek duten dinamismoaren nozio hori, zeinak ezartzen baitu abiarazten denetik amaitzen den arte programak egingo duen bidea. Bide horren urrats bakoitza kode-instrukzio batek irudikatzen du. Horri lotuta, garrantzitsua da azpimarratzea ezen, programa bat instrukzio-sekuentzia batez osatuta egon arren, ez dituela zertan haietako guztiak ordena jakin batean exekutatatu. Aitzitik, programak instrukzioen azpimultzo bat exekutatuko du, baldintza jakin batzuen arabera.

Baldintza horiek programatzaileak ezartzen ditu zuzenean, programaren exekuzio-fluxua aldatzen edo kontrolatzen duten instrukzioen bitartez. Bestalde, programaren exekuzio-fluxua alda daiteke programan erabiltzen diren aldagaien balio puntualaren arabera. Ondorengo adibideetan ikusiko dugun bezala, programa baten fluxua kontrolatzeko erabiltzen diren instrukzioak aldagaien erabilerarekin konbinatu ohi dira.





Baina programa baten fluxua sententzia konplexuagoen bidez nola kontrolatu aztertu aurretik, ikus dezagun adibide xumeago bat. Ondorengo zerrendak 7 instrukziok osatutako programa baten egitura orokorra erakusten du. Egitura horren arabera, programa hori exekutatzeko duen ordenagailuak, lehenik, **1. sententzia** prozesatuko du. Gero, **2. sententzia** prozesatuko du. Ondoren, **goto etiketa** instrukzioa prozesatuko du. Hona iritsita, irakurleak pentsa lezake programaren exekuzio-fluxuak **4. sententzia** exekutatzeko jarraituko duela, ordena sekuentzialean. Hala ere, **goto** instrukzioak, programa batean fluxu-kontrolaren adibiderik errazena denak, eragingo du programak **etiketa:** instrukziora «jauzi egitea». Hain zuzen ere, **gotoren** hitzez hitzeko itzulpena «hona joan» da, hau da, programaren kodearen beste puntu batera joatea. Bestela esanda, **goto** instrukzioak baldintzarik gabeko aldaketa edo jauzia eragiten du programaren exekuzio-fluxuan, halako moldez non betearazi beharreko hurrengo instrukzioa (**5. sententzia**) **etiketa:** instrukzioaren ondoren idatzitakoa izango den.

1. sententzia
2. sententzia
- goto etiketa
3. sententzia
4. sententzia
- etiketa:
5. sententzia

Esan bezala, baldintzarik gabeko jauziek ez dute inolako baldintzarik behar exekutatzeko, baldintzapeko kontrolaren sententziak, berriz, ideia honetan oinarritzen dira: baldintza edo adierazpen bat ebaluatzea programa baten exekuzio-fluxuan aldaketak eragiteko. Ildo horretan, gogoan izan baldintzapeko sententziak direla konputagailu bati erabakiak hartuz jardutea ahalbidetzen dioten egitura oinarritzkoenak. Ezagutzen duzun baldintzapeko sententziaren adibidean, «baldin eta txakurrak zaunka egiten badu, orduan piztu lorategiko argiak», **baldin eta** gako hitzak baldintza bat jartzen badu galdera gisa (zaunka egiten du txakurrak?). Galderaren erantzuna baiezkota bada, **orduan** gako hitzaren ondoren dagoen zatia izango da exekutatu den ekintza.

⚠ ADI

Baldintzarik gabeko jauzi-sententziak gehiegi erabiltzen dituen kodea *espageti kode* bihurtu liteke, sententzia horiek zaildu egiten baitute programa baten exekuzio-fluxuaren egitura. Espageti-plater batekin konparatuta sortzen da termino hori, eta bertan kodea bera hari nahasi batzuen modura irudikatzen da. Espageti korapilatuak programaren exekuzio-fluxuaren metafora da. Hala, oro har, ez da gomendatzen baldintzarik gabeko jauzi-sententziak erabiltzea.



Baldintzapeko sententzia batean ebaluatu beharreko baldintza lotuta egon ohi da aldagai jakin baten edo, are, batzuen balioa kontsultatzearekin. Hurrengo kode-zerrendak Python programazio-lengoaian idatzitako zatia erakusten du, zeinak egoera horren adibide erraz bat ematen duen. 1. eta 2. lerroetan **a** eta **b** aldagaiak adierazten dira, eta horiei 13. eta 7. balioak esleitzen zaizkie, hurrenez hurren. 4. lerroan baldintzapeko sententzia bat agertzen da; **if** gako hitzak adierazten du Python lengoaian. Ondoren, ebaluatu beharreko baldintza agertzen da, **a > b**. Horrela, baldin eta a-ren balioa b-ren balioa baino handiagoa bada, orduan 5. lerroko sententzia exekutatu da. Hala ez bada, hau da, a-ren balioa b-ren balioa baino txikiagoa edo berdina bada, 7. lerroko sententzia exekutatu da. Begira ezazu 6. lerroan beste gako hitz bat erabiltzen dela, Python lengoaian (**else**), baldintzaren ebaluazioa balio faltsuaren balioak denean exekutatu diren sententziak taldekatzeko.



```
a = 13
b = 7

if a > b:
    print ('a-ren balioa b-rena baino handiagoa da')
else:
    print ('a-ren balioa b-rena baino txikiagoa edo berdina da')
```

Pentsatuko duzun bezala, kodea exekutatzeko baduzu, emaitza honako hau izango da:

```
a-ren balioa b-ren balioa baino txikiagoa edo berdina da
```

Baldintzapeko sententziak zailtzeko aukera dago, bata bestearekin kateatuz, egoera gehiago barnean hartzeko.

⚠ ADI

Baldintzapeko sententziak taldekatzeko erabiltzen den terminoari *habiaraketa* esaten zaio. Hala bada, baldintzapeko sententzia batzuk habiaratu edo kateatu ditzakezu egoera edo baldintza desberdinak kontrolatzeko, eta ekintza multzo desberdinak erlazionatuko dituzu epaia gobernatzen duen baldintzaren ebaluazioaren emaitza zein den.



Illo horretan, aurreko kode-zatia zabaldu egin daiteke, jarraian erakusten den bezala. Arreta jarri 6. lerroko **elif** sententziari. Funtsean, programari zera adierazten ari gatzaizkio: aurreko baldintzaren ebaluazioa, hau da, 4. lerroarena, faltsua izan bada, orduan probatu beste baldintza hau, 6. lerroan definitutakoa. Azkenik, eta aurreko adibidean bezala, 8. lerroko **else** sententzia exekutatu beharreko ekintza adierazteko erabiltzen da (9. lerroa), baldin eta aurreko baldintza guztien ebaluaziotik faltsuak direla ondorioztatu bada.

```
a = 13
b = 7

if a > b:
    print ('a-ren balioa b-rena baino handiagoa da')
elif a == b:
    print ('a-ren balioa b-ren berdina da')
else:
    print ('a-ren balioa b-rena baino txikiagoa da')
```

Beste programazio-lengoaia batzuek **if-then-else** motako sententzia zenbait sententzia bakar batean biltzen dituzte. Adibidez, Java programazio-lengoiak **switch** sententzia ematen du, kodearen ondorengo adibidean erakusten den bezala.

```
int eguna = 3;
switch (eguna) {
    case 1:
        System.out.println («Astelehena»);
        break;
    case 2:
        System.out.println («Asteartea»);
        break;
    case 3:
        System.out.println («Asteazkena»);
        break;
    /* Gainerako egunak hemen: Osteguna, ostirala, larunbata */
    case 7:
        System.out.println («Igandea»);
        break;
}
```



2. lerroko **switch** sententziaren adierazpena behin bakarrik ebaluatzen da. Ondoren, adierazpen horren balioa **switch** sententziaren balio bakoitzarekin alderatzen da; balioak sententzia hori osatzen duten **case** sententzietan lotuta daude. Adibide honetan, **eguna** aldagaiaren balioa alderatzen da **case** sententzien balio bakoitzarekin. Hala, **egunaren** balioa, lehenik eta behin, 1 balioarekin alderatuko da, eta gero, 2. balioarekin, etab. Kointzidentzia bat dagoenean, hau da, konparazioa egiazkoa denean, dagokion **case** sententziaren ondoren dagoen kodea exekutatzen da. Adibide honetan, 10. eta 11. lerroetako kodea exekutatuko da, eta programa exekutatzearen emaitza **asteazkena** izango da.

Bestalde, eta dakizunez, programazio-lengoaiek kode-zatiak errepikatze sententziak eskaintzen dituzte. Normalean, errepikapen hori egingo da baldintza jakin bat egiazkoa denean. Adibidez, eta Python programazio-lengoiara itzulita, **while** sententziak aukera ematen du sententzia multzo bat exekutatze, baldin eta hura gobernatzen duen baldintza egiazkoa bada. Hurrengo kode-zerrendak kasu zehatz bat erakusten du.

```
i = 1;

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

Ikus dezakezun bezala, nahiko erraza da identifikatzea **while** begiztako kodea, hau da, 4. eta 5. lerroetako kodea, 5 aldiz exekutatuko dela. Horrek esan nahi du **while** begiztak 5 iterazio egingo dituela. Begizta gobernatzen duen baldintzak dakar iterazio bakoitzean galdetzea ea **i** aldagaiaren balioa 5 edo txikiagoa den. Konparazio hori egiten den lehen aldian, **i** aldagaiaren balioa 1 da. 1-en balioa 5 edo txikiagoa denez, baldintza egiazkoa dela ebaluatzen da, eta, beraz, begiztaren kodea (4. eta 5. lerroak) exekutatu egiten da. Kode horrek **i** aldagaiaren balioa erakusten du pantailako, eta, hauxe da garrantzitsua, **i** aldagaiaren edukari unitate bat gehitzen dio. Ondoren, programaren kontrol-fluxua 3. lerroa itzultzen da, eta

```
self.dt = dt

def kinetic_energy(self) -> np.ndarray:
    ekin = np.sum(np.sum(0.5 * self.m * self.vt**2, axis=1))
    assert ekin.shape == self.t.shape
    return ekin

def potential_energy(self) -> np.ndarray:
    raise NotImplementedError()

def energy(self) -> np.ndarray:
    return self.kinetic_energy() + self.potential_energy()

def force(self, t_index):
    raise NotImplementedError()

class ConstantGravityParticleSystem(ParticleSystem):
    def force(self, t, x, v):
        return np.array([0, 0, -self.g * self.m])

    def potential_energy(self, t, x, v):
        > * print(xb, sep, end, file) st[:, :2], as
        assert np.allclose(self.t, t)
        print(np.ans)
        return np.ans

class AerodynamicParticleSystem(ConstantGravityParticleSystem):
    def force(self, t, x, v):
        rho = 1.2
        cw = 0.45
        A = 100e-4
        fdiss = -rho*cw*A * np.abs(v)**3*v / 2
        fg = super(AerodynamicParticleSystem, self).force(t)
        return fg + fdiss

class NewtonPropagator:
    def __init__(self, system: ParticleSystem):
        self.system = system

    def run(self):
        print("running {} steps".format(len(self.system.t)))
        for index, t in enumerate(self.system.t[:-1]):
            self.step(index)

    def step(self):
        raise NotImplementedError()

class VelocityVerletPropagator(NewtonPropagator):
    def step(self, t_index):
```



baldintza berriz ebaluatzen du. Orain, *i* aldagaiaren balioa 2 da, eta 5ekin alderatzen da berriz. Prozesu hori errepikatu egingo da, harik eta *i* aldagaiak 6 balioa duen arte. Puntu horretan, **while** begiztak ez du iterazio gehiago exekutatu.

Azkenik, programazio-lengoaia gehienek begiztak malgutasun handiagoz idazteko aukera ematen duten sententziak dituzte. Klasikoenetako bat **for** sententzia da. Eskuarki, halako sententziek aukera ematen dute funtzionaltasun handiagoa sartzeko begizta deklaratzeko orduan; adibidez, aldagaiei balioa esleitzea eta, are, begiztaren exekuzioa gobernatzen duen aldagaia eguneratzea bera. Horrela, programatzaileak malgutasun handiagoa du kodea idazteko orduan. Python-en beste adibide bat aztertuko dugu.

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

Kode-zati horrek aurreko zatia emaitza bera sortzen du; orduan **while** sententzia erabili zen. Bestela esanda, 2. lerroko kodea 5 aldiz exekutatu da (begiztak 5 iterazio egingo ditu). Aurreko zatian ez bezala, *i* aldagaia 1. lerroko **for** begiztaren sententzian bertan adierazten da. Gainera, **range** sententziaren bidez, adierazten ari gara begiztak iteratu egingo duela *i* aldagaiaren balioa 1etik 6ra bitarteko tartean dagoen bitartean, hau da, *i* aldagaiaren balioa 1, 2, 3, 4 edo 5 den bitartean. Ikusiko duzun bezala, aurreko zerrendako funtzionalitate bera birsortzeko gauza izan gara 2 kode-lerro baino ez erabiliz; aurrekoak 4 kode-lerro zituen (lerro zuria kenduta).

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



Eduki digitalak
sortzea

A1 maila 3.4 Programazioa

Estilo- gidaliburuak





Estilo-gidaliburuak

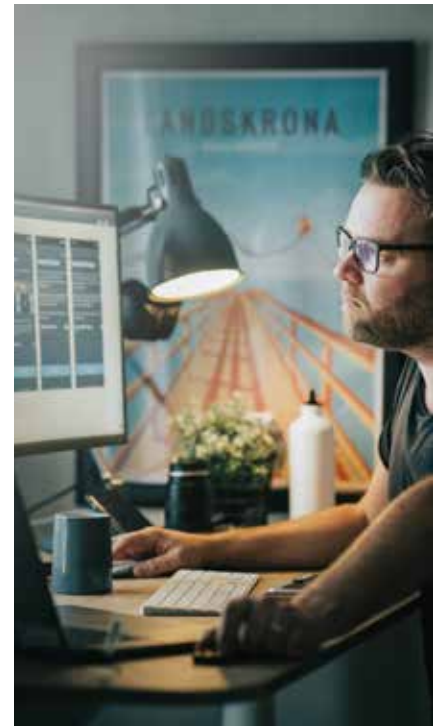
Estilo-gidaliburuak programazio-konbentzio batzuk definitzen dituzten dokumentuak dira; hau da, jardunbide egokiak, programazio-metodoak eta bestelako arauak definitzen dituzte programen garapen sendoa bermatze aldera. Adibidez, estilo-gidaliburuak programen zenbait alderdi dokumentatzen dituzte; besteak beste, gure aldagaiak nola izendatu behar ditugun, edo kodea koskatzeko zer erabili behar dugun, zuriuneak edo tabulazioak.

Estilo-gidaliburuak batez ere software-proiektuak taldean garatzeko erabili ohi dira, baina banakako proiektuetan ere badago oso frogatuta haien eraginkortasuna; programa baten kodea askoz ere errazago irakurri eta ulertuko dugu kodeak estilo-gidaliburu bati jarraitzen badio.

Estilo-gidaliburu ugari daude, programazio-lengoaia adina, baita bat baino gehiago ere lengoaia bakoitzeko. Programatzaile gisa, nahiago dugun estilo-gidaliburua hautatuko dugu kontuan hartuta zer euskarri eta dokumentazio dagoen eta erabiltzeko erraza den. Garrantzitsuena da gure programa garatzeko aukeratzeko dugun estilo-gidaliburuari jarraitzea eta proiektu berean estilo-gidaliburu bat baino gehiago ez nahastea.

Estilo-gidaliburuak ematen dizkiguten onuren artean, honako hauek nabarmentzen dira:

- Kodearen irakurgarritasuna hobetzen dute, errazago irakurtzeko eta ulertzeko.
- Beste garatzaile batzuek gure programei ekarpenak egiteko aukera errazten dute. Era berean, aukera ematen digute beste programa batzuen garapenean azkar inplikatzeko, hau da, dagoeneko ezagutzen ditugun estilo-gidaliburuak erabiltzen dituztenetan inplikatzeko.
- Benetan garrantzitsua den horretan zentratzen laguntzen digute, hau da, gure programak garatzeko lanean.





Estilo-gidaliburuaren adibide bat **PEP 8** edo **Python Koderako Estilo Gidaliburua** da, zeina Python lengoaiaren egileak berak, Guido van Rossum-ek, sortu zuen. Estilo-gidaliburu horrek honako hauei buruzko argibideak ematen ditu, besteak beste: kode- antolaketa, aldagaien izendapena nahiz lengoaiaren beste elementu batzuen, eta testu-kateetan komak edo komatxoak erabiltzea.

Adibidez, PEP 8k aldagaiak eta funtzioak letra xehez adieraztea gomendatzen du, eta hitzak gidoi baxuekin bereiztea. Hona hemen arau horrekin bat datozen aldagai-izen baliozkoen adibide batzuk: **gainditu_kopurua** eta **kontagailua**; adibide okerrak izango dira, ordea, **Gainditukopurua** eta **KONTAGAILUA**.

Estilo-gidaliburuarekin ez ezik, gure programen irakurgarritasuna hobetu eta etorkizunean irauin dezaten lortuko dugu, halaber, kodearen zatirik konplexuenak eta gure iritziz nabarmendu beharrekoak dokumentatuz. Kodeari iruzkinak erantsiz gero, zuzenean programaren kodean bertan egingo dugu dokumentazio hori, eta oharrak edozein programatzailek berrikusi ahal izango ditu garapen-etapan.

Testuinguru horretan, PEP 8 estilo-gidaliburuak jarraibideak ere ematen ditu kodean iruzkinak nola erabili jakin dezagun; esate baterako, irakurleari benetako ekarpena egiten dioten iruzkinak gehitzeko eskatzen du (eta jakina dena ez gehitzea, ordea), eta ingelesezko iruzkinak erabiliz dokumentatzea, besteak beste:

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

```
let a = m.split(" ")  
switch(a[0]){  
  case "connect":  
    if(a[1]){  
      if(clients.has(a[1])){  
        ws.send("connected");  
        ws.id = a[1];  
      }else{  
        ws.id = a[1]
```



DigitAll

Gaitasun
digitaletan
prestakuntza



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>