



Gaitasun  
digitaletan  
prestakuntza

# 3

## Eduki digitalak sortzea





Gaitasun  
digitaletan  
prestakuntza



Eduki digitalak  
sortzea

# ***C1 maila***





## Eduki digitalak sortzea

# AURKIBIDEA

### 3.1 EDUKIAK GARATZEA

- **AAko tekniketan oinarrিতuta testua automatikoki sortzea. GPT: kasu praktiko bat**
- **Diseinu grafikoko bideak sortzea eta editatzea**
- **3D diseinurako tresnak: libreak eta pribatiboak**
- **Hodeian adimen artifiziala erabiltzea bideo berriak sortzeko**
- **Zure webgunea eta eduki digitalak irisgarriak edo erabilerrazak al dira?**
- **Eduki-kudeatzaileek webguneak sortzeko duten funtzionaltasuna handitzea**

### 3.2 EDUKI DIGITALA INTEGRATZEA ETA BIRLANTZEA

- **Sentsore eta eragingailu motak**
- **Eduki digitalen benetakotasuna eta egiaztapena**

### 3.3 EGILE-ESKUBIDEAK ETA JABETZA INTELEKTUALEKO LIZENTZIAK

- **Copyrighta erregistratzea eta obra bat ustiatzea**

### 3.4 PROGRAMAZIOA

- **Programazio-paradigmak. Ikuspegi orokorra**
- **Garapen integratuko inguruneak (IDE). Ikuspegi orokorra**
- **Salbuespenak. Zer dira eta zertarako balio dute?**
- **Python-eko fitxategiak prozesatzea**
- **Objektuetara bideratutako programazioa Oinarrizko printzipioak eta kontzeptuak**
- **Kode-probak. Oinarrizko alderdiak**



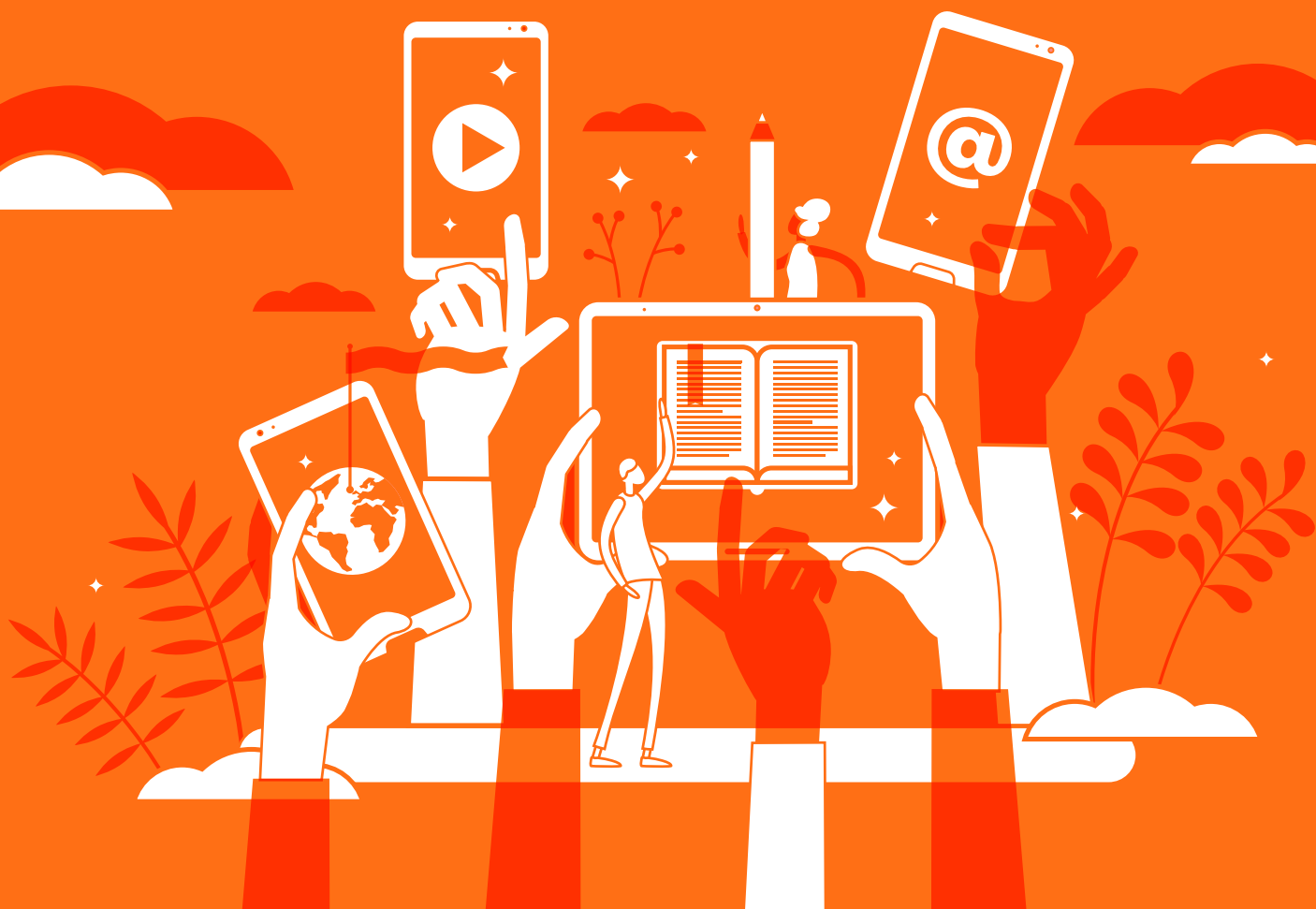


# DigitAll

Eduki digitalak  
sortzea

## 3.1

### EDUKIAK GARATZEA





Eduki digitalak  
sortzea

**C1 maila** 3.1 Edukiak  
garatzea

**AAko tekniketan  
oinarrituta testua  
automatikoki  
sortzea. GPT: kasu  
praktiko bat.**





# AAko teknikan oinarrituta testua automatikoki sortzea. GPT: kasu praktiko bat

## Sarrera

Testu-sorkuntza automatikoak aurrerapen nabarmenak izan ditu adimen artifizialeko tekniken (AA) garapenari esker. Teknika horiei esker, makinek aukera dute testu koherenteak eta testuinguru aldetik adierazgarriak ulertzeko eta sortzeko; kasu batzuetan, gizaki batek egingo lukeen antzeko moduan. Testu honetan, AA horiek garatzeko erabili diren teknika batzuk aztertuko ditugu, eta GPTrekin eginiko adibide batzuk jarriko ditugu, hura baita, segur aski, testua sortzeko gehien erabiltzen den AA.



## Testua automatikoki sortzeko AA ereduak

AA batek testua sortzeko hainbat bide daude. Logikoa denez, haietako bakoitzak tresna batzuk erabiltzea proposatzen du, eta sortutako testuak ezaugarri jakin batzuk izango ditu. Jarraian, haietako batzuk zerrendatuko ditugu, eta, ondoren, GPT ereduari buruz zehatzago hitz egingo dugu.

- **Arauetan oinarritutako ereduak:** ereduok aurrez definitutako algoritmoak eta arau espezifikoak erabiltzen dituzte testua sortzeko. Agertoki sinpleetan emaitza onargarriak sortzen badituzte ere, testuinguru konplexuagoetara egokitzeko gaitasun mugatua dute. ChatScript da arauetan oinarritutako teknologia bat; hasiera batean chatbotak sortzeko diseinatu zen.
- **Hizkuntza-eredu estatistikoak:** ereduok estatistika-teknikak erabiltzen dituzte datu masiboetan ereduak eta egiturak aztertzeko. Ikaste automatikoaren bidez, ereduok testua sortzen dute, hitzak eta sekuentziak gertatzeko probabilitatean oinarrituta. Spacy, hizkuntza naturalak prozesatzeko software-liburutegi bat, era horretako ereduaren barruan dago.
- **Neurona-sare errepikariak (RNN):** RNNak testuaren sekuentzia eta testuingurua kontuan hartzen dituzten AA ereduak dira. Neurona-sare horiek modu sekuentzialean prozesatzen dute informazioa, eta horrek aukera ematen



die epe luzeko mendekotasunak atzitzeko eta testu koherente eta arina sortzeko. Adibidez, TensorFlow-ek funtzionalitate ugari eskaintzen ditu RNN ereduak entrenatzeko eta hedatzeko, eta horrek testu sekuentziala eta testuinguru aldetik koherentea sortzea errazten du.

## Zer da GPT-4?

GPT-4 (Generative Pre-trained Transformer 4) testua automatikoki sortzeko alorrean agertu den azkenetako bat da. Eredu hau *Transformer* arkitekturan oinarritzen da. Arkitektura horrek eraginkortasuna frogatu du hizkuntza naturala prozesatzeko ataza ugarian.

GPT-4k **ikaste automatikoko** teknikak erabiltzen ditu hurrengo hitza edo esaldia iragartzeko, emandako testuingurua kontuan hartuta. Datu masiboetako entrenamendu intentsiboaren bidez, GPT-4k hizkuntzaren ezagutza sakona lortu du, eta testu koherente eta nabarmena sor dezake askotariko alorretan. RNNek ez bezala, Transformer-en oinarritutako ereduak erabiltzen duten arkitekturak aukera ematen du testuan epe luzeko erlazioak atzemateko, hitzen ordenaren mende bakarrik egon beharrik gabe.

Bi alderdi garrantzitsu argitu behar ditugu GPT-4 eraginkortasunez erabiliko badugu. Alde batetik, garrantzitsua da azpimarratzea sistemari zenbait jarraibide eman beharra dagoela. Jarraibideok hurrengo mailan aztertuko ditugu, eta emaitza optimizatzea ahalbidetuko digute. Bestalde, ez dugu ahaztu behar sistema horrek muga batzuk badituela ChatGPTtik hara jotzen badugu; nabarmenena da hura entrenatzeko erabilitako datu-baseak 2021. urtea baino lehenagokoak direla, eta urte horretatik aurrera, ondorioz, munduaren ezagutza mugatua duela. Hala ere, ordainpeko bertsioan Internetarako konektagarritasuna duten pluginak gehitzeko aukera dago, edo Bing-en txata ere erabil dezakegu, zeina Internetera konektatuta dagoen.

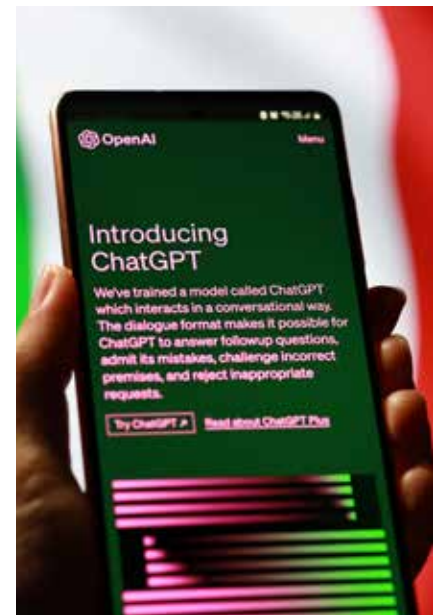




## Zer egin daiteke GPT-4rekin?

Tresna horrek hamaika aukera emango dizkigu, baita testu-sortzaile gisa soilik erabiltzen badugu ere. Errazena edozein gauza azaltzeko eskatzea da: gertaera historiko bat, gailu teknologiko batek nola funtzionatzen duen, errezeta bat nola prestatzen den, etab. Baina testu konplexuagoak ere lor ditzakegu, hala nola abestiak edo poemak.

Gaitasun honetan landu ditugun gaiak dagokienez, GPT-4 lagungarria izan dakiguke multimedia-eduki asko sor dezagun; adibidez, eskatu ahal diogu honako hau adierazteko: nola aplikatu geruza-maskara bat irudi batean; nola aplikatu efektuak testu baten gainean eta nola sortu webgune bat WordPress erabiliz.



### **i** Informazio gehiago

Coursera, UdeMy eta edX plataformek, besteak beste, adimen artifizialarekin, hizkuntza naturalaren prozesamendurekin eta testu-sorkuntzarekin lotutako ikastaro ugari eskaintzen dituzte. Hona hemen ikastaro arrakastatsuenetako batzuk: Stanforderko Unibertsitatearen «Natural Language Processing» (Coursera) eta «Deep Learning Specialization» (Coursera).





Eduki digitalak  
sortzea

**C1 maila** 3.1 Edukiak  
garatzea

# Diseinu grafikoko bideak sortzea eta editatzea





## Diseinu grafikoko bideak sortzea eta editatzea

Bideak sortzea eta editatzea tresna ahaltzu eta moldakorrenetako bat da diseinu grafikoaren esparruan. Lehen begiratuan konplexua dirudien arren, haren gaitasunak eta haiek aplikatzeko modua argi ulertuz gero, aukera-mundu bat irekiko dio diseinatzaile grafikoari.

### Zer da diseinu grafikoko bide bat eta zein da haren erabilgarritasuna?

Bideak (*paths*, ingelesez) funtsezko tresna bat dira diseinu grafiko digitalean. Marra bektorialak dira, eta manipulatu egiten dira forma, lerro eta kurba konplexuak sortzeko. Bideak oso moldakorrak dira; efektuak eta aldaketak aplikatzeko hautaketa-eremuak definitzeko nahiz formak marrazteko erabiltzen dira.

Bideak funtsezkoak dira; izan ere, zehazki definitzen dute irudi baten zer zati aukeratzen den; besteak beste, kolore-aldaketa bat eta desfokuratze-efektu bat aplikatzeko. Gainera, bideetan oinarritutako formek honako abantaila hau dute: guztiz editagarriak eta eskalagarriak dira, kalitatea galdu gabe, eta horregatik, balio handiko tresnak dira logotipoak, ikonoak eta tamaina-doikuntzak behar dituen beste edozein elementu grafiko diseinatzeko.

### Sortzea eta editatzea

GIMP softwarea erabiliko dugu bide baten sorrera eta hasierako edizioa urratsez urrats azaltzeko.

#### 1 | Ireki GIMP eta zure irudia

Ireki GIMP eta kargatu landu nahi duzun irudia. Horretarako, egin klik goiko menu-barrako «Fitxategia» aukeran, hautatu «Ireki» eta bilatu zer irudi erabili nahi duzun.





## 2 | Ireki bideen tresna

Egin klik GIMP-en tresna-paneleko bideen tresnan. Tresna hori Boligrafo/luma itxura duen ikono batekin irudikatzen da, eta bektoredun marra bat ateratzen da handik. Tresna-kutxa aurkitzen ez baduzu, aktibatu honela: joan menuko «Leihoak» atalera joanda; gero, hautatu «Tresna itxi berriak», eta, azkenik, «Tresnak».

## 3 | Sortu zure bidea

Egin klik zure bidea hastea nahi duzun lekuan. Horrek lehen nodoa edo aingura sortuko du. Gero, egin klik zure lerroa edo kurba amaitzea nahi duzun lekuan. Horrek bigarren nodo bat sortuko du. Lerro zuzen bat sortu nahi baduzu, utzi horrela. Baina kurba bat nahi baduzu, egin klik saguaren ezkerreko botoiarekin eta arrastatu bigarren nodoa. Nodotik kolore horiko bi lerro agertu direla ikusiko duzu, eta haiek arrastatuz kurba doitu duzu.

## 4 | Sortu forma konplexuagoak

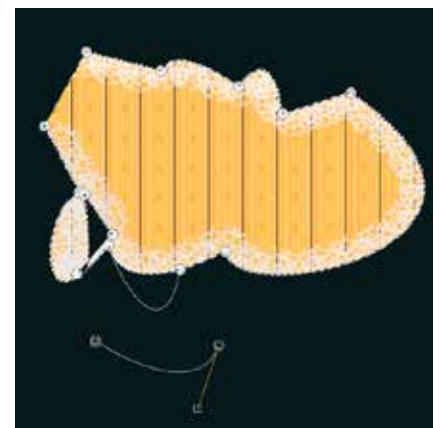
Forma konplexuagoak sortzeko, egin klik nodo gehiago gehitzeko. Bukatu ondoren, egin klik lehen nodoan bidea ixteko.

## 5 | Editatu zure bidea

Bidea edozein unetan doitu dezakezu, klik eginez eta nodoak arrastatuz. Aukera duzu, halaber, nodoen arteko lerroetan klikatzeko eta haiek arrastatzeko: kurbak doitzeko, mugitzeko, kolore solidoz edo patroiaz betetzeko eta trazu bat jartzeko edo ez.

## 6 | Aplikatu zure bidea

Zure bidearekin pozik zaudenean, zenbait gauzetarako erabil dezakezu; adibidez, hautaketa bat egiteko. Horretarako, joan bideen panelera (ikus ezin baduzu, hemen aurkituko duzu: «Leihoak» > «Tresna itxi berriak» > «Bideak»), egin klik eskuineko botoiarekin sortu berri duzun bidean, eta hautatu «Bidetik hautapenera».



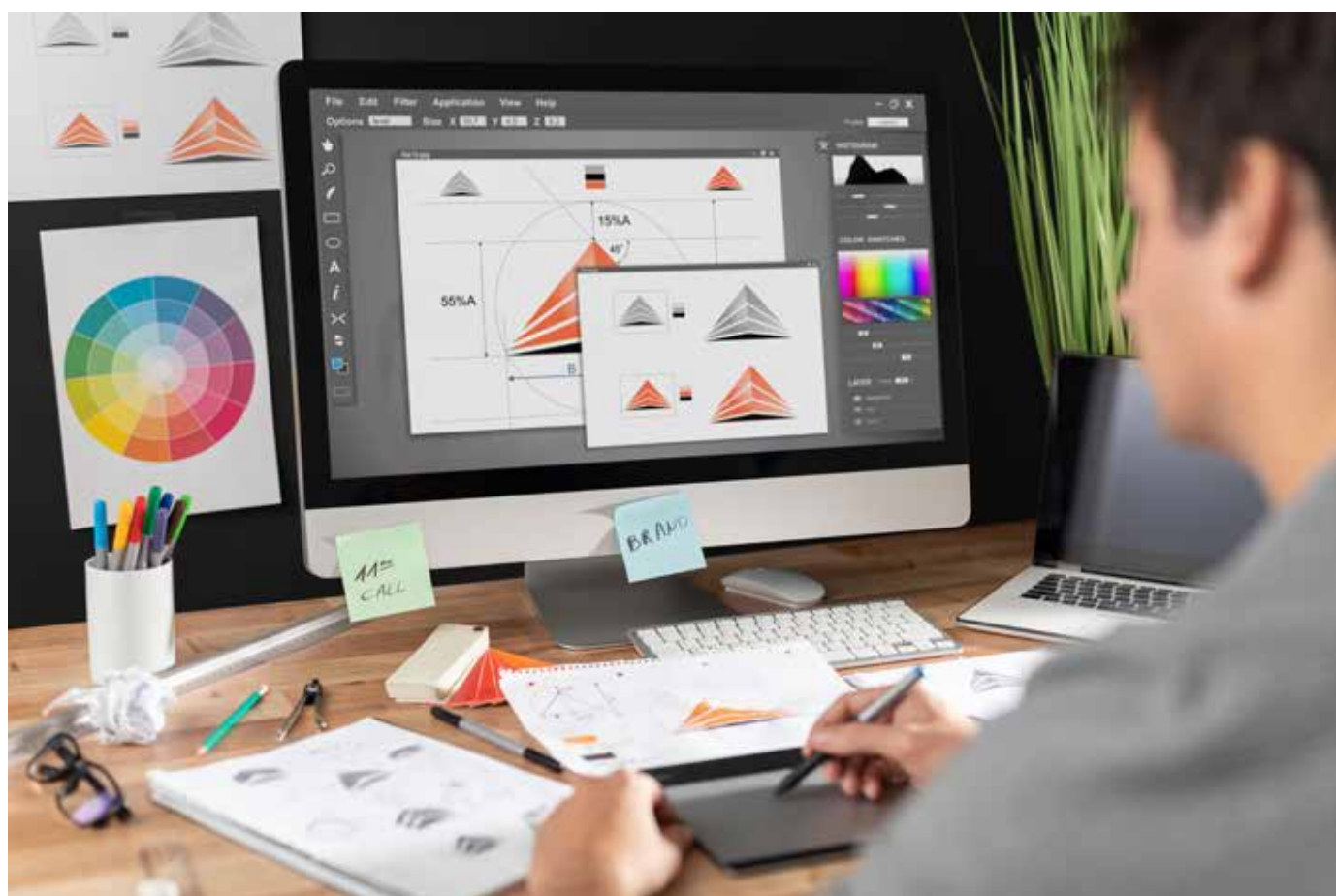


Begiratu orokorra eman berri diogu diseinu grafikoko bideak sortu eta editatzeari. Goragoko mailetan, sakonago aztertuko dugu tresna horren erabilera nola maximizatu, adibide praktikoak eta tekniko zehatzagoak emanez.

### **i** Informazio gehiago

Bideen tresnak ikasteko kurba motela du, baina hura behin ikasita, oso erabilgarria da. Hasieran, egokia izango da bide sinpleak eta oinarritzko marrazketa-praktikekin lan egitea, harik eta bideen jokabidearekin ohitzen garen arte. Pixkanaka, esperimentatu forma konplexuagoekin eta hautaketa zehatzagoekin.

[e.digitall.org.es/ruta-gimp](http://e.digitall.org.es/ruta-gimp)





Eduki digitalak  
sortzea

**C1 maila** 3.1 Edukiak  
garatzea

# 3D diseinurako tresnak: libreak eta pribatiboak

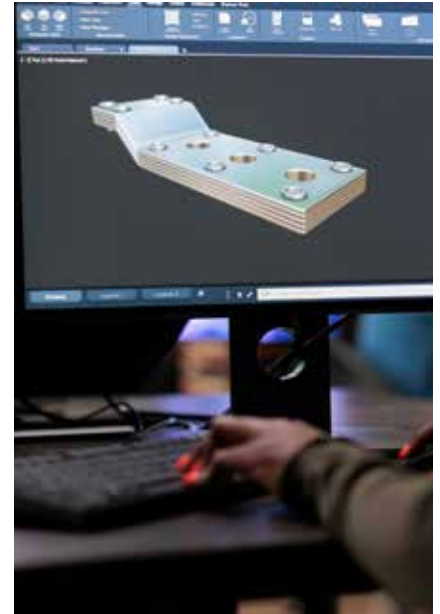




## 3D diseinurako tresnak: libreak eta pribatiboak

### Sarrera

3D diseinuak errotik aldatu du objektu birtualak kontzeptualizatzeko eta sortzeko modua. Modelatzeko, animatzeko eta renderizatzeko gaitasun aurreratuei esker, 3D diseinua funtsezko tresna bihurtu da askotariko diziplinetan (besteak beste, arkitektura, industria-diseinua, medikuntza, ingeniari-tza eta arte digitala). Testu honetan, 3D diseinuaren garrantzia aztertuko dugu, bai eta gure ideiak mundu birtualean gauzatzeko zer tresna dauden ere (libreak eta pribatiboak).



### 3D diseinurako tresnak

#### CAD softwarea

Jarraian ikusiko dugun bezala, aipatuko ditugun tresnetako askok CAD letrak dituzte izenean. CAD softwareaz ari garenean (*Computer-Aided Design*, ingelesez; ordenagailuz lagundutako diseinua, gaztelaniaz), honetaz ari gara: formatu digitaleko diseinuak sortzeko, aldatzeko eta optimizatzeko tresnak ematen dituen software espezializatuaren kategoriaz.

Hala, CAD softwareak aukera ematen die diseinatzaileei objektuen edo sistemen 2D edo 3Dko eredu birtualak sortzeko. Horrela, tresna eta funtzio espezifikoak erabiliz, erabiltzaileek aukera dute geometria zehatza marrazteko, murrizketak eta dimentsioak aplikatzeko, simulazioak egiteko eta diseinuen propietate fisikoak aztertzeko.

CAD softwareak, jarraian aipatuko ditugun programek bezala, zenbait abantaila dituzte ohiko diseinu-metodoen aldean. Alde batetik, ingurune digital batean lan egitean, diseinatzaileek aldaketa azkar eta zehatzak egin ditzakete diseinuetan, askotariko aukerak azter ditzakete eta haien bideragarritasuna ebaluatu dezakete fisikoki egin aurretik. Bestalde, teknologia horren bidez eskuzko marrazketa prozesu automatizatu batez ordeztzen denez, programa horiekin egindako proiektuak diseinu klasikoak baino denbora laburragoan egiten dira.



## Tresna libreak

Software libreak, beste hainbat eremutan bezala, demokratizatu egin du 3D diseinurako sarbidea, doakoa eta kode irekikoa baita. Segur aski, Blender da 3Dko modelatze eta animazioko software libre erabiliena. Izan ere, profesional askoren gogokoena da, oso ahaltsua delako.

Hasieran konplexua ematen badu ere, Blender nahiko erraza da erabiltzen, eta interfaze intuitiboa du. Oso erabilia da zenbait abantaila dituelako: alde batetik, plataforma anitzeko softwarea da, eta hari esker sorkuntza-prozesu osoa kontrolatuko dugu (modelatzea, animazioa, simulazioa, renderizatzea, etab.). Bestalde, erabiltzaileen komunitate zabalak etengabe eguneratzen du, baliabideak gehitzen ditu eta hura erabiltzen ikasteko lineako tutorialak egiten ditu.

Hala ere, Blender tresnaz gain, 3D diseinua egiteko bestelako tresna libreak daude. Hona hemen erabilienetako batzuk:

- **FreeCAD:** plataforma anitzeko kode irekiko 3D modelatzaile parametrikoko bat da, bizitza errealeko era guztietako tamainetako objektuak diseinatzeko egina.
- **OpenSCAD:** kode irekiko CAD programa bat da, eta 3D ereduak sortzen ditu script edo gidoiak oinarri hartuta. Erabiltzen duen programazio-lengoaiarekin ohitu ondoren, gidoi horiek oinarri hartuta, 3D eredu konplexuak sortu ahal izango ditugu.
- **3D Builder:** Microsoftek garatutako aplikazio hau egokia da esperientziarik gabeko erabiltzaileentzat, eta 3D objektuak ikustea, sortzea eta pertsonalizatzea ahalbidetzen du.
- **Tinkercad:** 3D diseinuak sortzeko doako online tresna da. Erabilerraza da, eta ez da beharrezkoa 3D diseinuan aurretiazko esperientziarik izatea. Izan ere, hezkuntzan erabiltzen da, haren oinarria baita forma sinpleak konbinatzea eredu konplexuak sortzeko, LEGO bat izango balitz bezala.

### OHARRA

Netflix-en *Next Gen* animazio-filma (2018) oso-osorik sortu zuten Blender programa erabiliz.





## Tresna pribatiboak

Tresna pribatiboek tresna eta ezaugarri aurreratu gehiago eskaintzen dituzte programa libreek baino, baina, batez ere, erabiltzailearen arazoak konpontzeko laguntza tekniko espezializatua. Gainera, oro har, interfaze atseginagoa izaten dute.

Jarraian, 3D diseinu-programa pribatibo ezagunenetako batzuk deskribatuko ditugu:

- **Autodesk 3ds Max:** software hau asko erabiltzen da 3D diseinuaren eta bistaratzearen industrian, haren jatorrizko bertsioak 90eko hamarkadakoak baitira. Tresna sorta zabala eskaintzen du modelatzeko, animatzeko, simulatzeko eta renderizatzeko. Oso erabilia da entretenimenduaren industrian, eta film askotan erabili dute (*Blade: Trinity*, *Misiao: Ezinezkoa* eta *Hellboy*).
- **Autodesk Maya:** Autodesk-ek garatutako beste programa bat da, eta asko erabiltzen da zinemaren eta animazioaren industrian. Ezaguna da 3D pertsonaia errealistak eta efektu berezi harrigarriak sortzeko gaitasunagatik. Izan ere, film oso ezagunetan erabili dute, hala nola *Avatar* filmean eta *The Matrix* sagaren sekuela batzuetan.
- **Cinema 4D:** 3D diseinuko software bat da, Maxon-ek garatutakoa. Bai profesionalak, bai hasiberriek erabiltzen dute, interfaze intuitibo eta erabilerraza duelako. Cinema 4D zinemaren, telebistaren eta bitarteko digitalen industrian erabiltzen da efektu bisualak, animazioak eta mugimenduzko grafikoak sortzeko. Haren potentzialaren bi adibide dira *Pacific Rim* eta *Tron: Legacy* filmak.
- **Houdini:** software honen ikuspegi prozesala nabarmentzen da, 3D ekoizpenaren arlo guztiak barne hartzen baititu. Film oso ezagunetako planoen erantzule da (*Frozen* eta *Zootopia*, besteak beste).

3D diseinu-programa pribatibo erabilieneen adibide batzuk baino ez dira izan. Bakoitzak bere ezaugarriak ditu, eta behar espezifikoaren arabera erabiltzen da.







## 3D inprimaketa: Objektu birtualak gauzatzea

3D inprimaketak are gehiago aldatu du 3D diseinua, sortutako objektu birtualak fisikoki gauzatzea ahalbidetu baitu. Hau da, 3Dko diseinu-programak erabil ditzakegu 3Dn inprima daitezkeen eredu digitalak sortzeko. Fabrikazio gehigarriko teknologiak erabiltzean, 3D inprimagailuek diseinuak azkar eta zehatz bihurtzeko objektu ukigarri. Horrek are gehiago zabaldu ditu 3D diseinua askotariko eremuetan aplikatzeko aukerak (besteak beste, prototipoak sortzea, medikuntza birsortzailea, arkitektura jasangarria eta produktuen pertsonalizazioa).

### Ondorioa

3D diseinua funtsezko tresna bihurtu da, ideiak eraginkortasunez eta zehatz bistaratzeko gaitasuna duelako. Bai tresna libreak, bai pribatiboek zeregin garrantzitsua dute prozesu horretan, eta dauden beharretara nahiz baliabideetara egokitutako aukerak eskaintzen dituzte. 3D inprimaketa integratuta, 3D diseinuak urrats bat gehiago eman du, objektu birtualak gauzatzea ahalbidetu baitu, eta horrek askotariko aukerak dakartza eremu ugarran.

#### Informazio gehiago

3D diseinuaren eta diseinatutako materialak 3Dn inprimatzearen arteko erlazio handiaren adibide bat dugu Ayúdame 3Dko lanetan. Ayúdame3D Espainiako erakundeak teknologiaren gizarte-balioa sustatzen du kontzientziario teknologiko-sozialeko programen bidez, mundu osoko kolektibo kalteberei laguntzeko. Horri esker, 3Dn inprimatutako besoak – tresdesi izenekoak – sortzen ditu, eta doan ematen dizkie desgaitasuna duten pertsonari.

[ayudame3d.org](http://ayudame3d.org)



Eduki digitalak  
sortzea

**C1 maila** 3.1 Edukiak  
garatzea

# Hodeian adimen artifiziala erabiltzea bideo berriak sortzeko





## Hodeian adimen artifiziala erabiltzea bideo berriak sortzeko

Adimen artifiziala (AA) erabiliz bideoak sortzea etengabeko eboluzioan ari den alorra da, eta teknika eta tresna berriak sortzen ari dira une oro. AAren laguntza baliabide digitalak sortzeko osagai integrala bihurtzen ari da; izan ere, bideoak sortzeko eta editatzeko modu horrek aukera ematen du denbora aurrezteko, efizientzia handitzeko eta sormenezko aukera berriak agertzeko.



### AAren ohiko erabilerak bideoak sortzean

Gaur egun, erabilera hauek dituzte, besteak beste:

#### 1 | Edizio-prozesuaren automatizazioa

AAn oinarrituta bideoa editatzeko tresnek erabiltzaileei aukera ematen diete bideoak testu-dokumentu bat editatuko luketen bezala editatzeko. Bideoaren edukiaren transkripzio automatikoak sortzen dituzte, eta horregatik, erabiltzaileek erraz ezaba ditzakete nahi ez dituzten zatiak, eta horretarako aski da dagokion testua transkripziotik kentzea.

#### 2 | Bideo-kalitatea hobetzea

Tresna horiei esker, askotariko zereginak egiten dira, hala nola audioa ezabatzea, mugimenduaren jarraipena egitea, hondoa ezabatzea eta isiluneak detektatzea. Funtzio horiei esker, erabiltzaileek edukiaren kalitatea hobetzen dute, trebetasun tekniko aurreratuak izan beharrik gabe.

#### 3 | Eduki sortzailea sortzea

Horien bidez, eduki berriak sor daitezke testuetatik edo irudietatik abiatuta, erabiltzaileak emandako jarraibideak oinarri hartuta, eta horrek eduki bisual bakarrak sortzeko aukera berriak ematen ditu.



## 4 | Pertsonalizazioa eta brandinga

AAko bideo-tresnek, gainera, enpresei laguntzen diete bideoak pertsonalizatzen, nor bere marka islatzeko eta abatar digitalak erabiliz bideoak sortzeko. Hori bereziki erabilgarria izango da azalpen-bideoak, gaikuntza-bideoak eta iragarkiak azkar sortzeko, aktoreak edo kamerariak kontratatu beharrik gabe.

### Bideoa sortzeko tresnak

Jarraian, doan edo ordainduta erabil ditzakezun tresna erabilgarrien adibide batzuk aurkeztuko ditugu.

#### 1 | Descript

Aukera ematen dizu zuk diozun guztiaren transkripzio bat, zenbait eszenarekin batera, sortzeko, bideo-pista automatikoki bereizita. Ezabatu nahi duzun transkripzioaren zatiak nabarmendu ditzakezu, eta Descript-ek editatu egingo ditu. Gainera, bideoa eszenatan zatitu dezakezu, eta kalitate handiko artxibo-materiala gehitu zure proiektuari, editorearen leihotik irten gabe.



#### 2 | Wondershare Filmora

Bideo tradizionala editatzeko tresna honek AAn oinarritutako utilitateak ditu. AAn oinarritutako funtzionalitateen artean honako hauek daude: murrizketa adimenduna, audio-zarata ezabatzea, audio-luzaketa, mugimenduaren jarraipena, hondoa ezabatzea eta isiltasuna antzematea. Esportatzean zure bideoaren konfigurazioa ere alda dezakezu, eta aukeratu formatua, bereizmena, kalitatea eta fotograma-abiadura. Gainera, ChatGPT du erantsia, eta ondorioz, AA erabiliz gidoiak, azpitoluak, deskribapenak eta abar ere sor daitezke.



#### 3 | Runway

Runwayk AAre ezaugarriak eskaintzen ditu bideoak fintzeko eta testutik bideoa nahiz testutik irudia sortzeko, eta aukera ematen dizu, orobat, zure AA ereduak irudiak sor ditzaten entrenatzeko ere bai. Bestelako ezaugarriak ere eskaintzen ditu, hala nola pantailaren atzeko plano berdea aldatzea (kroma), objektuak ezabatu eta ordezteak eta irudi-sekzioak ordezteak.





## 4 | Peech

Eduki-marketin taldeentzat diseinatuta dago. Aukera ematen dizu zure marka-kita gehitzeko, Peech-ek zure bideo guztiak automatikoki markatu ditzan. Bideo berri bat gehitzen duzun bakoitzean, Peech-ek elementu horiek gehitzen ditu, azpitu pertsonalizagarriekin batera.



## 5 | Synthesia

Tresna honen bidez, bideoak sor ditzakezu Aaren abatar digitalak erabiliz. Askotariko hizkuntzak onartzen ditu, baina baliteke abatarrak guztiz sinesgarriak ez izatea pantaila handietan ikusten direnean.



### AA BIDEZKO BIDEO-SORTZAILEEN EZAUGARRIAK

Izena	Egokia honetarako	Plataforma	Doakoa
<b>Descript</b>	Gidoia editatuz bideoa editatzeko	Windows, Mac (zenbait funtziotarako webgunea)	Bai, ordubeteko transkripzioa eta ur-marka
<b>Wondershare Filmora</b>	AAko tresnak erabiliz bideoa fintzeko	Windows, Mac, iOS, Android	Bai, ur-marka
<b>Runway</b>	AA sortzailearekin esperimentatzeko	Webgunea	Bai, 125 krediturekin, 3 proiektu
<b>Peech</b>	Eduki-marketin taldeentzat	Webgunea	Bai, hilean 2 bideo
<b>Synthesia</b>	Abatar digitalak erabiltzeko	Webgunea	Ez

### **i** Informazio gehiago

Online ikastaroak daude (Coursera, machine learning A-Z, etab.), eta AARI, haren tresnei, teknologiari eta joerei buruzko ezagutza zehatzak ematen dituzte. Hala ere, zer tresna espezifiko ikasi nahi duzun, baliteke tresna horretarako tutorialak edo ikastaro espezifikoak bilatu behar izatea. Esteka honetan dituzu dokumentu hau idazteko unean zeuden aukera onenak:

[unite.ai/best-ai-video-generators](https://unite.ai/best-ai-video-generators)



Eduki digitalak  
sortzea

**C1 maila** 3.1 Edukiak  
garatzea

Zure webgunea  
eta eduki digitalak  
irisgarriak edo  
erabilerrazak al  
dira?





# Zure webgunea eta eduki digitalak irigarriak edo erabilerrazak al dira?

## Zure webgunea eta edukiak irigarriak diren ebaluatzeko protokoloa

Aurreko moduluetan azaldutako irizpideak ezagutu ondoren, jarraitu beharreko protokolo edo prozedura bat proposatuko dugu, webgune batek irigarritasun-baldintza nagusiak betetzen dituen egiaztatzeko (testuari, irudiari, bideoari eta soinuari dagokienez).

### 1 | Testua egiaztatzea:

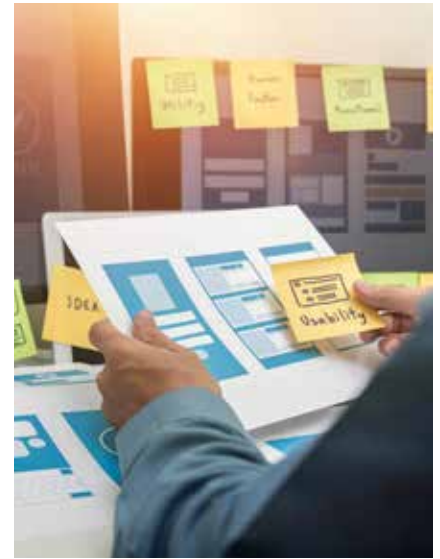
- Egiaztatu letra-tipoaren tamaina irakurgarria dela eta aukera dagoela erabiltzailearen lehentasunen arabera doitzeko.
- Egiaztatu informazioa transmititzeko ez direla erabiltzen kolore asko, pastel-koloreak eta oso kolore deigarriak erabiltzen; izan ere, baliteke horrek ulermena zailtea ikusmen-desgaitasuna eta daltonismoa duten pertsonentzat.
- Aztertu goiburuen egitura egokia dela (H1, H2, etab.), nabigatzea eta edukia ulertzea errazteko.
- Ziurtatu kontraste handia dagoela testu-kolorearen eta atzeko planoko kolorearen artean.
- Aukeratu erraz irakurtzeko moduko tipografia sinplea, apaingarri askoko bat aukeratzeak balitekeelako irakurketa zailtea.

### 2 | Irudiak egiaztatzea:

- Ziurtatu irudi garrantzitsu guztiek ordezeko testu deskribatzailea dutela (ALT atributua). Horri esker, ikusmen-desgaitasuna duten pertsonak irudien edukia ulertuko dute pantaila-irakurgailuen edo bestelako laguntza-teknologiaren bidez.
- Egiaztatu irudiak baino erabiltzen ez dituzten estekek eta botoiek ordezeko testua edo titulu-etiketa argigarriak dituztela.

### 3 | Bideoak egiaztatzea:

- Ziurtatu bideoek entzumen-desgaitasuna duten pertsonentzako azpitoluak dituztela. Azpitoluek zehatzak eta sinkronizatuak izan behar dute, eta behar bezala adierazi behar dituzte elkarrizketa eta soinu-efektu nabarmenak.





- Egiaztatu audio deskribatzailea duen bideo-bertsio bat ematen zaiela ikusmen-desgaitasuna duten pertsoneri. Audio deskribatzailea narrazio gehigarri bat da, eta bideoan ikusten diren ekintza garrantzitsuak deskribatzen ditu.
- Egiaztatu bideoa erreproduzitzeko kontrolak irisgarriak direla eta teklatura erabiliz erraz erabiltzeko modukoak direla.
- Aztertu ea gainjar daitekeen zeinu-hizkuntza darabilen pertsona edo abatar baten bideoa.

#### **4 | Soinua egiaztatzea:**

- Ziurtatu audio-elementuek erreprodukzio-kontrolak dituztela, soinuaren bolumena pausatzeko, eteteko eta doitzeko.
- Egiaztatu audio-edukiaren testuzko alternatiba bat edo transkripzio bat ematen dela, entzumen-desgaitasuna duten pertsonen edukia eskura dezaten.

#### **5 | Laguntza-teknologiarekiko bateragarritasun-probak:**

- Egin probak laguntza-teknologia erabiliz, hala nola pantaila-irakurgailuak (adibidez, JAWS, NVDA eta VoiceOver) eta teklatuak saguen ordeztu. Horri esker ebaluatu ahal izango duzu webguneko edukia nola aurkezten den eta harekin nola interakzionatzen duten ikusmen- eta mugikortasun-desgaitasuna duten pertsonen.

#### **6 | Nabigazioa eta egitura egiaztatzea:**

- Ziurtatu webguneko nabigazioa argia, koherentea eta ulerterraza dela. Erabili nabigazio-elementuetarako etiketa egokiak, eta eman adierazle bisual argiak erabiltzaileak jakin dezan webguneko zer ataletan dagoen.
- Egiaztatu esteka deskribatzaileak erabili direla, estekaren helmuga argi adierazita, eta ez esteka generikoak (adibidez, «klikatu hemen»).
- Egiaztatu tabulazio-ordena logikoa dela eta webgunean nabigatu daitekeela teklatura bakarrik erabiliz, sagua erabili beharrik gabe.
- Ziurtatu formularioak irisgarriak edo erabilerrazak direla, etiketa argiak dituztela, eta lotura egokiak dituztela sarrera-eremuen eta dagozkien etiketen artean.







## 7 | Kolore-kontrastea egiaztatzea:

- Ziurtatu testuaren eta hondoaren arteko kolore-kontrastea nahikoa dela irakurgarritasuna errazteko. Egiaztatu web-erabilerraztasunaren jarraibideetan ezarritako kontraste-irizpideak betetzen direla.
- Egiaztatu elementu interaktiboek, hala nola botoiek eta estekek, kontraste egokia dutela errazago identifikatu eta erabiltzeko.

## 8 | Askotariko nabigatzaile eta gailuekiko bateragarritasun-probak:

- Egin itzazu erabilerraztasun-probak web-nabigatzaile ezagun batzuetan (Chrome, Firefox, Safari eta Edge), webgunea plataforma guztietan irisgarria dela ziurtatzeko.
- Egiaztatu irisgarria dela askotariko gailuetan, hala nola mahaigaineko konputagailuetan, tabletetan eta gailu mugikorretan, pantaila guztietan esperientzia irisgarria bermatzeko.



## 9 | Kodea baliozkotzea:

- Erabili HTML eta CSS balidatzeko tresnak, webgunearen kodeak gomendatutako estandarrak eta jardunbide onenak betetzen dituela ziurtatzeko. Kode garbi eta ondo egituratua funtsezkoa da irisgarritasunerako.

## 10 | Errendimendu-probak:

- Egin webgunearen errendimendu-probak, karga azkarra eta esperientzia arina bermatzeko. Karga-atzerapenek baliteke eragin negatiboa izatea desgaitasuna duten pertsonengan, batez ere laguntza-teknologiak erabiltzen dituztenengan.

## 11 | Hizkuntza- eta irakurgarritasun-akatsak ebaluatzea:

- Egiaztatu gramatika-, ortografia- edo puntuazio-akatsak ez dietela edukia ulertzea zailtzen desgaitasun kognitiboak eta irakurmenaren arlokoak dituztenei.
- Ziurtatu esaldien eta lerrokaden egitura argia eta zehatza dela, irakurtzea eta ulertzea errazteko.

## 12 | Interaktibitate- eta funtzionaltasun-probak:

- Egiaztatu elementu interaktiboak, hala nola formularioak, botoiak eta menu zabalgarriak, irisgarriak direla eta teklatura bakarrik erabiliz erraz erabiltzeko modukoak direla.



- Ziurtatu animazioak eta edukiaren eguneratze dinamikoak irigarriak direla, eta erabiltzaileei ez dietela distrakziorik edo nahasmendurik eragiten.

### 13 | Hizkuntza-etiketa egiaztatzea:

- Ziurtatu webguneak behar bezala erabiltzen duela HTML hizkuntza-etiketa. Hori bereziki garrantzitsua da laguntza-teknologiak erabiltzen dituztenentzat, edukiaren ahoskera eta hizkuntza egokitzeko aukera ematen baitie.

### 14 | Tabulazio bidezko nabigazioa ebaluatzea:

- Egiaztatu elementu interaktiboen tabulazio-ordena logikoa eta iragartzeko modukoa dela. Horrek bermatzen du teklatura nabigatzeko erabiltzen duten pertsonak orriko elementu guztiak behar bezala atzitu eta erabili ahal izatea.

### 15 | Kontraste-probak zoom-maila desberdinetan:

- Egin kolorearen kontraste-probak nabigatzailearen zoom-maila desberdinetan. Ziurtatu kontrasteak irisgarritasun-estandarrak betetzen dituela, baita erabiltzaileak orriaren zoom-maila handitzen edo murrizten duenean ere.

### 16 | Edukiaren irakurgarritasuna egiaztatzea pantaila-tamaina desberdinetan:

- Ebaluatu edukia askotariko pantaila-tamainatan irakur daitekeen, eta ziurtatu gailu mugikorretan eta pantaila txikiagoetan testua behar bezain handia eta irakurgarria dela.

### 17 | Teklatu-probak:

- Egiaztatu webguneko elementu eta ekintza interaktibo guztiak teklatuaren bidez atzitu eta erabili daitezkeela. Hori funtsezkoa da sagua edo antzeko beste gailu batzuk erabili ezin dituzten pertsonentzat.

### 18 | Testua birdimentsionatzeko probak:

- Ziurtatu webgunearen edukia % 200era arte birdimentsionatu daitekeela, informazioa, funtzionaltasuna eta diseinua galdu gabe. Hori bereziki garrantzitsua da ikusmen-desgaitasuna duten pertsonentzat, baldin eta, argi irakurtzeko, testuaren tamaina handitu behar badute.





## 19 | Formularioak eta errore-mezuak baliozkotzea:

- Egiaztatu webguneko inprimakiek etiketa argiak, errore-mezu deskribatzaileak eta balidazio egokia dituztela, interakzioa eta erabiltzaileak akatsak zuzentzea errazteko.

## 20 | Aurkikuntzak eta hobekuntzak dokumentatzea:

- Dokumentatu irisgarritasuna egiaztatzekeo prozesuan zehar identifikatutako aurkikuntza, akats eta hobekuntza guztiak. Erregistratu ezarritako aldaketak, eta egin ezazu webgunean, irisgarritasuna hobetzeko, egindako eguneratzeen jarraipena.

Web-erabilerraztasuna prozesu jarraitua eta dinamikoa da. Eduki ekintza-plan bat identifikatutako irisgarritasun-oztopoei heltzeko, eta egin aldizkako eguneratzeak webgunearen irisgarritasuna hobetzeko, argitaratzen diren jarraibide eta estandar berrietan oinarrituta.

## Beste webgune batzuk automatikoki ebaluatzen dituzten webguneak

Webgune batek beharrezko diren irisgarritasun-irizpideak betetzen dituen ebaluatzeko, webgune batzuek beste webgune batzuen analisi automatikoak egiten dituzte URLaren bidez. Hona hemen zenbait adibide:

- **WebAIM's WAVE** (*Web Accessibility Evaluation Tool*): WAVE doako online tresna bat da, web-irisgarritasunaren analisi automatizatua egiteko aukera ematen du. Aski da webgune baten URLa sartzea, eta WAVEk txosten zehatza sortuko du, antzemandako irisgarritasun-arazoak nabarmentzeko.
- **Axe by Deque Systems**: Axe Deque Systems-ek garatutako irisgarritasun-tresnen suite bat da. Web-nabigatzaileentzako luzapen bat du, eta haren bitartez webguneak aztertzekeo aukera dago irisgarritasun- edo erabilerraztasun-arazoak aurkitzeko. Garapenerako lan-fluxuetan integratzeko API bat ere eskaintzen da.





- **Lighthouse by Google:** Lighthouse Googlek garatutako kode irekiko tresna bat da; web-orrien kalitaterako auditoria automatikoak eskaintzen ditu. Auditoria orokorrez gain, Lighthousek web-erabilerraztasunerako berariazko zenbait auditoria hartzen ditu barne.
- **Tenon.io:** Tenon.io web-irigarritasunaren analisirako plataforma bat da, eta webguneak aztertzea ahalbidetzen du, irigarritasun-arazoak aurkitzeko. Garapeneko lan-fluxuekin integratzeko API bat eskaintzen du, eta online erabiltzaile-interfaze bat ere eskaintzen du, analisi azkarrak egiteko.

 **AXE BY DEQUE SYSTEMS**

[deque.com/axe](https://deque.com/axe)

 **LIGHTHOUSE BY GOOGLE**

[e.digitall.org.es/lighthouse](https://e.digitall.org.es/lighthouse)

 **TENON.IO**

[tenon.io](https://tenon.io)

### Informazio gehiago

Irigarritasunari buruzko nazioarteko estandarrei buruz gehiago jakin nahi baduzu, honako hau kontsultatu: W3C Web Accessibility Initiative (WAI).

[e.digitall.org.es/wai](https://e.digitall.org.es/wai)



Eduki digitalak  
sortzea

**C1 maila** 3.1 Edukiak  
garatzea

**Eduki-  
kudeatzaileek  
webguneak  
sortzeko duten  
funtzionaltasuna  
handitzea**





# Eduki-kudeatzaileek webguneak sortzeko duten funtzionaltasuna handitzea

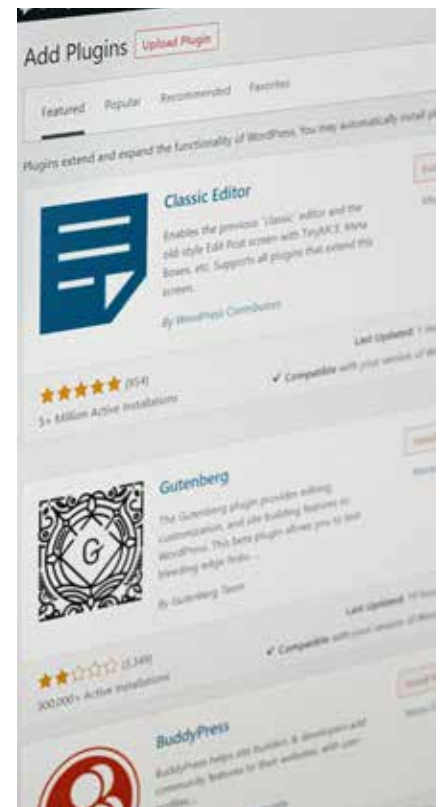
## Sarrera

Eduki-kudeatzaileek funtsezko zeregina betetzen dute webguneak sortzeko orduan, eta erabiltzaileei aukera ematen diete edukia eraginkortasunez administratzeko eta argitaratzeko. Hala ere, baliteke kudeatzaile horien oinarriko funtzionaltasuna nahikoa ez izatea webgune jakin baten behar guztiak asetzeko. Hor esku hartzen dute osagarriek edo pluginek; tresna hauek eduki-kudeatzaileetan instala daitezke, haien konplexutasuna eta gaitasuna handitzeko. Edo, bestela esanda, pluginen bidez, webgunea gure behar partikularretara egokitu ahal izango dugu. Testu honetan, osagarri horien garrantzia aztertuko dugu, eta erabilera-adibideak ikusiko ditugu eduki-kudeatzaile ezagunenetako bat erabiliz: WordPress.

## Pluginen bitartez funtzionaltasuna handitzea

Osagarriak edo pluginak aplikazio txikiak dira, eta eduki-kudeatzaileekin integratzen dira, funtzionalitate eta ezaugarri berriak gehitzeko. Tresna horiek modu egokia ematen dute eduki-kudeatzaile baten gaitasunak erabiltzailearen **beharren** arabera **personalizatzeko eta handitzeko**. Osagarriek askotariko helburuak dituzte, harremanetarako formularioak besterik gabe gehitzetik merkataritza elektronikoko sistema konplexuak sortzeraino.

Jarraian, bi plugin mota orokor deskribatuko ditugu: ordainpekoak eta doakoak. Oro har, ordainpeko pluginek funtzionalitate aurreratuagoak eta espezializatuagoak ematen dituzte. Adituek eta enpresek garatu ohi dituzte soluzio horiek, eta denbora nahiz baliabideak ematen dituzte haiek garatzeko. Ordainpeko osagarriek laguntza tekniko sendoagoa, aldizkako eguneratzeak eta ezaugarri eskusiboak ematen dituzte. Bestalde, doako osagarrien aukera zabala ere badago eduki-kudeatzaileentzat. Borondatezko ekarpena egiten duten programatzaileen eta diseinatzaileen komunitateak garatzen ditu osagarri horiek. Nahiz eta doako osagarriek ordainpekoek adina funtzio ematen ez dituzten, aukera bikaina dira aurrekontu mugatua dutenentzat eta oinarriko funtzionalitateak behar dituztenentzat.





## Pluginak WordPress-en erabiltzea

WordPress-en unibertsoan, imajina ditzakegun funtzionalitate guztietarako pluginak daude. Aurreko mailetan WordPress-en ezaugarriak deskribatzean esan dugun bezala, pluginek pertsonalizatzea ahalbidetuko digute programazio-ezagutzak izan beharrik gabe.

WordPress-en pluginak instalatzea prozesu erraza da, eta bi metodoren bidez egin daiteke: WordPress-en biltegiaren bidez edo eskuzko instalazio baten bidez. Testu honetan deskribatuko dugu pluginak nola instalatu WordPress-en gordailuaren bidez.

WordPress-en plugin bat instalatzeko, gure administrazio-panelean «Pluginak» fitxa hautatuko dugu alboko menuan (1. irudia). Plugin-kategorietan nabigatu dezakegu, gako-hitzen araberako bilaketak egin ditzakegu eta ezagunenen edo gomendatuenen zerrendak esploratu ditzakegu. Plugin bakoitzak bere xehetasun-orria du biltegian, eta, bertan askotariko informazioa ematen da: funtzionalitatea, ezaugarriak, eskakizunak eta erabiltzaileen iruzkinak.

Instalatu beharreko plugina aurkitu ondoren, aski dugu «Instalatu orain» fitxa sakatzea, eta WordPress-ek automatikoki deskargatu eta instalatuko du plugina gure webgunean. 2. irudian, adibide gisa, Europako araudirako DBEO cookie-en aholkularia izeneko plugina ikusiko dugu, zeinak aukera emango baitigu Datuak Babesteko Erregelamendu Orokorrera egokitzeko.





1. irudia. WordPress-en administrazio-paneleko «Pluginak» menuaren ikuspegi orokorra.



2. irudia. Europako araudirako DBEO cookie-en aholkularia izeneko pluginaren xehetasuna. Gure webgunean instalatuko dugu beheko eta eskuinaldeko botoia sakatuz gero, eta lagunduko digu «Onartu cookieak» botoi ezaguna sortzen.





1. irudian ikusten dugun bezala, aukeretako bat kategoria ezagunen arabera bilatzea da. Hona hemen batzuk:

- **Bilaketa-tresnetarako optimizazioa (SEO):** bilaketa-tresnetan webguneen ikusgarritasuna eta posizionamendua optimizatzeko.
- **Merkataritza elektronikoa eta enpresak:** online dendak sortzeko eta ordainketak kudeatzeko.
- **Segurtasuna:** webgunea eraso eta mehatxuetatik babesteko.
- **Formularioak:** harremanetarako formulario pertsonalizatuak eta inkestak sortzeko.
- **Irudi-galeriak:** irudiak webgunean modu erakargarrian erakusteko.

### WordPress-eko *plugin* ezagunenak

WordPress-erako doako plugin ezagunenetako batzuk hauek dira: Yoast SEO, Contact Form 7, Elementor Website Builder, WooCommerce eta Akismet Anti-Spam. Bestalde, WordPress-eko ordainpeko plugin ezagunenetako batzuk hauek dira: Yoast-en ordainpeko bertsioak eta WooCommerce-n oinarritutako merkataritza elektronikoa kudeatzeko pluginak.





## Ondorioa

Eduki-kudeatzaileak tresna ahaltzuak dira webguneak sortzeko, baina batzuetan haien oinarriko funtzionaltasuna ez da nahikoa behar guztiak asetzeko. Pluginek esku hartzen dute hor, eta erabiltzaileei aukera ematen diete eduki-kudeatzaileen gaitasunak modu erraz eta pertsonalizatuan areagotzeko. Hala ordainpeko nola doako osagarriak aukeratzeko, askotariko aukerak dituzte eskura erabiltzaileek WordPress-en unibertsoan webguneak pertsonalizatzeko eta esperientzia hobetzeko, bai beraiantzat bai bisitariantzat. Pluginekin esploratzea eta esperimentatzea eduki-kudeatzaileen funtzionaltasuna handitzeko eta webguneak hurrengo mailara eramateko modu eraginkorra da.

### Informazio gehiago

Asko ikasiko duzu plugin espezifikoaren funtzionamenduari buruz WordPress-en gaztelaniazko plugin-direktorioan. Webgune honetan WordPress-erako doako plugin ugari eta kode irekiko batzuk daude. Plugin bakoitzaren ezaugarriak, beste erabiltzaile batzuen balorazioak, instalatutako WordPress-en egungo bertsioarekiko bateragarritasuna eta instalazio aktiboen kopurua ikusiko dituzu. WordPress-en gordailua WordPress.org-eko lantaldeak mantentzen eta administratzen du, horregatik, gordailuan jasotako plugin guztiak WordPress-eko lantaldeak berrikusten eta onartzen ditu argitaratu aurretik, eta horrek kalitatea eta segurtasuna bermatzen laguntzen du.

[es.wordpress.org/plugins](https://es.wordpress.org/plugins)

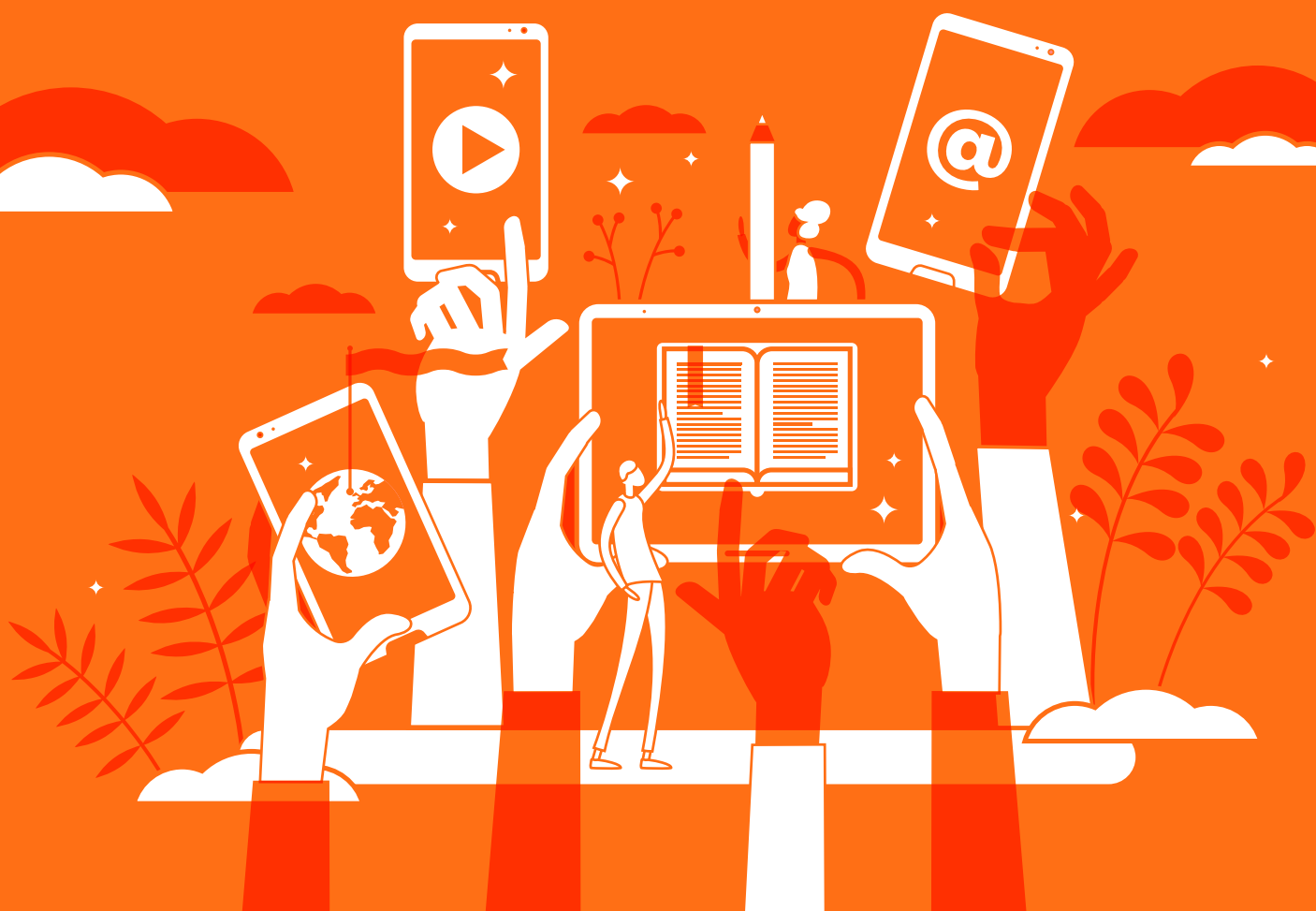


# DigitAll

Eduki digitalak  
sortzea

## 3.2

### EDUKI DIGITALA INTEGRATZEA ETA BIRLANTZEA





Eduki digitalak  
sortzea

**C1 maila** 3.1 Edukiak  
garatzea

# Sentsore eta eragingailu motak





## Sentsore eta eragingailu motak

### Sentsoreak

Software/hardware sistema bat eraikitzeko, mikrokontrolagailuak erabili behar dira, hala nola Arduino eta Raspberry Pi ordenagailua. Eraiki beharreko proiektu elektronikoaren arabera aukeratuko dugu mikrokontrolagailu mota. Arduinoren abantaila da ataza berehala exekutatzeko duela. Raspberry Pi, zenbaketa-potentzia handiagoa eta sistema eragilea duenez, egokiagoa da mikroordenagailu funtzional eramangarri gisa erabiltzeko. Bi plaka mota horiek aukera ematen dute **sentsore** eta **eragingailu** izeneko beste gailu batzuk sisteman integratzeko. 1. irudiak gailu elektroniko horien adibide bat erakusten du.



Sentsoreak magnitude fisikoak edo kimikoak (**instrumentazio-aldagaiak**) neurtzeko eta **transduttore** baten bidez **aldagai elektriko** bihurtzeko gai diren gailuak dira.

Instrumentazio-aldagaia aldagai elektriko bihurtzeko, hauek erabiltzen dira: erresistentzia termiko bat (adibidez, tenperatura), fotodiodo bat (argi-intentsitatea) edo kristal piezoelektriko bat (presioa). Azken horrek polaritatea hartzen du presio mekanikoa jasaten duenean. Aldagai elektrikoak hauek izango dira: erresistentzia, kapazitatea, tentsioa, korronea eta abar. Sentsorearen irteerak analogikoak, digitalak edo komunikazio-busen bidez konektatuak izango dira.

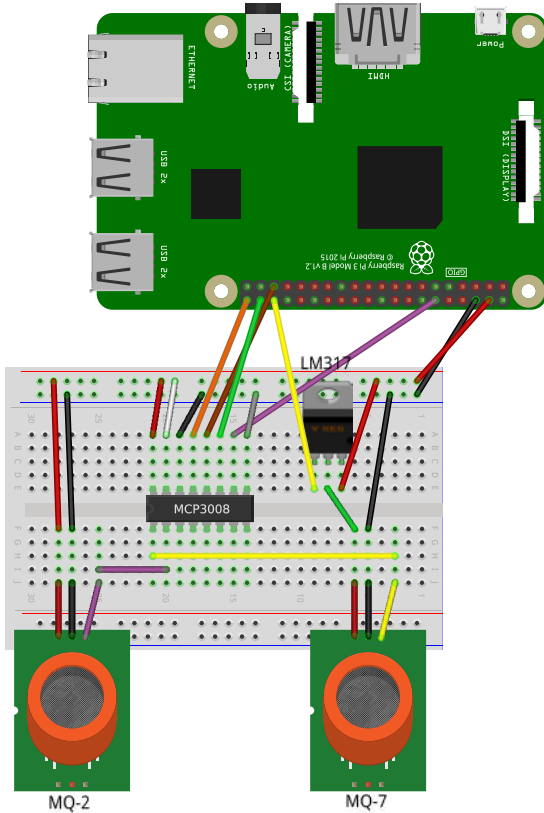
Sentsoreek neurketak egiten dituzte, eta prozesu horrek jarraian deskribatzen diren parametro batzuk ditu ezaugarri. **Domeinua** gailuak neur ditzakeen balio posibleen multzoa da. **Bereizmena** zera da: sentsoreak detekta dezakeen instrumentazio-aldagaiaren gutxieneko aldaketa. **Zehaztasuna** egin dezakeen errore handiena da. **Jitoak**: beste magnitude batzuk dira, eta hauek ere, nahi denak ez ezik, emaitzazko neurrietan eragiten dute. Zeroaren errorea (*offset*): aldagai elektrikoaren balioa, instrumentazio-aldagaia nulua denean. Hona hemen sentsore mota nagusiak:



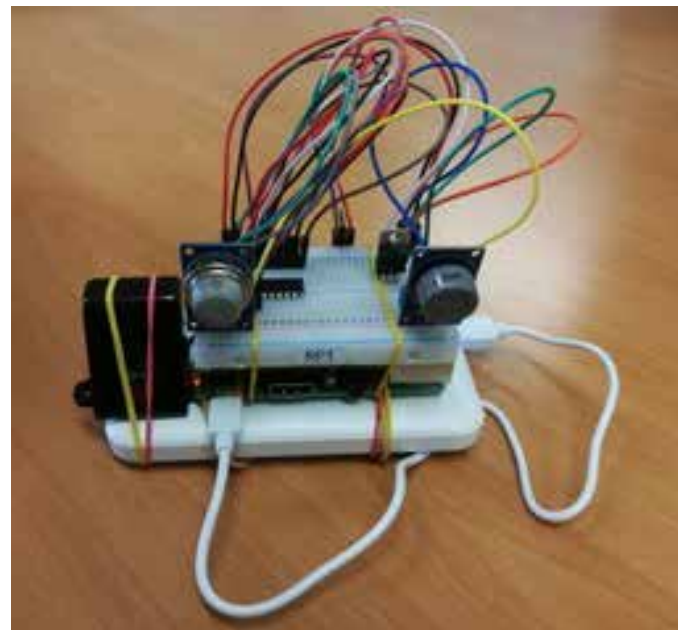
Mota	Deskribapena	Ezaugarriak
<b>Desplazamendua eta distantzia</b>	Desplazamendu linealak, posizioak edo distantzia linealak neurtzen dituzte. Gailu horien tartea milimetrotik ehunka metrora artekoa da.	Batzuek kontaktu fisikoa dute (marruskadurarekin), kontakturik gabekoak ere badira, laserrean edo ultrasoinuetan oinarritutakoak.
<b>Azelerazioa</b>	Azelerazioa edo bibrazioa neurtzen dute g gutxi batzuetatik milaka g-etaraino.	Mota askotakoak daude: piezorresistiboak, piezoelektrikoak eta azelerometro kapazitiboak.
<b>Giroskopioak eta erreferentzia inertzialak</b>	Espazioan orientazioa neurtzea edo mantentzea ahalbidetzen dute objektu baten mugimendua gertatzean.  Erreferentzia inertzialak posizioa eta abiadura angeluarra neurtzeko sistemak dira.  Sentsore horiei sentsore girokopiko esaten zaie.	Giroskopioak eta azelerometro triaxialak barne hartzen dituzte.
<b>Pare- eta tortsio-sentsoreak</b>	Ardatz bati jasanarazten zaion tortsio-indarra neurtzen dute. Errotazio-elementuak aztertzeko erabiltzen dira.	Estatikoak edo dinamikoak dira. Transduktoreak tortsioa tentsio-aldakuntza bihurtzen du.
<b>Presioa eta emaria</b>	Azalera-unitateko indar bat tentsio baliokide bihurtzen dute. Likidoen presioa neurtzeko nabarmentzen dira (besteak beste, ura, olio eta balazta-likidoak). Emari-sentsoreek kanal batean denbora-unitateko dabilen material-kantitatea (pisua edo bolumena) neurtzen dute.	Aplikazioak dituzte presio-etengailuen eraikuntzan edo mailen monitorizazioan.
<b>Ingurumen-parametroak</b>	Honako hauek neurtzeko aukera ematen dute: temperatura, hezetasuna, presio barometrikoa, CO2, NOx, PM10, PM2.5 eta abar.	Aplikazioak dituzte nekazaritzan ureztatzearen kontrol adimenduna egiteko eta hirietako emisioak kontrolatzeko.
<b>Presentzia-sentsoreak</b>	Objektu baten presentzia detektatzen dute, bai eta erreferentzia-puntuarekiko hurbiltasuna ere. Tripulatu gabeko ibilgailuetan eta zaintza autonomoan aplikatzen dira.	Igorle/hartzaile pare bat dute. Aplikazioetan erabiltzen dira ateak automatikoki irekitzeko/ixteko, argiak pizteko/itzaltzeko, pantailak automatikoki pizteko.
<b>Sentsore akustikoak</b>	Aplikazioak ditu ahots-komandoen detekzioan eta zaintza adimendunean.	Uhin akustikoak bulkada elektriko bihurtzen dituzte.
<b>Argi-sentsorea</b>	Argiarekiko sentikorra den azalera dute, eta erresistentzia-balio desberdinak sortzen dituzte.	Argi-maila neurtzeko erabiltzen dira, eta eguna edo gaua den hautemateko, eta argiztapena zuzentzeko.



1. irudiko adibidean Raspberry Pi gailu bat dago, zirkuitu elektriko txiki batera konektatuta; bi ingurumen-sentsore hartzen ditu barne, GLP gasak eta karbono monoxidoa neurtzeko, hurrenez hurren. 2. irudian emaitza errealak erakusten da.



1. irudia. Adibide honetan Raspberry Pi gailu bat dugu, ingurumen-parametroen bi sentsorera konektatuta.



2. irudia. 1. irudian erakusten den eskemaren benetako emaitza.

## Eragingailuak

**Eragingailuak** izeneko gailuek energia hidraulikoa, pneumatikoa edo elektrikoa eraldatzen dute sistematik kanpoko elementu baten gaineko ekintza sortzen duen prozesu bat aktibatzean.

Adibidez, eragingailu batek mikrokontrolagailu baten (adibidez, Arduino eta Raspberry Pi) agindua jasoko du, eta ponpa baten motorra martxan jarri.


Taula honetan eragingailu mota nagusiak laburbiltzen dira:



Mota	Deskribapena	Erabileraren adibideak
<b>Mekanikoak</b>	Sarrera baten ekintza birakaria mugimendu lineal bihurtzen dute.	Katu mekanikoak. Objektuak zehazki eta behin eta berriz jartzeko erabiltzen dira.
<b>Pneumatikoak</b>	Aire konprimatua ekintza mekaniko bat eragiten duen indar eragile bihurtzen dute.	Pintza pneumatikoak, objektuak leku batetik jaso eta beste leku batean uzteko.
<b>Hidraulikoak</b>	Indar hidraulikoa erabiltzen dute bultzatzeko eta kanpoko indarra lortzeko. Potentzia handia behar denean erabiltzen dira.	Garabietan eta hondeamakinetan erabiltzen dira, baita balbulak automatizatzeko ere.
<b>Elektrikoak</b>	Indarraren iturria energia elektrikoa da.	Buelta anitzeko balbulak, motorrak eta erreleak.
<b>Termikoak</b>	Mugimenduaren bidez lortzen dute indarra, eta anplifikazioa, hedapen termikoaren bidez.	Berokuntza- eta hozte-sistemen automatizazioa.

Sentsoreak Arduinoko eta Raspberri Pi-ko plaken sarreretara konektatuta daude; eragingailuak, berriz, irteeretara konektatzen dira. Gainera, **periferikoak** konekta daitezke; hardware gailuak dira, eta aukera ematen dute informazioa biltegitartzeko eta sistemaren kanpoaldearekin trukatzeko. Periferikoen adibideak: LCD pantailak, teklatuak, kanpoko memoriak, inprimagailuak, kamerak, mikrofonoak, zenbakizko displayak, etab.

**Programazioaren paradigmak. Ikuspegi orokorra** (C1 mailako 3.4 konpetentzia) izeneko dokumentua funtsezkoa da sentsoreak eta eragingailuak askotariko eremuetan kontrolatzeko, hala nola robotikan eta automatizazio industrialean. Programazio-lengoaiak erabiliz programak sor daitezke sentsoreekin elkarreraginean jardun dezaten, eta horrela, inguruneari buruzko informazioa biltzeko eta eragingailuetara bidaltzeko, ekintza espezifikoak egin ditzaten. Gainera, sentsoreek hartutako informazioa multimedia-edukia sortzeko erabil daiteke, hala nola irudiak eta bideoak, bildutako datuen adierazpen bisuala emateko. Hori erabilgarria da ingurumen-monitorizazioko eta segurtasuneko aplikazioetan.



**PROGRAMAZIO-  
PARADIGMAK.  
IKUSPEGI OROKORRA**

Erreferentziako dokumentua:  
**A3C34C1D01**

**⚠ ADI**

Sentsore bat aukeratzekoan, ezaugarriak arretaz aztertu behar ditugu, eta kontuan hartu behar dugu gure plakarekin bateragarria dela (tentsioa/potentzial-diferentzia eta korrontearen intentsitatea/ampere kopurua) eta haren domeinua gure sistemaren magnitudeekin bat datorrela. Aukeraketan eragina dute, halaber, erabilerraztasunak eta liburutegiak nahiz dokumentazioa egoteak.





Eduki digitalak  
sortzea

**C1 maila** 3.2 Eduki digitala integratzea  
eta birlantzea

# Eduki digitalen benetakotasuna eta egiaztapena





## Eduki digitalen benetakotasuna eta egiaztapena

Gaur egun, **informazioaren gizartea** deritzon bizi gara. Bertan, pertsonak informazioa eta ezagutza sortu, aldatu, eskuratu eta parteka ditzakete, produkzio-prozesuak eta, azken batean, bizi-kalitatea hobetzeko.

**Informazioaren gizartean** Informazioaren eta Komunikazioaren Teknologiek (IKT) aukera ematen dute jarduera ekonomiko, sozial eta kulturaletan informazioa eskuratzeko, sortzeko eta banatzeko.

Paradoxikoki, informazioaren gizartearen ondorioetako bat **desinformazioa** da. Informazioa eskuratzek ez dakar ezinbestean informatuta egotea, batez ere informazioa **eskergea, heterogeneoa** (edozein erabiltzailek sor dezake edukia) eta, maiz, kontraesankorra bada. Ildo horretan, informazioaren gizarteak herritarrei erronka hau planteatzen die: **zentzu kritikoz** filtratzea eta bereiztea informazio erabilgarria nahiz egiazkoa eta halakoa ez den informazioa.

**Auzitegi-analisi informatikoa** konputazioaren zientzien arlo bat da, eta barne hartzen ditu baliabide digitalak (hardwarea eta softwarea) **identifikatu, zaindu, berreskuratu eta aztertzeko** aukera ematen duten teknikak.

Auzitegi-analisi informatikorako tresna ugari daude informazio digitalari buruzko erabakiak hartzeko eta gertaerak eta ebidentziak aurkezteko.

Hona hemen haietako batzuk: **FotoForensics** ([fotoforensics.com](http://fotoforensics.com)), zeinak ahalbidetzen baitu URL batetik edo gailu lokal batetik datorren irudi baten gainean askotariko analisiak egitea, aldatu duten edo ez egiaztatzeko; **AmpedAuthenticate** ([ampedsoftware.com](http://ampedsoftware.com)) tresnak, berriz, irudi bat erreferentzia gisa hartuta, aukera ematen du askotariko analisiak egiteko eta irudi digital baten prozesatze-historia ezagutarazteko, eduki manipulatu den zehazteko. Azkenik, **ForensicToolkit (FTK)** ([exterro.com/forensic-toolkit](http://exterro.com/forensic-toolkit)) tresnak soluzio integratua ematen du, gailu digital batetik informazioa berreskuratzeko eta haren edukiaren auzitegi-analisia egiteko.





Desinformazioa eragiten duen edukiaren adibide argi bat **fake news** direlakoak dira. Sarean argitaratutako **albiste faltsuak** dira, eta helburutzat dute irakurleak engainatu, manipulatu eta nahasaraztea, bai eta, horretaz gain, pertsona edo erakunde bat gorestea edo haren izen ona galaraztea. Albiste horiek egiazkotasun-itxura izaten dute komunikabide digitaletan, batez ere sare sozialetan.

Duela gutxi, ikus-entzunezko eduki faltsuak agertzen ari dira sare sozialetan eta komunikabide digital askotan. Eduki horietako batzuek umore-asmoa dute, baina beste batzuek *fake news*en antzeko helburuak dituzte. **Deepfake** izenez ezagutzen diren eduki horietan honelako bideoak nabarmentzen dira: eduki faltsuak sortzeko asmoz pertsonaia errealean aurpegi eta ahotsak integratzen dituzten bideoak.

*Deepfake*, edo **faltsutze sakona**, deritzonak barne hartzen ditu adimen artifizialeko tekniken bidez sortutako eduki digitalak, hala nola gezurrezko irudiak eta bideoak.

*Deepfake*ak sortzeko, adimen artifizialeko teknika bat erabiltzen da, hain zuzen **ikaste sakoneko** teknika bat, sare neuronaletan oinarritua. Teknika horri **Sare Sortzaile Antagonikoa (SSA)** esaten zaio. Bertan, bi sare daude. Sare batek oinarri hartzen du lehendik dauden irudien eta bideoen datu-base handi bat, ikasteko eta berriak sortzeko. Bigarren sareak bereizten du lehenengoak sortutako irudiak eta bideoak faltsuak diren edo ez. Bigarren sareak ez badu hautematen lehenengo sareak sortutako eduki bat faltsua dela, orduan *deepfake*a prest dago.

Horregatik guztiagatik, funtsezkoa da eduki digital baten **benetakotasuna** bermatzea eta **manipulatu** duten edo ez **egiaztatzea**, desinformazioa eta bestelako delituak (iruzur, ohore eta irudiarekin lotutakoak) eragozteko. Batzuetan, irudietan eta bideoetan dauden akatsen bitartez hautemango dugu edukia manipulatu dutela. Detekzio formalago eta zorrotzagoa egiteko, baina, aurretik ezagutu ditugun auzitegi-analisiko tresnez gain, *deepfake*en aurkako teknologia batzuk ere saiatzen dira haiek detektatzen, edukia autentifikatzeko eta eduki bat *deepfake* gehiago sortzeko erabiltzea eragozteko.



Hala ere, askoz baliabide gehiago bideratzen dituzte *deepfake*ak sortzeko teknologiarra, haiek hautemateko tresnetara baino. Eduki horien garatzaileek, gainera, haiek hautemateko moduari buruz argitaratutako ikerketak baliatzen dituzte beren teknologia hobetzeko eta sistema horiei ihes egiten dien eduki faltsua sortzen jarraitzeko. Horregatik, *deepfake*ak detektatzeko software eskuragarria ez dago erabiltzaileentzat irekita, oro har.

Hauek dira publiko orokorrak eskura duen *deepfake*en aurkako software erabiliak:

- **Sensity:** eduki faltsua detektatzeko web-plataforma bat da. **Askotariko formatuetako fitxategiak igotzeko** aukera ematen du, dokumentuak barne, iruzurrezko edukia den jakiteko. Eskura dituen tresnen artean, honako hauek nabarmentzen dira: **nortasuna egiaztatzea, *deepfake*ak detektatzea, aurpegi-ezagutza eta iruzurrezko dokumentuak ezagutzea**. Tresnak lortzeko, konpainiarekin **harremanetan jartzeko formulario** bat bete behar da, sarbidea eskatzeko.
- **Deepware Scanner: *deepfake*ak detektatzeko** kode irekiko proiektua da. **Komando-lerroko interfaze** batean oinarrituta exekutatzeko da, eta gailuaren direktorio bateko bideoak eskaneatzeko aukera ematen du, *deepfake* bat zein diren detektatzeko. Sensity-k ez bezala (SSAn oinarritutako sistema baitu), Deepware Scanner-ek **sare neuronal konboluzional**etan oinarritutako ereduak erabiltzen ditu. Kode irekiko proiektua denez, jakin daiteke sistema hori diseinatuta nola dagoen, eta kontsultatu ahal da *deepfake*ak detektatzeko erabiltzen diren aurre-entrenatutako ereduak iturburu-kodea.
- **Fake Profile Detector:** Google Chromeren doako luzapen bat da, eta **irudi** bat *deepfake*a den identifikatzeko aukera ematen du. Oso erraz instala daiteke, eta, erabiltzeko, aski da saguaren eskuineko botoiarekin klik egitea aztertu nahi dugun irudian, eta ondoren, «**check fake profile picture**» aukeran. Une horretan, jakinarazpen bat agertuko da goiko eskuineko ertzean, balitekeela irudi faltsua izatea ohartarazteko.

#### OHARRA

Irudi eta bideoetan *deepfake*ak detektatzeko honelakorik dagoen egiazta dezakegu begi hutsez: argiztapeneko hutsaltasunak eta irudian edo frame batean itzaldurarik dagoen; eremu batzuetan bereizmena aldatzen den, eta ageriko akatsak edo hutsak dauden.



SENSITY

[sensity.ai/deepfakes-detection](https://sensity.ai/deepfakes-detection)



DEEPWARE SCANNER

[github.com/deepware](https://github.com/deepware)



Laburbilduz, atal honetan gomendio praktiko batzuk daude, begi-ikuskapenaren bidez eduki digitala autentifikatzeko, bai eta auzitegi-analisi informatikoko tresna batzuk eta *deepfake*en aurkako bestelako teknologiak ere, erabiltzailek eduki digitalen originaltasuna autentifika eta egiazta dezaten.



**FAKE PROFILE  
DETECTOR**

[e.digitall.org.es/fake-profile](https://e.digitall.org.es/fake-profile)

**i Informazio gehiago**

Auzitegi-analisiko beste tresna bat **TinEye** ([tinEye.com](https://tinEye.com)) da.

**i Informazio gehiago**

Deepfakeak detektatzeko softwarea: [e.digitall.org.es/deepfake](https://e.digitall.org.es/deepfake)



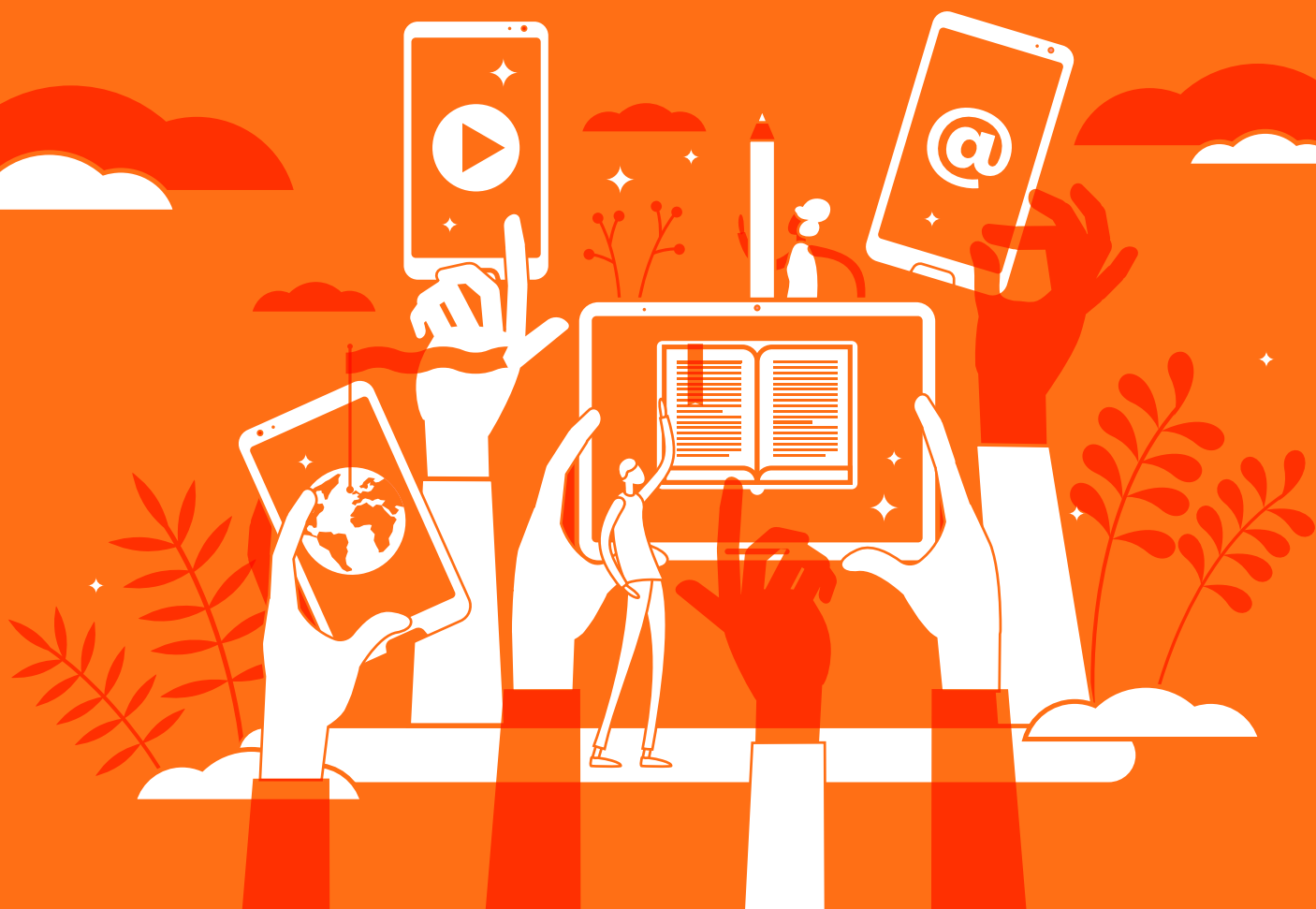


# DigitAll

Eduki digitalak  
sortzea

## 3.3

### EGILE-ESKUBIDEAK ETA JABETZA INTELEKTUALEKO LIZENTZIAK





Eduki digitalak  
sortzea

**C1 maila** 3.3 Egile-eskubideak eta jabetza  
intelektualeko lizentziak

# Copyrighta erregistratzea eta obra bat ustiatzea





## Copyrighta erregistratzea eta obra bat ustiatzea

Dokumentu honen edukiak jabetza intelektuala telematikoki erregistratzeko mekanismoari buruzko ezagutzak ematen ditu; hain zuzen, Gaztela-Mantxako Erkidegoko kasu jakin baterako, baina mekanismo bera izango da beste edozein erkidegotan, alde batera utzita Andaluzia eta Madrilgo Erkidegoa, sistema propioak baitituzte.

Erregistroa telematikoki eskatzeko, Kultura eta Kirol Ministerioaren egoitza elektronikoaren bidez ([cultura.sede.gob.es/procedimientos](http://cultura.sede.gob.es/procedimientos)), prozeduren zerrendara jo behar dugu eta «Propiedad Intelectual» kategoria hautatu; haren barruan, «Registro de la Propiedad Intelectual» azpikategoriara joan behar dugu, eta bertan, «**Solicitud de primera inscripción en el Registro de la Propiedad Intelectual**» hautatu ([e.digitall.org.es/primera-inscripcion](http://e.digitall.org.es/primera-inscripcion)).

Ondoren, «LEHEN INSKRIPZIO»ko lan artistiko, zientifiko eta abar erregistratzeko prozesua deskribatuko dugu.

### ⚠ ADI

«LEHEN INSKRIPZIO»ko eskaera egiten da aurrez Jabetza Intelektualaren Erregistroan (IPR) inskribatu EZ den obra baten inskripzioa eskatu nahi denean, honako kasu hauetan:

- 1 | Inskribatu gabeko obren eskaerak, obren egileek, ekoizleek eta abarrek aurkeztuak.
- 2 | Inskribatu gabeko obren eskaerak, egileek, ekoizleek eta abarrek hirugarren bati haien eskubideak *inter vivos* eskualdatu badizkiote.
- 3 | Inskribatu gabeko obren eskaerak, egileek, ekoizleek eta abarrek euren oinordekoei *mortis causa* eskualdatu badizkiote.

Jabetza Intelektualaren Erregistroan lehen inskripzioa eskatzeko prozedura hasteko, lehen esandako webgunean, interesdunak orrian ageri den «Registrar» botoian klik egin beharko du. Prozedura zenbait eremutakoa denez, ondoren, sarbidea zer autonomia-erkidegotarako eskatzen den hautatu behar da; kasu zehatz honetan, Gaztela-Mantxako Autonomia Erkidegoa hautatu behar da.

### ⚠ ADI

Sartzean, erregistroa egiteko behar diren baldintza orokorren eta teknioen berri ematen da. Eskaera sinatzeko erabiliko den identifikazio-metodoa aukeratu behar da: **OINARRIZKO SINADURA** bidez (CI@ve bidez) edo **ZIURTAGIRIDUN SINADURA** (*Autofirm@* bidez; beraz, aplikazioak instalatuta egon beharko du).





Identifikazio-metodoa aukeratu ondoren, leiho bat irekiko da erregistroa egiteko online formularioarekin. Lau fitxa izango ditu: Eskatzailearen datuak, obra, egileak eta dokumentazio gehigarria. Eskatzailearekin harremanetan jartzeko datuen fitxan datu batzuk daude jada, baina «Otros datos del solicitante» bete beharko da, hala nola «Sexo» eta «Dirección». Erregistratzeko, beste fitxetako datuak osatu behar dira, hala nola «Obra», «Autores» eta «Dokumentación adicional» datuak, honako irudi hauetan ikusten den bezala:

1. irudia. Fitxa: «Datos del solicitante».

2. irudia. Fitxa: «Obra».

3. irudia. Fitxa: «Autores».



4. irudia. Fitxa: «Dokumentación adicional».

Datu guztiak bete ondoren, orri-oinen Jabetza Intelektualaren Erregistroaren Tasari buruzko informazioa dago. Hautatutako «Opción de pago» adierazi beharko da; hau da, aldeztatik banku-erakunderen batean egindako ordainketaren egiaztatzea eranstea edo ordainketa telematiko egitea. Azken kasu horretan, banku-entitatea menu zabalgarri batean aukeratu ahal izango da, eta, ondoren, ordaintzeko modua: banku-kontu bidez edo kreditu-txartel bidez. Erregistrarako behar diren datu guztiak sartu ondoren, «Enviar» botoia sakatu beharko da (formularioaren beheko aldean).

Dena zuzen badago, leiho bat agertuko da, interesdunari eskaera sina dezan, bidali aurretik. Bestela, aurkitutako akatsak zuzentzeko eskatuko du.

Eskaera sinatu ondoren, pantaila bat agertuko da, bertan adieraziko da eskaera behar bezala erregistratu dela, eta eskaeran sartutako datu guztiak erakutsiko dira, barne hartuta sortutako espediente-zenbakia, erregistro-zenbakia eta data. Aukera egongo da eskaera-erregistroaren egiaztatzea PDF formatuan deskargatzeko, bai eta eskaeraren datuak PDF formatuan dituen dokumentu bat ere.

Interesdunak obra bat erregistratu ondoren, sistemak web-helbide bat emango du identifikazio-alearen kopia aurkezteko. Erantsi beharreko dokumentuak aldatu egingo dira obraren tipologiaren arabera. Dokumentuek baimendutako fitxategien motaren eta tamainaren gaineko baldintzak bete beharko dituzte.

#### Informazio gehiago

Dokumentu osoa, lehen inskripzioko eskaera telematiko nola aurkeztu behar den zehatz deskribatuta, hemen dago eskuragai: [e.digitall.org.es/manual-primera-inscripcion](http://e.digitall.org.es/manual-primera-inscripcion)

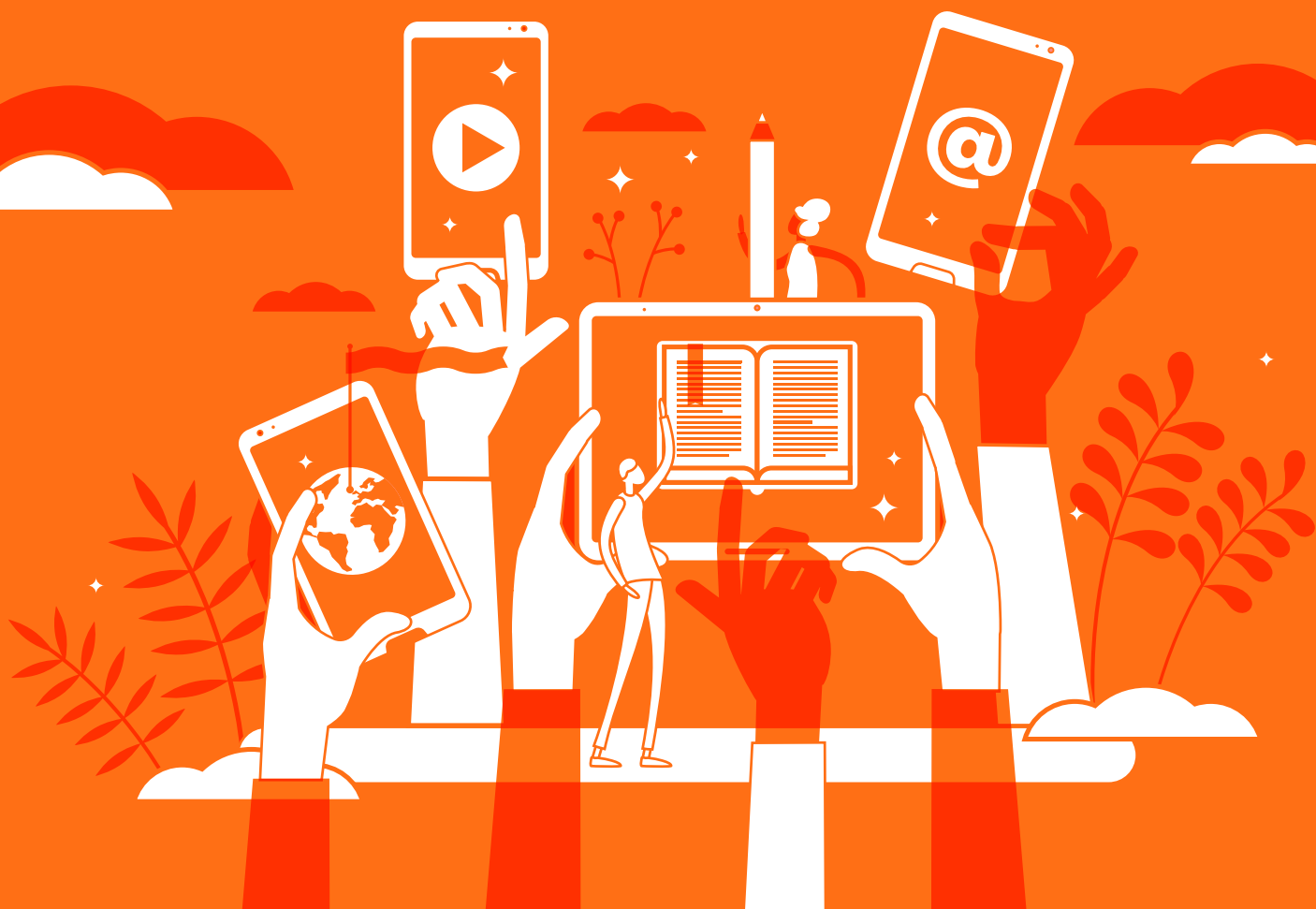


# DigitAll

Eduki digitalak  
sortzea

## 3.4

### PROGRAMAZIOA





Eduki digitalak  
sortzea

**C1 maila** 3.4 Programazioa

# Programazio- paradigmak. Ikuspegi orokorra





## Programazio-paradigmak. Ikuspegi orokorra

Programazioaren historian zehar, programak garatzeko modua aldatuz joan da, arazo gero eta konplexuagoak eta askotarikoagoak konpontzeko beharraren arabera. *Programazio-paradigma* esaten zaio programa baten diseinua eta idazketa nola landu zehazten duen ikuspegiari edo estiloari, eta prozesu osoa gidatzen duten arau, teknika eta printzipioez osatutakoa da. Programazio-paradigmak estuki lotuta daude programazio-lengoaiekin, programazio-lengoaia bakoitza programazio-paradigma bat edo gehiago onartzeko diseinatuta baitago. Izan ere, programazio-lengoaia batzuk paradigma bakarreko espezifikoak dira, eta beste batzuk, berriz, paradigma anitzekoak dira, eta zenbait paradigma onartzen ditu.

Gaur egun, paradigma desberdinak daude, eta bakoitzaren ezaugarri espezifikoek gainerakoetatik bereizten dituzte (besteak beste, datuak irudikatzeko moduari eta eragiketak eta kontrol-fluxua jorrazteko moduari lotuta). Aniztasun hori gertatu da paradigma zaharrentzat hartzen denaz geroztik (hau da, **programazio inperatiboa**) izandako bilakaeraren ondorioz. Izan ere, XX. mendeko 50eko hamarkadaren amaieran sortutako lehen programazio-lengoaletan dagoen haren jatorria. Paradigma horren oinarrian ideia hau dago: programa bat instrukzio-sekuentzia bat da, eta instrukziook ordena espezifiko batean exekutatu dira. Horrela, ikuspegi horren arabera, programak honako honetan dautza: arazo bat konpontzeko egin behar diren urratsen sekuentziaren deskribapen zehatza eta xehea egitean; hau da, arretagunea da arazoa «nola» konpontzen den. Lengoaia tipikoak dira, batzuk jada erabiltzen ez badira ere, honako hauek: Fortran, C, Pascal, Basic eta Cobol.

1960ko hamarkadaren amaieran, programak konplexuagoak egin ziren, eta argitasuna, kalitatea eta garapen-denbora hobetzeko beharra sortu zen. Horren ondorioz, **programazio egituratuaren** paradigma sortu zen. Oinarrian duen printzipioa da programa guztiak idatzi behar direla oinarritzko hiru kontrol-egiturak (sekuentzia, hautaketa eta iterazioa) eta moduluak edo prozedurak erabiliz. Ada, Algol eta Modula-2 programazio egituratuko lengoaiak dira. Bestalde, C lengoaia nagusiki programazio inperatiboko lengoaia bat da, baina haren ezaugarri batzuk direla-eta programazio egituraturako egokia ere bada.





La modu paraleloan garatzen da **programazio deklarati**boaren paradigma; horren arabera, paradigma inperatiboan ez bezala, programak idaztean zehazten da «zer» egin behar den, eta ez hori «nola» egin behar den. Horretarako, instrukzioak erabili beharrean, arauak eta propietateak erabiltzen dira. Programazio-lengoaia moderno batzuk, hala nola datu-baseetarako SQL edo web-orriak diseinatzeko HTML, paradigma horretan oinarritzen dira. Ikuspegi horren barruan, bi paradigma garrantzitsu daude:

- **Programazio logikoa**; horren arabera, programa adierazpen logikoetan oinarrituta eraikitzen da. Programak gertakari eta arau batzuk ezartzen ditu, eta logika matematikoa erabiltzen du ondorioak ateratzeko. Prolog programazio-lengoaia, 1970eko hamarkadan garatutakoa, paradigma hori inplementatzen lehenetariko bat izan zen.
- **Programazio funtzionala**: funtzio matematikoak bakarrik erabiltzea eta adierazpenak ebaluatzea baino ez ditu oinarri. Ikuspegi hori ez zen ospetsu bilakatu XX. mendeko 80ko hamarkadara arte. Lisp, Haskell eta Scheme funtsean ikerketa eta irakaskuntzarako erabiltzen diren lengoaia funtzionalak dira. Batez ere komunitate akademikoan erabili bada ere, azken urteotan ospe handiagoa lortu du softwarearen industrian, datu eta prozesu konkurrenteen kopuru handiak eraginkortasunez erabiltzeko gaitasuna duelako. Clojure, Scala eta F # lengoaiak, besteak beste, industrian erabiltzen diren programazio funtzionaleko lengoaien adibideak dira.

Hamarkada bat geroago, XX. mendeko 90eko hamarkadan, **objektuei bideratutako programazioaren (OBP)** paradigma zabaldu zen, informazioa bitzta errealean gertatzen den moduaren antzera irudikatzeko modu gisa, bai eta kodea berrerabiltzea ahalbidetzeko ere. Ikuspegi horren arabera, programa bat elkarriz eragiten dioten objektuen bildumatzat hartzen da; programa bera ere objektu bat izango litzateke. Objektuei bideratutako programazio-lengoaia ezagunenetako batzuk Java, Python eta C++ dira. Lengoaia moderno eta paradigma anitzeko horiek, gainera, programazio inperatiboa eta egituratua ahalbidetzen dute.





Gaur egun web-aplikazioak edo aplikazio mugikorrek garatzeko asko erabiltzen da, halaber, **gertaeretara zuzendutako programazioaren** paradigma da, nahiz eta 1980ko hamarkadaren hasieran sortu zen. Sistema banatuetan eta denbora errealean interakzioak modu asinkronoan erabiltzeko beharretik sortzen da, hala nola kontrol industrialeko eta telekomunikazioetako sistemetan. Hala ere, erabiltzailearen interfaze grafikoaren garapenak eman dio ospea. Java eta C++ programazio-lengoaiek onartzen dute programazio mota hori, baina zabaltzen ari dira beste lengoaia berriago batzuk, hala nola JavaScript eta C #.

Paradigma horiek guztiak programazio klasikoaren barruan sartzen dira. Gaur egun, paradigma guztiz desberdina hedatzen ari da, **programazio kuantikoaren**. Horren oinarria da mekanika kuantikoaren printzipioak erabiltzea programak garatzeko. Programa horietan, eragiketak gainjarri eta lotu egiten dira, eta eragiketa bat baino gehiago paraleloan egiteko aukera dago. Lengoiaren adibide bat Q# da.





Eduki digitalak  
sortzea

**C1 maila** 3.4 Programazioa

Garapen  
integratuko  
inguruneak (IDE).  
Ikuspegi orokorra







# Garapen integratuko inguruneak (IDE). Ikuspegi orokorra

## Sarrera

Garapen Ingurune integratuak [IDE siglekin ezagutzen dira, ingelesezko izenagatik duten nomenklaturagatik (*Integrated Development Environments*)], funtsezko tresnak dira programatzaile eta garatzaileentzat. Ingurune horiek softwarea garatzeko prozesua sinplifikatzen eta hobetzen duten tresna eta funtzionalitate integratuak ematen dituzte.

Dokumentu honetan, IDEen ezaugarri nagusiak aztertuko ditugu, baita haien erabilerarekin lotutako abantailak eta eragozpenak ere. Gainera, haien ikuspegia eta Python-en interprete interaktiboaren ikuspegia alderatuko ditugu.

Jarraitu aurretik, komeni da bideo hau ikustea:



### GARAPEN-FRAMEWORKEN HASTAPENAK

Bideo honetan, IDE kontzeptuari eta haren ezaugarri nagusiei buruzko oinarritzko informazioa dago.

[e.digitall.org.es/A3C34B1V03](https://e.digitall.org.es/A3C34B1V03)

## Zer da IDE bat?

IDE bat programazio-ingurune bat da, softwarea sortzera bideratutako hainbat tresna biltzen ditu. Haien artean, honako hauek nabarmendu behar ditugu:

### 1 | Kode-editorea

Iturburu-kodea idazteko eta editatzeko interfaze bat ematen du. Tresna ahaltsua da, eta arinago idaztea nahiz akatsak murriztea ahalbidetzen du. Haren ezaugarriak dira, besteak beste, sintaxia nabarmentzea, automatikoki osatzea eta testuinguru-iradokizunak.

### 2 | Araztailea

Kodea arazteak erroreak identifikatzea eta aurkitu ondoren haiek konpontzea ahalbidetzen du. IDEek arazteko askotariko aukerak ematen dizkiote garatzaileari, hala nola haustura-puntuak txertatzea eta aldagaiak ikuskatzea.





### 3 | Nabigazioa kodean

Kodearen barruan azkar nabigatzea eta elementu jakin batzuk erraz eta eraginkortasunez bilatzea errazten du. Funtzioak, motak eta aldagaiak bilatzeko aukera ematen du. Gainera, proiektuaren ikuspegi hierarkikoak ematen ditu, kodearen egitura hobeto ulertzeko.

### 4 | Bertsioak kontrolatzeko sistemekin integratzea

Bertsioak kontrolatzeko sistemak erabiltzea ohikoa da programatzaileen artean. Horregatik, IDEak daude bertsioak kontrolatzeko sistema hedatuenetan. Bertsioak kontrolatzeko sistema batek aukera ematen du kodean denboran zehar egindako aldaketen jarraipena egiteko. Horrela, garatzaileek beren kodearen bertsioak kudeatu eta kontrola ditzakete, kanpoko aplikazioetara jo beharrik gabe.

### 5 | Probak

Software-probak funtsezko etapa bat dira edozein aplikazio sortzean. Funtzionalitate berriak diseinatzea eta garatzea bezain garrantzitsua da probak diseinatzea eta garatzea. IDEek tresnak eta funtzionaltasunak ematen dituzte proba-prozesua errazteko eta softwarearen kalitatea ziurtatzeko.

### 6 | Kanpoko tresnekin integratzea

IDEek kanpo-tresna ugari integratzea ahalbidetzen dute aplikazioak garatzeko prozesua osatzeko. Tresna horiek barne hartzen dute softwarea diseinatu eta planifikatzea, bai eta hura garatu, optimizatu eta probatzea ere.

Garapen-ingurune integratua (IDE) softwarea sortzera bideratutako programa informatiko bat da, eta garatzaileek gehien erabiltzen dituzten tresnak interfaze grafiko berean integratzen ditu.





## IDEen abantailak

IDE bat osatzen duten tresna nagusiak deskribatu ondoren, erraza da software-garatzaillei dakarzkieten abantailak identifikatzea. Abantailarik garrantzitsuena, beste alderdi batzuetan eragina duena, zera da: produktibitatea handitzea garatzeko prozesuaren etapa guztietan. Garatzaileek interfaze grafiko bakarria dute, eta interfaseak barne hartzen ditu softwarearen bizi-zikloaren etapa guztiak (analisi eta diseinua, kodifikazioa, probak eta ondorengo mantentze-lanak) gauzatzeko behar diren tresna guztiak.

Abantaila horren arrazoia da garatzaileek erraz txandakatu ditzaketela proiektuaren zeregin edo fase desberdinak, aplikazioa etengabe aldatu beharrik gabe. Gainera, IDEak oso konfiguragarriak dira, eta horrek dakar garatzaile bakoitzaren behar eta gustu partikularretara egokitzea.

Nabarmendu behar da IDE bat erabiltzeak produktibitatea handitzen duela programak kodetzeko behar den denbora murrizten baitu, eta horretaz gain, kode fidagarri eta sendoagoa sortzen ere laguntzen du. Bi abantaila horiek zenbait funtzio erabiltzearen ondorio dira (automatikoki osatzea, sintaxia nabarmentzea eta akatsak denbora errealean zuzentzea). Ezaugarri horiek IDEen editorean daude eskuragarri.

Garatzaile gehienek taldean lan egiten dutenez, ezinbestekoa da lan-estandar komun batzuk ezartzea. IDEek emandako txantiloiak erabil ditzakete taldeko kide guztiek, eta horrek baterako lana errazten du. Gainera, bertsioak kontrolatzeko tresnen integrazioak kodea modu seguruan partekatzea errazten du.

Laburbilduz, IDEek garatzaileen produktibitatea handitzen dute, instalatzeko denbora murrizten dute eta garatzeko zereginak arintzen dituzte.



## IDEa vs Python-eko interpretatzailea

Aurreko gaitan interpretatzaile interaktiboaren kontzeptua aztertu dugu, bereziki Python-eko interpretatzaile interaktiboa. Python-eko interpretatzaile interaktiboa programa bat da. Programak Python-en idatzitako instrukzioak irakurtzen ditu, ebaluatzen edo prozesatzen ditu, eta exekutatzen ditu. Hala ere, IDEaren kontzeptua askoz ere konplexuagoa da, eta, gainera, interpretatzaileak eta konpiladoreak ditu tresnen artean.



### INTERPRETATZAILE INTERAKTIBOAK

*Bideo honetan, interpretatzaile interaktiboaren kontzeptua azaltzen da, bai eta programazio-tresna gisa ematen dituen abantailak ere. Zehazki, Python-eko interpretatzaile interaktiboa jorratzen du, eta hura instalatzeko eta exekutzeko oinarrizko nozioak ematen ditu.*  
[e.digitall.org.es/A3C34B2V02](https://e.digitall.org.es/A3C34B2V02)

IDEek abantaila ugari dituzte, baina programa bat garatzeko orduan abantailak eta desabantaila nagusiak aztertu behar dira. Haietako bat ikaste-kurbaren konplexutasuna da; hain zuzen ere, adituak ez diren erabiltzaileentzat. Gainera, garatzailearen esperientzia ez ezik, garatu nahi den programa mota ere hartu behar da kontuan.

Jarraian dagoen taulan Python-en interpretatzaile interaktiboarekin alderatzen dira IDEen erabilera, ezaugarriak eta ikaste-kurba.

	IDEak	Python-eko interpretatzailea
<b>Erabilera</b>	Proiektu konplexuetan gomendatua.	Proba azkarrak garapen errazak egiteko gomendatua.
<b>Ezaugarriak</b>	<ul style="list-style-type: none"> <li>- Kode-editorea, barne hartuta sintaxia nabarmentzea eta automatikoki osatzea.</li> <li>- Integrazioa kanpoko tresnekin eta liburutegiekin.</li> <li>- Produktibitate eta arintasun handiagoa softwarearen garapenean.</li> </ul>	<ul style="list-style-type: none"> <li>- Editore erraza.</li> <li>- Python-eko liburutegi estandarerrako sarbidea.</li> <li>- Python-eko kodea erraz eskuratu eta exekutatzea.</li> </ul>
<b>Ikaste-kurba</b>	Konplexua.	Erraza.



Laburbilduz, askotariko faktoreen mende dago IDE bat edo beste tresna bat (adibidez, Python-eko interpretatzaile interaktiboa) erabiltzea, eta haien artean daude, besteak beste, proiektuaren konplexutasuna, behar espezifikoak eta garatzailearen esperientzia. IDEak egokiagoak dira ezaugarri aurreratuak eskatzen dituzten proiektu konplexuetarako; Python-en interpretatzaile interaktiboa, aldiz, egokiagoa da garapen sinpleagoak eta proba azkarrak egiteko.

#### OHARRA

Python-ek oinarrizko IDE bat du oinarrizko instalazioan: IDLE (Integrated Development and Learning Environment). IDE baten oinarrizko ezaugarriak ditu, hala nola kode-editore bat (sintaxia nabarmentzea, automatikoki osatzea, testu-bilaketa barne), eta kodea modu interaktiboan exekutatzeko gaitasuna. Aukera ematen du, gainera, Python-eko script-ak sortzeko eta exekutatzeko, kodea urratsez urrats arazteko, dokumentazio integratua atzitzeko eta proba unitarioak exekutatzeko.

#### Informazio gehiago

Hona hemen Python-en programatzaileen arteko IDE ezagunenetako batzuk:

- **PyCharm:** [jetbrains.com/pycharm](https://jetbrains.com/pycharm)
- **Spyder:** [spyder-ide.org](https://spyder-ide.org)
- **Pydev:** [pydev.org](https://pydev.org)





Eduki digitalak  
sortzea

**C1 maila** 3.4 Programazioa

**Salbuespenak.  
Zer dira eta  
zertarako balio  
dute?**





## Salbuespenak. Zer dira eta zertarako balio dute?

Oro har, salbuespen bat zera da: programa bat exekutatzean izaten den gertaera, zeinak aurrez definitutako exekuzio-fluxua aldatzen duen. Ildo horretan, salbuespen bat gertatzea nahi ez den zerbait da, urruntzen baita programa batean espero izatekoa den portaeratik. Hala ere, posible da exekuzio-denboran gertatzen diren salbuespenak atzematea eta erabiltzea, hau da, programa bat exekutatzean ari denean, normaltasuna berreskuratzeko eta hura exekutatzean jarraitzeko.

Programazio-lengoaia moderno gehienek, hala nola Python-ek, salbuespenak definitzeko, atzemateko eta kudeatzeko euskarri natiboa hartzen dute barne. Horrek esan nahi du lengoaiak berak instrukzio espezifikoak ematen dituela salbuespenak tratatzeko, eta, horrela, errazten duela exekuzio-denboran gerta litezkeen erroreen aurreko programa sendoak garatzea. Hori da salbuespenak kudeatzeko zeregin nagusia.

Bestalde, salbuespenak honela sailkatzen dira: i) programazio-lengoaiak jasotako salbuespenak eta ii) erabiltzaileak zehaztutako salbuespenak. Lengoaiak jasotako salbuespenaren adibide klasiko bat agertzen da instrukzio bat zati zero egiten saiatzen denean. Python-en, salbuespen hori honako honek definitzen du: `ZeroDivisionError`. Erabiltzaileak zehaztutako salbuespen-adibide bat hau izan liteke: 9 digitu zehazki ez dituen telefono-zenbaki bat gordetzen saiatzea. Azken adibide horri dagokionez, erabiltzaileak definitu beharko luke salbuespen mota berri hori, datu-egitura berri baten definizioarekin gertatu bezala.



### ⚠ ADI

Salbuespenak erabiltzean, hiru eragiketa adierazgarri mota daude: salbuespenak abiaraztea, salbuespenak atzematea eta salbuespenak erabiltzea. Abiaraztea zera da: programatzaileak salbuespen bat jaurtitzeko duen gaitasuna; eta normalean, programaren beste atal batek hura jaso eta erabiliko du. Atzemateak kode-mailako instrukzio bat esplizituki detektatzea dakar. Azkenik, aurretik atzemandako salbuespena tratatzeko definitutako iturburu-kodeari lotuta dago erabilera.



## Python-eko salbuespenen erabileraren adibidea

Atal honetan, Python-eko salbuespenen erabilera-adibide simple bat ikusiko dugu. Helburua ez da sakontzean ea Python-en salbuespenak nola definitzen eta erabiltzen diren. Aitzitik, asmoa da, besterik gabe, aurretik ikusi dugun salbuespenaren kontzeptuaren adibideak ematea, esandako lengoaiaren bidez.

Ildo horretan, hurrengo kode-zatiak erakusten du nola lor daitekeen zenbaki bat teklatura erabiliz, eta ondoren, **balio** izeneko aldagai batean biltegitatu. Gero, aldagaiaren edukia erakutsiko da.

```
>>> balioa = int(input("Sartu zenbaki bat..."))
Sartu zenbaki bat...7
>>> balioa
7
```



Zoritarrez, kode horrek huts egingo du erabiltzaileak, adibidez, testu-kate bat sartzen duenean. Kasu honetan, Python-eko interpretatzaileak balio bat itxarongo du (osoko zenbaki bihurtzearen ondorioz), eta, horregatik, **ValueError** salbuespena emango du.

```
>>> balioa = int(input("Sartu zenbaki bat..."))
Sartu zenbaki bat... Igantzi
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'Igantzi'
```

Python-en, aurrerago xehetasun handiagoz ikusiko duzun bezala, posible da kode-segmentu bat biltzea gerta litezkeen salbuespenak atzemateko. Horretarako, try-except blokeak erabil ditzakezu. Hurrengo kode-zatiak atzeman eta kudeatzen du erabiltzaileak teklatura erabiltzean egindako akatsa; hau da, gure aurreko adibidean, osoko zenbaki ez beste zerbait idaztea.

```
>>> try:
...     balioa = int(input("Sartu zenbaki bat..."))
... except ValueError:
...     print("Zenbaki bat sartu behar duzu")
...
Sartu zenbaki bat...Irun
Zenbaki bat idatzi behar duzu
```





Eduki digitalak  
sortzea

**C1 maila** 3.4 Programazioa

# Python-eko fitxategiak prozesatzea





## Fitxategiak prozesatzea Python-en

Lehen ikusi dugun bezala, Python-ek zenbait funtzio ditu fitxategiak irekitzeko, irakurtzeko, idazteko eta ixteko. Funtzio horiek era guztietako fitxategiei aplikatzen zaizkie, baina gidaliburu honetan emango dira, hain zuzen, testu sinpleko fitxategiekin lan egiteko jarraibideak.

### Erroreen oinarritzko kudeaketa

Erroreak kudeatzea edozein programaren zati erabakigarria da. Python-en, erroreak **try/except** blokeak erabiliz kudeatzen dira.

Existitzen ez den fitxategi bat irekitzen saiatuz gero, Python-ek errore bat emango du: **FileNotFoundError**. Errore hori **try/except** bloke bat erabiliz atzeman eta kudea daiteke, programaren exekuzioa bat-batean amaitu ez dadin. Adibidez:

```
# Errore-kudeaketaren adibidea
try:
    with open("archivo_inexistente.txt", "r") as archivo:
        print(archivo.read())
except FileNotFoundError:
    print("Fitxategia ez da aurkitu.")
```

Kasu horretan, fitxategia aurkitzen ez bada, pantailan «Fitxategia ez da aurkitu.» mezua agertuko da, programa errore batekin amaitu ordez, eta programa exekutatzen jarraituko du blokearen ondoren sententzia gehiago badaude.

### Testu-kate bat bilatzea fitxategi baten edukian

Fitxategiak prozesatzean, zeregin komun bat da testu-kate espezifiko bat bilatzea fitxategiaren edukiaren barruan. Jarraian, adibide batean ikusiko dugu hori nola egin, baita sortzen diren erroreak nola kudeatu ere:





```
# Fitxategi bateko testu-bilaketaren adibidea
fitxategi_izena = "fitxategia.txt"
bilaketa_katea = "bilatu beharreko testua"
try:
    with open(fitxategi_izena, "r") as fitxategia:
        edukia = fitxategia.read()
        if bilaketa_katea in edukia:
            print("Bilaketa-katea fitxategian aurkitu da.")
        else:
            print("Bilaketa-katea ez da fitxategian aurkitu.")
except FileNotFoundError:
    print (f"{fitxategi_izena} fitxategia ez da aurkitu.")
```

Kode horretan, `fitxategi_izena` aldagaiak adierazitako fitxategia irekitzen da, haren edukia irakurtzen da, eta, ondoren, `in` eragilea erabiliz, egiaztatzen da ea bilaketa-katea fitxategiaren edukian dagoen. Bilaketa-katea fitxategian bada, mezu bat agertuko da hura adierazteko. Ez bada, beste mezu bat agertuko da. Fitxategia aurkitzen ez bada, errore-mezu bat agertuko da.

Adibide honek behar bezala funtzionatzen du eduki gutxiko fitxategietarako. Eduki oso handiko fitxategiak baditugu (gigabyteak), oso litekeena da teknika horrek ez funtzionatzea; izan ere, programak ezin izango du fitxategia osorik kargatu sistemaren memorian bilaketa egiteko.

Hori konpontzeko aukera bat izan liteke fitxategiaren edukia lerroz lerro irakurtzea, memorian, gehienez ere, fitxategiaren lerro bakarra aldi berean kargatzeko. Adibidez:

```
# Fitxategi bateko testu-bilaketaren adibidea (bertsio hobetua)
fitxategi_izena = "fitxategia.txt"
bilaketa_katea = "bilatu beharreko testua"
try:
    with open(fitxategi_izena, "r") as fitxategia:
        lerroa = fitxategia.readline()
        while lerroa:
            if bilaketa_katea in lerroa:
                print("Bilaketa-katea fitxategian aurkitu da.")
                break
            lerroa = fitxategia.readline()
        else:
            print("Bilaketa-katea ez da fitxategian aurkitu.")
except FileNotFoundError:
    print (f"{fitxategi_izena} fitxategia ez da aurkitu.")
```



Kasu horretan, fitxategia ireki eta lerroz lerro irakurriko da. Bilaketa-katea fitxategiaren lerroren batean badago, mezu bat agertuko da hura adierazteko, eta zikloa etengo da; alegia, ez dugu fitxategia prozesatzen jarraitu beharrik izango. Ez badago, eta lerro guztiak irakurri badira, beste mezu bat agertuko da. Fitxategia aurkitzen ez bada, errore-mezu bat agertuko da.

Esan bezala, kodearen bertsio hori eraginkorragoa izango da memoriari dagokionez, oso fitxategi handiekin lan egitean; izan ere, memoriaren fitxategi-lerro bakarra baino ez du aldi berean, fitxategiaren eduki osoaren orde.

## Ondorioa

Fitxategiak askotariko atazak egiteko prozesatzen dira; besteak beste, datuak analizatzeko eta atazak automatizatzeko. Aurretik, zenbait teknika ezagutu ditugu fitxategiak irekitzeko eta irakurtzeko, erroreak erabiltzeko eta fitxategi baten edukian testu-kateak bilatzeko.

### Informazio gehiago

Python-en erroreak kudeatzeari buruz gehiago jakiteko, kontsulta ezazu gaztelaniazko dokumentazio ofiziala: [e.digitall.org.es/excepciones](https://e.digitall.org.es/excepciones)





Eduki digitalak  
sortzea

**C1 maila** 3.4 Programazioa

**Objektuetara  
bideratutako  
programazioa.  
Oinarrizko  
printzipioak eta  
kontzeptuak**





## Objektuetara bideratutako programazioa. Oinarriko printzipioak eta kontzeptuak

Programatzeko orduan, programazio-lengoaiek askotariko programazio-filosofia edo -paradigmei jarraitzen diete, kontuan hartuta zer errealitate modelatu nahi duten eta zer arazo mota konpondu nahi dituzten.

Ingurutzen gaituen errealitatean askotariko objektuez beteta dago (besteak beste, aulkiak, mahaiak eta autoak), eta objektuok programazioaren bidez konpontzeko moduko arazoak sor ditzakete. Errealitate hori irudikatzeko, **objektuei bideratutako programazioaren (OBP)** paradigmak bereziki azpimarratzen ditu objektuen diseinua eta manipulazioa, era guztietako programetarako soluzioa diseinatzeko funtsezko elementu gisa. Jarraian, paradigma horren funtsezko printzipioak eta kontzeptuak deskribatuko dira.

### Funtsezko kontzeptuak

Bizitza errealeko arazo baten konponbidea modelatzeko orduan, askotariko objektuekin lan egin behar da, hala nola aulki eta mahai batekin, arazoaren parte izan baitaitezke. Horrenbestez, objektu horiek formalki deskribatu beharra dago, baina ez ezaugarri guztiak, programarako garrantzitsuenak baizik.

### Abstrakzioa

Abstrakzioa OBPre oinarrietako bat da. Modelatu nahi den errealitaterako funtsezkoak diren ezaugarriak ezagutzeko prozesu bat da. Aukera eman behar du programa baterako garrantzirik ez duten xehetasunak alde batera uzteko, eta ezarri beharreko gutxieneko xehetasunak baino kontuan ez hartzeko.

Demagun pertsonak parte hartzen duten programa bat ezarri nahi dela. Lehenik eta behin, **Pertsona**ren kontzeptua modelatu behar da. Pertsona hori definitzeko oinarriko elementuak bi motatakoak dira: **atributuak eta ekintzak**. **Atributuak** pertsona definitzen duten ezaugarriak edo aldagaiak dira, hala nola izena eta adina. **Ekintzak**, berriz, atributu horien gainean jarduten duten metodoak izango dira.

Mota	Pertsona
Atributuak	izena adina
Ekintzak	galdetuzena urteakBetetzea



Pertsona batek ezaugarri gehiago ditu, hala nola sexua eta NANA. Alabaina, abstrakzio-prozesu bat eginez, ezarri beharreko programarako beharrezkoak diren gutxieneko atributuak eta ekintzak zein diren zehaztu behar da. Irudi honetan Pertsona mota baten eskema orokorra ikusten da. Kontuan hartuta programa batean Pertsona bat izango denaren definizio edo eskema orokor hori, OBPn **klase** kontzeptuari dagokio.

## Klasearen kontzeptua

Orain arte, Python-en datuak kudeatzeko, datu-mota primitiboak erabili dira, hala nola *int*, *float* eta *string*. Horri esker, datu-mota simple bat irudikatzen aukera dago, hala nola osoko zenbaki bat eta testu-kate bat. Hala ere, egoera batzuetan beharrezkoa da datu-mota konplexuagoak modelatzea, beste datu sinpleago batzuen konposizio gisa eraturakoak. Hartarako, datu-egitura konplexuagoen kontzeptua erabil daiteke (adibidez, bektoreak eta matrizeak). Normalean homogeneous izaten dira, hau da, datu-mota bera izaten dute; adibidez, balio guztiak zenbakizkoak dira eta guztiak testu-kateak dira. Era berean, heterogeneous izan daitezke, hau da, balio osoak balio errealekin eta testu-kateekin konbina daitezke, adibidez.

Hainbat datu-motaren gainean lan egiteaz gain, datuak manipulatzeko eragiletak gehitzen badira, **datu-mota abstraktu** baten definizioa izango dugu aurrean. Beraz, **klasearen** kontzeptua datu-mota abstraktu bat da —hau da, mota berri bat—, osoko zenbaki eta testu-kate gisa ezagutzen ditugun oinarrizkoen gain.

Laburbilduz, **klase** batek entitate bat adierazten du, eta entitate horren atributuak ezarriko dira klase horren aldagaien bidez eta funtzio-multzo baten bidez egin ditzakeen ekintzen bidez. Datu-mota abstraktutzat har daiteke, eta barne hartzen ditu jarduketaren xede diren datuak edo aldagaiak eta datu horien gainean egin daitezkeen ekintzak edo metodoak. Hona hemen lehen ikusitako **Pertsona** klasearen Python-eko ezarpenaren adibide bat:





### #PERTSONA KLASEAREN ITURBURU-KODEA

```
class Pertsona (object):
    def __init__(self, izena, adina):
        self.izena = izena
        self.adina = adina
    def galdetuIzena(self):
        return self.izena
    def urteakBetetzea (self):
        self.adina += 1
```

**GaldetuIzena** funtzioak aukera ematen du ezarritako Pertsona objektu bakoitzaren izena atributua kontsultatzeko, eta **urteakBetetzea** funtzioak, berriz, aukera ematen du Pertsona objektu bakoitzaren adina aldagaiaren balioa eguneratzeko. Horrela, kapsularatze-printzipioa aplikatzen da, OBPren beste zutabeetako bat. **Adina** eta **izena** datuak babestuta daude, ez dira zuzenean eskuratzen, kontsultatzeko eta manipulatzeko bereziki diseinatutako metodoen bidez soilik.

Gainera, **\_\_init\_\_** metodoa nabarmendu behar da; metodo horri **metodo eraikitzailea** edo objektu baten atributuak (adina eta izena) hasieratzeko metodo esaten zaio.

## Objektuaren kontzeptua

Horrela, klase bat datu-mota berri bat da —arestian ikusitako *boolean* eta *int* motan bezala—, eta datuok modu abstraktuan balio logiko edo oso bat adierazten dute. Hala ere, balio jakin baten gainean jarduteko, aldagai bat definitu behar zen. OBPren antzeko kontzeptua dago; alegia, objektu bati erreferentzia egiten dion kontzeptua.

Analogiaz arrazoituta, aldagai bat sortzeko modua izen bat eta hasierako balio bat ematea da. Objektu bat sortzeko modu bera dago, hau da, erreferentziari izen bat ematea eta haren atributu guztien balioak esleitzea.

```
aldagaiIzena = hasierakoBalioa
erreferentziaIzena = atributuen hasierako balioak
```

Objektu batek zenbait atributu dituzenez, ez da hasierako balio bakarra egongo, batzuk baizik. Horregatik, metodo eraikitzaileak atributu guztiak aldi berean hasieratzen ditu.

```
pepe = Pertsona(izena = "Joxe", adina = "20") #objektu bat hasieratzea
```





Klase bat behin bakarrik definitzen da, baina programa batean behar diren adina objektu —hau da, pertsona— sortuko dira.

Objektuen ezaugarri nagusietako bat haien **egoera** da. **Egoerak** atributuen egungo egoera adierazten du, hau da, aldagaien balioa. Egoera objektu dinamikoa da. Hala, pertsona batek gaur 20 urte izan ditzake gaur, eta bihar 21, hau da, bere egoerak bilakaera izan dezake denborarekin.

Python-en, objektu baten edozein ekintza edo metodo deitzeko, “.” puntu eragilearekin batera sortutako erreferentzia erabiliko da. Horrela, nahi badugu pertsona batek urte bat gehiago betetzea edo nola deitzen den galdetzea, aski da metodoak honela deitzea:

```
pepe.urteakBetetzea()  
print(pepe.galdetuIzena())
```

## Objektuen/klaseen arteko erlazioak

Klaseak eta objektuak ez dira besterik gabe existitzen diren elementu estankoak. Bizitza errealean bezala, objektuek elkarri eragiten diote, eta erlazioak ezartzen dituzte. Askotariko erlazioak daude, eta predikatu mota desberdinek bereizten dituzte.

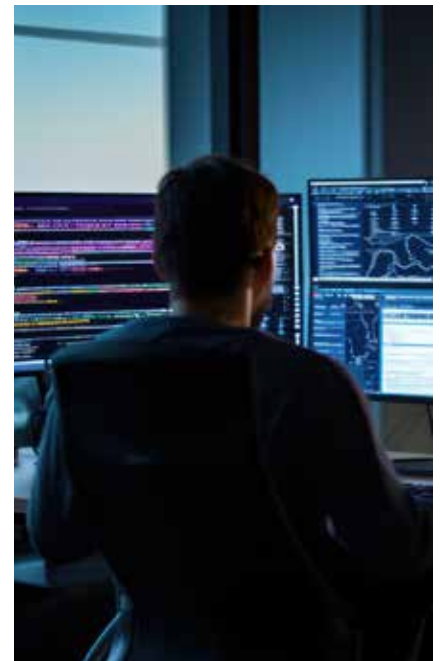
### 1 | Elkartze-erlazioak

Honelako predikatuen bidez identifikatzen dira: «...bat edo batzuk ditu»

Objektuen arteko egiturazko erlazio bat adierazten dute, hau da, denboran irausten duen erlazio bat. Adibide bat da: auto batek **zenbait** gurpil **ditu**. Autoa klase baten modura adierazten da, eta markaren eta matrikularen atributuak ditu, eta gurpil bat adieraziko da marka, presio eta diametroaren tankerako atributuak dituen klase gisa.

### 2 | Mendekotasun-erlazioak

Honelako predikatuen bidez identifikatzen dira: «...bat edo batzuk erabiltzen/ezagutzen ditu»





Objektuen arteko denbora-erlazioa adierazten dute. Adibide bat hau da: pertsona batek boligrafo bat **erabiltzen du**. Aurreko adibidean ez bezala, autoak beti ditu gurpilak, baina pertsona batek ez du beti boligrafoa soinean; hau da, behar denean erabiltzen du, eta harekiko erlazioa galtzen du erabili ondoren.

### 3 | Herentzia-erlazioak

Herentzia da OBPren oinarrizko beste zutabeetako bat. Herentzia-erlazioak honelako predikatuen bidez identifikatzen dira: "...bat da"

Erlazio hauek objektuek ezaugarri eta portaera komunak dituzte, hau da, metodoak eta atributuak, baina elkarren arteko desberdintasun batzuk ere badaude, eta bakarrak dira haien ondorioz.

Adibide bat dira Animalia, Arraina eta Txakurra klaseak. Arraina animalia mota bat **da** eta txakurra animalia mota bat **da**, eta horrenbestez, gauza komunak izango dituzte; adibidez, biek adina dute, eta izen bat jar dakieke. Hala ere, ezaugarri batzuk ez dira komunak, hala nola mugitzeko modua: batak igeri egiten du, eta bestea oinez doa.

Herentzia da, gainera, **polimorfismo**aren oinarria, OBPren beste zutabeetako bat. Aukera ematen du, adibidez, Animalia klaseko matrize batek Arrain klaseko edo Txakur klaseko zenbait objektu edukitzeko.

Erlazio horietako bakoitza bere modura ezartzen da lan egiten dugun programazio-lengoaian.

#### Informazio gehiago

Python.org. Klaseen definizioa: [e.digitall.org.es/python-clases](https://e.digitall.org.es/python-clases)  
Microsoft Company. Zer da OBPa? [e.digitall.org.es/poo](https://e.digitall.org.es/poo)



Eduki digitalak  
sortzea

**C1 maila** 3.4 Programazioa

# Kode-probak. Oinarrizko alderdiak





## Kode-probak. Oinarrizko alderdiak

**Softwarea garatzeko zikloan**, zenbait etapa daude, betekizunak analizatzetik sistema funtzionamenduan mantentzeraino. Ziklo horretako etapa kritikoenetako bat proben etapa da. Kode-probak, edo testak, softwarea garatzeko edozein prozesuren parte dira. Modua ematen dute idatzitako kodeak espero bezala funtzionatzen duela eta zehaztutako betekizunak betetzen dituela ziurtatzeko.

### Software-garapenaren zikloa

Softwarea garatzeko zikloak etapa hauek ditu:

- 1 | Betekizunak:** garatu beharreko sistemaren betekizunak jaso eta aztertzen dira.
- 2 | Diseinua:** sistemaren arkitektura erabakitzen da eta osagaiak diseinatzen dira.
- 3 | Garapena:** sistema ezartzen da diseinuetan oinarrituta.
- 4 | Probak:** egiaztatzen da sistemak betekizunak betetzen dituela eta behar bezala jokatzen duela.
- 5 | Integrazioa:** sistemaren osagai guztiak konbinatzen dira eta osotasun gisa egiaztatzen dira.
- 6 | Mantentze-lanak:** funtzionamenduan dagoen sistemaren jarraipena egiten da, akatsak zuzentzen dira eta behar berrietara egokitzen dira.

**Probak** egiten dira eraikitako sistemaren eta jatorrizko betekizunen arteko erroreak, akatsak eta desadostasunak identifikatzeko. Balio dute kodean akatsak aurkitu eta konpontzeko ez ezik, baita sistemak betekizunak betetzen dituela egiaztatzeko eta erabiltzaileak zer behar duen baliozkotzeko ere. Probak modu bat dira softwarearen kalitatea bermatzeko eta integrazio-etapan dauden arazoen kopurua murrizteko.

Hurrengo orrietan, proben bidez gidatutako garapenaren ikuspegia (*Test-Driven Development* edo *TDD*) aztertuko dugu, bai eta zer erlazio duen kode-probekin eta Python-en nola ezar daitezkeen ere.





## Proben bidez gidatutako garapena (TDD)

Proben bidez gidatutako garapena (*Test-Driven Development*) softwarea garatzeko metodologia bat da, eta oinarritzat du garapen-ziklo oso labur bat errepikatzea: lehenik, garatzaileak proba-kasu automatizatu bat idazten du, nahi den hobekuntza edo funtzio berri bat definitzeko; gero, proba hori gainditzeko behar den gutxieneko kodea sortzen du, eta, azkenik, kode berria maila onargarrira iristeko hobetzen du.

### TDD ikuspegia vs. Tradizionala

Ikuspegi tradizionalan, softwarearen garapenak honako ordena honi jarraitu ohi dio: i) kodea idazten da, ii) proba bat egiten da behar bezala funtzionatzen duela egiaztatzeko eta iii) akatsak aurkitzen badira, zuzendu egiten dira eta berriz probatzen da. Prozesu horrek jarraitzen du kodeak proba guztiak gainditu arte.

Aitzitik, TDDk erabat aldatzen du ikuspegi hori. Produkzio-kodea idatzi aurretik, garatzaileak kode berrirako proba bat idatziko du. Hasieran, proba horrek huts egingo du, probatzen ari den kodea oraindik ez delako existitzen. Gero, garatzaileak proba gainditzeko behar den gutxieneko kodea idatziko du. Azkenik, kodea findu eta hobetu egiten da, jokabidea aldatu gabe.

### TDD Python-en

Python-ek, maila handiko eta helburu orokorreko programazio-lengoaia denez, zenbait tresna ematen ditu TDD metodologia eraginkortasunez ezartzeko.

Adibidez, demagun bi zenbaki batzen dituen funtzioa ezarri nahi dela. TDD ikuspegiarekin, lehenik, funtzio honetarako proba bat idatziko da:

```
import unittest

class TestSuma(unittest.TestCase):
    def test_batura(self):
        emaitza = batu(1, 2)
        self.assertEqual(emaitza, 3)

if __name__ == '__main__':
    unittest.main()
```



Kode hori exekutatzean, probak huts egingo du, oraindik ez baita `sum()` funtzioa definitu. Horren ondoren, funtzioa ezarriko da proba gainditzea lortzeko:

```
def sum(a, b):
    return a + b
```

Eta probak berriro egingo dira. Orain gaindituko dira, eta gure funtzioak proban definitu dugun itxaropena betetzen duela adieraziko du horrek.

## Ondorioa

Kode-probak softwarea garatzeko prozesuen zati integrala dira. Modu bat ematen dute ziurta dezagun garatutako softwareak espero bezala jokatzen duela eta definitutako baldintzak betetzen dituela. Bestalde, Proben bidezko gidatutako garapena (TDD) izeneko metodologiak software-garapenaren erdigunean jartzen ditu probak, eta sustatu egiten du kalitate handiko kodea idaztea erabiltzailearen beharrak betetzeko eta erroreak agertzea minimizatzen.

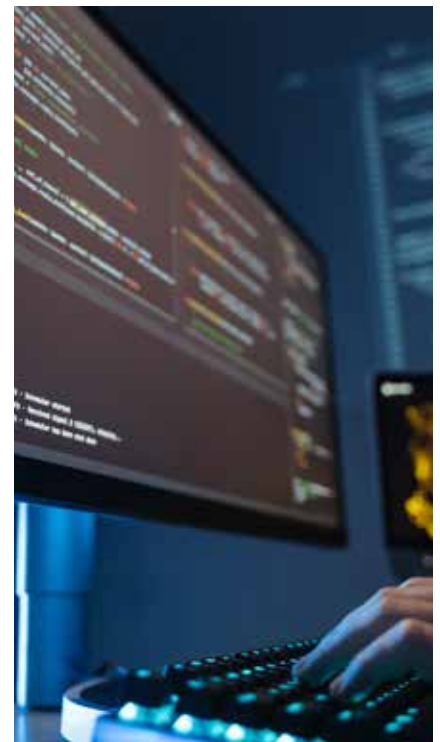
Python, dituen proba-tresnei eta bere sintaxi argi eta zehatzari esker, TDD metodologia ezartzeko aukera bikaina da. Probak ekoizpen-kodearen aurretik idazteak koderako itxaropenak argi eta garbi definitzen laguntzen du, akatsak hautematea errazten du, eta kode garbi eta mantengarria idaztea sustatzen du.

Probak inbertsio bat dira, eta etorkizunean denbora eta ahalegin handia aurrezten dute; izan ere, ustekabeko akatsak agertzea eragozten dute, eta akatsak detektatzea eta zuzentzea errazten dute. Probetan oinarritutako garapenaren ikuspegia hartzeak balio du softwarearen kalitatea hobetzeko, eta garatzeko prozesua efizienteagoa eta eraginkorragoa izan dadin.

### Informazio gehiago

Python-eko proben bidez gidatutako garapenari buruz gehiago jakiteko, kontsulta ezazu modulu hauen gaztelaniazko dokumentazio ofiziala:

- **unittest:** [e.digitall.org.es/unittest](https://e.digitall.org.es/unittest)
- **doctest:** [e.digitall.org.es/doctest](https://e.digitall.org.es/doctest)





# 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>