Mondragon Unibertsitatea

Goi Eskola Politeknikoa

GBL

GRADU BUKAERAKO LANA

ITURRIEN KUDEAKETARAKO PANEL ETA DATU BASEAREN SORRERA

ARANZADI OLAZABAL, MARTXEL

2023 – 2024

INFORMATIKA INGENIARITZAKO GRADUA

GRADU BUKAERAKO LANA GRADU BUKAERAKO LANA GRADU BUKAERAKO LANA GRADU BUKAERAKO LANA GRADU BUKAERAKO LANA

ITURRIEN KUDEAKETARAKO PANEL ETA DATU BASEAREN SORRERA

GRADU BUKAERAKO LANA, MUko Goi Eskola Politeknikoan AURKEZTUA

HONAKO TITULU HAU LORTZEKO: Ingenieritza Informatikoa

IKASTURTEA: 2023/2024

EGILEA: Martxel Aranzadi Olazabal

ZUZENDARIA: Javier Arkauz

TUTOREA: Eneko Arza

PROIEKTUA EGIN DEN ENPRESA/ERAKUNDEA:

☒ Lanaren egileak, Mondragon Unibertsitateko Goi Eskola Politeknikoari, doan eta ikerketa eta irakaskuntza helburu soilekin, dokumentu honen erreprodukzio eta komunikazio publikorako eskubideak baimentzen dizkio, baldin eta: jatorrizko egilea aipatzen bada, eta obraren erabilera komertziala ez bada.

☒ **Aitorpena – EzKomertziala – PartekatuBerdin (by-nc-sa):** ez da onartzen jatorrizko obraren erabilera komertzialik. Obra deribatuaren banaketa jatorrizko obra arautzen duen lizentzia berdinarekin egin behar da.

ORIGINALTASUN ADIERAZPENA

Ni Martxel Aranzadi Olazabalek

Adierazten dut Gradu Bukaerako Lan hau originala dela, nire lan pertsonalaren emaitza, eta ez dela aldez aurretik aurkeztu beste titulu edo kalifikazio profesional bat lortzeko. Besteen iturrietatik hartutako ideiak, formulazioak, irudiak eta ilustrazioak behar bezala aipatu eta erreferentziatu dira.

Laburpena

Proiektuaren helburua web-aplikazio bat sortzea da, informazio-iturri digitalen kudeaketa hobetzeko. Erabiltzaileek enpresaren softwareko kontu pertsonalean iturriak gehitzeko gaitasuna dute. Aldi berean, administrazio-panel bat garatu da, enpresako kudeatzaileek erabiltzaileek gehitutako iturriak berrikusi eta ebaluatzeko aukera izan dezaten. Panel horrek iturriak kategorizatzeko eta etiketatzeko prozesua errazten du, ondoren iturria katalogora atxikitzeko. Proiektuaren emaitzen hartean, funtsezkoenak datu-base bat inplementatzea, administrazio-panel funtzional bat garatzea eta iturrien katalogoa software nagusian eraginkortasunez integratzea dira. Laburbilduz, proiektuak informazio-iturri digitalak kudeatzeko irtenbide integral eta eraginkorra eskaintzen du, erabiltzailearen esperientzia eta enpresaren produktibitatea hobetuz.

Hitz gakoak: Iturri digitala, web-aplikazioa, administrazio-panela, kategorizazioa, etiketatzea, katalogoa.

Garapen Jasangarriko Helburuetan (GJH) eragina: 8 - Lan duina eta hazkunde ekonomikoa, 9 – Industria, berrikuntza eta azpiegitura, 12 – Ekoizpen eta kontsumo arduratsua.

Resumen

El proyecto se enfoca en la creación de una aplicación web destinada a mejorar la gestión de fuentes de información digital. Por una parte, los usuarios tienen la capacidad de añadir fuentes digitales en su cuenta personal para tener un seguimiento de ellas. Paralelamente, con el panel administrativo que se ha desarrollado, los gestores de la empresa podrán revisar y evaluar las fuentes añadidas por los usuarios. Este panel facilita el proceso de categorización y etiquetado de las fuentes para que luego se pueda añadir al catálogo y asi crear un catálogo organizado y útil para el usuario. Los resultados del proyecto incluyen la implementación de una base de datos sólida y segura, el desarrollo de un panel administrativo funcional y la integración del catálogo de fuentes en el software principal. En resumen, el proyecto ofrece una solución integral y eficiente para a gestión de fuentes de información digital, mejorando la experiencia del usuario y la productividad empresarial.

Palabras claves: Fuente digital, aplicación web, panel administrativo, categorización, etiquetado, catálogo.

Impacto en los Objetivos de Desarrollo Sostenible (ODS): 8 - Trabajo decente y crecimiento económico, 9 – Industria, innovación e infraestructura, 12 – Producción y consumo responsables.

Abstract

The project focuses on the creation of a web application aimed at improving the management of digital information sources. On the one hand, users have the ability to add digital sources to their personal account in order to keep track of them. At the same time, with the administrative panel that has been developed, company managers will be able to review and evaluate the sources added by users. This panel facilitates the process of categorising and tagging the sources so that they can be added to the catalogue and thus create an organised and useful catalogue for the user. The results of the project include the implementation of a robust and secure database, the development of a functional administrative panel and the integration of the source catalogue into the main software. In summary, the project provides a comprehensive and efficient solution for managing digital information sources, improving user experience and business productivity.

Keywords: Digital information sources, web application, administrative panel, categorising, tagging, catalogue.

Sustainable Development Goals (SDG) impact: 8 - Decent work and economic growth, 9 – Industry, innovation, and infrastructure, 12 – Responsible consumption and production.

Eskerrak

Nire eskerrik beroena adierazi nahi diot INNGUMAri, Gradu Amaierako Lan hau egiteko aukera eskaini didalako.

Imobach Ramirez, Joseba Karrikiri eta Aitor Fernandez nabarmendu eta eskertu nahi ditut, proiektuaren garapen osoan zehar izan dudan laguntzagatik. Arlo honetan duten ezagutza zabala, pazientzia eta dedikazio aparta funtsezkoak izan dira lanaren etapa bakoitzean. Plangintza eta diseinu fasetik inplementazio eta hedapeneraino, gida ezinbestekoa izan da.

Gainera, IDEKOko IKT departamentu osoari eskerrak eman nahi dizkiot elkarlaneko giroagatik eta proiektu hau behar bezala burutzeko beharrezkoak diren baliabideak emateagatik. Bertako departamentuan egon naizelako lanean.

Edukien aurkibidea

Irudien aurkibidea viii
Taulen aurkibidea ix
Sinboloak eta laburdurak x
1. Sarrera 1
1.1 Arazoak 1
1.2 Aurrekariak 2
1.3 Artearen egoera 3
1.4 Helburuak 3
1.4.1 Helburu teknikoak 3
1.4.2 Ikaskuntza-helburuak 4
1.4.3 Eragin soziala 4
1.4.4 Segurtasuna 4
1.4.5 Ingurumen-jasangarritasuna 4
1.4.6 Ikuspegi ekonomiko eta industriala 5
1.5 Proiektuaren plangintza 5
1.5.1 Analisi- eta diseinu-fasea 5
1.5.2 Administrazio-panelaren garapena 5
1.5.3 Katalogo zentralizatuaren integrazioa 5
1.5.4 APIaren garapena 5
1.5.5 Katalogo eleanitzaren ezarpena 5
1.5.6 Proiektuan zehar egin diren bilerak 6
1.6 Baldintza plegua 6
1.6.1 Garapenerako baldintza teknikoak 6
2. Garapena 8
2.1 Analisia 8
2.1.1 Frameworkaren aukeraketa: Next.js 8
2.1.2 Datu-basea hautatzea: MySQL MyISAM Motorrarekin 9
2.1.3 APIrako teknologien hautaketa 9
2.1.4 Katalogoaren Atalaren Garapenaren Analisia Software Nagusian 10
2.2 Datu basearen garapena 10
2.2.1 Datu basearen taulak: 10
2.2.2 Datu basearen indizeak 12
2.3 Iturrien administrazio panelaren garapena 13
2.3.1 Panelaren atalak 13
2.3.2 Panelaren funtzionalitateak 15
2.3.3 Panelaren diseinua 15
2.3.4 Garapen teknikoa 16
2.4 APIaren garapena 20
2.4.1 HTTP eskaera metodoak 21
2.4.2 Jatorrizko iturrien endpointak 21
2.4.3 Katalogoko iturrien endpointak 24
2.4.4 Kategorien endpointak 27

2.4.5 Etiketen endpointak 32
2.4.6 Apiaren segurtasuna 37
2.4.7 Apiaren hedapena 37
2.5 Katalogoaren Atalaren Garapena Software Nagusian 38
2.5.1 MVC Frameworka 38
2.5.2 Frontend-eko aldaketak 38
2.5.3 Backend-eko aldaketak 39
2.6 Hizkuntza eleanitzaren garapenak 39
2.6.1 Datu basearen tradukzioen taula berria (jos_idkreader_translations) 39
3. Emaitzak 42
3.1 Sistemaren egitura 42
3.2 Administrazio-paneleko emaitzak 43
3.2.1 Iturri guztien atala 43
3.2.2 Katalogoko iturrien atala 46
3.2.3 Kategorien atala 48
3.2.4 Etiketen atala 50
4. Memoria ekonomikoa 52
4.1 Erabilitako ekipoaren kostua 52
4.2 Softwarearen kostua 52
4.3 Langileen kostua 52
4.4 Koste totala 53
4.5 Inbertsioaren errentagarritasuna eta itzulera 53
5. Ondorioak 54
5.1 Alderdi teknikoak 54
5.2 Alderdi metodologikoak 54
5.3 Alderdi ekonomikoak 54
5.4 Eragina Garapen Jasangarriko Helburuetan (GJH) 55
5.4.1 Industria, Berrikuntza eta Azpiegitura (9. helburua) 55
5.4.2 Lan Duina eta Hazkunde Ekonomikoa (8. helburua) 55
5.4.3 Ekoizpen eta Kontsumo Arduratsua (12. helburua) 55
6. Etorkizuneko ildoak 57
6.1 Administrazio-panela handitzea 57
6.2 Kategorizazioaren automatizazioa eta hobekuntza 57
6.3 Datuak aztertzeko tresnak integratzea 57
6.4 Sistema eleanitzaren hedapena 57
7. Balorazio pertsonala 59
8. Bibliografia 60

Irudien aurkibidea

Irudia 1: Gantt diagrama 7
Irudia 2: Datu basearen arkitektura 13
Irudia 3: Panelaren diseinua 17
Irudia 4: Karpeten banaketa 18
Irudia 5: Datu base finalaren arkitektura 41
Irudia 6: Sistemaren arkitektura 43
Irudia 7: Iturri guztien pantaila 44
Irudia 8: Informazio gehiago opzioaren pantaila 45
Irudia 9: Iturria katalogora gehitzearen pantaila 45
Irudia 10: Tradukzioak gehitzeko pantaila 46
Irudia 11: Katalogoko iturrien pantaila 47
Irudia 12: Iturria editatzearen pantaila 48
Irudia 13: Konfirmazioko mezuaren pantaila 49
Irudia 14: Kategorien atalaren pantaila 49
Irudia 15: Kategoria berria sortzearen pantaila 50
Irudia 16: Konfirmazio mezuaren pantaila 51
Irudia 17: Etiketen atalaren pantaila 51
Irudia 18: Etiketa berri bat sortzearen pantaila 52
Irudia 19: Etiketa berri bati tradukzioak gehitzearen pantaila 52
Irudia 20: Garapen jasangarriko helburuak 57

Taulen aurkibidea

Taula 1 Erabilitako ekipoen kostua 52
Taula 2 Langileen kostua 53

Sinboloak eta laburdurak

GJH	Garapen Jasangarriko helburuak
GBL	Gradu Bukaerako Lana
CI	Competitive Inteligence
API	Application Programming Interfaces
SSG	Static Site Generation
SSR	Server Side Rendering
SEO	Search Engine Optimization
SVG	Scalable Vector Graphics
AWS	Amazon Web Services
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JWT	Jason Web Token

1. Sarrera

INNGUMA IDEKO[1] zentro teknologikoak sortutako spin-off teknologikoa da, eta DANOBATGROUPen parte da[2]. INNGUMAren softwarea 10 urte baino gehiagoz egon da merkatuan, baina 2019an enpresa gisa merkaturatu zen bere merkatu kuota zabaltzeko. Gaur egun, INNGUMAk presentzia du Espainian eta Latinoamerikan, eta 90 bezero baino gehiago ditu, besteak beste, RENFE, CAF, Orona, Mondragon Corporation, Eroski, Angulas Aguinaga eta Metro Madrid enpresa liderrak. Adimen Lehiakorreko eta Zaintza Teknologikoko softwarearen erreferente nazional gisa, INNGUMAk informazioa hainbat iturritatik atzematea erraztuko duten tresnak garatzen ditu, hala nola webguneetatik, sare sozialetatik eta datu-baseetatik[3].

Dokumentu hau INNGUMArako web-aplikazio berri bat garatzeko proiektuaren memoria da, informazio-iturri digitalen kudeaketa hobetzeko. Bertan, helburuak, erabilitako metodologiak, lortutako emaitzak eta garapen-prozesuan lortutako ondorioak zehazten dira.

Proiektuak informazio digitala kudeatzeko irtenbide eraginkorrago eta zentralizatuago baten beharrari heltzen dio, INNGUMAren Adimen Lehiakorreko eta Zaintza Teknologikoko tresna aurreratuak emateko misioarekin bat eginez. Memoria honetan egindako lanak barne hartzen ditu beharren analisia, aplikazioaren diseinua eta inplementazioa, bai eta aplikazioaren funtzionaltasunaren eta eraginkortasunaren ebaluazioa ere. Proiektuaren atalak deskribatzen dira, datu-base sendo bat sortzea, iturriak berrikusteko eta kategorizatzeko administrazio-panel bat garatzea eta iturrien katalogoa enpresaren software nagusian integratzea barne.

Proiektu honen garrantzia informazio digitalaren kudeaketa eta irisgarritasuna nabarmen hobetzean datza, antolamendu eraginkorragoa eta kategorizatutako eta etiketatutako iturrietarako sarbide zentralizatua ahalbidetuz. Hobekuntza horrek erabiltzailearen esperientzia optimizatzeaz gain, enpresaren produktibitatea ere areagotzen du, informazio garrantzitsua eta ondo antolatua eskuratzea errazten baitu.

Proiektu honen ekarpen nagusien artean daude informazio-iturriak kudeatzeko irtenbide teknologiko berritzaile bat, kategorizazio- eta etiketatze-tresna eraginkorrak ezartzea eta katalogo irisgarri eta zentralizatu bat sortzea.

1.1 Arazoak

Proiektu honen arazoa INNGUMAk Adimen Lehiakorreko eta Zaintza Teknologikoko softwarea hobetzeko duen beharretik sortzen da, iturriak kudeatzeko sistema eraginkorragoa eta eskuragarriagoa ezarriz. INNGUMAren softwarea erreferentea izan da merkatuan, eta informazioa hainbat iturritatik atzitzea erraztu du, hala nola webguneetatik, sare sozialetatik eta datu-baseetatik. Hala ere, ez zuen iturriak kudeatzeko sistema zentralizaturik, ezta informazioa hobeto antolatu eta erabiltzeko aukera emango zuen web interfaze grafikorik ere.

INNGUMA softwarearen erabiltzaileek informazio-iturriak gehitu zitzaketen beren kontuetara, baina iturri horiek ez zeuden sistematikoki kategorizatuta edo etiketatuta. Antolaketarik ezak zaildu egiten zuen informazio garrantzitsua kudeatzea eta eskuratzea, eraginkortasun operatiboa eta erabaki informatuak hartzeko gaitasuna murriztuz.

Arazo hori konpontzeko, iturrien katalogo bat sortu behar da, iturriak kudeatzeko sistema oso batekin batera, web interfaze grafikoaren mailan. Sistema horri esker, enpresako kudeatzaileek erabiltzaileek gehitutako iturriak berrikusi, kategorizatu eta etiketatu ahal izango dituzte. Antolatutako iturriak katalogo zentralizatu batean sartuko dira, erabiltzaile guztientzat eskuragarri, eta informazio garrantzitsua bilatzea eta hautatzea erraztuko dute.

1.2 Aurrekariak

INNGUMA softwareari esker, enpresek zaintza teknologikoa egin dezakete, eta informazioa lortu erabiltzaileak pertsonaliza ditzakeen hainbat iturritatik. Erabiltzaileek zehaztu dezakete zer informazio den garrantzitsua eta baliagarria haientzat. Inguruneari buruz informatuta egoteak aukera ematen die enpresei erabaki garrantzitsuak hartzeko, informazio egiaztatuan oinarrituta, eta horrek areagotu egiten du erabaki zuzen eta estrategikoak hartzeko probabilitatea.

Aplikazioa oso erabilgarria denez, merkaturatzea erabaki zen. Merkatuan izan zuen arrakasta ikusi ondoren, eta IDEKOk ezin zuenez eguneratzeak eta aplikazioaren garapena kudeatzen jarraitu, INNGUMA enpresa independente gisa sortu zen rol horiek bere gain hartzeko. INNGUMA gaur egun garapen berriez eta aplikazioaren mantentzeaz arduratzen da.

Gaur egun, INNGUMAk lantalde txikia baina eraginkorra du. Talde horretan sartzen dira marketineko espezialistak, produktuaren banaketaren eta komunikazioaren arduradunak, software-garatzaileak, programa mantentzeaz eta hobetzeaz arduratzen direnak. Tamaina txikikoa izan arren, INNGUMAk bezero-oinarri zabala du nazioan eta nazioartean, bereziki Latinoamerikan, eta Kutxabank, Orona eta Soraluce bezalako enpresak ditu bezeroen artean. Aplikazioaren aldakortasunari eta erabilgarritasunari esker, hainbat bezerok konfiantza izan dute haien zerbitzuetan.

Proiektu hau hasi aurretik, INNGUMAren katalogoa softwarearen zati integrala zen. Katalogo horrek aukera ematen zien erabiltzaileei beren kontu pertsonaletan informazio-iturriak gehitzeko, erabiltzaile bakoitzarentzat garrantzitsuenak diren iturrien jarraipen pertsonalizatua eta antolatua egiteko. Iturriak pertsonalizatzeko gaitasunak aukera ematen zien erabiltzaileei informazioa beren behar espezifikoetara egokitzeko, zaintza teknologikoaren eraginkortasuna eta efizientzia hobetuz.

Katalogoak informazioa biltzeko plataforma bat eskaintzeaz gain, iturrien antolaketa eta kategorizazioa ere errazten zuen. Erabiltzaileek deskribapenak eta etiketak gehi ziezazkieketen iturriei, eta horrek lagundu egiten zuen nabigagarritasuna eta katalogoaren barruko bilaketa hobetzen. Funtzionaltasun hori funtsezkoa izan zen informazioa kudeatzeko irtenbide sendoa eta pertsonalizagarria behar zuten enpresek softwarea har zezaten. Egungo proiektuaren ikuspegia gaitasun horiek hobetzea eta zabaltzea izan da, katalogoa INNGUMAren erabiltzaile guztientzat tresna baliotsua eta eraginkorra izaten jarraituko duela ziurtatuz.

1.3 Artearen egoera

Gaur egun, informazio digitalaren kudeaketa eta adimen lehiakorra etengabe garatzen ari diren eremuak dira, enpresek lehiakor eta ondo informatuta mantentzeko duten beharrak bultzatuta. Adimen lehiakorrak (CI Competitive Intelligence) hainbat iturritako informazioa biltzea eta sistematikoki aztertzea eskatzen du, erabaki estrategikoak hartzen laguntzeko. Sheila Wright-ek dioen moduan[4], CIak merkatuan aldaketak aurreratzeko eta erabaki informatuagoak hartzeko aukera ematen die enpresei.

Zaintza teknologikoko eta informazioaren kudeaketako softwareak bilakaera izan du funtzionalitate aurreratuak sartzeko, hala nola datu-bilketa automatizatzea, datuak denbora errealean aztertzea eta informazioa kategorizatzea. Inoreader[5], Digimind[6] eta Feedly[7] tresnek soluzio sendoak eskaintzen dituzte informazio digitala kudeatzeko, eta enpresei informazio kritikoa biltzeko, aztertzeko eta partekatzeko aukera ematen diete.

Hala ere, dauden soluzio askok optimizatzeko borrokatzen duten alderdi kritiko bat web interfaze grafikoa eta iturriak modu zentralizatuan kudeatzeko gaitasuna dira. CI plataforma baten eraginkortasuna, neurri handi batean, datu-bolumen handiak antolatzeko eta sailkatzeko gaitasunaren araberakoa da, azken erabiltzaileentzat eskuragarri eta erabilgarri izan dadin. Gartnerren arabera[8], erabiltzaile interfaze intuitibo bat eta beste enpresa plataforma batzuekin integratzeko gaitasuna funtsezko ezaugarriak dira informazioa kudeatzeko edozein softwareren erabilgarritasuna maximizatzeko.

INNGUMAk, adimen lehiakorreko softwarearen eta zaintza teknologikoaren merkatu nazionaleko liderra den aldetik, behar kritiko hori aitortu du. Enpresak hainbat iturritatik informazioa jasotzeko tresna aurreratuak garatu ditu, baina iturriak kudeatzeko sistema hobetzeko beharra identifikatu zuen. Iturriak kudeatzeko sistema oso batek, katalogo zentralizatu batekin eta web interfaze grafiko intuitibo batekin, informazioaren antolaketa hobetzeaz gain, haren erabilerraztasuna eta erabilgarritasuna ere hobetzen ditu erabaki estrategikoak hartzeko.

Proiektu honetan hartutako metodologiak barne hartzen ditu web garapeneko teknologia modernoen erabilera, datu base sendoak eta erabiltzailea ardatz duten diseinu ikuspegiak. Iturriak kategorizatzeko eta etiketatzeko administrazio-panel bat ezartzea eta INNGUMAren software nagusian katalogo irisgarri bat txertatzea bat datoz informazio digitala kudeatzeko jardunbide onenekin.

1.4 Helburuak

Proiektuaren helburu nagusia INNGUMArako web-aplikazio bat garatzea da, informazio-iturri digitalen kudeaketa eraginkorra eta zentralizatua ahalbidetzeko. Helburu hori MARTE/SMART[9] printzipioen arabera ezartzen da, hau da, argia, neurgarria, lorgarria, erronkaria, denboralizatua eta espezifikoa izan behar du. Gainera, gizarte-inpaktuari, segurtasunari eta osasunari, ingurumen-iraunkortasunari eta alderdi ekonomiko eta industrialei lotutako elementuekin lerrokatuko da.

1.4.1 Helburu teknikoak

Lehenengo helburu espezifikoa kanpoko administrazio-panel bat garatzea da, INNGUMAko kudeatzaileek erabiltzaileek gehitutako iturriak berrikusi, kategorizatu eta etiketatu ahal izan ditzaten.

Administrazio-panel horrek iturrien antolaketa hobetuko du, informazioaren kudeaketa zentralizatua eta eraginkorra ahalbidetuz.

Bigarren helburua datu-base sendo eta eraginkor bat sortzea da, informazio-iturriak administratzeko. Datu-base horrek datu-bolumen handiak erabiltzeko gai izan behar du, biltegiratutako informaziorako sarbide azkar eta segurua ahalbidetuz.

Hirugarren helburua da lehendik dagoen iturrien katalogo zentralizatua datu-base berritik zuzenean balioak lortzen dituen katalogo batekin ordezkatzea, katalogo dinamikoa eginez. Horri esker, erabiltzaileek informazio garrantzitsua azkar eta eraginkortasunez bilatu eta hautatuko dute, informazioaren eskuragarritasuna eta erabilgarritasuna nabarmen hobetuz.

Aipatutako helburuez gain, API bat garatu beharko da, administrazio-panelaren eta datu-basearen arteko komunikazioa errazteko. API horrek datuen integrazio arina eta efizientea ahalbidetu beharko du, erabiltzaileek eta sistemaren kudeatzaileek egindako eragiketetan koherentzia eta segurtasuna bermatuz.

Gainera katalogo eleanitz bat ezarri nahi da, administrazio-panelean etiketei, kategoriei eta iturriei itzulpenak gehitzeko aukera emanez. Horrela, katalogoa eskuragarria eta erabilgarria izango da ikus-entzule gehiagorentzat, eta hainbat hizkuntzatako erabiltzaileek errazago erabili ahal izango dute.

1.4.2 Ikaskuntza-helburuak

Ikaskuntza-helburuei dagokienez, proiektuaren helburua web-garapeneko, datu-baseetako eta informazioaren kudeaketako gaitasunak indartzea da. Garatzailea metodologia arinen aplikazioaz, teknologia modernoen erabileraz eta erabiltzailea ardatz duen diseinuaren ikuspegiaz baliatuko da.

Helburua da garatzaileak trebetasun aurreratuak eskuratzea web-teknologiak eta datu-baseak erabiltzen, bai eta interfaze grafiko intuitiboak inplementatzen ere. Gainera, proiektuak aukera emango du software-ingeniaritzaren esparruan funtsezkoak diren proiektuak kudeatzeko trebetasunak hobetzeko.

1.4.3 Eragin soziala

Proiektuak erabiltzaileentzako informazio digitalaren irisgarritasuna eta antolamendua hobetuko ditu, erabaki informatuak eta estrategikoak hartzea erraztuz, eta horrek eragin positiboa izango du gizartean, enpresa-ingurune eraginkorragoa eta lehiakorragoa sustatuko baitu.

1.4.4 Segurtasuna

Sistema osoan datuen babesa eta konfidentzialtasuna bermatzeko segurtasun-neurri integralak ezartzea izango da lehentasunezko helburua. Horrek barne hartuko du Flask sistemarekin Pythonen garatutako APIaren sekurizazioa, bai eta administrazio-panela eta datu-basea ere. Segurtasun-teknikak erabiliko dira, hala nola datuak zifratzea, erabiltzaileak autentifikatzea, roletan oinarritutako sarbide-kontrola eta datuak baliozkotzea, sisteman dagoen informazioa babesteko.

1.4.5 Ingurumen-jasangarritasuna

Proiektuak jasangarritasuna sustatuko du, informazioa kudeatzeko baliabide fisikoen beharra murriztuz, baliabide digitalen erabilera eraginkorragoa sustatuz eta ingurumen-inpaktua murrizten lagunduz.

1.4.6 Ikuspegi ekonomiko eta industriala

Ikuspegi ekonomikotik, proiektuak enpresen produktibitatea handituko du, informazioaren kudeaketa optimizatuta. Horri esker, INNGUMAk merkatuan duen posizioa sendotu eta bezeroen kuota handitu ahal izango du, kostu operatiboetan aurreztuz, efizientzia hobetuz eta informazio-iturrien kudeaketan erredundantziak murriztuz.

1.5 Proiektuaren plangintza

INNGUMAren iturriak kudeatzeko web-aplikazio bat garatzeko proiektuaren plangintza hainbat fasetan antolatzen da, eta jarduera espezifikoak, mugarriak eta entregagarriak argi eta garbi zehaztuta daude. Proiektuan zehar, aldizkako bilerak egiten dira, aurrerapenaren komunikazio eraginkorra eta etengabeko jarraipena bermatzeko.

1.5.1 Analisi- eta diseinu-fasea

Proiektuaren lehen fasea eskakizunak eta helburuak zehaztean oinarritzen da. Etapa horretan, bilerak egiten dira stakeholder-ekin, haien beharrak eta itxaropenak ulertzeko. Datu-basea diseinatu eta garapenerako programazio-lengoaia egokiena hautatzen da. Fase hau amaitzeko, administrazio-panela diseinatu da, garapenaren hurrengo etapetarako oinarri gisa balioko duena.

1.5.2 Administrazio-panelaren garapena

Bigarren fasean, datu-basea sortu eta administrazio-panela garatzen da. Etapa honetan, iturriak kudeatzeko beharrezkoak diren funtzionalitateen inplementazioa sartzen da. Proben prozesua fase honen erdian hasten da eta amaierara arte jarraitzen du, sistemaren kalitatea eta funtzionaltasuna bermatzeko.

1.5.3 Katalogo zentralizatuaren integrazioa

Hurrengo etapan, katalogo zentralizatu bat integratu behar da, balioak datu-base berritik zuzenean ateratzeko. Katalogoaren diseinua gutxi aldatuko den arren, backend-ean aldaketa esanguratsuak egingo dira datu-basearekin konektatuta dagoela ziurtatzeko. Dagoen katalogoaren egitura eta funtzionaltasuna zehatz-mehatz aztertuko dira, eta egokitu egingo da datuen arkitektura berria islatzeko.

1.5.4 APIaren garapena

Administrazio-panela eta katalogo zentralizatua amaitu ondoren, API bat ezarri nahi da, administrazio-panelak datu-baserako sarbidea izan dezan iturri-kudeaketa mota guztiak egiteko. APIa sendoa eta segurua izateko diseinatuko da, sistemaren osagaien arteko komunikazio eraginkorra erraztuz eta eragiketen datuen osotasuna eta eraginkortasuna bermatuz.

1.5.5 Katalogo eleanitzaren ezarpena

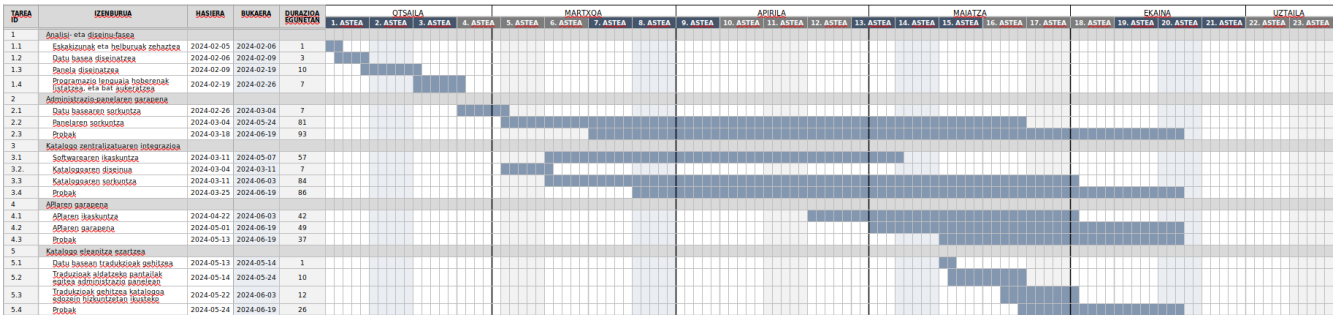
Azkenik, katalogo eleanitza ezarri nahi da. Fase honek barne hartzen ditu datu-basean itzulpenak gehitzea, itzulpen horiek kudeatzeko administrazio-panelean interfazeak garatzea eta funtzionalitate horiek integratzea, katalogoa hizkuntza askotan eskuragarri egon dadin. Itzulpenak katalogoan behar bezala erakusten direla ziurtatzeko probak egiten dira.

1.5.6 Proiektuan zehar egin diren bilerak

Asteazkenero bilera bat egiten da garatzaileekin, aurrerapena berrikusteko, arazo teknikoak konpontzeko eta zereginak planifikatzeko. Bilera horiei esker, garapen-taldeak sinkronizatu eta proiektuaren aurrerapenaren ikuspegi argia izan dezake, izan daitezkeen oztopoak garaiz identifikatuz eta lehentasunak behar bezala egokituz.

Horrez gain, bilera bat egiten da proiektuko parte-hartzaile guztiekin, garatzaileekin eta salmenta-taldearekin barne. Bilera horiek ikuspegi zabalagoa dute, eta azken erabiltzailearen eta merkatuaren ikuspuntuak lortzea ahalbidetzen dute. Salmenta-taldeak ideiak eman ditzake funtzionalitate berriek bezeroen beharrak hobeto asetzeko eta produktua merkatuan kokatzen laguntzeko.

Ondoren ageri den Gantt diagramak proiektuaren plangintza zehatza erakusten du. Kronograma honek proiektuaren fase desberdinak nabarmentzen ditu, hasierako analisi eta diseinutik hasi eta katalogo eleaniztuna ezarri arte. Fase bakoitza jarduera espezifikoetan banatuta dago, bakoitzaren iraupenarekin, eta aurrerapenaren ikuspegi argia ahalbidetzen du. Plangintzak etapa bakoitzeko garapenaren kalitatea bermatzeko proba-aldiak ere barne hartzen ditu.



Irudia 1: Gantt diagrama

1.6 Baldintza plegua

Atal honetan, proiektua garatzeko beharrezkoak diren baldintzak zehazten dira, eta betekizun teknikoak eta gauzatze-zehaztapenak aurkezten dira.

1.6.1 Garapenerako baldintza teknikoak

Proeiktua behar bezala egiteko, hainbat software eta ekipo erabili dira. Jarraian, ekipoaren zehaztapenak eta beharrezko software-tresnak deskribatzen dira:

1.6.1.1 Ekipoaren zehaztapenak

Garapen osoa ekipo bakar bat erabiliz egin da, Acer Nitro portatil bat, ezaugarri hauekin:

Prozesadorea: Intel Core i7-8750H CPU 2.20 GHz-era

RAM memoria: 24 GB (23.8 GB erabilgarri)

Pantaila: 1920x1080ko bereizmena

Sistema eragilea: Ubuntu 22.04.4 LTS

Biltegiratzea: 120 GB-ko SSD diskoa eta 930 GB-ko HDD diskoa

Grafika: NVIDIA Corporation GP107M

1.6.1.2 Softwarearen zehaztapenak

Proiektua aurrera eramateko, softwareko hainbat aplikazio eta tresna erabili dira. Hona hemen aplikazio eta tresna horiek:

Visual Studio Code: Proiektuaren kodea definitzeko erabiltzen den testu-editorea, bai erabiltzailearen aldean, bai zerbitzariaren aldean.

GitLab: Web aplikazio bat, interfaze grafiko batekin, garatzaileei fitxategiak modu zentralizatuan partekatzeko eta eguneratzeko aukera ematen diena.

Docker: Software-edukiontzien hedapena automatizatzen duen sistema eragilea, lan-ingurunea tokian altxatzeko erabiltzen dena.

Microsoft Teams: Bideo-deiak eta txatak konbinatzen dituen plataforma bat, INNGUMAko langileen arteko komunikazioa erraztuz.

PhpMyAdmin: PHPn oinarritutako tresna bat, web nabigatzailearen bidez MySQLn datuak erabiltzeko, funtsezkoa INNGUMA aplikatzeko behar diren datuak gorde eta manipulatzeko.

DataGrip eta DBeaver: Datu-baseak administratzeko tresna aurreratuak, datu-egiturak kudeatzeko eta optimizatzeko erabiltzen direnak.

Postman: APIak administratzeko tresna bat, proiektuaren garapenean proba eta kudeaketa erraztuz.

1.6.1.3 Aplikazioa gauzatzeko baldintzak

INNGUMAren erabiltzaileek erraz erabili ahal izango dituzte aplikazioa eta agente berria. Exekuzioa modu hauetan egin daiteke:

Edozein ordenagailuren bidez, eramangarria zein mahai gainekoa izan, erabiltzaileak aukeratutako web nabigatzaile bat erabiliz (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera edo Safari).

Gailu mugikor edo tablet bat erabiliz eta web nabigatzaile bat erabiliz, ordenagailu eramangarri batean edo mahai gaineko ordenagailu batean bezala.

2. Garapena

2.1 Analisia

INNGUMArako iturriak kudeatzeko web-aplikazioa garatzean, funtsezko hainbat erabaki tekniko hartu dira, proiektuaren behar espezifikoetan oinarrituta. Jarraian, erabilitako framework-a, datu-basea eta biltegiratze-motorra hautatzeko arrazoiak zehazten dira.

2.1.1 Frameworkaren aukeraketa: Next.js

Web aplikazioa garatzeko, Next.js[10] aukeratu da framework nagusi gisa. Aukera hori egiteko arrazoiak askotarikoak dira, eta ezaugarri aurreratuetan eta errendimendu handiko aplikazioak erabiltzeko gaitasunean oinarritzen dira:

Errenderizatu Hibridoa eta Estatikoa: Next.js sistemak zerbitzariaren aldea errenderizatzeko (SSR) eta orri estatikoak sortzeko (SSG) aukera ematen du. Horrek malgutasuna eta errendimendu optimizatua ematen ditu, orriak azkar kargatzeko eta erabiltzailearen esperientzia hobetzeko aukera emanez.

Jarduera eta SEO: Next.js-ek SSR maneiatzeko duen gaitasunak nabarmen hobetzen du SEOa, bilaketa-motorrek edukia eraginkorrago indexatu baitezakete. Gainera, renderizazio estatikoak karga-denborak hobetzen ditu, funtsezkoa baita erabiltzaileen erabilgarritasunerako eta atxikipenerako.

Garatzeko erraztasuna eta eskalagarritasuna: Next.js-ek egitura argia eta osagai berrerabilgarriak eskaintzen ditu, eta horrek proiektua garatzea eta mantentzea errazten du. React sistemarekin integratzeak erabiltzaile-interfaze modernoak eta dinamikoak modu eraginkorrean eraikitzea ahalbidetzen du.

Bateragarritasuna eta Ekosistema: Next.js tresna eta zerbitzu anitzekin integratzen da, autentifikazio-sistemak, datu-baseak eta hedapen-zerbitzuak barne. Horrek funtzionalitate berrien inplementazioa eta proiektuaren etorkizuneko eskalagarritasuna errazten ditu.

2.1.1.1 Beste frameworkekin alderatzea

- Angular: Angularrek[11] eskala handiko aplikazioak garatzeko ingurune sendoa eskaintzen duen arren, bere ikaskuntza-kurba nabarmenagoa da eta bere egitura zurrunagoa da Next.js-en aldean. Gainera, errendimenduari dagokionez, Angular astunagoa izan daiteke. Angular indartsua den bitartean, bere errendimendua SSR terminoetan ez da Next.js-ek eskaintzen duen eraginkortasun mailara iristen.

- Vue.js: Vue.js[12] erabilerraztasunagatik eta errendimenduagatik da ezaguna. Hala ere, Next.js-ek euskarri hobea eskaintzen dio SSRri eta integrazio natiboagoa Reactekin, eta hori hobeto lerrokatzen da proiektuaren beharrekin. Vue.js, moldakorra bada ere, ez da iristen Next.js markaren SSR errendimendu-maila berera, eta, beraz, ez da hain egokia edukia azkar kargatu eta eguneratu behar duten aplikazioetarako.

2.1.2 Datu-basea hautatzea: MySQL MyISAM Motorrarekin

Aplikazioan datuak eta iturriak biltegiratzeko eta kudeatzeko, MySQL[13] aukeratu da datu-baseak kudeatzeko sistema gisa, MyISAM[14] biltegiratze-motorra erabilita. Hauek dira aukeratzeko arrazoiak:

Errendimendua: MyISAM irakurketa-eragiketetan errendimendu handia izateagatik da ezaguna, eta hori erabakigarria da iturrietako datu-kopuru handiak kudeatu behar direnean. Aplikazioak datuak azkar eta eraginkortasunez kontsultatu behar dituenez, motor hau egokia da gure beharretarako.

Sinpletasuna eta abiadura: MyISAM sinpleagoa da egituran eta administrazioan, InnoDB bezalako beste motor batzuekin alderatuta. Hautaketa-eragiketetan duen abiadurak eta transakzioen gainkargarik ezak ezin hobea egiten dute lehentasuna datuak azkar irakurtzea duten aplikazioetarako.

Euskarria eta komunitatea: MySQL gehien erabiltzen den datu-baseetako bat da, eta euskarri komunitario zabala du. Horrek arazoak konpontzea eta beste garatzaile batzuen esperientzian oinarritutako jardunbide egokiak ezartzea errazten du.

Datuen Administrazioaren malgutasuna: MySQL MyISAMekin erabiltzeak datuen eta eskemen administrazio malgua ahalbidetzen du, hazkundea eta proiektuaren premia berrietara egokitzea erraztuz, errendimendua arriskuan jarri gabe.

2.1.2.1 Datu-baseen beste sistema batzuekiko alderaketa

PostgreSQL: PostgreSQL-ek[15] ezaugarri aurreratuak eta oso fidagarriak eskaintzen baditu ere, konfigurazioa eta administrazioa konplexuagoak izan daitezke. MySQL-ek, MyISAMekin, errendimendu handiagoa eskaintzen du irakurketa-eragiketetan, eta hori lehentasuna da gure proiektuarentzat. PostgreSQL bikaina da datuen osotasuna eta transakzio konplexuak behar dituzten aplikazioetarako, baina MySQL egokiagoa da kontsulta azkar eta eraginkorrak egiteko.

MongoDB: NoSQL[16] datu-base gisa, MongoDB[17] primerakoa da egituratu gabeko datuak gordetzeko. Hala ere, proiektuaren beharrak hobeto lerrokatzen dira MySQL bezalako sistema erlazional batekin, erabiltzen ditugun datuen izaera egituratua dela eta. MongoDB egokiagoa litzateke egituratu gabeko datu-bolumen handiak eta aldakorrak erabiltzen dituzten aplikazioetarako, baina gure irakurketa intentsiboko eragiketetarako eta ondo definitutako datu-egituretarako, MySQL da aukerarik onena.

Erabaki tekniko horiek ziurtatzen dute aplikazioa sendoa eta eskalagarria izango dela, eta datuen karga esanguratsua maneiatzeko gai izango dela, INNGUMAri adimen lehiakorreko eta zaintza teknologikoko softwarearen merkatuan lider izaten jarraitzeko aukera emanez.

2.1.3 APIrako teknologien hautaketa

APIaren garapen-prozesuan eragina du aldez aurretik Flask[18] liburutegia erabiliz Pythonen inplementazio bat egoteak. Muga horrek hizkuntza eta garapen-esparrua hautatzeko dauden aukerak murrizten baditu ere, abantaila nabarmenak ditu, oinarri sendoa ematen baitu hasteko. Flask eta Python sistemekin jarraitzean, dagoen azpiegitura aprobetxatzen da, eta horrek inplementazioa sinplifikatzen du eta lehendik dagoen kodearekiko koherentzia errazten du. Python lengoaiarekin eta Flask sistemaren egiturarekin ohitzeko ikaskuntza-aldia beharrezkoa den arren, ezarritako oinarri teknikoak garapen

eraginkorragoa lortzeko eta Inngumaren sistema nagusiarekin arazorik gabe integratzeko bidea errazten du.

2.1.4 Katalogoaren Atalaren Garapenaren Analisia Software Nagusian

INNGUMAren software nagusian katalogoaren atala hobetzeko, lehendik dagoen azpiegitura teknologikoaren barruan lan egitea hautatu da. APIarekin bezala, erabaki hori sisteman dagoeneko ezarrita dauden teknologietara egokitzeko beharrean oinarritzen da. Horrek esan nahi du PHP[19] erabili behar dela Joomlaren backend eta Bootstrap-en framework CSSarekin, horiek baitira softwarean erabiltzen diren teknologiak. Aukera horrek lehendik zeuden azpiegitura eta teknologiak ulertzeko ikaskuntza-prozesu bat ekarri badu ere, katalogoa garatzeko oinarri sendoa eta ezarria baliatzeko abantaila ere eskaintzen du. Dagoen softwarearen funtzionamendua eta egitura ulertzea izan da zailtasun nagusia, baina teknologia horien barruan lan egiteak integrazio arina eta efizientzia hobetua ematen du katalogoaren garapenean.

2.2 Datu basearen garapena

Datu basearen aukeraketa egin ondoren, datu-basearen egitura diseinatu da, beharrezko taulak eta erlazioak definituz. Hainbat taula nagusi sortu dira:

2.2.1 Datu basearen taulak:

2.2.1.1 Katalogoaren iturrien taula (jos_idkreader_catalog_sources):

Taula honetan honako eremu hauek sartzen dira:

- id: Katalogoaren iturriaren identifikatzailea.
- real_source_id: Katalogoari gehitu zaion iturriaren identifikatzailea. Horrela, iturriaren ezaugarriak lor daitezke, hala nola irudia, URLa, etab.
- catalog_category_id: Iturria zein kategoriari gehitu zaion. Katalogoko iturri batek kategoria bati erreferentzia egin behar dio beti.
- title: Katalogoan agertuko den iturriaren izenburua. Jatorrizko iturrikoaren desberdina izan daiteke.
- language: Iturriko hizkuntza.
- description: Katalogoan agertuko den deskribapena. Jatorrizko iturriarenaren desberdina izan daiteke.

2.2.1.2 Katalogoaren kategorien taula (jos_idkreader_catalog_categories):

Taula honetan honako eremu hauek sartzen dira:

- id: Kategoriaren identifikatzailea.
- parent: Aita kategoriako identifikatzailea, halakorik badago.

- name: Kategoriaren izena.
- description: Kategoriaren deskribapena.
- icon: Kategoriari lotutako ikonoa.

2.2.1.3 Katalogoko etiketen taula (jos_idkreader_catalog_tags):

Taula honetan honako eremu hauek sartzen dira:

- id: Etiketaren identifikatzailea.
- name: Etiketaren izena.

2.2.1.4 Etiketen eta katalogoko iturrien arteko erlazioen taula (jos_idkreader_catalog_tag_sources):

Taula honetan honako eremu hauek sartzen dira:

- catalog_tag_id: Etiketaren identifikatzailea.
- catalog_source_id: Katalogoaren iturriaren identifikatzailea.

Taula horiek informazioa modu eraginkorrean atzitzeko eta antolatzeko diseinatu dira, sisteman datuak erraz kudeatzeko eta berreskuratzeko.

Egitura zehaztu ondoren, datu-basea inplementatu da MySQL erabiliz. Prozesu horretan, aldez aurretik ezarritako diseinuaren arabera sortu dira taulak, eta bakoitzak azterketa-fasean identifikatutako baldintzak betetzen dituela ziurtatu da. Ondoren, taulak datuekin populatu dira, hasierako probak egiteko eta oinarrizko funtzionaltasuna egiaztatzeko.

Datu-basearen arkitektura irudikatzeko, taulen egitura eta haien arteko erlazioak erakusten dituen irudi bat erantsi da. Irudi horrek datu-basearen irudikapen bisuala ematen du, haren diseinua eta funtzionamendua ulertzen lagunduz.

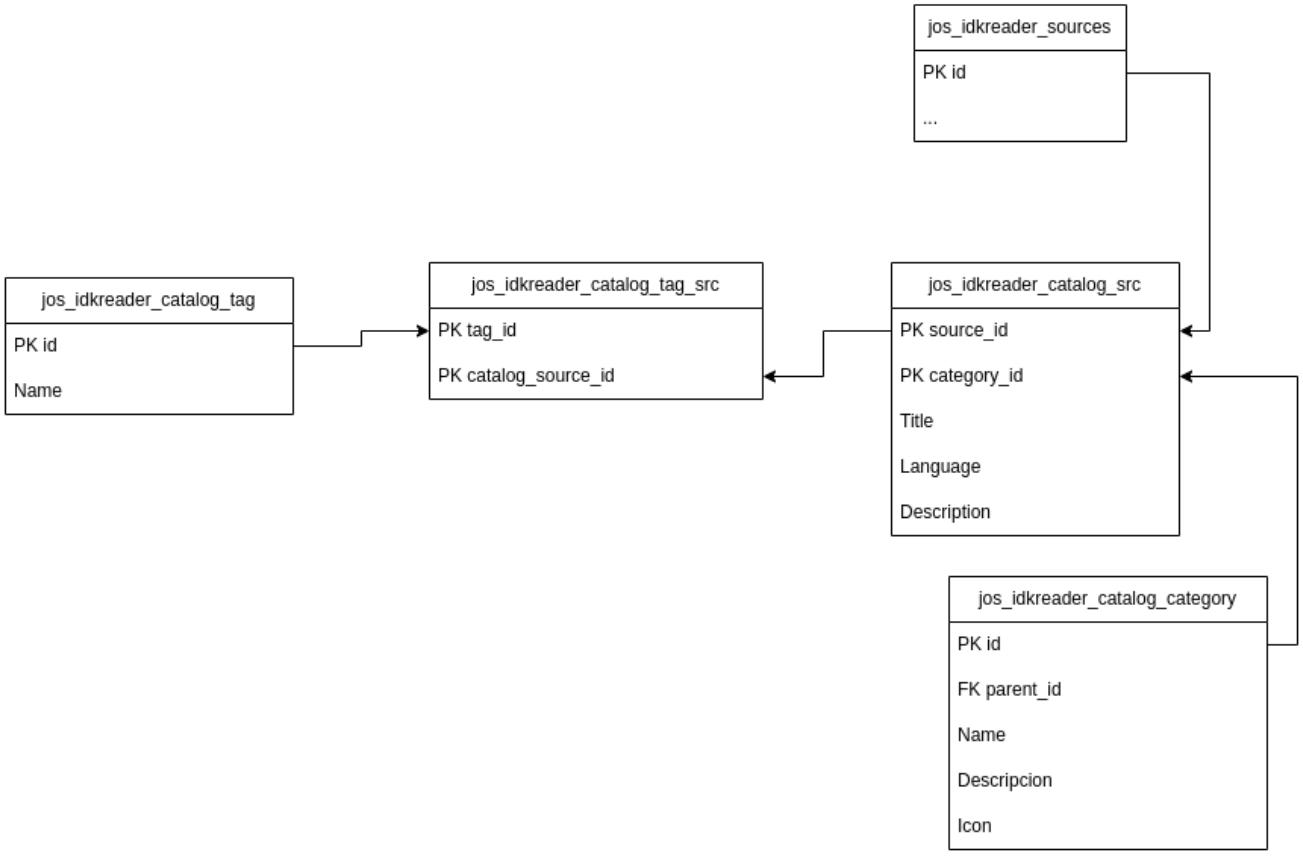


Figure 3: Datu basearen arkitektura

2.2.2 Datu basearen indizeak

Datu-basea sortzean, kontsulten eraginkortasuna kontuan hartu da, hainbat taulatan datu asko erabiliko baitira. Errendimendu optimoa bermatzeko, indizeak sortu dira tauletako hainbat funtsezko eremutan.

Datu-baseetako indizeak egitura gehigarriak dira, taula bateko eremu baten edo gehiagoren gainean sortzen direnak. Bere funtzio nagusia eremu horietan balio espezifikoak bilatzen dituzten kontsultak azkartzea da. Indizeek aukera ematen dute datu-baseak erregistroak azkar aurkitzeko, taula osoa eskaneatu behar izan gabe, eta hori bereziki onuragarria da taula handietan.

Hona hemen kontsulten eraginkortasuna hobetzeko sortutako indizeak:

2.2.2.1 Katalogoaren iturrien taula (jos_idkreader_catalog_sources):

Indize bat sortu da real_source_id eremuan, benetako identifikatzailean oinarritutako iturri espezifikoak bilatzen dituzten kontsultak bizkortzeko.

Title eremuan ere indize bat sortu da, izenburu bidezko bilaketen azkartasuna hobetzeko.

2.2.2.2 Katalogoko kategorien taula (jos_idkreader_catalog_categories):

Indize bat sortu da name eremuan, kategoria-izenaren araberako bilaketen eraginkortasuna hobetzeko.

2.2.2.3 Katalogoko etiketen taula (jos_idkreader_catalog_tags):

Indize bat sortu da name eremuan, etiketa-izenaren araberako bilaketen eraginkortasuna hobetzeko.

2.2.2.4 Etiketen eta katalogoko iturrien arteko erlazioen taula (jos_idkreader_catalog_tag_sources):

Indize konbinatuak sortu dira catalog_tag_id eta catalog_source_id eremuetan, etiketak iturriekin lotzen dituzten kontsultak bizkortzeko.

Indize horiek kontsulten erantzun-denbora murrizten laguntzen dute, erabiltzaileak katalogoko datuen kudeaketan eta sarbidean duen esperientzia eta efizientzia hobetuz.

2.3 Iturrien administrazio panelaren garapena

Administrazio-panela garatzeko, administratzaileekin komunikazio estua izan da, haien behar eta eskakizun espezifikoak ulertzeko.

2.3.1 Panelaren atalak

Elkarrizketa horietan, identifikatu da panelak funtsezko lau atal izan behar dituela.

2.3.1.1 Erabiltzaileek gehitutako iturrien atala:

Atal honen helburua erabiltzaileek plataformari gehitzen dizkioten iturri guztiak zerrendatzea da taula baten laguntzarekin. Funtsezkoa da zerrenda hori denbora errealean eguneratzea, administratzaileei iturri berriak berehala monitorizatzeko eta berrikusteko aukera emanez.

Taulak zutabe hauek ditu:

- ID: Iturriaren identifikatzaile bakarra.
- Izenburua: Iturriaren izenburua.
- Sortze data: iturria sortu zen data.
- Azken bilketaren data: Iturritik informazioa bildu zen azken aldia.
- Mota: Iturri mota da.
- Egoera: iturria aktibo, arriskuan edo erorita dagoen adierazten du.
- Aukerak: Azken zutabean, botoi bat sakatzean, aukera zerrenda bat zabaltzen da, iturria katalogoan eransteko edo iturriari buruzko informazio gehiago ikusteko.

2.3.1.2 Katalogoko iturrien atala:

Atal honetan, katalogoa osatzen duten iturri guztiak erakutsiko dira taula batean. Zerrendan iturri bakoitzari buruzko xehetasun garrantzitsuak jasoko dira, hala nola izenburua, kategoria, hizkuntza eta deskribapena. Horrek erraztu egingo die administratzaileei katalogatutako iturriak kudeatzea eta eguneratzea.

Taulak zutabe hauek ditu:

- Izenburua: Iturriaren izenburua katalogoan.
- Kategoria: Iturria zein kategoriatakoa den.
- Etiketak: Iturriari lotutako etiketak.
- Egoera: Iturriaren egoera adierazten du.
- Hizkuntza: Iturriko hizkuntza.
- Deskribapena: Iturriaren deskribapen laburra.
- Aukerak: Azken zutabean, iturria editatzeko edo katalogotik ezabatzeko aukera ematen duen botoia.

2.3.1.3 Kategorien atala:

Atal honek sisteman eskuragarri dauden kategoria guztien zerrenda osoa aurkeztuko du taula baten laguntzarekin. Kategorien hierarkia ere kudeatu ahal izango da, gurasoen eta seme-alaben kategoriak esleituz, dagokionaren arabera.

Taulak zutabe hauek ditu:

- Izena: Kategoriaren izena.
- Kategoria Aita: Kategoria nagusia (hala badagokio).
- Deskribapena: Kategoriaren deskribapen laburra.
- Aukerak: kategoria editatzeko edo ezabatzeko botoiak barne.

2.3.1.4 Etiketen atala:

Kategorien atalaren antzera, panelaren zati honek sortu diren etiketa guztiak zerrendatuko ditu taulak.

Taulak zutabe hauek ditu:

- Izena: Etiketaren izena.
- Aukerak: Etiketa editatzeko edo ezabatzeko botoiak.

2.3.2 Panelaren funtzionalitateak

Horrez gain, funtzionalitate espezifikoak ezarriko dira administrazio-panelaren barruan, iturrien kudeaketa eta sailkapenak errazteko:

2.3.2.1 Iturria katalogora gehitzea:

Administratzaileek iturri berriak gehitu ahal izango dizkiote katalogoari zuzenean paneletik. Prozesu horretan iturri berriei dagozkien kategoriak eta etiketak esleituko dira, eta behar bezala antolatuta daudela ziurtatuko da. Horrez gain, titulua eta deskripzioa aldatzeko aukera izango dute, katalogoan eurek nahi bezala ager dadin.

2.3.2.2 Iturria katalogotik editatu edo kentzea:

Administratzaileek katalogoko iturri bat editatu edo kendu ahal izango dute.

2.3.2.3 Kategoria berria sortzea:

Kategoria berriak sortzeko aukerari esker, administratzaileek katalogoa modu dinamikoan zabaldu eta antolatu ahal izango dute, behar eta informazio mota berrietara egokituz. Kategoria sortzean, aita kategoria bat eransteko aukera izango dute. Aita kategoriarik esleitzen ez bazaio, kategoria hori aita izango da, baina kontuan izan behar da ezin izango zaiola iturririk gehitu.

2.3.2.4 Kategoria editatu edo ezabatzea:

Administratzaileek kategoria bat editatu edo ezabatu ahal izango dute.

2.3.2.5 Etiketa berria sortzea:

Kategoria bat sor daitekeen bezala, etiketek aukera bera dute. Baina etiketa bat sortzeko beste modu bat ere badago: katalogoaren iturri bat editatzen denean edo katalogoari iturri bat gehitzen zaionean, etiketak eransteko aukera ematen dizu, hemen datu-basean ez dagoen etiketa bat gehitzen baduzu, sistemak automatikoki bat sortuko du izen horrekin.

2.3.2.6 Etiketa editatu edo ezabatzea:

Administratzaileek etiketa editatu ahal izango dute, eta etiketa bat jada garrantzitsua edo beharrezkoa ez bada, administratzaileek etiketa ezabatu ahal izango dute. Prozesu horrek bermatuko du etiketa erabilgarri eta egokiak bakarrik mantenduko direla sisteman, beharrezkoa ez den informazioa metatzea saihestuz.

2.3.3 Panelaren diseinua

Administrazio-panela osatuko duten atalak eta izan beharko dituen funtzionaltasunak ezagutu ondoren, panelaren egitura diseinatzeari ekin zaio. Diseinu bisual honek sekzioak eta funtzioak nola antolatu eta aurkeztuko diren kontzeptualizatzen laguntzen du, administratzaileek errazago garatu eta erabil ditzaten.

Jarraian, administrazio-panelaren diseinuaren irudi bat erantsi da. Irudi honek sekzioen, nabigazio-botoien eta lan-eremu nagusien kokapena erakusten du, ezarriko den interfazearen ikuspegi argia emanez. Atal guztiak taula batekin irudikatzea erabaki da.



Figure 5: Panelaren diseinua

2.3.4 Garapen teknikoa

Administrazio-panela garatzeko, Next.js erabiltzea erabaki da, funtzionaltasun aurreratuak dituzten web-aplikazioak sortzea errazten duen React framework bat. Next.js-en, funtsezkoa da funtsezko bi kontzeptu ulertzea: karpeta bidezko bideratzea eta zerbitzariaren edo bezeroaren aldeko ekintzak.

2.3.4.1 Karpeta bidezko bideratzea

Next.js-en, bideratze-sistema proiektuaren karpeta-egituran oinarritzen da. Pages karpetaren barruko fitxategi bakoitza aplikazioaren ibilbide bihurtzen da automatikoki. Adibidez, pages/index.ts fitxategi bat ibilbideari dagokio /, eta pages/catalog.ts fitxategi bat, berriz, /catalog ibilbideari dagokio. Konbentzio horri esker, kodea intuizioz antola daiteke, eta administrazio-paneleko atalen arteko nabigazioa errazten du.

Proiektuaren egitura erakusteko, karpetak Next.js aplikazioaren barruan nola banatu diren erakusten duen irudi bat erantsi da. Egitura horrek administrazio-panelaren atalak islatzen ditu, eta errazago ulertzen da nola erabiltzen den aplikazioa.

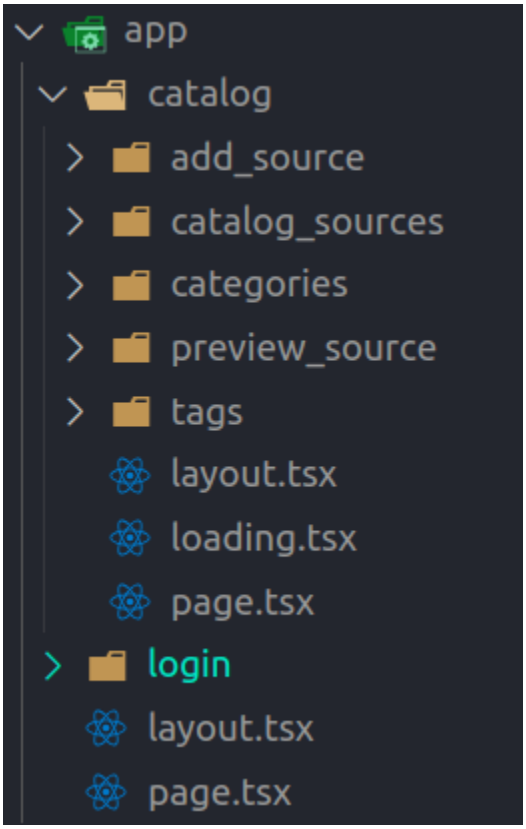


Figure 7: Karpeten banaketa

2.3.4.2 Server-Side eta Client-Side Actions

Next.js-ek zerbitzariaren aldean zein bezeroaren aldean ekintzak erabiltzeko aukera ematen du, erabiltzailearen errendimendua eta esperientzia optimizatzeko malgutasuna emanez. Zerbitzariaren aldeko ekintzek (Server-Side Rendering edo SSR) zerbitzarian eduki dinamikoa sortzea ahalbidetzen dute bezeroari bidali aurretik, eta hori baliagarria da maiz aldatzen diren eta hasierako karga azkarra eskatzen duten datuetarako. Bestalde, bezeroaren aldeko ekintzak (Client-Side Rendering edo CSR) egokiak dira interakzio azkarretarako eta denbora errealean eguneratzeko, orria kargatu beharrik gabe.

2.3.4.3 Proiektuko beste karpeta batzuk

App-karpetaz gain, aplikazioaren egituran eta funtzionamenduan funtsezko rolak betetzen dituzten beste karpeta batzuk daude:

- Components: Karpeta honek egitura modular eta eraginkor bat mantentzen laguntzen duten osagai berrerabilgarri guztiak ditu. Osagai horien adibideak dira aplikazioaren hainbat zatitan erabiltzen diren taulak, formularioak eta botoiak.

- Helpers: Hemen gordetzen dira APIrekin elkarreragiten duten ekintza guztiak. Funtzio horiek datu-basetik datuak lortu, sortu, eguneratu edo ezabatzeko beharrezkoak diren HTTP eskaerak egiteaz arduratzen dira.

- Lib: Karpeta honetan aplikazio osoan erabiltzen diren konstanteak, motak eta utilitateak definitzen dira. Horren barruan sartzen dira konfigurazio orokorrak, funtzio osagarriak eta kodea mantentzeko koherentzia eta erraztasuna bermatzen duten datu motak.

2.3.4.4 Panelaren segurtasuna

Administrazio-panelaren segurtasuna bermatzeko eta baimendutako kudeatzaileei soilik sarbidea mugatzeko, autentifikazio-sistema bat ezarri da, NextAuth funtzionalitatea eta ibilbideak babesteko middlewarea erabiliz.

- NextAuth configurazioa

NextAuth Next.js aplikazioetan autentifikazioa inplementatzea errazten duen liburutegia da. Proiektu honetan, NextAuth konfiguratu da saioaren hasiera eta erabiltzaileen autentifikazioa kudeatzeko. Prozesua NextAuth konfiguratuz hasten da, kredentzialen hornitzaile bat onartzeko. Horri esker, kudeatzaileek saioa has dezakete erabiltzailea eta pasahitza erabiliz. Kredentzialak mugatuta daude erabiltzailea posta elektroniko bat izan dadin, pasahitzak bestalde gutxienez 6 karaktere izan behar ditu, honekin, egon daitezkeen sarrera maltzurrak sanetizatu ditugu.

- Autentifikazio-prozesua

Kudeatzaile bat administrazio-panelean sartzen saiatzen denean, saio-hasierako orri bat aurkezten zaio. Hemen, zure erabiltzaile-izena eta pasahitza sartu behar dituzu. NextAuth-ek baliozkotu egiten ditu kredentzial horiek datu-basearen aurka; erabiltzailea kontrastatua bada, administratzailearentzako sarbidea duen begiratzen da. Kredentzialak zuzenak badira eta erabiltzaileak sarbidea badu, modu seguruan biltegiratzen den erabiltzaile-saio bat sortzen da, panelean sartzeko aukera emanez.

- Middleware bidezko ibilbideak babestea

Erabiltzailea autentifikatuta dagoenean, funtsezkoa da administrazio-paneleko ibilbide sentikorrak babestea. Hori lortzeko, middleware erabiltzen da, eskaerak maneiatzeko prozesuan gauzatzen diren funtzioen multzoa. Middlewareak erabiltzaileak saio aktiborik duen egiaztatzen du, ibilbide babestuetara sartzen utzi aurretik. Erabiltzailea autentifikatuta ez badago, sesioa hastera berbideratuko zaio.

- Ezarritako segurtasunaren abantailak

Datuen babesa: Administrazio-panelerako sarbidea erabiltzaile baimenduei mugatzeak informazio sentikorra babesten du, eta ziurtatzen du baimendutako langileek bakarrik egin ahal izango dituztela aldaketak sisteman.

Segurtasun zentralizatua: NextAuth-ek saioen autentifikazioa eta kudeaketa maneiatzen ditu, aplikazioan segurtasuna inplementatzearen eta mantentzearen konplexutasuna murriztuz.

Malgutasuna: NextAuth-ek autentifikazio-hornitzaile ugari (GitHub, Google, Facebook) integratzeko eta baimen-logika proiektuaren behar espezifikoen arabera pertsonalizatzeko aukera ematen du. Proiektu honetarako ez da izan beharrezkoa holako autentifikazio sistema ezartzea, erabiltzaile gutxik edukiko dutelako panelerako sarbidea.

2.3.4.5 Panelaren garapenean erabilitako liburutegiak

Administrazio-panela garatzeko, hainbat liburutegi erabili dira, panelaren funtzionaltasuna eta itxura errazago ezartzeko eta hobetzeko. Jarraian, erabilitako liburu-denda nagusiak eta proiektuan nola lagundu duten deskribatzen da:

- Tanstack taula

Datuak tauletan bistaratzeko eta manipulatzeko, @tanstack/react-table liburutegia erabili da. Liburutegi honek React aplikazioetan taulak maneiatzeko soluzio malgu eta indartsua eskaintzen du, ezaugarri aurreratuen inplementazioa ahalbidetuz, hala nola:

  - Iragaztea: Iragazteko aukerak gehitu dira, erabiltzaileek taulen barruan datu espezifikoak bilatu eta bistaratu ahal izan ditzaten.
  - Antolamendua: Tauletako datuak erraz antola daitezke irizpide desberdinen arabera, informazioaren erabilgarritasuna eta eskuragarritasuna hobetuz.
  - Pertsonalizazioa: Liburu-dendak taulen itxura eta portaera pertsonalizatzeko aukera ematen du, administrazio-panelaren behar espezifikoetara egokitu daitezen.

- Tailwind CSS

Erabiltzaile-interfazea diseinatzeko eta estilizatzeko, Tailwind CSS[20] erabili da, diseinu oso pertsonalizagarria eta eraginkorra ahalbidetzen duen CSS framework erabilgarria. Proiektu honetan Tailwind CSS erabiltzearen abantaila nagusiak honako hauek dira:

  - Estilo sendoak: Tailwindek aurrez definitutako erabilgarritasun-klaseak eskaintzen ditu, aplikazio osoaren diseinua bermatzen dutenak.
  - Garapen azkarra: Erabilgarritasun-klaseak erabiltzean, berehala aplika dakizkieke estiloak osagaiei, CSS pertsonalizatua idatzi beharrik gabe.
  - Pertsonalizazioa: Tailwindek aukera ematen du aurrez zehaztutako estiloak erraz pertsonalizatzeko konfigurazio-fitxategi bat konfiguratuz, diseinua administrazio-panelaren behar espezifikoetara egokituz.

- Heroicons

Panelaren interfazean erabiltzen diren ikonoetarako, heroicons[21] aukeratu da. Liburutegi honek kalitate handiko SVG ikono sorta zabala eskaintzen du, integratzen eta pertsonalizatzen errazak direnak.

Heroiconsen ikonoak erabiltzailearen esperientzia eta panel barruko nabigazioa hobetzeko erabili dira, estetika moderno eta argia emanez.

2.3.4.6 Sistemaren hedapena

Administrazio-panelaren hedapen eraginkorra eta eskalagarria bermatzeko, Docker[22] edukiontziak erabiltzea erabaki da. Docker aplikazio bat eta bere gelak edukiontzi batean paketatzeko aukera ematen duen plataforma bat da, eta horrek bermatzen du aplikazioa uniformeki exekutatzen dela edozein ingurunetan.

- Docker erabiltzearen abantailak
  - Eramangarritasuna: Dockerrek aukera ematen du administrazio-panela Docker euskarria duen edozein ingurutan exekutatzeko, aplikazioak garapen, proba eta ekoizpenean modu berean funtzionatzen duela ziurtatuz.
  - Isolamendua: Docker edukiontzi bakoitza ingurune isolatu batean exekutatzen da, eta horrek mendekotasunen arteko gatazkak saihesten ditu eta aplikazio ugari host berean exekutatzea ahalbidetzen du, interferentziarik gabe.
  - Eskalagarritasuna: Dockerrek aplikazioaren eskalagarritasuna errazten du, edukiontziaren instantzia anitzen sorreraren bidez, lan-karga handitzeko.
  - Inplementatzeko erraztasuna: Edukiontziak erraz kudeatu eta zabal daitezke orkestrazio tresnen bidez, hala nola Docker Compose, eta horrek etengabeko hedapen prozesua errazten du.
- Hedapen-prozesua
  - Dockerfile sortu: Dockerfile fitxategi bat sortu da, administrazio-panelerako edukiontziaren irudia nola eraiki zehazten duena. Artxibo honetan, bulegoak instalatzeko eta ingurunea konfiguratzeko behar diren jarraibide guztiak jasotzen dira.
  - Docker irudia eraiki: Dockerfilea erabiliz, administrazio-panelerako Docker irudia eraiki da.
  - Edukiontzia sortzu: Behin irudia eraikita, irudi horretatik abiatuta edukiontzi bat sortu eta exekutatu da.
  - Docker Compose orkestrazioa: Hainbat zerbitzu eta edukiontzi kudeatzeko, Docker Compose erabili da. Docker-compose.yml fitxategi batek zehazten ditu beharrezko zerbitzuak eta nola konfiguratu eta exekutatu behar diren.
  - Hodeian hedatzea: Azkenik, produkzioan hedatzeko, hodeiko zerbitzuen hornitzaile bat erabili da, AWS gisa.

2.4 APIaren garapena

Atal honetan, APIan ezarritako endpointak deskribatzen dira. Endpointak eskaerak jaso eta bakoitzean dauden eragiketak definitzen dituzten ibilbide espezifikoak dira.

2.4.1 HTTP eskaera metodoak

Endpoint bakoitzerako, HTTP eskaera desberdinak ezarri dira[23], egin nahi den ekintza adierazteko:

- GET: Datu edo baliabide espezifikoak lortzeko erabiltzen da.
- POST: Baliabide berri bat sortzeko erabilia.
- PUT: Dauden baliabideak eguneratzeko erabiltzen da.
- DELETE: Baliabide bat ezabatzeko erabiltzen da.

2.4.1.1 HTTP eskaeren erantzunak

APIaren erantzunek, egindako eskaeraren emaitza adierazten dute, eta HTTP egoera-kode desberdinekin adierazten dira. Hona hemen gehien erabiltzen diren erantzunak:

- 200: Eskaera arrakastatsua izan da eta eskatutako baliabidea itzuli da.
- 201: Baliabide bat sortzeko eskaera arrakastatsua izan da eta baliabidea sortu egin da.
- 204: Eskaera arrakastatsua izan da, baina ez da edukirik itzultzen.
- 400: Eskaera ez da baliozkoa edo ezin izan zen prozesatu bezeroaren akats baten ondorioz.
- 404: Eskatutako baliabidea ez dago.
- 500: Errorea gertatu da zerbitzarian.

Informazio mota desberdinak maneiatzeko lau karpetatan banatu dira: jatorrizko iturriak, katalogoaren iturriak, kategoriak eta etiketak.

2.4.2 Jatorrizko iturrien endpointak

Atal honetan, APIan jatorrizko iturriak kudeatzeko inplementatutako endpointak deskribatzen dira. Ondoren, sortutako endpointak, horien metodoak eta egiten dituzten eragiketak zehazten dira:

2.4.2.1 Lortu jatorrizko iturri bat IDa erabiliz

- Endpoint:/original_source/< int: original_source_id>
- Deskripzioa: Jatorrizko iturri bat bueltatzen du emandako IDaren arabera.
- Sarrera-parametroak: Nahi den iturriaren identifikatzailea
- Metodoa: GET

- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta iturria aurkitzen bada, hau bueltatua izango da JSON formatuan 200eko egoera-kodearekin
  - 204 egoera-kodea:
    - Iturria ezin izan bada aurkitu, 204ko egoera-kodea bueltatuko du.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.2.2 Jatorrizko iturrien zerrenda iragazia lortu

- Endpoint: /original_sources_list
- Deskripzioa: Jatorrizko iturrietatik iragazitako zerrenda bat itzultzen du, iragazkien eta emandako orrialdearen arabera.
- Sarrera-parametroak:
  - page_size (int): Orriaren tamaina.
  - current_page (int): Oraingo orrialdearen zenbakia.
  - column_filters (list): Zutabeei aplikatutako iragazkien zerrenda.
  - sorting (list): Antolamendu-irizpideen zerrenda.
- Metodoa: POST
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta iragazkiekin bat datozen jatorrizko iturriak aurkitzen badira, APIak 200 egoera-kodea eta jatorrizko iturrien zerrenda iragazita duen JSON bat bueltatuko du.
  - 204 egoera-kodea:
    - Iragazkiekin bat datozen jatorrizko iturririk aurkitzen ez bada, APIak 204 No Content egoera-kodea erantzungo du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik.

- 500 egoera-kodea:
  - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIak 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.2.3 Iturri bat gehitu katalogora

- Endpoint: /add_source_to_catalog
- Deskripzioa: Endpoint honek katalogoan iturri espezifiko bat gehitzeko aukera ematen du.
- Sarrera-parametroak:
  - source_id (int): iturriaren ID.
  - source_title (string): Iturriaren izenburua.
  - source_lang (string): Iturriaren hizkuntza.
  - source_category (int): katalogoaren kategoriaren ID.
  - source_description (string): Iturriaren deskribapena.
  - source_tags (string): Iturriari lotutako etiketak.
- Metodoa: POST
- Erantzunak:
  - 201 egoera-kodea:
    - Sortzeko eskaera behar bezala prozesatzen bada eta iturria katalogoari gehitzen bazaio, APIak 201 Created egoera-kode batekin eta katalogoari erantsitako iturriaren IDa duen JSON batekin erantzungo du.
  - 400 egoera-kodea:
    - Sarrera-parametroak baliogabeak badira edo falta badira, APIak 400 Bad Request egoera-kodea izango du. Horrek adierazten du eskaera ezin izan zela prozesatu bezeroaren akats baten ondorioz.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.2.4 Jatorrizko iturri baten item-ak lortu

- Endpoint: /original_source_items/<int:original_source_id>
- Deskribapena: Endpoint honek jatorrizko iturri bati lotutako itemak itzultzen ditu, emandako IDaren arabera.
- Sarrera-parametroak:
  - original_source_id (int): Lortu nahi diren itemak dituen jatorrizko itemaren Ida.
- Metodoa: GET
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta jatorrizko iturrirako itemak aurkitzen badira, APIak 200 egoera-kodea eta itemen xehetasunak jasotzen dituen JSON bat izango ditu.
  - 204 egoera-kodea:
    - Jatorrizko iturriari lotutako itemik ez badago emandako IDarekin, APIak 204 No Content egoera-kodea izango du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik. Kasu horretan, erantzunaren gorputza hutsik egongo da.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.3 Katalogoko iturrien endpointak

2.4.3.1 Katalogoko iturria lortu Idaren arabera

- Endpoint: /catalog_source/<int:catalog_source_id>
- Deskribapena: Endpoint honek katalogoko iturri espezifiko baten xehetasunak itzultzen ditu, emandako IDaren arabera.
- Sarrera-parametroak:
  - catalog_source_id (int): Lortu nahi den katalogoaren iturriaren Ida.
- Metodoa: GET
- Erantzunak:
  - 200 egoera-kodea:

- Eskaera behar bezala prozesatzen bada eta katalogoaren iturria aurkitzen bada, APIak 200 egoera-kode batekin eta katalogoaren iturriaren xehetasunak jasotzen dituen JSON batekin erantzungo du.

◦ 404 egoera-kodea:

- IDa duen katalogoaren iturria ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du katalogoan ez dagoela zehaztutako IDa duen iturririk.

◦ 500 egoera-kodea:

- Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.ç

2.4.3.2 Katalogoko Iturrien Zerrenda Iragazia lortutako

- Endpoint: /catalog_source_list
- Deskribapena: Endpoint honek katalogoko iturrietatik iragazitako zerrenda bat itzultzen du, iragazkien eta emandako orrialdearen arabera.
- Sarrera-parametroak:
  - page_size (int): Orriaren tamaina.
  - current_page (int): Oraingo orrialdearen zenbakia.
  - column_filters (list): Zutabeei aplikatutako iragazkien zerrenda.
  - sorting (list): Antolamendu-irizpideen zerrenda.
- Metodoa: POST
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta iragazkiekin bat datozen katalogoko iturriak aurkitzen badira, APIak 200 egoera-kode batekin eta katalogoko iturrien zerrenda iragazia duen JSON batekin erantzungo du.
  - 204 egoera-kodea:
    - Ez badago iragazkiekin bat datorren katalogoko iturririk, APIak 204 No Content egoera-kodea izango du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik. Kasu horretan, erantzunaren gorputza hutsik egongo da.

- 500 egoera-kodea:
  - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.3.3 Katalogoko iturri bat eguneratzea

- Endpoint: /update_catalog_source
- Deskribapena: Endpoint honek katalogoko iturri espezifiko baten xehetasunak eguneratzeko aukera ematen du.
- Sarrera-parametroak:
  - catalog_source_id (int): eguneratu beharreko katalogoaren iturriaren ID.
  - source_title (string): Iturriaren izenburu berria.
  - source_lang (string): Iturriaren hizkuntza berria.
  - catalog_category_id (int): Katalogoaren kategoriako ID berria.
  - source_description (string): Iturriaren deskribapen berria.
  - source_tags (string): Iturriari lotutako etiketa berriak.
- Metodoa: PUT
- Erantzunak:
  - 200 egoera-kodea:
    - Eguneratzeko eskaera behar bezala prozesatzen bada, APIak 200 egoera-kodea eta katalogoaren iturriaren xehetasun eguneratuak jasotzen dituen JSON bat izango ditu.
  - 400 egoeraren kodea:
    - Sarrera-parametroak baliogabeak badira edo falta badira, APIak 400 Bad Request egoera-kodea izango du. Horrek adierazten du eskaera ezin izan zela prozesatu bezeroaren akats baten ondorioz.
  - 404 egoera-kodea:
    - IDa duen katalogoaren iturria ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du katalogoan ez dagoela zehaztutako IDa duen iturririk.
  - 500 egoera-kodea:

- Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.3.4 Iturri bat Katalogotik ezabatu ID bidez

- Endpoint: /delete_catalog_source/<int:catalog_source_id>
- Deskribapena: Endpoint honek katalogoko iturri espezifiko bat ezabatzeko aukera ematen du, emandako IDaren arabera.
- Sarrera-parametroak:
  - catalog_source_id (int): Ezabatu nahi den katalogoaren iturriaren Ida.
- Metodoa: DELETE
- Erantzunak:
  - 200 egoera-kodea:
    - Ezabatzeko eskaera behar bezala prozesatzen bada, APIak 200 egoera-kodearekin erantzungo du eta katalogoaren iturria ezabatu dela baieztatuko du.
  - 404 egoera-kodea:
    - IDa duen katalogoaren iturria ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du katalogoan ez dagoela zehaztutako IDa duen iturririk.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.4 Kategorien endpointak

Atal honetan, APIan kategoriak kudeatzeko inplementatutako endpointak deskribatzen dira:

2.4.4.1 Kategoria bat lortzea IDaren arabera

- Endpoint: /category/<int:category_id>
- Description: Endpoint honek kategoria zehatz baten xehetasunak itzultzen ditu, emandako IDaren arabera.
- Sarrera-parametroak:
  - category_id (int): Lortu nahi den kategoriaren ID.

- Metodoa: GET
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta kategoria aurkitzen bada, APIak 200 egoera-kodea eta kategoriaren xehetasunak jasotzen dituen JSON bat izango ditu.
  - 404 egoera-kodea:
    - Emandako IDa duen kategoria ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du ez dagoela zehaztutako IDa duen kategoriarik.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.4.2 Kategoria guztiak lortu

- Endpoint: /categories
- Description: Endpoint honek sisteman erregistratutako kategoria guztien zerrenda itzultzen du.
- Sarrerako parametroak: Bat ere ez.
- Metodoa: GET
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta kategoriak aurkitzen badira, APIak 200 egoera-kode batekin eta sistemaren kategoria guztien zerrenda jasotzen duen JSON batekin erantzungo du.
  - 204 egoera-kodea:
    - Sisteman erregistratutako kategoriak aurkitzen ez badira, APIak 204 No Content egoera-kodea izango du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik. Kasu horretan, erantzunaren gorputza hutsik egongo da.
  - 500 egoera-kodea:

- Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.4.3 Aita kategoriak lortu

- Endpoint: /parent_categories
- Description: Endpoint honek azpikategoriak dituzten kategoria guztien zerrenda itzultzen du.
- Sarrerako parametroak: Bat ere ez.
- Metodoa: GET
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta aita-kategoriak aurkitzen badira, APIak 200 egoera-kode batekin erantzungo du, eta sistemaren kategoria aita guztien zerrenda jasotzen duen JSON batekin.
  - 204 egoera-kodea:
    - Sisteman erregistratutako aita-kategoriarik ez badago, APIk 204 No Content egoera-kodea izango du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik. Kasu horretan, erantzunaren gorputza hutsik egongo da.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.4.4 Kategorien zerrenda iragazia lortu

- Endpoint: /category_list
- Deskribapena: Endpoint honek kategorien zerrenda iragazi bat itzultzen du, iragazkien eta emandako orrialdearen arabera.
- Sarrera-parametroak:
  - page_size (int): Orriaren tamaina.
  - current_page (int): Oraingo orrialdearen zenbakia.
  - column_filters (list): Zutabeei aplikatutako iragazkien zerrenda.

- sorting (list): Antolamendu-irizpideen zerrenda.

- Metodoa: POST
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta iragazkiekin bat datozen kategoriak aurkitzen badira, APIak 200 egoera-kodea eta kategorien zerrenda iragazia duen JSON bat izango ditu.
  - 204 egoera-kodea:
    - Iragazkiekin bat datozen kategoriak aurkitzen ez badira, APIak 204 No Content egoera-kodea izango du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik. Kasu horretan, erantzunaren gorputza hutsik egongo da.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.4.5 Kategoria berri bat sortu

- Endpoint: /create_category
- Description: Endpoint honek sisteman kategoria berri bat sortzeko aukera ematen du.
- Sarrera-parametroak:
  - name (string): Kategoria berriaren izena.
  - parent_category_id (int, aukerakoa): aita kategoriako ID, hala badagokio, bestela 0 bat, honek kategoria aita dela esan nahiko du.
  - description (string): Kategoria berriaren deskribapena.
- Metodoa: POST
- Erantzunak:
  - 201 egoera-kodea:
    - Sortzeko eskaera behar bezala prozesatzen bada, APIak 201 egoera-kodea eta sortutako kategoria berriaren xehetasunak jasotzen dituen JSON bat izango ditu.
  - 400 egoera-kodea:

- Sarrera-parametroak baliogabeak badira edo falta badira, APIak 400 Bad Request egoera-kodea izango du. Horrek adierazten du eskaera ezin izan zela prozesatu bezeroaren akats baten ondorioz.

◦ 500 egoera-kodea:

- Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.4.6 Kategoria bat eguneratzeak

- Endpoint: /update_category
- Description: Endpoint honek kategoria zehatz baten xehetasunak eguneratzeko aukera ematen du.
- Sarrera-parametroak:
  - category_id (int): eguneratu beharreko kategoriaren ID.
  - name (string): Kategoriaren izen berria.
  - parent_category_id (int, aukerakoa): Aita kategoriako ID berria, hala badagokio.
  - description (string): Kategoriaren deskribapen berria.
- Metodoa: PUT
- Erantzunak:
  - 200 egoera-kodea:
    - Eguneratzeko eskaera behar bezala prozesatzen bada, APIak 200 egoera-kodea eta kategoriaren xehetasun eguneratuak jasotzen dituen JSON bat izango ditu.
  - 400 egoera-kodea:
    - Sarrera-parametroak baliogabeak badira edo falta badira, APIak 400 Bad Request egoera-kodea izango du. Horrek adierazten du eskaera ezin izan zela prozesatu bezeroaren akats baten ondorioz.
  - 404 egoera-kodea:
    - Emandako IDa duen kategoria ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du ez dagoela zehaztutako IDa duen kategoriarik.
  - 500 egoera-kodea:

- Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.4.7 Kategoria bat ezabatzea IDaren arabera

- Endpoint: /delete_category/<int:category_id>
- Deskribapena: Endpoint honek kategoria espezifiko bat ezabatzeko aukera ematen du, emandako IDaren arabera.
- Sarrera-parametroak:
  - category_id (int): ezabatu nahi den kategoriaren Ida.
- Metodoa: DELETE
- Erantzunak:
  - 200 egoera-kodea:
    - Ezabatzeko eskaera behar bezala prozesatzen bada, APIak 200 egoera-kodearekin erantzungo du eta kategoria ezabatu dela berretsiko du.
  - 404 egoera-kodea:
    - Emandako IDa duen kategoria ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du ez dagoela zehaztutako IDa duen kategoriarik.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIak 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.5 Etiketen endpointak

Etiketak administratzeko eta kudeatzeko endpointak.

2.4.5.1 Etiqueta bat lortzea IDaren arabera

- Endpoint: /tag/<int:tag_id>
- Deskribapena: Endpoint honek etiketa espezifiko baten xehetasunak itzultzen ditu, emandako IDaren arabera.
- Sarrera-parametroak:
  - tag_id (int): Lortu nahi den etiketaren Ida.

- Metodoa: GET
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta etiketa aurkitzen bada, APIak 200 egoera-kodea eta etiketaren xehetasunak jasotzen dituen JSON bat izango ditu.
  - 404 egoera-kodea:
    - Emandako IDa duen etiketa ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du ez dagoela zehaztutako IDa duen etiketarik.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.5.2 Etiketa guztiak eskuratzea

- Endpoint: /tags
- Description: Endpoint honek sisteman erregistratutako etiketa guztien zerrenda bat itzultzen du.
- Sarrerako parametroak: Bat ere ez.
- Metodoa: GET
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta etiketak aurkitzen badira, APIak 200 egoera-kode batekin eta sistemaren etiketa guztien zerrenda jasotzen duen JSON batekin erantzungo du.
  - 204 egoera-kodea:
    - Sisteman erregistratutako etiketarik ez badago, APIk 204 No Content egoera-kodea izango du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik. Kasu horretan, erantzunaren gorputza hutsik egongo da.
  - 500 egoera-kodea:

- Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete

2.4.5.3 Etiketen zerrenda iragazia lortu

- Endpoint: /tag_list
- Deskribapena: Endpoint honek etiketen zerrenda iragazi bat itzultzen du, iragazkien eta emandako orrialdearen arabera.
- Sarrera-parametroak:
  - page_size (int): Orriaren tamaina.
  - current_page (int): Oraingo orrialdearen zenbakia.
  - column_filters (list): Zutabeei aplikatutako iragazkien zerrenda.
  - sorting (list): Antolamendu-irizpideen zerrenda.
- Metodoa: POST
- Erantzunak:
  - 200 egoera-kodea:
    - Eskaera behar bezala prozesatzen bada eta iragazkiekin bat datozen etiketak aurkitzen badira, APIak 200 egoera-kodea eta etiketen zerrenda iragazia duen JSON bat izango ditu.
  - 204 egoera-kodea:
    - Iragazkiekin bat datozen etiketarik ez badago, APIak 204 No Content egoera-kodea izango du. Horrek adierazten du eskaera behar bezala prozesatu dela, baina ez dago erantzunik itzultzeko daturik. Kasu horretan, erantzunaren gorputza hutsik egongo da.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.5.4 Etiketa berria sortu

- Endpoint: /create_tag
- Description: Endpoint honek sisteman etiketa berri bat sortzeko aukera ematen du.
- Sarrera-parametroak:

- name (string): Etiketa berriaren izena.
- description (string): Etiketa berriaren deskribapena.

- Metodoa: POST
- Erantzunak:
  - 201 egoera-kodea:
    - Sortzeko eskaera behar bezala prozesatzen bada, APIak 201 egoera-kodea eta sortutako etiketa berriaren xehetasunak jasotzen dituen JSON bat izango ditu.
  - 400 egoeraren kodea:
    - Sarrera-parametroak baliogabeak badira edo falta badira, APIak 400 Bad Request egoera-kodea izango du. Horrek adierazten du eskaera ezin izan zela prozesatu bezeroaren akats baten ondorioz.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.5.5 Etiketa bat eguneratu

- Endpoint: /update_tag
- Deskribapena: Endpoint honek etiketa espezifiko baten xehetasunak eguneratzeko aukera ematen du.
- Sarrera-parametroak:
  - tag_id (int): eguneratu beharreko etiketaren ID.
  - name (string): Etiketaren izen berria.
  - description (string): Etiketaren deskribapen berria.
- Metodoa: PUT
- Erantzunak:
  - 200 egoera-kodea:
    - Eguneratzeko eskaera behar bezala prozesatzen bada, APIak 200 egoera-kodea eta etiketaren xehetasun eguneratuak jasotzen dituen JSON bat izango ditu.

- 400 egoera-kodea:
  - Sarrera-parametroak baliogabeak badira edo falta badira, APIak 400 Bad Request egoera-kodea izango du. Horrek adierazten du eskaera ezin izan zela prozesatu bezeroaren akats baten ondorioz.
- 404 egoera-kodea:
  - Emandako IDa duen etiketa ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du ez dagoela zehaztutako IDa duen etiketarik.
- 500 egoera-kodea:
  - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.5.6 Etiketa bat ezabatu IDaren arabera

- Endpoint: /delete_tag/<int:tag_id>
- Deskribapena: Endpoint honek etiketa espezifiko bat ezabatzeko aukera ematen du, emandako IDaren arabera.
- Sarrera-parametroak:
  - tag_id (int): Ezabatu nahi den etiketaren Ida.
- Metodoa: DELETE
- Erantzunak:
  - 200 egoera-kodea:
    - Ezabatzeko eskaera behar bezala prozesatzen bada, APIak 200 egoera-kodearekin erantzungo du eta etiketa ezabatu dela baieztatuko du.
  - 404 egoera-kodea:
    - Emandako IDa duen etiketa ez badago, APIak 404 Not Found egoera-kodea izango du. Horrek esan nahi du ez dagoela zehaztutako IDa duen etiketarik.
  - 500 egoera-kodea:
    - Eskaera prozesatzean akatsen bat gertatzen bada datu-basean izandako arazo baten edo barruko beste errore baten ondorioz, APIk 500 Internal Server Error egoera-kode batekin erantzungo du, eta horrek esan nahi du ezin izan zela eskaera behar bezala bete.

2.4.6 Apiaren segurtasuna

Segurtasuna funtsezko zutabea da gure APIa diseinatzeko eta ezartzeko orduan. Datuak babesteko eta eskaeren benetakotasuna bermatzeko, hainbat segurtasun-neurri ezarri ditugu endpoint guztietan. Atal honetan, erabiltzaileen baliozkotzea eta autentifikazioa nola egiten den zehazten da, bai eta Flask-ek datuen osotasuna eta segurtasuna bermatzeko ematen dituen segurtasun-neurri gehigarriak ere.

2.4.6.1 Erabiltzailearen balidazioa

Administrazio-paneleko backend-etik gure APIra iristen den eskaera bakoitzak autentifikazio-token bat dauka. Token hori saioa hasteko prozesuan lortzen da eta erabiltzailearen saioan biltegiratzen da. Erabiltzailearen baliozkotzea bi etapa nagusitan egiten da. Lehenik eta behin, eskaera bat jasotzean, APIak autentifikazio-tokena baliozkotzen du. Token hau JWT (JSON Web Token) bat da, sistemaren sekretu bat erabiliz egiaztatzen dena. Hasierako egiaztapen horrek bermatzen du tokena ez dela aldatu eta benetakoa dela. Tokena baliozkoa bada, APIak hurrengo balidazio-etapari ekingo dio. Bestela, APIak autentifikazio-akats batekin erantzuten du, baimenik gabeko sarbidea eragotziz.

Tokena balidatu ondoren, APIak bigarren egiaztapen bat egiten du, lotutako erabiltzaileak katalogoko panelean sartzeko baimena duen zehazteko. Segurtasun-geruza gehigarri horrek ziurtatzen du baimendutako erabiltzaileek bakarrik jardun ahal izango dutela katalogoaren kudeaketako endpointekin. Erabiltzaileak beharrezko baimenak baditu, APIak eskaera prozesatzen jarraitzen du. Bestela, eskaera gelditu egiten da eta baimen-errore bat itzultzen da. Baliozkotze-prozesu bikoitz horrek bermatzen du baimendutako eta baimendutako erabiltzaileek soilik eskuratu eta aldatu ahal izango dituztela datuak APIaren bidez, baimenik gabeko sarbideen aurkako defentsa-lerro sendo bat eskainiz.

2.4.6.2 Inputen segurtasuna

Flask, APIrako erabiltzen den frameworkak, hainbat segurtasun-neurri eskaintzen ditu jasotako datuak seguruak direla bermatzeko. Neurri horien artean inputen sanitizazioa dago, ahultasun arruntak prebenitzen laguntzen duena, hala nola SQL injekzioa eta XSS motako erasoak (Cross-Site Scripting). Inputak sanitizatzea prozesu kritikoa da, eta sarrera-datuak garbitzea eta baliozkotzea eskatzen du, datu horiek espero diren motakoak eta formatukoak direla eta kode maltzurrik ez dutela ziurtatuz.

Flask sistemak sarrerako datuak garbitzeko eta baliozkotzeko mekanismo integratuak ditu, eta horrek ziurtatzen du jasotako datuek segurtasun-estandarrak betetzen dituztela. Flask sistemaren berezko gaitasunez gain, liburutegi gehigarriak eta balidatzaile pertsonalizatuak erabiltzen ditugu gure aplikazioaren datu espezifikoak egiaztatzeko, segurtasun-geruza gehigarri bat gehituz. Balidatzaile horiek ziurtatzen dute datuak koherenteak izatea eta mehatxurik gabe egotea, horrela sistemaren osotasuna babestuz.

2.4.7 Apiaren hedapena

APIa hedatzeko, zerbitzaririk gabeko arkitektura (serverless) erabili dugu Amazon Web Servicesen[24] (AWS). Ikuspegi horrek hainbat abantaila eskaintzen ditu eskalagarritasunari, kostuei eta mantentze-lanei dagokienez.

Serverless hodeiko konputazio eredu bat da, non zerbitzuen hornitzaileak automatikoki kudeatzen duen kodea exekutatzeko behar den azpiegitura. Zerbitzari fisiko edo birtualen administrazioaz arduratu beharrean, garatzaileek kodea idatzi eta hedatu dezakete.

2.5 Katalogoaren Atalaren Garapena Software Nagusian

Software nagusiaren barruan katalogoaren atala garatzean, batez ere datuen integraziora eta kudeaketara bideratu da, Joomlan[25] backend-erako eskaerak eginez. Lan guztia zuzenean zerbitzarian egin da, Joomlaren berezko funtzionaltasuna erabiliz, kanpoko API bat sortu beharrik gabe.

2.5.1 MVC Frameworka

Joomlaren arkitektura MVC diseinu ereduan oinarritzen da[26]. Eredu honek negozio-logika (model), erabiltzaile-interfazea (view) eta aplikazioaren fluxua (controller) bereiztea errazten du.

2.5.1.1 Model: CatalogSources modeloa sortu da, katalogoko datuak nola egituratzen diren eta datu-basearekin nola elkarreragiten duten definitzen duena. Horren barruan sartzen dira datuak lortu, txertatu, eguneratu eta ezabatzeko SQL kontsultak. Modelo honen laguntzarekin, bistak behar dituen informazio eta datu gustiak bistaratu ahal izango dira.

2.5.1.2 View: Katalogoaren ikuspegia katalogoaren lehen bertsioaren datuak aurkezteko diseinatuta zegoen. Horren barruan sartzen dira txantiloiak sortzea eta Joomlaren osagaiak erabiltzea katalogoaren administrazio-orriak errenderizatzeko. Bertan, modelo berriko funtzioei deitu beharko zaie, datu base berriko informazioa lortu ahal izateko.

2.5.1.3 Controller: Katalogoaren kontrolatzailea ere garatu zen logika eta erabiltzailearen interakzioak kudeatzeko katalogoaren lehen bertsioan. Doikuntzak egin ditu funtzionalitate berriak integratzeko eta katalogoaren kudeaketa hobetzeko. Aldaketa hauen artean, bistari modeloko funtzioak pasa ahalko zaizkio.

2.5.2 Frontend-eko aldaketak

Hasiera batean backend-ean soilik zentratzea planifikatu bazen ere, aldaketa batzuk egin dira frontend-ean, erabiltzailearen esperientzia hobetzeko.

2.5.2.1 Iturrien eta kategorien deskribapenak

Katalogoan iturrietarako eta kategorietarako deskribapen zehatzak gehitu ziren. Horrek erabilgarritasuna hobetzen du eta erabiltzaileei iturri eta kategoria bakoitzaren edukia eta helburua hobeto ulertzeko aukera ematen die. Deskribapenek testuinguru gehigarria eskaintzen dute eta katalogoaren barruko nabigazioan eta hautaketan laguntzen dute.

2.5.2.2 Erabiltzailearen interfazea hobetzea

Backendaren funtzionalitate berriei eusteko, doikuntza txikiagoak egin ziren erabiltzailearen interfazean. Horren barruan sartzen da eremu berriak sartzeko formularioak eguneratzea, hala nola deskribapenak, eta iturrien eta kategorien zerrenden aurkezpen bisuala hobetzea.

2.5.3 Backend-eko aldaketak

2.5.3.1 Akatsak baliozkotzeko eta maneiatzeko hobekuntzak

Aipatutako eguneratzeez gain, hobekuntzak egin dira backend-aren akatsak baliozkotzeko eta maneiatzeko. Katalogoarekin elkarreragiten duen eskaera bakoitza zenbait balidaziotatik pasatzen da, datuen osotasuna bermatzeko. Balidazio horien barruan sartzen dira iturriak badaudela egiaztatzea, sartutako datuak baliozkoak direla egiaztatzea eta erabiltzaileak ekintza jakin batzuk egiteko baimena ematea.

2.5.3.2 Kontsultak optimizatzea

Datu-baserako kontsultak optimizatu ditu, sistemaren errendimendua hobetzeko. Horren barruan sartzen dira indizeak gako-tauletan erabiltzea eta eragiketa arruntetarako kontsulta eraginkorragoak ezartzea, hala nola iturriak katalogoan bilatzea eta iragaztea. Kontsulten optimizazioak ziurtatzen du sistemak datu-bolumen handia erabili ahal izango duela abiadura edo eraginkortasuna arriskuan jarri gabe.

2.5.3.3 Monitorizazioa eta Logak

Monitorizazio- eta logging-tresnak ezarri ziren sistemaren errendimendua gainbegiratzeko eta edozein arazo denbora errealean detektatzeko. Horren barruan sartzen dira datu-baseari egindako kontsulten monitorizazioa, APIren errendimendua eta erabiltzaileen jarduera. Logs zehatzei esker, arazoak modu eraginkorragoan identifikatu eta konpon daitezke.

2.6 Hizkuntza eleanitzaren garapenak

Garapenaren atalean, alderdi garrantzitsuenetako bat hizkuntza anitzeko funtzionaltasuna inplementatzea izan da. Funtzionalitate hori funtsezkoa da aplikazioa audientzia global batek erabili ahal izateko, erabiltzaileek sistemarekin beren hizkuntza gogokoenean elkarreragiteko aukera izan dezaten.

Itzulpenak kudeatzeko prozesua automatizatu egin da, eraginkortasuna hobetzeko eta eskuzko lana murrizteko. Hori lortzeko, taula berri bat ezarri da itzulpenak gordetzeko datu-basean. Taula honek elementua, hizkuntza, itzulitako testua eta beste metadatu garrantzitsu batzuk identifikatzeko eremuak ditu.

2.6.1 Datu basearen tradukzioen taula berria (jos_idkreader_translations)

Datu-basean taula bat gehitu zen itzulpen guztiak gordetzeko. Taularen egiturak itzulpen guztien kudeaketa zentralizatua eta automatizatua ahalbidetzen du:

- language: Eremu honetan itzulpenaren hizkuntza-kodea gordetzen da (adibidez, 'es' erderaren kasuan, 'en' inglesaren kasuan). Nahitaezko eremua da, eta aukera ematen du itzulpena zein hizkuntzatan dagoen identifikatzeko.
- element_id: Eremu honek itzulpenaren elementuaren identifikatzaile bakarra gordetzen du. Nahitaezko eremua da, eta aukera ematen du itzulpena datu-baseko elementu espezifikoarekin lotzeko.

- table_name: Eremu honek taularen izena adierazten du elementua dagoen datu-basean. Itzulpenak modu eraginkorragoan antolatzen eta bilatzen laguntzen duen nahitaezko eremua da.
- table_field: Eremu honek itzultzen ari den datu-basearen taularen barruko eremua zehazten du (adibidez, izenburuarentzako 'title', deskribapenerako 'description'). Nahitaezko eremua da, eta itzulpena eremu egokian aplikatzen dela bermatzen du.
- translation: Eremu honek itzulitako testua jasotzen du. Aplikazioan agertuko den benetako itzulpena gordetzen duen aukerako eremua da.

Taula berri horren ondoren, datu-basearen arkitektura osoa horrelakoa da:

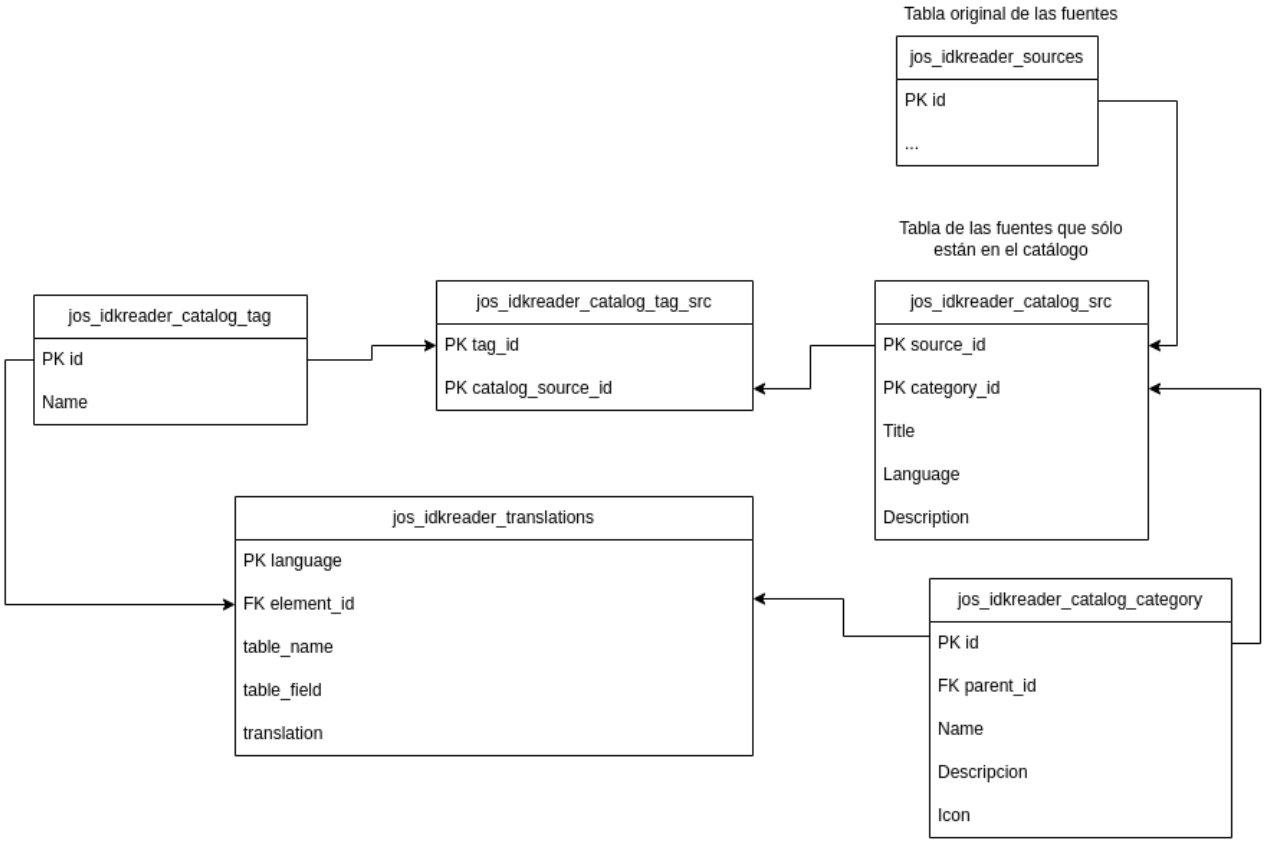


Figure 9: Datu base finalaren arkitektura

Kategoria, etiketa edo iturri berri bat gehitzen denean, itzulpenak administrazio-panel berriaren bidez kudeatzen dira. Horrek nabarmen murrizten du eskuzko lana, eta itzulpen guztiak eguneratuta eta koherenteak izatea bermatzen du.

Itzulpenak automatikoki sinkronizatzen dira software nagusiaren katalogoaren atalarekin, eguneratze guztiak erabiltzailearen interfazean berehala islatuko direla bermatuz. Elementu batek itzulpena aurkitzen ez badu, itzulita ez dagoela esan nahiko du, orduan taulak berez dakarren balioa hartuko da.

Horrela, lortzen dena da .ini fitxategiak aldatu behar ez izatea, datu guztiak, itzulpenak zein bestelakoak, datu-basetik jasoko liratekeelako. Eta MYSQLren kontsulta soil batekin nahi den itzulpena lortu ahal izango da, beharrezko eremu guztiak gehituz.

3. Emaitzak

Atal honetan, proiektuaren garapenean zehar lortutako emaitzak aurkezten dira. Sistema-egitura oso eta sendo bat ezartzeak informazio-iturri digitalen kudeaketa optimizatzea ahalbidetu du, azken erabiltzailearen esperientzia eta kudeatzaileen eraginkortasun operatiboa hobetuz. Jarraian, proiektuaren funtsezko arloetan lortutako lorpenak zehazten dira, eta osagai nagusiak eta integraziotik eta hedapenetik eratorritako onurak nabarmentzen dira.

3.1 Sistemaren egitura

Sistemaren arkitektura informazio-iturri digitalen kudeaketa eraginkorra eta arina eskaintzeko diseinatu da. Jarraian, sistemaren egituraren laburpena aurkezten da, diagrama argigarri batekin batera:

- Katalogoaren-kudeaketa - NEXTJS (nodejs):

Iturrien katalogoaren kudeaketa eta aurkezpena maneiatzen du. Administratzaileei iturrien itzulpenak berrikusi, kategorizatu eta kudeatzeko aukera ematen die.

- API - Flask:

Flask liburutegiarekin egindako API bat sortu da Python mikroframeworkean. Frontend eskaerak maneiatzen ditu eta datu-basean eragiketak egiten ditu. Bitartekari gisa jarduten du, datuak prozesatuz eta frontend eta datu-basearen arteko komunikazio eraginkorra ziurtatuz.

- Datu-basea - MySQL:

Datu-base erlazionalak kudeatzeko sistema, sistemaren datu guztiak biltzen dituena. Erabiltzaileentzako taulak, iturriak, kategoriak, etiketak eta itzulpenak barne hartzen ditu. Datu guztiak kudeatu eta gordetzeko erabili da MySQL hizkuntako datu basea.

- Software Nagusia - PHP (Apache):

Software nagusia, Apache serbitzari batean altxatuta dago, bertan katalogoa inplementatu da, erabiltzaileei erreztasunak emateko iturri berriak ezagutu ditzaten.

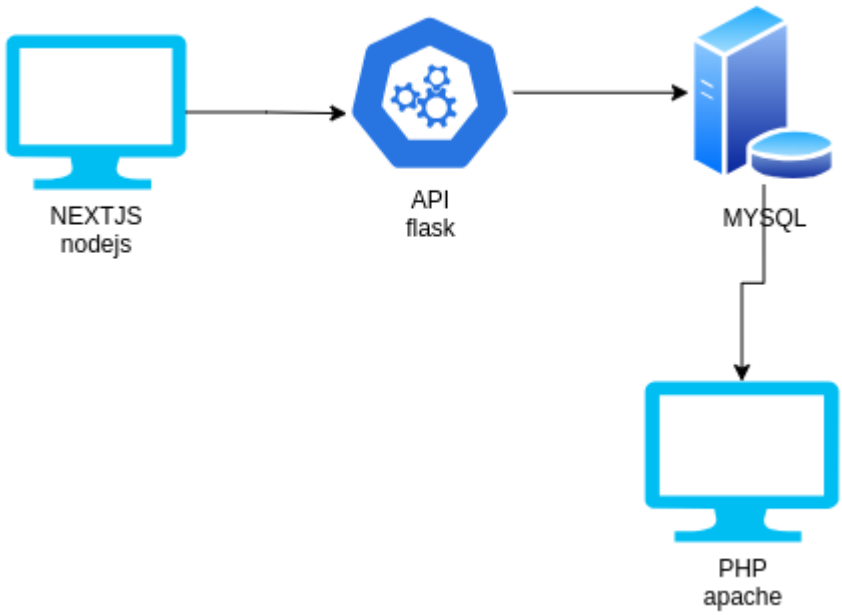


Figure 11: Sistemaren arkitektura

3.2 Administrazio-paneleko emaitzak

Administrazio-panelaren interfazea argia eta antolatua izateko diseinatu da, eta kudeatzaileei erabiltzaileek gehitutako informazio-iturri guztien ikuspegi osoa eman zaie.

3.2.1 Iturri guztien atala

Interfaze printzipalak iturri guztien zerrenda ematen du, eta xehetasun garrantzitsuak erakusten ditu, hala nola izena, kategoria eta berrikuspen-egoera. Kudeatzaileek iturriak iragazi, bilatu eta ordenatu ditzakete, kudeaketa eraginkorragoa izan dadin.

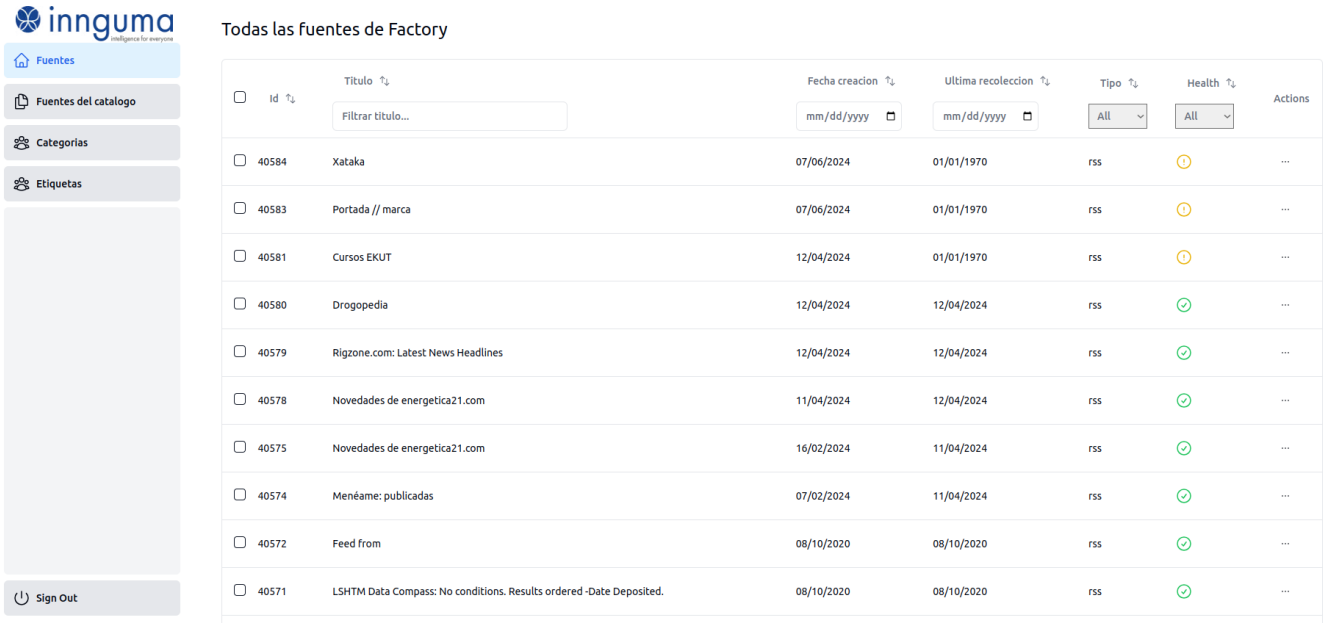


Figure 13: Iturri guztien pantaila

3.2.1.1 Akzio gehigarriak

Gero iturriek akzio espezifiko batzuk dituzte, atal honetan iturrien informazio gehiago ikusi dezakezu edo katalogora gehitu automatikoki.

- Informazio gehiago

Informazio gehiagoren kasuan, ze frekuentziarekin jasotzen diren iturriaren datuak, eta zenbat pertsonek duten iturri hau gehituta beraien kontutan ikusi daiteke, hau administratzaileentzat oso interesgarria da, iturri baten arrakasta zenbatekoa den ebaluatzeko eta gero horren arabera katalogora gehitu ahal izango dute.



Figure 15: Informazio gehiago opzioaren pantaila

- Katalogora gehitu iturria

Gero, katalogora gehitu nahi bada, formulario moduko bat agertuko da, bertan ze kategoriatan gehitu nahiko den aukeratu beharko da. Gero ere iturria etiketatu ahalko da hizkuntza, izenburua eta deskribapena ezartzen ere.

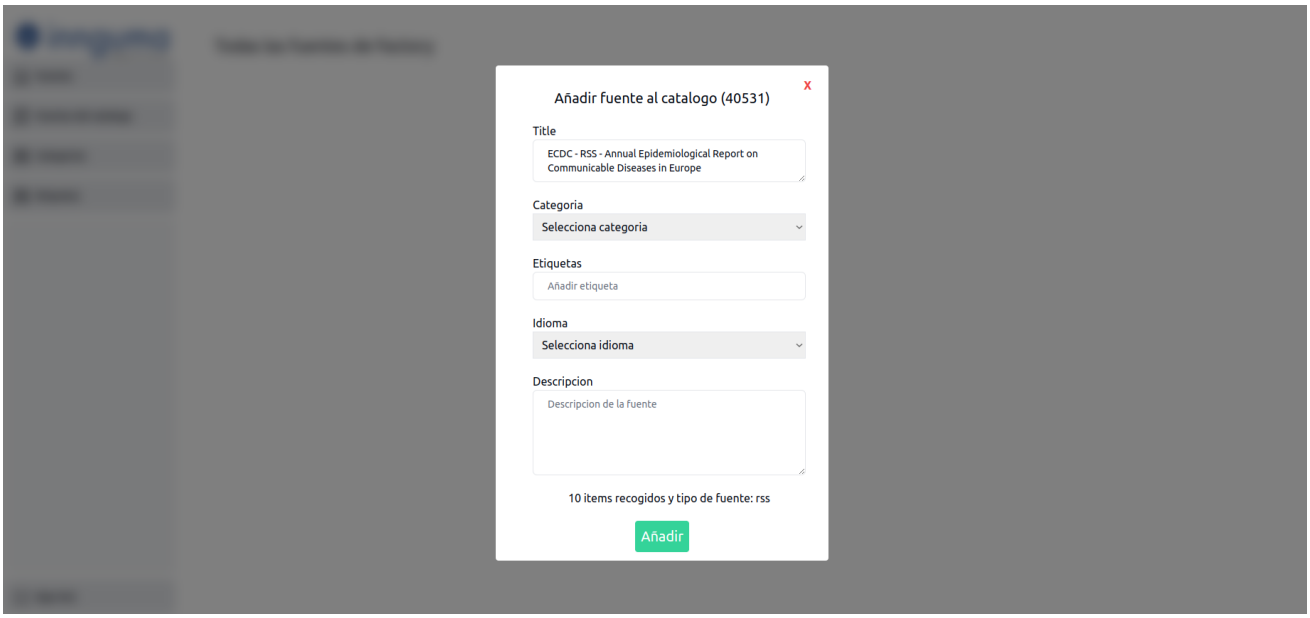


Figure 17: Iturria katalogora gehitzearen pantaila

- Iturriari tradukzioak gehitu

Iturria gehitu ostean, beste formulario bat ateratzen da, iturriaren tradukzioak ezartzeko.

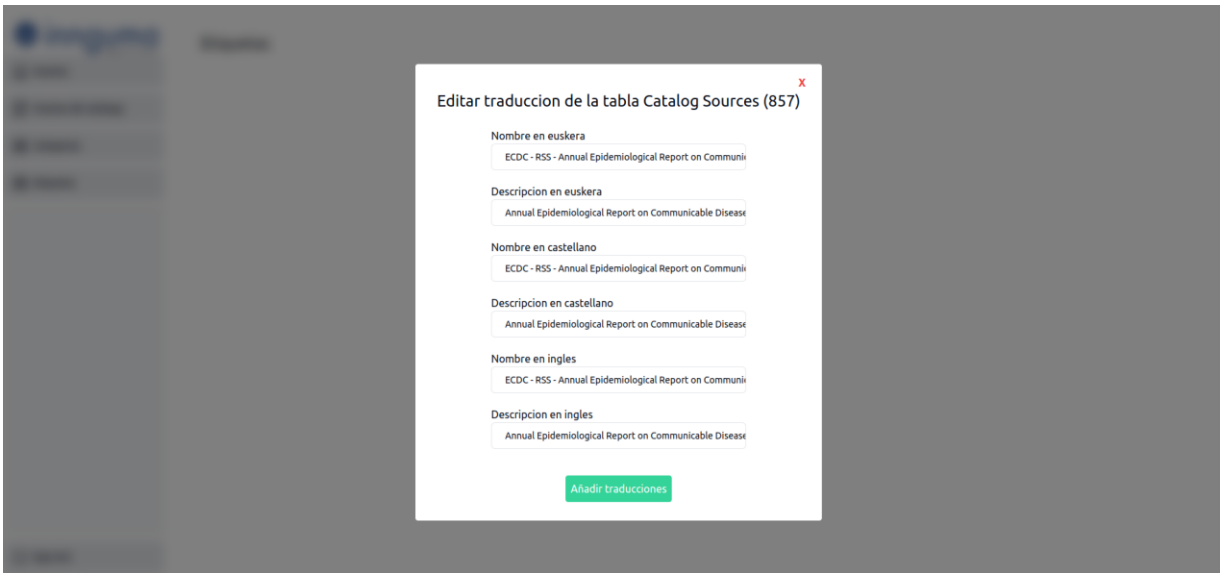


Figure 19: Tradukzioak gehitzeko pantaila

3.2.2 Katalogoko iturrien atala

Katalogoaren iturriak modu antolatuan aurkezten dira, informazioa bilatzeko eta iragazteko interfaze intuitibo batekin. Iturri bakoitzak bere kategoriak eta etiketak ditu, eta erabiltzaileei informazio garrantzitsua erraz eskuratzeko aukera ematen die.

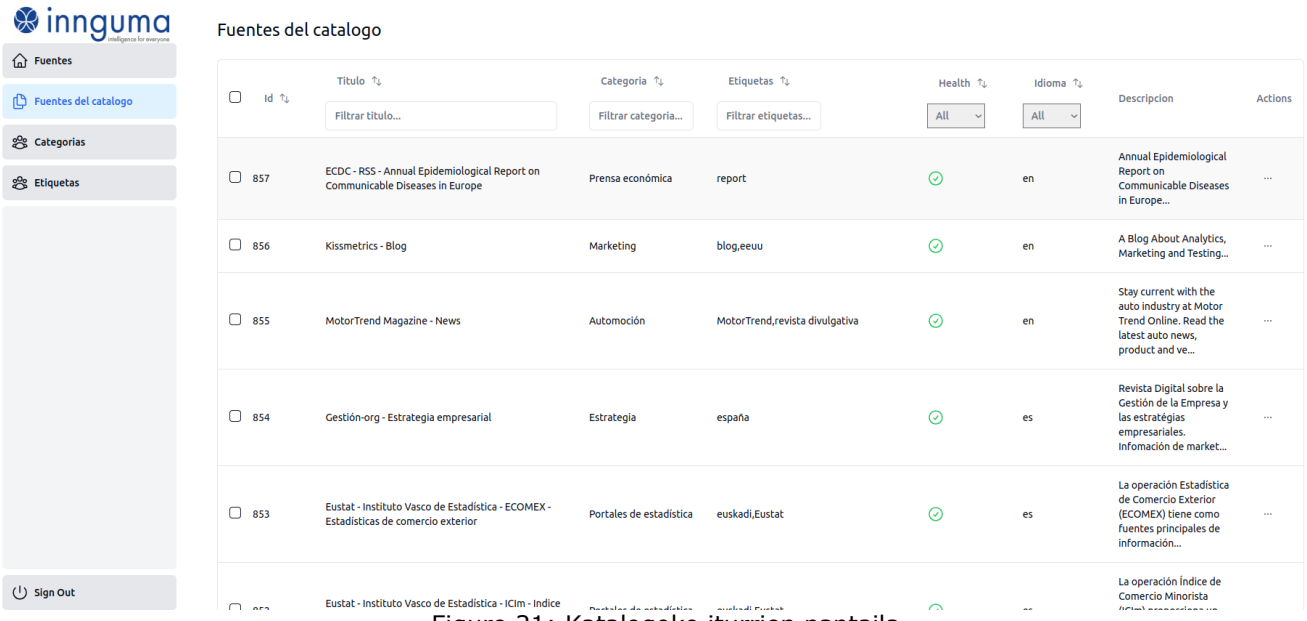


Figure 21: Katalogoko iturrien pantaila

3.2.2.1 Akzio gehigarriak

Akzioen opzioetan hemen, iturria editatzea, iturriaren tradukzioak aldatzea eta iturria katalogotik kentzean dago.

- Iturria editatzea

Iturria editatzeko pantailan, iturriaren informazioa aldatu ahal izango da, izenburua, kategoria, etiketak, hizkuntza eta deskripzioa barne.

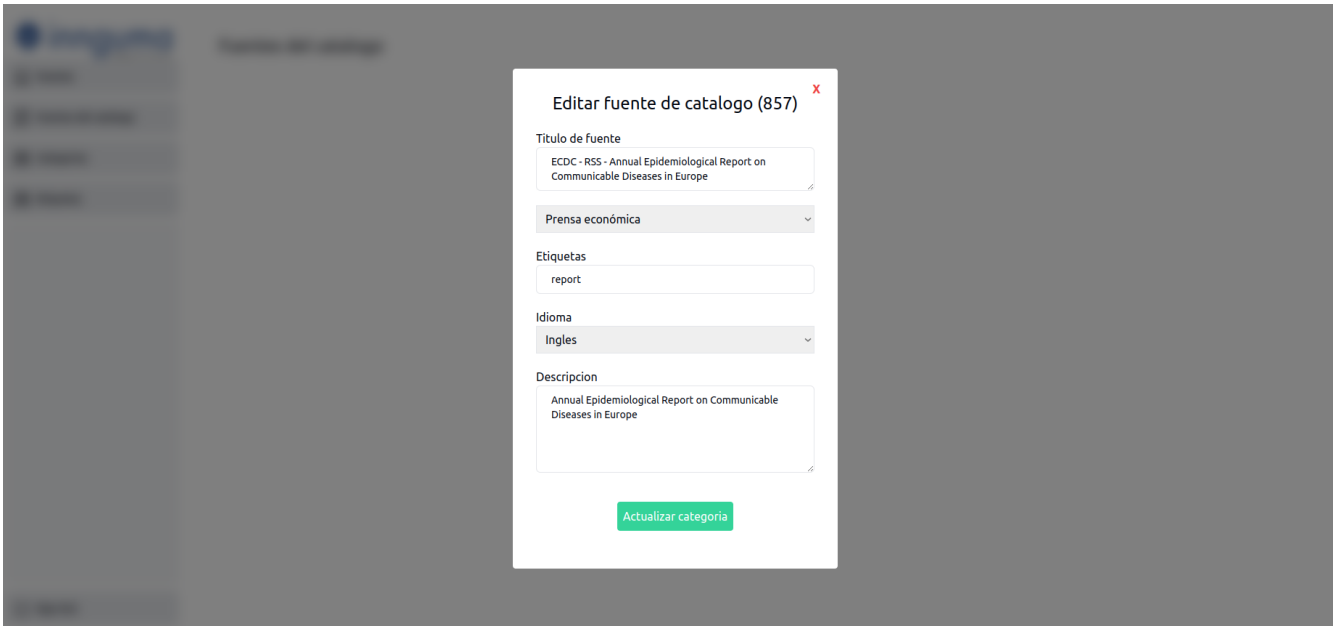


Figure 23: Iturria editatzearen pantaila

- Tradukzioak aldatzea

Tradukzioak aldatzeko pantailan, iturriaren tradukzio guztiak aldatu izango ahal dira, izenburua eta deskripzioa barne.

- Iturria katalogotik kentzea

Iturria katalogotik kendu ahal izango du administratzaileak, hau iturria jauzita egoten bada edo pasatu ahalko da. Hemen konfirmazio modal bat agertuko da iturria ezabatu nahi dela bermatzeko.

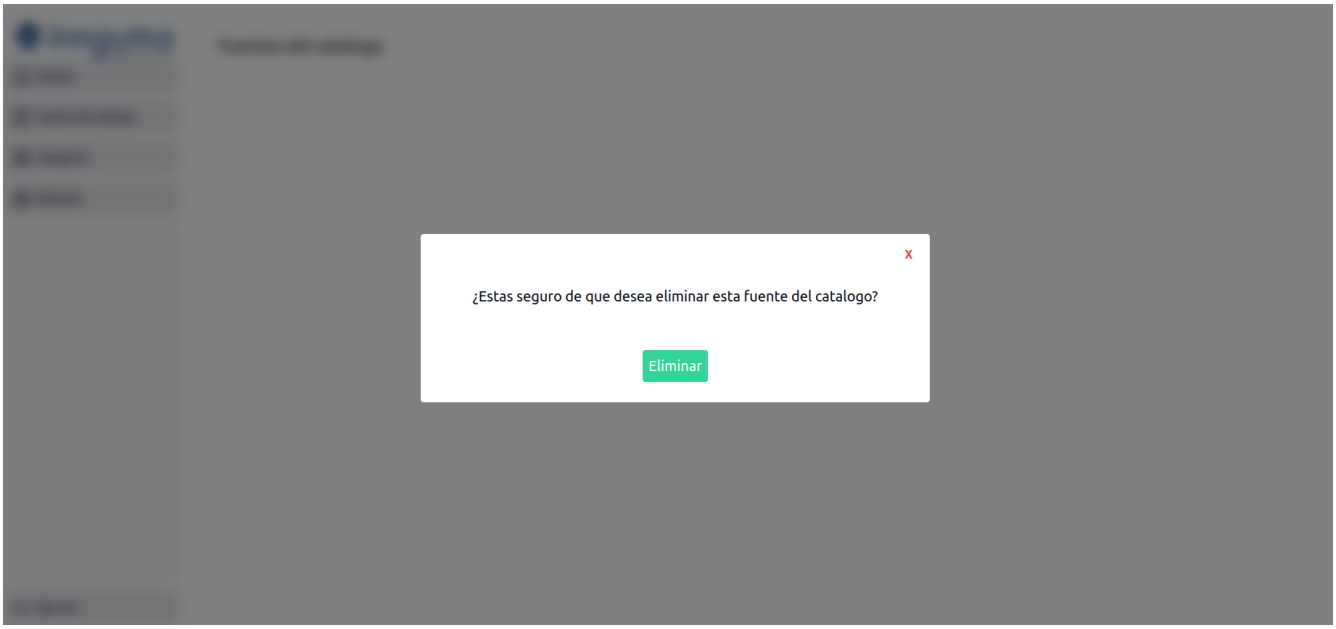


Figure 25: Konfirmazioko mezuaren pantaila

3.2.3 Kategorien atala

Kategorien atalak katalogoan nabigatzea errazten du, eta erabiltzaileei aukera ematen die iturriak gai espezifikoen arabera taldekatuta aztertzeko. Kudeatzaileek kategoriak gehitu eta alda ditzakete administrazio-paneletik, katalogoa beti eguneratuta eta ondo antolatuta dagoela ziurtatuz. Kategoria baten izenaren gainean sakatzerakoan kategoria horretan esleituta dauden iturri guztiak listaratuko dira katalogoko atalean.

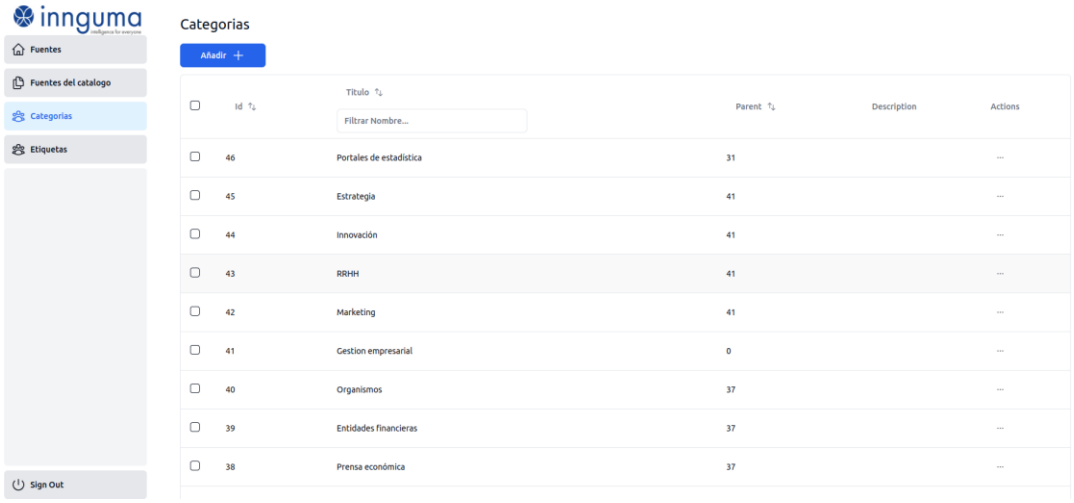


Figure 27: Kategorien atalaren pantaila

3.2.3.1 Kategoria berria sortu

Atal honetan, kategoria berriak sortzeko aukera dago, kategoria gehitzerakoan izen bat jarri behar zaio kategoria berriari, kategoria aita bat ere esleitu behar zaio, ez bazaio kategoriarik esleitzen bera izango da kategoria aita. Ondoren deskripzio bat gehitu beharko zaio eta azkenik software printzipalean edukiko duen ikonoa, ikonoa fontawomekoa izan beharko du.

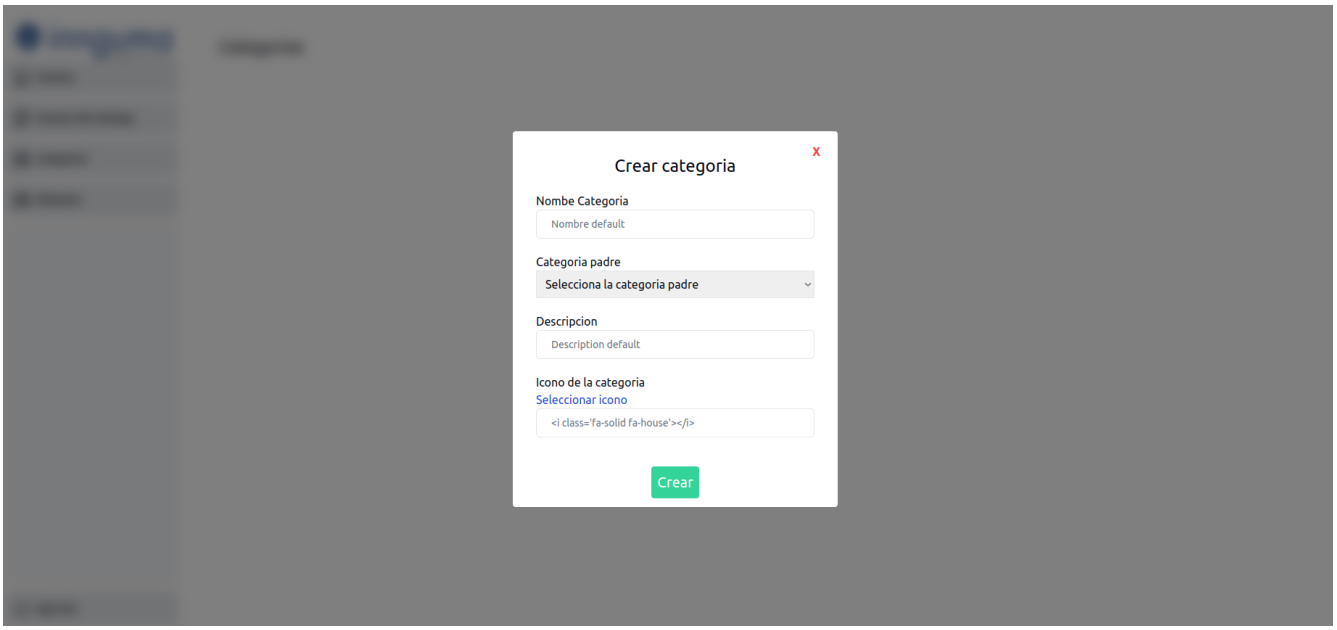


Figure 29: Kategoria berria sortzearen pantaila

3.2.3.2 Akzio gehigarriak

Kategorien akzio gehigarriak, editatzea, tradukzioak aldatzea eta kategoria ezabatzea dira. Kategoria ezabatzen bada, konfirmazio pantaila bat agertzen da, kategoria horretara esleituta dauden iturri guztiak katalogotik kanpo gelditukо direla abisatzen.



Figure 31: Konfirmazio mezuaren pantaila

3.2.4 Etiketen atala

Etiketek iragazteko geruza gehigarri bat ematen dute, eta erabiltzaileei aukera ematen diete bilaketa zehatzago fintzeko. Kategoriekin bezala, etiketak administrazio-paneletik kudeatzen dira, eta beharren arabera sortu eta aldatzeko aukera ematen dute. Etiketaren izenarem gainean sakatzean, katalogoko atalera berbideratuko da, etiketa horren iturri guztiak zerrendatzen.

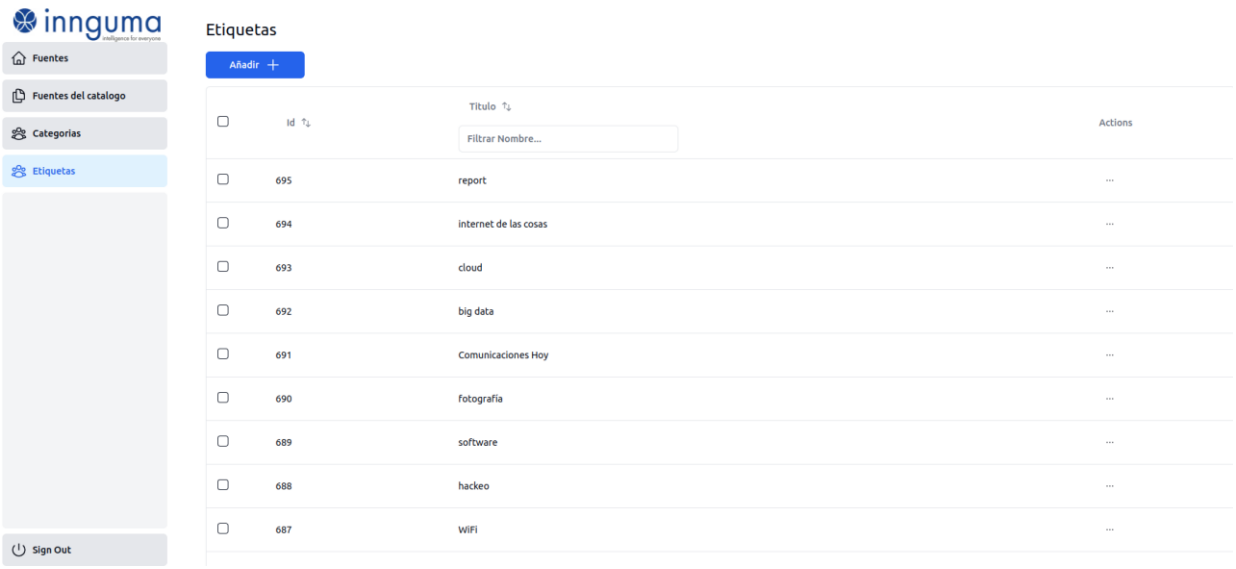


Figure 33: Etiketen atalaren pantaila

3.2.4.1 Etiketa berri bat sortzea

Etiketa berri bat sortzerako orduan, etiketek izena bakarrik dutenez, bertan tradukzioak aldatzea erabaki da.



Figure 35: Etiketa berri bat sortzearen pantaila

Tradukzioak aktibatzea erabakitzen bada izena hiru hizkuntzetan traduzitzeko aukera emango da. Aukeraketa hori botoi baten bidez egin da.

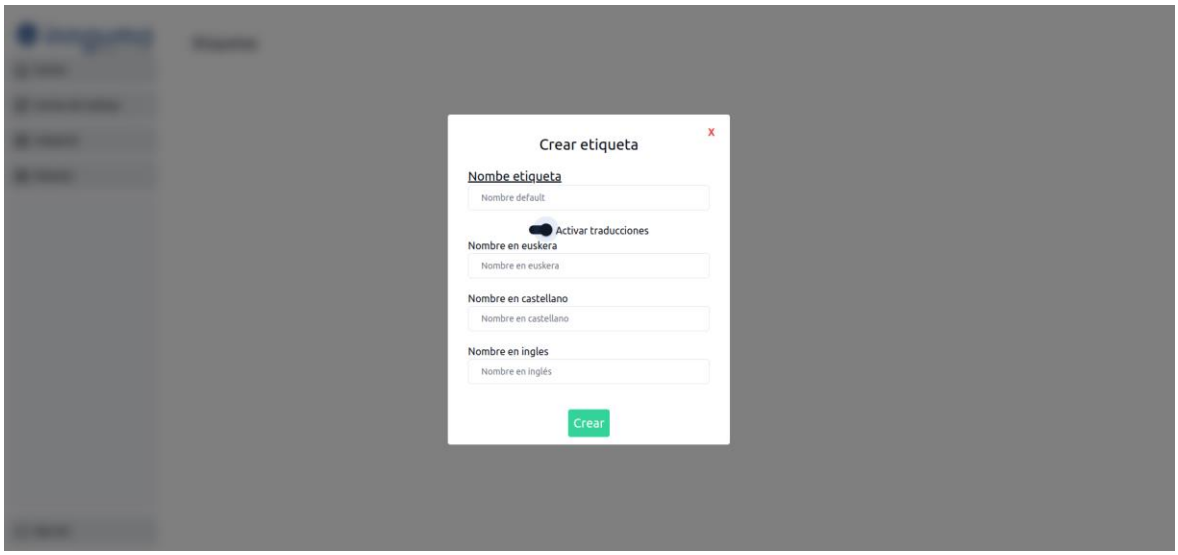


Figure 37: Etiketa berri bati tradukzioak gehitzearen pantaila

4. Memoria ekonomikoa

Proiektuaren memoria ekonomikoaren helburua INNGUMArako iturriak kudeatzeko web-aplikazioaren garapenarekin eta inplementazioarekin lotutako kostuen ikuspegi zehatza eta ulergarria ematea da. Azterketa horrek kostu zuzenak, ekipoa eta erabilitako softwarea barne hartzen ditu, baita inbertsioaren errentagarritasunaren eta itzuleraren ebaluazioa ere. Helburua proiektua teknikoki bideragarria ez ezik ekonomikoki errentagarria ere izango dela bermatzea da, finantza-baliabideak nola banatzen diren eta epe labur eta luzera zer onura ekonomiko espero diren ulertzeko esparru argi bat eskainiz.

4.1 Erabilitako ekipoaren kostua

Atal honetan proiektua garatzeko erabilitako ekipo guztien kostua zehazten da.

Materiala	Kantitatea	Prezioa
Ordenagailu portatila	1	850€
Pantaila gehigarria	1	90€
Arratoia	1	15€
Guztira	3	955€

Taula 1 Erabilitako ekipoen kostua

4.2 Softwarearen kostua

Proiektuaren garapena planifikatu da dauden baliabideen erabilera maximizatzeko eta kostu gehigarriak minimizatzeko. Bereziki, AWSn eskuragarri dagoen Docker edukiontzien azpiegitura erabili da aplikazioa hedatzeko. Horri esker, softwarearen kostu gehigarriak saihestu dira, ez baita beharrezkoa izan lizentzia edo kanpoko zerbitzu berriak eskuratzea aplikazioa hedatzeko eta kudeatzeko.

4.3 Langileen kostua

Jarraian, proiektuan parte hartzen duten langileen kostuak zehazten dira. Azpimarratu behar da enpresan egindako eskabidearen ordu guztiak zenbatu direla, ez bakarrik proiektua garatzeko orduak. Beraz, zerikusia duten jarduera guztiak hartu dira kontuan, prestakuntza, eskakizunen definizioa, diseinua, garapena, memoriaren sorrera eta lotutako beste zeregin batzuk barne. Ikuspegi integral horri esker, proiektuan inbertitutako ahaleginaren eta giza baliabideen ebaluazio zehatza egin daiteke, inplikatutako lan-kostuen irudi osoa emanez.

Langilea	Dedikatutako orduak	Kostua
Martxel Aranzadi Olazabal	30 ECTS (750 horas)	3500€
Guztira	-	3500€

Taula 2 Langileen kostua

4.4 Koste totala

Proiektuaren kostua, guztira, **4455€** -koa izan da, garatzailearen soldatari eta erabilitako ekipoari dagokiona. Ez da gastu gehigarririk egin, software librea baliatu delako eta AWSn dauden Docker edukiontziak berrerabili direlako aplikazioa hedatzeko, kanpoko softwarearen bidezko ordainketa saihestuz.

4.5 Inbertsioaren errentagarritasuna eta itzulera

Proiektu hau produktu independentea ez denez eta Innguma aplikatzeko funtzionaltasun berri bat denez, ezin da zehaztu zenbat diru lortu den soilik administrazio-panel honen inplementazioarekin. Hala ere, funtzionaltasun hori funtsezko osagaia da, enpresaren eraginkortasun operatiboa nabarmen hobetuko duena.

Administrazio-panel honek ez du iraungitze-datarik, eta aplikazioa amaitu arte egongo da martxan. Beraz, epe luzerako inbertsiotzat ere har daiteke. Bere garapenak eta abiarazteak iturrien kudeaketa etengabe hobetzen lagunduko dute, Inngumaren plataforma eraginkor eta garrantzitsu mantentzen dela ziurtatuz. Inbertsio estrategiko horrek barne-prozesuak optimizatzeaz gain, bezeroei eskainitako produktuaren balioa ere handitzen du, denboran zehar inbertsio iraunkorreko itzulera ziurtatuz.

5. Ondorioak

Proiektuak planteatutako helburuak betetzea lortu du, INNGUMA aplikazioan iturriak kudeatzeko eta kategorizatzeko irtenbide osoa eta eraginkorra eskainiz. Jarraian, alderdi teknikoei, metodologikoei, ekonomikoei eta Garapen Jasangarriko Helburuetan (GJH)[27] duten eraginari buruzko ondorioak aurkeztuko ditugu:

5.1 Alderdi teknikoak

Next.js framework gisa aukeratzea oso egokia izan da, errendimendu bikaina duelako eta zerbitzariaren aldea errenderizatzeko gaitasuna duelako (SSR). Teknologia horri esker, erabiltzaile-interfaze azkar eta eraginkor bat sortu da, azken erabiltzailearen esperientzia nabarmen hobetuz. Bestalde, MySQLren inplementazioa MyISAM motorrarekin erabakigarria izan da datu-bolumen handia maneiatzeko. Motor hau irakurketa-eragiketetan duen abiaduragatik eta eraginkortasunagatik da ezaguna, eta horrek datu-basearen errendimendu optimoa bermatu du. Erabaki teknologiko horiei esker, sistema sendoa eta arina garatu ahal izan da, enpresaren egungo eta etorkizuneko beharrak asetzeko.

5.2 Alderdi metodologikoak

Metodologia arinen erabilera eta aldizkako bilerak funtsezkoak izan dira proiektua arrakastaz garatzeko. Garatzaileekin astero egiten diren bilerek erraztu egin dute arazo teknikoak azkar koordinatzea eta konpontzea. Laguntzailearekin egindako bilerek orientazio eta feedback espezializatu baliotsua eman dute, erabaki teknikoak zuzenak izango zirela ziurtatuz. Gainera, parte-hartzaile guztiekin egindako bilerei esker, salmenta-taldea barne, ikuspegi zabala lortu ahal izan da, azken erabiltzailearen eta merkatuaren ikuspuntu baliotsuak txertatuz. Elkarlaneko ikuspegi malgu horri esker, aldaketetara egokitu ahal izan da, eta garapen arin eta efizienteari eutsi ahal izan zaio.

5.3 Alderdi ekonomikoak

Proiektua modu eraginkorrean gauzatu da kostuei dagokienez. Doako edo kostu txikiko tresnak erabiltzeak, hala nola Visual Studio Code, GitLab eta Docker, eskura dauden baliabideak maximizatzea ahalbidetu du, azken produktuaren kalitatea arriskuan jarri gabe. Katalogo eleaniztunaren garapenak balio esanguratsua gehitu dio produktuari, merkatu globalean duen erakargarritasuna eta lehiakortasuna zabalduz. Proiektu hau esparru ekonomiko kontrolatu baten barruan ezartzeak erakutsi du posible dela irtenbide teknologiko aurreratuak garatzea gehiegizko gastuetan erori gabe.

5.4 Eragina Garapen Jasangarriko Helburuetan (GJH)

Administrazio-panelaren garapenak eta Innguman informazio-iturrien kudeaketa hobetzeak nabarmen laguntzen dute NBEk ezarritako Garapen Iraunkorreko Helburuetan (GIH). Proiektu honetarako GJH garrantzitsuenak honako hauek dira:

5.4.1 Industria, Berrikuntza eta Azpiegitura (9. helburua)

Azpiegitura erresiliente bat garatzeko, industrializazio inklusibo eta jasangarria sustatzeko eta berrikuntza sustatzeko helburua hainbat azpi-helburutan islatzen da:

5.4.1.1 Azpiegitura fidagarriak, jasangarriak, erresilienteak eta kalitatezkoak garatzea (9.1. helburua)

Administrazio-panelak informazioaren kudeaketa hobetzen du, Inngumaren azpiegitura teknologikoa sendoagoa eta eraginkorragoa izan dadin. Informazio-iturriak zentralizatzean eta antolatzean, sistemaren fidagarritasuna eta iraunkortasuna areagotzen dira.

5.4.1.2 Azpiegitura modernizatzea eta industriak birmoldatzea, jasangarriak izan daitezen (9.4. helburua)

Iturriak kudeatzeko API bezalako teknologia aurreratuak ezartzeak Inngumaren azpiegituraren modernizazioan eta jasangarritasunean laguntzen du. Aplikazioa hedatzeko AWSn Docker edukiontziak erabiltzeak ikuspegi berritzaile eta jasangarria erakusten du.

5.4.1.3 Industria-sektoreen gaitasun teknologikoa hobetzea herrialde guztietan (9.5. helburua)

Next.js bezalako teknologia modernoen erabilerak eta dauden sistemekin integratzeak Inngumaren gaitasun teknologikoa hobetzen du, enpresa bere sektorean berrikuntzan lider gisa kokatuz.

5.4.2 Lan Duina eta Hazkunde Ekonomikoa (8. helburua)

Helburu horren bidez, hazkunde ekonomiko inklusiboa eta iraunkorra, enplegu osoa eta produktiboa eta guztiontzako lan duina sustatu nahi dira.

5.4.2.1 Produktibitate ekonomikoaren maila handiagoak lortzea dibertsifikazioaren, modernizazio teknologikoaren eta berrikuntzaren bidez (8.2. helburua)

Iturrien kudeaketa hobetzeak eta administrazio-panel eraginkor bat ezartzeak Inngumako langileen produktibitatea handitzen du. Modernizazio teknologikoari esker, langileek modu eraginkorragoan eta tresna hobeekin lan egin dezakete.

5.4.2.2 Garapenera bideratutako politikak sustatzea, ekoizpen-jarduerak, lanpostu zintzoen sorrera, ekintzailetza, sormena eta berrikuntza bultzatzeko (8.3. helburua)

Administrazio-panelaren garapenak sormena eta berrikuntza sustatzen ditu enpresaren barruan. Langileak tresna aurreratuekin hornitzeak produkzio-jarduerak garatzen eta lan-giro eraginkorragoa eta txukunagoa sortzen laguntzen du.

5.4.3 Ekoizpen eta Kontsumo Arduratsua (12. helburua)

Helburu horren helburua kontsumo eta ekoizpen iraunkorreko modalitateak bermatzea da.

5.4.3.1 Enpresak praktika jasangarriak hartzera eta txostenak aurkezteko zikloan iraunkortasunari buruzko informazioa sartzera bultzatzea (12.6. helburua)

Administrazio-panelaren garapenak eta teknologia eraginkor eta iraunkorren erabilerak erakusten du Inngumak ekoizpen arduratsuarekin duen konpromisoa. Informazioaren kudeaketa eta eraginkortasun operatiboa hobetzean, Inngumak praktika iraunkorragoak hartzen ditu.

5.4.3.2 Mundu osoko pertsonek garapen iraunkorrerako informazio eta ezagutza garrantzitsuak izango dituztela ziurtatzea (12.8. helburua)

Administrazio-panelak aukera ematen die erabiltzaileei informazioa modu antolatuago eta eraginkorragoan eskuratzeko, datu fidagarrietan eta eguneratuetan oinarritutako erabakiak hobeto hartzen lagunduz. Horrek enpresari ez ezik, bezeroei eta bazkideei ere egiten die mesede.



Figure 39: Garapen jasangarriko helburuak

6. Etorkizuneko ildoak

INNGUMArako iturrien katalogoaren eta administrazio-panelaren garapenak mugarri esanguratsua ezarri du zaintza teknologikorako informazioaren kudeaketan eta antolaketan. Hala ere, beti daude aukerak lortutako emaitzak zabaltzeko eta hobetzeko. Ondoren, egindako lanarekin jarraitzeko etorkizunean egin litezkeen ideia eta azterlan batzuk aurkezten dira:

6.1 Administrazio-panela handitzea

Administrazio-panelak eboluzionatu egin lezake, iturri guztiak kudeatzeko tresna integral bihurtzeko. Horrek, iturrien katalogoaren administrazioa ez ezik, iturrien osasunaren monitorizazio aktiboa ere barne hartuko luke. Adibidez, iturri batek eskuragarri egoteari uzten badio edo erortzen bada, sistemak jakinarazpen bat bidal diezaioke kudeatzaileari. Era berean, kudeatzaileak panela erabili ahal izango du eroritako iturria ordeztu dezaketen iturri alternatiboak bilatzeko eta ebaluatzeko, informazio-fluxuaren jarraitutasuna ziurtatzeko.

6.2 Kategorizazioaren automatizazioa eta hobekuntza

Ikaskuntza automatikoko algoritmoak ezartzea, iturrien kategorizazioa eta etiketatzea hobetzeko. Adimen artifiziala erabiliz, sistemak kudeatzailearen erabakietatik ikas lezake eta iturri berrietarako kategorizazio automatikoak iradoki, denbora aurreztuz eta doitasuna hobetuz.

6.3 Datuak aztertzeko tresnak integratzea

Administrazio-panela datuak aztertzeko eta bistaratzeko tresna aurreratuekin integratzea, hala nola Tableau edo Power BI tresnekin. Horri esker, kudeatzaileek iturriak antolatu eta administratu ez ezik, bildutako informazioari buruzko azterketa sakonak egin, joerak identifikatu eta txosten zehatzak modu eraginkorragoan sortu ahal izango dituzte.

6.4 Sistema eleanitzaren hedapena

Katalogoaren gaitasun eleanitzak zabaltzen jarraitzea. Horrek, etiketen eta kategorien itzulpena ez ezik, iturrietako edukietarako itzulpen automatikoko zerbitzuen integrazioa ere barne har lezake, erabiltzaileei informazioa beren hizkuntza gogokoenean eskuratzeko aukera emanez, iturriko jatorrizko hizkuntza kontuan hartu gabe.

Segurtasuna eta pribatutasuna hobetzea

Segurtasun- eta pribatutasun-neurri aurreratuak ezartzea, administrazio-panelak erabiltzen dituen datuak babesteko. Horrek barne har litzake datuen zifratzea, faktore anitzeko autentifikazioa eta aldizkako segurtasun-auditoretzak, informazio sentikorraren babesa bermatzeko.

Etorkizuneko ildo horiek egungo sistemaren funtzionaltasuna eta eraginkortasuna hobetzeaz gain, INNGUMArentzat ere aukera berriak irekiko lituzkete berrikuntzari eta hazkundeari dagokienez. Arlo

horiek aztertzen eta garatzen jarraitzeak bermatuko du tresnak garrantzitsua eta baliotsua izaten jarraitzea erabiltzaileentzat, haien behar aldakorretara egokituz eta adimen lehiakorreko eta zaintza teknologikoko teknologiaren abangoardian mantenduz.

7. Balorazio pertsonala

Proiektu hau oso esperientzia interesgarria izan da, batez ere administrazio-paneleko teknologia eta programazio-lengoaia hautatzeko izan dudan askatasunagatik. Autonomia horri esker, hainbat aukera alderatu eta ebaluatu ahal izan ditut, eta, azkenik, Next.js aukeratu, gero eta ospe handiagoa eta zerbitzariaren aldea errenderizatzeko gaitasun sendoak dituelatko (SSR). Next.js zerotik erabiltzen ikastea, React hizkuntzaz gain, erronka bizigarria izan da, eta oinarri sendoa eman dit web garapeneko teknologia modernoetan.

Next.jsekin eta Reactekin lan egitearen esperientzia baliagarria izan da trebetasun tekniko aurreratuak eskuratzeari dagokionez, eta, horrez gain, inbertsio bat izan da nire etorkizun profesionalean. Teknologia horiek etengabe ari dira eboluzionatzen, eta gero eta gehiago eskatzen dira lan-merkatuan. Hori dela eta, teknologia horiek menderatzeak aukera ugari ematen ditu garapen profesionalerako eta hazkunderako.

Pertsonalki, proiektu honetako datu-baseen kudeaketaren alderdiaz gozatu dut bereziki. Bi urte lehenago arlo honetan praktikak egiten eman nuen esperientziak oinarri sendoa eman dit, baina proiektu bat zerotik garatzea, MySQL eta MyISAM datu-baseen motor gisa erabiliz, esperientzia aberasgarria izan da. Aukera horri esker, ingurune praktiko batean aplikatu eta zabaldu ditut nire ezagutzak, datu-baseak administratzeko grina eta arlo horretan ditudan trebetasunekiko konfiantza indartuz.

8. Bibliografia

[1] Ideko, «Centro Tecnológico | IDEKO». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.ideko.es/
[2] «Ezagutu Danobatgroup - Danobatgroup I Grupo Industrial». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.danobatgroup.com/eu/taldea
[3] «Innguma Home», Innguma. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.innguma.com/
[4] S. Wright, D. W. Pickton, y J. Callow, «Competitive intelligence in UK firms: a typology», *Mark. Intell. Plan.*, vol. 20, n.º 6, pp. 349-360, nov. 2002, doi: 10.1108/02634500210445400.
[5] «Inoreader – Build your own newsfeed», Inoreader. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.inoreader.com/
[6] D. Inc, «Digimind | Plataforma de Inteligencia Social y Competitiva». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.digimind.com/es/
[7] «Feedly: Track the topics and trends that matter to you». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://feedly.com/
[8] «Using Design Systems to Build Good UX Products», Gartner. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.gartner.com/smarterwithgartner/software-engineers-need-design-systems-to-build-products-with-good-ux
[9] P. Camarelles, «Salud y Prevención PAPPS: ¿Que es el método M.A.R.T.E?», Salud y Prevención PAPPS. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://educacionpapps.blogspot.com/2018/09/que-es-el-metodo-marte.html
[10] «Docs | Next.js». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://nextjs.org/docs
[11] «Angular». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://angular.dev/
[12] «Vue.js». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://vuejs.org/
[13] «MySQL». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.mysql.com/
[14] «MySQL :: MySQL 8.4 Reference Manual :: 18.2 The MyISAM Storage Engine». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://dev.mysql.com/doc/refman/8.4/en/myisam-storage-engine.html
[15] «PostgreSQL: About». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.postgresql.org/about/
[16] «Bases de datos no relacionales | Bases de datos de gráficos | AWS», Amazon Web Services, Inc. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://aws.amazon.com/es/nosql/
[17] «MongoDB Atlas Database | Multi-Cloud Database Service», MongoDB. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.mongodb.com/products/platform/atlas-database
[18] «API — Flask Documentation (3.0.x)». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://flask.palletsprojects.com/en/3.0.x/api/
[19] «PHP: ¿Qué es PHP? - Manual». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.php.net/manual/es/intro-whatis.php
[20] «Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.» Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://tailwindcss.com/
[21] «tailwindlabs/heroicons». Tailwind Labs, 28 de junio de 2024. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://github.com/tailwindlabs/heroicons
[22] «Docker: Accelerated Container Application Development». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.docker.com/

[23] «HTTP response status codes - HTTP | MDN». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
[24] «Amazon Elastic Compute Cloud Documentation». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://docs.aws.amazon.com/ec2/?icmpid=docs_homepage_featuredsvcs
[25] T. J. Project, «Joomla! - Content Management System to build websites & apps», Joomla! Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.joomla.org/about-joomla.html
[26] «MVC Framework Introduction», GeeksforGeeks. Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://www.geeksforgeeks.org/mvc-framework-introduction/
[27] «Garapen Jasangarrirako Helburuak (GJH) - Gobernu Programa». Accedido: 28 de junio de 2024. [En línea]. Disponible en: https://programa.irekia.euskadi.eus/eu/sustainable_dev_goals

U BUKAERAKO LANA GRADU BUKAERAKO LANA GRADU BUKAERAKO LANA GRADU BUKAERAKO LANA GRADU BUKAERAKO LANA