

Arkitektura Koadernoa IO-System

1. Helburua

Dokumentu honek deskribatu egiten ditu proiektuan zehar, eta zehazki IO-System azpisistemaren atalean, filosofia, erabakiak, baldintzak, justifikazioak eta garrantzizkoa izan daitekeen edozer.

2. Arkitektura helburuak eta filosofia

CMS baten bitartez kudeatutako web-aplikazioa izango da. Helburua metodologia jarraitzen duten proiektuen informazioa gordetzea da. Rol bakoitzak metodologian dituen ataza berdina bete beharko ditu.

Azpisistemak hurrengo helburu nagusiak ditu:

- Erabiltzaileen autentikazioaren kudeaketa.
- Erabiltzaileen rola eta baimenak kudeatzea.
- Bizi-zikloa definitzen duen metodologiaren informazioa gordetzea.
- Proiektuak sortzea eta bere informazioa gordetzea.
- Bizi-zikloan zehar sortutako dokumentuak gordetzea.
- Sortutako edukia guztia bistaratzea.

3. Hipotesiak eta menpekotasunak

CMS baten erabilera datuen sarrera/irteerarako irtenbide egokiena da. Webgune bat sortzeko aukera ematen duen tresna erabilerraza izateaz aparte, ez da baliabide tekniko aurreratuera etengabe jo behar. Kudeaketa, administrazioa eta mantentze-lanak egiteko laguntza ematen du kanpoko baliabiderik erabili gabe.

Datu-base erlazionalak prozesu baten ezagutza gordetzeko modurik egokiena da, datuen independentzia, emaitzen koherentzia eta datu-basearen produktibitatea handitzea lortuz.

IO-System azpisistemaren arkitektura ModelEditor azpisistemarekin erlazionatuta dago. Izan ere, ModelEditor-ekin definitutako ereduaren datuak erabili beharko ditu. Hala ere, dependentzia ez da ProWF proiektuan bezain zuzena, ModelEditor sistemaren arkitektura abstraktuagoa delako.

Horrez gain, Drupal CMSarekiko dependentzia izango du, sistema horren gainean eraikita dagoelako. Beraz, Drupal-en bertsioak kudeatu beharko dira. Drupalek aldi berean dependentziak dituen, horiek ere kudeatu beharko dira. Horregatik, Composer erabiliko da, dependentzien kudeaketa errazteko.

4. Betekizun arkitekturalki esanguratsuak

Sistemaren betekizunak Ikuspegia eta Betebeharren Espezifikazioa dokumentuetan definitu dira.

5. Erabakiak, mugak eta justifikazioak

Azterketa sakon bat egin eta aukera bakoitza ebaluatu ostean, Drupal CMSa erabiltzea izan zen erabakia, hurrengo arrazoiengatik:

- Drupalen erraza da edukia gehitzea/sortzea. Eduki pertsonalizatu motak malguak dira eta aukera asko eskaintzen dituzte.
- Guneari gehitzeko hainbat modulu eskuragarri daude bere webgunean eta proiektu honetarako oso erabilgarriak diren moduluak aurkitu ziren.
- Erabiltzaileak administratzea erraza da, rol berriak sortu eta baimenak zehaztu ditzakeen sistema integratu batekin. Funtzionalitate hori oso komenigarria zen proiektu honentzat.

- Mundu mailan garrantzitsuenak diren teknologia saltzaileen sailkapenak argitaratzen dituzten Gartner eta Forrester erakundeek txostenetan, CMS atalean, liderra den Acquia enpresak Drupal erabiltzen du oinarri bezala.

Proiektu honetan saiakera bat egin da BPM eta workflow-lengoaiarik erabili gabe. Izan ere, eredutik sortutako datu-basea eta Drupal-eko moduluak bakarrik erabili dira.

6. Mekanismo arkitekturelak

- **Prozesuak kudeatu:** Jarraitu beharreko prozesua bistaratuko da, bizi-zikloko zehaztasun guztiekin: faseak, iterazioak, jarduerak, atazak etab. Prozesu ingeniariak prozesuan aldaketak egiteko aukera izango du.
- **Proiektuak kudeatu:** Proiektu kudeatzaileak proiektuak sortu eta aldatzeko aukera izango du. Proiektuan kideak gehitu beharko ditu prozesuaren rola betetzeko.
- **Artefaktuak idatzi:** Erabiltzaile bakoitzak bere rolari dagozkion artefaktuak bete beharko ditu txantiloietan oinarrituta. Artefaktu horiek DOC eta PDF dokumentu moduan igoko dira.

7. Geruzak edo arkitektura frameworka

IO-System azpisistemak honako osagaiak izango ditu.

- **Datu-basea:** Datu-base erlazionalak ereduaren informazio garrantzitsuen gordeko du, Drupal webgunerako beharrezkoa izan daitekeena. Hau da, OpenUP eta ABRD metodologiaren faseak, iterazioak, jarduerak, atazak, artefaktuak, rola, etab.
- **Drupal datu-basea:** Drupal sistemaren datu-basea eduki guztia gordetzeaz arduratzen da, fitxategiak izan ezik. Adibidez, erabiltzaileak, rola, baimenak eta eduki motak gordetzeaz.
- **Drupal datu inportatzailea:** Datu-inportatzailea Drupal modulu multzo bat izango da. Hauen ardura aurretik aipatutako lehenengo datu-basetik Drupal datu-basera edukia inportatzea da. Horretarako, nodoak sortu beharko dira, eta Drupal arduratuko da edukia gordetzeaz.
- **Drupal fitxategiak:** Fitxategiak datu-basetik kanpo gordeko dira. Gure kasuan fitxategi gehienak artefaktuei dagozkienak izango dira, DOC eta PDF dokumentuak.
- **Drupal interfazea:** Interfazean edukia bistaratu eta aldatzeko aukera guztiak egongo dira. Esan bezala, edukia Drupal datu-basean gordeko da. Gainera, erabiltzailearen kontuekin zerikusia duten aukerak ere egongo dira. Horrez gain, administratzaileak aukera gehiagarri asko izango ditu webgunea kudeatzeko interfaze bidez.

