

Informatika Fakultatea

Informatika Ingeniaritzako Gradua

▪ Gradu Amaierako Lana ▪

Software Ingeniaritza

Business Intelligence kultura-kudeatzaile entitate batean

Jokin Indias Alza

2020- otsaila



Laburpena

Munduan konexio gehiago daude gizakiak baino, eta bai edo bai, gure bizitzeko eta erlazionatzeko moduak dakarren hazkunde teknologikoak, ezinbesteko mugarri bat markatu du, datuak dituenak zerbait oso baliotsua du. Adibidez, Paypal, Booking, Uber, Spotify, ... ez dira ez bankuak, ez hotel bat, ezta taxi bat, ezta ere diskografia bat, baina datuen jabe dira eta ustiatzen eta zerbitzatzen dute diru asko irabazteko.

Egoera honen aurrean, **DATUAN** pentsatzen hasi behar da eta Donostia Kulturak biltzen dituen datuak mimatu. Estrategia bat da ezin dena segundo bat ere atzeratu.

Donostia Kulturak arazo larri bat du beraiekin nolabaiteko harremana duten pertsonen eta entitateen **datuen sakabanatzearekin**. Sakabanatze hau datuen esplorazioa zailtzen du, **business intelligence** erremintekin.

Gaien Aurkibidea

Laburpena.....	ii
Gaien aurkibidea.....	iii
Irudi eta taulen zerrenda.....	vii
1. Sarrera	1
1.1 Donostia Kultura.....	2
1.2 Adimedia Comunicación Interactiva.....	2
1.3 Landutako proiektua.....	3
2. Plangintza	5
2.1 Irismena.....	6
2.1.1 Betebeharrak.....	6
2.1.2 Mugak.....	6
2.1.3 Proiektuaren irismena.....	7
2.1.4 Irismenaren kudeaketa.....	8
2.1.5 Proiektuaren deskonposaketa.....	8
2.2 Lan-metodologia.....	10
2.3 Emangarri nagusiak.....	10
2.4 Baliabideak.....	11
2.4.1 Software baliabideak.....	11
2.4.2 Hardware baliabideak.....	11
2.5 Plangintza.....	11
2.5.1 Egutegia.....	11
2.5.2 Gantt diagrama.....	12
2.5.3 Lan-karga.....	15
2.5.4 Mugarriak.....	15
2.6 Kudeaketa plana.....	15
2.6.1 Komunikazio plana.....	15
2.6.2 Arriskuak.....	16
2.6.3 Aldaketak.....	18
2.6.4 Kalitatea.....	19
2.6.4.1 Kalitatearen adierazleak.....	19
2.6.4.2 Kalitatea ziurtatzea.....	19
2.6.4.3 Kalitate kontrola.....	19

3. Oinarri tekniko-teknologikoak	21
3.1 Business Intelligence.....	22
3.1.1 Abantailak.....	22
3.1.2 Desabantailak.....	22
3.2 Data Warehouse.....	23
3.2.1 OLAP.....	24
3.2.2 Modelo dimentsionala.....	25
3.2.3 Pixkanaka aldatzen diren dimentsioak.....	27
3.3 ETL prozesua.....	27
3.3.1 Ateratzea.....	28
3.3.2 Eraldatzea.....	29
3.3.3 Kargatzea.....	30
4. Informazio-bilketa	33
4.1 ETL prozesua.....	34
4.1.1 ETL erremintak.....	34
4.1.1.1 Oracle Warehouse Builder.....	34
4.1.1.2 Informatica PowerCenter.....	35
4.1.1.3 Talend Open Source Integration Software.....	35
4.1.1.4 Pentaho Data Integration- Kettle.....	36
4.1.2 Kodea sortzeko teknologiak.....	36
4.1.2.1 JavaScript (Node.js).....	37
4.1.2.2 Python.....	37
4.1.2.3 Java.....	38
4.1.2.4 PHP.....	39
4.2 Data Warehouse.....	39
4.2.1 Teradata.....	39
4.2.2 Oracle.....	40
4.2.3 Amazon Web Services (AWS).....	41
4.2.4 MySQL.....	41
4.2.5 PostgreSQL.....	42
4.2.6 Panoply.....	42
4.2.7 SAS.....	43
4.3 Datuen sarbiderako APIa.....	43
4.3.1 Protokoloak.....	43
4.3.1.1 SOAP.....	44
4.3.1.2 REST.....	44
4.3.1.3 GraphQL.....	45
4.3.1.4 OData.....	45
4.3.2 Informazio trukaketa.....	46
4.3.2.1 XML.....	46
4.3.2.2 JSON.....	46
4.3.2.3 YAML.....	47
4.3.3 APIaren garapen teknologia.....	47
4.3.3.1 Java.....	48
4.3.3.2 JavaScript.....	49
4.3.3.3 Python.....	49

4.3.3.4	PHP.....	49
4.3.4	Merkatuan dauden APIak.....	50
4.3.4.1	Teradata.....	50
4.3.4.2	Oracle.....	50
4.3.4.3	MySQL.....	50
4.3.4.4	SAS.....	51
4.4	Kudeaketa aplikazioa.....	51
4.4.1	PHP.....	51
4.4.1.1	Medoo.....	52
4.4.1.2	Laravel.....	52
4.4.1.3	CodeIgniter.....	52
4.4.2	Java.....	53
4.4.2.1	Spring MVC.....	54
4.4.2.2	Struts 2.....	54
4.4.2.3	Hibernate.....	54
4.4.3	JavaScript.....	54
4.4.3.1	Express.....	55
4.4.3.2	Koa.....	55
4.4.3.3	React.....	56
4.4.3.4	Angular.....	57
4.5	Datu-garbiketa.....	57
4.5.1	Datu-garbiketa erremintak.....	57
4.5.1.1	Talend Open Source Integration Software.....	57
4.5.1.2	Pentaho Data Integration- Kettle.....	57
4.5.1.3	Winpure.....	58
4.5.1.4	Data Ladder.....	58
4.5.2	Datu-garbiketarako garapen teknologiak.....	59
4.5.2.1	Pandas.....	59
4.5.2.2	Dplyr.....	59
4.6	Datu-iturriak.....	59
4.6.1	Ezaugarriak.....	60
4.6.2	Zer datu interesatzen zaizkigu?.....	60
5.	Proposatutako soluzioa	63
5.1	ETL prozesua.....	64
5.2	Data Warehouse.....	64
5.2.1	DBKS.....	64
5.2.2	Modeloaren diseinua.....	65
5.3	Datuen sarbiderako APIa.....	65
5.4	Kudeaketa aplikazioa.....	66
5.4.1	Zerbitzari aldea.....	66
5.4.2	Bezero aldea.....	66
5.5	Datu-garbiketa.....	67
5.6	Datu-iturriak.....	67
6.	Soluzioaren alternatiba	71
6.1	ETL prozesua.....	72

6.2	Data Warehouse.....	72
6.3	Datuen sarbiderako APIa.....	72
6.4	Kudeaketa aplikazioa.....	73
6.5	Datu-garbiketa.....	73
6.6	Datu-iturriak.....	73
7.	Jarraipena eta kontrola	75
7.1	Burututako lana.....	76
7.1.1	Desbiderapenak.....	76
7.1.2	Bukaerako Gantt diagrama.....	77
7.2	Arriskuak.....	79
7.3	Aldaketak.....	79
7.4	Kalitatea.....	80
8.	Ondorioak.....	81
8.1	Proiektuaren inguruko ondorioak.....	82
8.2	Ondorio pertsonalak.....	82
8.3	Etorkizunerako gomendioak.....	83
	Bibliografia.....	85
	A Eranskina. Bilera aktak.....	87

Irudi eta Taulen zerrenda

IRUDIAK

2.1 irudia	LDE eskema.....	8
2.2 irudia	Gantt diagrama.....	14
3.1 irudia	OLAP kuboak.....	25
3.2 irudia	Izar modeloa.....	25
3.3 irudia	Elur-maluta modeloa.....	26
3.4 irudia	Galaxia modeloa.....	26
5.1 irudia	Datu biltegiaren modeloaren diseinua.....	65
7.1 irudia	Bukaerako Gantt diagrama.....	78

TAULAK

2.1 taula	Egutegia	13
2.2 taula	Mugarriak.....	15
2.3 taula	Interesatuaren taula.....	16
3.1 taula	Kargatzeko metodoen ezaugarriak.....	30
4.1 taula	PHP eta Java konparazioa.....	53
5.1 taula	xx_socios_eki.....	67
5.2 taula	dk_acymailing_subscriber.....	68
5.3 taula	dk_acymailing_listsub.....	68
5.4 taula	dk_acymailing_lists.....	69
5.5 taula	dk_content.....	69
5.6 taula	dk_flexicontent_fields_item_relations.....	69
5.7 taula	dk_flexicontent_field.....	70
5.8 taula	xx_socios_sorteos.....	70
7.1 taula	Ordu desbideraketak.....	76

1

Sarrera

Kapitulu honetan *Donostia Kultura* (arazoa duen entitatea) eta *Adimedia Comunicación Interactiva*-ren (arazoa konponduko duen enpresa) nondik norakoa azalduko da. Baita ere proiektu honetan jorratuko denaren sarrera bat.

1.1 DONOSTIA KULTURA

Donostia Kultura Donostiako Udalaren kultur eragile nagusia da. Hainbat udal-ekipamendu eta udal-zerbitzu kudeatzeaz gain, kultura-diziplina guztietan programazioa eta jaialdiak antolatu eta kultur ekimen berriak garatzen ditu, eragile pribatuekin zenbaitetan. Donostia Kultua Enpresa Erakunde Publiko bat da.

Herritar orori eskuratzeko aukera eskaini eta kultura hirian barrena hedatzeko xedez, Donostia Kulturak 18 zentro kudeatzen ditu, San Telmo Museoa eta Victoria Eugenia Antzokia bezalako erakin ezagunetatik hasi eta herritarrengandik gertu dauden eta askotan beren auzoetatik haratago doazen zentro eta liburutegien sareraino. Hona hemen eskaintzen dituen zerbitzu eta jarduerak:

- Antzerkia eta dantza
- Musika
- Zinema eta bideoak
- Jaiak
- Kultur Ekintza Zerbitzua
- Liburutegi Zerbitzua
- San Telmo Museoa

Donostia Kulturak 220 langile baino gehiago ditu. Jarduera eta zerbitzu guztiak ongi garatzeko, honako hiru unitate zentral hauek ditu: Zerbitzu Orokorrak, erakundeko administrazio-gaietaz arduratzeko, Komunikazio eta Irudi Unitatea, programazioa eta Donostia Kulturaren irudia zabaltzeko, bazkideak kudeatzeko eta abar. Ekoizpena, azkenik, programazio osoari euskarri tekniko, logistiko eta antolaketari dagokionaz arduratzen da.

1.2 ADIMEDIA COMUNICACIÓN INTERACTIVA

Adimedia zerbitzu digitalen eta kultura digital agentzia geldiezin bat da, profil sortzaile eta teknologiko nabarmena duen ekipo espezializatu batekin. Negozio, denda, enpresa, erakunde, zerbitzu, ... Interneteko presentzia erraza izatea ahalbidetzen dute, pertsonak-proiektuak-markak batzeko.

Euskadiri webgune erakargarriak, praktikoak eta baliotsuak eskaintzea da Adimediaren misioa. Eta hau guztia lortzeko haxe de beraien kultura:

- **Konpromisoa:** beraien bezeroekin, lagunekin eta hornitzaileekin, ikuspegi kolaboratibo eta garden batekin.
- **Eboluzioa:** bazkide proaktibo gisa, aurreratzea eta profesionaltasuna erakutsiz.
- **Sendotasuna:** haien planteamendu eta emaitzen kalitatean, balioan, ezagueran, haien esperientziari esker.

1.3 LANDUTAKO PROIEKTUA

Donostia Kulturak behar bat dauka eta beraz arazo bat dauka. Beharra, dituen datu guztiak analizatu eta ebaluatu. Baina hori gauzatzeko arazo bat dauka, datuen sakabanatzea. Lehen azaldu bezala, zerbitzu eta eskaera asko eskaintzen ditu, toki askotan, eta horregatik datuak oso sakabanatuta daude. Sakabanatze hau, datuen analisia asko zailtzen du, beraz Adimediatic soluzio bat eskaini zaio Donostia Kulturari.

2

Plangintza

Proiektua aurrera eramateko egin den plangintza azalduko da kapitulu honetan.

2.1 Irismena

Donostia Kultura pertsona eta entitateen datu banaketarekin duen arazoa konpondu.

Banaketa honek datu ustiapena zailtzen du, Donostia Kulturaren zerbitzuak hobetzeko.

2.2.1 Betebeharrak

Bi betebehar nagusi daude:

- Alde batetik, datu-biltegi bakar bat sortzea. Ingelesez data warehouse bezala agertzen da. Datu biltegi horretan kalitatezko, baliagarriak eta ordenatuak diren datuak egongo dira. Donostia kulturak datu-sakabanapena dauka, hau da, ez ditu datu guztiak toki berdinean gordetzen, baizik eta jatorriaren arabera leku eta teknologia desberdinekin gordetzen da. Beraz atal honek daukan betebeharra da datu guzti horiek batzea modu logiko eta ordenatu batean, toki bakar batean.
- Beste alde batetik, datuen ustiapena. Donostia Kulturak dauzkan datu guztiak ustiatu nahi ditu, beraien bazkideei zerbitzu hobeagoak eskaintzeko. Horretarako “business intelligence” erremintak erabiliko dira. Honekin lortzen duguna da datuen estatistika zabalak sortzea, eta honekin lortzen dena da, Donostia Kulturak hartu behar dituen erabakiak hobeagoak eta zehatzagoak izatea.

Bi betebehar hauek erlazionaturik daude, hau da, bata bestearen beharra dute. Donostia Kulturak “business intelligence” tresnak erabiltzen hasi nahi du, eta horretarako ezinbestekoa da kalitatezko datu biltegi bat izatea, bestela datu-sakabanaketak “business intelligence” egitea asko zailtzen du, eta gainera gerta daiteke lortutako emaitzak ez izatea errealak, hau da, erroreak edukitzea edo datu batzuk ez hartzea edo gaizki hartzea emaitzen eraketarako.

2.2.2 Mugak

Hasiera batean aurkitu dugun muga bakarra denbora da. Proiektua 300 ordukoa izan behar da, beraz oso mugatuak egongo gara alde horretatik. Proiektua aurrera joango den heinean, arazo gehiago suertatuko dira, beraz muga gehiago.

Denborarekin larri egongo garela jakinda, Adimediak lehentasun batzuk jarri ditu proiektuaren inplementaziorako, ordena honetan:

1. Datu-biltegia sortzeko analisisa, hau da, ze pausu egin eta nola egin behar diren datu-biltegia sortzeko.
2. Datu-biltegia sortu eta martxan jarri.
3. Business Intelligence tresna edo trenak datu-biltegian martxan jarri (Tresna hauek Donostia Kulturak aukeratuko ditu).

2.2.3 Proiekturen irismena

Proiektu honen denbora dela eta ezin gara betebeharrak guztiak inplementatzera iristen, beraz Alexis Mindegiarekin izandako bileran 2019/10/11n, hartu dugun erabakia izan da bakarrik aurreko atalean dauden lehentasunetatik bakarrik lehenengoa garatzea, hau da, datu-biltegia sortzeko beharrezkoa den analisia.

Analisi honek gai hauek jorratuko ditu:

- Datu iturriak identifikatu
 - Non daude? (paperean, euskarri fisikoan, online-zerbitzari batean, ...)
 - Zein da jabea?
 - Erabilitako teknologia
 - Zertarako erabiltzen dira?
 - Zein datu dira interesgarriak edo garrantzitsuak?
 - Zer formatu dute datu horiek?
 - Nola atzitu daitezke datuak? (Web-zerbitzuak, API, ...)
 - Sinkronizazio maiztasuna (egunero, minutuero, segundoero, ...)
- Datuen kalitatea
 - Definizioa
 - Zein datu balio dute? Zeintzuk ez?
- Datuen sinkronizazioa: ETL prozesua
 - Zer da ETL prozesua?
 - Zein teknologia erabiliko dugu garapenerako?
 - Framework bat erabiltzea komenigarria da?
- Data Warehouse (Datu-biltegia)
 - Zer da datu-biltegi bat?
 - Zer ezaugarri ditu?
 - Zer abantaila ditu?
 - Zer DBKS erabiliko dugu? (MySQL, PostgreSQL, Oracle, ...)
- Datuen garbiketa
 - Nola eguneratu datuak? (Pertsona bat hil bada, edo entitate bat desegin bada)
 - Nola normalizatu datuak? (datu ez baliagarriak garbitu)
 - Nola gauzatu deduplikazioa? (Datu errepikapenak hauteman eta ezabatu)
- Datuen sarbiderako APIa
 - Zein teknologia erabiliko dugu garapenerako?
 - Zein protokolo erabiliko dugu?

- Informazio trukaketa zein formatutan gauzatuko da?
- Framework bat erabiltzea komenigarria da?
- Datuak kudeatzeko aplikazioa
 - Zein teknologia erabiliko dugu garapenerako?
 - Framework bat erabiltzea komenigarria da?

Beraz proiektu hau proiektu handi baten analisisa izango da, ondoren proiektu osoa inplementatzeko balio behar izan duena.

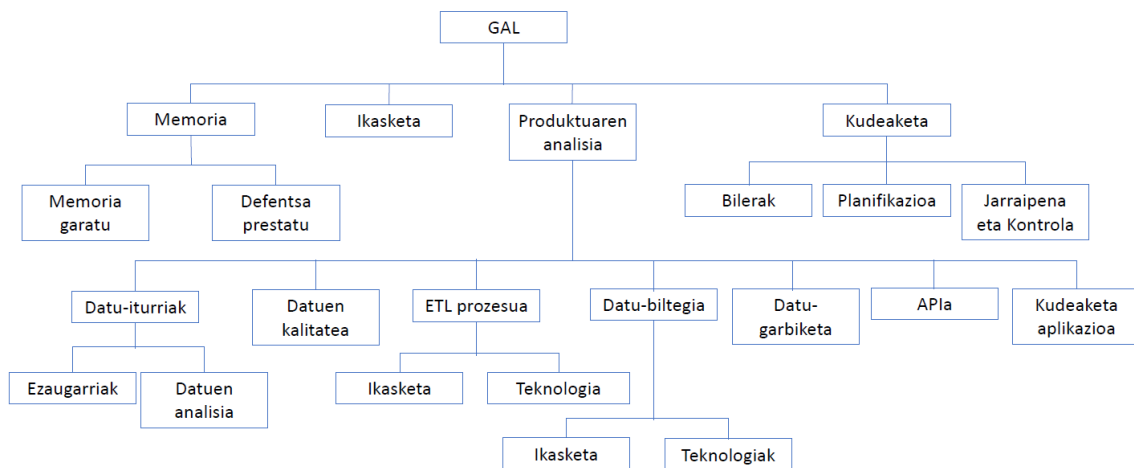
2.2.4 Irismenaren kudeaketa

Gerta daiteke hasiera batean adostutako irismena ezin izan betetzea. Denboragatik, atzerapenengatik, edo bezeroak beren helburuak aldatzen dituelako. Aldaketa hauek proiektuaren irismena aldatzea ekar dezake, eta beren mugak aldatzea ere.

Baina esan beharra dago proiektuaren arrakasta dela gure helburu nagusia, eta aldaketa guztiak horren inguruan egon behar dira. Beraz, aldaketa bat egonez gero, plangintza osoa berregin egin beharko zen, irismen eta muga berrietara egokituz.

2.2.5 Proiektuaren deskonposaketa

LDE-ak proiektuaren irismena definitzen du eta lana ataza eta azpiatazetan banatzen du, era hierarkiko batean. Azpiataza bakoitza programatu eta gainbegiratu daiteke, baita bere kostuak estimatu ere. Hasiera batean planteatu dugun LDA honako hau da:



2.1 Irudia: LDE eskema

Gradu amaierako lana lau azpiataletan banatu da, eta azpiatal horiek azpiatal gehiagotan banatu dira. Hemen azpiatal horien azalpen bat:

- **Memoria:** Gradu Amaierako Lanean garatu beharreko memoria eta defentsaren prestakuntza.
 - **Memoria garatu:** Proiektuaren memoriaren garapena. Non proiektuaren irismena, garatu den analisi osoa, jarraipena eta eta kontrola, eta ondorioak azalduko diren
 - **Defentsa prestatu:** Fakultateko Tribunalaren aurrean aurkeztu beharreko memoria honen defentsaren prestaketa.
- **Ikasketa:** business intelligence inguruko kontzeptuak ikasi eta barneratu.
- **Produktuaren analisia:** produktua inplementatu ahal izateko beharrezkoa den analisia. Analisi honetan hainbat gai jorratuko dira, produkturen kalitatea ahal den handiena izateko. Gai horien artean datu-iturriak, ETL prozesua edo datu biltegia daude.
 - **Datu-iturriak:** datu-biltegiak izango dituen datu iturrien analisia.
 - **Ezaugarriak:** datu-iturri bakoitzak dituen ezaugarriak, besteak beste, non dagoen kokatua edo zer teknologia erabiltzen duen.
 - **Datuen analisia:** datu-iturri bakoitzaren datuen analisia. Atal honetan ikusiko dugu zer datu hartuko ditugu eta zein maiztasunarekin sinkronizatuko dugu gure datu-biltegiarekin.
 - **Datuen kalitatea:** atal honetan erabakiko dugu, Donostia Kulturarekin batera, zein datu gorde behar diren datu-biltegiak eta zer formatutan gordeko diren.
 - **ETL prozesua:** datu-biltegia eguneratua izateko beharrezkoa den ETL prozesua nola gauzatuko den.
 - **Ikasketa:** ETL prozesu bat nola egiten den.
 - **Teknologia:** zer teknologia erabiliko dugu ETL prozesua inplementatzeko.
 - **Datu-biltegia:** datu-biltegiak izango dituen ezaugarriak.
 - **Ikasketa:** datu-biltegi batek dituen ezaugarriak ikasi.
 - **Teknologia:** zer teknologia erabiliko dugu datu-biltegia garatzeko.
 - **Datu-garbiketa:** datu desegokien garbiketa nola gauzatuko den analisia.
 - **APIa:** datu-biltegia atzitzeko beharrezkoa den APIa garatzeko behar den analisia.
 - **Kudeaketa aplikazioa:** datu-biltegia kudeatzeko beharrezkoa den aplikazioa garatzeko behar den analisia.
- **Kudeaketa:** produktuaren plangintza eta azpiatalen jarraipena eta kontrola. Baita ere proiektuaren arriskuen kudeaketa-plana edo kalitate plana atal honetan jorratuko dira.
 - **Bilerak:** unibertsitean naiz enpresan egin diren bilerak.
 - **Planifikazioa:** proiektua garatzeko beharrezkoa den planifikazioa.
 - **Jarraipena eta Kontrola:** proiektua aurrera joan ahala egin behar den jarraipena eta kontrola, soluzioaren kalitatea bermatzeko.

2.2 Lan-metodologia

Proiektua aurrera eraman ahal izateko oso garrantzitsua da metodologia bat ondo garatzea. Metodologia hau akatsak ekiditeko balio behar digu, horrela analisiaren kalitate hobereana lortzen.

Horretarako proiektua ataletan banatu dugu, aurreko puntuan azalduta, horrela atalka proiektua garatzen joango gara. Honekin lortu nahi duguna da denboraren kontrol handiagoa, denbora-desbiderapenak ahal den guztia ekidituz, eta atal bakoitzaren jarraipen optimoena lortzen saiatu.

Esan bezala, kalitate hobereana lortzea da helburua, eta horretarako atalak banaka inplementatuko ditugu, eta ez gara beste batekin hasiko garatzen hari garena bukatu arte. Baita ere, ataza bakoitzaren bukaeran emangarri bat sortuko dugu, egindako analisiaren informazio guztiarekin. Emangarri horiek Adimediaren eskutan jarriko dira, behin emangarria bukatutakoan. Beraz atal bakoitzaren bukaeran honako hau egingo dugu:

1. Egin dugun lanaren balorazioa: aurretik finkatutako helburuak bete diren hausnartu eta egindako lana baliagarria den proiektuarentzat ikusi. Hau da, egindako analisia egokia dela proiektu honentzat, eta aurrerago, inplementatzerako garaian, analisi honek balio digun.
2. Hurrengo ataleko helburuak finkatu: argi eduki zer lortu nahi dugun, eta beraz analisia ildo horretatik garatu.
3. Garapena: helburuak argi edukita, analisia inplementatu.
4. Dokumentazioa: analisi guztia dokumentu batean plasmatu, modu argi eta antolatuta batean, edozeinek ulertu ahal izateko.

2.3 Emangarri nagusiak

Proiektuaren emangarriak bi talde nagusitan banatu ditugu:

- **Analisia eta ikasketa:** lehenago aipatu bezala proiektuaren analisia ataletan banatu ditugu, eta atal bakoitzak emangarri bat izango du, analisi guztiarekin. Baieta ere, aurre ikasketa bat egingo da, eta baita ere dokumentu bat izango du.
- **Memoria:** memoriarekin erlazionatutako emangarriak.
 - **Memoria:** dokumentu hau bera, non proiektu osoaren nondik norakoak azalduko dira, batik bat, plangintza, jarraipena eta kontrola, gauzatutako analisia, ondorioak eta abar...
 - **Aurkezpena:** proiektu hau bera defendatzerako garaian erabiliko den euskarria.

2.4 Baliabideak

2.4.1 Software baliabideak

Proiektu honetan ez dugu azken produktu bat garatu, beraz, ez dugu kodea garatu. Horregatik ez dugu software baliabide handirik erabili, soilik funtsezko softwarea.

Alde batetik, analisi osoa dokumentatzeko LibreOffice 5.4.2.2 dokumentu editorea erabili dugu, Windows 10 sistema eragilearen gainean. Beste alde batetik PHPMyAdmin erabili dugu datu-iturriak aztertzeko. Software hau nabigatzaile batean exekutatzen da, eta horretarako Google Chrome nabigatzailea erabili dugu.

Baita ere, enpresako kideekin koordinazioa izateko eta plangintzaren kontrola izateko Google Drive-eko erremintak erabili dira, horrela dokumentuak eta kalkulu orriak konpartitzeko eta edozein lekutatik atzigarri izateko aukera dugu.

Azkenik, memoria idazteko, baita ere LibreOffice erabili dut, beste dokumentazio guztia euskarri honekin erabili dugulako. Eta azkenik aurkezpena gauzatzeko Microsoft Office PowerPoint erabili dut.

2.4.2 Hardware baliabideak

Kasu honetan ere, ez dugu hardware baliabide handirik erabili, arrazoi berdinegatik.

Hasteko, ezinbestekoa denez, konputagailua. Bi konputagailu erabili dira proiektua aurrera eramateko. Lehenik, enpresak emandako mahaigaineko ordenagailua, non analisi guztia gauzatu den. Bigarrenik, ordenagailu pertsonala, eramangarria, non memoria eta aurkezpena garatzeko erabili den.

Baita ere, Adimediako zerbitzari baten erabilera baimenak izan ditut, dokumentazio guztiaren segurtasun kopia bat izateko, eta enpresako lankide guztiak eskuragarri zezaten dokumentazio osoa.

2.5 Plangintza

2.5.1 Egutegia

375 orduko proiektu bat izango da, non 300 ordu enpresan egin diren eta 75 ordu dokumentazio guztia garatzen eta antolatzen. Lan-metodologian azaldu den moduan, lana atalka egingo da, horrela denbora handi bat zati txikiagotan banatzen dugu eta denbora hobetu kontrola dezakegu, desbideraketa gutxienak izaten saiatuz.

Proiektua gauzatzeko honako estimazioa egin da, [2.1 taula](#) honetan agertzen da.

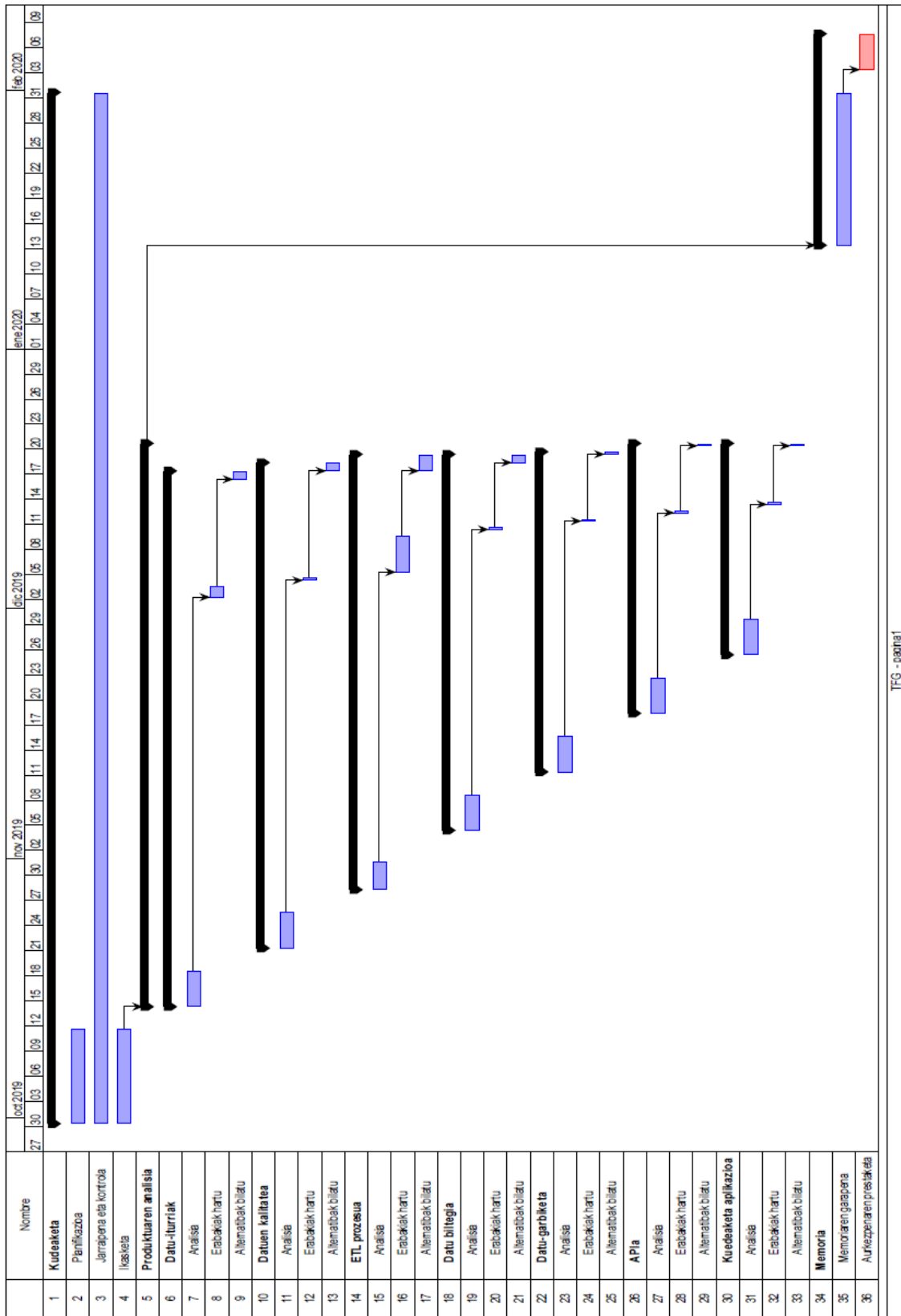
Ikus dezakegu, proiektu osoa 375 ordu behar dituela aurreikusten dugula, ordu horiek 78 egunetan banatuak. 2019ko irailaren 30ean hasiera eman genion proiektuari eta 2020ko otsailaren 7an bukatuta egon behar du.

2.5.2 Gantt diagrama

Aurreko planifikazioarekin batera, Gantt diagrama bat egin dugu. Honekin, denboran ikus dezakegu nola gauzatuko diren atalak, eta haien arteko dependentziak. Garatu egin digun Gantt diagrama [2.2 Irudian](#) azaltzen da.

Kodea	Atazaren izenburua	Estimazioa (orduak)	Hasiera	Amaiera	Irupena (egunak)
1	Kudeaketa	32	2019/09/30	2020/02/01	83
1.1	Planifikazioa	10	2019/09/30	2019/10/11	10
1.2	Jarraipena eta kontrola	22	2019/09/30	2020/02/01	73
2	Ikasketa	36	2019/09/30	2019/10/11	10
3	Produktuaren analisia	234	2019/10/14	2019/12/20	50
3.1	Datu-iturriak	38	2019/10/14	2019/09/16	8
	Analisia	23	2019/10/14	2019/10/18	5
	Erabakiak hartu	10	2019/12/02	2019/12/03	2
	Alternatibak bilatu	5	2019/12/16	2019/09/16	1
3.2	Datuen kalitatea	31,5	2019/10/21	2019/12/17	6,5
	Analisia	24	2019/10/21	2019/10/25	5
	Erabakiak hartu	5	2019/12/04	2019/12/04	1
	Alternatibak bilatu	2,5	2019/12/17	2019/12/17	0,5
3.3	ETL prozesua	38	2019/10/28	2019/12/18	8
	Analisia	24	2019/10/28	2019/11/01	4
	Erabakiak hartu	9	2019/10/05	2019/12/09	2
	Alternatibak bilatu	5	2019/12/17	2019/12/18	2
3.4	Datu-biletegia	33	2019/11/04	2019/12/19	8
	Analisia	24	2019/11/04	2019/11/08	5
	Erabakiak hartu	5	2019/12/10	2019/12/10	1
	Alternatibak bilatu	4	2019/12/18	2019/12/19	2
3.5	Datu-garbiketa	30,5	2019/11/11	2019/12/19	6,5
	Analisia	24	2019/11/11	2019/11/15	5
	Erabakiak hartu	4	2019/12/11	2019/12/11	1
	Alternatibak bilatu	2,5	2019/12/19	2019/12/19	0,5
3.6	APIa	31,5	2019/11/18	2019/12/20	6,5
	Analisia	24	2019/11/18	2019/11/22	5
	Erabakiak hartu	5	2019/12/12	2019/12/12	1
	Alternatibak bilatu	2,5	2019/12/20	2019/12/20	0,5
3.7	Kudeaketa aplikazioa	31,5	2019/11/25	2019/12/20	6,5
	Analisia	24	2019/11/25	2019/11/29	5
	Erabakiak hartu	5	2019/12/13	2019/12/13	1
	Alternatibak bilatu	2,5	2019/12/20	2019/12/20	0,5
4	Memoria	73	2020/01/13	2020/02/07	20
4.1	Mermoriaren garapena	58	2020/01/13	2020/01/31	15
4.2	Aurkezpenaren prestaketa	15	2020/02/03	2020/02/07	5
		375	2019/09/30	2020/02/07	78

2.1 Taula: Egutegia



TFG - pagin 1

2.2 Irudia: Gantt diagrama

2.5.3 Lan-karga

Aurreko ataletan ikusi den moduan, proiektua 78 egunetan banatzen da, 375 ordu betez. Eguneroko lan-karga berdina da, enpresarekin adostutako ordutegia betetzen delako. Egunero 5 orduko lan-karga dago goizeko zortzitatik eguerdiko ordu batetara arte. Baita ere esan beharra dago enpresan irailaren 30etik, abenduaren 20ra arte egon naizela, proiektua garatzen. Ondoren memoria etxetik egin dut, baina honen dedikazioa aurreko berdina izan da, 5 ordu egunero.

Lan-dedikazioa ez da handiagoa izan beste ikasgaiak eta unibertsitate kanpoko kurtso bat dela medio. Ez dugu ahaztu behar proiektu honen erdian Gabonak egon direla, beraz oporretarako etenaldi bat egon da, Gantt diagraman agertzen den moduan, abenduaren 20tik urtarrilaren 13ra arte.

Proiektua, nahiz eta produktuaren betebeharrak guztiak ez bete, ordu dedikazio handia behar du, 375 ordu estimatzen direlarik, proiektua nahiz memoria garatzeko.

2.5.4 Mugarriak

Esanguratsuak diren epe-mugak hauek dira:

Mugarria	Data
Proiektuaren hasiera	2019/09/30
Dokumentazioaren hasiera	2019/09/30
Dokumentazio osoaren entrega	2019/12/20
Memoriaren garapenaren hasiera	2020/01/13
Memoriaren behin-behineko bertsioa	2020/01/22
Defentsaren eskaera	2020/01/24
Memoriaren behin-betiko bertsioa	2020/02/02
Memoria entregatu	2020/02/02
Proiektuaren defentsa	2020/02/10-2020/02/14

2.2 Taula: Mugarriak

2.6 Kudeaketa plana

2.6.1 Komunikazio plana

Proiektu honen interesatuak honako hauek dira:

Interesatua	Rola	Kontaktu-informazioa
Imanol Usandizaga Lombana	Irakaslea/Zuzendaria	imanol.usandizaga@ehu.eus
Juanjo Pascual Coca	Enpresako tutorea	info@adimedia.net
Donostia Kultura	Enpresa bezeroa	-
Jokin Indias Alza	Ikaslea/garatzaila	jindias001@ikasle.ehu.eus

2.3 Taula: Interesatuen taula

Komunikazio plana bi taldetan banatu dugu, modu honetan:

- **Gradu amaierako lanaren kudeaketaren komunikazio plana**

Komunikazio plan honetan bi bide erabiliko ditugu komunikazioak gauzatzeko. Alde batetik posta elektronikoa erabiliko dugu, edozein zalantza proiektuaren zuzendariarekin harremanetan jartzeko. Beste alde batetik, zalantza handiren bat edo arazo handiren bat badago, zuzendariarekin bilera bat egongo da. Lehenik, posta elektronikoa bidez noiz eta non izango den bilera adostuko da eta ondoren bilera gauzatu da.

Ikaslearen lana izango da bilera hauen aktak betetzea.

- **Produktuaren kudeaketaren komunikazio-plana**

Produktuaren garapenarekin zerikusik duen komunikazioak planifikatu ditugu. Hasteko komunikazio hauek guztiak enpresako tutorearekin izango dira. Alde batetik posta elektronikoa bidez, eta beste alde batetik bilera presentzialen bidez, enpresan bertan. Bata elkarren ondoan lan egiteak komunikazioa asko errazten du.

Edozein zalantza edo arazo edukiz gero, eta horiek bezeroarekin kontsultatu behar badira, ni ez naiz zuzenean haiekin komunikatu behar. Baizik eta lehenik eta behin tutoreari arazoa azaldu, eta berak kudeatu egingo du. Honek enpresako beste lankide batekin harremanetan jarriko da arazo hori konpontzeko edo zuzenean bezeroarekin harremanetan jarriko da.

Edozein bilera eduki ezker, ikaslea izango da aktak idaztearen arduraduna.

2.6.2 Arriskuak

Hainbat arrisku aurreikusten ditugu proiektu honetan, nahiz eta planifikazio bat egin dugun hau ez gertatzeko, gerta daiteke ondoko arazoren bat gertatzea:

- **Denboraren aurreikuspen txarra:**

Nahiz eta planifikazio zehatz bat eginda, gerta daiteke ordu horiek gaizki estimatzea. Hau da, ordu gutxiago behar izatea, edo ordu gehiago behar izatea. Hau ez gertatzeko planifikazio garaian oso zehatzak izaten saiatu behar gara, beste era batean esanda, zehaztapen guztiak kontuan hartu behar ditugu, adibidez, langile kopurua, gure eskura ditugun baliabideak,

langileak duten jakintza eta esperientzia, besteen artean. Denbora desbiderapen bat gertatuz gero, jarraipen eta kontrol txostenean islatu beharko genuke, eta ondoren planifikazioa berrantolatu, proiektua epean bukatzeko.

- **Kategoria:** Kudeaketa
- **Eragina:** Oso handia
- **Lehentasuna:** Oso handia
- **Probabilitatea:** Ertaina

- **Bezeroaren konpromiso falta:**

Gure proiektuan, aurretik azaldutako moduan, bezeroaren datu-iturrietan baliatu behar gara datu-biltegia sortzeko. Beraz datu-iturri horien informazioa behar dugu, non dagoen kokatuta edo zer teknologia erabiltzen duten adibidez. Informazio guzti hori bezeroak eman behar digu gure analisia kalitatezkoa izan dadin. Gerta litekeena da, bezero honek ez ematea eskatutako informazioa edo denbora asko pasatzea informazioa eman arte. Honek baita ere planifikazioan du eragina, analisi bat egiterako garaian ez badugu beharrezko informazio guztia, ezin dugu lan hori burutu, beraz planifikazioa berrantolatu beharko zen eta jarraipen eta kontrol txostenean islatu.

- **Kategoria:** Garapena
- **Eragina:** Handia
- **Lehentasuna:** Handia
- **Probabilitatea:** Txikia

- **Teknologien ezagutza falta:**

Nahiz eta ez erabili teknologia askorik eta oso konplexurik, gerta daiteke teknologiaren bat ez ezagutzea, eta beraz aurreikusitako denbora baino gehiago erabiltzea ataza batzuetan. Beraz, aurreko arriskuetan bezala planifikazioa berriz antolatu beharko zen, eta arazo eta aldaketa guztiak jarraipen eta kontrol txostenean adierazi.

- **Kategoria:** Garapena
- **Eragina:** Ertaina
- **Lehentasuna:** Ertaina
- **Probabilitatea:** Txikia

- **Komunikazio falta:**

Komunikazio planean agertzen diren interesatuen arteko komunikazio falta egon daiteke. Komunikazio falta hau gerta daiteke komunikazio-baliabideren baten arazoengatik, edota lehen azaldutako bezeroaren konpromiso faltagatik. Adibide bat jarritz, bezeroarekin kontaktuan jarri nahi bagara, posta elektronikoa bat bidaltzen diogu, baina hauek ez badute erantzuten mezu honi, atzerapenak gertatu daitezke planifikazioan. Gerta daiteke ere, komunikazio gutxi

egoteagatik interesatuen artean, gaizki ulertuak egotea, eta horrela proiektua gaizki bideratu, atzerapen handiak sortuz.

- Kategoria: Kudeaketa
- Eragina: Ertaina
- Lehentasuna: Ertaina
- Probabilitatea: Txikia

- **Informazio galerak:**

Gerta liteke bat batean ordenagailua itzaltzea eta momentuan editatzen ari garen dokumentua guztiz ez gordetzea. Beraz azken aldiz gordetako bertsioa izango dugu, eta ordura arte egindakoa galtzea. Horretarako gomendagarria da editatzen ari garen fitxategia aldiro gordetzea.

Baina baita ere gerta daiteke ordenagailua apurtzea eta ordura arte egindako guztia galtzea. Honek proiektua ia zerotik berriz hasia eragiten du. Hau ekiditeko, gomendagarria da segurtasun kopiak egitea. Gure proiektuan egingo duguna da, egunaren bukaeran lan guztia zerbitzari batera igo. Gainera egunean behin, zerbitzari horren segurtasun kopia egiten da. Baita ere, noiz behinka, dokumentu guztiak Google Drive-era igotzen dira, edozein lekutatik atzigarri izateko.

- Kategoria: Kudeaketa/Garapena
- Eragina: Handia
- Lehentasuna: Handia
- Probabilitatea: Txikia

2.6.3 Aldaketak

Proiektu bat garatzen hari garen bitartean ohikoa da aldaketaren bat egin behar izatea. Aldaketa hauek ondo kudeatzea oso garrantzitsua da, gaizki kudeatzen badira arrisku oso handia bihurtu daitekeelako, eta beraz, proiektuaren kalitatea asko jaitea. Baita ere, esan beharra dago, aldaketa hauek egoki kudeatzen badira, oso onuragarria izan daitekeela proiektuarentzat. Horregatik aldaketaren bat egotea ez du suposatzen proiektuaren porrota, betiere aldaketa ondo kudeatzen bada.

Kudeaketa hori egokia izateko pausu hauek jarraituko ditugu:

1. Aldaketa beharra identifikatu.
2. Aldaketa eskaera gauzatu.
3. Aldaketa eskaera ebaluatzen da, proiekturen zuzendariarekin edo enpresako tutorearekin.
4. Aldaketa horren inguruan erabakiak hartu.
5. Aldaketa gauzatu, aldaketa aurrera eramateko erabakia hartu bada.
6. Aldaketa dokumentatu, jarraipen eta kontrol txostenean.

2.6.4 Kalitatea

Proiektu baten ezinbesteko betebeharrak bat kalitatea da. Gure proiektuan baita ere kalitatea bermatu nahi dugu. Horretarako kalitatea hiru azpiataletan banatuko dugu:

- Kalitatearen adierazleak
- Kalitatearen ziurtatzea
- Kalitatearen kontrola

2.6.4.1 Kalitatearen adierazleak

Proiektuak kalitate on bat duela adierazten duten adierazleak honako hauek dira:

- Plangintzaren jarraipena: aurreikusitako denborak eta mugak zehazki betetzea.
- Dokumentuen tamaina: dokumentuak oso handiak direla ez du zehazki esan nahi analisiak kalitate handiagoa duenik, baina bai izan daiteke informazio asko bildu egin dela, beraz analisi aberatsago bat egin dela.
- Dokumentuen argitasuna: analisiak modu koherente eta argi batean dokumentatua egotea.
- Analisiaren koherentzia: analisia gure arazoan bideratu dela argi ikustea, eta ez egotea beharrezkoa ez dugun informaziorik.

2.6.4.2 Kalitatea ziurtatzea

Aurreko adierazleak betetzen direla egiaztatuko dira. Horretarako aldiro konprobaketak egingo dira. Gure kasuan atal bakoitza bukatzerakoan, adierazle horiek betetzen direla zirtatuko gara, eta ez badira betetzen beharrezko aldaketak egingo ditugu horiek betetzen saiatzeko.

2.6.4.3 Kalitate kontrola

Proiektua arrakasta izan dadin kalitate minimo bat izan behar du. Proiektu hau minimo horretara iristeko egiaztapen bakar bat bete behar du. Egiaztapen hau da, burututako analisia baliagarria izatea azken produktua sortzeko. Hau da, datu-biltegia inplementatzen hasten garen garaian, analisi honetan oinarritzea. Hau lortzen bada, kalitatezko proiektu bat lortu dugula esan genezake.

3

Oinarri tekniko-teknologikoak

Proiektuan zehar ikusitako oinarrizko termino teknologiko eta teknikoen azalpena

3.1 Business Intelligence

Negozio Adimena (BI) termino generiko bat da, aplikazioak, azpiegiturak eta erremintak, eta informazioaren sarbidea eta analisia egiteko aukera ematen duten jardunbide egokiak biltzen dituen, erabakiak eta errendimendua hobetzeko eta optimizatzeko.

Negozio Adimenaren lehenengo urteetan, iturri desberdinetako informazioaren bilketan bideratzen zen, ondoren leku bakar batean batzeko, eta horrela, enpresei Data Warehouse baten sarbidez hornitu.

Hala ere, datuen bilketa eta biltegiatzea Negozio Adimenaren zati bat da. Horregatik Negozio Adimenaren inplementazio asko ez dute arrakasta izaten, ez dutelako epe luzerako etorkizuna kontuan hartzen, datuen koherentziaren eta zehaztasunaren aldetik.

Negozio Adimenaren sistema bat inplementatzeko pausu hauek jarraitu behar dira:

1. Enpresa baten datu baseetatik datuak ateratzen ditugu.
2. Datu horiek garbitzen eta eraldatzen dira. Ondoren datu horiek datu biltegian gordetzen dira.
3. Azkenik, BI sistema erabiliz, erabiltzaileak kontsultak egin dezake, edo txostenak eskatu edo edozein analisi egin.

3.1.1 Abantailak

Negozio Adimenaren sistema bat edukitzeak hainbat abantaila izan dezake enpresa edo erakunde batentzat. Alde batetik lortzen ditugun analisiak kalitate hobea dutenez, produktibitatea handitzen da. Erabaki hobea hartzeko ahalmena izango dugu.

Baita ere, datu guztiak leku bakar batean izango ditugu, eta ez dugu iturri askotara joan behar analisi bat egiteko. Datuak toki bakar batean edukitzeak beste abantaila batzuk dakar, batetik, analisisien denbora txikitzen du, datu-biltegian datuak era optimoan antolatuta daudelako, eta bestetik, analisisien kalitatea handitzen du, datu biltegi batean datuak garbiak daudelako.

3.1.2 Desabantailak

Baina, Negozio Adimenak, bere desabantailak ditu. Hasteko, oso konplexua da. Negozio Adimenaren sistema bat inplementatzea ez da gauza erraza. Datu guztiak nola hartzen eta eraldatzen diren diseinatzea lan handia dauka, kalitate maila on bat eduki nahi badugu. Baita ere, inplementazioak denbora handia dakar. Datuak hartzeko prozesua eta datu horiek garbitzeko eta eraldatzeko prozesu horiek inplementatzea denbora handia behar du.

Azkenik, aurreko arrazoiengatik beste desabantaila bat sortzen da, kostua. Denbora handia behar izatea eta oso konplexua izatea kostu handiak dakar.

3.2 Data Warehouse

Datu biltegi bat datu base bat da, non enpresa baten sistema desberdinen informazioa biltegitzen du. Hau da, biltegi bateratu bat da, non iturri desberdinetako informazioa integratu eta maneiatu egiten da, ondoren prozesatu eta aztertu egin ahal izateko, ikuspegi desberdinetatik.

Datu biltegia DBKS (Datu Baseak Kudeatzeko Sistema) baten ganean inplementatzen da, eta bertan datuak txertatzeko karga periodikoak egiten dira, datu-iturri desberdinetatik, ETL erremintak erabiliz. Erremintak hauek ahalbidetzen dute datu biltegia beste datu base batzuekin lotzea, eraldaketak definitzea, datuak ondo integratuak egoteko, eta karga periodikoak automatizatzea.

Datu biltegi bat egiterakoan, hiru arkitektura desberdin kontuan hartu dezakegu:

1. Maila bakarreko arkitektura:

Maila bakar bat edukitzearen helburua biltegitutako datuen kopurua gutxiagotzea da. Helburu hau datuen erredundantzia ezabatzea da. Arkitektura hau ez da gehiegi erabiltzen errealitatean.

2. Bi mailako arkitektura:

Datu biltegia eta fisikoki erabilgarri dauden iturriak banatzen ditu. Arkitektura hau ez da hedagarria eta ezta ere ez ditu erabiltzaile kantitate handirik onartzen. Baita ere, konektibitate arazoak ditu, sarearen mugak direla eta.

3. Hiru mailako arkitektura:

Arkitektura erabiliena da. Goi-, erdiko- eta behe-mailaz osatuta dago:

1. **Behe-maila:** datu biltegiaren zerbitzarian dagoen datu basea da beheko maila. Datu base erlazional sistema bat izaten da. Datuak garbitu, eraldatu eta kargatu egiten dira maila honetan, backend erremintak erabiliz.
2. **Erdiko-maila:** erdiko-maila datu biltegi batean OLAP zerbitzari bat da, ROLAP edo MOLAP modeloak erabiliz inplementatzen dena. Erabiltzailearentzat, datu basearen ikuspegi abstraktu bat aurkezten dio. Maila hau, baita ere, bitartekari moduan jotzen du, bukaerako erabiltzailea eta datu basearen artean.
3. **Goi-maila:** erabiltzaile maila da. Datu biltegiara konektatu eta datuak ateratzen duten APIak eta erremintak dira goi-maila. Kontsulta erremintak, txosten erremintak, analisi erremintak edo datu-meatzaritzako erremintak izan daitezke.

Datu biltegi batek honako ezaugarriak ditu:

- **Gai bati bideratuta:** gai baten informazioa ematen digu eta ez enpresa edo erakunde batek momentu horretan dituen eragiketak. Datu base batek aldiuneko informazioa gordetzen du, aldaketaren bat badago datu basea eguneratzen da, eta aurreko informazioa galdu egiten da. Aldiz, datu biltegia informazio guztia gordetzen du, analisi sakonak egin ahal izateko.
- **Integratua:** neurri unitate berdina erabiltzen da. Hau da, datu base askotatik informazioa hartzen denez, informazio hori eraldatzen da, datu guztiak koherentzia izateko. Adibidez,

distantzia bat adierazi nahi badugu gure datu biltegian, gerta daiteke distantzia horiek datu base desberdinetan egotea eta bakoitzean unitate desberdinetan egotea, beraz, datu biltegiara eramaten ditugunean eraldatu egin behar dira, guztiak neurri berdina izateko.

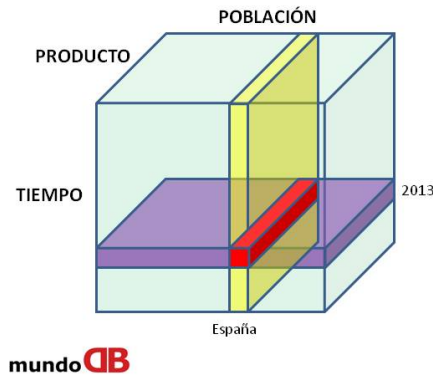
- **Denboraren aldaera adierazi:** datuen historikoa gordetzen da, azterketa sakonak egin ahal izateko. Gehienetan denbora elementu bat izan ohi dute datu biltegiako elementuek.
- **Ez-hegakorra:** datuak ez dira ezabatzen datu-biltegi batean. Datuak bakarrik txertatzen dira. Datu base batean datu bat eguneratzen bada, ondoren ez da datu biltegian eguneratzen, beste berri bat sartzen da, denbora marka batekin. Horrela, oraingo egoera, egoera zaharra eta noiz gertatu den aldaketa daukagu gordeta. Honekin lortu egin duguna da, analisi sakonagoak egiteko aukera.

3.2.1 OLAP

Datuen kudeaketa sistema OLAP motakoa da. Bi motatako sistemak daude OLAP eta OLTP. Bakoitzak bere abantaila eta desabantailak dauzka. Hemen oso laburki azalduta:

- **OLAP** (online prozesamendu analitikoa): sistema bat edo gehiagoren datu bilduak aztertzen ditu. Datu ilara askoren irakurketa konplexu handietarako optimizatua dago. Datu baseko datuak normalean ez daude normalizatuak. Transakzio askorik ez direnez egiten, OLAP ez da kezkatzen datu integritateaz. Datu biltegiak erabiltzen dute.
- **OLTP** (online transakzioen prozesamendua): irakurketa eta idazketa txikientzat optimizatua dago, errendimendua obetzeko. Datu baseko taulak normalizatuak egon behar dute. Datu base operazioetan erabiltzen da gehien bat. OLTP sistema bat, OLAP sistema baten iturri izaten da. Datuen integritatea zaintzen saiatu behar du, transakzio asko egongo direlako.

Funtzionatzeko, OLAP aplikazioek datu base bat erabiltzen dute, multidimentsionala dena, baita ere OLAP kuboak deitzen dena. Hainbat dimentsio dituen datu base bat da, horregatik bere izena, multidimentsionala modeloa. Datu base erlazional bat bi dimentsiotakoa bezala hartu dezakegu, eta datu base multidimentsional bat atributu bakoitza dimentsio bat bezala har dezakegu. Hona hemen hiru dimentsiotako kubo baten adibidea:



3.1 Irudia: OLAP kubo

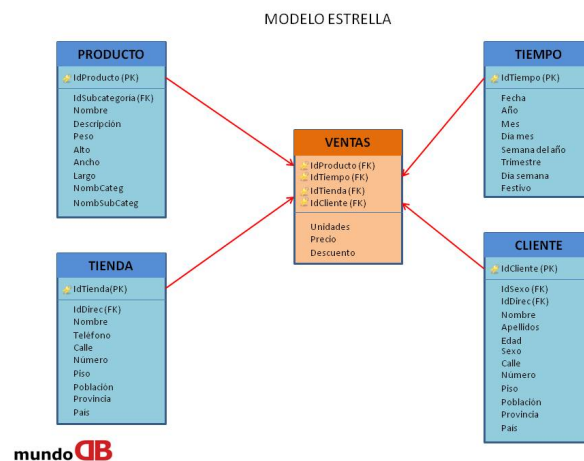
Kubo honetan ikusi daiteke zer produktu saldu egin diren herrialde eta denbora baten arabera, era optimo batean. Kubo hauek datuak era konkretu batean antolatzea eskatzen du, modelo dimentsional batean.

3.2.2 Modelo dimentsionala

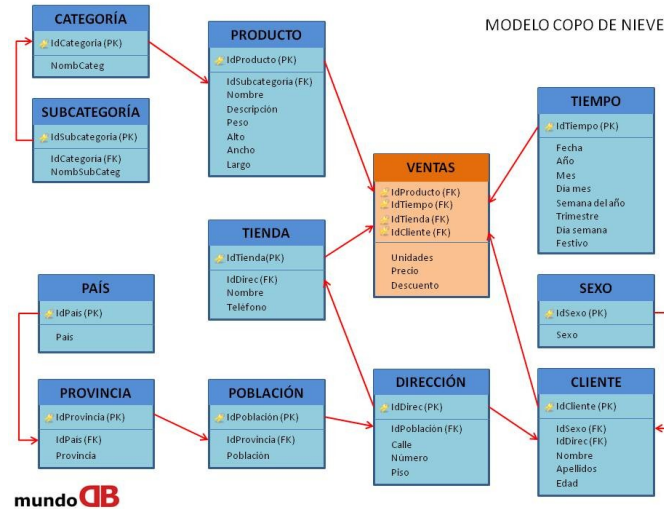
Mota honetako datu-modeloa elementu hauek dauzka:

- **Gertaeren taula:** etengabe erregistratu behar diren enpresa baten gertaerei buruzko indizeak eta neurriak ditu (salmenten bolumena, adibidez).
- **Dimentsioen taulak:** gertaera taulan dauden datuak deskribatzen duen atributuak dauzka. Erreferentzia datu batzuk dira, gertaera taulan dauden gertaerentzat.

Hierarkia gabeko modelo multidimentsional bat **izar modeloa** deitzen zaio. Eta hierarkiak baditu, **elur-maluta modeloa**. Hona hemen modelo bakoitzaren adibide bat:



3.2 Irudia: Izar modeloa

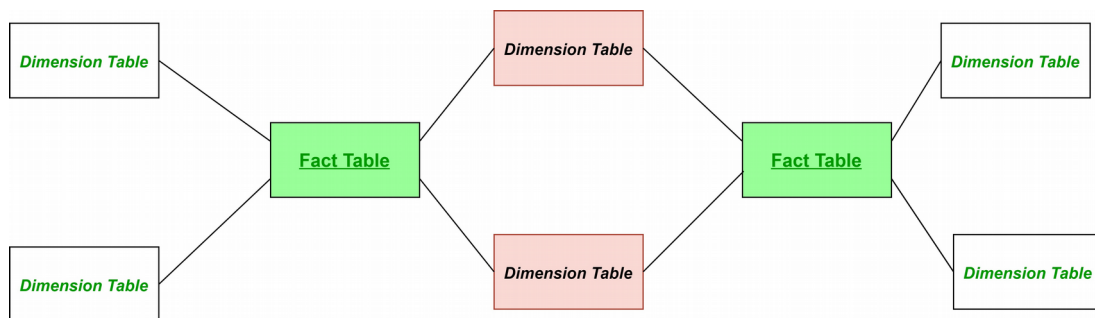


3.3 Irudia: Elur-maluta modeloa

Bi irudietan kolore laranja duen taula gertaeren taula da, eta beste guztiak, urdin kolorekoak, dimentsioen taulak.

Izar modeloa, datu basea guztiz desnormalizatua dago, horregatik egiten diren kontsultak azkarragoak dira. Ikusten denez diseinua oso sinplea da eta kuboen prozesamendua azkarragoa da. Aldiz, elur-maluta modeloa, datu basea konplexuagoa da eta normalizatuago dago. Kuboaren prozesamendua motelago izan daiteke.

Eskema mota gehiago daude, baina bi hauek dira erabilienak. Beste bat aipatzeko, baita ere nahiko erabilia dena, **galaxia modeloa** da. Hau bi izar modeloen konbinazioa da. Bi gertaera taula daude eta dimentsio taulak komunean dituzte, irudi honetan agertzen den moduan:



3.4 Irudia: Galaxia modeloa

Modelo dimentsional bat gauzatzeko pausu hauek jarraitu behar dira:

1. **Negoioaren prozesuak identifikatu:** identifikatu datu biltegi batek eduki beharko lukeen negozio prozesua. Marketing, Salmentak, Giza Baliabideak, etab. Izan daitezke, behar dugun analisiaren arabera.
2. **Zehaztasun maila identifikatu:** pausu honetan galdera hauek erantzun behar ditugu:

1. Eskuragarri dauden produktu guztiak gorde behar ditugu edo batzuk bakarrik?
2. Produktuaren informazioa hiler, astero, egunero edo orduero biltegiatu behar da?
3. Aurreko bi erantzunak nola eragiten du datu biltegiaren tamainari?
3. **Dimentsioak identifikatu:** dimentsio hauetan gorde behar dira nahi ditugun datu guztiak.
4. **Gertaerak identifikatu:** aztertu nahi den datua da. Datu horren informazio guztia dimentsio tabletan dago, gertaera taulan bakarrik taula horiekin erlazioa dago. Baita ere informazio txiki bat egon daiteke, zenbakizkoak normalean, kostua adibidez.
5. **Eskema egin:** modeloa implementatuko litzateke pausu honetan. Eskema bat datu base baten taulen antolakuntza da.

3.2.3 Pixkanaka aldatzen diren dimentsioak

Sistema transakzional batean informazioa eguneratu daiteke, baina bakarrik azken bertsioa gordetzen dute. Baina datu biltegi batean, aldaketen historial guztia gorde behar dugu, momentu bakoitzean gertatu dena jakiteko. Beraz, datu-iturri batean aldaketa bat badago, nola gauzatuko da aldaketa hori datu biltegiari? Pixkanaka aldatzen diren dimentsioak (Slowly Changing Dimensions) erabiliko dugu.

Bi mota daude, orain azalduko direnak:

1. **1 motakoak: gainean idatzi:** informazio berria, informazio zaharraren gainean idazten da, ez dira historikoak gordetzen eta oraingo bertsioa bakarrik egongo da. Mota hau erabiltzen da datu batean erroreren bat badago edo ez zaigunean interesatzen historikoa gordetzea.
2. **2 motakoak: aldaketen historiala:** denboran zehar egondako informazio guztia gordetzen du. Datu-iturri batean aldaketa bat egon bada, datu biltegiari ez da informazio hori eguneratzen, baizik eta beste ilara berri bat sartzen da, hasiera data batekin. Aurreko bertsioari amaiera-data bat txertatzen zaio.

ETL prozesua izan behar du aldaketa hauen arduraduna. Jakin behar du zein erabili kasu bakoitzean.

3.3 ETL prozesua

ETL (Extract, Transform and Load) datu-iturrietatik datuak ateratzearen, eraldatzearen eta azkenik datu biltegiari sartzearen arduraduna den prozesu bat da.

Betidanik, prozesu hau kode pertsonalizatu baten bidez gauzatu da. Honek arazo handiak ekarri ditu, informazioa geroz eta handiagoa eta konplexuagoa bilakatzen joan den heinean. Horregatik, gaur egun, ohikoagoa da ETL erremintak erabiltzea prozesu hau gauzatzeko.

Orain definitu egingo dugu zehazki zer den atal bakoitza, hau da, ateratzea, eraldatzea eta kargatzea, egin behar duena:

3.3.1 Ateratzea

Izenak adierazten duen moduan, pausu hau datu-iturrietatik datuak ateratzean datza. Gauza asko hartu behar dira kontuan pausu hau behar den bezala aurrera eramateko, orain azalduko direnak.

Hasteko, etapa honen planifikazioak datu-iturri bakoitzaren datuen tamainen estimazioak barne izan behar du. Baita ere, datuak atera egin behar dira datu-iturrietan eragin negatiborik izan gabe. Hau da, datu-iturriek berdin-berdin funtzionatzen jarraitu behar dute datuek ateratzen diren garaian. Datu-iturria ezin da aldatu, ezta datu-biltegiaren beharretara moldatu.

Prozesu hau diseinatzea esan nahi du, erabakiak hartu behar direla bi alderdi nagusi hauetaz:

- Zer ateratze metodo aukeratu dugu? Honek datu-iturrian, garraio prozesuan eta biltegia eguneratzeko denboran eragiten du.
- Nola emango ditugu datuak ondorengo prozesamendua egiteko? Honek garraio metodoan eta datuak garbitzearen eta eraldatzearen beharretan eragiten du.

Aterako diren datuen kantitatearen estimazioa eta ETL prozesuaren etapa (hasierako karga edo datuen mantentzea) baita ere eragin dezake nola ateratzen diren datuak erabakian, ikuspegi logiko eta fisiko batetik. Funtsean, erabaki behar dugu nola atera datuak ikuspegi logikotik eta fisikotik:

- **Ateratze logikoko metodoak:** datuen ateratzea hiru modutan izan daiteke:
 - **Eguneraketa jakinarazpena:** datuak ateratzen dira datu-iturriak datuek eguneratu diren jakinarazpen bat ematen duenean.
 - **Ateratze inkrementala:** sistema batzuek ezin dute jakinarazpenik eman, bana bai identifikatu dezakete eguneratu diren datuak eta datu horien laburpen bat eman. Mota honen desabantaila bat izan daiteke sistemak datu ezabatuak ez ezin identifikatu ahal izana.
 - **Ateratze osoa:** beste sistema batzuk ezin dute identifikatu eguneratutako datuak, beraz datu guztiak birkargatzea aukera bakarra da. Metodo honek aurreko ateratzearen kopia bat gordetzea eskatzen du, horrela jakin dezakegu eguneratu diren datuak. Azken baliabide bezala erabiltzea, eta taula txikiekin erabiltzea gomendatzen da.
- **Ateratze fisikoko metodoak:**
 - **Online ateratzea:** ateratze prozesua zuzenean iturrira konektatzen da, edo bitarteko sistema batera konektatzen da, datuak modu aurrekonfiguratu batean gordetzen dituena.
 - **Offline ateratzea:** datuak ez dira iturritik zuzenean ateratzen, baizik eta iturritik kanpo antolatzen dira. Datuak egitura bat daude iadanik edo beste ateratze prozesu batek sortuak dira. Ondorengo egiturak kontuan hartu behar dira:

- Formatu generiko bat duten fitxategi lauak. Objektuaren informazio gehigarria behar da, ondorengo prozesamendurako.
- Oracle-ren formatu espezifiko bateko fitxategiak irauli. Barruko objektuen informazioa barne sartzen da.
- Erregistroak berregin eta artxibatu. Informazioa beste fitxategi batean dago.
- Garraiatzeko taula-espazioak.

Ez dago ateratze metodo ideal bakar bat, hau datu-iturriaren eta negozio beharren araberakoa izango da.

3.3.2 Eraldatzea

Eraldatze prozesuan datuak garbitu eta beharrezko elkarketa egiten da, analisirako prestatuz. Arkitekturaren aldetik, bi modu daude eraldaketa egiteko:

- **Zenbait pausutako eraldaketa:** ETL prozesu klasikoa da. Ateratako datuak prestaketa gune batera mugitzen dira, eta hor bertan eraldaketa gauzatzen, datu-biltegian kargatu baino lehen.
- **Biltegian egiten den eraldaketa:** prozesuaren ordena zertxobait aldatuko zen, orain ez zen ETL prozesu bat izango, baizik eta ELT prozesu bat. Datuak ateratzen dira eta zuzenean datu biltegian kargatzen dira. Eraldaketa datu biltegian gauzatzen da.

Azken urteetan aldaketa bat egon da, eraldaketa biltegian bertan egiteko joerarekin, faktore hauengatik:

1. Datu base modernoek errendimendu eta eskalagarritasun hobetoengatik.
2. Datu baseen kapazitatea eraldaketak SQL lengoia egiteko.

Datuak eraldatzeko modu desberdinak daude:

- **Scripting-a:** SQL-en edo Python-en idatzitako script-en bidez gauzatutako eraldaketa.
- **ETL erremintak instalazioetan:** ETL erremintek prozesua automatizatzen dute. Normalean, enpresako leku batean kokatzen dira, eta esperientzia eta kostu handiak behar izaten dute.
- **ETL erremintak hodeian:** hodeian egoten dira, eta hornitzailearen azpiegitura aprobetxatu dezakegu.

Eraldaketa, pasu batzuk jarraitzen dira datuak nahi dugun formatua lortzeko. Kasu batzuetan, datuak garbitu egin behar dira, ondoren eraldatu ahal izateko. Behin datuak garbitu direnean, ondorengo pausuak jarraitu behar dira:

1. **Datuen aurkikuntza:** datuak iturriko formatuan identifikatu eta ulertu. Honek laguntzen du erabakitzen zer gertatu behar den datuekin, nahi dugun formatua lortzeko.
2. **Datuen mapaketa:** eraldaketa planifikatzen da.
 1. Zehaztu: zer datu mugituko diren zehaztu. Baita ere maiztasuna zehaztuko da.
 2. Datuak esleitu: iturburu-eremuak helmuga eremuekin bat etorri behar du.
 3. Transformazioa: zer datu behar duten eraldaketa zehaztu.
 4. Froga: frogapen sistema batekin eraldaketa horien frogak egin.
3. **Kodea sortu:** eraldaketa gauzatzeko kodea sortzen da.
4. **Kodea exekutatu:** eraldaketa prozesua martxan jartzen da eta datuak formatu egokia lortzen dute.
5. **Egiaztapena:** eraldatutako datuak egiaztatzen dira, lortu nahi zen formatu dutela konprobatuz.

3.3.3 Kargatzea

Behin datuak eraldatu direnean, datu horiek datu biltegian kargatzeko prozesua da. Pausu honetan, datu biltegiak behar duen lana kontuan hartu behar da. Datu kopuruaren, egituraren, helmugaren eta karga motaren arabera, gure helmuga-sistemari eragin egingo dio, positiboki edo negatiboki.

Datu biltegi batean datuak kargatzeko bi metodo daude:

- **Karga osoa:** datuen irauli osoa, datu biltegia lehen aldiz kargatzen denean ematen da metodo hau.
- **Karga gehikorra:** datuen karga tarteka egiten da. Azken ateratzearen data gordetzen da, horrela bakarrik data horretatik aurrera egindako eguneraketak txertatuko dira. Bi mota bereizten ditu, kargatu beharreko datu kantitatearen arabera:
 - Transmisioko karga gehikorra: datu kantitate txikiak kargatzeko hobeagoa.
 - Sorta bidezko karga gehikorra: datu kantitate handiak kargatzeko hobeagoa.

	Karga Osoa	Karga gehikorra
Ilaren sinkronizazioa	Datu-iturrietako ilara guztiak	Bakarrik erregistro berriak eta eguneratuak izan direnak
Denbora	Denbora gehiago	Denbora gutxiago
Zailtasuna	Baxua	Altua

3.1 Taula: Kargatzeko metodoen ezaugarriak

Karga prozesua exekutatzen ari den momentuan, ez dugu utziko erabiltzaileei datu biltegiara sartzen. Hau egitearen arrazoia honako hau saihestea da:

- **Datuen analisi ez-osoia:** karga egiten ari den momentuan, gertaeren taulak eta dimentsioen taulak ez daude guztiz beteta. Erabiltzaileek analisi bat egiten badute, datuak guztiz sartu gabe oraindik, emaitzak okerrak izango liratekeen.
- **Kontsultaren errendimendu arazoa:** erabiltzaile batek kontsulta bat martxan jartzen badu, gertaeren eta dimentsioen taula behar dituena, momentu horretan beste operazio bat egiten ari da. Beraz kontsultaren errendimendua kaskarra izango litzateke, eta baita ere karga prozesuaren errendimendua.

Beste alde batetik, datu biltegi baten karga prozesuan operazio hauek parte hartzen dute:

- **Txertatze operazioa:** datu biltegian, datu berriak txertatu dezakegu, edo dimentsio datuen bertsio berriak txertatzea (pixkanaka aldatzen diren dimentsioak).
- **Eguneratze operazioa:** operazio hau datu biltegian dauden datuak eguneratzeko balio du. Datu-iturrietan eguneratu diren datuak, datu biltegian eguneratzen ditugu (pixkanaka aldatzen diren dimentsioak).
- **Upsert operazioa:** operazio berezi bat da, ilara bat txertatzerako garaian ilara hori datu biltegian badago eguneratu egiten dugu, eta ez badago datu-biltegian txertatu egingo dugu.
- **Karga masiboko utilitatea:** aurreko hiru operazioak errenkadaka egiten da. Datu askorekin tratatu egin behar badugu, errenkadako prozesamendua oso motela izango litzateke. Beraz, karga masiboko utilitatea erabiliko dugu. Honek, sarrerako multzoa hartzen du (fitxategi lau bat) eta edukia taulan txertatzen du.

4

Informazio-bilketa

Kapitulu honetan, egin behar dugun analisia egiteko behar dugun informazio osoa azalduko da. Hau da, bildu egin den informazio osoa azaldu egingo da.

4.1 ETL prozesua

Aurretik azaldutako ETL prozesutik, bi galdera sortu dira honen inguruan. Alde batetik, ETL erreminta bat erabiliko dugu edo guk sortutako kode bat erabiliko dugu prozesua gauzatzeko? Beste alde batetik, ETL erreminta bat erabiltzen badugu, non exekutatu dugu erreminta, gure instalazioetan edo hodeian?

Aurreko galderari erantzuteko ETL erreminta batzuk aztertu ditugu, eta baita ere kode bat sortzeko teknologia aztertu ditugu.

4.1.1 ETL erremintak

Ondoren aztertutako ETL erremintak azalduko dira. Baina lehen, azaldu egingo dugu zer informazio bilatu egin den erreminta bakoitzean. Izan dira kasu batzuk, hornitzaileak ez duela informazio guztia eman. Hona hemen bilatu egin diren ezaugarriak erreminta bakoitzeko:

- Kostua
- Erabiltzeko erraztasuna
- Euskarria
- Erremintaren eskakizunak
- Abiadura
- Datuen kalitatea
- Kontrol erremintak
- Konektibitatea

4.1.1.1 Oracle Warehouse Builder

Datu integrazioaren alderdi guztietarako erreminta bakarra da Oracle Warehouse Builder. Erreminta honek Oracle-ren datu baseaz baliatzen da datuak eraldatzeko kalitate handiko informazioan. Erreminta honek hiru ezaugarri talde dauzka:

- **Basic ETL:** ETL gaitasun oinarrizkoen multzoa da eta datu biltegi sinpleentzat egokia da.
- **Enterprise ETL:** negozio datuen bilketarako funtzionalitate aurreratuak eta datuen integrazio proiektuak.
- **Application Adapters for OWB:** SAP eta Oracle ERP aplikazioentzako konektibitatea.

Ezaugarri talde bakoitzak lizentzia bat dauka, baina hornitzaileak ez digu haien kostua zehazten. Baina bai jakina da, Oracle-ren produktuak ez direla batere merkeak.

Ezaugarri taldea gure beharren arabera aukeratu beharko genuke, gure kasuan *Enterprise ETL* aukeratuko genuke, datu biltegi handi bat aurreikusten dugulako. Erreminta azkarra da eta datuen kalitatea bermatzen digu.

Beste alderdi garrantzitsu bat kontuan hartzeko konektibitatea da. Gure datu-iturriak ezin dira edozein izan. Oracle Warehouse Builder bi modu dauzka iturrietara konektatzeko. Lehenengoa

Oracle Heterogeneous Services” da. Honekin Oracle datu baseetara konektatu gaitzke, eta beste batzuetara ere, DB2, SQL Server, Teradata datu baseetara adibidez. Baina baita ere ODBC estandarraren bidez konektatu gaitzke.

4.1.1.2 Informatica PowerCenter

Aditu askoren ustetan, Informatica ETL aukera hobereena da, bere ezaugarriengatik, ahalmenagatik, lan aukerengatik, eta etorkizuneko eta profesional hazkundeagatik. Erreminta honek, aukera ematen digu edozein lukutatik datuak ateratzeko, edozein motatako datuak eraldatzea, edozein modutara, eta azkenik edozein lekutan kargatzea.

Lehen esan bezala, Informatica PowerCenter aditu askorentzat ETL erreminta hobereena da, arrazoi hauengatik:

- %100eko arrakasta-tasa.
- Konektibitate handiena eskaintzen digun erreminta da.
- Garapen funtzioak eskaintzen dituen bakarra da. Konfigurazioaren kudeaketa eta bertsioren kontrola eskaintzen du.
- Eguneratutako datuak identifikatzen ditu.
- Informatica da ezaugarri hauek eskaintzen dituen bakarra.
- Plataforma bakar batek ETL ziklo osoa automatizatzen du.
- Formazio erreza dauka.

Baita ere, azken urteetan Informatica PowerCenter hazten ari da. Hori aurreko arrazoiengatik da, baina ez hori bakarrik. Baita ere, aukeratu dezakegu non egongo den zerbitzaria, gure instalazioetan edo hodeian. Baita ere interfaze grafiko bat du, datuak monitorizatzeko, datuak mantentzeko eta datuekin sortutako gatazkak konpontzeko. Datuak beti seguruak mantentzen dira.

Baina ez Informatica PowerCenter ez dira soilik abantailak, baita ere desabantaila batzuk dauka. Alde batetik, oso garestia da, 2400€-tik aurrera hileroko eskuragarri dago. Beste alde batetik, Informatica soilik ETL prozesuan zentratzen da. ETLren liderra da merkatuan. Baina ez du datu kalitatearen ezaugarriak eskaintzen.

4.1.1.3 Talend Open Source Integration Software

Kode irekiko erreminta bat da, eta beraz, guztiz doakoa da. Erreminta honen ezaugarri garrantzitsu bat da ETL eta ELT egiteko aukera ematen digula.

Konektibitatearen aldetik, Talend aukera handiak ematen digu, datu base askotara konektatu gaitzkeelako, besteak beste, DB2, Amazon Redshift, JavaDB, PostgreSQL, Oracle, SQLite, ... Baita ere, JDBC estandarraz baliatu gaitzke. Baina ez bada nahikoa, Talend ordainpeko bertsiorekin bat dauka, konektore gehiagorekin. Edo beste aukera bat da, gure konektorea sortzea. Kode irekiko denez, gure konektoreak sortu ditzakegu.

Kode irekiko datu integrazioaren liderra da Talend, eta hori ezaugarri hauengatik da:

- Paketatutako aplikazioak (ERP, CRM, etab.), datu baseak, konputagailu zentralak, web-zerbitzu fitxategiak, etab. konektatzeko ahalmena.
- Datu biltegietan, data marts-etan, OLAP aplikazioetan idazketa eskaintzen du, analisirako edo txostenen sorkuntzarako esate baterako.
- Osagai integratu aurreratuak kontsulten tratamendu automatikoa egiteko.

Aipatu bezala, ordainpeko bertsio bat eskaintzen digute. Bertsio honek konektore gehiago, lankidetzak gaitasunak, programazioa, administrazioa eta jarraipena, datuen kalitatea eta aditu baten laguntza eskaintzen digu.

4.1.1.4 Pentaho Data Integration – Kettle

Erreminta hau baita ere kode irekikoa da, eta Apache 2.0 lizentziapean dago. Horregatik, baita ere, guztiz doakoa da.

Java aplikazio eta liburutegi bat da. Talend-en ezaugarriak eta osagaiak Pentaho-renak baino ulergarriagoak dira, baina honek ez du esan nahi ezin dituela ETL prozesu konplexuak inplementatu. Pentaho honako abantailak eskaintzen digu:

- Doakoa: bere erabilerak ez du gasturik suposatzen lizentzietan.
- Ordainpeko edizioa: edizio honek zerbitzu-teknikoa eskaintzen du eta soluzio konplexuagoak garatzeko ahalmena.
- Komunitate handia: aktibitate handia foroetan, baita beta-frogak garatzen.
- Teknologia aurreratuak eta erabilera erraza: doako eta ordainpeko bertsioetan azken teknologekin lan egiten da, beti instalazio eta konfigurazio errazak mantenduz.
- Programa asko: modularra izatea eta funtzio desberdinak burutzeko programak izatea, haien artean datuen integrazioa, Pentaho-ren abantaila handi bat da.

Konektibitatearen aldetik, baita ere abaniko handi bat eskaintzen du. Datu base erlazioaletara (Oracle, DB2, MySQL, ...), NoSQL datu basetara (MongoDB, Cassandra, ...), datu base analitikoetara (Snowflake, Teradata, SAP HANA, ...), business applications-etara (SAP, Google Analytics, ...), eta fitxategietara (XML, JSON, Excel, ...) konektatu daiteke.

Aukera interesgarri bat ezan daiteke datuen integrazioarako software bat aukeratzeko garaian. Datu integrazio adimentsu bat egin dezakegu, kalitate hobereana lortzeko asmoz.

4.1.2 Kodea sortzeko teknologiak

Programazio lengoia batzuk aztertu egin ditugu, eta haien barruan, lagungarriak izan daitezkeen liburutegiak edo erremintak.

4.1.2.1 JavaScript (Node.js)

ETL aplikazio askotan Node.js erabiltzen da. Arrazoi nagusia da honen asinkrono izaera. Ehunka ilara dituzten datu baseak oso azkar prozesatu daitezke Node.js-ri esker. Itxaron denbora murrizten du eta prozesuaren eraginkortasuna handitzen laguntzen du.

Beste arrazoi bat Node.js erabiltzeko da, JSON-ekin era natiboan funtzionatzen duela. Baita ere Node.js liburutegi asko izatea oso baliagarri gertatzen da. Gainera, JavaScript datuen ikuskapena errazten du.

4 ETL erreminta aurkitu ditugu Node.js-arentzat:

1. **Empujar:** datuak bidali eta segurtasun kopiak egiten ditu, beste datu-operazioen artean. Datuen operazioak seriean edo paraleloan egiteko gaitasuna dauka. Kode irekiko erreminta da eta hainbat datu baseetara konekta daiteke, besteen artean, MySQL, Amazon Redshift, Elasticsearch, FTP eta S3. Bere dokumentazioan azaltzen da nola inplementatu dezakegun gure konektoreak.
2. **Nextract:** kode irekiko ETL erreminta da, eta oso aukera ona programatzaile hasiberrientzat eta erdi-mailakoentzat. Garatzaileen lana errazteko egin zen, Java-n egindako ETL erreminta batzuen aurrean. Npm moduluak erabil dezake. Datu base batzuekin bateragarria da, besteak beste, PostgreSQL, MSSQL, MySQL, MariaDB, SQLite3 eta Oracle. CSV eta JSON fitxategiak hartu eta sortu dezake. Makina bakar batean exekutatu dezake, beraz ez da gomendatzen Big Datan erabiltzea.
3. **Extraload:** fitxategietatik datu baseetara eta datu baseen artean datuak mugitzen dituen ETL erreminta da. ETL modu azkar eta erraz batean funtzionatzea egiten du. Dokumentazioan aurkitu dezakegu konektore gehiago nola instalatu.
4. **Datapumps:** oinarrizko ETL erreminta bat da eta "bonbak" erabiltzen ditu sarrera irakurtzeko eta irteeran idazteko. ETL lan konplexu bat sortzeko bomba talde bat sor dezakegu. Konfigurazio gehigarria behar da, ez duelako ETL lan guztia egiten bere kabuz.

4.1.2.2 Python

Python garatzaileen komunitateak kode irekiko erreminta asko sortu dituzte, ETL errazago egiten dutenak. Ezin izan dugu erreminta guztiak aztertu, baina aipatuena orain azalduko dira:

1. **Airflow:** erreminta honen teknologia nagusia Zuzendutako Grafiko Aziklikoen eraikuntzan datza. Honek, programatzaileari ahalbidetzen dio zereginak beste langileei banatzea oso modu errazean. Erabiltzaile interfaze praktikoa batekin dator. Airflow oso hedagarria eta eskalagarria da.
2. **petl:** Python pakete bat bat, ETL-arentzat. Erabiltzaileari Python taulak sortzea ahalbidetzen dio, datuak hainbat iturri posibleetatik hartuz, CSV, XLS, HTML, TXT, JSON, eta abar modukoak. Ondoren datu basera edo fitxategi batera bidaltzen ditu datuak. ETL lanentzat diseinatu egin da.
3. **Panoply:** datu biltegitratze soluzio moduan diseinatu da, funtzionalitate guztiekin. Panoply prozesuaren pauso guztiak maneiatzen ditu, datuen ateratzea sinplifikatuz edozein datu-iturritik.

4. **pandas**: datuen azterketa eta manipulazio erreminta kit bat da. Datu markoak gehitzen ditu, R estilotik Python-era. Honek egiten duena da, datuen manipulazioa, garbiketa eta analisia askoz errazagoak izatea. Pandas prozesuaren pausu bakoitza gidatu dezake, biltegitarte formatu gehienetatik datuak ateratzen utziz.
5. **Bonobo**: datu hodiak sortzeko erremintak dauzka eta hauek datu-iturri asko prozesatu dezakete paraleloan. SQLAlchemy hedapen bat dauka eta hodia SQL datu base batera zuzenean konektatu dezake. Erraz funtzionatzen jartzeko diseinatu da. ETL prozesuaren grafikoen irudikatzaile bat dauka ere, prozesuaren jarraipena errazten duena.

4.1.2.3 Java

ETL prozesu konplexuak inplementatu dezakegu, baita ere, Java erabiliz. ETL erreminta batzuk baino are gehiago egin dezakegu Java-rekin. Hona hemen ETL prozesuak inplementatzeko lagungarriak izango diren tresna batzuk:

1. **Apache Spark**: tresna honen hasiera azkarrak erakusten digu nola idatzi Java aplikazio autonomo bat. Gainera funtzionalitate gehiago lortu ahal dugu Spark dituen paketeekin. Erreminta asko ditu datuen prozesamendurako eta eraldaketarako. Kalkuluak paraleloan exekutatzeko diseinatuta dago. Baita ere beste erreminta batzuk ditu, datuen transmisio azkarrerako, ikaskuntza automatikorako eta grafikoen prozesamendurako.
2. **Jaspersoft ETL**: kode irekiko ETL erreminta, txosten dinamikoak txertatu ahal dituen gure Java aplikazioan edo webean. Datuak atera dezake edozein iturritatik eta 10 formatu desberdinetara esportatu. Errez erabiltzeko aukera bat da datuen prozesamendurako.
3. **Scriptella**: kode irekiko ETL erreminta da, Java-n idatzia. Datuen eraldaketa lana sinplifikatzeko sortu zen. Datu baseen arteko ETL script-tak onartzen ditu, eta funtzionatu egiten du datu-iturri askorekin.
4. **Pentaho Kettle**: ETL erreminten atalean aipatu dugu tresna hau. Java-z idatzita dago eta bere metodo propioa erabiltzen du datuen integrazioarako, Pentaho-ren metadatueta oinarrituta. Datuak mugitu eta eraldatu, lanak sortu eta exekutatu, iturri askotatik datuak atera eta gauza gehiago egin dezakegu. Baina ezin dugu eraldaketak sekuentziatu.
5. **Talend Open Source Data Integrator**: erreminta hau baita ere lehen aipatu dugu. ETL erreminta bat da eta Java kodeko liburutegiak izan dezake. Talend uzten digu kodearekin lan egiten, fitxategiak administratzen, eta datu handiak eraldatzen eta integratzen. Garapen grafikoko eta diseinu tresnak ditu, eta ehunkada osagai eta datuen prozesamendurako konektoreak.

4.1.2.4 PHP

PHP-rekin batera ETL erreminta bat behar da. PHP-en ETL liburutegiak egokiak dira ETL komando lerrotik egiten duten programatzaileentzat. Egia esan ez ditugu PHP erreminta askorik aurkitu, hemen aurkitu direnak:

1. **PHP ETL**: ondo dokumentatu dagoen ETL liburutegi bat da, eta ETL gauzatzen du komando lerrotik. Iturrientzat eta datu biltegiarentzat, PHP ETL bateragarria da datu base hauekin: MySQL, PostgreSQL, SQL Server eta SQLite. Eraldaketak JSON kodifikatzera eta dekodifikatzera, zutabeak berrizendatzera eta moztera, errepikatutako ilarak ezabatzea eta maiuskulak eta minuskulak bihurtzera mugatzen da. Lan handia behar du konfiguratzeko eta erabiltzeko.
2. **YaEtl**: grafiko gidatuak erabiltzen ditu, nodo exekutagarriez osatua, eraldatzeko, batzeko eta datu fluxuak adarkatzeko iturri askotatik. Sarrera eta irteera formatu asko onartzen ditu. Nodoaren diseinuak denbora aurrezten laguntzen du, ETL prozesuak kateatzen dituelako eta zeregin konplexuak sortu eta antolatzen ditu.

4.2 Data Warehouse

Aurreko kapituluaz azaldutako moduan datu biltegi bat DBKS sistema baten gainean inplementatuta dago. Puntu honetan DBKS desberdinak aztertu ditugu, eta horiek aztertze irizpide hauek jarraitzen saiatu gara:

1. **Eskalagarritasuna**: aukarik hoberena datu basearen diseinu malgua ematen duena da. Baita ere, negozioren betebeharrei egokitzen zaion sistemaren arkitektura, horrela hardware eta aplikazioen inbertsioaz hobeto baliatzen gara.
2. **Potentzia**: DBKSak soluzio konplexuei euskarria eman behar du.
3. **Erabilgarritasuna**: inaktibitate denbora minimoa izan behar du.
4. **Kudeaketa**: kontrol puntu bakarra izateak sistemaren kudeaketarako aukerarik hoberena da.
5. **Kostua**: jabetza osoaren kostu minimoa izatea gomendagarria da.
6. **Elkarreragingarritasuna**: weberako, barruko saretara eta zerbitzu korporatiboetarako sarbide integrala izatea, ezaugarri batzuk dira kontuan hartu beharrekoak.

Orain aztertutako DBKS azalduko dira:

4.2.1 Teradata

Teradataren EDWak (Enterprise Data Warehouse) enpresei hornitzen die biltegiatze hibridoaren gaitasun eta azterketa sendoekin eta eskalagarriekin. Biltegiatze hau lortzen da egituratuta eta ez egituratuta diren datuen monitoreen bidez. Teradata-k baita ere soluzio bat eskaintzen du hodeian oinarrituta. Erlazional eta zutabe ahalmenak konbinatzen ditu, NoSQL ahalmen

mugatuekin batera. Memorian dagoen datu base baten errendimenduak ez du kostu gehigarriarik, memoria gordetzen dituelako gehien erabili diren datuak. Hauek dira Teradata erabiltzearen abantailak eta desabantailak:

- Abantailak:
 - Oso eskalagarria.
 - Datu kantitate handiak prozesatzen ditu oso erraz.
 - Aurretik eraikitako funtzionalitate asko dauzka.
 - Nabigatzeko erraza.
- Desabantailak:
 - Enpresa txikientzat garestia izan daiteke (hornitzaileak ez digu lizentziaren kostua zehazten).
 - Hodeian izateko aukera beste hornitzaileak baino garestiagoa.
 - Instalazioak denbora asko behar du.

4.2.2 Oracle

Oracle 12c Database datu optimizatuen prozesamenduaren estandarra da, eskalagarria eta errendimendu altukoa. Datu biltegitako plataforma espezializatua Oracle Exadata Machine da. Abangoardiako plataforma honek gaitasun aurreratuak eskaintzen ditu, Flash biltegitratzea S/I gainkargarentzat, adibidez.

Oracle 19c bertsio berria da, eta Oracle 12c eta 18c familiaren kide da. Datu base honetan Autonomous Data Warehouse-rekin aurkitzen gara, Oracle-ren datu base berria dena eta optimizatua eta kudeatuta dago datu biltegitako lan kargetarako, Oracle Database-ren errendimendurekin. Pentsamenduaren abiaduraren prozesamendua eskaintzen du azterketa sofistikatueterako. Datuen zutabeko prozesamendua inplementatzen du baita ere, datu biltegiaren analitika bizkortzeko. Oracle 19c bideragarri dago Linux, Windows, Solaris, HP/UX eta AIX plataformetan, eta Oracle Cloud-en baita ere. Hauek dira Oracle-ren abantailak eta desabantailak:

- Abantailak:
 - Konfigurazio erraztasuna eta potentzia informatikoa.
 - Funtzionalitate asko eta pertsonalizagarriak.
 - Tresna grafiko intuitiboak.
- Desabantailak:
 - Konplexua. Ikaskuntza kurba handia erabiltzaile batzuentzat.
 - Eskarmentu gabeko datu base administratzaile batentzat, konfigurazioak asko zaildu daitezke.
 - Garestia.

4.2.3 Amazon Web Services (AWS)

Amazon ingurune bat eskaintzen du, datuak biltegitzeko erremintaz eta baliabidez beteta, bere hodeiko zerbitzuak osatzen dutenak. Hiru zerbitzu eskaintzen digu datu biltegitzako:

- **Amazon Redshift:** Amazon-en datu biltegi zerbitzua da, hodeian kokatua. Petabyte-ak prozesatzeko gai da oso azkar. Erreminta errentagarria da datu biltegitzarako, edozein datu aztertu ditzakeena SQL estandarra erabiliz. Hona hemen bere abantaila eta desabantailak:
 - Abantailak:
 - Analisirako datu guztiak batera mantentzen ditu.
 - Erabiltzaileak datu multzo handiak biltegitzatu ahal dute sarbide azkarreko datu baseetan.
 - Redshift-en cluster-ak oso azkarrak eta inplementatzeko errazak dira.
 - Desabantailak:
 - Datu analisi batzuk zailak dira, beren tamainaren eta datuen zehaztasunaren arabera.
 - Bere datu base propioa denez bere sintaxiarekin, ezin dira %100 postgres sintaxia edo baliabideak erabili.
 - Mahaigaineko bezeroentzat ez dago aukera askorik.

Bere kostua 0,25€ ordukoa da instantzia batentzat, baina datu kantitate handiak biltegitzatu nahi baditugu, 1000€ terabyte-ekoa da.

- **AWS Data Pipeline:** AWS zerbitzuen artean datuak mugitzeko diseinatuta dagoen web zerbitzu bat da.
- **Elastic MapReduce:** Hadoop¹ soluzio bat eskaintzen du, administratzeko erraza dena, AWS zerbitzuen plataformaren gainean.

4.2.4 MySQL

MySQL datu base transakzional bikaina da, baina lan analitiko bat egiterako garaian ez, batez ere datu-iturri asko eta datu kantitate handiak daudenean. MySQL InnoDB motorrean oinarritzen da, online transakzioen prozesamendurako (OLTP) optimizatua dagoena. Hau esan nahi du, ez dela oso ona kontsulta analitikoak egiteko, konputazionalki intentsuak direnak. Beraz MySQL ez dirudi soluzio hobereana gure datu biltegitzako.

1 Lizentzia librearen bidez aplikazio banatuak onartzen dituen software framework-a da.

4.2.5 PostgreSQL

PostgreSQL kode irekiko datu base aurreratuena da. Oso malgua da, beraz, datu base erlazional sinple baten modura funtziona dezake, eta baita ere, denbora segiden datu base bezala edo kostu txikiko datu biltegi eraginkor bat bezala. Inguru lokal batean erabiltzen baldin bada, produktuaren kostua 0€-koa izango litzateke. PostgreSQL integratu daiteke datu-meataritza, OLAP edo txosten kanpo erremintekin. Aukera on bat izan daiteke erremindu handiko eta kostu txikiko datu biltegi bat nahi badugu. Hona hemen PostgreSQL datu biltegi bezala erabiltzearen abantaila eta desabantailak:

- Abantailak:
 - Ezaugarri askorekin etortzen da, garatzaileei laguntzeko beren aplikazioak sortzen, administratzaileei datu integritatea babesten eta akatsei tolerantzia den inguruneak sortzen laguntzeko, eta datuak administratzen laguntzeko, berdin du datuen tamaina handia edo txikia den.
 - Doakoa eta kode irekikoa izateaz gain, PostgreSQL oso hedagarria da.
- Desabantailak:
 - Komunitatearen laguntza lortzea zerbait zaila izan daiteke.
 - Eskalarekin arazo informatuak daude.
 - Erreplikazioa oraindik ez dago oso ondo inplementatuta.

4.2.6 Panoply

Lehenengo “hodeiko datu biltegi adimentsua da”. Informazioa lortzeko denborarik azkarrena eskaintzen du industriari, garapena eta kodifikazioa ezabatuz, administrazioan, integrazioan eta datu eraldaketan erabiltzen direnak. Panoply hodeian oinarritutako datu biltegi eskalagarria, erabilgarritasun handia eta integratutako kontsulta azkarrak eskaintzen ditu. Puntatik puntara administrazioa eman egiten du, datu prestaketaren lan guztiak automatizatzen. Hona hemen Panoply-k dituen abantailak eta desabantailak:

- Abantailak:
 - Kontsulten optimizazio handia.
 - Materializazio automatikoaren ikuspegiak.
 - Bezero arreta paregabea.
 - Konfiguratzeko eta hasteko erraza.
- Desabantailak:
 - Materializazio prozesuekin noizbehinkako arazoak.
 - Integratutako bistaratze erreminten falta.
 - Erabiltzaile web interfazea motela izan daiteke.

4.2.7 SAS

Datuen erreminta garrantzitsuenetarikoa da, erabiltzaileei iturri desberdinetatik datuak atzitzea eskaintzen duena. Analisi konplexuak egin dezake. Aktibitateak toki zentraletatik kudeatzen da, erabiltzaileei urrunetik erremintara sartzeko aukera ematen duena. Prozesatu gabeko datu fitxategiak kanpoko datu baseetan ikus daitezke, eta datuak datu erreminta desberdinekin administratu daitezke. Hemen SAS erabiltzearen abantailak eta desabantailak:

- Abantailak:
 - Interfaze grafiko ulergarria.
 - Entrenamendua barne dauka.
 - Modelo konplexuak sortzeko erraztasuna.
 - Kalitatearen ezagutzaren oinarria barne.
- Desabantailak:
 - Erroreak ez dira erraz ulertzekoak.
 - Oso garestia izan daiteke konpainia batzuentzat.
 - Ikaskuntza kurba handia eskarmentu gabeko erabiltzaileentzat.

4.3 Datuen sarbiderako APIa

Hasteko API hau datu biltegia martxan dagoenean erabiliko dugu. Horrela datu biltegian dauden datuak atzitu egin ahal izango dugu, eta baita ere hirugarren bati sarbidea eman nahi badiogu ez da zuzenean datu biltegia konektatzen, baizik eta APIra. Horrela segurtasunean eta errendimenduan irabazten dugu.

API bat zehazteko galdera asko erantzun behar ditugu. Alde batetik zer protokolo erabiliko dugu. Beste alde batetik, zer formatutan gauzatuko dugu informazio trukaketa. Baita ere, zer teknologia erabiliko den garapenerako. Eta azkenik, ea merkatuan dagoen API bat erabiltzea komenigarriagoa den.

Beraz aztertutako informazio guztia azalduko dugu:

4.3.1 Protokoloak

APIak API webetara eboluzionatu duten bitartean, esfortzu handiak egin dira beraien diseinu sinplifikatzeko eta inplementazioa errazteko.

4.3.1.1 SOAP

Protokolo estandar bat da, bi objektu desberdin, prozesu desberdinetan daudenak, nola komunika daitezkeen XML bidez definitzen duena. SOAP (Simple Object Access Protocol) zerbitzuak HTTP protokoloaren bidez funtzionatzen dute gehienetan, baina ez dago protokolo horretara lotuta, baizik eta FTP, POP3, TCP, eta abar erabili dezake. Baina HTTP protokolo nagusia da. Baita ere web zerbitzu izenarekin ezagutzen dugu SOAP.

SOAP protokolo hoberena da Server to Server edo Partner to Partner komunikaziorako, protokolo oso sendoa delako, metadatuak erantsi daitezkeelako atributuen bitartez (JSON ezin du egin) eta izen espazioak definitu daitezkeelako. SOAP formatu pisutsu bat da, beren tamaina eta prozesamenduagatik.

Bere abantailen artean, TCP garraio protokoloa erabiltzen duenez, HTTP, SMTP edo JMS protokoloak erabiltzeko aukera dugula. Oso estandarizatua dago SOAP, horregatik, erregela konkretuak daude mezuak bidaltzeko, datuak bidaltzeko formatua beti XML da.

Enpresa ingurunetan oso erabilia da, bere abantailengatik. Gainera segurtasuna haientzat oso garrantzitsua da.

Baina bere desabantailak ditu, haien artean oso estandarizatua egotea (nahiz eta lehen abantaila bezala aipatu). Honengatik ez da batere malgua, eta errore handiak egoten dira garapenean estandarrak ez badira ezagutzen. TCP protokoloa erabiltzen duenez beste protokoloak baino errendimendu gutxiago dauka.

4.3.1.2 REST

REST² teknologia malguago bat da eta datuak HTTP protokoloaren bidez bidaltzen du, baina honek HTTP eskaintzen dituen metodoak erabiltzea uzten du, GET, POST, PUT, DELETE eta PATCH bezalakoak. Baita ere, HTTP erantzun kodeak erabiltzen ditu (404, 200, 404, 409). REST hain malgua da, edozein datu mota bidali dezakegula, datu mota Header Content-Type bidez definitzen delako, eta honek XML, JSON, Bitarrak, eta abar diren fitxategiak bidaltzea uzten digu. Baina formatu erabiliena JSON da, JavaScript-en bidez interpretagarria delako. HTML formularioak REST zerbitzu bat apuntatu dezake arazorik gabe.

Beste abantaila bat JSON erabiltzeko bere errendimendua da. Pisu gutxiago dauka eta horregatik azkarrago prozesatzen da.

REST MVC modeloa jarraitzen duten web aplikazioetan datuak hartzeko erabili beharko zen. Baita ere prozesamendu baliabide kaskarrak ditugunean gomendagarria erabiltzea, banda zabalera txikia dugunean, aplikazio mugikorretan edo robotikan.

Baita ere, REST indar handia hartzen ari da NodeJS-rekin eta NoSQL datu baseekin, MongoDB adibidez.

Berez ez da REST estandar bat existitzen, metodoak egiten azaltzen duenik, SOAPekin alderatuta. Baita ere SOAP seguruagoa da REST baino. Beste desabantaila bat da, datu multzo handien tratamendua edo dei bat baino gehiago egitea konexiokoaren ezintasuna. Baina azken

2 REST (Reprentational State Transfer) HTTPren gainean exekutatzen den arkitektura bat.

biak konpondu egin dira, HTTP 2.x protokoloarekin, eta honen inplementazioa RESTfull³ zerbitzuetan .

4.3.1.3 GraphQL

GraphQL kode irekiko kontsulta eta datuen manipulaziorako lengoai bat da APIentzat. API web garapenean arreta jartzen du eta RESTekin eta beste arkitektura batzuekin konparatu eta egiaztatu egin da. Bezeroei eskatutako datu egitura definitu dezakete eta zerbitzariak datu egitura berdina bueltatuko du, horrela datu kantitate handiak bueltatzea saihestuko dugu. Lengoiaren malgutasuna eta aberastasuna konplexutasuna gehitzen dio, eta gerta daiteke GraphQL ez merezi erabiltzea API sinpletarako.

Hauek dira GraphQL-ren abantaila handienak:

- Batzuetan baliabide bakar bat behar da, baina zerbitzariak denak bueltatzen ditu, GraphQL, ordea, aplikazioa da baliabidearen kontrola duena, banda zabalera eta denborak murriztuz.
- GraphQL-rekin baliabide desberdinetara sar gaitzke dei bakar batekin.
- APIen bertsioikatzea ezabatzea saiatzten da, atributuen erabilera deprekatuz.
- GraphQL datu basearen sarbidearen menpekotasuna banatzen du.

Beste alde batetik, nahasgarria dirudien arren, SQL ez du zerikusik GraphQL-rekin, SQL datu baseekin lan egiteko pentsatuta dagoelako eta GraphQL bezeroak eta zerbitzariak komunikatzeko. GraphQL edozein programazio lengoiaiaz erabili daiteke, besteak beste, Node.js, Python, Ruby, C#, Java, etab.

GraphQL zerbitzuak orokorrean JSON erabiliz erantzuten dute, baina ez da beharrezkoa JSON erabiltzea. JSON oso ezaguna da bezero eta API garatzaileentzat, eta irakurtzeko oso erraza da. Gainera, GraphQL-ren sintaxia JSON-en inspiratuta dago. RESTekin ezin dugu aukeratu zer datu nahi ditugun JSON erantzunean, aldiz, GraphQL-n, zehazki aukeratu dezakegu behar duguna.

4.3.1.4 OData

OData (Open Data Protocol) jardunbide egokien multzo bat definitzen du, API REST-ak eraikitze eta kontsumitzeko.

OData, datu-baseetarako SQL estandarraren, web baliokidea da. SQL lengoai bat da, non DBKS baten datu iragaziak, ordenatuak eta proiektuak eskatu ahal ditugu, eta OData sintaxi bat da operazio berdinak eskaintzen dituena, baina HTTP protokoloaren bidez eskuragarri dauden datu iturrietan.

OData-k manipulazio aukerak dauzka kontsulten gainean, REST bakarrik erabiliz ezingo genuena lortu. Baita ere modu bat dauka kontsumituko den informazioaren metadatuak kontsultatzeko.

3 RESTfull erreferentzia egiten du REST arkitekturaz inplementatuta dagoen zerbitzariari.

OData-k formatu desberdinak erabiltzen ditu datuak eta datu-modeloa irudikatzeko. JSON formatua estandarra da datuak irudikatzeko. Datu-modeloa irudikatzeko Eskema Arruntan Definizio Lengoia (CSDL dira bere siglak ingelesez) erabiltzen da, OData zerbitzuetan dagoen datu-modeloaren XML irudikapen bat definitzen duena.

4.3.2 Informazio trukaketa

Helburua da informazioa trukaketa definitutako estandarren bidez makinaren artean, eta berdin du zer sistema eragile duten edo zer programazio lengoia egin den softwarea. Hauek dira erabilerak:

- Zerbitzaritik eta zerbitzarira datuak sartu eta atera.
- Aplikazioaren konfigurazioa fitxategi bakar batean. Fitxategi hauek konfigurazio parametroak eta beren baloreak dituzte.
- Aplikazioaren osagai desberdinen artean datu transferentzia.
- Bitarteko datu biltegiak.

4.3.2.1 XML

Estandar hau garatuta dago web erabileran egokitzeko.

Sistemen arteko datu trukaketa egiteko XML da gehien erabilitako formatu bat. Estandar honen formatuak testuan oinarrituta dago informazio egituratua irudikatzeko: datuak, dokumentuak, konfigurazioa, etab.

XML-ak informazioa egituratua dauka, ondo definitutako partez osatuta dago, eta parte horiek beste parte batzuk dituzte bere barnean. Informazio egituratu hori irudikatzeko etiketak erabiltzen dira. Etiketa bat elementu bat da eduki bat izan dezakeena, elementuak edo biak, elementu hutsak izan dezakeen arren. Etiketa bat beti "<" eta ">" baloreen artean definitzen da eta etiketa isteko "</>" erabiltzen da. Etiketa horren barruan atributu bat, batzuk edo bat ere ez izan dezake.

XML-ren abantaila bat JSON-en gainean da XML metadatuak izan dezakeela etiketetan atributu eran. Baita ere XML-k mekanismo bat dauka izenen arazoak konpontzeko, aurrizkiak. Aurrizki hauekin programatzaileak aukera dauka bi entitate desberdinak izen berdinarekin izendatzeko.

Beste abantaila bat da, ia nabigatzaile guztiak XML prozesatzeko ahalmena dutela.

4.3.2.2 JSON

Informazioaren trukaketarako formatua, irakurgarriagoa gizakiarentzat eta XML bezain eraginkorra makinaren arteko komunikaziorako. JavaScript programazio lengoia erabiltzeko batean oinarrituta dago. Estandar hau programazio lengoia erabiltzeko eta exekutatzeko duen sistemaren independente da.

JSON izen eta balio parez edo baloreen lista ordenatuen bilduma batez eraikita dago. Programazio lengoai askotan antzeko elementuak ditugu, adibidez, objektuak, egitura bat, hiztegi bat, array bat, ... Elementu edo objektu bakoitza giltza batez hasi eta bukatu egiten da. Izen bakoitza bi puntuz jarraitzen da eta zuhaitzaren adar berdinean dauden elementuak komaz banatzen dira.

JSON-en abantaila bat da XML askoz zehatzagoa dela. JSON oso tamaina txikia dauka eta denbora gutxiago behar du konprimitzeko. Horregatik, transmisioa eta prozesamendua azkarragoa da. Beste alde batetik, JSON-en prozesamendua XML-rena gaitzen du CPUaren baliabideen erabileran.

JSON-en beste abantaila bat da bere objektuen eta matrizeen irudikapenak mapaketa zuzena egiten uzten duela erabiltzen ari garen programazio lengoaiaren datu egiturekin. Lehen esan bezala JSON JavaScript-en azpimultzo bat da, horregatik JavaScript kodea erabiltzea oso naturala da. JavaScript-entzat oso onuragarria da, baina beste lengoaietarako ez. Ordea, 175 JSON erreminta baino gehiago daude 64 programazio lengoai desberdinetarako.

4.3.2.3 YAML

Datuak serializatzeko lengoai bat da, JSON eta XML bezalakoak. Koska formatua erabiltzen du egitura definitzeko. Hauek dira abantaila batzuk:

- Komentarioak onartzen ditu, JSON ez.
- Datu mota konplexuekin lan egin dezake.
- YAML dokumentua JSON-ena baino txikiagoa da.
- YAML JSON-en supermultzo bat da.
- YAML dokumentu baten barruan JSON dokumentu bat izan dezakegu.
- YAML dokumentu batek YAML multzo bat baino gehiago izan dezake.

Baina baita ere, bere desabantailak ditu:

- JSON-en prozesatzailea YAML-ena baino errazagoa da.
- Informazio trukaketan, JSON egokiagoa da.
- YAML-en prozesatzailea oso konplexua da.

YAML-n abantailak eta desabantailak ikusten, oso egokia da konfigurazio fitxategientzat erabiltzea. Baina APIen erabilpenean JSON gomendagarriagoa da.

4.3.3 APIaren garapen teknologia

Programazio lengoai baten aukeraketa faktore desberdinetan oinarrituko da:

- Lengoaiarekin esperientzia.
- APIak sortzeko dituen framework-ak.
- Euskarri eta komunitate aktibo bat duen.

4.3.3.1 Java

Java-ren bidez APIak sortzeko Spring Framework-a aurkitu dugu, kode irekikoa. Spring honako ezaugarri hauek ditu:

- **Datuen atzipena:** DAO, JDBC, ORM, Marshalling XML euskarria.
- **Transakzioen kudeaketa.**
- **Integrazioa:** urruneko komunikazioa, JMS, JCA, JMX, posta elektronikoa, ...
- **Frogak (testing).**
- **Alderdietan oinarritutako programazioa.**
- **MVC (Modelo Vista Controlador).**
- **Seguritatea.**
- **Taldeko datuen prozesamendua.**
- **Urruneko administrazioa.**
- **POJO (Plain Old Java Object):** ez du programatzailea behartzen klaseak heredatzen edo interfazeak inplementatzen.

Hauek dira Spring erabiltzearen abantailak:

- Era modular batean antolatua dago. Nahiz eta pakete eta klase asko izan, bakarrik arduratu behar gara gure garapenak behar dituztenetan, eta besteak alde batera utzi.
- Aplikazioa frogatzea prozesu sinple bat da.
- Teknologien salbuespen zehatzak (JDBC, Hibernate edo JDO sortutakoak) itzultzeko API bat dauka.
- Transakzioen kudeaketarako interfaze bat eskaintzen du.
- Teknologia desberdinen txantilo bat dauka, haien artean honakoak aurkitzen ditugu: JDBC, Hibernate eta JPA. Horrela ez da beharrezkoa kode luzerik idaztea, txantiloiekin lana murrizten dugulako.
- Dokumentazio handia dauka webean, eta komunitate handi bat dauka.

4.3.3.2 JavaScript (Node.js)

API bat garatzeko JavaScript-en bidez Express framework-a aurkitu dugu, Node.js-ren framework ezagunena. Mekanismo hauek eskaintzen ditu:

- Eskaeren kudeatzaileen idazketa, HTTP aditz desberdinekin URL bide desberdinetan.
- Bisten renderizazio motorren bidezko integrazioa.
- Aplikazioetan ezarpenak ezarri, adibidez zer portu erabiliko den.
- “Middleware” prozesu eskaerak gehitu.

Express.js zerbitzari framework bat da Node.js web aplikazioentzat, zehazki diseinatuta dagoena horri bateko, horri askotako eta hibridoak diren web aplikazioak sortzeko. Node.js-ren zerbitzari framework estandarrean bihurtu da. GET, PUT, POST eta DELETE eskaerak behar dituen aplikazio bat oso erraza da inplementatzea Express.js bidez.

4.3.3.3 Python

Python-en API bat garatzeko Django REST framework-a aztertu dugu. Hauek dira Django erabiltzearen abantailak:

- Nabigatzailetik nabigagarria den APIa, horrela garatzaileen lana azkartzen da.
- Integrazioa autentifikazioarekin, OAuth1-n edo OAuth2-n oinarrituta.
- ORM bidez datuak serializatu daitezke.
- Dokumentazio oso ona eta komunitate handi bat, kode irekikoa izanik.

Hiru osagaietan oinarritzen da:

- **Router-ak:** gure APIaren url-ak definitzeko erremintak dira, modu erraz eta antolatuan.
- **Bistak:** Django-ren *class-view* klasetara luzapenak dira.
- **Serializadoreak:** gure APIak bueltatuko dituen erantzunak nola izango diren zehazki definitzeko, eta nola prozesatuko ditugun iristen diren eskaerak definitu.

Django REST oso erreminta indartsua da, osoa eta heldua, APIak eraikitzea modu azkar eta erraz batean ahalbidetzen diguna.

4.3.3.4 PHP

Kasu honetan Laravel framework-a aukeratu dugu. Kode irekikoa da eta beste framework-en abantailaz baliatzen saiatzen da, eta baita ere PHPren azken bertsioak erabiltzen, garapenerako. Bere filosofia da PHP kodea garatzea modu elegante eta sinple batean, MVC modeloan oinarrituta. Laravel dituen ezaugarri nabariak hauek dira:

- **Blade:** txantilo sistema bat da, Laravel bidez bistak sortzeko.
- **Eloquent:** Laravel daukan ORMa⁴, datu baseak kudeatzeko prozesuak modu erraz batean egiteko.
- **Routing:** bideen antolatze eta kudeatze sistema.
- **Middleware:** kontrolatzaile batzuk dira, zerbitzariaren eskara bat egin aurretik eta ondoren exekutatzen direnak.
- **Komunitatea eta dokumentazioa:** azpimarratzeko puntu bat da, komunitate oso handia dagoelako eta dokumentazioa kalitate handikoa delako.

Laravel-en potentzia integrazioan, eskalagarritasunean eta mantentze erraztasunean datza.

4.3.4 Merkatuan dauden APIak

DBKS sistema batzuen hornitzaileak baita ere APIak eskaintzen dituzte, datu sarbidea errazteko.

4.3.4.1 Teradata

Teradata-k Data Mover Restfull API eskaintzen du. HTTPS estandarraren funtzioak erabiliz interfaze bat eskaintzen digu datuak mugitzeko. Edozein REST bezero erabilia Data Mover-en RESTfull zerbitzura sartu gaitzake.

Baita ere, lanak definitu eta editatu egin dezakegu, datu zehatzen objektuak kopiatzeko, taulak, erabiltzaileak, bistak, ... bezalakoak, Teradata sistema batetik beste sistema batera. Datu kantitate handiak kopiatzea ahalbidetzen digu Teradata sistema batetik beste sistema batera.

4.3.4.2 Oracle

Oracle REST Data Services (ORDS) REST interfazeen garapena errazen du, Oracle Database-aren datu erlazionalentzako. Erantzunak JSON bidez ematen ditu.

4.3.4.3 MySQL

MySQL-k API C eskaintzen digu, C aplikazio batetik sarbide zuzena izateko MySQL-ra. Hau metodo nagusia da MySQL-ra konektatzeko, eta komando lerroko bezeroak erabiltzen dute naiz beste MySQL konektoreak eta bitartekoen APIak.

4 **ORM (Object-Relational mapping):** Objektu-erlazional mapaketa.

4.3.4.4 SAS

SAS Viya da eskaintzen den APIa eta REST printzipioetan antolatuta dago. API honekin, SAS baliabideak sortu eta sartu gaitzke, edozein bezero teknologia erabiliz. Honek SAS Viya-ren baliabideak integratzen errazten du.

4.4 Kudeaketa aplikazioa

Puntu honetan bi galdera erantzun behar ditu. Alde batetik zein teknologia erabiliko dugun bezero naiz zerbitzari aldean, eta beste alde batetik, ea komenigarria zaigun framework bat erabiltzea. Ondoren bi galdera hauei erantzuteko azterketa dago:

4.4.1 PHP

PHP kode irekiko lengoaiarik ezagunena da eta oso egokia da web garapenerako eta gainera HTML-an txertatu daiteke. Komando asko erabili ordez HTML ikuskatzeko, PHP orriak HTML kodea txertatuta daukate. PHP kodea etiketa batzuen barruan dago.

PHP kodea zerbitzarian exekutatzen da, HTMLa sortzen eta bezerora bidaliz. Bezeroak script-a exekutatzearen emaitza jasoko du, baina honek ez daki zein den kodea. Zerbitzari prozesatu daiteke HTML fitxategi guztiak PHP-ren bidez prozesatzeko.

PHP zerbitzari aldeko script-en programazioan zentratuta dago. Hiru alderdi daude non PHP script-ak exekutatzen dituztenak:

- Zerbitzari aldeko script-ak.
- Komando lerroko script-ak.
- Mahaigaineko aplikazioak idatzi.

Erabili daiteke sistema eragile nagusi guztietan. Baita ere, gaur egungo ia zerbitzari guztiak PHP onartzen dute. Modu honetan, sistema eragilea eta web zerbitzaria aukeratzeko askatasuna izango dugu.

PHP-ren ezaugarri garrantzitsuenetariko bat da datu base askorentzako euskarriak dituela. Datu basera sartzea oso erraza da, datu base zehatzaren luzapena erabiliz, edo PDO erabili. Honekin ODBC luzapena erabiliz edozein datu basetara konektatu gaitzke.

Baita ere framework asko aurkitu ditugu, programazio lanak errazteko asmoz. Hauek aztertu egin ditugu:

4.4.1.1 Medoo

Datu basetarako framework bat da. Oso arina da, fitxategi bat behar duelako bakarrik, erabiltzeko eta ikasteko erraza da eta errendimendu alturako optimizatua dago. Oso egokia da edozein proiektuentzat, garapena PHP-n eta SQL datu base bat behar baldin badu.

SQL kontsulta batzuk onartzen ditu, sinpleak naiz konplexuak. Baita ere, datu mapaketa onartzen du eta SQL injekzioa saihesten du. SQL datu base guztiak onartzen ditu, haien artean, MySQL, MSSQL, SQLite, MariaDB, PostgreSQL, Sysbase, Oracle eta askoz gehiago.

Framework PHP guztiekin ondo lan egiten du, Laravel, CodeIgniter, Yii, Slim, etab. bezalakoak.

4.4.1.2 Laravel

Framework hau ezaguna da bere sintaxi dotoregatik, oso ulergarria dena eta plazer bat dena lan egiteko. Laravel-ekin oso azkar hasi gaitzke gure proiektuetan lan egiten. Oinarrizko elementu asko alde batera utzi dezakegu, funtzio asko gure eskura izango ditugulako, erabiltzaileen autentifikazioa, sesioen administrazioa edo cache biltegitratzea bezala.

Laravel-en nukleoa oso egonkorra da, errendimenduaren aldetik, eta gainera framework-ak handitu dezakegu luzapenak erabiliz. Beste liburutegi batzuekin integratzen da.

Hauek dira bere ezaugarri nagusiak:

- Luzapenak erabiliz Laravel-en funtzionalitate nagusia handitu daiteke.
- Integratutako funtzioak dauzka, oinarrizko gauzetan denbora ez galtzeko.
- Beste liburutegi batzuekin Laravel integratu.
- Beste lan batzuk era asinkronoan exekutatu, bigaren maila batean, errendimendua hobetzeko.

Laravel datu baseen integrazioa asko errazten du. Kontsulta eraikitzaile bat dauka eta ORM Eloquent erremintarekin mapaketa asko errazten da. Gaur egun, datu base hauek onartzen ditu:

- MySQL
- PostgreSQL
- SQLite
- SQL Server

4.4.1.3 CodeIgniter

MVC arkitektura erabiltzen duen PHP framework bat da. Honek web aplikazioak sortzen lagunduko digu, oso eskalagarriak direnak, tamaina txiki batekin. Dokumentazio bikain bat dauka. Errendimendu egonkor bat dauka, honek esan nahi duena da, oso egokia dela aplikazio arinak sortzeko zerbitzari apaletan ondo exekutatzeko.

Hauek bere ezaugarri nagusiak:

- Framework arina, errendimenduan pentsatuta egina.
- Framework-aren erraztasuna eta dokumentazio bikaina.
- Aplikazio eskalagarriak MVC arkitektura erabiliz.

4.4.2 Java

Java ikasteko oso zaila den programazio lengoia bat da, ez hainbeste bere sintaxiagatik edo objektuei orientatuta dagoelako, baizik eta bere zabaltasunagatik. Java barruan liburutegi asko aurkitu dezakegu, edozein gauza egiteko.

Filosofia bakar bat dauka: “behin idatzi eta exekutatu edozein lekutan”. Beste era batean esanda, behin programatzen duzu eta edozein sistemetan exekutatu dezakezu. Hau Java-ren Makina Birtualari (JVM) esker da, eta diseinatuta dagoena edozein sistema eragiletan funtzionatzeko.

Java objektuei zuzenduta dago, seguruenik garatzaileei gehien gustatuko zaien ezaugarria da. Objektuei zuzendutako programazioa paradigma bat da arazoak konpontzen laguntzen diguna, objektuetan plasmaturik eta ondoren kodera eramanez modu eraginkor eta azkar batean.

Java-k kode gutxiago idazten laguntzen dugu, horrela kalitatezko kodea izango dugu eta gure kodifikazio praktikak hobetuko ditugu. Hau dudarik gabe, objektuei zuzendutako programazioari esker da.

Datu baseen sarbidea JDBC estandarraren bidez egiten da Java-n, sarbide uniforme bat eskaintzen digu, berdin du zein den DBKS. Beraz, Java-z idatzitako aplikazioak ez dute DBKS-ren espezifikazioak ezagutu behar, bakarrik JDBC-ren funtzionamendua ulertu behar da.

PHP eta Java-ren artean konparaketa bat egiteko, taula hau sortu dugu:

	PHP	Java
Kostuak	Merkeago	Garestiago
Errendimendua	Errendimendu ona	Errendimendu ona
Seguritatea	Seguritate gutxiago	Seguragoa
Sintaxia	Lagungarria	Konplexua
Eramangarritasuna	Eramangarria	Eramangarria
Garapen denbora	Denbora gutxi	Denbora gehiago
Euskarria eta laguntza	Ona	Ona

4.1 Taula: PHP eta Java konparazioa

Baita ere framework batzuk aztertu ditugu, kodeketa errazteko. Hona hemen aztertutakoak:

4.4.2.1 Spring MVC

Web aplikazioetarako eskala handiko Java framework-a da, ingeniariari erreminta ahaltu bat eskaintzen die web aplikazioak garatzeko. Zerbitzu asko eskaintzen ditu: API REST, SOAP web zerbitzuak, seguritatea, etab. hauek dira Spring-en abantaila batzuk:

- Froga datu txertaketa errazak, POJO erabilpenari esker.
- Modularitate obetua, kodean irakurgarritasuna eragiten duena.
- Modulu desberdinen akoplamendu malgua.
- Erabilpen malguko dependentzien txertaketa.

Garatzaileek 20 modulu inguru eskura dituzte, eta gainera aukera dezakete zeintzuk erabili eta zeintzuk ez. Moduluak kategoria desberdinetan antolatzen dira eta horietako bat Data Access/Integration da. Kategoria honetako moduluak diseinatuta daude Java aplikazioei eskaintzeko beharrezko tresnak beste aplikazioekin elkarreragiteko, adibidez datuen sarbidea kontrolatzeko. Adibidez, spring-jdbc moduluak abstrakzio kapa bat dauka bezeroak nola sartzeko den datu basean definitzen duena, eta JDBC kodifikazioa aurrezten digu.

4.4.2.2 Struts 2

Java EE-kin eginiko web aplikazioak egiten dute garatzaileentzat framework hau izugarritzko aurkikuntza bat da. Apache erreminta multzo handi bat eskaintzen ditu enpresei orientatutako web aplikazioak sortzeko, garapen prozesua optimizatzen hasieratik bukaerara. Struts 2 aukera egokia da webak datu karga handi bat jasango badu.

Garatzaile askok ez dute oso arin aurkitzen berarekin lan egiteko.

Struts 2 MVC framework bat da eta ez datu base framework bat, baina JPA / Hibernate integrazioa onartzen du.

4.4.2.3 Hibernate

Java-ren mapaketa framework bat da eta objektu-erlazional inpedantziarekin hautsi egiten du. Hibernate ezaugarri handiekin hornituta dago, garatzaileei datuak atzitzen laguntzen diena. ORM framework aurreratu bat da, Java objektu datu basean operazioak egiten uzten diguna.

Hibernate edozein datu baserekin komunikatu daiteke, kodearen eguneraketa xamurrekin. Baina bere desabantailak ere baditu, Hibernate-k ez ditu txertaketa handiak egiten uzten, JDBC-k egiten dituen kontsultak saihesten. Framework hau ez da gomendagarria proiektu txikiatarako.

4.4.3 JavaScript

Orain arte aztertutako programazio lengoaiak zerbitzari aldekoak dira, baina JavaScript bi aldetarako balio du, zerbitzarirako naiz bezerorako.

Backend garatzaileak JavaScript mota berezi bat erabiltzen dute, Node.js. Node.js ez da programazio lengoai bat, bakarrik software ingurune bat da, JavaScript exekutatzen duena. Liburutegi asko dauzka eta hauek datuen sarrera eta irteera eskaintzen dute, beste gauzen artean. Modu honetan JavaScript erabili dezakegu nabigatzailearen kanpotik, eta backend lanak exekutatu.

Node.js-ren beste abantaila bat abiadura da. HTTP naiz S/I eskaerak oso azkarrak dira, ez du lan handia eskatzen eskaerak prozesatzea, horrela instantzia bakar batek eskaera asko prozesatu dezake hardware simple batekin.

Node.js datu base mota asko onartzen ditu, berdin du datu base erlazional bat den edo NoSQL datu base bat. Baina, NoSQL datu baseak, MongoDB bezalakoak, egokiagoak dira Node.js-rentzat. Datu base bat atzitzeko, lehenengo gauza datu base horren kontroladorea instalatzea da. Ondoren kontroladore horren bidez datu basea atzitu dezakegu. Hauek dira onartzen diren datu base erlazionalak: SQL Server, Oracle, MySQL, PostgreSQL, SQLite, etab. gainera, baita ere, NoSQL datu base hauek onartzen dira: MongoDB, Cassandra, LevelDB, Redis, etab.

Hauek dira Node.js erabiltzeko aztertu ditugun framework-ak:

4.4.3.1 Express

Node.js-ren framework famatuena eta erabiliena da. Framework nahiko handia da eta gure betebeharrak guztiak konponduko ditu. Erabilpen kasu txikietarako edo oso konkretuentzat, gerta daiteke Express ez izatea aukerarik hoberena.

Express web aplikazioentzat oinarritzko ezaugarri kapa bat ematen digu, baina ez ditu Node.js-ren ezaugarriak estaltzen.

Datu base bat konektatzeko Node.js-ren funtzionalitate berdinak erabiliko ditugu, framework-ak ez du laguntza gehigarri ematen kasu honetarako.

4.4.3.2 Koa

Koa web framework-en belaunaldi berria da. Express sortu zuten garatzaileak sortu dute. Hauek dira beren ezaugarri batzuk:

- Tamaina murriztua.
- Metodoetan adierazpen handiagoa, hau da, beren izenak argiagoak direla.
- Egonkortasun handiagoa web aplikazioak eta APIak sortzerako garaian.

Honekin saiatu nahi dira framework hau Express baino hobetua izaten. Koa funtzio asinkronoaz baliatzen da, horrela, deien erantzunez desegiten gara eta errorearen kudeaketa handitzen dugu.

Baina lehen esan bezala JavaScript bezero aldeko programazio lengoia da, berez honetarako sortu egin zen. Eta hauek dira JavaScript bezero aldean erabiltzearen abantailak:

- Konpilatzailea ez da beharrezkoa, nabigatzaileak HTMLarekin interpretatzen dute.
- Ikasteko erreza da.
- Erroreak erraz detektatzen dira.
- Orriko elementu konkretu batzuei bakarrik esleitu daiteke.
- Sarrerak egiaztatzeko erabili daitezke eta datuen eskuzko egiaztapenen beharrak murriztu.
- Webak interaktiboagoak izaten laguntzen du.
- Beste programazio lengoaiak baino azkarragoa eta arinagoa da.

Baina bere ahultasunak ere badauzka, adibidez, mehatxuei ahula da. Gainera, erabiltzaile baten konputagailuan kode maltzurra exekutatzeko erabili daiteke. JavaScript kodea nahiko handia izan daitezke ere. Eta azkenik, JavaScript modu desberdinetan ager daiteke gailu desberdinetan, eta honek sendotasun eza sor dezake.

Hona hemen aztertu ditugu framework-ak, JavaScript bezero aldean erabiltzeko:

4.4.3.3 React

Facebook sortutako liburutegi bat da eta bisten sorkuntzan oinarritzen da. Berezitasun batzuk ditu, gertaera ereduak bezala, aipatutako bistak eguneratzen dituena datuak aldatu egin direnean, gure garapenari denbora errealeko karga sortuz.

Erabiltzailearen interfazeak sortzeko oinarritua dago. React bakarrik gure aplikazioaren bista kontrolatzeaz arduratzen da. Bere ezaugarri nagusien artean DOM birtualaren erabilpena aurkitzen da. Liburutegi honekin lan egiten dugunean, kontzeptu garrantzitsuena da gure garapena osagaietara bideratu behar dugula. Osagaiak gure kodea pieza txikietan banatzeko modu bat da, eta modu honetan berrerabili dezakegu osagai bakoitza.

React erabiltzaileen interfazeen txantiloak eta JavaScript logika konbinatzen ditu, gauza bat inor ez duela lehenago egin. Eraitza JSX deitzen da. Lehen aipatu bezala, React osagaiak erabiltzen ditu, eta markaketa eta logika fitxategi berdinean gordetzen ditu. Baita ere, XMLren lengoai antzeko bat erabiltzen du, markaketa kodean bertan idazteko aukera ematen duena. JSX abantaila handi bat da garapenerako, dena toki bakar batean dugulako.

React norabide bakarreko lotura erabiltzen du. Modeloaren egoera aldatzen bada erabiltzailearen interfazea automatikoki eguneratzen da. Baina alderantziz ez, erabiltzailearen interfazean aldaketa bat egiten badugu, modeloaren egoera ez da aldatzen. Horregatik norabide bakarrekoa da.

4.4.3.4 Angular

Angular MVC framework bat da web frontend garapenerako. SPA (Single-Page Applications) aplikazioak sortzen uzten dugu. HTML hiztegia handitzen ahalbidetzen digu direktiba eta atributuekin, semantika mantentzen eta kanpoko liburutegiak erabili gabe.

Angular-en programazioa TypeScript bidez egiten da, lengoi hau JavaScript-en supermultzo bat da.

Angular React baino funtzionalitate gehiago dauzka “erabiltzeko prest”. Ez dugu erabaki behar zer liburutegi erabili, zuzenean programatzen hasen gara. Baina, hau desabantaila bat izan daiteke, malgutasun gutxiago daukalako. Hau da, Angular ematen diguna erabili behar dugu.

Angular DOM erregularra erabiltzen du, React-en kontrakoa. Honek HTML etiketen zuhaitzen egitura osoa eguneratzen du. Orri berdinean datu eskaera askorekin bagaude, errendimenduan txarto eragin dezake.

Angular lotura bidirekzionala erabiltzen du. Honekin erabiltzailearen interfazean aldaketa bat egiten badugu, modeloa automatikoki aldatuko da. Eta gainera, alderantziz ere, hau da, modeloan aldaketa bat gauzatzen bada, interfazean automatikoki eguneratu egingo da.

4.5 Datu-garbiketa

Datu-garbiketa gauzatzeko mi modu daude. Lehenik, erreminta bat erabiliz, edo gure kodea erabiliz. Beraz, gure azterketa bide horretatik joan da, lehenik datu garbiketa egiten duten erremintak aztertu ditugu eta ondoren zer teknologia egokiak izango liratekeen kalitatezko garbiketa bat egiteko.

4.5.1 Datu-garbiketa erremintak

Bi mota bereizi egin ditugu aztertutako erreminten artean. Alde batetik ETL erremintak datu-garbiketa barne dutenak, eta beste alde batetik datu-garbiketa erreminta espezifikoak.

4.5.1.1 Talend Open Source Integration Software

Erreminta hau lehenik ezaguna zaigu, ETL atalean ikusi dugulako. Beraz ETL erreminta bat da eta datu-garbiketa gauzatzen du. Datu kalitatearen erroreen erabakiak automatizatu dezakegu eta erregelak aplikatu. Baita ere datu kalitatearen atari bat dauka, kontrolagailuekin, txostenekin eta panelekin. Azkenik, Talend aberasketa, harmonizazioa, parekatze zehaztugabea eta deduplikazioa eskaintzen digu.

4.5.1.2 Pentaho Data Integration – Kettle

Erreminta hau baita ere lehenik aztertu dugu, ETL prozesuan, beraz, baita ere, ETL erreminta bat da, datu-garbiketa barne daukana. Pentaho erabiltzaileari iturri desberdinetako datuak

garbitzen eta prestatzen ahalbidetzen dio eta baita ere aplikazioen arteko datu migrazioa. Pentaho-ren ahalmenak datuen kalitatean laguntzen dute erregelak betetzen ez duten datuak identifikatzen, sendotasun gabeko eta erredundanteak diren datuak garbitzen eta deduplikatzen, eta azkenik, izenak, helbideak, helbide elektronikoak eta datu telefonikoak balioztatzen, estandarizatzen eta zuzentzen.

4.5.1.3 Winpure

Datu-garbiketa erreminta ezagunetariko bat da. Gainera nahiko merkea da eta esfortzurik gabe datu kantitate handiak garbitu dezake, errepikapenak ezabatu, zuzendu eta estandarizatu dezake. Datu basetatik, kalkulu orritatik, CRM eta toki gehiagotatik datuak garbitu dezake. Bere ezaugarrien artean hauek aurkitzen ditugu, datuen garbiketa aurreratua, bateragarritasun zehaztugabea, datuen arazketa oso azkarra, edizioa hizkuntza batzuetan, besteen artean.

Besteen artean desberdintzen duena da bere ahalmena datuak konparatzeko eta horrela kointzidentzia erreal gehiago aurkitzea faltsuak baino. Kointzidentzia hauek aurkitzeko funtzio hauek ditu:

- **Izengoitiaren kointzidentzia:** pertsonen izenak, inizialak eta ezizenak identifikatze ditu.
- **Enpresen izenen kointzidentzia:** metodo aurreratuak definizio desberdinak dituzten enpresen izenak identifikatzeko.
- **Kaleen helbideen kointzidentzia:** helbide berdinak identifikatzeko laburdura desberdinekin.
- **Telefono zenbakien kointzidentzia aurreratua:** kode internazionalekin, bertako kodeekin edo hutsuneekin dauden zenbakiak identifikatzeko.

Lizentziaren prezioa 999\$ gainditzen du.

4.5.1.4 Data Ladder

Data Ladder erabiltzaileei beren datuak gehiena aprobetxatzen laguntzen dedikatzen da. Laguntza hori aberasketa erremintekin, profilen sorketarekin, deduplikazioarekin eta datuen aberasketarekin gauzatzen da.

Gatu garbiketa plataforma oso ikusgarria, datuen kalitatearen arazoak aurkitzeko eta konpontzeko diseinatu egin dena.

Bikoizketak identifikatu eta ezabatzen ditu edozein iturritatik. Deduplikazio software azkarrena eta zehatzena dela kontsideratzen da.

Ia edozein datu base integratu dezakegu, SQL Server, Oracle, Teradata, MySQL, PostgreSQL, MongoDB, Google Big Query, Snowflake eta gehiago bezalakoak. Gainera fitxategi mota desberdinak integratu dezakegu, Excel, Google Spreadsheets, CSV/TSV, XML o TXT moduak.

4.5.2 Datu-garbiketarako garapen teknologiak

Gure garapenarekin datu-garbiketa gauzatu dezakegu, baina bakarrik bi liburutegi aurkitu egin ditugu gure lana erraztu egingo dutenak arlo honetan. Hauek Pandas liburutegia da Python-entzat, eta Dplyr liburutegia R-entzat.

4.5.2.1 Pandas

NumPy liburutegiaren luzapen bezala idatzita dago liburutegi hau eta datuen manipulaziorako eta analisirako egina dago, Python programazio lengoaiarentzat. Datu egiturak eta operazioak taulak manipulatzeko eskaintzen ditu. Hauek dira Pandas-en ezaugarri batzuk:

- DataFrame datu mota, datuen manipulaziorako.
- Datuen lerrokatzea.
- Datu multzoen berregituraketa eta segmentazioa.
- Datu egituretan zutabeen txertaketa eta ezabatzea.
- Datuen nahasketa eta bateratzea.

4.5.2.2 Dplyr

Dplyr paketeak funtzio asko eskaintzen ditu, data frame-ak manipulatzeko eta horrela errepikapenak, erroreak gauzatzeko probabilitatea eta idatzi beharrezko karaktere kopurua murrizteko. Bere gramatika oso ulergarria da. Pakete honek pausu hauek oso erraz eta azkar egitea laguntzen du:

- Aukerak ezabatzeagatik, datu manipulazioaren erronkan pentsatzea laguntzen du.
- “Aditz” sinpleak eskaintzen ditu, datu manipulaziorako bat datozen funtzioak, gure ideiak kodean itzultzen laguntzeko.
- Software eraginkorrak erabiltzen ditu, horregatik denbora gutxiago pasatzen dugu konputagailuari itxoitzen.

4.6 Datu-iturriak

Atal honetan arazo handiak izan ditugu, Donostia Kulturak ez duelako beren datu iturriak erraztu. Beraz Adimediak bere kontrolpean duen datuak erabili ditugu. Donostia Kulturaren web orriko datu basea izan da lortu dugun datu-iturri bakarra. Beraz, DK Webaren datu basea aztertuko dugu.

4.6.1 Ezaugarriak

Donostia Kulturaren web orria kudeatzeko erabiltzen da. Online zerbitzari batean kokatzen da, Adimedian bertan. Datuen jabea Donostia Kultura da berez, baina haien kudeaketaz Adimedia arduratzen da. Erabilitako teknologia MySQL da, eta PHPMyAdmin bidez datuak kudeatzen dira.

Datu hauek, lehen esan bezala web orria kudeatzeko erabiltzen dira. Web orri honetan Donostia Kulturako bazkideak logeatu daitezke, beraz bazkideen datuak daude. Datu hauek EKI datu basetik hartzen dira, egunero. Hau da, bazkideen datuak EKI datu basean daude, Donostiako Udalaren datu basea, eta egunero, gauean, fitxategi konprimitu bat prestatzen da, eguneratu diren bazkideekin edo bazkide berrieekin. Fitxategi horrekin DK Webaren datu basea eguneratzen da. Baita ere, datu gehigarriak daude bazkide hauen kudeaketarako, webean sartzeko behar duten pasahitza bezala.

Beste alde batetik, zozketen datuak daude. Beste era batean esanda, ze zozketa daude webean eta zer bazkide daude apuntatuta zozketa horietara. Baita ere buletinak aurkitzen ditugu. Buletinak, Donostia Kulturak bazkideei informazioa bidaltzeko erabiltzen du. Arlo desberdinak daude, zinea, musika, antzerkia, etab. bezalakoak, eta bazkide bakoitza nahi duen buletinera apuntatu daiteke, eta horrela nahi duen informazioa bakarrik jaso. Orduan, buletin hauen informazioa baita ere datu basean gordetzen da.

Azkenik, web orria Joomla eduki kudeaketa sistema erabiltzen du, eta honen kudeaketa egiteko beste taula batzuk ditu datu baseak, baino ez dute informazioa gehitzen gure datu biltegari.

4.6.1 Zer datu interesatzen zaizkigu?

Datu basearen taula guztiak ikusita bi talde berezitu dezakegu, alde batetik dk aurrizkia dutenak eta beste alde batetik xx aurrizki dutenak. dk aurrizkia duteak Joomla sortutakoak dira eta xx dutenak berriz Adimediak sortutakoak.

Guri interesatzen zaizkigun datuak bazkideen datuak dira, eta bazkide hori zer zozketa eta zer buletinera apuntatuta dagoen. Bazkideen datuak xx tauletan dago. Buletinen informazioa berriz dk tauletan dago eta azkenik, zozketen informazioa xx eta dk tauletan dago.

Taula hauek kontuan hartuko ditugu:

- **xx_socios_eki:** Lehen esan bezala, datu hauek berez EKI datu basean daude, eta datu biltegia osatzeko egokiena hortik datuak hartzea izango litzateke. Baina nola ez dugun sarbiderik, datu base honetatik egingo dugu analisisa.

Bazkideen datu pertsonalak gordetzen dira taula honetan, ikusten denez. Datu hauek guztiak datu-biltegira eramango ditugu.

- **dk_acymailing_subscriber:** Buletinetara apuntatuta dauden bazkideak.
- **dk_acymailing_listsub:** Taula honetan zer bazkide zer buletinera dagoen apuntatuta agertzen dira.
- **dk_acymailing_list:** Taula honetan buletinaren informazioa aurkitzen da.

- **dk_content:** Web orrian agertzen diren artikuluen informazioa. Artikulu batzuk zozketak dira, beraz zozketa horiek interesatzen zaizkigu.
- **dk_flexicontent_fields_item_relations:** Taula honetan artikulua bakoitzean agertzen diren eremuak daude. Adibidez, zozketa bat baldin bada, zozketatzen dena eta zozketara apuntatzeko eremuak agertuko dira, besteak beste.
- **dk_flexicontent_field:** Aurreko taulan azaldu diren eremuan deskribapena agertzen da taula honetan.
- **xx_socios_sorteos:** Zer bazkide dauden apuntatuta zozketetara agertzen da.

Atal honetako azterketa Donostia Kulturarekin batera egin beharko litzateke, jakiteko zeintzuk diren interesatzen diren datuak, eta datu-iturri guztien informazioa izateko. Horregatik datu guzti hauek hartuko ditugu gure datu biltegia inplementatzeko. Etorkizun batean proiektu hau aurrera jarraitzen badu, azterketa hau berriz egin beharko litzateke. Alde batetik aldaketak egon daitezkeelako datu base honetan, eta beste alde batetik, datu-iturri guztiak aztertu beharko liratekeelako.

Baita ere arrazoi honengatik, atal hau azkenerako utzi dugu, Donostia Kulturak informazio gehiago ematearen zain.

Gure analisiarekin jarraituz, datu base hau atzitzeko zuzenean datu basera konektatu behar gara gaur egun. Ez dago ez APIrik ez web zerbitzurik datuak atzitzeko, baina Adimediak ez du baztertzen etorkizun batean API bat inplementatzea datu basea atzitzeko.

Baita ere, datu biltegia eguneratua mantentzeko, periodikoki datuak hartu behar ditugu datu base honetatik. Datu biltegia ahal den eguneratuena mantendu behar dugu. Baina sinkronizazio maiztasuna oso txikia baldin bada, datu baseari gehiegizko lan-karga sortu diezaiokegu. Horregatik, maiztasun egoki bat esleitu behar diegu datuei, datu biltegia eguneratuen mantentzeko eta datu basea ez kargatzeko, bere lan nagusia (DK webaren datu basea izatea) berdin egiten jarraitzea.

5

Proposatutako soluzioa

Kapitulu honetan, aurreko atalean egindako azterketan oinarrituta erabaki bat hartuko dugu, gure arazorako soluzio optimoena izanik.

5.1 ETL prozesua

Aurreko azterketa ikusita, galdera bat dugu erantzuteko: ETL erreminta bat erabiltzen dugu edo gure garapen pertsonala egingo dugu?

Eskuzko kodifikazio metodoak egunak edo asteak behar dute, datuen kantitatearen eta konplexutasunaren arabera. Kode pertsonalizatua oso egokia da prozesua automatizatzeko oso erraza denean edo lana ez daukanean konplexutasun handirik. Gehienetan ETL erreminta batek ingurune bisual bat eskaintzen du, ETL prozesuaren zati handiak ikusten uzten digularik.

Azken urteetan ETL erremintek garatzaile batentzat, diseinatzaileentzat Photoshop erreminta bezala bihurtu da, ezinbestekoa. ETL erremintak erabiltzeko prest daude, erreminta hartu eta ETL prozesuak diseinatzen hasi gaitzke. Gainera, kode pertsonalizatu bat ezin da ETL erreminta batekin berdindu, nahiz eta arretaz egina eta dokumentatuta egon arren.

Beraz, hau guztiagatik, **ETL erreminta** bat erabiltzea erabaki dugu.

Baina orain beste galdera bat sortzen zaigu, zein ETL erreminta erabiliko dugu? Aztertutako erremintak ikusita, bere kalitateagatik bat besteen gainean dagoela uste dugu. Merkatuan dagoen ETL erreminta indartsuena da, eta ezaugarri hoberenak eskaintzen ditu, horregatik merkatuko liderra da. Erreminta hori **Informatica PowerCenter** da.

Baina hala ere, erreminta honek bi desabantaila handi dauzka. Alde batetik, datu kalitatearen ezaugarria ez duela, eta beste alde batetik, bere kostua oso handia dela eta enpresa guztiek ezin dute bere prezioa jasan.

5.2 Data Warehouse

Oraingoan bi galdera erantzun behar ditugu. Lehenik, zein DBKS erabiliko dugun, eta bigarrenik, zein izango den datu biltegi honen diseinua.

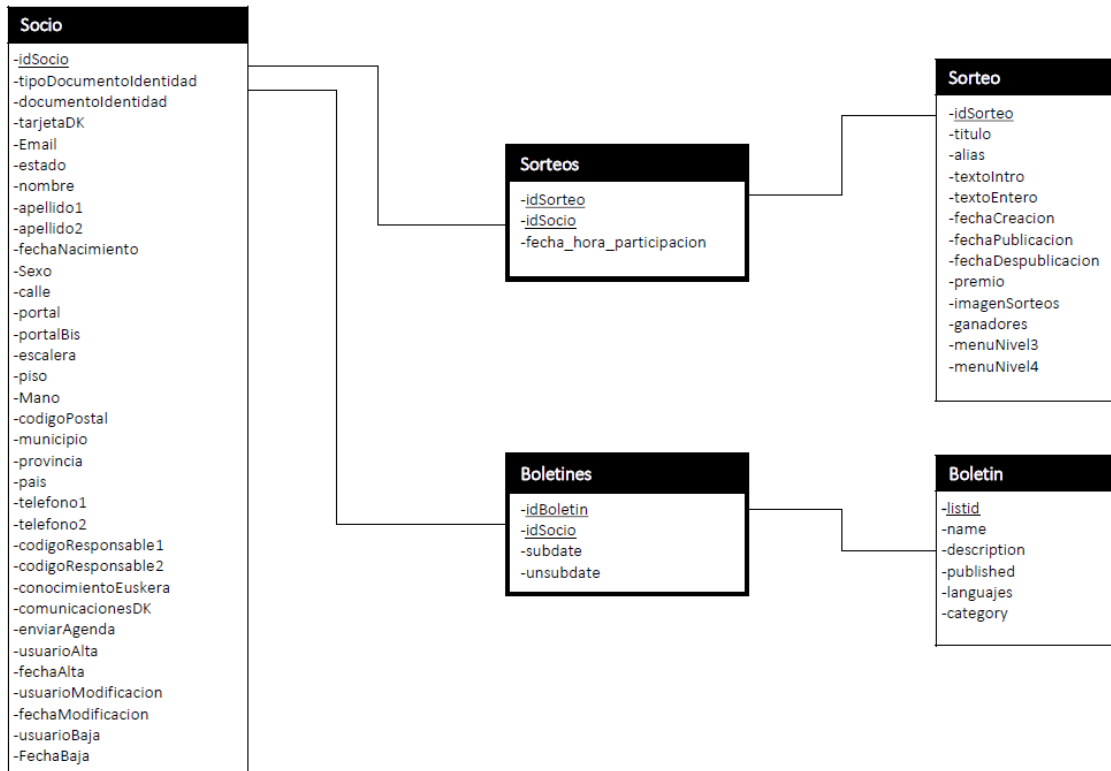
5.2.1 DBKS

Aزتututako DBKSen artean lau iruditu zaizkigu baliagarriak gure soluziorako. Sistema horiek Panoply, Teradata, Oracle eta PostgreSQL dira. Azkeneko hau kode irekikoa da, eta beraz doakoa, baina ala ere oso software baliagarria iruditzen zaigu. Lau hodeian egoteko aukera dute (Panoply zehazki bakarrik hodeian egon daiteke) eta bakoitzak bere ezaugarri desberdinak ditu.

Baina iruditzen zaigu, bat besteen gainetik dagoela, eta hori **Oracle** da. DBKS oso indartsua da, geroz eta datu biltegia handiagoa denean, geroz eta administrazioa eta errendimendua hobetzen da. Bere indarrarengatik, konfigurazioa zerbait zaildu daiteke, baina behin hori lortu egin dela, datu biltegi oso ahalmentsua izango dugu. Desabantaila bat jartzeko, bere kostua da. Oracle oso garestia da.

5.2.2 Modeloaren diseinua

Datu-biltegiak modelo zehatz bat dute, eta hirugarren kapituluaz azaldu den moduan izar eskema edo elur-maluta eskema erabilienak dira. Baita ere aipatu genuen galaxia eskema, eta hori da erabili duguna. Hemen eskema:



5.1 Irudia: Datu biltegiaren modeloaren diseinua

Galaxia eskema batean bi gertaeren taulak agertzen ziren, eta dimentsio taulak komunean zituzten. Kasu honetan, ikusi dezakegun moduan, *Sorteos* eta *Boletines* taulak gertaeren taulak dira. *Socio*, *Sorteo* eta *Boletin* taulak dimentsio taulak dira. Agerian denez *Socio* taula bi gertaeren taulak konpartitzen duten dimentsio taula da.

Orduan, modu honetan jarrita, egingo diren analisi eta azterketa guztiak datu hauetaz baliatuz, zozketen eta buletinen ingurukoak izango dira, guzti horiek bazkideekin lotuta egonik.

5.3 Datuen sarbiderako APIa

Kapitulu honetako beste ataletan bezala, galdera bat erantzun behar dugu. Merkatuan dagoen API bat erabiltzen dugu, edo gure API propioa garatzen dugu?

Galdera hau erantzuteko oso garrantzitsua da jakitea zer DBKS aukeratu dugun. Horien hornitzaile batzuek APlak eskaintzen dituztelako, aurreko kapituluan azaldutako moduan. Oracle izan da aukeratutako DBKS eta hornitzaileak Oracle REST Data Service (ORDS) eskaintzen digu baita ere. ORDS Oracle lizentziaren barruan sartzen da, beraz ez du kostu gehigarririk proiektuarentzat. Beraz oso ideia ona izango litzateke **Oracle REST Data Service** gure API bezala aukeratzea, lehenik Oracle aukeratu badugu gure DBKS bezala.

5.4 Kudeaketa aplikazioa

Kasu honetan bi galdera ditugu, aurreko kapituluko azterketa ikusita. Lehenik, zein teknologia erabiliko dugun zerbitzari aldea garatzeko, eta bigarrenik, zein teknologia erabiliko dugu bezero aldea garatzeko.

5.4.1 Zerbitzari aldea

Aztertutako teknologien artean Java iruditu zaigu aukera oso ona. Lehenik eta behin objektuei orientatua dagoelako, eta hau asko errazten du datuekin lan egitea. Baita ere, Javak, framework batzuk eskaintzen ditu datuen atzipenerako. Hibernate erreminta aztertu dugu, eta oso egokia da datu base erlazional batekin lan egiteko. Java objektuen eta modelo erlazionalaren arteko mapaketa egiten laguntzen du. Baita ere oso egokia izan daiteke Spring MVC framework-a erabiltzea. Honek aplikazioa garatzen lagunduko digu, kodeketa asko erraztuz bere ezaugarriekin. Beraz, gure aukera **Java** erabiltzea da, **Spring MVC** eta **Hibernate**-rekin batera.

5.4.2 Bezero aldea

Kasu honetan ez dago zalantza handirik, **JavaScript** da ia-ia da bezero teknologi bakarra. Frontend garapenerako teknologia erabiliena da HTML eta CSSrekin batera. Esan beharra dago ez dela bezero teknologia bakarra, baina ez dira oso ezagunak. Beraz hemen dugun galdera da ea framework bat erabiltzen dugun.

Egia esan oso interesgarri da framework bat erabiltzea. Oso aukera ona React iruditzen zaigu, alde batetik bere ezaugarriengatik eta erabiltzen duten erabiltzaile kopuruengatik, eta beste alde batetik, Adimedian esperientzia dagoelako erreminta honekin. Beraz ez ginateke zerotik hasi behar teknologia honekin.

Laburbiltzeko, bezeroaren garapenerako **JavaScript** erabiliko dugu, **React** liburutegien laguntzaz.

5.5 Datu-garbiketa

Hasteko, ETL prozesuaren atalean Informatica PowerCenter erreminta erabiltzea erabiltzea erabaki dugu. Erreminta honek ez dauka datu-garbiketaren ezaugarria, beraz soluzio bat bilatu behar dugu.

Lehenik eta behin, nola gauzatuko dugu datu-garbiketa? Gure garapena eginez edo merkatuan dagoen erreminta bat erabiliz? Egia esan, gure garapen propioa egitea oso zaila iruditzen zaigu. Alde batetik, aztertutako liburutegi horiekin ez dugulako eskarmenturik, eta beste alde batetik, oso liburutegi gutxi daudelako (bi horiek bakarrik aurkitu ditugu) eta oso informazio gutxi dagoelako horren inguruan. Beraz, ez dugu aukera ona ikusten gure kode pertsonalizatua garatzea.

Orduan, datu-garbiketarako erreminta bat erabiliko dugu. Oraingoan lehen aztertuko bi erreminten artean gaude, Winpure eta Data Ladder. Bien artean indartsuena eta ahalmentsuena **Data Ladder** iruditzen zaigu, nahiz eta ezin izan dugun jakin bere lizentziaren kostua.

5.6 Datu-iturriak

Lehenik eta behin, argi gelditu behar zaigu zein diren datu-iturriak (kasu honetan datu-iturri bakarra). Eta haien barruan zein den eremua, haren mota, eta batez ere, zer den eremu hori, hau da, zer informazio gordetzen duen:

- **xx_socios_eki**

Zutabea	Mota	Deskribapena
codigoParticipante	int(6)	Parte-hartzaile bakoitzaren identifikadorea
tipoDocumentoIdentidad	tinyint(1)	Nortasun agiri mota
documentoIdentidad	varchar(16)	Nortasun agiria
tarjetaDK	varchar(16)	Bazkide zenbakia
email	varchar(50)	Posta helbidea
estado	varchar(2)	EKIn daukan egoera
nombre	varchar(30)	izena
apellido1	varchar(30)	Lehen abizena
apellido2	varchar(30)	Bigarren abizena
fechaNacimiento	date	Jaiotze data
sexo	varchar(1)	Neska: M Mutil: V
calle	varchar(150)	Kalea
portal	int(4)	Ataria
portalBis	varchar(1)	Atariak beste zenbaki bat baldin badu haren

		barruan
escalera	varchar(2)	Eskailera
piso	varchar(5)	Solairua
mano	varchar(3)	Eskua
codigoPostal	varchar(5)	Posta kodea
municipio	varchar(30)	Hiria
provincia	varchar(30)	Probintzia
pais	varchar(40)	Herrialdea
telefono1	varchar(15)	Lehen telefono zenbakia
telefono2	varchar(15)	Ordezko telefono zenbakia
codigoResponsable1	int(6)	Adin gutxikoa bada, bere arduradunaren identifikazioa
codigoResponsable2	int(6)	Adin gutxikoa bada, bere bigarren arduradunaren identifikazioa
conocimientoEuskera	varchar(1)	Euskara badakien azaltzen du.
comunicacionesDK	tinyint(1)	Komunikazio lehentasunak
enviarAgenda	varchar(3)	Agenda nola jasotzen duen
usuarioAlta	varchar(9)	Zein eman dion altan
fechaAlta	datetime	Noiz eman den altan
usuarioModificacion	varchar(9)	Zein eguneratu dituen datuak
fechaModificacion	datetime	Noiz eguneratu diren datuak
usuarioBaja	varchar(9)	Zein eman du bajan bazkidea
FechaBaja	datetime	Noiz eman den bajan bazkidea

5.1 Taula: xx_socios_eki

- dk_acymailing_subscriber

Zutabea	Mota	Deskribapena
subid	int(10)	Identifikatzailea
email	varchar(200)	Posta helbidea (bazkidearekin erlazionatzeko erabiliko da)

5.2 Taula: dk_acymailing_subscriber

- dk_acymailing_listsub

Zutabea	Mota	Deskribapena
listid	smallint(11)	Buletinaren identifikatzailea, dk_acymailing_list taulan dagoena

subid	int(11)	Bazkidearen identifikatzailea, dk_acymailing_subscriber taulan dagoena
subdate	int(11)	Buletinean apuntatu den data
unsubdate	int(11)	Buletinetik bajaran eman den data

5.3 Taula: dk_acymailing_listsub

- dk_acymailing_list

Zutabea	Mota	Deskribapena
listid	smallint(10)	Buletinaren identifikadorea
name	varchar(250)	Buletinaren izena
description	text	Buletinaren deskripzioa
published	tinyint(11)	Buletina publikatua dagoen adierazten du
languajes	varchar(250)	Buletinaren hizkuntzak
category	varchar(250)	Buletinaren kategoria

5.4 Taula: dk_acymailing_list

- dk_content

Zutabea	Mota	Deskribapena
id	int(11)	Artikuluaren identifikatzailea
title	varchar(255)	Artikuluaren izenburua
alias	varchar(255)	Artikuluaren ezizena
introtxt	mediumtext	Artikuluaren sarrerako testua
fulltext	mediumtext	Artikuluaren testu osoa
created	datetime	Artikuluaren sortze data
modified	datetime	Artikuluaren eguneratze data
publish_up	datetime	Artikuluaren publikazio data
publish_down	datetime	Artikuluaren despublikazio data

5.5 Taula: dk_content

- dk_flexicontent_fields_item_relations

Zutabea	Mota	Deskribapena
field_id	int(11)	Eremuaren identifikadorea
item_id	int(11)	Artikuluaren identifikadorea (dk_content taulakoa)
value	mediumtext	Eremuaren balioa

5.6 Taula: dk_flexicontent_fields_item_relations

- **dk_flexicontent_field**

Zutabea	Mota	Deskribapena
id	int(11)	Eremuaren identifikadorea
field_tipe	varchar(50)	Eremu mota
label	varchar(255)	Eremuaren etiketa
description	varchar(255)	Eremuaren deskribapena

5.7 Taula: dk_flexicontent_field

- **xx_socios_sorteos**

Zutabea	Mota	Deskribapena
codUsr	int(7)	codigoParticipante, xx_socios_eki taulakoa
id_sorteos	int(11)	Zozketaren identifikatzailea (artikuluaren identifikatzailea)
fecha_hora_participacion	datetime	Zozketen parte hartu den data eta ordua

5.8 Taula: xx_socios_sorteos

Datu guzti hauek hartuta eta eraldatuta, lehen azaldutako datu-biltegia lortuko dugu. [5.1 irudiko](#) modeloa lortzeko eraldaketa batzuk gauzatu egin behar dira.

Hasteko, *xx_socios_eki* taulako datu guztiak datu biltegiaren *socio* taula izango da, aldaketarik gabe. Ondoren, datu biltegiko *boletines* taula, datu baseko *dk_acymailing_listsub* taulako eremuak hartuko ditugu. Baina lehendik, datu basean bi taula hauek ez daude lotuta, *dk_acymailing_listsub* taulan bazkide identifikatzaile bat dago, baina ez da erlazionatzen *xx_socios_eki* taularekin, baizik eta *dk_acymailing_subscriber* taularekin. Taula horretan bazkideen informazioa da, baina guri interesatzen zaiguna posta elektronikoaren helbidea da. Helbide hori *xx_socios_eki* taulako helbidearekin erlazionatzen ditugu, horrela buletina bazkidearekin lotuko dugu. Azkenik datu biltegiko *boletin* taula betetzeko, *dk_acymailing_list* taula erabiliko dugu.

Azkenik zozketak ditugu. Datu biltegiko *sorteos* taula datu baseko *xx_socios_sorteos* taula da. Azkeneko taula hau, berez, *xx_socios_eki* taularekin erlazionatzen da. Azkenik, *sorteo* taula gelditzen zaigu. Alde batetik, datuak *dk_content* taulatik hartuko ditugu, hemen artikuluen datuak daudelako, baina nola informazio gehiago nahi dugun, *dk_flexicontent_field* taulara joango gara. Hemen artikuluen eremuak daude, bere izenekin. Zozketek, eremu jakin batzuk dituzte, beraz horien identifikatzaileak eta izenak hartuko ditugu, datu biltegian izen hori izango delako taularen zutabearen izena. Ondoren identifikadore horrekin eta artikuluaren identifikadorearekin *dk_flexicontent_fields_item_relations* taulan eremu horren balioa hartuko dugu, eta datu biltegiara eraman.

Bukatzeko, datu-biltegian dauden datu guztien motak zehazteko falta dira. ETL prozesua gehiegi ez luzatzeko, datu basean daukaten datu-mota berdinak mantenduko ditugu, datu biltegiko DBKS uzten badu.

6

Soluzioaren alternatiba

Kapitulu honetan, aurreko atalean proposatutako soluzioaren alternatiba planteatzen dugu. Beti gerta daiteke softwarea bateragarria ez izatea, edo ez izatea aurrekontu nahiko nahi duguna egiteko.

6.1 ETL prozesua

Gure soluzioan Informatica PowerCenter aukeratu egin dugu prozesu hau gauzatzeko. Baina egia da erreminta oso garestia dela eta proiektu honek ez izatea hainbeste aurrekontu erreminta hau lortzeko.

Beraz, proposatzen dugun alternatiba honako hau da. **Pentaho Data Integration – Kettle** erabiltzea. Kode irekiko erreminta da, baina ala ere oso ahalmentsua da. Egia da, ez dela hain boteretsua Informatica baino, baina gauza asko egin dezakegu. Gainera kode irekikoa denez, funtzionalitate gehiago eta hobekuntzak egin ahal ditugu, **Java** lengoia erabilia.

Azkenik, erreminta honekin, harri batez bi kolpe egin dezakegu. Hau da, ETL erreminta honek, datu-garbiteta barnean dauka, horrela arazo hori baita ere konpontzen dugu, eta ez dugu beste erreminta bat behar edo garatzen denbora pasa behar.

6.2 Data Warehouse

Oraingo honetan, bi arrazoiengatik izan daiteke beste aukera baten beharra. Gure soluzioan Oracle DBKS aukeratu dugu eta gerta daiteke oso garestia izatea edo sistemarekin ez izatea bateragarria, nahiz eta azken hau probabilitate gutxi izatea.

Beraz, beste soluzio ekonomiakoago bat bilatze arren, **PostgreSQL** aurkitu dugu. Kode irekikoa da beraz, ez du kosturik izango. DBKS honek hazkundean dago, geroz eta erabiliagoa da. Beste sistemen ezaugarri antzekoak ditu eta oso indartsua da baita ere. Beraz aukera egokia iruditzen zaigu, Oracle ondoren. Gainera, Adimedian, datu base honekin lanean hasi nahi dute.

Aipatzeko, MySQL ez genuke inoiz hartuko datu biltegirako. Oso egokia da datu base transakzionalentzat, baina ez OLAP sistema batentzat, hau da, datu biltegi batentzat.

Azkenik, modeloaren diseinuan ez dugu aldaketarik planteatuko, hori delako soluziorik optimoena, ditugun datuekin.

6.3 Datuen sarbiderako APIa

Kasu honetan, datu biltegiaren menpe gaude. Beraz, azkenean PostgreSQL aukeratzen badugu datu biltegi bezala, APIa ere aldatu behar egingo dugu, Oracle REST Data Service, Oracle datu basearentzat delako. PostgreSQL ez du API bat eskaintzen datu atzipenerako, beraz gu geuk garatu egin beharko dugu.

Lehenik, erabiliko dugu protokoloa aukeratuko dugu. Aztertutako protokoloen artean **REST** protokoloa aukerarik hoberena iruditu zaigu. Lehenik eskaintzen duen ezaugarriengatik. Bigarrenik, DBKsek eskaintzen duten APIak REST direlako, beraz, honengatik baita ere aukera ona dirudi. Eta azkenik, Adimedian, eskarmentua dute REST APIak garatzen. Baina, ez hori bakarrik,

OData erabiltzea ideia ezin hobea iruditzen zaigu, API RESTak garatzeko jardunbide egokiak ematen ditu, eta datu base edo datu biltegi bat atzitzeko oso egokia da.

Ondoren informazioaren trukaketaren formatua aukeratu behar dugu. REST arkitekturan, edozein formatu aukeratu dezakegu, baina erabiliena **JSON** da, beraz hori aukeratuko dugu. Gainera, JSONentzat euskarri gehiago dago beste formatuentzat baino.

Eta azkenik garapen teknologia aukeratu behar dugu. Aztertutako teknologien artean, hiru dira atentzia deitu digutenak. Horiek, PHP, Java eta JavaScript (Node.js) dira. Hiruek ezaugarri bikainak eskaintzen dituzte APIak garatzeko, eta baita ere framework ezin hobeak dituzte. Horregatik, gauza bakar batengatik PHP aukeratu dugu. Adimedian, PHP-rekin esperientzia handia dute, eta beste biek ez. Horregatik, gure aukera **PHP** erabiltzea da **Laravel** framework-a erabilia.

6.4 Kudeaketa aplikazioa

Atal honetan ez dago gehiegi esaterik. Alde batetik bezero aldeko teknologia ia bakarra JavaScript da, eta gainera nabigatzaile guztietan bateragarria da. Beste alde batetik, zerbitzari aldea dago. Ez dugu arazorik ikusten Java erabiltzeari, baina gerta daiteke sistema batean bateragarria ez izatea, probabilitate oso gutxi izanik. Horregatik, **PHP** eta **JavaScript** (Node.js) proposatzen dugu, haiekin batera framework-ak erabiliz, **Laravel** eta **Express**, hurrenez hurren.

6.5 Datu-garbiketa

Atal honetan bi egoera egon daitezke. Alde batetik aukeratutako erreminta, Data Ladder, sistemarekin bateragarri ez izatea edo bere lizentzia kostu handiegia izatea, eta orduan, beste bat aukeratu behar izatea. Eta beste alde batetik, ETL prozesuan definitu dugu alternatiba aukeratu behar izatea.

Hasteko, Data Ladder ezin badugu aukeratu, proposatzen dugun beste aukera **Winpure** da. Winpure baita ere datu-garbiketa erreminta oso ahalmentsua da eta, gainera, eskuragarriagoa da, Data Ladder baino.

Ondoren, gerta daiteke azkenean **Pentaho** erabiltzea ETL prozesurako. Kasu horretan datu-garbiketa ETL prozesuaren barnean egingo zen eta erreminta hori izango zen arduraduna.

6.6 Datu-iturriak

Atal honetan ez dago alternatibarik bilatu behar, datu-iturria kasu guztietan berdin tratatu behar dugu, ez dago parametririk hau aldatu ahal dutenik.

7

Jarraipena eta kontrola

Kapitulu honetan, bigarren atalean egindako planifikazioaren jarraipena eta kontrola egingo dugu.

7.1 Burututako lana

7.1.1 Desbiderapenak

Proiektua planifikatu genuen garaian, 375 ordu estimatu egin genuen. Atal honetan estimatutako eta burututako orduen arteko alderaketa ikusi egingo dugu, eta bien artean egon den desbiderapena.

Kodea	Atazaren izenburua	Estimazioa	Errealia	Desbideraketa
1	Kudeaketa	32	32	0
1.1	Planifikazioa	10	15	-5
1.2	Jarraipena eta kontrola	22	17	5
2	Ikasketa	36	81	-45
3	Produktuaren analisia	234	182	52
3.1	Datu-iturriak	38	24	14
	Analisia	23	15	8
	Erabakiak hartu	10	5	5
	Alternatibak bilatu	5	4	1
3.2	Datuen kalitatea	31,5	0	31,5
	Analisia	24	0	24
	Erabakiak hartu	5	0	5
	Alternatibak bilatu	2,5	0	2,5
3.3	ETL prozesua	38	29	9
	Analisia	24	20	4
	Erabakiak hartu	9	7	2
	Alternatibak bilatu	5	2	3
3.4	Datu-biltegia	33	31,5	1,5
	Analisia	24	23	1
	Erabakiak hartu	5	7	-2
	Alternatibak bilatu	4	1,5	2,5
3.5	Datu-garriketa	30,5	32,5	-2
	Analisia	24	24	0
	Erabakiak hartu	4	7	-3
	Alternatibak bilatu	2,5	1,5	1
3.6	APIa	31,5	32,5	-1
	Analisia	24	24	0
	Erabakiak hartu	5	7	-2
	Alternatibak bilatu	2,5	1,5	1
3.7	Kudeaketa aplikazioa	31,5	32,5	-1
	Analisia	24	24	0
	Erabakiak hartu	5	7	-2
	Alternatibak bilatu	2,5	1,5	1
4	Memoria	73	75	-2
4.1	Mermoriaren garapena	58	60	-2
4.2	Aurkezpenaren prestaketa	15	15	0
		375	370	5

7.1 Taula: Ordu desbideraketak

Taulan ikusten den moduan, estimatutako ordu kopurua nahiko gerturatzen da ordu errealetara. Baina atazak ikusita, desbideraketa handiak egon dira. Orain banaka ataza guztiak azalduko ditugu, eta zergatik gertatu diren ordu desbideraketa horiek.

Hasteko kudeaketa dago. Guztira ez dago aldaketarik egon, baina, planifikaziorako ordu gehiago behar izan ditugu, proiekturen magnitudea gutxietsi dugulako hasiera batean, beraz proiektuaren planifikazioa denbora gehiago behar izan du. Aldiz, jarraipena eta kontrola oso ondo eraman dugu eta estimatutako denbora gutxiago behar izan dugu.

Ondoren ikasketa dator. Lehen esan bezala, proiektuaren magnitudea gutxietsi da, beraz hasiera batean denbora gutxiago beharko genuela pentsatu genuen. Horregatik atal honetan denbora desbideraketa oso handia egon da, zehazki 45 ordu gehiago behar izan dira.

Jarraitzeko, analisi osoa dator. Atal honetan baita ere desbideraketa handi bat egon da, batez ere atal honen azpiatal batean, *datu-kalitatea* azpiatalean. Atal hau ezin izan dugu egin, Donostia Kulturaren konpromiso faltagatik. Azpiatal hau gauzatzeko beharrezkoa genuen jakitea zer datu ustiatzea interesatzen zitzaien Donostia Kulturari, eta baita ere jakin zer den kalitatezko datu bat haientzat, horrela datu biltegi borobil bat diseinatzeko. Hau guztiagatik, azpiatal hau garatu gabe egon da, beraz, etorkizun batean, Donostia Kulturak berriro interesa badauka azpiatal hau garatu egin beharko zen haiekin batera.

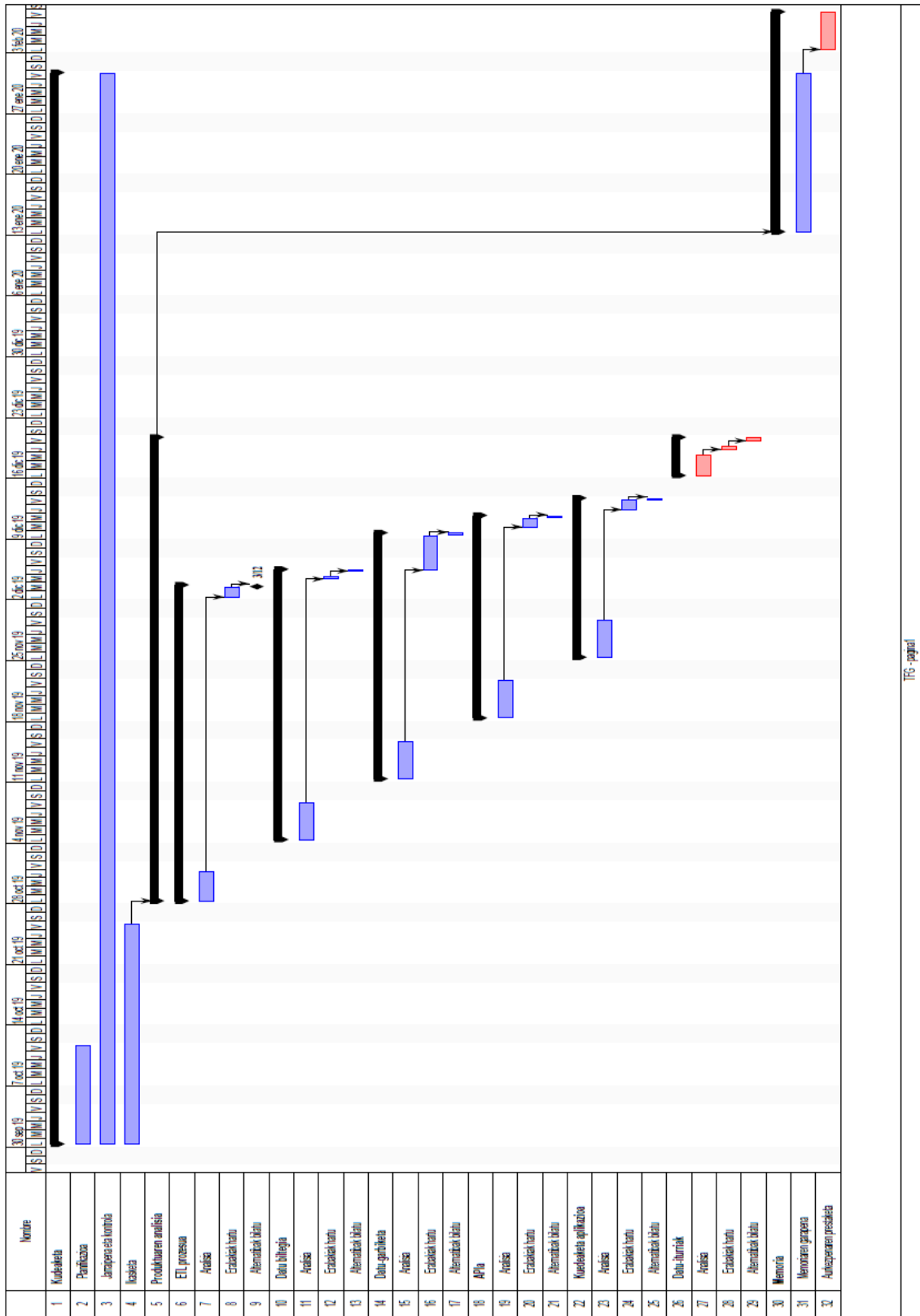
Baita ere, arrazoi berdinagatik, datu-iturrien atala bukaerarako utzi behar izan dugu, eta Adimediak dauzkan datu baseetan analisia oinarritu. Horregatik denbora gutxiago behar izan da. Gainera, azpiatal hau baita ere berriz egin beharko litzateke, Donostia Kulturak interesa jartzen duenean, datu-iturri guztiak aztertzeko. Beste azpiataletan ez dira aldaketa handirik egon.

Bukatzeko memoriaren atala dugu. Honetan ez dugu aldaketa handirik izan, bakarrik memoriaren garapenerako bi ordu gehiago behar izan ditugula, besterik ez.

Azkenean 370 orduko proiektua izan da, aldaketa oso nabariekina, baina planifikazio egokia egin denez, oso errez aurrera eraman ditugu arazo horien soluzioak.

7.1.2 Bukaerako Gantt diagrama

Aurreko ataleko desbideraketak egon direnez, Gantt diagrama baita ere aldatu egin da.



7.1 Irudia: Bukaerako Gantt diagrama

7.2 Arriskuak

Plangintzan arrisku hauek aurreikusi genituen, eta horrela eraman ditugu aurrera:

- **Denboraren aurreikuspen txarra:**

Arrisku hau oso probabilitate handia zeukan gertatzeko, eta aurreko atalean ikusi denez, nahiko nabaria izan da. Bukaeran ondo kudeatu dugula nabari da, ordu kopuru totalaren desbiderapena oso txiki izan delako. Baina esan beharra dago, arrisku hau beste arrisku batekin konpondu dela. Denbora estimazio txar bat egin dugu, proiektuaren magnitudea gutxietsi dugulako eta ikaskuntzarako denbora gehiago behar genuelako. Baina hau, ondoren azalduko dugun beste arrisku batekin konpondu dugu. Horregatik analisiaren azpiatal bat egin gabe geratu da, beraz, desbideraketak konpentsatu egin dira.

- **Bezeroaren konpromiso falta:**

Nahiz eta probabilitatea txikia izanik, arazo hau izan dugu oztopo bezala proiektuan zehar. Análisisan asko eragin du, alde batetik, azpiatal bat ezin izan delako osotasunean egin, datu-iturrien azterketa, eta beste azpiatal bat ez da egin, datu kalitatearen azterketa. Beraz, azkeneko azpiatal hau bezeroak berriz interesa duenean egin beharko da. Datu-iturrien azpiatala, zerbait konpontzea lortu dugu, Adimediak zeukan datu baseaz baliatuz. Baina hau baita ere, bezeroak berriz interesa duenean egin beharko da.

- **Teknologien ezagutza falta:**

Arazo hau gauzatzeko probabilitatea oso txikia zen, ez zeudelako teknologia gehiegirik erabiltzeko. Eta horrela izan da, ez dugu arazorik eduki, erabili ditugun teknologiak lehendik ezagunak zirelako, beraz ez dute suposatzen denbora galera bat.

- **Komunikazio falta:**

Bezeroaren konpromiso faltagatik, baita ere komunikazio falta egon da, eta puntu horretan egondako arazo berdinak gertatu dira. Beste alde batetik, enpresan egon den komunikazioa ona izan da, ez da egon inongo interferentziarik.

- **Informazio galerak:**

Arazoa hau ezta ere ez zaigu gertatu, beharrezko neurriak hartu ditugulako hasieratik hau ez gertatzeko. Neurri hauek planifikazioaren kapituluan azaldu genituen.

7.3 Aldaketak

Proiektuan zehar aldaketa batzuk egon dira, orain azalduko direnak:

- **Irismenaren aldaketa:** hasiera batean produktu osoa garatzea pentsatzen genuen, hau da, business intelligence produktu osoa. Baina ondoren, magnitudea kontuan hartuta, proiektu horren bideragarritasuna ez genuen argi ikusten, denboragatik. 300 orduko proiektu batean ezin zen hori guztia egin, beraz, produktu horren analisi sakona egitea erabaki genuen, Adimediak jarri zuen betebeharrak minimoa.
- **Datu-iturriak:** datu-iturrien analisia egiterako garaian, Donostia Kulturak oraindik ez zigun bere datuen informazioa eman. Beraz atal hau atzeratzea erabaki genuen, datuak lortzen genituen arte. Baina azkenean ez ziguten informazio hau eman. Orduan, geneukan informazioarekin lan egin behar genuen. Informazio hau, datu base bakar bat da, Adimediak kudeatutakoa, eta Donostia Kulturaren web orria kudeatzeko erabiltzen da.
- **Datuen kalitatea:** atal hau gauzatzeko, Donostia Kulturaren laguntza behar genuen baita ere. Baina nola ez zuten interesik ez zuten laguntza hori eman. Jakin behar genuen zer datu interesatzen zitzairen, horrela ondorengo analisiak egin ahal izateko. Informazio hau ez genuenez eskuratu, atal hau egin gabe gelditzea erabaki da, eta etorkizun batean, Donostia Kulturak interesa berreskuratzen badu, atal hau egin beharko da.

7.4 Kalitatea

Kalitatea ziurtatzeko [2.6.4.1](#) atalean azaldutako kalitate adierazleak kontuan hartuko ditugu:

- Plangintzaren jarraipena: aurreikusitako denborak ez dira zehazki bete, atal batzuen denbora estimazioa txarra izan delako eta beste atal batzuk ezin izan direlako bete. Beraz, arrazoi honengatik, proiektuaren kalitatea zerbait jaisten da.
- Dokumentuen tamaina: analisiaren azpialde bakoitzeko dokumentu bat garatu dugu eta tamaina handikoak, informazio asko biltzea lortu delako. Honek analisiaren kalitatea handitzen du.
- Dokumentuen argitasuna: dokumentuak modu argian antolatu dira, memoria honen antzerako egitura dute. Beraz, hau ere kalitatea handitzen du.
- Analisiaren koherentzia: informazioa biltzeko garaian beti gure arazoa kontuan hartu dugu, ez dugu beste informaziorik bilatu, ezta informazio gutxi bilatu ere. Orduan, analisiaren kalitatea handitzen da honekin.

8

Ondorioak

Kapitulu honetan, proiektu honen ondorioak azalduko dira.

8.1 Proiektuaren inguruko ondorioak

Proiektu hau Business Intelligence soluzio bat garatzeko aurretik egin behar den analisi sakonean datza. Hau da, Business Intelligence soluzioa lortzeko eman beharreko pausuak eta nola eman pausu horiek. Egia esan ez da nahi genuen soluzio borobilena, baina ez gure aldetik egondako arazo batengatik, baizik eta bezeroak hasiera batean zuen interesa galdu duelako. Beraz ez da gelditu esperotako analisi zehatza.

Baina honek ez du esan nahi ez gaudela konforme emaitzarekin. Analisi oso hona lortu dugula uste dugu, eta etorkizun batean bezeroak interesa berreskuratzen badu, analisi hau oso baliagarria izango da. Ez dira datu-iturri guztiak aztertu ezta ere nola nahi dugun eduki gure datu biltegia, hau da, zer izango den kalitatezko datu bat guretzat. Baina ETL prozesua, Data Warehouse, Datu-garbiketa, datu atzipenerako APIa eta kudeaketa aplikazioa sakonki aztertu ditugu. Zer teknologia dauden gaur egun, ahalmentsuenak naiz ekonomikoenak, eta baita ere zeintzuk diren jardunbide egokiak atal horiek aurrera eramateko modu egokienean.

Orduan, azken finean aztertu egin duguna da edozein Business Intelligence soluzio batek behar duen analisia, gure datuen ezaugarrietan sartu gabe. Gaur egungo edozein negozio adimenerako inplementazio bat gure analisia jarraitu dezake.

Baita ere, bezeroaren aldetik espezifikazio gehiago faltan bota ditugu. Adibidez, non kokatuko zen datu biltegia. Hodeian edo zerbitzari batean, adibidez. Edo zerbitzari batean kokatuta egongo bazen, zer ezaugarri izango zituen zerbitzari horrek. Informazio hau oso ondo etorriko zen erabakitzeke ETL edo ELT prozesua gauzatzea, [3.3 atalean](#) azaldu dugun bezala. ELT prozesu bat gauzatzeko zerbitzari horrek prozesamendu ahalmen handia izan beharko zuen, baina ETL prozesurako bitarteko sistema bat beharrezkoa dugu eraldaketa gauzatzeko.

Beste alde batetik, esan beharra dago, dokumentu honetan azaldutako teknologiak ez direla bakarrak. Askoz gehiago daude, baina dokumentu honetan azaldu dugun moduan, aukeratutako hauek izan dira egokiak gure arazorako. Beste teknologia batzuk erabili daitezke, eta egokiak izan daitezke, baina dokumentu honetan azaldutakoak seguru egokiak direla.

Laburbiltzeko, egindako proiektua oso ona iruditu zaigu, nahiz eta ez aztertzea nahi genuen guztia. Emaitza ona da, eta honen gainean lanean hasi gaitzke, Donostia Kulturarentzat Business Intelligence soluzio bat inplementatzen, beti bere konpromisoa handiagoa bada, bestela proiektua porrota aseguratua izango du hasieratik.

8.2 Ondorio pertsonalak

Ondorio pertsonal asko atera egin ditut proiektu honetatik. Hasteko, erronka handi bat izan da, magnitude honetako proiekturik inoiz ez dudalako aurrera eraman. Gainera proiektu erreal bat izan da, ez unibertsitateko proiektu akademiko bat. Modu honetan, mundu errealean garatzen diren proiektuak ikusi ditut, eta zer egin behar den aurrera eramateko.

Harrigarria iruditu zait 300 ordu baino gehiagoko proiektu batek bakarrik beste proiektu handi baten analisia dela, eta ez dugula azken produktu bat garatu. Honekin konturatu naiz, produktu erreal bat honelako analisi bat behar duela gutxienez, ondoren produktuak arrakasta izan dezan eta kalitate on bat izan dezan.

Baita ere, Adimedia bezalako enpresa batek izan dezakeen arazoak ikusi ditut. Adimediak produktu informatikoak garatzen ditu beste enpresa edo erakunde batzuentzat, beraz garatzen dituen produktuak ez dira beraientzat. Horrela beste batzuen ezaugarriekin jokatu behar du eta haiekin etengabeko komunikazioa mantentzea. Baina gauzak ez dira beti hain politak, guri gertatu zaigun bezala, Donostia Kulturak azkenean interesa galdu duela proiektu honetan. Beraz, arazo hauei nola aurre egin ikasi dut, eta zer arrisku dauden zure bezeroan konfiantza edukitzeaz.

Beste alde batetik, graduan zehar ikasitakoa aplikatzen hasi naiz. Alde batik datu baseen inguruko ezagutzak aplikatzeko aukera izan ditut, eta gainera ezagutza horiek handitzeko, datu biltegien inguruan. Baita ere, Software Ingeniaritzaren beste irakasgaietan ikasitakoa erabili dut, adibidez APIaren edo kudeaketa aplikazioaren teknologiak aztertzeko. Eta noski, proiektua aurrera eramateko kudeaketa on bat egitea beharrezkoa izan da, graduan zehar baita ere ikasitakoa. Honekin plangintza osoa eta jarraipena eta kontrola burutu egin dira, eta proiektuaren arrakastan parte hartu dute.

Baina ez hori bakarrik, proiektu hau hasi aurretik, ez nuen Business Intelligence inguruko ezagutzarik, ezta Business Intelligence terminoa entzun ere. Proiektu honi esker hau aldatu da, eta mundu honen inguruko ezagutzak asko handitu ditut. Eta ez hori bakarrik, gainera arazo erreal batean ezartzeko aukera izan dut, eta horrela gehiago eta azkarrago ikasteko aukera izan dut.

8.3 Etorkizunerako gomendioak

Informatikaren mundua etengabeko eboluzio batean dago, eta Business Intelligence ez da gutxiago. Garatzaileak hobekuntzak eta teknologia berriak garatzen ari dira. Gainera enpresa handiak etekin asko ateratzen dute honekin, beraz diru asko inbertitzen dute.

Honengatik, etengabe ari dira teknologia berriak eta teknika berriak ateratzen, horregatik, gure analisi hau ezin da betirako erabili. Etorkizun batean teknologia hobetoagoak egongo dira eta teknika berriak eta hobetoagoak egongo dira. Beraz, Donostia Kulturak interesa berrekuratzen badu etorkizun labur batean analisi hau oraindik erabilgarri egon daiteke, baina etorkizun luzeago batean, analisia berriz egin beharko litzateke, garai horretan egongo diren teknologia berriak kontuan hartuz.

Baita ere, dokumentuan zehar aipatzen joan garen moduan, atal batzuk betetzeko falta dira, Donostia Kulturarekin batera. Horregatik, etorkizun batean dokumentu hau baliagarria izateko, atal horiek bete behar dira lehendik. Gainera, atal horiek gure datuak kontuan hartzen dute, beraz, nahiz eta eginak egon, berriz egingo beharko lirateke, datuak etengabe aldatzen direlako, datu berriak sartuz edo daudenak eguneratuz.

Bibliografía

Donostia Kultura

https://eu.wikipedia.org/wiki/Donostia_Kultura

<https://www.donostiakultura.eus/index.php?>

[option=com_flexicontent&view=items&cid=4&id=14698&Itemid=21&lang=es](https://www.donostiakultura.eus/index.php?option=com_flexicontent&view=items&cid=4&id=14698&Itemid=21&lang=es)

Adimedia Comunicación Interactiva

<https://www.adimedia.net/agencia/hola/>

Business Intelligence

<https://www.guru99.com/business-intelligence-definition-example.html>

https://www.sinnexus.com/business_intelligence/

Data Warehouse

<https://www.powerdata.es/data-warehouse>

<https://www.guru99.com/data-warehouse-architecture.html>

<https://www.businessintelligence.info/serie-dwh/como-construir-datawarehouse.html>

<https://www.guru99.com/dimensional-model-data-warehouse.html>

ETL – ELT

<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/qu-son-los-procesos-etl>

<https://blog.bi-geek.com/4-tipos-herramientas-etl/>

<https://es.talend.com/resources/elt-vs-etl/>

<https://www.guru99.com/etl-vs-elt.html>

<https://blog.panoply.io/etl-vs-elt-the-difference-is-in-the-how>

API

<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

<https://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>

https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

Datu-garbiketa

<https://blog.es.logicalis.com/analytics/data-cleansing-y-sus-fases-contra-los-problemas-de-calidad-de-datos>

<https://empresas.blogthinkbig.com/tus-datos-mas-limpios-casi-sin-frotar/>

<https://www.geotab.com/blog/data-cleaning/>

Hadoop

<https://hadoop.apache.org/>

https://en.wikipedia.org/wiki/Apache_Hadoop

<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/328879/c-mo-se-relacionan-big-data-y-hadoop>

SOAP

<https://www.oscarblancarteblog.com/2017/03/06/soap-vs-rest-2/>

<https://www.chakray.com/es/que-diferencias-hay-entre-rest-y-soap/>

<https://www.guru99.com/comparison-between-web-services.html>

REST

<https://www.oscarblancarteblog.com/2017/03/06/soap-vs-rest-2/>

<https://www.chakray.com/es/que-diferencias-hay-entre-rest-y-soap/>

<https://www.guru99.com/comparison-between-web-services.html>

<https://es.stackoverflow.com/questions/2512/cual-es-la-diferencia-entre-rest-y-restful>

ORM

https://en.wikipedia.org/wiki/Object-relational_mapping

<https://www.campusmvp.es/recursos/post/que-es-un-orm.aspx>

A Eranskina: bilera aktak

1. Bilera akta

Data: 2019/09/12

Ordua: 10:30

Bertaratuak: Imanol Usandizaga (Zuzendaria), Jokin Indias (Ikaslea)

Lekua: Informatika Fakultatea, Donostia

Gai-zerrenda:

- Proiektuarentzat zuzendaria bilatu

Hitz egindakoa eta hartutako erabakiak:

- Imanol zuzendaria izateko prest dago.
- Proiektuaren nondik norakoa azaldu.
- Zuzendariak proiektuaren eta memoriaren inguruko gomendioak eman.

2. Bilera akta

Data: 2019/10/11

Ordua: 11:00

Bertaratuak: Alexis Mindegia (Adimediako zuzendaria), Jokin Indias (Ikaslea)

Lekua: Adimedia, Irun

Gai-zerrenda:

- Proiektuaren irismenean aldaketak

Hitz egindakoa eta hartutako erabakiak:

- Denboraz larri gaudela konturatu gara, eta seguruenik proiektu osoa ezin izango dugu garatu.
- Betebehar minimoak finkatzen ditugu, hau da, proiektuaren analisisa.

3. Bilera akta

Data: 2019/10/25

Ordua: 11:30

Bertaratuak: Juanjo Pascual (Adimediako tutorea), Jokin Indias (Ikaslea)

Lekua: Adimedia, Irun

Gai-zerrenda:

- Proiektuaren betebeharrak minimoak egiteko jarraibideak

Hitz egindakoa eta hartutako erabakiak:

- Irismena guztiz finkatu.
- Irismen hori aurrera eramateko jarraitu beharrezko pausuak.
- Juanjo Donostia Kulturarekin kontaktuan jarriko da, datu-iturriak lortzeko.

4. Bilera akta

Data: 2019/12/16

Ordua: 10:00

Bertaratuak: Juanjo Pascual (Adimediako tutorea) Jokin Indias (Ikaslea)

Lekua: Adimedia, Irun

Gai-zerrenda:

- Datu-iturrien arazoa

Hitz egindakoa eta hartutako erabakiak:

- Donostia Kulturak ez ditu datu-iturrien informazioa eman.
- Arazo hau aurrera eramateko, Juanjok erabaki du Donostia Kulturaren datu basea erabiltzea, Adimediak kudeatutakoa.
- Datu base horren nondik norakoa azaldu du Juanjok.

5. Bilera akta

Data: 2020/01/27

Ordua: 11:30

Bertaratuak: Imanol Usandizaga (Zuzendaria), Jokin Indias (Ikaslea)

Lekua: Informatika Fakultatea, Donostia

Gai-zerrenda:

- Memoriaren egoera

Hitz egindakoa eta hartutako erabakiak:

- Memoria ondo bideratuta dagoela adierazi du Imanolek.
- Jarraitzeko gomendio batzuk eman ditu.
- Memoria entregatzearen eta defentsaren zalantzak argitu ditugu.