

emeri la zabal zazu



Universidad del País Vasco  
Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA  
ESCUELA DE INGENIERÍA DE BILBAO

INDUSTRIA INGENIARITZA TEKNIKOKO ATALA

SECCIÓN INGENIERÍA TÉCNICA INDUSTRIAL

--

Sinadura DATA	Sinadura DATA
------------------	------------------



## LABURPENA

InLog Consulting enpresak garatutako WMS (*WareHouse Management System*) aplikazioaren zabaltzea eta aplikazio horri funtzionalitate gehiago gehitzea izan da proiektu honen helburu nagusia. Aplikazio honek logistikako operadore batek dituen funtzionalitate batzuk ditu.

WMS aplikazioa web nabigatzaileetan exekutatzeko da. Enpresak fakturatzeko atal honek dituen funtzionalitate batzuk beste ingurune batean baditu (ORACLE FORMS –en) bezero batentzat inplementatuta. Aukera hau WMS aplikaziora hedatu nahi zuten haiek dituzten beste bezeroentzat baita ere eskuragarri egoteko.

Merkatuan dagoeneko existitzen dira logistika aplikazioak, baina aplikazio honek erabiltzaileei aukera gehiago emango dizkie WMS aplikazioak duen atal berri honekin, Fakturazioarekin zerikusia duten prozesuak automatizatu bai dira.



## EDUKIEN AURKIBIDEA

LABURPENA .....	1
IRUDIEN AURKIBIDEA .....	8
TAULEN AURKIBIDEA.....	14
SARRERA.....	16
Aurretik jakin beharrekoa .....	16
Zer da logistikako operadore bat? .....	16
Zer da Fakturatze-Modulua?.....	18
Zergatik aukeratu dut proiektu hau? .....	20
HASIERAKO PLANTEAMENDUA .....	22
Proiektuaren helburuak .....	22
Proiektuaren irismena.....	23
PROIEKTUAREN FASEAK .....	23
LDE diagraman sakonduz .....	26
Denbora plangintza .....	29
Software baliabideak.....	31
Hardware baliabideak .....	32
Arkitektura .....	32
Bezero/zerbitzari arkitektura .....	33
Barne egitura.....	34
Ebaluazio ekonomikoa .....	39
Eskulana.....	39
Hardware gastuak .....	40
Software gastuak.....	40

Gastuak guztira.....	41
Arriskuen kudeaketa .....	42
AURREKARIEN AZTERKETA .....	45
Merkatuan dagoen softwarea.....	45
Garraioen kudeaketa.....	45
Biltegien kudeaketa.....	46
BETEKIZUNEN BILKETA .....	47
Aktoreen hierarkia .....	47
Erabilpen kasuak .....	47
Kautotzea .....	48
Kautotutako erabiltzailearen aukerak.....	48
ANALISIA & DISEINUA .....	54
Garapenerako aukeratutako lengoaiak.....	54
Domeinuaren ereduak .....	54
Entitateak .....	56
Klase diagrama .....	57
Klase garrantzitsuak .....	59
Datu-basearen entitate/erlazio diagrama.....	63
Prozesu esanguratsu baten sekuentzia diagrama: Fakturen inprimatzea .....	67
GARAPENA.....	69
Fakturatze-modulua sakonean.....	69
Funtzionalitateetan sakonduz .....	71
Proiektuaren antolakuntza.....	74
Fakturatze-modulua: funtzionalitateen azalpena.....	82

JSP.....	83
JavaScript.....	88
XML.....	88
.sql PACKAGE.....	90
Funtzionalitateen azalpena.....	91
Garapenarekin arazoak.....	93
Saioan informazioa mantendu.....	93
Arazoaren konponbidea.....	99
JasperReports.....	102
EGIAZTAPENA ETA EBALUAZIOA.....	107
Proben txantiloia.....	107
ONDORIOAK ETA ETORKIZUNEKO LANA.....	110
Ondorio orokorrak.....	110
Ondorio pertsonalak.....	114
Etorkizunerako lana.....	114
Bibliografia.....	116
Liburuak.....	116
Kontsultatutako Webguneak.....	116
A ERANSKINA.....	118
Filtroen arabera bilatu sekuentzia diagrama.....	118
Erregistroa gehitu sekuentzia diagrama.....	119
B ERANSKINA.....	121
C ERANSKINA.....	125
Erabiltzaile gida.....	125

Sarbidea lortu.....	125
Kautotzea .....	125
Atalaren hautaketa.....	126
Menu nagusia.....	126
Oinarrizko funtzionalitateak.....	127
Funtzionalitate bereziak.....	131
E ERANSKINA .....	135
Lan ingurunea prestatu .....	135
JDK instalatu (1.6.0_45).....	135
JBOSS instalatu .....	136
Git errepositorioen kudeaketa.....	136
Eclipse IDE –aren konfigurazioa .....	136





## IRUDIEN AURKIBIDEA

Irudia 1 - Banaketa prozesua.....	16
Irudia 2 - Kontrol modulua .....	17
Irudia 3 - Bezeroen modulua.....	17
Irudia 4 - Garraio modulua .....	17
Irudia 5 - Fakturatze modulua.....	17
Irudia 6- LDE diagrama .....	25
Irudia 7 - Gantt diagrama .....	30
Irudia 8 - Arkitektura .....	33
Irudia 9- MVC osagaiak.....	35
Irudia 10 - Aktore Hierarkia.....	47
Irudia 11 - Kautotzea .....	48
Irudia 12 - Fakturatze-modulua erabilpen kasuak .....	48
Irudia 13 - Erabilpen & Azpi erabilpen kasuak .....	50
Irudia 14 - Domeinuaren eredua.....	55
Irudia 15 - Menu.....	57
Irudia 16 - Pantaila prototipoa.....	57
Irudia 17 – Klase Diagrama.....	58
Irudia 18 - Action klaseko geruza nagusiak .....	61
Irudia 19 - PLSQL Developer Diagrama .....	64
Irudia 20- PL SQL Developer diagrama 2.....	65
Irudia 21 - Fakturen inprimatzea (sekuentzia diagrama) .....	67
Irudia 22 - LOV popup .....	70
Irudia 23 - Proiektuaren antolaketa .....	74

Irudia 24 - src osagaiak.....	75
Irudia 25 - Action & Form.....	76
Irudia 26 – Managerrak.....	76
Irudia 27 - DAO & SQL query.....	77
Irudia 28 - Textos Properties.....	77
Irudia 29 - Literalak gazteleraz.....	77
Irudia 30 - Literalak ingelesez.....	78
Irudia 31 - WebContent osagaiak.....	78
Irudia 32- JavaScript fitxategiak.....	79
Irudia 33 - Jasper fitxategiak.....	80
Irudia 34 - Konfigurazio fitxategiak.....	81
Irudia 35 - JSP fitxategiak.....	81
Irudia 36 - Menu nagusia.....	82
Irudia 37 - Azpi atalen direktorioak.....	82
Irudia 38 - JSP atalak.....	83
Irudia 39 - Main: Tag - en definizioa.....	84
Irudia 40 - Atalak eta geruzak.....	84
Irudia 41 - Filter: Iragazpen atala pantaila.....	84
Irudia 42 - Eremu baten definizioa.....	85
Irudia 43 - Botoi baten definizioa.....	85
Irudia 44 - Detail: Emaitzen bistaratzea.....	85
Irudia 45 - Detail: Ekintzaren definizioa.....	86
Irudia 46 - Detail: Ezabatze aukeraren definizioa.....	86
Irudia 47 - Detail: Eguneraketa aukeraren definizioa.....	86

Irudia 48 - Detail: Eliminar botoia .....	86
Irudia 49- Edit: Erregistro baten sorrera .....	87
Irudia 50 - Edit: Eremuak.....	88
Irudia 51 - JavaScript: Funtzioen definizioa.....	88
Irudia 52 - XML: resultMap sortzen.....	89
Irudia 53 - XML: Query definizioa.....	89
Irudia 54 - XML: Parameter Map definizioa .....	90
Irudia 55 - XML: Prozedimenduari deia.....	90
Irudia 56 - Package: Prozedimenduaren definizioa.....	90
Irudia 57 - Fakturazio-Tarifa goiburua.....	94
Irudia 58 - Fakturazio-tarifa lerroa .....	94
Irudia 59 – Directo kalkulatzte-formula pertsonalizatzeko popup-a .....	96
Irudia 60 – Escalado kalkulatzte-formula pertsonalizatzeko popup-a .....	97
Irudia 61 - Atributuen deklarazioa .....	99
Irudia 62 - Ekintza eta DDBB esleipena HashMap ean.....	101
Irudia 63 - Faktura motaren hautaketa.....	102
Irudia 64 - Faktura laburtua 1 .....	103
Irudia 65 - Faktura laburtua 2 .....	104
Irudia 66 - Faktura zehaztua.....	105
Irudia 67 - Gantt denbora erreala .....	113
Irudia 68 - Filtroen arabera bilatu (sekuentzia diagrama) .....	118
Irudia 69 - Erregistro berri bat sortu (sekuentzia diagrama) .....	119
Irudia 70 - URL sarbidea .....	125
Irudia 71 – Kautotzea .....	125

Irudia 72 - Aplikazioko atalak .....	126
Irudia 73 - Fakturatze-modulua: Menu nagusia.....	126
Irudia 74 - Conceptos de Facturación menu bidea .....	126
Irudia 75 - Bilaketa iragazpena.....	127
Irudia 76 - Bilatzea: emaitzen bistaratzea.....	128
Irudia 77 - Checkbox aukeratua .....	128
Irudia 78 - Erregistro sortzea.....	129
Irudia 79 - Erregistroa editatzea.....	130
Irudia 80 - Ezabatze mezua .....	131
Irudia 81 - Dena ondo joan denaren mezua.....	131
Irudia 82 - Faktura formatua (mezua).....	132
Irudia 83 - Faktura nabigatzailean.....	132
Irudia 84 - LOV botoiak.....	133
Irudia 85 - LOV popup .....	133
Irudia 86 - JDK bertsioa ORACLE web orrialdean .....	135
Irudia 87 - Eclipse instalatzailea .....	136
Irudia 88 - JBoss Tools instalazioa MarketPlace-tik.....	137
Irudia 89 - Jaspersoft instalazioa MarketPlace-tik .....	137
Irudia 90 - Proiektua inportatu.....	138
Irudia 91 - Existing Projects aukera .....	139
Irudia 92 - Zerbitzaria instalatzen.....	139
Irudia 93 - Create new runtime.....	140
Irudia 94 - JBoss eta JDK adierazten.....	140
Irudia 95- Runtime deskarga .....	141

Irudia 96 - Runtime deskarga .....	141
Irudia 97 - Proiektuak zerbitzarian gehitu.....	142



## TAULEN AURKIBIDEA

Taula 1 - Tarifen kudeaketa.....	18
Taula 2 - Fakturen kudeaketa.....	19
Taula 3 - Prozesuen zerrenda.....	22
Taula 4 - Denbora estimazioa.....	28
Taula 5 - Prozesuen denbora estimazioa .....	28
Taula 6 - Atalen banakapena.....	29
Taula 7 - Eclipse plugin -ak .....	31
Taula 8 - Hardware baliabideak: PC .....	32
Taula 9 - Hardware baliabideak: Server .....	32
Taula 10 - Erabilitako JSTL etiketak .....	38
Taula 11 - Gastuak laburtuta .....	41
Taula 12 - Arriskuen laburpena .....	43
Taula 13 - Prozesuen funtzionalitatea.....	73
Taula 14 - Pantaila baten osagaiak.....	83
Taula 15 - Fakturazio bidea (prozesuak) .....	91
Taula 16 - Ekintza esleipena 0-tik.....	100
Taula 17 - Directo mota aldatzen ekintza esleipenak .....	100
Taula 18 - Escalado mota aldatzen ekintza esleipenak .....	101
Taula 19 - Proben txantiloia .....	108
Taula 20 - Estimaturako denbora vs. denbora erreala.....	111
Taula 21 - Prozesuen estimaturako denbora vs. denbora erreala.....	111
Taula 22- Proba gidioa (gaztelerako bertsioa) .....	123

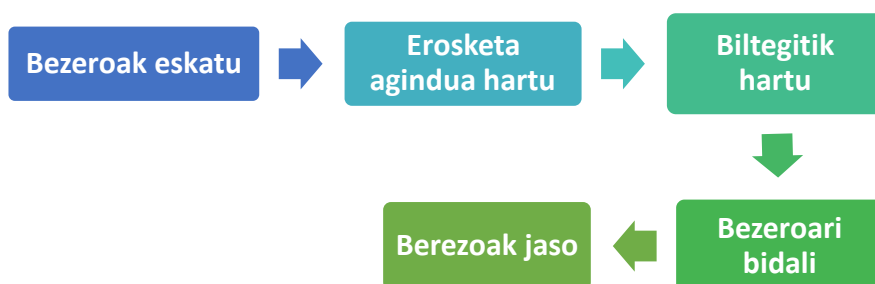




## SARRERA

### Aurretik jakin beharrekoa

Gaur egun gauza askotarako erabiltzen ditugu banaketa-zerbitzuak eskaintzen dituzten enpresak, erakunde kopuru zenbatezinek erabiltzen dituzte baita ere. Baina, zertarako? Eguneroko gauza batzuentzako erabiltzen direla esan genezake, adibidez, Amazon bidez erositako SmartWatch berria norbaitek ekarri beharko digu etxera edo Amazonek berak, beharko du haren biltegietara norbaitek bidaltzea berak ondoren salduko duena. Banaketa-zerbitzua, beraz, kontsumitzaileak edo erosleak produktu zehatz bat beste edozeini erosi (edo eskatu) dion momentutik erosleak edo kontsumitzaileak jaso arte jarraituko duen banaketa-prozesua izango da.



*Irudia 1 - Banaketa prozesua*

Egun, banaketa-zerbitzuetako enpresak jarraitzen dituzten barne prozesuak banatu behar duten materiala edo produktuak haien helburuetara heltzeko guztiz automatizatuta edo ia guztiz automatizatuta daude. Normalean, hau guztia kudeatzeko logistikako operadore bat erabiltzen dute enpresek.

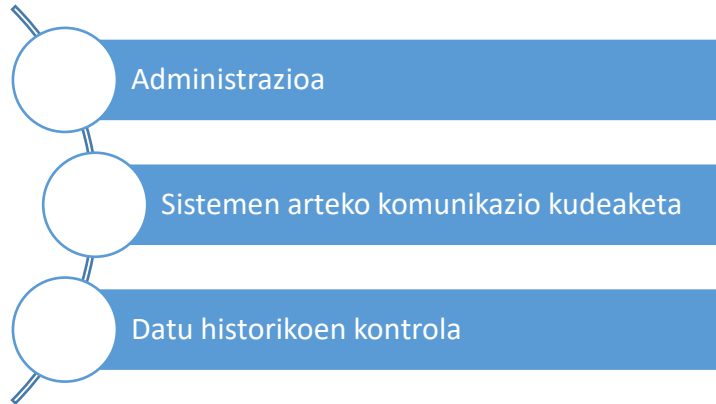
### Zer da logistikako operadore bat?

Bezero batek dituen betekizunak kontuan izanda, haren hornikuntza katearen diseinuaz arduratuko den erakundea da, adibidez, hornikuntzaren, garraioaren, biltegitratzearen eta distribuzioaren diseinuaz.

Logistikako operadore batek kudeatzen dituen funtsezko eragiketak burutzen ditu. Hauek dira **operadore logistiko bat osatzen duten oinarriko moduluak**:

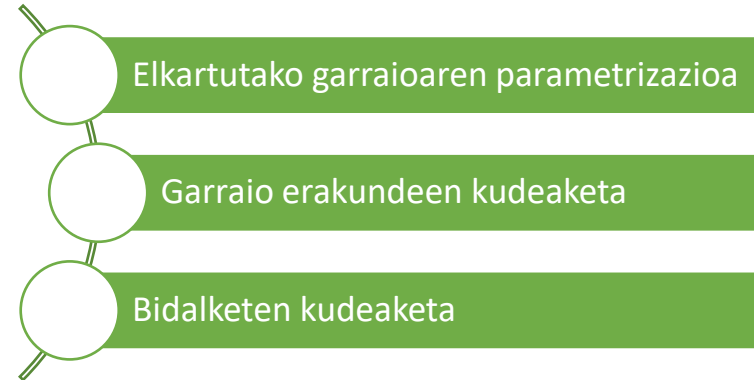
1. Kontrol-modulua
2. Garraio-modulua
3. Bezeroen-modulua
4. Fakturatze-modulua

### 1.KONTROL-MODULUA



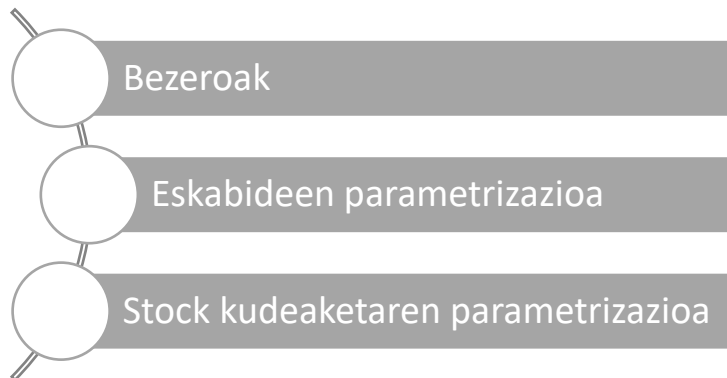
*Irudia 2 - Kontrol modulua*

### 2.GARRAIO-MODULUA



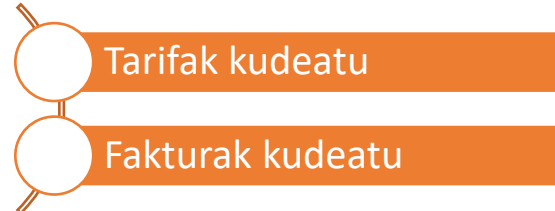
*Irudia 4 - Garraio modulua*

### 3.BEZEROEN-MODULUA



*Irudia 3 - Bezeroen modulua*

### 4.FAKTURATZE-MODULUA



*Irudia 5 - Fakturatzte modulua*

Proiektu honek jadanik inplementatuta dagoen logistikako operadore bati atal berri bat gehituko dio, Fakturatze-modulua hain zuzen. Beraz, In-LOG WMS (*Warehouse Management System*) aplikazioari zati berri hau gehituko zaio. Aplikazioak duen helburu nagusia biltegien kudeaketa aurrera eramatea eta kalitate handiko zerbitzu bat ematea da. In-LOG WMS logistikako operadore batean oinarritzen den web aplikazio bat da. Biltegiekiko eragiketak haririk gabeko gailuekin aurrera eramaten dira denbora errealean.

## Zer da Fakturatze-Modulua?

Fakturatze-modulu berri honekin hainbat gauza automatizatuko dira, adibidez, logistikako operadore batek emandako zerbitzu desberdinak kudeatzeko tarifa-zerrenda eskemaren eta fakturazioaren parametrizazio bat erabili genezake.

Beherago azalduko diren bi atalen kudeaketa erraztuko luke. Logistikako operadoreak besteei emandako zerbitzuak kudeatzeko erabiliko da, besteak beste, tarifak eta fakturak

Modulu honek dituen funtzionaltasunak hauek izango lirateke:

1. Tarifen kudeaketa: garraio enpresei egiten zaien likidazioa eta bezeroei egiten zaien fakturazioa.

Taula 1 - Tarifen kudeaketa

MANTENU OROKORRAK
<ul style="list-style-type: none"> <li>• Zergak</li> <li>• Banketxeak</li> <li>• Monetak</li> <li>• Bihurtzeak</li> <li>• Negozio-lineak</li> </ul>
FAKTURATZE-KONTZEPTUEN MANTENUA
FAKTURATZE TARIFEN MANTENUA
<ul style="list-style-type: none"> <li>• Garraioa</li> <li>• Manipulazioa</li> <li>• Biltegiratzea</li> <li>• Kudeaketa</li> </ul>
FAKTURATZE BEZEROAK
<ul style="list-style-type: none"> <li>• Datu Orokorren mantenua</li> </ul>

2. Fakturen kudeaketa: ziklo logistikoa zehar operadoreak emandako zerbitzuengatik fakturak sortu, kontzeptu egozgarrietaz baliatuz.

Taula 2 - Fakturen kudeaketa

**FAKTURAZIORA IGAROBIDEA**

- Kontzeptuak
- Garraiora lotutako tarifak
- Manipulazioa
- Biltegiatzea

**FAKTUREN SORRERA**

**FAKTUREN IXTEA**

**INPRIMATZEA ETA KONTSULTAK**

- Fakturen jaulkipena
- Fakturen kontsulta orokorra

## Zergatik aukeratu dut proiektu hau?

Idea praktikak egiten ditudan enpresan sortu zen, enpresak berak atal berri hau gehitu nahi dio aplikazioari. Dagoeneko antzeko fakturatze-modulu batekin lan egiten dute, baina hori bezero batentzat eskusiboki dago, gainera inplementatuta dagoen ingurunean pixka bat atzeratuta geratu da, ORACLE FORMS –ekin bait dago inplementaturik.

In-LOG WMS aplikazioak oraindik ez ditu logistikako operadore batek dituen atal guztiak, helburu nagusia hau izango litzateke: aplikazioa logistikako operadore bat bihurtzeko beste atal bat gehitzea.

Merkatuan badaude beste logistikako operadoreak diren beste aplikazio batzuk baina aplikazio honekin eta modulu hau gehitu ondoren aplikazioa orain arte erabili duten erabiltzaileei aukera gehiago emango dizkie, logistika arloarekin fakturazioarekin zerikusia duten prozesuak “automatizatzea” lortuko da.

Azkenik, oso interesgarria iruditu zitzaidan aplikazioa bere Cloud –ean dagoelako, beraz, ez dago bezero batek duen gailu guztietan instalatu beharrik, ezta eguneratzeak egin ondoren berrinstalatu beharrik ere.

Gaur egun enpresa batek haien barruko kudeaketarako Cloud aplikazio bat edukitzea da eskari nagusia. Aplikazio mota hauek alde positibo ugari dituzte, hemen esanguratsuenak aipatuko dira:

- Aplikazioa martxan jartzeko nabigatzaile bat besterik ez da behar.
- Exekutatzen ari den gailuan ez du memoria askorik behar (Nabigatzaileak berak martxan ipintzen dituen errekurtsioak kontuan izan barik).
- Gailuan ez da ezer ez instalatu behar.
- URL batekin baduzu aplikaziorako sarbidea.
- Aplikazioa exekutatu daitekeen gailu kopurua oso handia izan daiteke.
- Eguneraketa prozesuak oso erraz egin daitezke (zerbitzarian aplikazioa aldatuz gero nabigatzaileek bertsio berria erabiliko dute).



## HASIERAKO PLANTEAMENDUA

### Proiektuaren helburuak

Proiektu honen oinarriko helburua dagoeneko implementatuta eta martxan dagoen aplikazio bati funtzionaltasun gehiago emango dion atal berri bat gehitzea da. Funtzionaltasun guzti hauek logistikako operadore baten fakturazioaren arloarekin zerikusia izango dute, adibidez, lehen pertsona batek faktore guztiak kontutan izan behar zituen faktura bat emititzeko edo faktura guztiak biltegiatuta zeuden lekuan nahi zuen faktura hori bilatu behar zuen ondoren inprimatzeko. Atal berri honek hori guztia (edo ia guztia) automatizatuko du. Zehazki honako hauek izango dira aplikazioa automatizatzekeko implementatuko diren prozesuak:

PROZESUA	FUNTZIONALTASUNA
<b>Fakturazio kontzeptuen kudeaketa</b>	Kontzeptuen sorrera, ezabaketa edo eguneratzea
<b>Manipulazioko fakturazio biderako jaurtiketa</b>	Erabiltzaileak aukeratutako eskarien fakturazio biderako jaurtiketa
<b>Biltegiatzeko fakturazio biderako jaurtiketa</b>	Jabearen eta egunaren arabera fakturazio biderako jaurtiketa
<b>Eskuzko fakturazio kontzeptuen kudeaketa</b>	Eskuzko kontzeptuen sorrera, ezabaketa edo eguneratzea
<b>Fakturazio tarifen mantentzea</b>	Tarifen sorrera, ezabaketa edo eguneratzea
<b>Jabeenganako tarifen elkartzea</b>	Jabeen eta tarifen arteko elkarketa
<b>Artikulu/tarifa elkartzea</b>	Artikuluen eta tarifen arteko elkarketa
<b>Manipulazioaren fakturazio bideko azterketa</b>	Aurretik egindako jaurtiketen berrikustea
<b>Biltegiatzeko fakturazio bideko azterketa</b>	
<b>Manipulazioaren fakturazio bideko desagitea</b>	Aurretik egindako jaurtiketen desagitea
<b>Biltegiatzeko fakturazio bideko desagitea</b>	
<b>Fakturen sortze automatizatua</b>	Fakturak modu automatikoan sortu
<b>Fakturen eskuzko sortzea</b>	Fakturak erabiltzaileak kontzeptu zehatz batzuk aukeratuz eratu
<b>Fakturen mantentzea</b>	Fakturen sorrera, ezabaketa eta eguneratzea
<b>Fakturen ixtea</b>	Fakturak ixteko prozesua
<b>Fakturen inprimatzea</b>	.pdf fitxategi bat sortu

Taula 3 - Prozesuen zerrenda



Biltegiatze zerbitzuak eskaintzen dituzten enpresei gauzak erraztuko dizkio baita ere. Haiek eskaintzen dituzten zerbitzuak modu automatizatuago batean kudeatuko lituzkeen, adibidez, biltegi bateko toki konkretu bat alokatzen duten bezeroek, leku hori erabiltzeagatik ordaindu beharko dute. Hori kudeatuko duten tarifak aplikatuko dira eta automatikoki bezero horrek ordaindu beharko dituen biltegiatze zerbitzuak faktura batean agertuko dira, pertsona batek edozein eragiketa egin barik.

Bestalde, helburu gisa hartu genezaken beste aspektu bat honako hau izango litzateke: logistikako operadore bat lortzea. Logistikako operadore bihurtzeko In-LOG WMS aplikazioak funtsezko beste bi modulu beharko lituzke. Atal berri hau gehituz aplikazioa logistikako operadore bihurtzeko beste pausu bat beteko genuke.

Laburbilduz, honako hauek izango lirateke helburu nagusiak:

- Fakturatze-atal berria In-LOG WMS aplikazioari gehitzea.
- Aplikazioa logistikako operadore bihurtzeko beste pausu bat betetzea.
- Fakturatze prozesuak automatizatzea.

## Proiektuaren irismena

### PROIEKTUAREN FASEAK

Proiektu bat aurrera eraman ahal izateko fase desberdinak jasan behar ditugula esan daitezke.

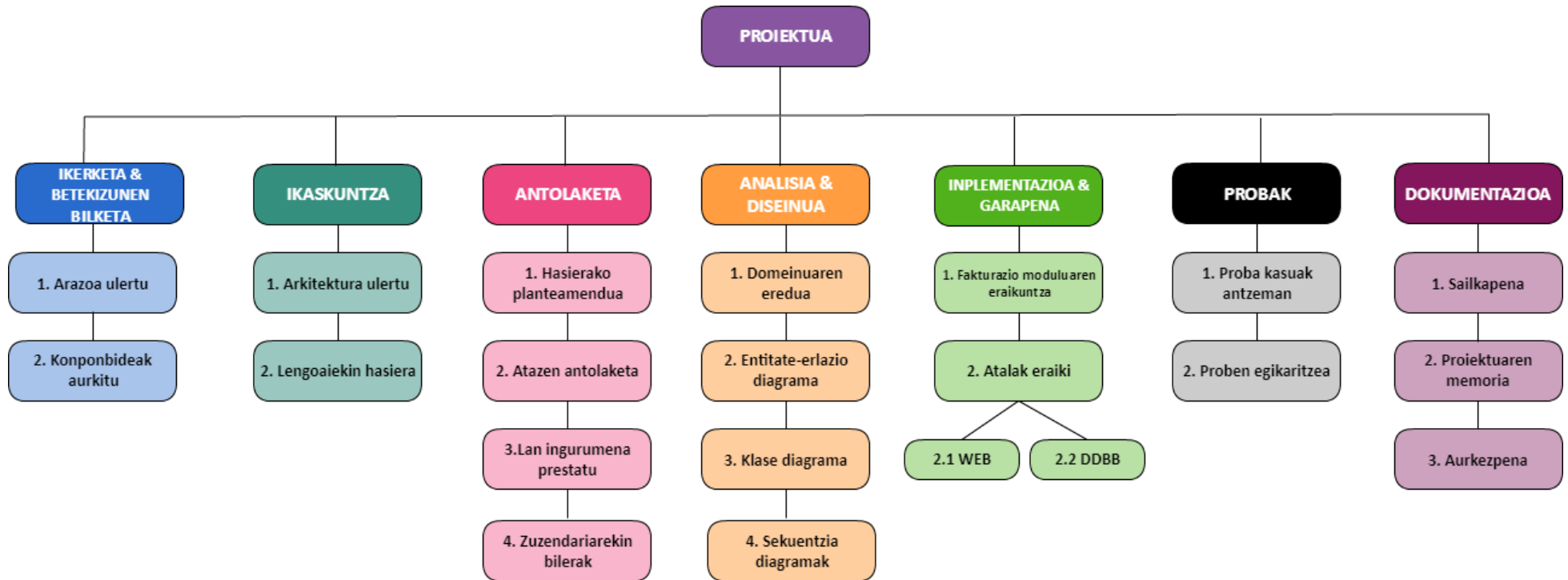
- **Ikerketa:** Ikerketa erabilgarria izango da honako atal hauei irtenbidea aurkitzeko:
  - Proiektuaren irismena identifikatu.
  - Proiektuaren helburu nagusiak identifikatu.
  - Arazoa konpontzeko erantzunak aurkitu.
  - Erantzun bakoitzeko haren alde onak eta txarrak antzeman.
- **Ikaskuntza:** Aurreko atalean ikusitako kontzeptuak praktikara eramateko fasea, beste modu batean esanda, erabiltzen ikasi proiekturako beharrekoa. Hurrengo pausua inplementatuta dagoen aplikazioaren egitura ulertzea izan da. Hau da, zertarako erabiltzen da arkitekturako atal bakoitza eta atal bakoitza inplementatuta dagoen lengoaien oinarritzko ezagutzak hartzea.
- **Antolaketa:** Proiektua aurrera eramateko modu antolatu batean ezarritako atazak. Ataza bakoitzeko behar diren erramintak edo materiala adieraziko da eta ataza bakoitza aurrera eramateko behar izango den denbora finkatu beharko da. Fase honetan ere proiektuaren zuzendariarekin izandako bilerak sartuko dira.
- **Analisia & Diseinua:** Aplikazioa inplementatzerako orduan eta dagozkion eragiketak aurrera eramateko modu eraginkor batean NOLA egin behar diren gauzak. Deskribatutako funtzionalitateen sistemaren osagaien diseinuan datza. Entitateen eta

sekuentzialaren arteko erlazioak eta eragiketen arteko elkarrekintza azaltzeko eraikiko diren diagramak.

Aurreko faseetan proposatutakoa aurrera eramateko nola antolatu beharko den definitu. Ondoren diseinu xehe bat burutuko da atal desberdinen zehazpenak deskribatuz.

- **Inplementazioa eta garapena:** Inplementazioa aurrera eramateko kontutan izan beharko dugu denbora plangintza eta plangintza horretan finkatutako denbora tarteak betetzen diren edo ez egiaztatu. Kodearen idazketa fasea, diseinuan finkatutakoa kode bihurtuko da.
- **Probak:** Behin proiektuko inplementazio osoa daukagula, berriro ere probak egiteko ordua da. Nahiz eta inplementazioan zehar probak egin diren, fase honetan proba kasu batzuk definituko dira proiektuaren funtzionaltasunak modu zuzen batean exekutatzeko; dena den atal honetan berriro ere funtzionalitate guztiak probatuko dira.
- **Dokumentazioa:** Proiektuan zehar gertatutako guztia adierazi "paperean". Hasierako ideia, proiektuak izango dituen helburuak, aurkitutako arazoak, nola konpondu diren, denboren estimazioa, etab... Izango duen dokumentua.

Testu honen azpian agertzen den diagraman errazago ikus daiteke atal bakoitza zertan datzan, bakoitzean zehazki zer egin den azaltzen bai da.



Irudia 6- LDE diagrama

LDE diagraman sakonduz

#### *Ikerketa eta Betekizunen bilketa*

1. **Arazoa ulertu:** Proiektuak izango duen funtzionalitatea ulertu, hau da, ulertu zertan datzan proposatutakoa.
2. **Konponbideak aurkitu:** Proposamenari nola egin ahal diodan aurre, beste hitzetan esanda, nola lortuko dut proiektua burutzea?

#### *Ikaskuntza*

1. **Arkitektura ulertu:** Aplikazioa barnetik ikusteko lehen kontaktua, aplikazioa osatzen duten osagaia desberdinak ikusi eta aplikazioaren zein funtzionalitaterako diren ulertu.
2. **Lengoaiekin hasiera:** Inplementazioarekin aurrera egiteko erabiltzen diren lengoaia desberdinak zerrendatu. Lengoaia horiei buruz erabilgarriak izango diren liburuetan informazioa bilatu eta adibideak garatu.

#### *Antolaketa*

1. **Hasierako planteamendua:** Hasiera batean antzeman daitezkeen prozesuak antzematea eta horren hasierako estimazio bat egitea.
2. **Atazen antolaketa:** Atazen/prozesuen zerrenda hartuta prozesu horiek antolatu.
3. **Lan ingurumena prestatu:** Proiektua garatzeko behar diren erraminten prestakuntza. Beharreko softwarea instalatu eta lan ingurunea prestatu.
4. **Zuzendariarekin bilerak:** Proiektuaren zuzendariari ideiak aipatu berak zuzenketak edo berrikusketak egiteko.

#### *Analisia & Diseinua*

1. **Domeinuaren eredu:** Arazoari konponbide bat aurkitzeko eraikitzen den diagrama, eredu honetan arazoa konpontzeko beharko diren gaiak agertuko dira. Proiektuan agertzen diren entitateen eta hauen atributu batzuk izango dituen diagrama.
2. **Entitate-erlazio diagrama:** Proiektuan beharko diren entitateen arteko erlazioak bilduko dituen diagrama.
3. **Klase diagrama:** Proiektuak behar izango duen klase diagramaren eraikitzea aplikazioa behar bezala ibiltzeko.
4. **Sekuentzia diagramak:** Aplikazioak izango dituen funtzionaltasun batzuen sekuentzia diagramak.

#### *Inplementazioa & Garapena*

1. **Fakturazio moduluaren eraikuntza:** Modulu berriaren hezurduraren eraikuntza. Menu egitura zehaztu eta aplikazioak dituen atalen desberdinen arteko komunikazio bideak garatu.
2. **Atalak eraiki:** Modu orokor batean esanda, modulu berriaren barne egongo diren funtzionalitate guztien garapena; beheko prozesu hauek guztiak modu zehatzago batean agertuko dira.

**Probak**

1. **Proba kasuak antzeman:** aplikazioak behar bezala ibiltzen den edo ez ikusteko burutu behar diren baldintzak zerrendatu.
2. **Proben egikaritzea:** Zerrendatutako baldintza horiek exekutatu.

**Dokumentazioa**

1. **Sailkapena:** Dokumentazioak zenbat atal izango dituen. Behin atalak argi daudenean, bakoitza zertan datzan ulertu eta bakoitzeko ideia nagusiak idatzi.
2. **Proiektuaren memoria:** Memoria beraren garapena, aurreko ideien sailkapena kontutan izanda memoria garatu.
3. **Aurkezpena:** Proiektuaren defentsa burutzeko dagokion aurkezpena garatu.

Jarraian taula batean azalduko da zehatzago fase eta azpi fase bakoitzak hartuko duen denboraren estimazioa:

FASE IZENA	DENBORA ESTIMAZIOA (orduak)
<b>Ikerketa &amp; Betekizunen bilketa</b>	<b>20</b>
Arazoa ulertu	10
Konponbideak aurkitu	10
<b>Ikaskuntza</b>	<b>40</b>
Arkitektura ulertu	10
Lengoaiekin hasiera	30
<b>Antolaketa</b>	<b>49</b>
Hasierako planteamendua	10
Atazen antolaketa	10
Lan ingurumena prestatu	4
Zuzendariarekin bilerak	25
<b>Analisia &amp; Diseinua</b>	<b>34</b>
Domeinuaren eredia	8
Entitate-erlazio diagrama	8
Klase diagrama	8
Sekuentzia diagramak	10
<b>Implementazioa &amp; Garapena</b>	<b>206</b>
Fakturazio Moduluaren Eraikuntza	30
Atalak eraiki ( <i>begiratu taula 5</i> )	176
<b>Probak</b>	<b>20</b>
Proba kasuak antzeman	10
Proben egikaritzea	10

<b>Dokumentazioa</b>	<b>75</b>
Saillapena	5
Proiektuaren memoria	50
Aurkezpena	20
<b>GUZTIRA</b>	<b>444</b>

Taula 4 - Denbora estimazioa

Gorago esan bezala hona hemen modu zehatzago batean prozesuak zerrendatuko dituen taula eta prozesu bakoitzeko estimatutako denborak:

ATALA/PROZESU IZENA	DENBORA ESTIMAZIOA (orduak)
Fakturazio kontzeptuen kudeaketa	12
Manipulazioko fakturazio biderako jaurtiketa	8
Biltegiratzeko fakturazio biderako jaurtiketa	8
Eskuzko fakturazio kontzeptuen kudeaketa	12
Fakturazio tarifen mantentzea	12
Jabeenganako tarifen elkartzea	12
Artikulu/tarifa elkartzea	12
Manipulazioaren fakturazio bideko azterketa	12
Biltegiratzeko fakturazio bideko azterketa	12
Biltegiratzeko fakturazio bideko desegitea	12
Manipulazioaren fakturazio bideko desegitea	12
Fakturen sortze automatizatua	8
Fakturen eskuzko sortzea	12
Fakturen mantentzea	12
Fakturen ixtea	8
Fakturen inprimatzea	12
<b>GUZTIRA:</b>	<b>176</b>

Taula 5 - Prozesuen denbora estimazioa

Baina, nondik atera da estimazio hau? Zergatik izango da ordu kopuru hori eta ez gehiago edo gutxiago? Honek guztiak justifikazio bat dauka, hasieran enpresak fakturatze-modulu berriak izango zituen atal guztien estimazio bat egin zuen. Estimazio hori aplikazioaren funtzionamenduaren jakitun batentzat eraiki zen.

Proiektua aplikazioaren “ez jakitun” batek inplementatu duenez gero, estimazio bakoitzari ordu kopuru txiki bat gehitu zaio esperientzia falta horrengatik.

## Denbora plangintza

Proiektua burutzeko atazak zeintzuk diren jakin ondoren, ataza horiek denboran zehar nola garatuko diren zehaztuko da. Horretarako, Gantt diagrama eraiki da prozesu bakoitzaren iraupena argiago ikusteko. Diagrama honek aurreko tauletan (*Taula 4 – Denbora estimazioa*) aipatutako prozesuak agertuko dira baita ere.

Demagun langile batek lanaldi partziala duela, beraz, astero 30 ordu lan egingo ditu. Gantt diagraman egindako plangintza ikusita, proiektuaren iraupena 6 hilabete eta erdikoa izango dela esan dezakegu. Kalkuluak egiteko demagun hilabete batek gutxi gora behera 4 aste dituela, hona hemen lortutako emaitzak:

$$\text{Hilabete kopurua} * \text{Aste kopurua} * \text{Lan orduak} = 6,5 * 4 * 30 = 780 \text{ ordu}$$

Ordu kopuru honi jai egun nazionalak eta autonomikoak eta asteburuko egunak kendu behar dizkiogu. Galduko diren orduen kalkuluaren estimazioa 68 ordu direla adierazi da. Kalkulu hau egin ondoren hauek izango dira proiektuari dedikatu ahal dizkiogun orduak guztira:

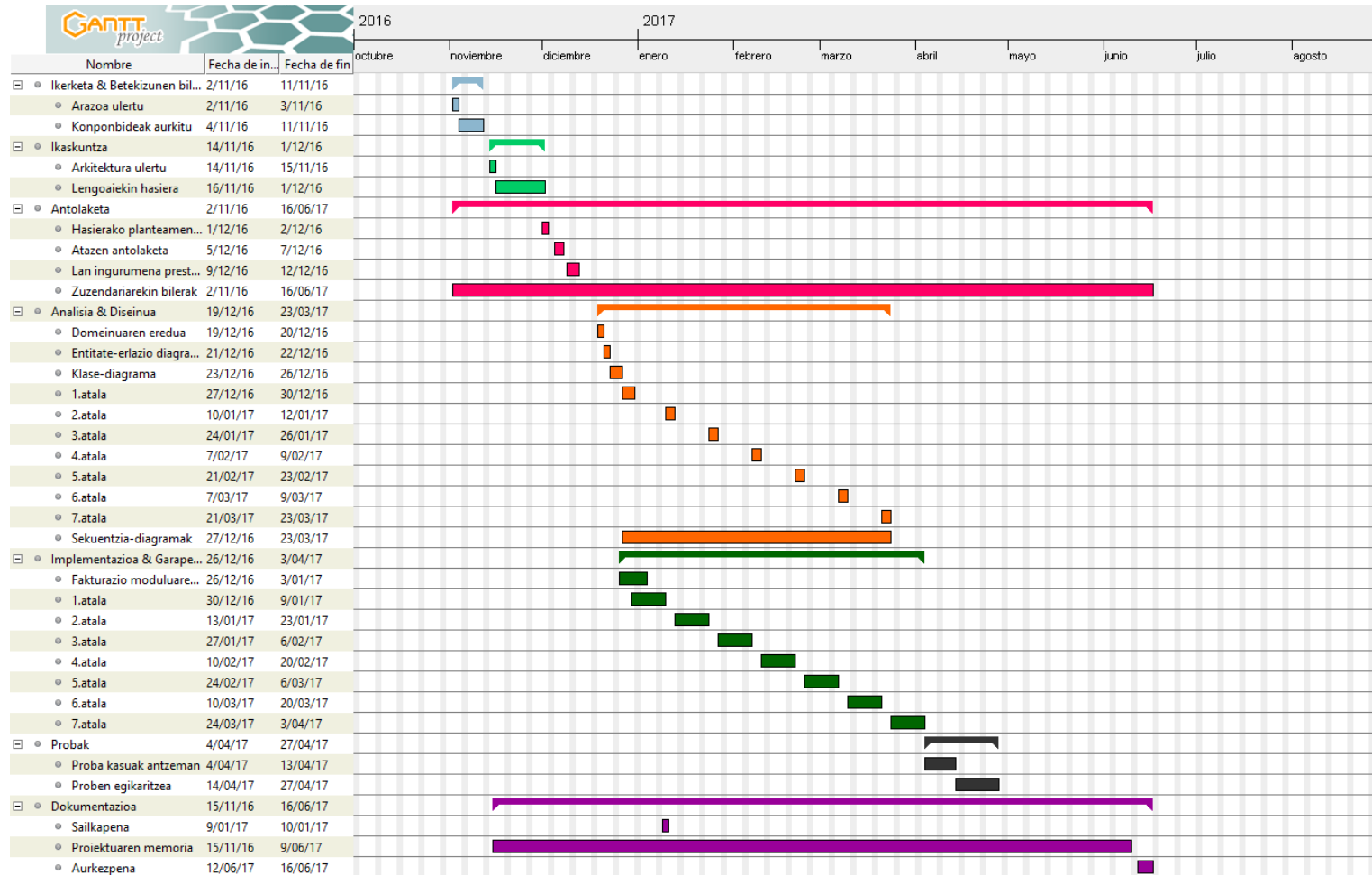
$$\text{Ordu kopurua guztira} - \text{Jai egunak} = 780 - 68 = 712 \text{ ordu}$$

Gantt diagraman ikusten denez gero, ataza batzuk batera garatuko dira, adibidez, dokumentazioa eta zuzendariarekin bilerak proiektu garapenaren osoan zehar paraleloan edo ia paraleloan egongo dira.

Iteratibo-inkremental motako garapen prozesua erabili da, hau da, diseinua eta inplementazioa batera garatu dira. Gantt diagraman ikusten denez gero, analisia eta inplementazioa ataletan banatu dira planifikazioa zehatzagoa egiteko. Atal bakoitzeko diseinua eta zer garatu den hobeto ulertzeko azaltzen hurrengo taula ikusi:

ATALA	ZEREGINA
1	Fakturazio kontzeptuak + Fakturazio biderako jaurtiketa (Manipulazioa + Biltegitratzea)
2	Eskuzko fakturazio kontzeptuak + Fakturazio tarifen mantenimendua
3	Jabeenganako tarifen elkartzea + Artikulu/tarifa elkartzea
4	Fakturazio bidezko azterketa (Manipulazioa + Biltegitratzea) + Fakturazio bideko desegitea
5	Fakturen sortze automatizatua + Fakturen eskuzko sortzea
6	Fakturen mantenua
7	Fakturen ixtea + Fakturen inprimatzea

Taula 6 - Atalen banakapena



Irudia 7 - Gantt diagrama



## Software baliabideak

Hona hemen proiektua aurrera eramateko erabili ohi diren software baliabideak:

- **Eclipse Java EE IDE for Web Developers MARS.2:** Software irekiko programazio erramintaz osatutako software plataforma bat da. Aplikazioaren WEB atala garatzeko erabilia, erraminta honekin batera plugin hauek erabili dira aplikazioaren atal hau gauzatzeko:

Plugin –aren izena	Zertarako?
<b>Red Hat (JBoss 7.1.1)</b>	<ul style="list-style-type: none"> <li>• Aplikazioa martxan jartzeko</li> <li>• Behin aplikazioa martxan zegoela probak egin ahal izateko</li> </ul>
<b>Report Design(Jaspersoft Studio)</b>	<ul style="list-style-type: none"> <li>• Txostenak garatzeko</li> <li>• Txostenak konpilatzeko</li> </ul>

Taula 7 - Eclipse plugin -ak

- **PL/SQL Developer:** Oracle inguruneko datu baseen garapenerako software integratua. Aplikazioaren Datu Basearen atala kodetzeko erabilia; erraminta honekin Datu-Base berara WEB atalarekin konektatzeko beharrezko package-ak inportatu dira eta bertan konpilatu baita ere. Fakturatze moduluak egiten dituen barne prozesuak ere erraminta honekin garatu eta konpilatu dira.
- **Google Chrome:** Googlek garatutako web nabigatzailea, kode irekiko osagaiekin konpilatuta dago. Aplikazioaren probak burutzeko eta JavaScript kodea pausoz-pauso debugatzeko erabilia.
- **Mozilla Firefox:** Kode irekiko web nabigatzailea. Aplikazioaren probak burutzeko erabilia.
- **Microsoft Office:** Proiektuaren dokumentazioa egiteko erabili den ofimatika erraminta. Excel proiektuaren plangintzarako, Word proiektuaren memoriarako hain zuzen.
- **Dropbox:** Proiektuaren dokumentazioa biltegitzako erabilitako Cloud biltegitzako zerbitzua.
- **Git:** Bertsioen kontrolerako erabiltzen den softwarea. Oinarritutako aplikazioaren errepositoria hartuz, fakturatze moduluaren aukera berriak gehituta aplikazioaren bertsio berri bat garatzeko erabilia. **TortoiseGit** erraminta erabili da bertsioen kontrola egiteko.
- **Redmine:** Proiektuen kudeaketarako erraminta. Redmine-n proiektuaren garapenean sartutako orduak adierazi ditut, proiektuaren jarraipen zehatza egiteko.
- **Cacoo:** Antolatzaile grafikoak eratzeko online erraminta. Erraminta honekin memorian zehar agertzen diren diagramak eratu ditut: LDE, UML, sekuentzia diagramak,
- **Zotero:** Bibliografiak kudeatzeko libre eta doako erraminta. Erabilitako liburuen bibliografia eta kontsultatutako webguneen zerrenda eratzeko erabilia.

## Hardware baliabideak

Proiektua inplementatzeko eta inplementatutakoarekin frogak egiteko honako hardware hau erabili da:

Ordenagailu eramangarria	ASUS R510J
Prozesadorea	Intel Core i7 2.6GHz
RAM memoria	8GB DDR3
Disko gogorra	1TB
Txartel grafikoa	NVIDIA GEFORCE 930M

Taula 8 - Hardware baliabideak: PC

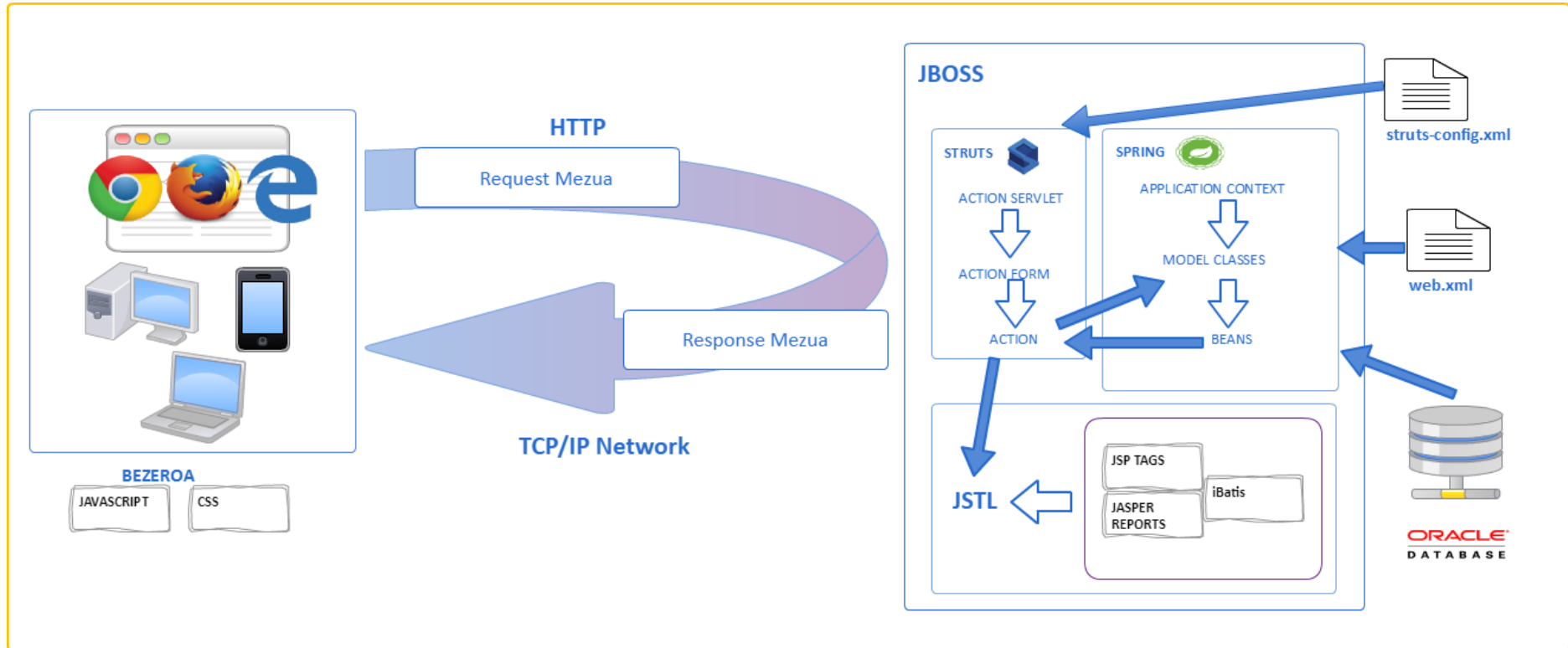
Zerbitzaria	
Sistema eragilea	CentOS 6.5
Linux distribuzioa	2.6.32-642.11.1.el6.x86_64
Aplikazioen Zerbitzaria	JBOSS 7.1.1
Java bertsioa	1.6.0_45-b06
DDBB bertsioa	Oracle 11g 11.2.0.4.0

Taula 9 - Hardware baliabideak: Server

## Arkitektura

Atal honetan oinarritutako aplikazioaren arkitektura nolakoa den eta prozesuen exekuzioa nola egiten den azalduko da. Aplikazioaren arkitektura bezero/zerbitzari arkitektura dela, lehenik eta behin haien arteko komunikazioa nola burutzen den azalduko da eta ondoren bai bezeroaren bai zerbitzariak duten egitura azaltzeko alde batetik Client Side izango dugu eta bestetik Server Side.

Bezero/zerbitzari arkitektura



Irudia 8 - Arkitektura

Barne egitura

*Client side*

CSS

Cascading Style Sheets, dokumentu bat pantailan nola agertuko den deskribatzen duen mekanismo bat da. Garatzaileei askatasun handia uzten die haien estiloak definitzeko. Kasu honetan HTML (Hyper Text Markup Language) eta XML (eXtensible Markup Language) dokumentuei estilo bat emateko erabiltzen da.

JavaScript

Web orrialdeetako jarrera zehatz batzuk programatzeko sortu zen programazio lengoaia da, erabiltzailearen elkarrekintzari erantzuna emateko eta web orrialdeentzako automatismo errazak sortzeko.

Aplikazio honetan formularioak balioztatzeke erabiliko da, hau da, Bidalitako datu guztiak baliozkotzat joko ditu edo ez. Gainera baliozkoak ez izatekotan dialogo bat erakutsiko dio erabiltzaileari pantailan.

*Server side*

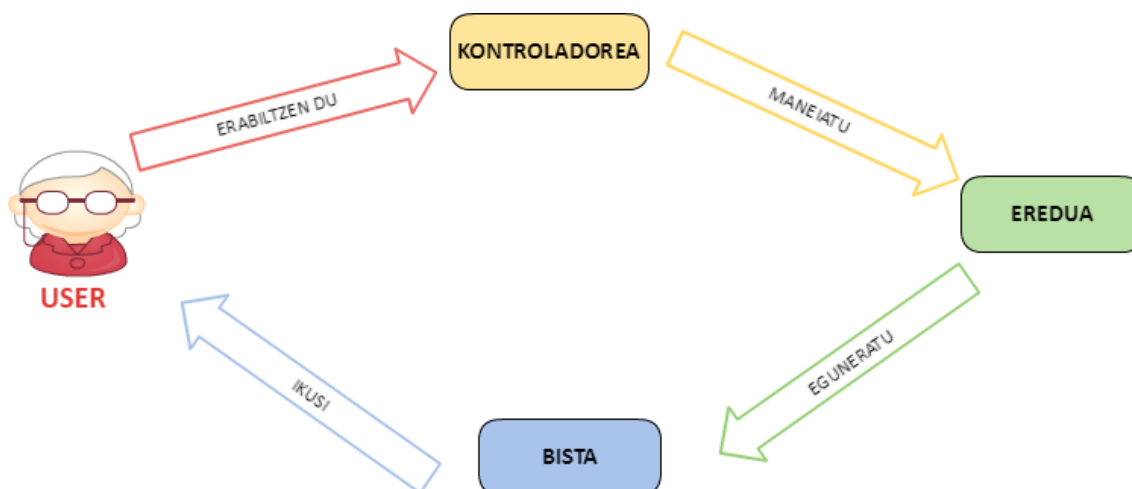
Struts

MVC (Kontroladore-bista ereduan) filosofian oinarritzen den framework bat da, web Java aplikazioak eraikitzeke erabilia. Struts –ek ez gaitu bakarrik lan egiteko API batekin hornitzen, garapen filosofia bat ematen digu baita ere. Framework bat erabiltzearen alde onenetariko bat egiten ditugun gauzak aurretik frogatu dituztela da, beraz, guk ez ditugu berriro ere frogatu behar.

Gorago esan dudan moduan framework honek MVC arkitekturan oinarritzen da eta arkitektura horretako atal bakoitza zertarako eta nola erabiltzen den azalduko da:

- **Kontroladorea:** Struts –ek emandako klase baten servleta da. Erabiltzaileak egiten dituen eskaera guztiak servet honetara berbidaltzeko web aplikazioa konfiguratu beharko da, web.xml fitxategia erabiliz.  
Nahi den ekintza exekutatzeko, kontroladoreak erabiltzaileak egindako eskaera guztiak dagozkien klasera bidaltzen ditu. Ekintzak (action) egikaritzen dituzten klase guztiek Action klasea heredatuko dute.
- **Bista:** JSP orrialdeak erabiliz inplementatuko da, ActionForm klaseak formularioetan sartutako datuak hartuko dituzte eta hauek baliozkotzat joko dituzte edo ez, eta etiketen liburutegiei ezker formularioekin lana asko erraztuko da eta erroreak erakustea utziko dute.
- **Eredua:** Garatzailearen esku egongo da ereduaren inplementazioa.

Struts-ek http (*Hypertext Transfer Protocol*) eskaera bat prozesatuko du, hurrengo irudian agertzen den moduan:



Irudia 9- MVC osagaiak

- ✓ Bezeroak egiten duen eskaera Struts –eko kontroladoreak hartuko du, eskaera hori ez da URL bat izango, ekintza (*action*) baten izena baizik. Kontroladoreak ekintza hori identifikatuko du eta ekintza bera egikarituko du.
- ✓ Egikaritutako ekintzak eredu eguneratuko du, eredu horretatik datuak lortuz eta beans –ean biltegituz. Beans erabiliz datu basera eta zerbitzari aldean dauden beste elementuetara sarbidea izango da.
- ✓ Buelatuko balioaren arabera, kontroladoreak hurrengo bista zein izango den erabakiko du.
- ✓ Bista bistaratzeko/eraikitze behar dituen datuak hartuko ditu.
- ✓ Bistak bezeroari datuak erakutsiko dizkio.

#### Baina nola eraikitzen da komunikazio bide hau?

struts-config.xml izena duen konfigurazio fitxategi bat egongo da bertan “erlazio” horiek definitzeko. Hau da, ekintza (*action*) bakoitzari esleitutako izena eta Java klase baten elkartzeko bat egingo du fitxategi honek.

Eskaera eta action guztien elkarketak <action-mappings> etiketa barruan kokatuko dira. Elkarketa honetan elementu (action) bakoitza <action> etiketa barruan ipiniko da non etiketa honek bi atributu izango dituen:

- path: Eskaera gordeko duen atributua.
- type: Eskaeran aipatutako ekintza horrek inplementatutako Java klasea.

#### Spring

Framework honek programazio eredu oso bat eta Java –n oinarritzen diren aplikazioentzako konfigurazioa eskuragarri ipintzen du. Transakzioen kudeatzaile gisa eta mendekotasunen injekzioak burutzeko erabili da aplikazio honetan. JavaEE aplikazioek

erabiltzen duten teknologien beste aukera bat da. Lehen, aplikazioen zerbitzariak errekurtso asko behar zituzten eta haiekin lan egitea oso zaila egiten zen.

Spring –ek dependenzien injekzioa edo ohiko objektuak negozio-objektu bihurtzearen ideia hedatu zuen. Ideien hedapen honek aplikazio arinago bat eta garapen erraza eta azkarrago bat onartzen zuten. Honi ezker negozio-geruzarako framework bat izatetik aplikazio osoko geruzetarako teknologia bat izatera bihurtu zen.

Baina zer da negozio-geruza? Lehenik eta behin kontzeptu hau zertan datzan azalduko da, geroago agertuko diren beste azalpenak ulertzeko errazago izango da eta.

- ***Negozio-geruza:*** exekutatzan diren programak dauden lekua da. Erabiltzailearen eskaerak hartzen dira eta prozesuaren ondoren emaitzak bueltan bidaltzen dira. Aurkezpen-geruzarekin komunikatzen da (*geruza honek erabiltzaileari sistema aurkeztuko dio*), eskaerak jaso ondoren emaitzak aurkezteko. Datu-geruzarekin komunikatzen da baita ere (*geruza honetan datuak daude eta datuen atzipenaz arduratuko da*) datu base kudeatzaileari eskaerak egiteko datuak lortu ahal izateko.

Spring –ek aukeratutako objektuen bizi zikloa kudeatu dezake. Framework –ak kudeatutako objektuei orokorrean Spring-eko *bean* deitzen zaie. *bean* objektuak singleton –ak dira, hau da, edukiontzian horren instantzia bakarra egongo da eta instantzia bat eskatzen den bakoitzean benetan lortuko dena objektu horretara erreferentzia izango da. Edukiontzi horren konfigurazioa `WebApplicationContext` klasearekin egiten da, *web.xml* fitxategian definituko da *listener* bat moduan.

Spring –ek oinarrizko hiru elementuekin hornitzen gaitu:

- **Enterprise zerbitzuak**
  1. Objektu transakzionalak bihurtu
  2. Objektua rol batzuei atzitu
  3. Objektua modu erremotoan eskuragarri
- **Estereotipo konfiguragarriak** : Estereotipo pertsonalizatuak definitu, adibidez, estereotipo berri batek negozio-objektu bat adieraziko du. Estereotipo hau automatikoki cacheatuko litzateke eta rol zehatz bat duten erabiltzaileentzat egongo litzateke eskuragarri.
- **Dependentzien injekzioa:** Objektu batek ematen duen zerbitzua, zerbitzu bera eskatzen duen beste objektu bati eman. Adibidez, aurkezpen-geruzako objektu bat negozio-geruzan dagoen objektu batekin komunikatu ahal izatea.

## Datuei sarbidea

### Datasources

Datu baseko taulak non dauden (ze datu basetan dauden) adieraziko Datasource –ak. Data source –ak softwareak datu base zehatz batera konektatu ahal izateko beharko duen informazioa gordeko du.

Data source –a definitzeko orduan datu basearen informazioa eta zein zerbitzarian dagoen kokatuta adierazi beharko zaio.

Kasu honetan JBOSS zerbitzaria erabiltzen denez, data source –ak JBOSS –ek duen standalone.xml fitxategian definituko dira. Bertan datu basearen izena (id), hara konektatzeko pasahitza eta IP-a adieraziz lortuko da datu baserako sarbidea.

## IBatis

Datuak mapeatzeko framework arina eta iraunkortasun API bat da. Datu baseko eskema bat aprobe txatu eta Java aplikazioentzako iraunkortasun geruza sortzeko erabili daiteke.

Hona hemen iBatis –ek eskaintzen dituen aisetasunak:

- JDBC (*Java Database Connectivity*) erabiliz idatzitako SQL –a exekutatu du.
- Objektuen ezaugarriak PreparedStatement -entzat (parametrizatu daitezkeen SQL sententziak) prestatzen ditu.
- Query baten emaitzak objektu batera edo objektuen zerrenda batera mapeatuko ditu.

iBatis –en konfiguraziorako honako fitxategi eta pausu hauek beharrezkoak izango dira:

- iBatis –en konfigurazio fitxategi bat egongo da, bertan parametro batzuk (adibidez cache –a aktibatuta dagoen edo ez), DataSource –a eta SQL mapeoak adieraziko dira.
- .xml fitxategiak (kasu honetan pantaila bakoitzak mota honetako fitxategi bat beharko du) exekutatu diren SQL sententziak, mapeatuko diren parametroak edo mapeatuko diren emaitzak izango ditu.
- SQL sententziak exekutatuzeko `SQLMapClient` interfazea inplementatuko duen klasea izango dugu.

## Txostenen sorrera JASPER Reports

Txostenak eraikitzeke erabilitako liburutegia. Txosten hauek .pdf fitxategi batean, monitorean, inprimagailu batera, etab... bidali eta gorde daitezke.

Java lengoia idatzia dago eta web aplikazioetan eduki dinamikoa sortzeko erabiltzen da. Kasu honetan, *Report* –en garapena garatu ahal izateko liburutegi hau erabili da eta Eclipse –k duen *JasperSoft Studio* plugin –ari ezker hauen diseinua egitea askoz errazagoa izan da.

## HTML JSP edukiaren sorrera

### TAGLIBS STRUTS

Formularioekin elkarrekintza burutzeko erabiltzen da. Hau horrela ibiltzeko Struts konfiguratu beharko da eta horretarako `struts-config.xml` konfiguratu da. Horretarako hemen azpian azalduko da zer adierazi beharko da konfigurazio fitxategi horretan:

- Aplikazioak izango duen funtzionamenduaren fluxua, hau da, zer ekintza burutuko diren eta ekintza horiek behar dituzten “bideen” adierazpena idatzi beharko da.
- Errore bat agertzekotan zer egin beharko da.
- Bidalitako eta balioztatutako informazioaren datu motak zein formularioarekin dauden elkarturik.

## JSTL

JavaServer Pages Standard Tag Library (*JSTL*), Java EE –ren osagai bat da. Java Server Pages –ak hedatzen ditu, etiketen lau liburutegi emanez. Web orrialde dinamikoak eraikitzeke erabiltzen dira aipatutako liburutegiak.

Esan bezala, etiketa liburutegi hauek JSP zehazpenetik hedatzen dira, JSP –ak aldi berean Servlet –etik hedatzen dira. JSTL API –ak gainera pertsonalizatutako etiketen liburutegiak eraikitzen uzten du.

Modulu hau eraikitzeke erabili diren etiketak hauek izan dira:

ETIKETA IZENA		
core	fmt	functions
<ul style="list-style-type: none"> <li>• Iterazioak</li> <li>• Baldintzapenak</li> <li>• URL manipulaziorako</li> <li>• ...</li> </ul>	Karaktere kateak zifretara bihurtu.	Funtzioak izango dituen etiketa.

Taula 10 - Erabilitako JSTL etiketak



## Ebaluazio ekonomikoa

Erabilitako errekurtsoek emango dituzten irabaziak konparatzeko eta neurtzeko izango da ebaluazio ekonomikoaren atal hau. Kontutan izan beharra dago proiektu hau merkatuan dagoen aplikazio baten parte dela, beraz, proiektu honen inplementazioarekin diru-sarrerak espero dira.

### Eskulana

Kalkulu hau lortzeko hainbat faktore kontutan izango dira. Alde batetik nire esperientzia, lan-merkatuan arlo honetan nire esperientzia 0-1 urte artean ezarriko dut; kontutan hartuko den beste faktore bat programatzaile urtero batek duen soldata izango da. Beraz, faktore bi hauek kontutan izanda 0 eta 1 urte arteko esperientzia daukan programatzaile batek 19.861€ kobratzen ditu urtero. Honetan oinarrituta hilabetero eta orduro kobratzen duenaren kalkulua egingo da.

$$\text{Soldata urtero} \div \text{Hilabete kopurua} = 19.861 \div 12 = 1.655,08\text{€/hilero}$$

Orduro zenbat kobratzen den kalkulatzeko demagun hilabete batek 4 aste dituela, egunero 8 orduko lana egiten dela eta astean 5 egunetan lan egiten dela.

$$\frac{\text{Hileroko Soldata}}{\text{Hileroko asteak} * \text{Egun bateko lan orduak} * \text{Astero egunak}} = \frac{1.655,08}{4 * 8 * 5} = 10,34 \text{ €/ordu}$$

10,34€ kobratuko dira lan egindako ordu bakoitzeko.

Proiektuaren planifikazioa egitean 535 orduko denbora estimazioa egin da eta 535 ordu hauen barruan 100 ordu arloaren ikerketa eta ikaskuntzara (40 ordu ikerketara eta 60 ordu ikaskuntzara) eskainiko dira. Ordu kopuru hori ez da kontutan hartuko eskulanaren kostua kalkulatzeko, beraz:

$$\text{Estimatutako orduak guztira} - \text{Ikerketa\&Ikaskuntza orduak} = 444 - 100 = 344 \text{ ordu}$$

Eskulaneko gastua honela geratuko litzateke:

$$\text{Ordu kopurua} * \text{Soldata orduro} = 344 \text{ ordu} * 10,34 \text{ €/ordu} = 3.556,96 \text{ €}$$

**Eskulan gastua guztira: 3.556,96€**

## Hardware gastuak

Atal honetan proiektuaren garapenerako erabili den material elektronikoaren gastuaren kalkulua burutuko da.

### *ASUS R510J ordenagailu eramangarria*

Ordenagailu eramangarri baten bizi iraupena 5 urtetan (60 hilabete) ezarrita dago eta honen salneurria 650€ izan zen orain dela urte bat.

$$\text{Hileroko amortizazioa} = 650/60 = 10,83 \text{ €}$$

Proiektuaren garapenean zehar erabiliko da ordenagailu eramangarri hau eta estimatutako iraupena 6 hilabete eta erdikoa da, beraz:

$$\text{Amortizazioa} = \text{Hileroko amortizazioa} * \text{Garapenaren iraupena} = 10,83 * 6,5 = 70,41 \text{ €}$$

**Hardware gastuak guztira: 70,41 €**

## Software gastuak

Atal honetan proiektuaren garapenerako erabili den softwarearen gastuaren kalkulua egingo da.

Erabili diren aplikazio gehienak doakoak dira beraz haien kostua 0€ -koa izango da, baina horietako bik ordainpekoak izan dira.

### *Microsoft office*

Dokumentazio eta planifikazioaren garapenerako erabili izan da. Software hau erabili ahal izateko lizentzia bat ordaindu beharra dago eta haren salneurria urte batengatik 69 € -koa izan da.

### *PL/SQL Developer*

Datu basearen atala garatzeko erabili da eta software hau erabili ahal izateko lizentzia bat ordaindu beharra dago, haren salneurria 183 € -koa da.

**Software gastuak guztira : 69 + 183 = 252 €**

Gastuak guztira

Gastuak modu argiago eta laburbildu batean ikusteko taula:

<b>KONTZEPTUA</b>	<b>GASTUA (€ -tan)</b>
Eskulana	3.556,96€
Hardware	70,41€
Software	252€
<b>GUZTIRA</b>	<b>3.879,37€</b>

*Taula 11 - Gastuak laburtuta*

## Arriskuen kudeaketa

Baliteke proiektuaren garapenaren zehar arriskuak agertzea, batzuk besteak baino handiagoak izan daitezke, baina oraingo atal honetan garapen honetan zehar izan ahal ditugun arriskuak antzemango ditugu eta hauei aurre egiteko hainbat ebazpen azalduko dira. Horretarako arrisku bakoitzarekiko haren agerpen probabilitatea kontutan izango da eta konponbide gisa prebentzio plan bat sortuko da.

Esan bezala arrisku guztiak desberdinak dira, haien larritasunaren arabera proiektua larritasun handiago batekin edo txikiago batekin kaltetuko dute.

Jarraian software garapen batean zehar aintzakotzat hartuko ditudan arriskuak zerrendatuko ditut, baina lehenago hiru taldeetan banatuko dira hobeto azaldu ahal izateko:

### 1. Garapenarekin aurrera jarraitzeko arazoak

Proiektu bat hasterako orduan garrantzitsua da landuko diren gaiak menperatzea, hau gertatu ezean baliteke ustekaberen bat gertatzea eta horrek proiektuaren entrega epea atzeratzea. Ez da ohikoena baina gertatu daiteke, beraz, kontutan izango den arrisku bat izango da.

### 2. Gaixotasunak edo lesioak

Arrazoi medikoengatik lanarekin aurrera egiteko sortutako ezgaitasun batean datza. Baliteke ezgaitasun hau lanean gauden bitartean edo lanetik kanpo sortzea.

### 3. Ekipamendu informatikoarekin arazoak

Proiektua garatzeko erabiltzen ditugun erraminta fisikoak edo erabiltzen diren softwareak hondatu daitezke, erabilera desegoki batengatik edo besteren arrazoiengatik.

Azpiko taulan arazo bakoitzarekiko prebentzioa, haren kontingentzia plana, agertzeko duen probabilitatea eta bakoitzak izango lukeen eragina azalduko da.

	<b>Garapenarekin aurrera jarraitu ezin</b>	<b>Gaixotasunak/lesioak</b>	<b>Ekipamendu informatikoarekin arazoak</b>
<i>Prebentzioa</i>	<ul style="list-style-type: none"> <li>• Aurretik behar den formazioa hartu.</li> <li>• Informazioa edo laguntza eman ahal dizkiguten baliabideak ezagutu (liburuak, webguneak,...)</li> </ul>	<ul style="list-style-type: none"> <li>• Lanpostuan jarrera egoki batean lan egin.</li> <li>• Monitorearekiko distantzia egokia mantendu.</li> <li>• Lanpostuan gaudela estresatzen garela nabarituz gero, atsedean bat egin.</li> </ul>	<ul style="list-style-type: none"> <li>• Aldiro ekipamendu osoari azterketak egin.</li> <li>• Astero babes-kopiak egin bai ekipamendu lokalean zein Cloud biltegitratze zerbitzu batean.</li> </ul>
<i>Kontingentzia Plana</i>	<ul style="list-style-type: none"> <li>• Soluzio bat aurkitzen saiatu.</li> <li>• Baliabideetaz baliatu soluzioa aurkitzeko.</li> </ul>	<ul style="list-style-type: none"> <li>• Medikarengana joan, berak diagnostikoa egin dezan.</li> <li>• Kontrako diagnostikoa ematekotan, baja hartu edo osasuntsu egon arte atsedendia hartu.</li> </ul>	<ul style="list-style-type: none"> <li>• Beste ekipamendu bat erabili eta guztia berriro instalatu.</li> <li>• Softwarearen kasuan, babeskopiak erabili informazioa berreskuratzeko.</li> </ul>
<i>Agertze Probabilitatea</i>	Arazoa aurkitu eta soluzio bat eman %45	Gaixotasuna agertu edo lesio bat jasan %20	<ul style="list-style-type: none"> <li>• Hardware hondatu %15</li> <li>• Software hondatu %5</li> </ul>
<i>Eragina</i>	Erdia	Baxua	Baxua

Taula 12 - Arriskuen laburpena



## AURREKARIEN AZTERKETA

In-LOG WMS biltegien kudeaketa aurrera eramatea eta kalitate handiko zerbitzua eskaintzen duen web aplikazioa da. Aplikazioaren gaur egungo egoera logistikako operadore batek burutzen dituen funtzionalitate batzuk eskaintzea da, logistikako operadore batek orokorrean dituen funtzionalitate guztiak<sup>1</sup> eskaintzea baitu helburu gisa.

Merkatuan badago funtzionaltasun hauek eskaintzen duen softwarea. Baina esan beharra dago aplikazio hauek eskaintzen dituzten zerbitzuak finkoak direla, hau da, aplikazioak dituzten atalak kasu gehienetan ezin dira enpresa batek bakarrik behar dituen zerbitzuetara moldatu.

In-LOG WMS aplikazioak zerbitzu pertsonalizatu bat eskaintzen du. Bezero batek aplikazioko atal bat aldatu nahi ezker, bezero horrentzat moldatuko litzateke aplikazioa. Egingo litzatekeen aldaketa baten kasua hobeto ulertzeko adibide txiki bat azalduko da:

Demagun bezero bat Espainiakoa dela eta beste bat Estatu Batuetakoa. Espainian telefono zenbaki baten luzera 9 zenbakikoa da, Estatu Batuetako telefono zenbaki batek, aldiz, 7 zenbaki ditu. Estatu Batuetako bezeroak telefono zenbakiak 7 zenbakiko luzera izatea nahi du, beraz, In-LOG aplikazioan aldaketa txiki hori egitea posiblea izango litzateke.

### Merkatuan dagoen softwarea

Logistikako operadore batek eskaintzen dituen zerbitzuak ematen dituzten aplikazio esanguratsuen zerrenda bat egingo da, haiek eskaintzen dutena ikusteko. Batzuk ez dituzte logistikako operadore batek eskaintzen dituen zerbitzu guztiak ematen, dena den, logistikako merkatuan oso ezagunak eta erabiliak direnez, zerrendan agertuko dira. Alde batetik garraioen kudeaketarako softwarea azalduko da eta bestetik biltegien kudeaketarako erabiltzen dena.

### Garraioen kudeaketa

- **SAP:** Garraioaren kudeaketa errazten du. Garraio enpresa eta logistika-zerbitzu hornitzaileentzat bideratuta dago. Finantza edo fakturazio modulua ere badu, gastuen eta irabazien kudeaketa burutzeko.
- **IBM Enterprise Content Management:** Garraio enpresei kudeaketarekin laguntzen die, globalizazio, bermatzea, hornikuntza-kate malguen beharra eta segurtasuna hobetzearekin hain zuzen. Beraren bizi ziklo osoan zehar sortutako dokumentu eta izan dituen komunikazioak harrapatzen ditu.

---

<sup>1</sup> Sarrera atalean azaldu dira logistikako operadore baten funtzionalitateak

- **TRANSICS:** Gastuen kudeaketari ezker, garraio kasu bakoitzerako aukera hoberena hautatzea posiblea izango da. Prozesuen eta dokumentuen bizkortasuna bermatzen du haren sistema automatizatuari ezker.

#### Biltegien kudeaketa

- **Aqua Intelligent Warehouse:** Biltegi-prozesuen optimizazio, kudeaketa eta kontrolarentzako erabilia. Biltegiatze kostuen murrizpenean eta erakundeari zerbitzuak hobetzen lagunduko dio.
- **SIGA:** Biltegien kudeaketarako sistema integratu bat da (SGA). Merkantziaren kontrola urrutiko gailuak erabiliz burutzen du.
- **SQL (Pyme):** Enpresei funtzionalitate berriak emateko erabiltzen den ERP –a. Trazabilitate eta stock-kudeaketarako programentzat erabilia. Prozesu batzuk automatizatzea lortuko du.

Behin merkatuan jadanik dauden aukeren azalpena eginda, esan beharra dago hauek guztiak ezin direla web nabigatzaile batekin erabili. Hau da, mahaigaineko aplikazioak dira eta gailu edo ordenagailu bakoitzeko instalazio bat egin beharko da.

In-LOG WMS aplikazioarekin, aldiz, ez da instalazio prozesurik egin behar, web nabigatzailea ireki eta URL –a adieraziz lortuko da aplikazioa.

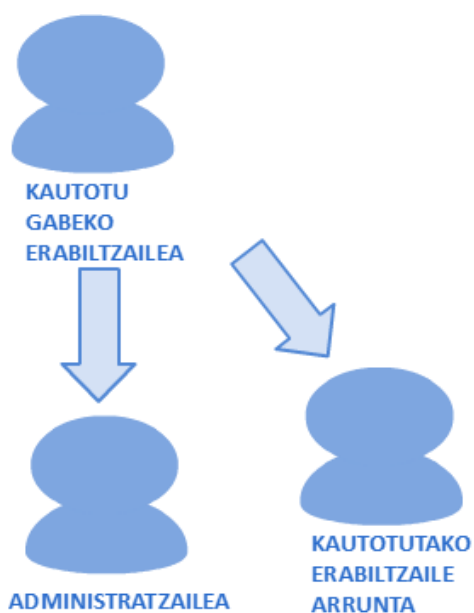


## BETEKIZUNEN BILKETA

Atal honetan dauden erabilpen kasuak azaltzeko aktoreen hierarkia eta domeinuaren ereduarekin azalduko dira.

### Aktoreen hierarkia

Aktoreen hierarkia azalduko da orain. Gainera aktore bakoitzarekiko deskribapen txiki bat emango da.



*Irudia 10 - Aktore Hierarkia*

Irudian ikusten den bezala bi erabiltzaile mota egongo dira. Alde batetik Administratzaile/Erabiltzaile rola izango duena eta bestetik soilik Erabiltzaile rola edukiko duena.

### Administratzaile / Erabiltzailea

Erabiltzaile arrunt bat izan izango da baina abantaila batzuekin. Aplikazioak duen Segurtasun modulurako baimenak izango ditu, modulu hori erabilia beste erabiltzaileei moduluak eta biltegiak atzitu ahalko dizkio.

### Erabiltzailea

Atzituta dituen aplikazioko atalak bakarrik erabili ahal izango ditu.

### Erabilpen kasuak

Erabilpen kasuak azalduko dira orain. Erabilpen kasu bakoitzeko haren funtzionamendua zein den eta deskribapen txiki bat egingo da. Aplikazio beran modulu berri

baten garapena burutuko denez, bakarrik kautotzea eta modulu horrekiko erabilpen kasuak izango kontutan izango dira.

### Kautotzea

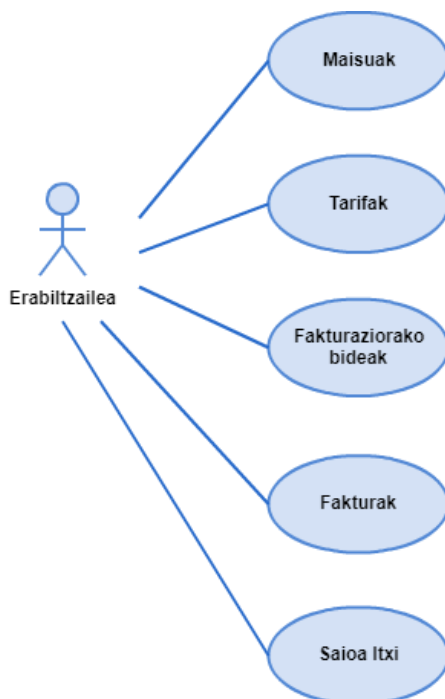
Aplikazioa erabiltzen hasteko lehenik eta behin erabiltzailea kautotu behar da. Horretarako Segurtasun modulutik erabiltzaile kode bat, pasahitza eta zein moduluetara izango duen sarbidea adierazi beharko du Administrazioaile rol –a duen beste erabiltzaile batek. 11. irudian kautotze kasua ikusi daiteke.



Irudia 11 - Kautotzea

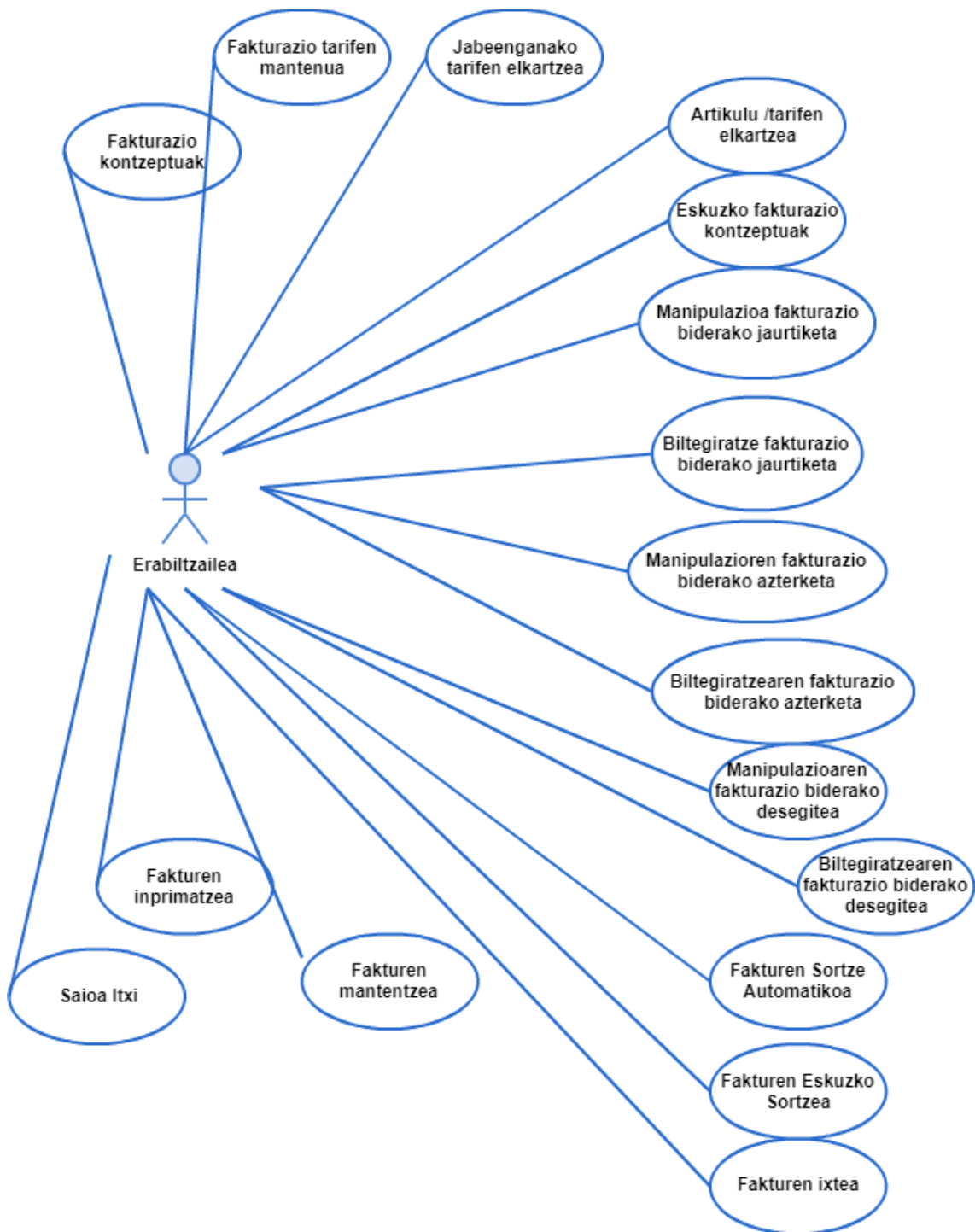
### Kautotutako erabiltzailearen aukerak

Erabiltzaile bat kautotuta dagoenean aplikaziorako sarbidea izango du, baina lehen esan dudan moduan, administrazioaileak atzitu dizkion moduluetara edo modulu bakoitzeko atal batzuetara izango du sarbidea. Orain adieraziko dira kautotutako erabiltzaile batek dituen aukerak. Bisualki errazago ikusteko begiratu 12. irudia.



Irudia 12 - Fakturatze-modulua erabilpen kasuak

Azpi erabilpen hauen atalak eta horien azpi atalen azalpena emango da orain. Modu argiago batean ulertu ahal izateko ikus 13. irudia.



Irudia 13 - Erabilpen & Azpi erabilpen kasuak

### Saioa itxi

Kautotuta dagoen erabiltzailea bere saioa itxi dezake. Aukera hau kautotu bezain laster izango du eskuragarri, baina erabilgarri dituen moduluen artean batera sartu eta gero eskuragarri izango du baita ere.

### Menu

Fakturatze-Modulua aukeratu eta gero erabiltzaileak ikusiko duen lehenengo gauza menu bat izango da. Bertan eskuragarri izango ditu aplikazioaren atal berri honek eskaintzen dituen funtzionalitateak. Hemen aipatzen diren funtzionalitate hauek guztiak Fakturatze-moduluaren lehenengo fasearentzako definitu diren atalak dira.

### Maisuak

Menu atal honek entitate-maisuen kudeaketarako sarbidea emango digu. Hona hemen atal hau osatzen duen azpiatala:

- **Fakturazio kontzeptuak:** Azpi atal honetan fakturazio-tarifekin erlazionatu daitezkeen fakturazio-kontzeptuen kudeaketa egongo da.

### Tarifak

Menu atal honek fakturazio tarifak kudeatzeko sarbidea izango du. Hona hemen atal hau osatzen duten azpiatalak:

- **Fakturazio tarifak:** Fakturazio tarifak kudeatzeko funtzionalitatea.
- **Tarifen elkartzea:** Tarifak jabeekin edo artikuluekin elkartzeko eta ondoren haiek kudeatzeko funtzionalitatea.

### Fakturaziorako bideak

Menu atal honek fakturaziorako bideak kudeatuko dituzten funtzionalitatera sarbidea emango digu. Hona hemen atal hau osatzen duten azpiatalen zerrenda:

- **Eskuzko fakturazio kontzeptuak:** Eskuzko fakturazio kontzeptuak kudeatu.
- **Fakturazio biderako jaurtiketa:** Manipulazio edo Biltegitratzeko fakturazio biderako jaurtiketa kudeatu.
- **Fakturazio biderako azterketa:** Jaurtitako Manipulazio edo Biltegitratzeko fakturazio bideen azterketa kudeatu.
- **Fakturazio biderako desegitea:** Jaurtitako Manipulazio edo Biltegitratzeko fakturazio bideen desegitea kudeatu.

### Fakturak

Menu atal honek Fakturen sorrera, bilatzea, ezabatzea eta aldaketa kudeatzeko sarbidea emango digu. Hona emen atal hau osatzen duten azpiatalen zerrenda:

- **Fakturen sortzea:** Fakturak sortzeko funtzionalitatea. Bi aukera egongo dira fakturak sortzeko:
  - *Automatikoa:* Fakturak modu automatikoan sortu.
  - *Eskuzko sortzea:* Fakturak eskuz sortu momentuan sortutako fakturazio kontzeptuekin.
- **Fakturen mantentzea:** Fakturen sortzea atalean sortutako fakturen kudeaketa. Hauek bilatu, eguneratu edo ezabatu ahal izango dira.
- **Fakturen ixtea:** Fakturak ixteko prozesua kudeatu. Behin Faktura bat itxita dagoela, ezin da horren gainean inolako aldaketarik egin.
- **Fakturen inprimatzea:** Aukeratutako faktura bakoitzeko .PDF fitxategi bat sortuko du prozesu honek. Bi Faktura moten artean aukeratu ahal izango da: laburtutako faktura eta zehaztutako faktura.



## ANALISIA & DISEINUA

Lanaren garapena nola antolatu den azalduko da atal honetan. Horretarako zati desberdinak azalduko dira: domeinuaren eredua, klase diagrama eta datu-basearen erlazio diagrama.

### Garapenerako aukeratutako lengoaiak

Lanaren antolaketa eta diseinuarekin hasi aurretik, proiektua garatzeko lengoaiak zergatik aukeratu diren azalpen txiki bat emango da.

Proiektua aurrera eraman ahal izateko hasita zegoen beste proiektu batera moldatu behar izan da. Garatuta dagoen aplikazio baten hedapena egin nahi denez, aplikazio horrek erabiltzen dituen baliabideetara moldatu behar izan da atal berri honen garapena.

Aurreko ataletan<sup>2</sup> azaldu da aplikazioak duen arkitektura eta arkitektura hori osatzen duten osagaiak ere bai.

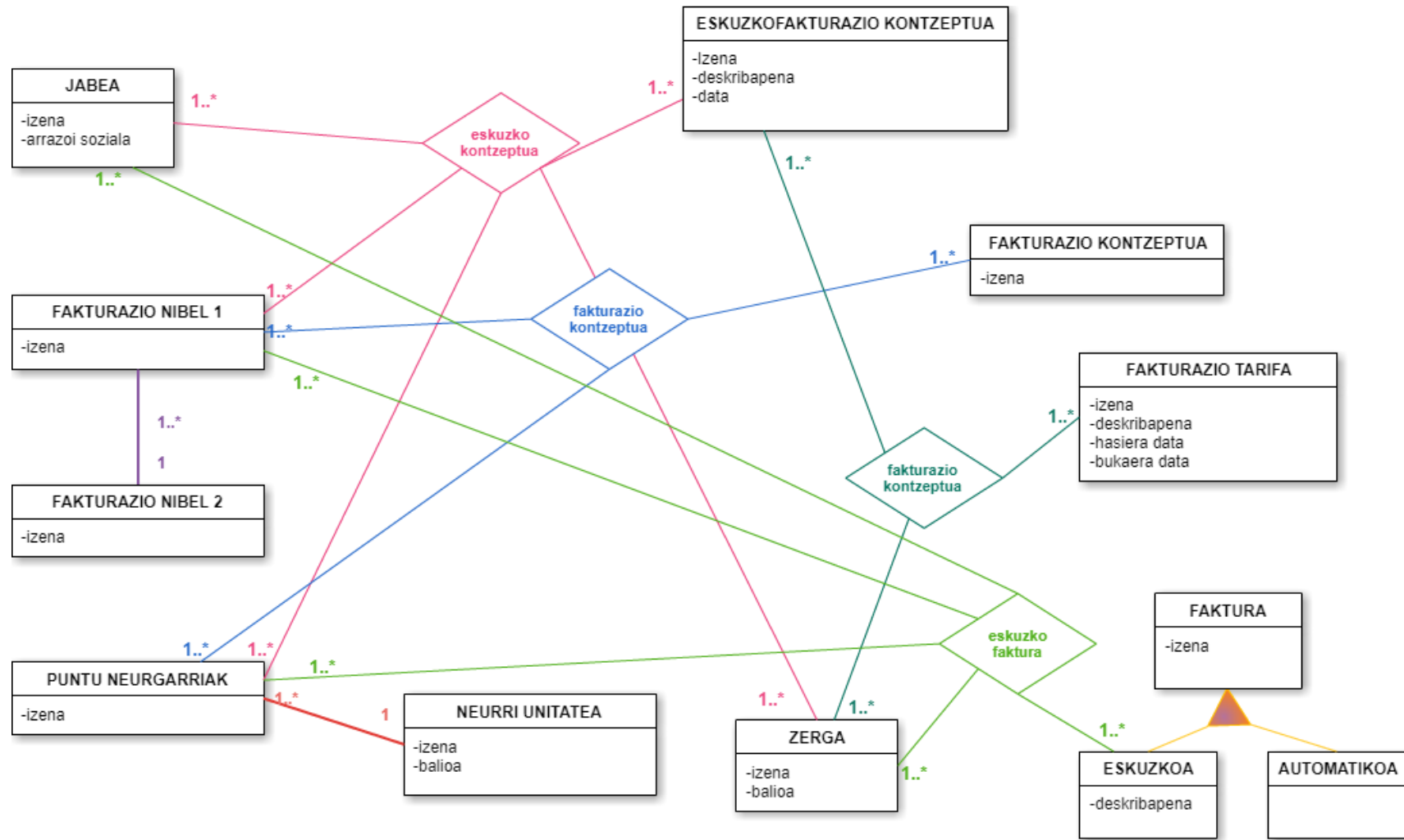
### Domeinuaren eredua

14. irudian proiektua aurrera eraman ahal izateko nola antolatu diren erabilitako datuen Domeinuaren Eredua agertuko da eta ondoren entitate bakoitza zertan datzan ikusteko azalpen txiki bat emango da.

---

<sup>2</sup> Arkitektura atalean aplikazioak erabiltzen dituen baliabideak azaldu dira





Irudia 14 - Domeinuaren eredua

## Entitateak

- Jabea: Jabe bakoitzaren informazioa gordeko duen entitatea da.
- Fakturazio Nibel 1: Fakturazio Nibel 1 bakoitzaren informazioa gordeko duen entitatea da.
- Fakturazio Nibel 2: Fakturazio Nibel 2 bakoitzaren informazioa gordeko duen entitatea da.
- Puntu Neurgarriak: Puntu Neurgarri bakoitzaren informazioa gordeko duen entitatea da.
- Neurri Unitatea: Neurri Unitate bakoitzaren informazioa gordeko duen entitatea da.
- Zerga: Zerga bakoitzaren informazioa gordeko duen entitatea da.
- Eskuzko Fakturazio Kontzeptua: Eskuzko Fakturazio bakoitzaren informazioa gordeko duen entitatea da.
- Fakturazio Kontzeptua: Fakturazio Kontzeptu bakoitzaren informazioa gordeko duen entitatea da.
- Fakturazio Tarifa: Fakturazio Tarifa bakoitzaren informazioa gordeko duen entitatea da.
- Faktura: Faktura bakoitzaren informazioa gordeko duen entitatea da. Faktura mota bi sortuko dira: Eskuzkoa eta Automatikoa. Biak Fakturak duen informazioa izango dute baina atributu batzuk gehituz.

## Klase diagrama

Klaseen arteko erlazioa azalduko da orain. Erlazio hauek hobeto ulertzeko ikus 17. irudia.

Fakturatze-modulua pantailaz dago osatuta, pantaila bakoitza aplikazio zati honek duen funtzionalitatea izango da. Pantaila bakoitzerako sarbidea menu nagusitik atzigarri egongo da 15. irudian ikusten den moduan.

> Maestros | > Tarifas | > Pasos a facturación | > Facturas |

Irudia 15 - Menu

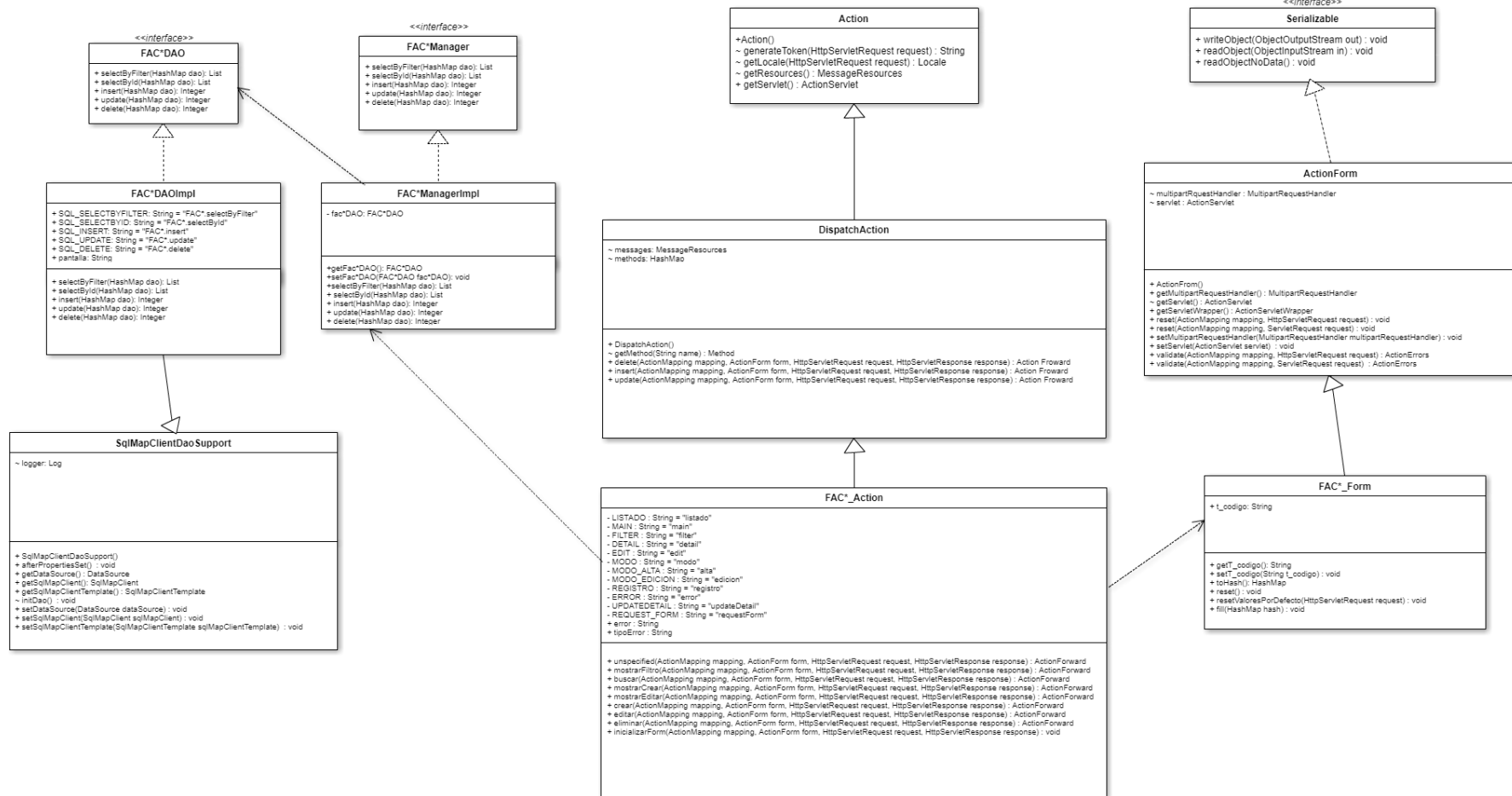
Behin funtzionalitate (pantaila izena) bat aukeratu dugula, pantaila bat ikusiko dugu; 16. irudian pantaila baten prototipoa agertzen da, pantaila batek izango duen itxura hobeto ulertzeko.

The screenshot shows a web application interface. At the top, there is a navigation menu with the following items: > Maestros | > Tarifas | > Pasos a facturación | > Facturas |. Below the menu, there is a search form with two input fields: 'Propietario:' with the value 'Prop1' and 'Tipo' with a dropdown menu showing 'Tipo 1'. Below the input fields are three buttons: 'Buscar', 'Crear', and 'Limpiar'. Below the buttons is a section labeled 'Resultados:' containing a table with three columns: 'Propietario', 'Denominación', and 'Tipo'. The table contains three rows of data:

Propietario	Denominación	Tipo
Sample 1	Sample 1	Sample 1
Sample 2	Sample 2	Sample 2
Sample 3	Sample 3	Sample 3

Irudia 16 - Pantaila prototipoa

Pantaila bakoitzak klase hauek guztiak izango ditu haren funtzionamendua egokia izateko. Hobeto ulertu ahal izateko Klase Diagrama orokor bat azalduko da, esan bezala, pantaila guztiek klase hauek izango dituzte salbuespen batzuk kenduta, orokorrean pantaila guztiek funtzionalitate berdinak izango dituzte. Klase diagrama hobeto ikusi ahal izateko honako URL – a eskuragarri dago: <https://goo.gl/JHNSZ9>



Orain klase bakoitza zehatz-mehatz zertarako erabiltzen den azalduko da:

- **Serializable:** Informazioa *byte* –tan bihurtzea posible izateko, interfaze hau implementatu beharko du klaseak.
- **ActionForm:** ActionMapping batekin edo gehiagorekin elkartuta dagoen *JavaBean*<sup>3</sup> bat da.
- **FAC\*\_Form:** Pantaila bat osatuko duten osagaiak definituko dira hemen. Datu horien aldaketak edo haiek bistaratzea kudeatzeko Java klase hau erabiliko da.
- **FAC\*Manager:** Alde logiko eta grafikoaren arteko bitartekari gisa lan egingo duen interfazea.
- **FAC\*ManagerImpl.java:** Alde logiko eta grafikoaren arteko bitartekari gisa lan egingo duen klasea, FAC\*Manager interfazeak dituen metodoak izango ditu eta horien inplementazioa klase honetan egongo da.
- **FAC\*DAO:** Datu basearekin elkarrekintza kudeatuko duen interfazea.
- **FAC\*DAOImpl:** Datu basearekin elkarrekintza kudeatuko duen klasea, FAC\*DAO.java interfazeak dituen metodoak izango ditu eta horien inplementazioa klase honetan egongo da.
- **Action:** http eskaera eta logikaren arteko eskaera bat prozesatzeko egokitzailea da. Kontrolatzaileak (ActionServlet) eskaera bakoitzeko Action egokia aukeratuko du. Adibidez, beharrezkoa balitz instantzia berri bat eratuko luke.
- **DispatchAction:** Ekintza batekin erlazionatuta dauden funtzioak elkartuko dituen mekanismoa. Modu honetan ez da beharrezkoa izango ekintza bakoitzeko funtzio bat eratzea.
- **FAC\*\_Action.java:** Bere izenak esaten duen moduan , pantailen ekintzak implementatzeko dagoen klasea izango da. Alde grafikotik (adibidez botoi bat sakatu ondoren) logikora pausua ematean, lehenengo kudeatzailea izango da.
- **SqlMapClientDaoSupport:** iBatis liburutegiaren klasea. Datuen atzipena beharko duten objektuek heredatuko dute.

## Klase garrantzitsuak

Atal honetan klase garrantzitsuenen azalpen zehatzago bat emango da. Lehenago azaldu direnak garrantzitsuak dira baita ere baina lehen aipatutako klaseak orokorragoak dira.

## Form

Formulario gisa erabiliko den klasea. Lehen azaldu den moduan klase honetan pantaila batek izango dituen eremu guztiak agertuko dira. Ondoren datuen kudeaketarako askoz errazagoa izango da informazioa modu honetan egotea.

---

<sup>3</sup> Java aplikazioak eratzeko osagai eredu bat dira *JavaBeans* –ak.

Pantailan agertuko den eremu bakoitza atributu bezala definituko da. Atributu hauek *String* motakoak izango dira. Hona hemen Form klaseak dituen metodoen azalpena:

- get\*(): Definitu den atributu bakoitzeko bere *get* metodoa ere inplementatuko da.
- set\*(): Definitu den atributu bakoitzeko bere *set* metodoa inplementatuko da.
- toHash(): Aurretik definitu diren atributu guztiak *HashMap* batean gordeko ditu. Hau horrela eginda askoz errazagoa izango da pantaila bateko eremuekin eragiketak egitea.
- reset(): Atributu bakoitzeko *set* metodoa erabilita, atributu guztiak “husten” ditu.
- resetValoresPorDefecto(): Atributuak lehenetsitako balioekin hasieratzen ditu.
- fill(): Metodoari bidaltzen zaion *HashMap* parametroa begiratuko du. Parametro horretan atributu bakoitzaren balioa dagoen edo ez konprobatuko du. Balioa egonez gero, *set* metodoa erabiliz Form klaseko atributuari esleituko dio.

## DAO

Datu-basearekin harremanetan ipintzen den klasea izango da hau. Hau da, klase hau erabilita datu basearen kontra eragiketak bidali daitezke edo datu basean beran egin behar diren prozesuen jaurtiketa exekutatu daiteke baita ere. Metodoen azalpena eman aurretik esan beharra dago orokorrean klase honek dituen metodoak azalduko direla, baliteke pantaila bateko beste funtzionalitatearen bat izatea baina datu-basearen kontrako edozein prozesu izanez gero, klase honetan definitu beharra dago. Hona hemen DAO klaseak dituen metodoen azalpena:

- selectByFilter(): Datu basearen kontra kontsulta bat bidaliko du. Kontsultaren erantzuna *List* batean bueltatuz.
- selectById(): Datu basearen kontra kontsulta bat bidaliko du. Kontsultaren erantzuna *List* batean bueltatuz.
- insert(): Erregistro baten sorrera exekutatu du metodo honek.
- update(): Aurretik existitzen den erregistro baten eguneraketa exekutatu du metodo honek.
- delete(): Aurretik existitzen den erregistro baten ezabaketa exekutatu du metodo honek.

## Manager

Lehen azaldu den moduan, Manager klasea alde logiko eta grafikoaren arteko bitartekari gisa lan egingo duen klasea da. Pantailari dagokion DAO –ak izango dituen metodo berdinak izango ditu eta DAO klasearen azalpena ematean esan den bezala, baliteke pantaila batzuk beste funtzionalitatearen bat izatea. Hau gertatzekotan hemen ere funtzionalitate horien metodoa garatu beharko litzateke. Hona hemen haien azalpena:

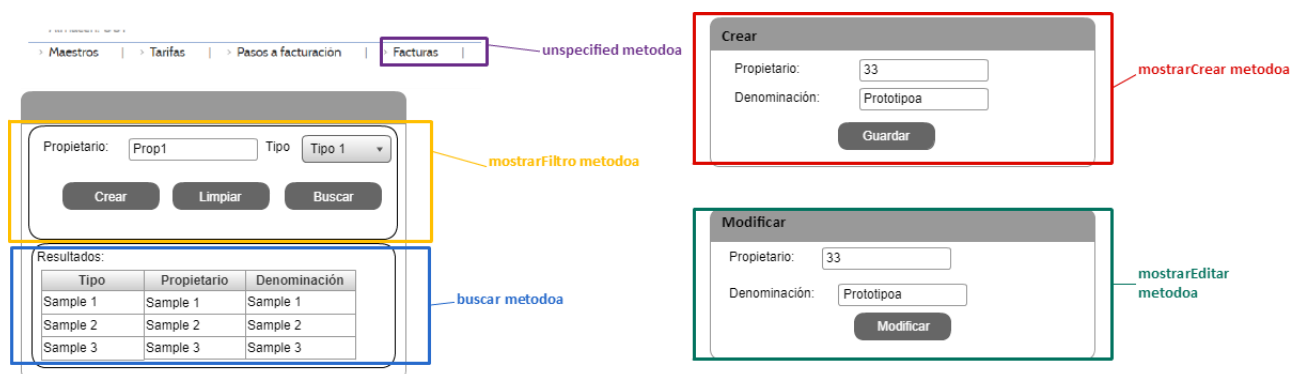
- getFAC\*DAO(): Managerrari dagokion DAO –a bueltatuko du beste metodoetan exekutatu ahal izateko.
- setFAC\*DAO(): DAO bat parametro gisa bidaliz, Manager klasean atributu bezala definitutako DAO –ari pasatuko zaio.
- selectByFilter(): Datu basearen kontra kontsulta bat bidaliko du. Kontsultaren erantzuna *List* batean bueltatuz.

- selectById(): Datu basearen kontra kontsulta bat bidaliko du. Kontsultaren erantzuna *List* batean bueltatuz.
- insert(): Erregistro baten sorrera exekutatu du metodo honek.
- update(): Aurretik existitzen den erregistro baten eguneraketa exekutatu du metodo honek.
- delete(): Aurretik existitzen den erregistro baten ezabaketa exekutatu du metodo honek.

### Action

Funtzionalitateak aurrera eraman ahal izateko lehenik eta behin pantailan (alde grafikoa) ekintza bat sortu beharko da. Hobeto ulertzeko, klasearen izenak esaten duen moduan ekintzak definituko dira bertan. Hau guztia kudeatu ahal izateko Action klaseak egongo dira; pantaila bakoitzak bere Action –a izango du.

Beherago azalduko diren metodoak modu argiago batean ulertu ahal izateko begiratu 18 . irudia pantaila baten geruzak edo atalak zeintzuk izango diren ulertzeko.



Irudia 18 - Action klaseko geruza nagusiak

Hona hemen klase honek izango dituen metodoak:

- unspecified(): Pantaila bat betetzeko (aurrerago azalduko da zer den pantaila bat "betetzea") erabiliko den metodoa. Pantaila batera sartu nahi izatean metodo hau exekutatu da zein bide "jarraitu" behar den adieraziz.
- mostrarFiltro(): Pantaila guztiek, edo ia guztiek, exekutatu nahi den bilaketa filtratzeko aukera izango dute. Metodo honek pantailako geruza hori beteko du.
- buscar(): Bilaketa exekutatu du metodo honek eta emaitzak multi erregistro batean bistaratuko ditu.
- mostrarCrear(): Pantaila guztiek, edo ia guztiek, elementu berri bat gehitzeko aukera izango dute. Metodo honek pantailako geruza hori beteko du.
- mostrarEditar(): Pantaila guztiek, edo ia guztiek, elementu bat eguneratzeko aukera izango dute. Metodo honek pantailako geruza hori beteko du.
- crear(): Datu basean erregistro berri baten sorrera exekutatu du metodo honek.
- editar(): Datu baseko erregistro baten aldaketak gordetzea exekutatu du metodo honek.

- eliminar(): Datu basean dagoen erregistro baten ezabaketa exekutatu du metodo honek.
- inicializarForm(): Parametro bezala bidaltzen zaion formularioa (Form klasekoa) hasieratu du Form klaseak duen *reset* metodoa erabiliz.



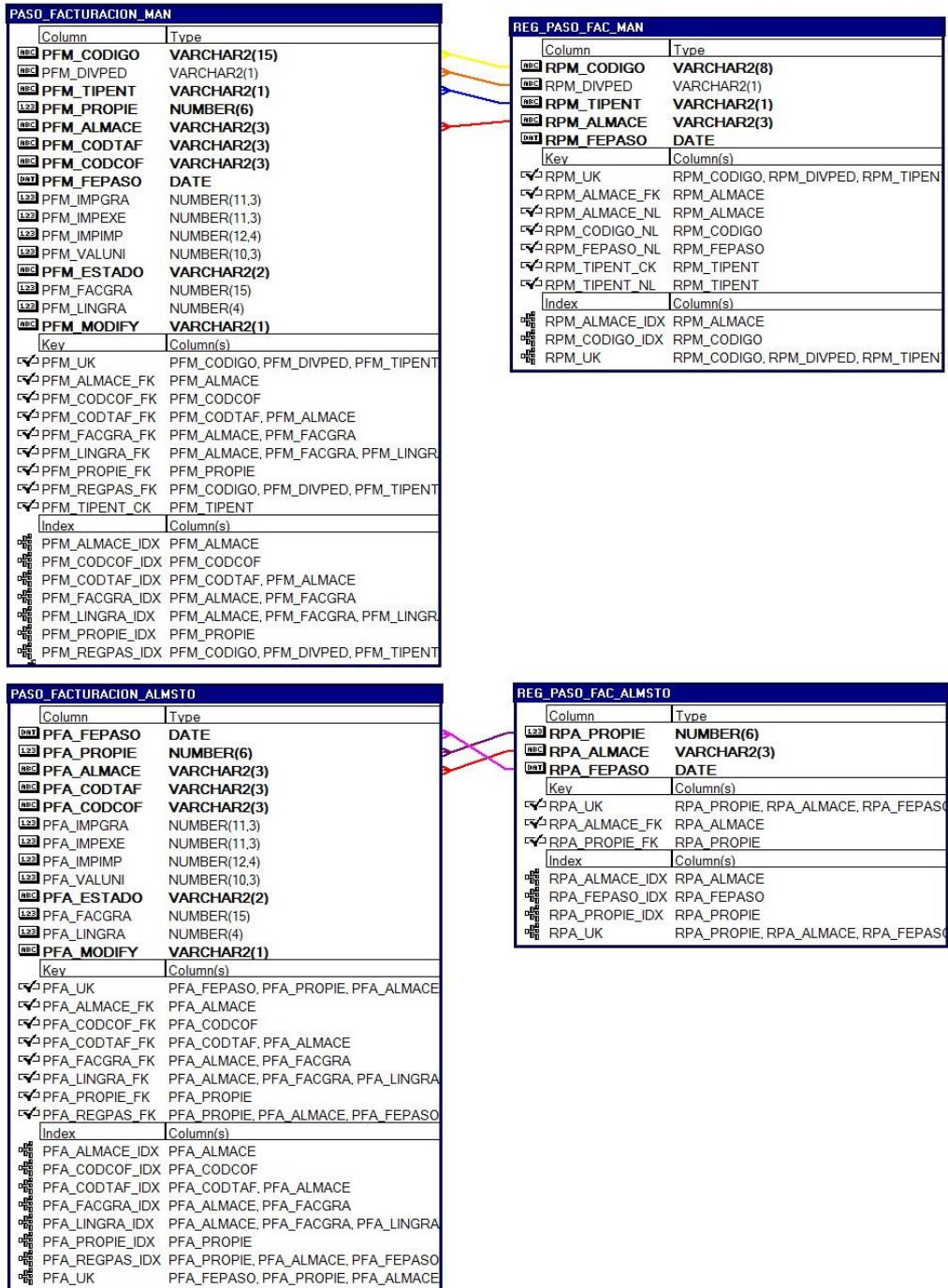
## Datu-basearen entitate/erlazio diagrama

Datu-basearen erlazio diagrama azalduko da orain. Datu-basea osatzen duten entitateak eta haien arteko erlazioen azalpena emango da.

19. irudia PL SQL Developer –etik esportatutako irudia da. Irudi horretan entitate erlazio diagrama erakusten da, baina ez denez oso argi ikusten diagrama askoz hobeto ikusteko eta ulertzeko honako URL hau eskuragarri dago: <https://goo.gl/JCLBnT>

20. irudian diagramaren beste zati bat agertzen da. Bertan Fakturazio bidea (Manipulazioko zein Biltegitratzekoa) prozesua exekutatu ondoren datuak gordeko dituzten taulak agertzen dira. Taula hauek beste diagramatik bananduta agertzeko arrazoia oso sinplea da; entitate guztiak elkarrekin ipiniz gero, ez litzateke ezer ulertuko. . irudian ez da erlazioirik agertzen baina bai *Primary Key* –ak, bai *Foreign Key* –ak agertzen dira. Entitateen bigarren diagrama hau hobeto ikusteko eta ulertzeko beste URL hau eskuragarri dago: <https://goo.gl/KJmGFm>

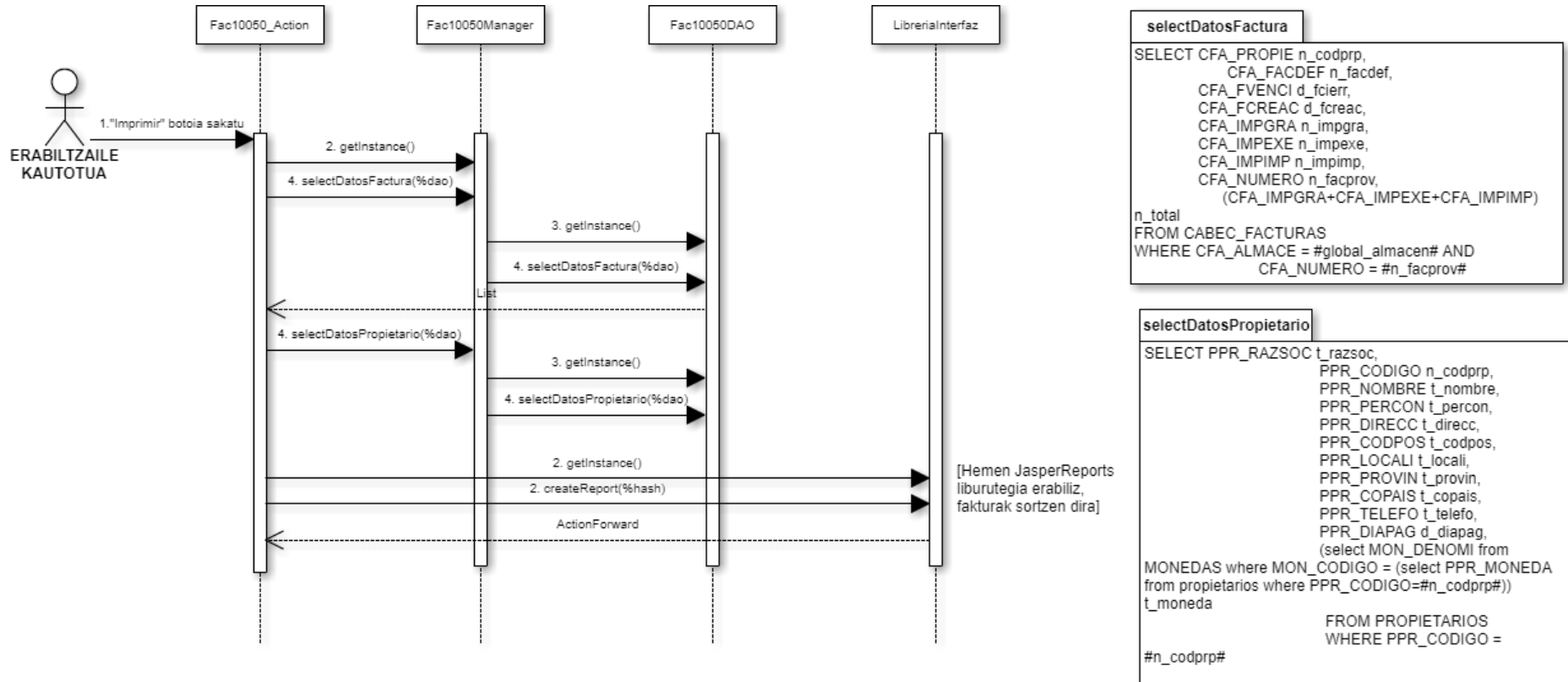




Entitate bakoitzaren azalpena hobeto ulertzeko taula bakoitzaren gaztelaniazko izena agertuko da itzultitako izenaren alboan.

- **Jabeak** (*PROPIETARIOS*): Jabeen informazioa gordeko duen taula.
- **Zergak** (*IMPUESTOS*): Zergen informazioa gordeko duen taula.
- **Artikuluak** (*ARTICULOS*): Artikuluen informazioa gordeko duen taula.
- **Biltegiak** (*ALMACENES*): Biltegien informazioa gordeko duen taula.
- **Puntu Neurgarriak** (*PUNTOS\_MEDIBLES*): Puntu neurgarrien informazioa gordeko duen taula. Puntu neurgarri bakoitzak Neurri unitate bat izango du.
- **Neurri unitateak** (*UNIDADES\_MEDIDA*): Neurri unitateen informazioa gordeko duen taula.
- **Fakturazio tarifa motak** (*TIPOS\_TARIFA\_FAC*): Fakturazio tarifa moten informazioa gordeko duen taula.
- **Fakturazio tarifa azpi motak** (*SUBTIPOS\_TARIFA\_FAC*): Fakturazio tarifa azpi moten informazioa gordeko duen taula. Fakturazio tarifa mota bat izango du azpi mota bakoitzak.
- **Fakturazio kontzeptuak** (*CONCEPTOS\_FACTURACION*): Fakturazio kontzeptuen informazioa gordeko duen taula. Kontzeptu bakoitzak fakturazio tarifa mota eta azpimota bat eta puntu neurgarri bat izango ditu.
- **Eskuzko Fakturazio kontzeptuak** (*CONCEPTOS\_FAC\_MANUALES*): Eskuzko Fakturazio Kontzeptuen informazioa gordeko duen taula. Eskuzko kontzeptu bakoitzak biltegi bat, zerga bat, fakturazio tarifa mota eta azpimota bat eta jabe bat izango ditu.
- **Fakturazio Tarifak** (*TARIFAS\_FACTURACION*): Fakturazio Tarifen informazioa gordeko duen taula. Tarifa bakoitzak biltegi bat eta fakturazio tarifa mota eta azpimota bat izango ditu.
- **Jabeen Tarifak** (*TARIFAS\_PROPIE*): Taula honetan tarifen eta jabeen arteko erlazioak gordeko dira. Erlazio hau osatzeko jabe bat, tarifa bat eta biltegi bat behar dira.
- **Artikuluaren Tarifak** (*TARIFAS\_ARTICU*): Taula honetan tarifen eta artikuluaren arteko erlazioak egongo dira. Erlazio bakoitzak artikulu bat, tarifa bat, jabe bat eta biltegi bat beharko du.
- **Faktura burua** (*CABEC\_FACTURAS*): Faktura baten burua gordeko duen taula. Faktura buru bakoitzeko biltegi bat, jabe bat egongo da. Hemen zein tarifa mota den ere adierazi beharko da (Automatikoa edo Eskuzkoa)
- **Faktura lerroak** (*LINEAS\_FACTURAS*): Faktura batek izango dituen lerroak gordeko dituen taula. Lerro bakoitza faktura buru batekin elkartuta egongo da. Lerro bat osatzeko faktura zenbaki bat eta fakturazio tarifa mota eta azpi mota bat behar dira.
- **Manipulazio fakturazio biderako erregistroa** (*REG\_PASO\_FAC\_MAN*): Jaurti ahal daitezkeen pasabideak gordeko dituen taula.
- **Biltegiratze fakturazio biderako erregistroa** (*REG\_PASO\_FAC\_ALMSTO*): Jaurti ahal daitezkeen pasabideak gordeko dituen taula.
- **Manipulaziozko fakturazio bidea** (*PASO\_FACTURACION\_MAN*): Jaurti diren Manipulazio fakturazio bideak gordeko dituen taula.
- **Biltegiratze fakturazio bidea** (*PASO\_FACTURACION\_ALMSTO*): Jaurti diren Biltegiratze fakturazio bideak gordeko dituen taula.

Prozesu esanguratsu baten sekuentzia diagrama: Fakturen inprimatzea



Irudia 21 - Fakturen inprimatzea (sekuentzia diagrama)





## GARAPENA

Dokumentazio osoan zehar azaldu den moduan, aurretik eraikita zegoen aplikazio bati funtzionaltasun gehiago gehitu zaizkio. Kasu honetan, funtzionaltasun horiek guztiak aplikazioaren modulu berri bat definituz gehituko dira.

Orain arte aplikazioa bi moduluk osatzen zuten:

1. Biltegiratze moduluak: Biltegi baten kudeaketa burutzen du. Modulu hau erabiltzaile batzuentzat atzitura egongo da. Sarrera eta irteera eskaeren kudeaketa, biltegi bakoitzeko inbentarioaren kudeaketa edo bezeroen kudeaketa aurrera eramateko erabiltzen da, besteak beste.
2. Segurtasun moduluak: Aplikazio beraren administrazio moduluak da. Atal hau erabilia erabiltzaile berriak sortu daitezke eta erabiltzaile horiek izango dituzten abantailak ezarriko dira; bakoitzak behar dituenak. Abantaila horien artean, adibidez, erabiltzaile batek eskuragarri izango dituen moduluak egongo dira. Baliteke 1 erabiltzaileak Biltegiratze eta Segurtasun moduluak eskuragarri izatea baina 2 erabiltzaileak bakarrik Biltegiratze moduluak eskuragarri izan dezake.

Orain arte aplikazioan "eraikita" zegoen guztia azaldu eta gero, proiektu honetan garatu diren funtzionalitate berriak azalduko dira eta hasteko modulu berria nola eraiki den azalduko da.

Fakturatze-moduluak aurretik aplikazioan zeuden atalen egitura jarraituko du, esan beharra dago Biltegiratze eta Segurtasun moduluak bananduta daudela inplementazioan, Segurtasun moduluak gauza gehiago kudeatzen dituelako.

Fakturatze atala, berriz, Biltegiratze atalarekin batera ibiliko da baina, zergatik? Hau arrazoi batengatik horrela inplementatzea erabaki zen; fakturazio-moduluak ez dauka zentzurik Biltegiratze-moduluak barik, modulu berri honek Biltegiratze-moduluak dituen hainbat funtzionalitate eta hainbat datu-egitura berdin izango dituelako eta Biltegiratze-moduluak kudeatzen dituen datu batzuk atal berri honen funtzionamendua egokia izan ahal izateko guztiz beharrezkoak bai dira.

Orain Fakturatze-moduluan sakon-sakon sartuko gara, honek dituen funtzionaltasun edo aukera guztiak xehetasun osoz azaltzeko.

## Fakturatze-moduluak sakonean

Garatutako proiektu hau aplikazioan gehituko den atal berri honen lehenengo atal bat da. Hasierako atal hau Fase 0 gisa izendatu da. Bezeroak inplementazio berri honek dituen oinarriko funtzionalitateak dituen fakturazio atal bat behar zuelako egin da.

Hasierako fase hau 16 pantailak osatzen dute, baina zer ulertzen dugu pantaila gisa? Orokorrean, pantaila menu nagusitik edo azpi-menu batetik atzigarri dugun aplikazioaren funtzionalitate bat izango da. Funtzionalitate bakoitzak aplikazioan efektu bat edo beste bat sortuko du, logika jarraituz, pantaila bakoitzak funtzionalitate bat izango du.

Orokorrean funtzionalitate horien artean erregistroen bilatzea, sortzea, eguneratzea edo ezabatzea egongo da; arrazoi honengatik sailkapen bat egingo da azalpena ez luzatzeko eta batez ere modu argiago eta erraz batean ikusteko.

Aplikazio honetan bi pantaila mota daude:

- **Pantaila nagusiak:** Menu nagusitik atzigarri dauden funtzionalitateak dira. Beherago azalduko da zeintzuk diren funtzionalitate horiek.
- **LOV (List of Values):** Bere izenak esaten duen moduan balio batzuen zerrenda bistaratzean datza. Pantaila arrunt bat inplementatzerako orduan jarraitu behar diren pausu berdinak eman behar dira LOV baten garapenerako. Menu nagusitik ez daude atzigarri, aldiz, LOV pantailak pantaila batetik atzigarri egongo da. Pantaila zehatz batek duen eremu baten balioen zerrenda bistartzeko erabiliko da. Beste ezaugarri esanguratsu bat da *popup* gisa bistaratzen direla. 22. irudian adibide bat erakusten da zertan datzan hobeto ulertzeko:

The screenshot shows the In-LOG WMS interface. On the left, the 'CREACIÓN' section is visible with fields for 'Concepto Facturación' (AAA), 'Denomi. Concepto', 'Nivel 1\*' (02), 'Nivel 2\*', 'Punto Medible\*', and 'Unidad Medida'. A search icon is highlighted on the 'Nivel 1\*' field. A popup window titled 'Nivel 2 (FAC00200lov)' is open, showing a search filter and a table of results.

**FILTRO DE BÚSQUEDA**

Nivel 2  Denominación

Buscar Limpiar Cancelar

**RESULTADO DE CONSULTA**

Encontrados 3 datos.

Nivel 2	Denominación	Nivel 1
01	MANIPULACION ENVIOS	02
02	MANIPULACION LLEGADAS	02
03	MANIPULACION DEVOLUCIONES	02

Irudia 22 - LOV popup

22. irudian dagoen lauki gorriaren barruan botoi bat ikusten da. Botoi bertan sakatuz gero popup -a irekiko da.



## Funtzionalitateetan sakonduz

Sailkapenarekin hasteko pantaila guztiak zerrendatuko dira (bakarrik haien izena idatziz, bakoitzaren eginkizuna ondoren azalduko da) eta bakoitzak duen funtzionalitatea bere alboan azalduko da 13. taulan:

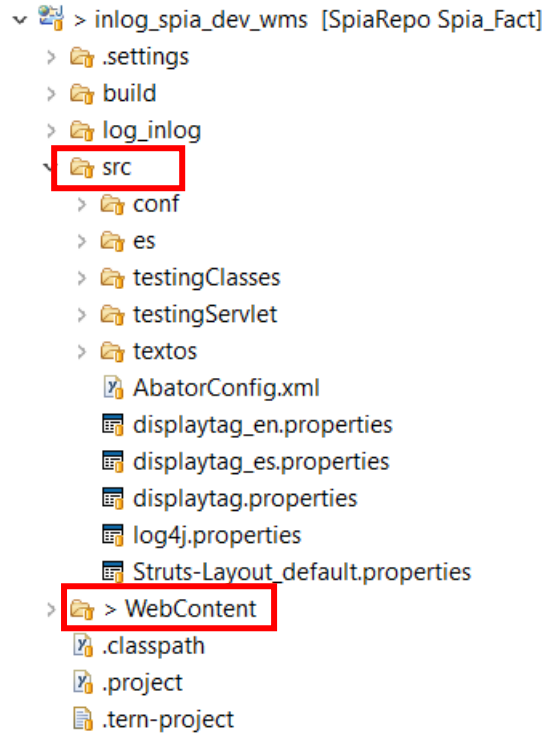
PANTAILA IZENA	ERREGISTROEKIN ERAGIKETAK				BESTE FUNTZIONALITATEA
	BILATZEA	SORRERA	EGUNERATZEA	EZABATZEA	
Fakturazio kontzeptuak	✓	✓	✓	✓	x
Fakturazio tarifen mantentzea	✓	✓	✓	✓	x
Jabeenganako tarifen elkartzea	✓	✓	✓	✓	x
Artikulu/tarifa elkartzea	✓	✓	✓	✓	x
Eskuzko fakturazio kontzeptuak	✓	✓	✓	✓	x
Manipulazioko fakturazio biderako jaurtiketa	✓	x	x	x	✓
Biltegiratzeko fakturazio biderako jaurtiketa	✓	x	x	x	✓
Manipulazioaren fakturazio bideko azterketa	✓	x	x	x	✓
Biltegiratzeko fakturazio bideko azterketa	✓	x	x	x	✓
Manipulazioaren fakturazio bideko desagitea	✓	x	x	x	✓
Biltegiratzeko fakturazio bideko desagitea	✓	x	x	x	✓
Fakturen sortze automatizatua	x	✓	x	x	✓
Fakturen eskuzko sortzea	✓	✓	x	x	✓

<b>Fakturen mantentzea</b>	✓	x	x	✓	x
<b>Fakturen ixtea</b>	✓	x	x	x	✓
<b>Fakturen inprimatzea</b>	✓	x	x	x	✓

Taula 13 - Prozesuen funtzionalitatea

## Proiektuaren antolakuntza

Orain garapena aurrera eramateko proiektua nola antolatu den azalduko da. Hau modu argiago batean ulertzeko, kodea nola antolatu den erakusten da 23. irudian. Zuhaitza osorik azalduko da, dauzkan azpiataletan sakonduz.

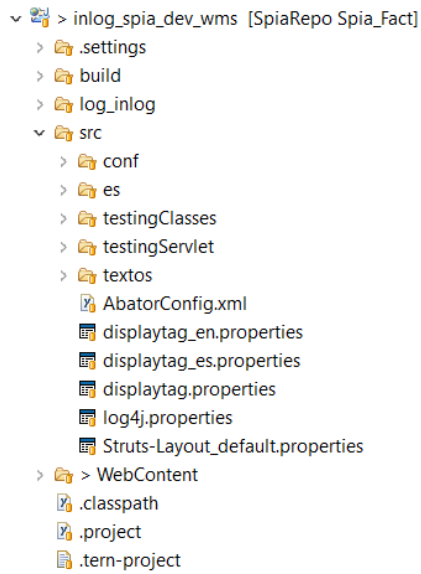


*Irudia 23 - Proiektuaren antolaketa*

## SRC

SRC –n aplikazioaren alde logikoa eta logiko eta grafikoaren arteko tartekariak egongo dira. Hau da, datu basearekin egiten diren konexioak edo eskatzen zaizkion datuak bertan kudeatuko dira edo pantaila bateko botoi bat sakatuz gero zer gertatuko den hemen kudeatuko du baita ere.

Orain proiektuaren alde honetan dauden aplikazioaren osagaiak eta azpi osagaiak azalduko dira. Hobeto ikusteko ikus 24. irudia.



*Irudia 24 - src osagaiak*

Gorago esan den moduan atal hau alde logikoa denez, hemen aplikazioaren alde garrantzitsuenak implementatu dira. Hemen garatu diren klaseen funtzionalitatea aurreko ataletan<sup>4</sup> azaldu da. Atal honetan azpi atal garrantzitsuenak *es* eta *textos* izango dira, gainontzekoak frogak exekutatzeko klaseak implementatzeko erabili bai dira.

*.properties* fitxategiak alde grafikoan erabili diren liburutegietan aldaketa batzuk egiteko erabili diren fitxategiak dira. Adibidez JSTL liburutegiak eskaintzen dituen etiketetaz aparte beste batzuk definitu dira bertan.

Beraz, alde logikoa nola dagoen antolatuta azaltzeko azpi atal hauen azalpena ematen da:

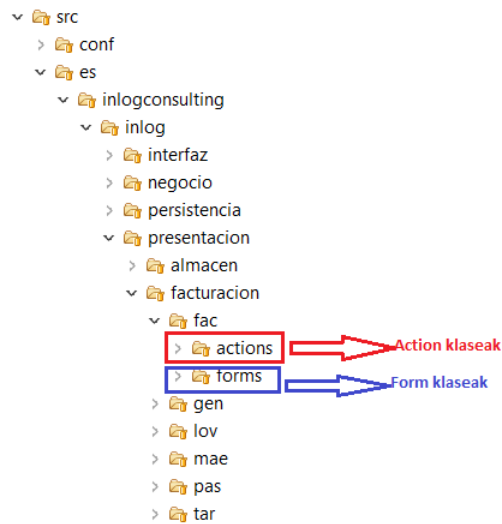
---

<sup>4</sup> Klase bakoitza zertarako erabiltzen den Klase Diagrama azaldu den atalean adierazi da.

es

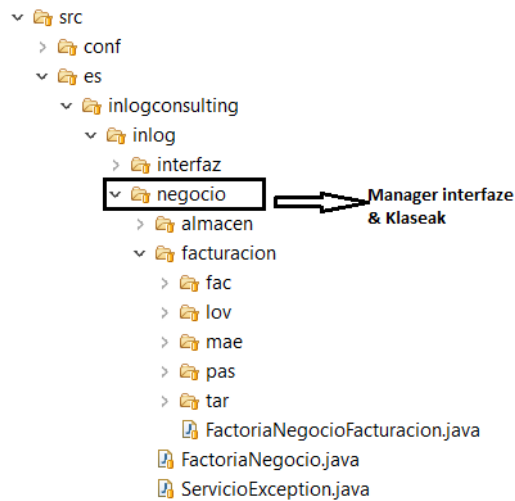
Azpi atal honetan honako klase hauek aurkituko dira eta jarraian dauden irudietan modu argiago baten ikusi daitezke egituraketa:

- Forms
- Action



Irudia 25 - Action & Form

- Managerrak (interfazea eta inplementazioa)



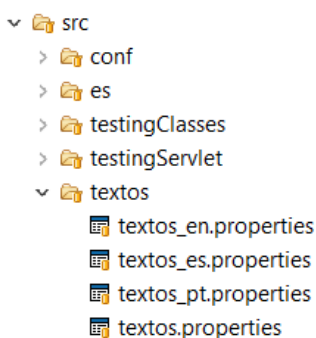
Irudia 26 – Managerrak

- DAO –ak (interfazea eta implementazioa)
- SQL query –ak definituta daude atal honetan. Pantaila bakoitzeko bat.



Irudia 27 - DAO & SQL query

#### textos



Irudia 28 - Textos Properties

Azpi atal honetan aplikazioan erabiltzen diren literalen definizioa dago. Baina, zertarako? Aplikazioa hizkuntza bat baino gehiagotan erabilgarria dagoenez, hizkuntza bakoitzak bere *textos\*.properties* fitxategia izango du. Literalen izena fitxategi guztietan berdina izango da baina bakoitza bere hizkuntzan, 29 eta 30. irudietan ikusten den moduan.

```
app.tit.aplicac3_3 = Informes
app.tit.aplicac3_4 = Mantenimientos
```

Irudia 29 - Literalak gazteleraz

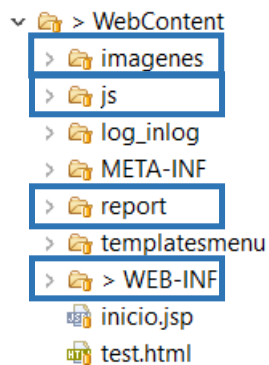
```
app.tit.aplicac3_3 = Reports  
app.tit.aplicac3_4 = Maintenance
```

*Irudia 30 - Literalak ingelesez*

## WebContent

WebContent –en alde grafikoa egongo da. Pantailen estiloak, itxura eta bakoitzak izango dituen osagaiak definituko dira.

Proiektuaren WebContent atalean dauden aplikazioko osagaiak eta azpi osagaiak azalduko dira. Modu argiagoan ulertzeko ikus 31. irudia.



*Irudia 31 - WebContent osagaiak*

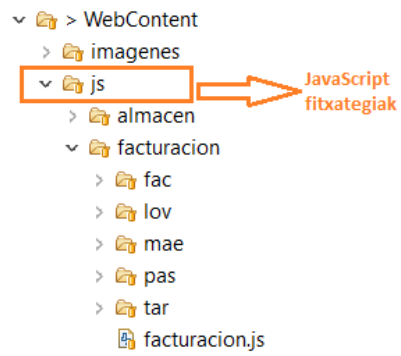
## imagenes

Atal honetan aplikazioan erabili diren irudiak gordeko dira. Adibidez, botoi baten itxura aplikazio osoan berdina izan behar da eta estilo berdina jarraitzeko irudi aurre definituak egongo dira.

## Js

Hemen js fitxategiak daude. Pantaila bakoitzeko bat definituta dago eta web aplikazioari dinamikotasuna emateko erabili da.

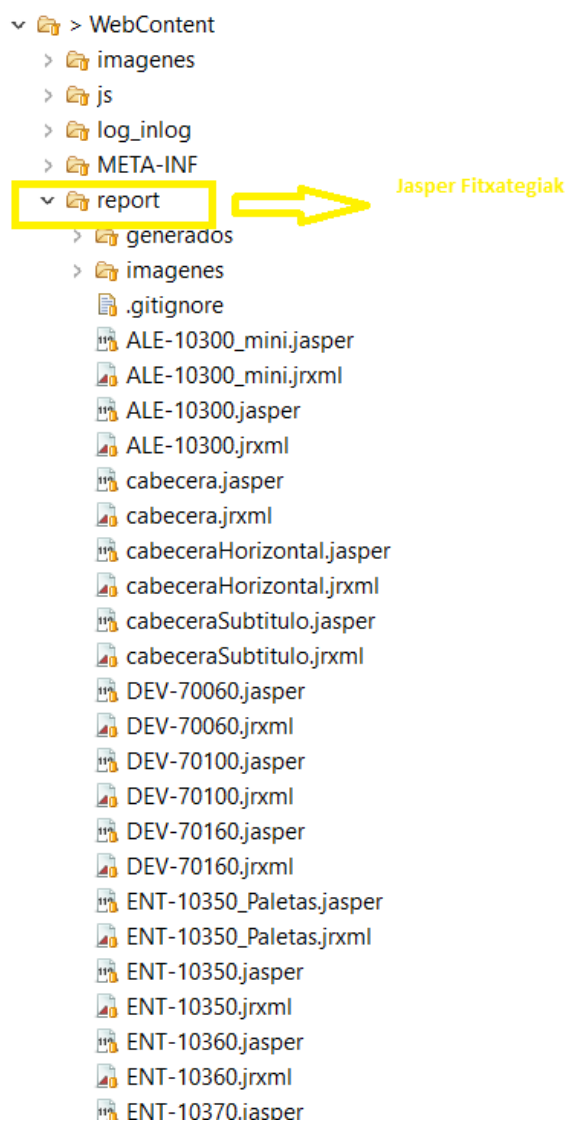




*Irudia 32- JavaScript fitxategiak*

## Report

Jasper Report fitxategi editagarriak hemen izango dira eta hauek konpilatzerakoan lortutako fitxategiak baita ere. Aplikazio osoan dauden *report* -ak hemen daude.

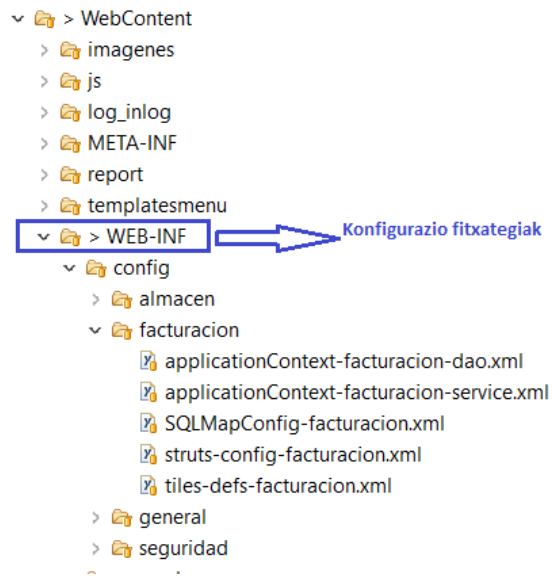


*Irudia 33 - Jasper fitxategiak*

## WEB-INF

Hemengo atal honetan hainbat gauza garrantzitsu daude. Alde batetik konfigurazio fitxategiak, hau da, bezero aldea (alde grafikoa) zerbitzariarekin harremanetan jartzeko aurredefinitu behar diren fitxategiak (edo hau lortzeko jarraitu beharreko pausuak) hemen definitu dira. Eta beste aldetik pantailen itxura (.jsp fitxategiak) hemen zehaztu da, hau da, zein izango den haien itxura, zer eremu izango duten, etab...

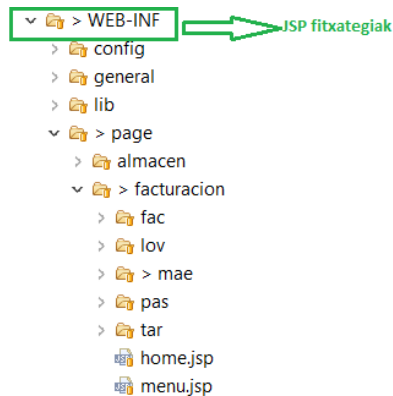
- *config* direktorioaren barruan konfigurazio fitxategiak aurkituko ditugu:



Irudia 34 - Konfigurazio fitxategiak

Aurrerago azalduko da konfigurazio fitxategi bakoitzaren funtzionamendua.

- page direktorioaren barruan .jsp fitxategiak aurkituko ditugu:



Irudia 35 - JSP fitxategiak

### Azpi atalen antolaketa

Azpi atal batzuetan ikusten denez, guztia beste azpi direktorio batzuetan antolatu da. Azpi direktorio hauek pantaila guztiak menu nagusiaren arabera ordenatzeko sortu dira. 36. irudian Fakturazio moduluak duen menu nagusia agertzen da eta azpi direktorioak honekin bat datozela ikusi daiteke 37.irudian:

> Maestros | > Tarifas | > Pasos a facturación | > Facturas |

Irudia 36 - Menu nagusia

- ▼ 📁 facturacion
  - > 📁 fac
  - > 📁 lov
  - > 📁 mae
  - > 📁 pas
  - > 📁 tar

Irudia 37 - Azpi atalen direktorioak

## Fakturatze-modulua: funtzionalitateen azalpena

Funtzionalitateen garapena pantaila guztietan berdina izan da kasu zehatz batzuk kenduta, dena den, kasu horien garapena nola burutu den azalduko da baita ere. Hasteko pantaila bat eraikitzeko bere osagaiak zeintzuk diren azalduko da.

Hiru multzotan banatuko dira osagai hauek; lehenengo alde logikoa izango dugu, ondoren alde grafikoa eta azkenik aplikazioaren eta Datu Basearen arteko konexioa. Alde logikoaren funtzionamenduaren osagaien azalpena<sup>5</sup> lehenago eman da, beraz multzo horretan ez da gehiagorik sakonduko.

Ulergarria izateko 14. taulan zerrendatuko dira pantaila baten osagaiak.

	OSAGAIA	ZEREGINA
<b>ALDE LOGIKOA</b>	DAO ( <i>Data Access Object</i> )	Datu Basearen kontra eragiketak bidali
	Manager	Alde logiko eta grafikoaren arteko bitartekari gisa lan egingo duen klasea da. Ekintza DAO a-rekin konektatu
	Form	Formulario gisa erabiliko den klasea
	Action	Ekintzen garapenaren inplementazioa
<b>ALDE GRAFIKOA</b>	SQL Query (.xml)	Pantaila batek datu basearen kontra exekutatu dituen QUERY –en definizioa
	JavaScript	Pantaila dinamizatzeko .js fitxategia
	JSP	Pantailaren definizio grafikoa

<sup>5</sup> Klase diagrama osatzen duten Java klaseen azalpena emandakoan hain zuzen.

<b>DATU BASEAREKIN KONEXIOA</b>	Package	Datu basean aldaketak eragingo dituen .sql <i>package-a</i>
---	---------	---

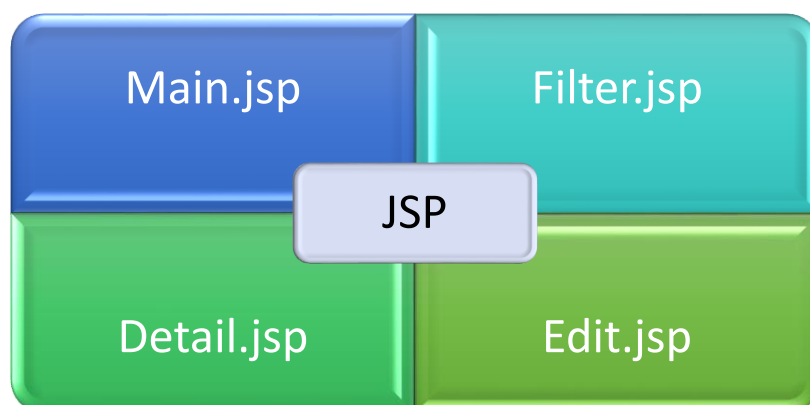
Taula 14 - Pantaila baten osagaiak

Orain pantaila baten alde grafikoa eraikitzeke behar diren atalen azalpena emango da.

## JSP

Goian azaldu den moduan pantaila baten definizio grafikoa osorik fitxategi hauekin definituko da. Hau da, atal honekin pantaila baten egitura definituko da; beraren itxura, zeintzuk izango diren bere osagaiak edo zer eremu izango dituen, adibidez.

Lehenago esan den moduan, orokorrean, pantaila guztien funtzionamendua berdina izango da (bilatzea, gehitzea, eguneratzea eta ezabaketa), horregatik kasu orokor horien alde grafikoa eraikitzeke behar diren azalpena emango da.



Irudia 38 - JSP atalak

38. irudian pantaila baten itxura osatzeko beharrezkoak diren azpi atalak azaltzen dira. Orain haien azalpen sakonago bat emango da.

### Main.jsp

Azpi atal hau ezinbestekoa izango da pantailen definizio guztietan. Hona hemen zeintzuk diren bere betebeharrak:

- Pantaila bakoitzerako “sarbide” gisa jokatuko du.
- .jsp fitxategi honetan pantaila bakoitzak izango dituen geruzak definituko dira eta geruza horietan zein izango den kargatuko den *display* -a.

39 eta 40. irudietan Main azpi atalaren funtzionalitateak modu grafikoan adierazi dira.

```

|<%@ taglib uri="http://struts.apache.org/tags-html"
<%@ taglib uri="http://struts.apache.org/tags-bean"
<%@ taglib uri="http://displaytag.sf.net/el"
<%@ taglib uri="http://inlog.inlogconsulting.es/tags/menu"
    prefix="html" %>
    prefix="bean" %>
    prefix="display"%>
    prefix="menu" %>

```

Erabiliko diren etiketen definizioa eta esleitutako ezizena

Irudia 39 - Main: Tag - en definizioa

```

<div id="filter" style="display:">
<div class="div_fieldset" onclick="gestionarDisplay('contenido_filter', 'img_display_Filter');">
<span class="img_ocultar" title="<bean:message key='sis.comun.verOcultar' />" id="img_display_Filter" >&nbsp;&nbsp;&nbsp;</span>
<span class="titulo_fieldset"><bean:message key='sis.comun.filter.title' /></span>
</div>
<div id="contenido_filter" style="display:block;"></div>
<script type="text/javascript">
<script>
</script>
</div>

```

Pantaila batek izango duen filter atalaren definizioa

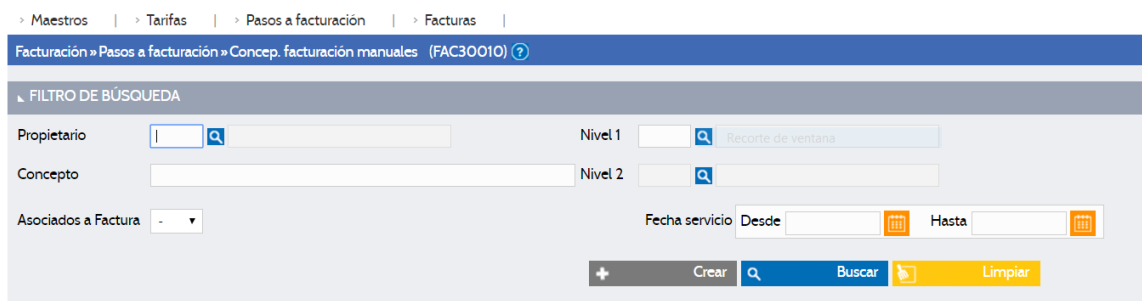
Filter atalaren edukia zein geruzatan erakutsiko den definitzen du

Irudia 40 - Atalak eta geruzak

Atalen definizioa pantaila bakoitzak izango dituen funtzionalitateen arabera izango da. Pantaila batek funtzionalitate guztiak izanez gero (kasu honetan bilatzea, sortzea, eguneratzea eta ezabatzea), beste bi atal definitu beharko lirateke Main.jsp fitxategian.

### Filter.jsp

Pantaila baten iragazpen atalaren alde grafikoa eraikitzeko erabilia 41.irudian ikusten den moduan.



Irudia 41 - Filter: Iragazpen atala pantaila

Pantaila batek erregistroren bilatze aukera izatekotan, orokorrean, emaitzak iragazteko aukera izango du. Beraz, azpi atal honen betebeharrak hauek izango dira:

- Emaitzak iragazteko aukera eman.
- Zeren arabera iragazi ahal diren emaitzak zehaztu.
- Erregistro berri bat sortzeko aukera eman (*Crear* botoia sakatuz).
- Bilatze aukera eman (*Buscar* botoia sakatuz).
- Filtratze eremuetan sartutako informazioa ezabatu (*Limpiar* botoia sakatuz).

42 eta 43. irudietan Filter azpi atalaren edukia modu grafikoago batean azaltzen da.

```
<td class="etiqueta" style="width:130px;max-width:130px;white-space:nowrap;">
  <span id="literal.t_codigo"><bean:message key="fct.comun.literal.CodigoCF.long" /></span>
</td>
<td style="width:50px;max-width:50px;white-space:nowrap">
  <html:text styleClass="caja1" styleId="t_codigo" property="t_codigo" maxLength="3" style="width:50px;"onkeypress="convertirMayuscula(
    event, this)"/>
</td>
```



Filter atalean definitu den eremua

Irudia 42 - Eremu baten definizioa

```
<html:submit styleClass="button buscar" property="botonBuscar" onclick="return validarFormFilter_FAC10050(this.form);"><bean:message key='
sis.comun.literal.Buscar.long' /></html:submit>
```



Botoi baten definizioa eta bertan sakatuz gero honek burutuko duen ekintzaren definizioa

Irudia 43 - Botoi baten definizioa

### Detail.jsp

Bilaketa baten emaitzak pantailaratzeko erabilia 44.irudian ikusten den bezala:

RESULTADO DE CONSULTA												
Encontrados 118 datos. Se muestran del 1 al 8.												
	Prop.	Razón Social	Fecha	NI	Denominación	NZ	Denominación	Denoms	Concepto	Total	Factura	Lin Fac.
<input type="checkbox"/>	0	RAZON SOCIAL	19/12/2016	02	MANIPULACION	01	MANIPULACION ENVIOS		PRUEBA 09	1.112,99		
<input type="checkbox"/>	0	RAZON SOCIAL	01/12/2016	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK		PRUEBA N	2,22		
<input type="checkbox"/>	0	RAZON SOCIAL	07/12/2016	02	MANIPULACION	02	MANIPULACION LLEGADAS		PRUEBA 2	2.333,00		
<input type="checkbox"/>	1	PROPIETARIO I DANI	01/03/2017	02	MANIPULACION	01	MANIPULACION ENVIOS		LVA CONCEPTO FACT	23.461,00		
<input type="checkbox"/>	1	PROPIETARIO I DANI	20/03/2017	02	MANIPULACION	01	MANIPULACION ENVIOS		LVA PRUEBA CONCEPTO 1	2.092,00	244	1
<input type="checkbox"/>	1	PROPIETARIO I DANI	20/03/2017	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK		LVA FAC MANUAL	103,00	241	1
<input type="checkbox"/>	1	PROPIETARIO I DANI	23/02/2017	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK		LVA PRUEBA 3	9.902,00	242	2
<input type="checkbox"/>	1	PROPIETARIO I DANI	05/03/2017	02	MANIPULACION	03	MANIPULACION DEVOLUCIONES		LVA PRUEBA FACT 4	95,00	242	3

Irudia 44 - Detail: Emaitzen bistaratzea

Esan den moduan pantaila batek egikaritutako bilaketa baten emaitzak bistaratu behar baditu, Detail azpi atal hau izan beharko du. Hauek izango dira azpi atal honen betebeharrak:

- Bilaketaren emaitzak erakutsi. 44.irudian ikusten den moduan bilaketaren emaitzak taula batean bistaratuko dira.
- Erregistroen ezabatzea burutzea. Taulan agertzen den *checkbox* –a aukeratuz eta ondoren “Eliminar” botoia sakatuz aukeraturako erregistroak ezabatuko dira
- Erregistro baten eguneratzea. Taulan agertzen den arkatz berdean sakatuz erregistro hori eguneratzeko aukera izango da eta pantaila horren Edit.jsp atalera berbidaliko gaitu.

45, 46, 47 eta 48. irudietan Detail azpi atalaren edukia modu grafikoago batean azaltzen da.

```
<display:table requestURI="/facturacion/mae/FAC10050_Action.do?accion=buscar" name="listado" uid="item_10050" class="displaytag"
cellspacing="0" cellpadding="0" pageSize="${registrosPorPagina}" export="true" excludedParams="ajax accion" clearStatus="
${clearStatus}" keepStatus="true" >
```



Bilaketa egikaritzeko ekintzaren definizioa

*Irudia 45 - Detail: Ekintzaren definizioa*

```
<display:column headerClass="icono" media="html" title="<input type='checkbox' id='ck_delete_detail_all' name='ck_delete_detail_all'
onclick='seleccionarTodos(\"ck_delete_detail\")' />" style="width:20px;">
<input class="checkbox" type="checkbox" id="ck_delete_detail" name="ck_delete_detail" value="${item_10050.t_codigo}" />
</display:column>
```



Erregistroen aukeratzea, ondoren aukeratutakoak ezabatzeko

*Irudia 46 - Detail: Ezabatze aukeraren definizioa*

```
<display:column media="html" headerClass="icono" style="text-align:center; width:20px;" title="&nbsp;">
<a href="javascript:ajaxUpdateRequest(PATH_WEB_FACTURACION+'facturacion/mae/FAC10050_ActionCrear.do?accion=mostrarEditar', 't_codigo' +
encodeURIComponent('${item_10050.t_codigo}'), 'contenido_edit'); displayModificar();"><html:img src="${initParam.imagenes}/botonEditar.gif"
border="0" titleKey="sis.comun.literal.Modifi.long"/></a>
</display:column>
```



Eguneraketa aukeraren definizioa (arkatz berdearen definizioa)

*Irudia 47 - Detail: Eguneraketa aukeraren definizioa*

```
<center>
<html:button styleClass="button_eliminar" onclick="displayEliminarDetail(); return validarFormDetail_FAC10050();" property="botonEliminar">
bean:message key='boton.eliminar'/></html:button>
</center>
```



Eliminar botoiaren eta honek burutuko duenaren ekintzaren definizioa

*Irudia 48 - Detail: Eliminar botoia*

## Edit.jsp

Erregistro berri baten sortzea edo jadanik existitzen den erregistro baten eguneratzea burutzeko erabilia. JSP fitxategia berdina izango da, sortze eta eguneratze aukeren arteko desberdintasuna oso txikia izango bai da. 49. irudian errazago ikusi daiteke pantaila baten Edit txantiloia erredia.



CREACIÓN

Conceptos de Facturación Manuales

---

Concepto\*

Literal para la Factura\*

Propietario\*

Nivel 1\*

Nivel 2\*

Fecha de aplicación del Concepto

Referencia Externa

Importes

---

Cantidad Gravada\*

Cantidad Exenta\*

Importe Impuesto

Impuestos

Código Impuesto\*   Valor

Guardar
Limpiar
Cancelar

*Irudia 49- Edit: Erregistro baten sorrera*

Pantaila batek erregistroak sortzeko edo eguneratzeko aukera izatekotan Edit azpi atala izan beharko du. Baliteke pantaila bateko eguneratze aukeran egonez gero, eremu batzuk eskurazinak izatea. Baina zeintzuk dira azpi atal honen betekizunak? Hona hemen horien zerrenda:

- **Erregistro baten sortzea:**
  - Pantailan erabiltzaileak bete beharko duen formularioa hutsik agertuko da. Bertan, erregistro berri hori sortzeko behar den informazioa erabiltzaileak bete beharko du.
  - “Limpiar” botoia sakatuz gero betetako eremuak hustuko dira, hau da, formularioa berriro ere kargatuko da.
- **Erregistro baten eguneratzea:**
  - Aukeratutako erregistro baten informazioa pantailaratuko da eremu desberdinetan. Eremu batzuk editagarriak izango dira eta beste batzuk ez.
- **“Guardar”** botoia sakatuz eta agertzen diren eremuak modu egokian betez gero, erregistro berria sortuko da. Eremu baten bat modu desegokian betez gero ordea, errore bat pantailaratuko da.
- **“Cancelar”** botoiak Detail atalera eramango gaitu (pantailak aukera hori izatekotan).

Edit azpi atalaren garapenaren gauza esanguratsuenak zeintzuk diren modu grafikoago batean ikusteko ikus ondorengo irudia:

```

<c:if test="${modo == 'edicion'}">
  <bean:define id="disabled">true</bean:define>
  <bean:define id="style">caja2</bean:define>
</c:if>
<c:if test="${modo == 'alta'}">
  <bean:define id="style">caja1</bean:define>
  <bean:define id="disabled_n1">true</bean:define>
  <bean:define id="ast_obligatorio"></bean:define>
</c:if>

```

- "modo" aldagiaren arabera eremuen itxura (estiloa) bata edo bestea izango da.
- Eremu batzuk editagarriak izango dira eta beste batzuk ez.

Irudia 50 - Edit: Eremuak

JSP fitxategi guztietan erabiliko diren etiketen (*tag*) izena adierazten da hasieran goialdean. Halaber, hauei esleitu zaien ezizena ere bertan definituko da.

## JavaScript

Pantaila bakoitzeko .js fitxategi bat egongo da eta hau pantaileri dinamikotasuna emateko erabiliko da.

Fitxategi honetan funtzio batzuk definitu dira ekintza batzuk burutzeko. Ekintza hauek pantaila bakoitzaren Action klasearekin lotuta egongo dira, beste batzuetan aurre definitutako funtzio batzuekin elkartuta daude.

```

function validarFormFilter_FAC10050(form)
{
  var error = validarForm(form);
  return submitForm(form, error, 'contenido_detail', 'displayBuscar()');
}

```

Irudia 51 - JavaScript: Funtzioen definizioa

51. irudian ikusten den moduan definitutako funtzioak beste funtzio batzuk erabiltzen ditu formularioa baliozkotzat jotzeko edo ez. Funtzio hauek JavaScript orokor batean (*general.js*) definituta daude eta aplikazioan dauden pantaila guztiek (fakturatze moduluak zein beste bi moduluak) funtzio hauetaz baliatzen dira balioztatze hauek burutzeko.

## XML

Atal honekin exekutatu diren Query -ak eta hauek bueltatu dituzten emaitzak kudeatuko dira. iBatis -ek Data Mapper aukera eskaintzen du, hau da, emaitzak *resultMap* batean utziko ditu eta *map* hau Java -ko HashMap klase batean bihurtuko da.

52. irudian berreskuratuko den informazioaren sailkapena egiten da, *resultMap* batean bihurtuz.

```

<resultMap id="FAC10050List" class="java.util.HashMap">
  <result column="t_codigo" property="t_codigo" javaType="java.lang.String" />
  <result column="t_codtcf" property="t_codtcf" javaType="java.lang.String" />
  <result column="t_denniv1" property="t_denniv1" javaType="java.lang.String" />
  <result column="t_codscf" property="t_codscf" javaType="java.lang.String" />
  <result column="t_denniv2" property="t_denniv2" javaType="java.lang.String" />
  <result column="t_denomi" property="t_denomi" javaType="java.lang.String" />
</resultMap>

```

*Irudia 52 - XML: resultMap sortzen*

Eta *resultMap* hau informazioarekin betetzeko honako Query hau definitu da (ikus 53. irudia):

```

<select id="selectByFilter" resultMap="FAC10050List" >
  SELECT
  COF_CODIGO t_codigo,
  COF_CODTCF t_codtcf,
  (select TTF_DENOMI from TIPOS_TARIFA_FAC where TTF_CODIGO = COF_CODTCF) t_denniv1,
  COF_CODSCF t_codscf,
  (select SUF_DENOMI from SUBTIPOS_TARIFA_FAC where SUF_CODIGO = COF_CODSCF and SUF_CODTTF = COF_CODTCF) t_denniv2,
  COF_DENOMI t_denomi

  FROM
  CONCEPTOS_FACTURACION

  <dynamic prepend="WHERE">
    <isNotEmpty property="t_codigo" prepend="AND">
      UPPER(COF_CODIGO) LIKE (#t_codigo#)
    </isNotEmpty>

    <isNotEmpty property="t_codtcf" prepend="AND">
      COF_CODTCF = #t_codtcf#
    </isNotEmpty>

    <isNotEmpty property="t_codscf" prepend="AND">
      COF_CODSCF = #t_codscf#
    </isNotEmpty>

    <isNotEmpty property="t_denomi" prepend="AND">
      UPPER(COF_DENOMI) LIKE (#t_denomi#)
    </isNotEmpty>
  </dynamic>

  ORDER BY
  COF_CODIGO
</select>

```

*Irudia 53 - XML: Query definizioa*

Prozedura hau SELECT –ak exekutatzeko bakarrik erabili da, gainontzeko query –ak (insert, update eta delete) burutzeko beste modu batean garatu egin da.

Ondoren sakonean azalduko den prozedimenduarekin elkartu dira ekintza hauek. Hau da INSERT, UPDATE eta DELETE –k datu basean eragiten dituzten aldaketak .sql *package* batekin elkartuta daude eta honek barruan prozedimenduak izango ditu.

Hasteko parametroen mapa (*parameter map*) bat definituko da 54. irudian adierazten den moduan. Parametroen definizioa egiten da, zein motakoak diren eta sarrera/irteerakoak diren adierazten da. Parametro hauek prozedimenduak behar dituenak izango dira, baten bat ez egonez gero errorea botako du.

```

<parameterMap id="insertMap" class="java.util.HashMap">
  <parameter property="t_codigo"      javaType="java.lang.String"      jdbcType="VARCHAR" mode="IN" />
  <parameter property="t_denomi"      javaType="java.lang.String"      jdbcType="VARCHAR" mode="IN" />
  <parameter property="t_codtcf"      javaType="java.lang.String"      jdbcType="VARCHAR" mode="IN" />
  <parameter property="t_codscf"      javaType="java.lang.String"      jdbcType="VARCHAR" mode="IN" />
  <parameter property="t_codpum"      javaType="java.lang.String"      jdbcType="VARCHAR" mode="IN" />
  <parameter property="po_error"      javaType="java.lang.Integer"     jdbcType="INTEGER" mode="INOUT" />
  <parameter property="po_men_java"   javaType="java.lang.String"      jdbcType="VARCHAR" mode="INOUT" />
  <parameter property="po_men_error"  javaType="java.lang.String"      jdbcType="VARCHAR" mode="INOUT" />
</parameterMap>

```

Irudia 54 - XML: Parameter Map definizioa

Hau definitu ostean, *package* –ak duen prozedimenduari deia definituko da 55.irudian adierazten den moduan:

```

<procedure id="insert" parameterMap="insertMap">
  {call PACK_FACT10050.ON_INSERT(?, ?, ?, ?, ?, ?, ?, ?)}
</procedure>

```

Irudia 55 - XML: Prozedimenduari deia

.sql PACKAGE

Package –an datu basean aldaketak eragingo dituzten prozedimenduak definituko dira. Pakete hauek datu basean biltegitratuko dira.

```

PROCEDURE ON_INSERT (p_codigo IN VARCHAR2, p_denomi IN VARCHAR2,
  p_codtcf IN VARCHAR2, p_codscf IN VARCHAR2,
  p_codpum IN VARCHAR2, p_error IN OUT NUMBER,
  p_men_java IN OUT VARCHAR2, p_men_error IN OUT VARCHAR2);

```

Irudia 56 - Package: Prozedimenduaren definizioa

Lehenik eta behin prozedimendu guztiak definituko dira eta ondoren haien implementazioa garatuko da.

## Funtzionalitateen azalpena

Fakturatze moduluak dituen funtzionalitate bakoitzaren funtzionamendua labur-labur azalduko da. Funtzionalitate orokorren azalpena emango da eta moduluak dituen funtzionaltasun “desberdinenak” ere bai.

### *Bilatzea*

Informazioa pantailaratzeko erabiltzen da, pantaila bakoitzeko informazio zehatza adieraziko da. Hau guztia bistaratzeko datu basearen kontrako kontsulta egikarrituko da bistaratze horrek behar dituen datuak eskatuz.

### *Sorrera*

Erregistro berriak gehitzeko erabiltzen da, pantaila bakoitzak datu zehatz batzuk beharko ditu erregistro berri hori gehitzeko.

### *Eguneratzea*

Dagoeneko existitzen den erregistro bat aldatzean datza.

### *Ezabatzea*

Jadanik dagoen erregistro baten edo gehiagoren ezabaketa exekutatzeko erabilia.

### *Beste funtzionalitate batzuk*

#### Fakturazio bidea: Manipulazio & Biltegitratzea

Manipulazio eta Biltegitratze fakturazio bideen kudeaketaz arduratuko diren funtzionalitateak. Baina, zer da fakturazio biderako prozesua?

- **Fakturazio bidea:** Entitate bakoitzak fakturatu behar dituen bukaerako zenbatekoaren kalkuluak burutuko dituen prozesua.

Kalkulu hauek egiteko tarifa bat, biltegi bat eta jabe bat beharrezkoak izango dira. Fakturazio bidea prozesuak 15. taulan agertzen diren prozesuak egin ahal izango ditu.

PROZESUA	ZER EGINGO DU?
<b>Jaurtiketa</b>	Manipulazio edo Biltegitratze fakturazio bidea prozesatu.
<b>Azterketa</b>	Aurretik jaurtitako fakturazio bideen azterketa.
<b>Desegitea</b>	Aurretik jaurtitako fakturazio bideen desegitea.

Taula 15 - Fakturazio bidea (prozesuak)

## Fakturak

Fakturak kudeatuko dituzten funtzionalitateak azalduko dira orain. Baina hasteko, zer da faktura bat?

Fakturazio bide baten edo gehiagoren batukaria da. Fakturazio bide horiek denbora tarte batean jabeei fakturatu egin behar zaizkie.

## Fakturen sortzea

Fakturak sortzeko garatutako prozesuak. Hauek sortzeko dauden aukerak jarraian azalduko dira.

- Automatikoa: Fakturen sortze automatikoa.
- Eskuzko sortzea: Fakturen eskuzko sortzea. Erabiltzaileak aplikazioko puntu honetan fakturak sortu ditzake eskuzko fakturazio batekin. Kontzeptu hau faktura sortzen den momentuan bertan sortuko da baita ere.

## Fakturen ixtea

Prozesu honetan faktura bat baieztatuko da eta behin betiko faktura sortuko da. Faktura bat itxi ondoren, faktura horren gainean ezingo da aldaketarik egin.

Faktura ixterakoan sistemak kode zenbaki bakarra eta sekuentziala atzitu dio. Kode hori fakturaren kode fiskala izango da.

## Fakturen inprimatzea

Faktura bakoitzeko .PDF fitxategi bat sortuko da. Fitxategi hori sisteman biltegitatu daiteke edo modu fisikoan inprimatu.

Inprimatzerako orduan bi faktura mota egongo dira:

- **Laburbildua**: Fakturaren datu orokorrak erakutsiko ditu.
- **Zehaztua**: Fakturaren datu zehaztuak erakutsiko ditu. Faktura mota honetan datu orokorrak eta fakturaren datu multzo bat agertuko da. Faktura hori osatzen dituzten lerroen xehetasunak, hain zuzen.

## Garapenarekin arazoak

Proiektu guztian gertatzen den moduan nahiz eta analisisa eta diseinua egiterako orduan kasu guztiak kontutan izan, ezustekoak agertu daitezke.

Inplementazioan zehar agertu diren ustekabe hauek ez dira oso larriak izan, baina proiektuan atzerapen txikiak sortu dituzte. Gainera, "arazo" hauei konponbide bat emateko prestakuntza estra bat hartu behar izan da.

Esan dudan moduan ezustekoa ez da oso larria izan baina proiektuaren garapenaren zehar agertu denez aipagarria da eta atal honetan hura zertan datzan azalduko da.

## Saioan informazioa mantendu

Izenburuan irakurtzen den moduan arazoa erabiltzaile baten saioan informazioa mantentzea izan da arazo aipagarriena.

Fakturazio-Tarifen mantentze pantaila garatzerakoan sortu zen arazo hau. Baina hobeto ulertzeko pausoz-pauso azalduko da pantaila honen funtzionamendua, zein ataletan agertu zen arazoa eta zein izan zen aurkitutako irtenbidea azkenean pantailaren funtzionamendua egokia izateko.

Fakturazio-Tarifen kudeaketa aurrera eramatea du helburu pantaila honek. Hau da, fakturazio-tarifak bilatu, sortu, eguneratu eta ezabatu ahal dira pantaila honekin. Azalpen honekin pantaila orokor bat dirudi, baina eragozpenak sortu/eguneratu inplementazioarekin agertu ziren.

### *Fakturazio-Tarifa*

Lehenik eta behin, zer behar da Fakturazio-Tarifa bat sortzeko? Ulergarria izateko sortze prozesu hau bi ataletan bananduko da. Beraz, tarifak izango dituen bi atalak hauek izango dira: Tarifaren goiburua eta lerroak.

### Tarifaren goiburua

Tarifaren atal hau bakarria izango da eta ondorengo parametroek osatzen dute, dena den, 57.irudian pantailako eremuak ikus daitezke.

- Tarifa kodea.
- Izendapena.
- Zein tarifa motari dagokion (bere kodea).
- Zein azpi-tarifa motari dagokion (bere kodea).

CREACIÓN

Tarifa de Facturación

Código Tarifa		Denominación*	PRUEBA PRUEBA
Nivel 1*	02		MANIPULACION
Nivel 2*	01		MANIPULACION ENVIOS

Irudia 57 - Fakturazio-Tarifa goiburua

## Lerroak

Fakturazio-tarifa batek derrigorrez lerroak izan beharko ditu, sortze pantailara sartzean goiburua bete behar da eta lehenengo lerro bat sortu beharko da ere bai.

Lerro bakoitza bakarra izango da eta ondorengo parametroek osatuko dute:

- Fakturazio-kontzeptua kodea (pantailan fakturazio-kontzeptu horren ezaugarriak bistaratzen dira erabiltzaileak informazio gehiago izan dezan).
- Zerga kodea.
- Atal finkoa.
- Atal aldakorra.

Linea de Tarifa

Concepto Facturación	CEN	CAJAS ENVIADAS
Punto Medible	30	CAJAS MANIPULADAS
Unidad Medida	08	CAJAS
Importe Impuesto*	01	IVA 10 Valor 10

Parte Fija	Parte Variable
Fijo <input type="checkbox"/> Exento <input type="checkbox"/>	Variable <input checked="" type="checkbox"/>
Valor Fijo	Limite Mínimo 12 Limite Máximo 2999
	Fórmula de Cálculo DI - DIRECTO

Irudia 58 - Fakturazio-tarifa lerroa

Arazoa lerro baten alde aldakorra sortzean edo aldatzean agertu zen. Alde aldakor hori ez delako Fakturazio-tarifa baten lerroen datu baseko taulan gordetzen, tarifa-lerro baten ezaugarri hori biltegitratzen duen taula beste bat da.

Atal aldakor horretan Kalkulatz-formula (*Fórmula de Cálculo*) bat aukeratu behar da. 58. Irudian zirkunferentzia urdina dagoen lekuan ikusi daitekeen moduan. Behin kalkulatz-formula aukeratu dugula (*Directo* edo *Escalado* –ren artean), **formula mota bakarra aukeratu**



**daiteke**, formula horren datuak sartu behar dira eta horretarako lupa botoian sakatu beharko dugu. Botoia sakatzerakoan *popup* bat bistaratuko da.

Directo kalkulatze-formula:

Facturación » Tarifas » Tarifas de facturación » Mant. Tarifas de facturación (FAC20020) ?

CREACIÓN

Tarifa de Facturación

Código Tarifa  Denominación\* PRUEBA PRUEBA

Nivel 1\*

Nivel 2\*

Línea de Tarifa

Concepto Facturación

Punto Medible

Unidad Medida

Importe Impuesto\*

Parte Fija

Fijo  Exento

Valor Fijo

Parte Variable

Variable

Límite Mínimo  Límite Máximo

Fórmula de Cálculo

Fórmula de Cálculo

Valores Directo

Importe Fijo\*

Importe/Unidad\*

Exento

Irudia 59 – Directo kalkulatze-formula pertsonalizatzeko popup-a

Escalado kalkulatzte-formula:

The image shows a software interface for creating a tariff. A modal window titled "Fórmula de Cálculo" is open, allowing for the configuration of calculation parameters. The background form includes fields for tariff code, denomination, levels, concept, measurable point, unit, and tax amount. The modal window has two main sections: "MODIFICACIÓN CREACIÓN" for setting calculation values (Hasta, Importe Fijo, Importe/Unidad, Exento) and "RESULTADO DE CONSULTA" which displays a message that no data was found matching the selected criteria, with an "Eliminar" button to clear the results. The modal window also features "Guardar" and "Limpiar" buttons in both sections, and a "Cancelar" button at the bottom.

Irudia 60 – Escalado kalkulatzte-formula pertsonalitzatzeko popup-a



59 eta 60. Irudietan ikusi daitekeenez aukeratutako kalkulatz-formularen arabera *popup* bat bistaratuko da edo beste bat.

Aipatutako irudietan “Guardar” botoi bat dago. Hasiera batean behin pantaila horien eremu guztiak bete ondoren botoi hori sakatuz gero, datu basera zuzenean agindu bat bidaltzen zen egindako aldaketak edo sortutako datu berriak gordetzeko.

Baina zer gertatuko litzateke formula batetik bestera aldatzerakoan? Aurretik sortutako edo aldatutako informazioa galduko litzateke?

Guztia ezabatzearen jokoera ez zen egokia, beraz, fakturatze-tarifa lerro horren sorrerak edo eguneratzeak sortutako “saioan” guztia gorde beharko zen. Lehen aipatutako *popup* leihoek daukaten “Guardar” botoi horrek benetan datu basera ez du agindurik bidaliko, guztia “gordeta” geldituko da eta lerro osoaren “Guardar” botoia sakatzerako orduan agindua bidaliko da datu basera egindako aldaketa guztiekin.

## Arazoaren konponbidea

Pantaila horrek duen Action Java klaseak kudeatuko du konponbidea. *request* parametroak saioa lortzeko metodo bat izango du. Saio horretan atributu desberdinak gorde daitezke, adibidez eguneratu nahi dugun tarifa baten kodea.

Kasu honetan *request* parametroan bi atributu “berezi” eratuko dira arazoari ebazpen bat emateko 61.irudian pantaila hasieratzen denean ikusi daiteke nola bi atributu horiek erazten diren. Atributuen mota zein izango den hau da:

- *Directo*: HashMap bat izango da, erregistro bakarra sortu edo aldatu ahal dezakegulako formulatze-kalkulu hau aukeratuz gero.
- *Escalado*: HashMap –ez betetako ArrayList bat, erregistro bat baino gehiago izan ahal ditulako formulatze-kalkulu honek.

```
public ActionForward unspecified(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception
{
    FAC20020_LinForm formulario = (FAC20020_LinForm) form;
    formulario.reset();

    formulario.setT_codtar(request.getParameter("t_codtar"));
    formulario.setT_codtcf(request.getParameter("t_codtcf"));
    formulario.setT_codscf(request.getParameter("t_codscf"));
    request.setAttribute("t_codtar", request.getParameter("t_codtar"));
    request.setAttribute("t_codtcf", request.getParameter("t_codtcf"));
    request.setAttribute("t_codscf", request.getParameter("t_codscf"));
}
/*
 * Se guardara en la sesion Los cambios que se hagan tanto en ESCALADO como en DIRECTO
 */
request.getSession().setAttribute("session_list", new ArrayList<HashMap>());
request.getSession().setAttribute("session_direct", new HashMap());
/*
 * Se guardara en la sesion el maximo idrang
 */
request.getSession().setAttribute("max_idrang", 0);
request.setAttribute(MSG_OK_CAB, request.getParameter(MSG_OK_CAB));
return mapping.findForward(FILTER);
}
```

### Irudia 61 - Atributuen deklarazioa

Saioan zehar bi kalkulatz-formulen informazioa gordeko da, nahiz eta *Directo* motatik *Escalado* –ra aldatu berriro ere lehenengoa aukeratuz gero, erabiltzaileak hasiera batean sartutako informazioa bertan izango du.

Egitura bakoitzak popup leihoak eskatzen dituen eremuen erreferentzia eta informazioa gordeko du (ikus 59 eta 60. irudiak). Horretaz aparte, HashMap bakoitzean zer ekintza burutu behar den erregistro horrekin eta datu basekoa den edo ez adieraziko da. Beherago agertzen den laburpenarekin hobeto ulertuko da ekintza eta esleipen hauen gaia:

#### 0-tik lerro bat sortzen

LERRO BERRI BAT SORTZEN		
	Directo	Escalado
<b>Ekintza</b>	A	A
<b>Datu basekoa da?</b>	N	N

Taula 16 - Ekintza esleipena 0-tik

Bi kalkulatzeko-formula motetan berdina adieraziko da. **A** ekintzak “Guardar” botoia sakatzerakoan momentuan hautatuta dagoen formula motako erregistroa sortuko du. Eta **N** esleitu zaio erregistro horiek ez direlako datu basean existitzen.

#### Lerro bat eguneratzen

Aukera honen azalpena burutzea pixka bat korapilatsua da, beraz, sinplifikatzeko azpi atalak egingo dira.

#### Lerroak Directo mota zuen...

LERROAK Directo MOTA ZUEN...		
	...eta horren gainean aldaketak egingo dira	
	Directo	Escalado
<b>Ekintza</b>	M	Hutsik
<b>Datu basekoa da?</b>	S	Hutsik
	...eta Escalado mota gordeko da orain	
	Directo	Escalado
<b>Ekintza</b>	B	A
<b>Datu basekoa da?</b>	S	N

Taula 17 - Directo mota aldatzen ekintza esleipenak

Lerroak Escalado mota zuen...

<b>LERROAK Escalado MOTA ZUEN...</b>		
	<b>...eta horren gainean aldaketak egingo dira</b>	
	Directo	Escalado
<b>Ekintza</b>	Hutsik	M
<b>Datu basekoa da?</b>	Hutsik	S
	<b>...eta Escalado mota gordeko da orain</b>	
	Directo	Escalado
<b>Ekintza</b>	A	B
<b>Datu basekoa da?</b>	N	S

Taula 18 - Escalado mota aldatzen ekintza esleipenak

Erabiltzaileak pantailan egiten dituen aldaketen arabera azkenean atributu horiek modu batean edo bestean geratuko dira. Beraz, konbinazio hauek edukita datu basean dagozkion ekintzak exekutatu dira.

```
hash.put("t_accion", "A");
hash.put("t_bbdd", "N");
```

Irudia 62 - Ekintza eta DDBB esleipena HashMap ean

## JasperReports

Fakturatze moduluan duen funtzionalitateen artean fakturen inprimatzea dago. Faktura hauek .pdf formatuan inprimatuko dira hauek gordetzeko edo inprimagailu batera bidaltzeko.

Honen garapena JasperReports liburutegia erabiliz eta Eclipse –k duen Report Design *plugin* –arekin egin da. Bi faktura mota inprimatu daitezke: laburtua edo zehaztua.

The screenshot shows a web application interface for invoice printing. The main area displays a table of invoices with columns for Cod. Prov., Cod. Def., Prop., Denominación, Fecha creación, Imp. sal., Exen., and Imp. impor. A dialog box is open, asking "¿Esta seguro de querer imprimir estas facturas?" (Are you sure you want to print these invoices?). The dialog box has a dropdown menu for "Tipo Factura" set to "RESUMIDA" and two buttons: "Imprimir" (Print) and "Cancelar" (Cancel).

Cod. Prov.	Cod. Def.	Prop.	Denominación	Fecha creación	Imp. sal.	Exen.	Imp. impor.
15	28	1	PROPIETARIO 1 DANI	09/03/17	1.111.000	2.222.000	3.333.000
16	27	1	PROPIETARIO 1 DANI	09/03/17	1.111.000	2.222.000	3.333.000
101		1	PROPIETARIO 1 DANI	10/03/17	54.407.398	9.340.000	17.396.695
106		1	PROPIETARIO 1 DANI	03/03/17			
123		1	PROPIETARIO 1 DANI	01/04/17			
124		1	PROPIETARIO 1 DANI	01/04/17			
126		1	PROPIETARIO 1 DANI	01/04/17			
127		1	PROPIETARIO 1 DANI	01/03/17			

Irudia 63 - Faktura motaren hautaketa

“Imprimir” botoia sakatu ondoren pantaila honen Action klaseak prozesu bat hasiko du eta JasperReport –ak behar dituen parametroak bidaliko dizkio Report –a bete ahal izateko.



**FACTURA RESUMIDA**

PROPIETARIO 1 DANI  
 PROPIETARIO 1 DANI  
**ATT -->**  
 DIRECCION con e  
 28000 LOCALIDAD  
 PROVINCIA PAIS

Fecha Factura	2017-03-10	Factura Provisional	101	PAG.	1 / 2
Concepto				Importe Total	
02 - MANIPULACION	01 - MANIPULACION ENVIOS	.....			120,000
02 - MANIPULACION	01 - MANIPULACION ENVIOS	.....			140,000
02 - MANIPULACION	01 - MANIPULACION ENVIOS	.....			500,000
02 - MANIPULACION	01 - MANIPULACION ENVIOS	.....			46.841,109
02 - MANIPULACION	01 - MANIPULACION ENVIOS	.....			2.000,000
LVA LINEA2					
LVA LINEA 2					
EXTTEXT					
02 - MANIPULACION	01 - MANIPULACION ENVIOS	.....			1,000
FOHER LAZO					
FOHER LAZO PETICION JOAQUIN					
02 - MANIPULACION	02 - MANIPULACION LLEGADAS	.....			3.000,000
03 - ALMACENAJE STOCK	01 - ALMACENAJE STOCK	.....			100,000
03 - ALMACENAJE STOCK	01 - ALMACENAJE STOCK	.....			110,000
03 - ALMACENAJE STOCK	01 - ALMACENAJE STOCK	.....			130,000
03 - ALMACENAJE STOCK	01 - ALMACENAJE STOCK	.....			5,289
CONCEPTO LVA					
PRUEBA PARA CALCULAR IMPUESTO					
REFEXTLVA					
03 - ALMACENAJE STOCK	01 - ALMACENAJE STOCK	.....			400,000
LVA LINEA 4					
LVA LINEA 4					
EXTTEXT					
03 - ALMACENAJE STOCK	01 - ALMACENAJE STOCK	.....			60,000
LVA LINEA 6					
LVA LINEA 6					
EXTTEXT					



Faktura laburtuaren garapenean zehar ez zen inolako arazorik egon, baina zehaztutako fakturak, aldiz, hainbat atzerapen sortu zituen. Ez da *report* simple bat, azpi *report* bat dauka.

Faktura zehaztuaren informazioa osorik lortzeko ezinezkoa zen bi *Query* lotzea, bata bestearekin ez bai zutelako erlaziorik. Beraz, aurkitu den konponbidea azpi-*report* bat erabiltzea da.

Bisualki bi fakturak berdinak dira, baina zehaztuaren kasuan fakturan informazio gehiago ateratzen da 66. irudian erakusten den moduan:

02 - MANIPULACION	01 - MANIPULACION ENVIOS		46.841,109
IGSFAC01 / 0 S CEN - CAJAS ENVIADAS		-----	4.000,000
IGSFAC01 / 0 S PGR - PEDIDOS GRABADOS		-----	750,000
IGSFAC01 / 0 S UEN - UNIDADES ENVIADAS		-----	10.042,000
IGSFAC03 / 0 S CEN - CAJAS ENVIADAS		-----	4.000,111

Irudia 66 - Faktura zehaztua



## EGIAZTAPENA ETA EBALUAZIOA

Atal honetan fakturatzeko moduluaren funtzionamendua egokia dela egiaztatzen da. Horretarako egiaztapen eta ebaluazio hau jarraitzeko proben-txantiloia erabili da. Pantaila bakoitzeko proba-eredu txantiloia bat bete da.

Lehenago esan den moduan ia pantaila gehienek jarrera berdina dute, beraz, probak guztientzako ia berdinak izan dira, funtzionalitate ez hain orokorrak dituzten pantailentzat berdina erabili da funtzionalitate berezi horiek gehiturik.

Esan beharra dago modulu hau aplikazio komertzial baten parte izangoenez, gainontzeko atalekin probak egiteko jarraitu diren pausuak ere proba hauek egiteko moldatu behar izan dela.

Proba hauek zeintzuk izan diren ikusteko euskaratutako txantiloia (eredua) jarraian izango da, dena den, .eranskinean jatorrizko txantiloia (gaztelaniaz) eskuragarri dago.

### Proben txantiloia

Egiaztatzeko aspektua	Eraitza
<b>BILAKETA</b>	
Fokoa lehenengo bilatze irizpidean kokatu behar da.	
Intro tekla sakatzerakoan "Buscar" botoia sakatuz gero izango lukeen jarrera izango du (check, combo -ak eta data-eremuak izan ezik).	
"Limpiar" botoiak pantailara menutik sartuko baginaren jarrera berdina izango du.	
Botoien izenak "Limpiar" eta "Buscar" dira.	
Tabulatzerakoan lehenengo "Buscar" botoia eskuragarri egongo da eta ondoren "Limpiar" botoia.	
Zenbakitzerako orduan bilatze irizpidea galtzen ez dela konprobatu.	
<b>MULTI-ERREGISTROA</b>	
Datuak erakusten dituen sareta taulako goiburuekin zutabetan egon behar da eta errenkadek ez dute sigi-saga egin behar.	
Laburdura duten zutabeetan tips -ak (mezu argigarriak) erabili.	
Tauletako zenbakiak (kantitateak adierazten dituzten zenbakiak) eskumaldean justifikatu behar dira eta haien bukaeran hutsune bat egon beharko da.	
Kantitateak adierazten dituzten zenbakiak maskara modu egokian ezarri behar dute (milako puntua eta hamartarra).	
Kodeak adierazten dituzten zenbakiak ezkerrean justifikatuko dira.	
Botoien izenak "Crear" eta "Eliminar" dira.	
Goiburuek lerro bat baino gehiagokoak ezin dira izan.	
Multi-erregistroko erregistro bakoitza lerro batekoa izango da.	
<b>TXERTATZEA</b>	
Sortze blokeko sartzekoan kurtsorea editagarria den lehenengo eremuan kokatuko da.	
Intro tekla sakatzean "Crear" botoia sakatuz gero izango lukeen jarrera izango du (check, combo -ak eta data-eremuak izan ezik).	

Eremuak hutsuneekin bakarrik idaztea ez da posiblea. Hori gertatzekotan nulo gisa interpretatuko dira.	
Botoiaren izena "Crear" da.	
"Volver"/"Cancelar" sakatuz gero, kontsulta berriro ere exekutatu da multi-erregistroa erakutsiz.	
Botoiaren izena "Guardar" da.	
<b>EDIZIOA</b>	
Edizio blokerak sartzerakoan kurtsorea editagarria den lehenengo eremuan kokatu da.	
"Guardar" botoia sakatzerakoan aldaketak gorde behar ditu.	
Intro tekla sakatzean "Guardar" botoia sakatuz gero izango lukeen jarrera izango du (check, combo -ak eta data-eremuak izan ezik).	
Eremuak hutsuneekin bakarrik idaztea ez da posiblea. Hori gertatzekotan nulo gisa interpretatuko dira.	
Botoiaren izena "Guardar"/"Modificar" izango da.	
"Volver"/"Cancelar" sakatuz gero, kontsulta berriro ere exekutatu da multi-erregistroa erakutsiz.	
<b>EZABAKETA</b>	
Multi-erregistroan ezabatu nahi diren erregistroak aukeratu dira eta ondoren ezabaketa botoia sakatu da.	
Ezabaketak bilaketa edo paginatzeak duten jarrera berdina izango du.	
Erregistroa/erregistroak ezabatu ondoren kontsulta berriro exekutatu da.	
<b>ESPORTAZIOA</b>	
"Exportar a Excel" sakatuz gero, .xls fitxategi baten deskarga hasiko da. Bertan, multi-erregistroan agertzen diren emaitzak agertuko dira.	
"Exportar a PDF" sakatuz gero, .pdf fitxategi baten deskarga hasiko da. Bertan, multi-erregistroan agertzen diren emaitzak agertuko dira.	

Taula 19 - Proben txantiloia

Emaitza zutabearen **OK** agertuko da dena ondo joan bada, **N/A** pantaila horrek funtzionalitate edo ezaugarri hori ez izatekotan eta **hutsik** geratuko da emaitza zuzena ez izatekotan.



## ONDORIOAK ETA ETORKIZUNeko LANA

Orain proiektuaren garapena amaitu ondoren ateratako ondorioak aztertzeko ordua heldu da. Ondorioak ateratzeko, hasiera batean planifikatutako lana vs. benetan egindako lana alderatuko da.

Hau guztia ikusi eta aztertu ondoren, proiektu honen gainean dagoen etorkizunerako lana aztertuko da ere bai.

### Ondorio orokorrak

Estimatutakoaren eta benetan lanean emandakoaren konparazioa egingo da. Hau egiteko hasiera bateko planifikazio taula berreskuratuko dugu eta bere alboan arlo horretan emandako denbora erreala adieraziko da (Ikus 20. taula).

FASE IZENA	DENBORA ESTIMAZIOA (orduak)	DENBORA ERREALA (orduak)
<b>Ikerketa &amp; Betekizunen bilketa</b>	<b>20</b>	<b>30</b>
Arazoa ulertu	10	10
Konponbideak aurkitu	10	20
<b>Ikaskuntza</b>	<b>40</b>	<b>70</b>
Arkitektura ulertu	10	20
Lengoaiekin hasiera	30	50
<b>Antolaketa</b>	<b>49</b>	<b>48</b>
Hasierako planteamendua	10	10
Atazen antolaketa	10	10
Lan ingurumena prestatu	4	8
Zuzendariarekin bilerak	25	20
<b>Analisia &amp; Diseinua</b>	<b>34</b>	<b>50</b>
Domeinuaren eredia	8	10
Entitate-erlazio diagrama	8	10
Klase diagrama	8	10
Sekuentzia diagramak	10	20
<b>Inplementazioa &amp; Garapena</b>	<b>206</b>	<b>230</b>
Fakturazio Moduluaren Eraikuntza	30	50
Atalak eraiki	176	200
<b>Probak</b>	<b>20</b>	<b>30</b>
Proba kasuak antzeman	10	10
Proben egikaritzea	10	20



<b>Dokumentazioa</b>	<b>75</b>	<b>135</b>
Sailkapena	5	5
Proiektuaren memoria	50	100
Aurkezpena	20	30
<b>GUZTIRA</b>	<b>444</b>	<b>593</b>

Taula 20 - Estimaturako denbora vs. denbora erreala

Ondorengo taulan (21. Taulan) modu sakonago batean ikusi daiteke pantaila bakoitza eraikitzeke estimaturako denbora vs. emandako denbora erreala.

<b>ATALA/PROZESU IZENA</b>	<b>DENBORA ESTIMAZIOA (orduak)</b>	<b>DENBORA ERREALA (orduak)</b>
Fakturazio kontzeptuen kudeaketa	12	12
Manipulazioko fakturazio biderako jaurtiketa	8	8
Biltegitratzeko fakturazio biderako jaurtiketa	8	8
Eskuzko fakturazio kontzeptuen kudeaketa	12	12
Fakturazio tarifen mantentzea	12	30
Jabeenganako tarifen elkartzea	12	12
Artikulu/tarifa elkartzea	12	12
Manipulazioaren fakturazio bideko azterketa	12	12
Biltegitratzeko fakturazio bideko azterketa	12	12
Biltegitratzeko fakturazio bideko desegitea	12	12
Manipulazioaren fakturazio bideko desegitea	12	12
Fakturen sortze automatizatua	8	8
Fakturen eskuzko sortzea	12	12
Fakturen mantentzea	12	30
Fakturen ixtea	8	8
Fakturen inprimatzea	12	30
<b>GUZTIRA:</b>	<b>176</b>	<b>230</b>

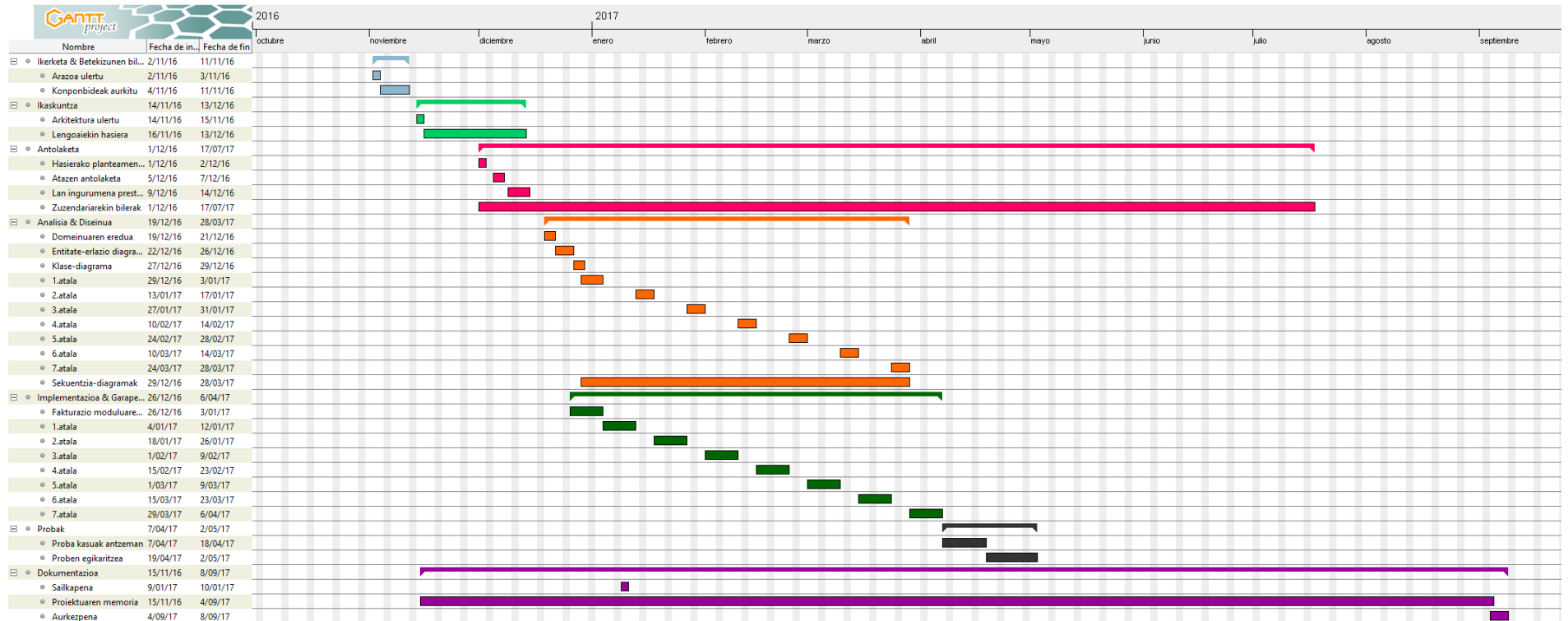
Taula 21 - Prozesuen estimaturako denbora vs. denbora erreala

Ataza batzuen eraikuntzarako denbora berdina izan dela ikusi daiteke da goiko taulan. Hau gertatu da estimazioa egiterako orduan arlo horretan garatzailearen ezjakintasuna kontutan izan delako. Baina ataza inplementatzerako orduan, aldiz, ez zuen denbora gehiagorik eraman aurretik beste ataza batzuk inplementatu nituelako.

Gainontzeko atazen konplexutasuna denbora erreal a adierazi den zutabeen ikusi daiteke argi eta garbi. Ataza horietan ezustekoak agertu dira edo ataza osorik eraikitzeke baliabide gehiago behar izan dira. Fakturen inprimatzean adibidez, lehenago<sup>6</sup> azaldu den

<sup>6</sup> Garapen atalean Fakturen Inprimatzearen azalpena emandakoan

moduan, *report* nagusia eraikitzeko azpi-*report* bat beharrezkoa izan da eta horren implementazioa aurretik ez zegoen hausnartuta.



Irudia 67 - Gantt denbora erreala

## Ondorio pertsonalak

Behin aurreko guztiarekin amaitu dudala inpresio pertsonalak ateratzeko ordua heldu da. Lehenik eta behin esan behar dudana da proiektu bat aurrera eramatea oso gogorra dela eta gauzak ondo eta txukun egiteko, hauek modu konstante batean egin behar direla.

Bestalde, proiektu honen garapenean zoriontsua izan naizela esan dezaket. Praktikak egiten nituen lekuan proposatu zidaten Fakturazio-moduluaren garapena eta praktika orduetan garatu ahal izana iraunkortasuna mantentzea baimendu dit.

Honetaz gain, hasiera batean nahiko frustragarria izan zen proiektua. Hasteko aurretik eraikita dagoen aplikazio batean eraiki nahi den modulu berri horrek zer egingo duen eta zertarako eraikiko den ulertu behar da, beraz, logistikako operadore baten funtzionalitateak zeintzuk diren aztertu eta ulertu behar izan ditut. Hau ulertu ondoren, aplikazioaren “tripetan” sartu behar izan naiz beraren arkitektura ulertzeko, erabilitako baliabideak (programazio lengoaiak, *framework* –a, erramintak, etab...) lehen ez nituelako erabili. Gainera, nahiz eta programazio lengoaiak menperatu eta aurretik *framework* horiekin lan egin duen norbaitentzat ere nahiko konplexua egiten zaio aplikazioaren garapenarekin hastea.

Azkenik atera ahal dudan konklusiorik garrantzitsuena eta nabariena asko ikasi dudala izan da. Hasiera batean nire burua proiektua aurrera eramateko kapaz ez nuen ikusten, baina esaten den moduan “*darraionarentzat da ihizia*”.

## Etorkizunerako lana

Memoria honen zehar esan dudan moduan, Fakturatze-moduluaren lehenengo fase bat izan da hau. Bezero batek logistikako operadore batek duen atal honen inplementaziorik sinpleena eskatu bai zuen. Bezeroak fakturazioarekin zerikusia duten gaiek oinarrizko funtzionalitateak edukitzeko.

Hasieran azaldu den moduan enpresak duen helburu nagusia WMS aplikazioa logistikako operadore baten dituen funtzionalitate guztiak edukitzea da. Gainera, gaur egungo framework batera moldatu nahi dute aplikazio osoa.



## Bibliografia

### Liburuak

Lafosse, J. (2010). *Struts 2*. ENI Ediciones.

Sarin, A. (2012). *Getting Started with Spring Framework*. Createspace Independent Publishing Platform.

### Kontsultatutako Webguneak

Alicante, Universidad. (2011). Hemendik hartua <http://www.jtech.ua.es/j2ee/publico/spring-2012-13/sesion01-apuntes.html> - *Spring framework -aren informazio praktikoagoa*

Alicante, Universidad. (2011). Hemendik hartua <http://www.jtech.ua.es/j2ee/publico/struts-2010-11/sesion02-struts-apuntes.html> - *Struts erramintaren adibide praktikoak*

Alvarez, C. (2014). Hemendik hartua <https://www.genbetadev.com/frameworks/que-es-spring-framework> - *Spring framework -aren informazioa hedatzeko*

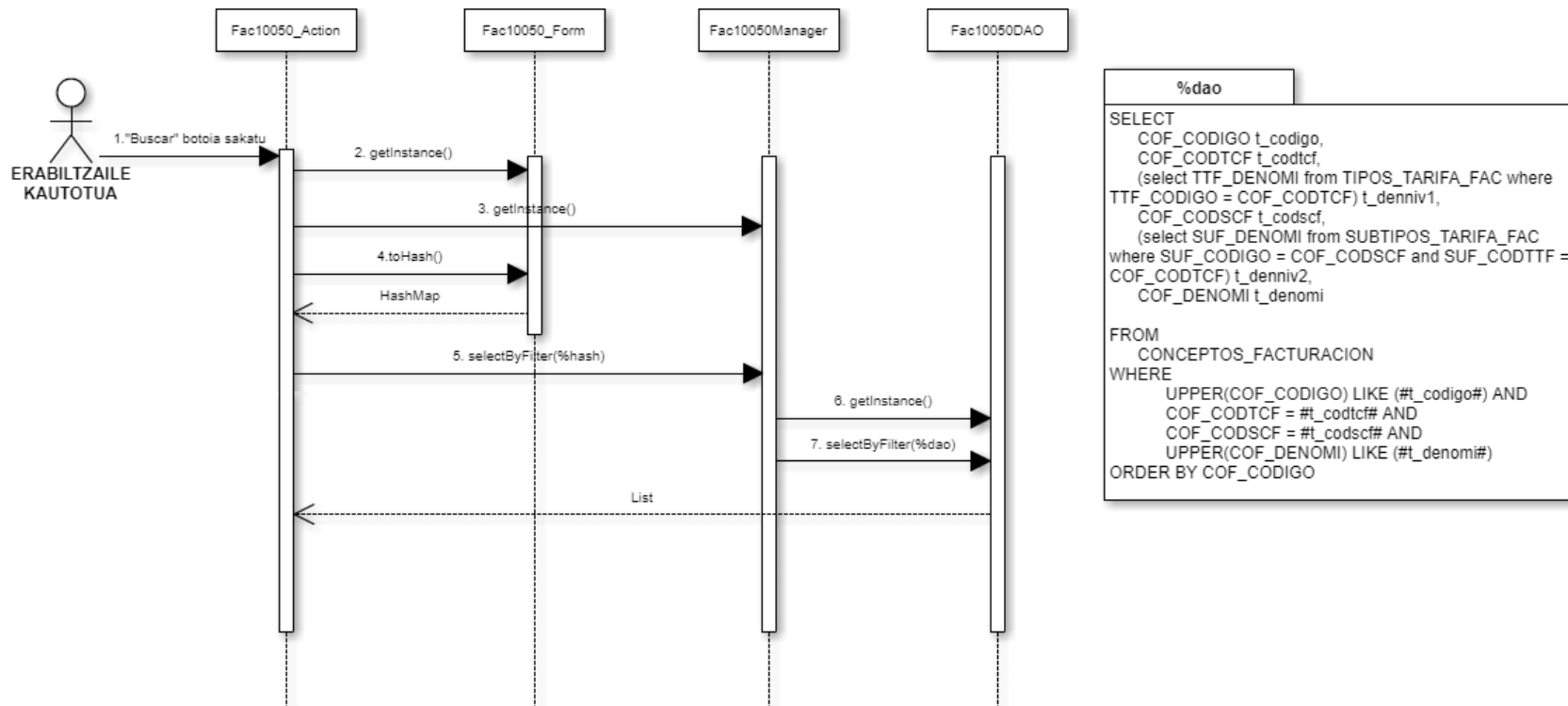
ORACLE. (2017). *ORACLE Help Center*. Hemendik hartua [https://docs.oracle.com/cd/E17984\\_01/doc.898/e14695/undrstnd\\_datasources.htm](https://docs.oracle.com/cd/E17984_01/doc.898/e14695/undrstnd_datasources.htm) - *Zer dira datasource -ak eta zertarako erabiltzen dira?*

unos, T. c. (2009). *Tus ceros y mis unos*. Hemendik hartua <https://mikiorbe.wordpress.com/2009/06/24/descripcion-de-ibatis-%C2%BFque-es-%C2%BFpara-que-sirve/> - *iBatis zer den azalpena modu errazago batean emanda*



## A ERANSKINA

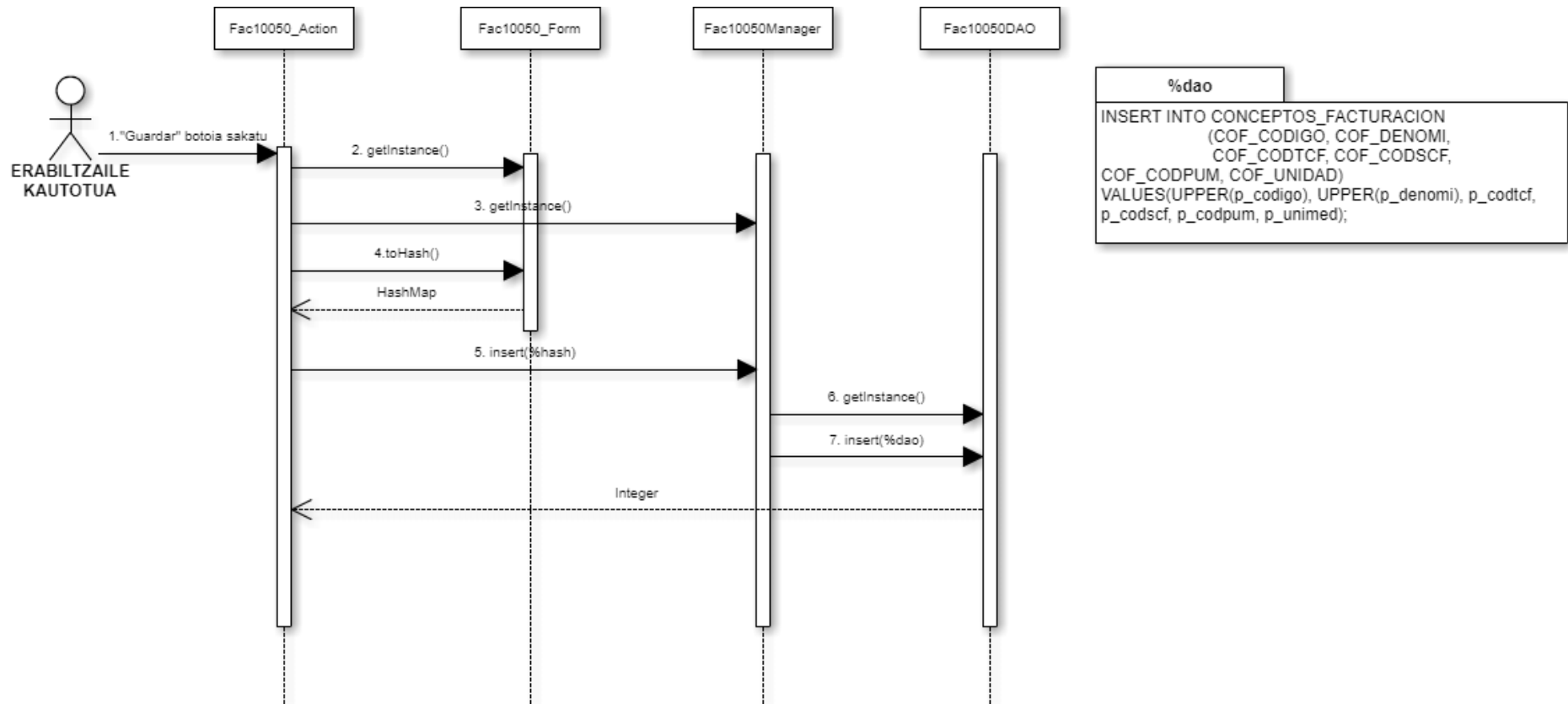
### Filtroen arabera bilatu sekuentzia diagrama



Irudia 68 - Filtroen arabera bilatu (sekuentzia diagrama)



Erregistroa gehitu sekuentzia diagrama



Irudia 69 - Erregistro berri bat sortu (sekuentzia diagrama)



## B ERANSKINA

Probak exekutatzeko jarraitutako gidoiaren bertsio orijinala erakusten da 22. taulan. Lehenago esan bezala pantaila bakoitzeko gidoi hau jarraitu da, kasu zehatz batzuentzat probatzeko kasu gehiago gehitu dira.

Aspecto a probar	Resultado
<b>ACCESO</b>	
Al entrar a la pantalla, aparece junto al título de la opción el código de pantalla.	OK
<b>BÚSQUEDA</b>	
El foco se debe posicionar en el primer criterio de búsqueda	OK
Al pulsar la tecla de intro es como si pulsásemos el botón de buscar (excepto checks, combos y campos fecha)	OK
El botón Limpiar tiene que tener el mismo comportamiento como si entrásemos a la opción desde el menú.	N/A
Los nombres de los botones son Limpiar y Buscar.	OK
Al tabular, se accede primero al botón Buscar y luego al botón Limpiar.	OK
Comprobar que no se pierde el criterio de búsqueda al paginar	OK
<b>MULTI-REGISTRO:</b>	
La rejilla que muestra los datos tiene que estar encolumnada con la cabecera y las filas no tienen que serpentear.	OK
Utilizar tips (mensajes aclaratorios) en las cabeceras de columnas que contengan abreviaturas.	N/A
Los números (que indiquen cantidades) en las tablas se tienen que justificar a la derecha y tendrán que dejar un espacio en blanco al final de su celda.	N/A
Los números que indiquen cantidades deben aplicar la máscara correctamente (punto de millar y decimal).	N/A
Los números que indiquen códigos se justifica a la izquierda.	OK
Los nombres de los botones son Crear y Eliminar.	OK
Las cabeceras no pueden ocupar más de una fila. Vigilar el ancho de las denominaciones.	OK
Cada registro del multiregistro no puede ocupar más de una fila. Vigilar las máscaras	OK
<b>INSERCIÓN</b>	
Cuando se entra en el bloque de alta, el cursor se tiene que posicionar en el primer campo editable.	OK
Cuando pulsa el botón Crear, tiene que ocasionar que se dé de alta el registro y que vuelva al bloque de inserción en disposición de volver a teclear otra alta y situando el foco en el primer campo editable, excepto en las pantallas cabecera/líneas o en aquellas que muestre una alerta (p. ej.: se ha creado el pedido xxxx).	OK
Cuando se pulsa la tecla de intro es como si hubieses pulsado el botón Crear. (excepto checks, combos y campos fecha).	OK
No se permite teclear datos con solo blancos. Si se teclean, se tiene que interpretar como nulos o si hay validación asociada, como si el dato introducido no existiera.	OK
El nombre del botón es Crear.	OK
Tras crear el registro se ejecuta de nuevo la consulta en el multiregistro (donde se vea el bloque multiregistro o tras pulsar volver).	OK
El nombre del botón es Guardar	OK
<b>EDICIÓN</b>	
Cuando se entra en el bloque de edición, el cursor se tiene que posicionar en el primer campo editable.	OK
Cuando se pulsa el botón guardar, tiene que guardar los cambios y tiene que desaparecer el bloque de edición, excepto en las pantallas cabecera/líneas.	N/A
Cuando se pulsa la tecla de intro es como si hubieses pulsado el botón Crear. (excepto checks, combos y campos fecha).	OK

Aspecto a probar	Resultado
No se permite teclear datos con solo blancos. Se tiene que interpretar como nulos.	OK
El nombre del botón es Guardar. (Modificar)	OK
Tras editar el registro se ejecuta de nuevo la consulta en el multiregistro (donde se vea el bloque multiregistro o tras pulsar volver).	OK
<b>BORRADO</b>	
Se seleccionará en el multiregistro que registros se desea borrar y posteriormente se pulsará el botón de borrado..	OK
El borrado tendrá el mismo comportamiento que el buscar o el paginar (independiente del bloque de edición).	OK
Tras borrar el registro se ejecuta de nuevo la consulta.	OK
<b>AYUDA</b>	
Se ha actualizado el documento de ayuda de pantalla con los cambios realizados	N/A
<b>EXPORTACIÓN</b>	
Al pulsar "Exportar a Excel" comienza la descarga de un fichero .xls, en el fichero aparecerá el resultado de la búsqueda mostrado en el multiregistro de la pantalla	OK
Al pulsar "Exportar a PDF" comienza la descarga de un fichero .pdf, en el fichero aparecerá el resultado de la búsqueda mostrado en el multiregistro de la pantalla	OK

Taula 22- Proba gidoia (gaztelerako bertsioa)



## C ERANSKINA

### Erabiltzaile gida

Atal honetan aplikazioak (Fakturatzte-Moduluak) dituen funtzionalitateak bisualki azalduko dira.

#### Sarbidea lortu

Aplikazioa web-nabigatzaile batean exekutatu behar da. Beraz aplikazioa erabiltzeko web nabigatzaile bat ireki behar dugu eta zerbitzarira bidea bertan adierazi. Baina zein da zerbitzarirako bidea?

Bezero bakoitzak “bide” bat izango du. Beste hitzetan esanda, sarbidea izateko zerbitzariaren IP helbidea beharko da. Zerbitzaria berdina izan daiteke bezero guztientzat, baina bertan bezero bakoitzak bere “lekua” izango du. Hobeto ulertzeko ikus 70. irudia adibide gisa.

192.168.1.218:8080/inlog\_spia\_dev/autenticacion/autenticar.do

*Irudia 70 - URL sarbidea*

#### Kautotzea

Behin sarbidea lortu dugula, aplikazioaren funtzionalitateak erabiltzen hasteko *User* eta *Password* bat beharrezkoak izango dira. Hauek administratzaile rola duen beste erabiltzaile batek sortu behar dizkigu baina gida honekin jarraitzeko demagun *User* eta *Password* –a aurretik sortuta ditugula.

Erabiltzaile ID-a eta pasahitza dagozkien eremuetan idatzi, hizkuntza aukeratu dezakegu (hizkuntzaren balio lehenetsia Gaztelania izango da) eta Aceptar botoia sakatuko dugu.

Usuario\*   
 Clave\*   
 Idioma

*Irudia 71 – Kautotzea*

## Atalaren hautaketa

Pantaila nagusi gisa 72. irudia agertuko zaigu. Bertan aplikazioak dituen hiru moduluak agertuko dira: Biltegi, Segurtasuna eta Fakturatze moduluak, hain zuzen.

Aurreko ataletan aplikazioaren modulu bakoitzak zer egiten duen azaldu da. Gure kasuan Fakturatze modulua funtzionalitateak ikusiko ditugu, beraz, modulu hori aukeratuko dugu hurrengo pausura joan ahal izateko.

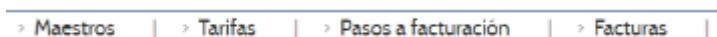


Irudia 72 - Aplikazioko atalak

Aplikazioko atalaren hautaketa egin ondoren *popup* bat pantailaratuko da. Leihotxo horretan zein instalazio eta zein biltegiaren gainean lan egingo dugun galdetuko digu. Horiek aukeratu ostean “Aceptar” botoia sakatu eta Fakturatze-modulu barruan egongo gara.

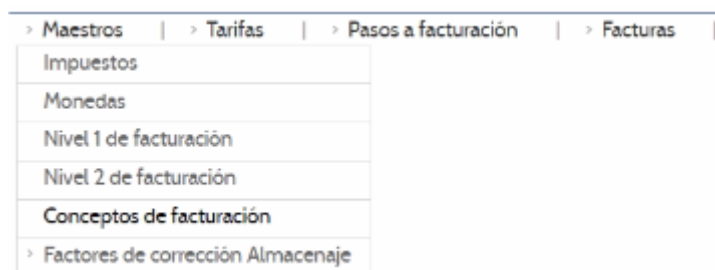
## Menu nagusia

Une honetan Fakturatze-moduluak duen menu nagusia ikusgarri izango dugu. (ikus 73. irudia).



Irudia 73 - Fakturatze-modulua: Menu nagusia

Funtzionalitateak ikusteko menuak eskaintzen digun hautabide bat aukeratuko dugu. Adibidez bide hau jarraituko dugu: Maestros > Conceptos de Facturación.



Irudia 74 - Conceptos de Facturación menu bidea



## Oinarrizko funtzionalitateak

Orain aplikazioak dituen oinarrizko funtzionalitateen erakustaldia egingo da.

### Bilaketa

Bilaketak bi atal nagusi ditu: bilaketa irizpideen arabera emaitzak lortu eta emaitzen pantailaratzea.

### Iragazpena

Bilaketa irizpideekin hasiko gara. Zertan datza hau? Bilaketa aukera duten pantaila bakoitzeko bilaketa irizpide batzuk egongo dira. Hau da, emaitzak lortzeko irizpide batzuk erabilia emaitza zehatzagoak lortu daitezke. Emaitza zehatzago horiek lortzeko informazio guztiaren artean iragazpen prozesu bat burutzen dela esan genezake.

*Irudia 75 - Bilaketa iragazpena*

75. irudian erakusten den moduan irizpide batzuk izango ditugu bilaketa egiteko. Zenbat eta irizpide gehiago idatzi, bilaketa orduan eta zehatzagoa izango da.

### Botoiak

Goiko irudian ikusten den moduan atal honek hiru botoi ditu (kasu gehienetan):

- “Crear”: Erregistro berri bat sortzeko.
- “Buscar”: bilaketa exekutatzeko (intro tekla sakatuz gero bilaketa exekutatu da baita ere).
- “Limpiar”: Eremuetan idatzitako bilaketa irizpideak hutsik utziko ditu.

### Emaitzak pantailaratu

Funtzionalitate honek ez dauka misterio askorik. Bilaketa irizpideak zeintzuk izan diren kontutan izango du aurkitutako informazioa bistaratzekoan.

Emaitzak taula batean adieraziko dira. (Ikus 76. irudia)

RESULTADO DE CONSULTA

Encontrados 32 datos. Se muestran del 1 al 8.

	Concepto Facturación	Denomi. Concepto	Nivel 1	Denominación	Nivel 2	Denominación
<input checked="" type="checkbox"/>	CAA	CAJAS ALMACENADAS1	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	CEN	CAJAS ENVIADAS	02	MANIPULACION	01	MANIPULACION ENVIOS
<input checked="" type="checkbox"/>	GEA	GENERICO ALMACENAJE	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	HAA	HUECOS ALMACEN OCUPADOS	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	HPA	HUECOS PICKING OCUPADOS	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	JRL	CONCEPTO FACTURACION JON	02	MANIPULACION	03	MANIPULACION DEVOLUCIONES
<input checked="" type="checkbox"/>	PAA	PALETAS ALMACENADAS	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	PEA	PESO ALMACENADO	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK

Exportar a Excel PDF Eliminar

Irudia 76 - Bilatzea: emaitzen bistaratzea

76. irudian emaitzen taula daukagu. Taula honek zutabe mota desberdinak ditu; lehenengo biak funtzionalitate “bereziak” dituzte, aurrerago azalduko den moduan, eta gainontzekoak lortutako informazioa erakusteko dira.

Gainera emaitzak orrialde desberdinetan zenbakituta erakutsiko dira.

Emaitzen taulan agertzen diren aipatutako lehenengo bi zutabeetan arreta jarriko dugu orain.

### Checkbox zutabea

Lehenengo zutabeen *checkbox* bat dago, hau nahi ditugun errenkadak aukeratzeko da. Pantaila gehienetan errenkaden hautaketa hau erregistro horiek ezabatzeko izaten da, baliteke beste pantaila batzuetan erregistroen hautaketa prozesu desberdin bat jarraitzeko izatea.

Gauden pantailan edozein errenkada aukeratu dugu, hau da, *checkbox* zutabeen sakatu.

	Concepto Facturación	Denomi. Concepto	Nivel 1	Denominación	Nivel 2	Denominación
<input checked="" type="checkbox"/>	AAA	DEMO PROBA	02	MANIPULACION	02	MANIPULACION LLEGADAS
<input checked="" type="checkbox"/>	CAA	CAJAS ALMACENADAS	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	CEN	CAJAS ENVIADAS	02	MANIPULACION	01	MANIPULACION ENVIOS
<input checked="" type="checkbox"/>	GEA	GENERICO ALMACENAJE	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	HAA	HUECOS ALMACEN OCUPADOS	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	HPA	HUECOS PICKING OCUPADOS	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK
<input checked="" type="checkbox"/>	JRL	CONCEPTO FACTURACION JON	02	MANIPULACION	03	MANIPULACION DEVOLUCIONES
<input checked="" type="checkbox"/>	PAA	PALETAS ALMACENADAS	03	ALMACENAJE STOCK	01	ALMACENAJE STOCK

Irudia 77 - Checkbox aukeratua

77. irudian ikusten denez gero, errenkada hori kolorez aldatu egin da.

### Eguneraketa zutabea

Orain arreta bigarren zutabeen jarriko dugu. Zutabe honetan arkatz berde bat izango dugu. Arkatz berde hau duten errenkada guztiak editagarriak izango dira eta honen gainean sakatuz gero, eguneraketa pantailara berbidaliko gaitu aplikazioak.

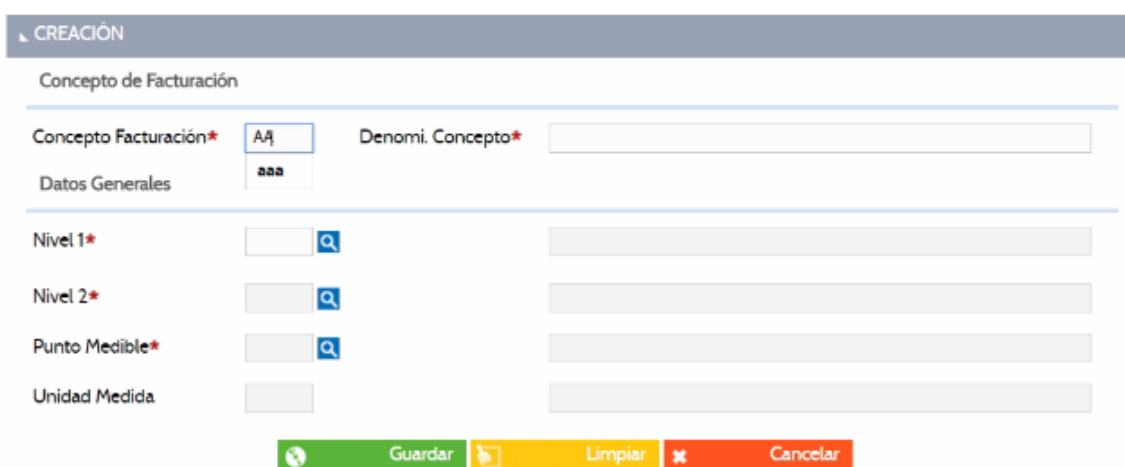
## Beste ezaugarri batzuk

Azaldutako atal nagusietaz aparte, bilaketa pantaila honek beste ezaugarri bat du: lortutako emaitzen esportazioa.

Emaitzen taula azpian “Exportar a “ aukera daukagu (76. irudian behealdean). Botoi hauetan sakatzerakoan bilaketaren emaitza guztiak .pdf edo .xls fitxategi batera esportatuko dira.

## Sortzea

Pantaila batzuk erregistroak sortzeko aukera izango dute. Aukera hau lehen aipatutako “Crear” botoiak emango digu.



CREACIÓN

Concepto de Facturación

Concepto Facturación\* AA Denomi. Concepto\*

Datos Generales aaa

Nivel 1\*

Nivel 2\*

Punto Medible\*

Unidad Medida

Guardar Limpiar Cancelar

Irudia 78 - Erregistro sortzea

Goiko irudian ikusten den moduan beste pantaila batera berbidaliko gaitu “Crear” botoiak. Pantaila honetan eremu batzuk agertuko dira, batzuk hutsik eta beste batzuk informazioarekin (adibide honetan guztiak hutsik daude).

Pantaila honen beheko aldean hiru botoi daude:

- “Guardar”: Erregistroa eratuko du. Dena ondo badao mezu bat agertuko zaigu, errore bat aurkitzekotan, aldiz, errorea pantailaratuko da mezu bat bezala. Dena ondo joan bada, hau da, erregistroa sortu bada, erregistroaren informazioa berriro erakutsiko da eremuetan baina oraingoan eguneraketa moduan izango da.
- “Limpiar”: Bilaketan bezala idatzitako eremuak hutsik utziko ditu.
- “Cancelar”: Atzera bueltatuko da. Bilaketa pantailara berbidaliko gaitu eta aurretik exekutututako kontsulta berriro ere exekututuko du emaitzak pantailaratu.

## Eguneraketa

Pantaila batzuk erregistroak editatzeko aukera emango dute. Eguneraketa aukera bi bide desberdin jarraituz lortu daiteke.

Sortze ataletan azaldu den moduan erregistroa errore barik sortu izanez gero, erregistro hori editatzeko aukera emango zaio erabiltzaileari.

Bigarren bidearen hasiera lehenago aipatu da baita ere. Bilaketa-emitzen taulan zutabe “berezi” bat geneukan, arkatz berde bat zuenak hain zuzen ere. Arkatz berde horrek eguneratze pantailara berbidaliko gaitu.

Irudia 79 - Erregistroa editatzea

79. irudian ikusten den moduan eguneratze pantaila honetan eremu batzuk agertuko zaizkigu. Batzuk editagarriak izango dira eta beste batzuk, aldiz, ez dira eskuragarri egongo. Hemen ere behealdean botoiak izango ditugu:

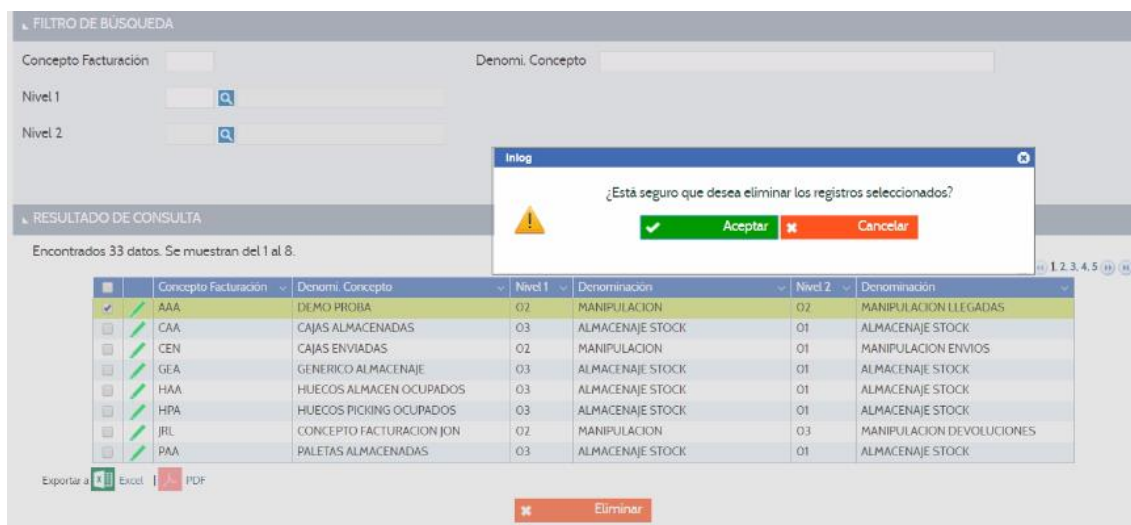
- “Modificar”: Erregistroa eguneratuko du. Dena ondo badao mezu bat agertuko zaigu, errore bat aurkitzekotan, aldiz, errorea pantailaratuko da mezu bat bezala. Dena ondo joan bada, erregistroaren informazioa berriro erakutsiko da.
- “Cancelar”: Atzera bueltatuko da. Bilaketa pantailara berbidaliko gaitu eta aurretik exekututako kontsulta berriro ere exekututako du emaitzak pantailaratuz.

### Ezabatzea

Bilaketa baten emaitzak pantailaratzen dituen pantaila batek, orokorrean, erregistroak ezabatzeko aukera emango digu.

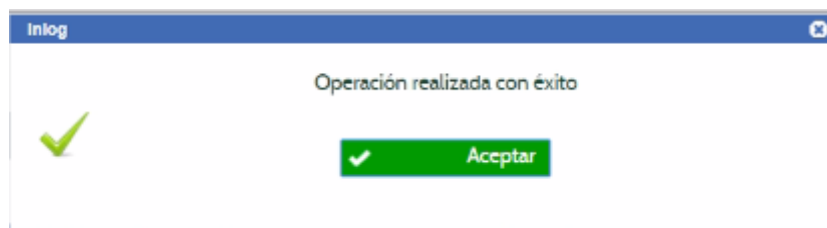
Ezabatze prozesua burutzeko soilik multi-erregistroko (emitzen taula) nahi ditugun erregistroak aukeratuko ditugu (*checkbox* –a dagoen zutabean, hau sakatu) eta taula azpialdean dagoen “Eliminar” botoia sakatuko dugu.

Orain 80.irudian ikusten den bezala mezu bat agertuko zaigu. Bertan aukeratutako erregistroen deuseztatze honekin ziur gauden ala ez galdetuko digu.



Irudia 80 - Ezabatze mezua

“Aceptar” sakatuz gero, hautatutako erregistroak ezabatuko dira mezua bat erakutsiz (ikus 81.irudia) eta “Cancelar” sakatzen badugu, berriz, mezua pantailatik kenduko da eta ez du ezer egingo.



Irudia 81 - Dena ondo joan denaren mezua

## Funtzionalitate bereziak

Azaldutako oinarritzko funtzionalitateetaz gain aipagarriak diren beste batzuk azalduko dira.

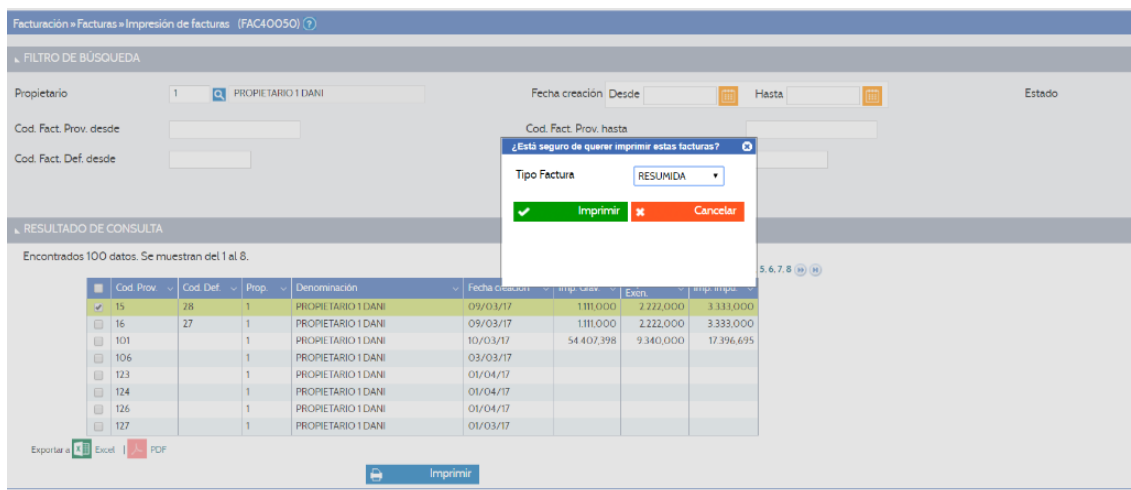
### Fakturen inprimatzea

Aplikazioak daukan aukera honekin Faktura baten informazioa inprimatu ahal izango dugu. Inprimatze hau .pdf bat sortuko du eta fitxategi honekin erabiltzaileak berak uste duena egin ahal izango du (fitxategia deskargatu eta gorde edo inprimagailu batetik atera, adibidez).

Inprimatze aukera hau erabiltzeko pantaila bide hau jarraituz dago: *Facturas > Impresión de Facturas* .

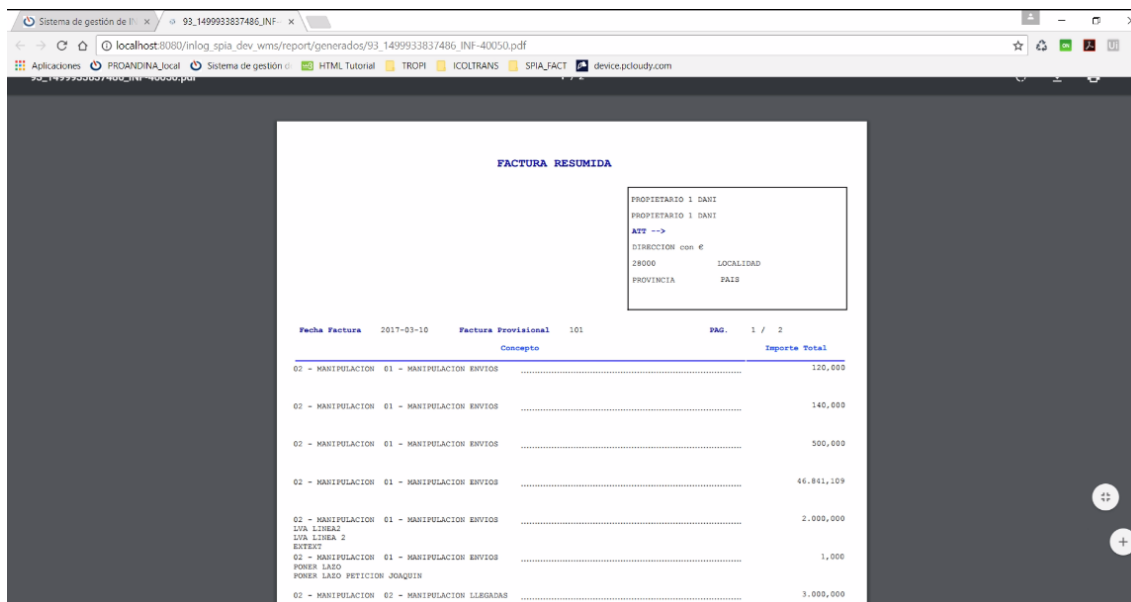
Bertan bilaketa pantaila agertuko da eta emaitzen taulan faktura bat edo bat baino gehiago aukeratu ahal izango ditugu haiek inprimatzeko. Ondoren “Imprimir” botoia sakatu eta

mezu bat agertuko da. Mezu horretan zein faktura-formatua inprimatu nahi dugun galdetuko zaigu. Bi aukera daude: laburtua (*Resumida*) eta zehaztua (*Detallada*) Ikus 82. irudia.



Irudia 82 - Faktura formatua (mezua)

Formatua aukeratu dugunean “Imprimir” botoia sakatzen badugu fakturak nabigatzailean agertuko dira (fitxa berri batean, ikus 83. irudia) eta “Cancelar” botoia sakatuz gero, ez du ezer egingo.



Irudia 83 - Faktura nabigatzailean

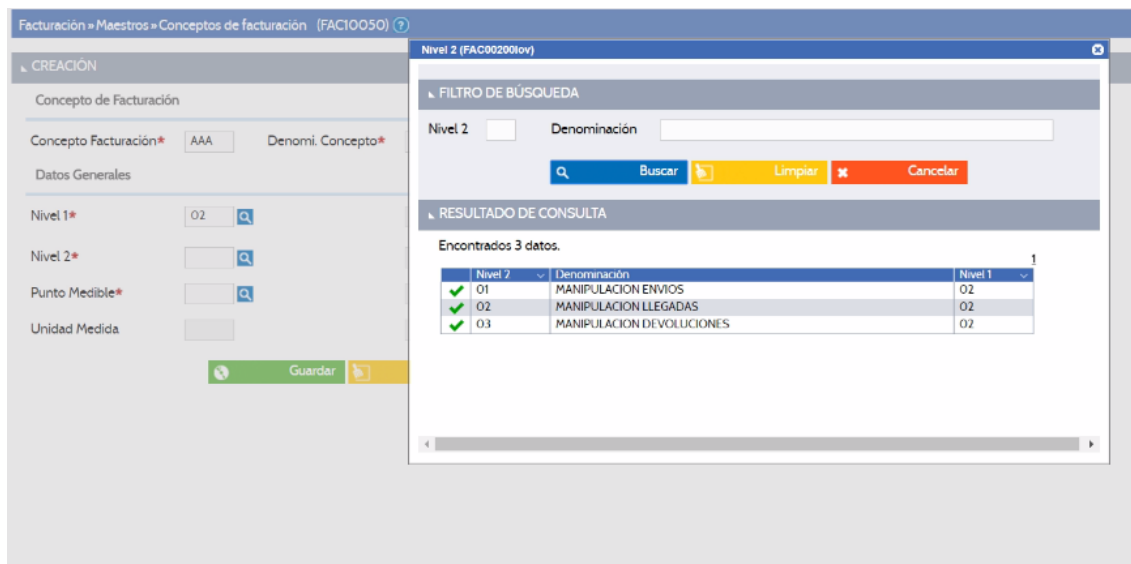
LOV leihok

Bilaketa, sortze eta eguneraketa pantailetan arreta jartzen badugu berriro ere, botoi batzuk zertarako direnen azalpena ez da eman. Botoi hauek lupa itxura dute, 84. irudian erakusten den moduan.



Irudia 84 - LOV botoiak

Honako botoia sakatzerakoan *popup* leiho bat pantailaratuko da (ikus 85. irudia). Leiho berezi honetan eremu baten informazioa bilatu ahal dezakegu. Adibidez, demagun erregistro bat sortzeko X eremu baten kodea behar dugula eta kode hori existitu behar dela (“Propietario” kodea behar izatekotan *propietario* hori aurretik existitu beharko da), baina erabiltzaile bezala ez dakizkigu guztien kodeak. Orduan LOV (*List of Values*) aukera sakatzen dugu eta leiho berezi horretan kode horren bilaketa exekutatu dugu.



Irudia 85 - LOV popup

Bilaketa exekutatuak emaitzak pantailaratuko dira eta emaitza horien artean erregistro bat aukeratu gero, pantaila nagusian dagokion eremuan idatziko da.





## E ERANSKINA

### Lan ingurunea prestatu

Atal honetan pertsona batek aplikazioaren garapenarekin jarraitu nahi izatekotan bere lan ingurunea nola eta zerekin prestatu beharko luke azalduko da.

### JDK instalatu (1.6.0\_45)

Ingurunea instalatzeko lehenengo pausua proiektuarekin bateragarria den Java JDK bat instalatzea izango da. Kasu honetan 1.6.0\_45 bertsioa erabiliko da.

JDK bertsioa instalatzeko Oracle –ren web orrialdera joango gara bertan, Java sailean *Downloads* –era joango gara. Gure ordenagailuarekin bateragarria den bertsioa deskargatuko dugu. Deskargatzen dugun bertsioa 32 edo 64 bitekoa den egiaztatzea garrantzitsua da, ondoren Eclipse bertsioa berdina izan behar delako.

Java SE Development Kit 6u45		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	65.46 MB	<a href="#">jdk-6u45-linux-i586-rpm.bin</a>
Linux x86	68.47 MB	<a href="#">jdk-6u45-linux-i586.bin</a>
Linux x64	65.69 MB	<a href="#">jdk-6u45-linux-x64-rpm.bin</a>
Linux x64	68.75 MB	<a href="#">jdk-6u45-linux-x64.bin</a>
Solaris x86	68.38 MB	<a href="#">jdk-6u45-solaris-i586.sh</a>
Solaris x86 (SVR4 package)	120 MB	<a href="#">jdk-6u45-solaris-i586.tar.Z</a>
Solaris x64	8.5 MB	<a href="#">jdk-6u45-solaris-x64.sh</a>
Solaris x64 (SVR4 package)	12.23 MB	<a href="#">jdk-6u45-solaris-x64.tar.Z</a>
Solaris SPARC	73.41 MB	<a href="#">jdk-6u45-solaris-sparc.sh</a>
Solaris SPARC (SVR4 package)	124.74 MB	<a href="#">jdk-6u45-solaris-sparc.tar.Z</a>
Solaris SPARC 64-bit	12.19 MB	<a href="#">jdk-6u45-solaris-sparcv9.sh</a>
Solaris SPARC 64-bit (SVR4 package)	15.49 MB	<a href="#">jdk-6u45-solaris-sparcv9.tar.Z</a>
Windows x86	69.85 MB	<a href="#">jdk-6u45-windows-i586.exe</a>
Windows x64	59.96 MB	<a href="#">jdk-6u45-windows-x64.exe</a>
Linux Intel Itanium	53.89 MB	<a href="#">jdk-6u45-linux-ia64-rpm.bin</a>
Linux Intel Itanium	56 MB	<a href="#">jdk-6u45-linux-ia64.bin</a>
Windows Intel Itanium	51.72 MB	<a href="#">jdk-6u45-windows-ia64.exe</a>

[Back to top](#)

*Irudia 86 - JDK bertsioa ORACLE web orrialdean*

[JDK deskargatzeko link -a](#)

## JBOSS instalatu

Orain zerbitzaria instalatu behar da. Horretarako JBOSS zerbitzaria hautatu da. Deskargatzeko JBOSS –en web orrialdera jo eta *Downloads* atalean 7.1 bertsioa hautatu eta .zip fitxategia deskargatu.

.zip fitxategia deskonprimatuko dugu. (Oso garrantzitsua da gogoratzea non egin dugun pausu hau!!!!).

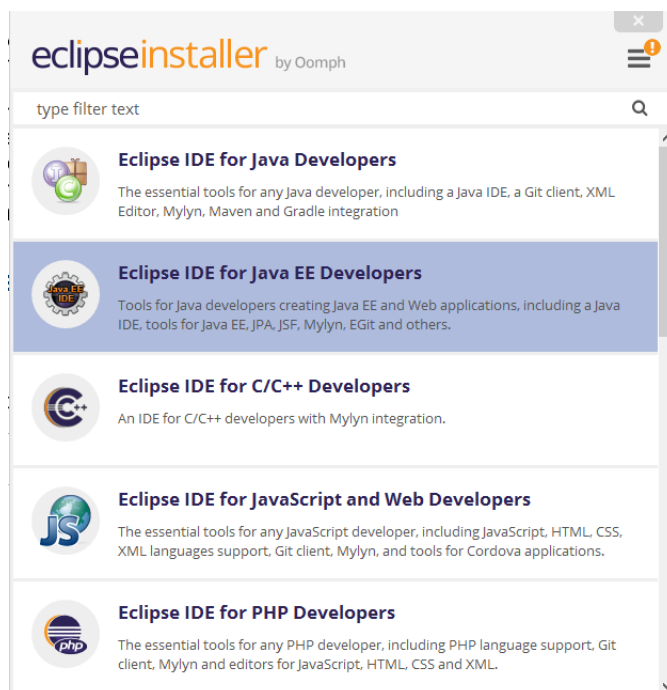
## Git errepositorioen kudeaketa

Errepositorioaren kudeaketa modu erraz batean kontrolatzeko TortoiseGit softwarea deskargatuko dugu bere web orrialdetik, *Downloads* ataletik hain zuzen.

## Eclipse IDE –aren konfigurazioa

### *Eclipse deskargatu*

Eclipse deskargatzeko haren web orrialdera joango gara eta *Downloads* ataletik instalatzailea deskargatuko dugu. Instalatzailea exekutatu dugu eta *Eclipse IDE for Java EE Developers* aukeratuko dugu.



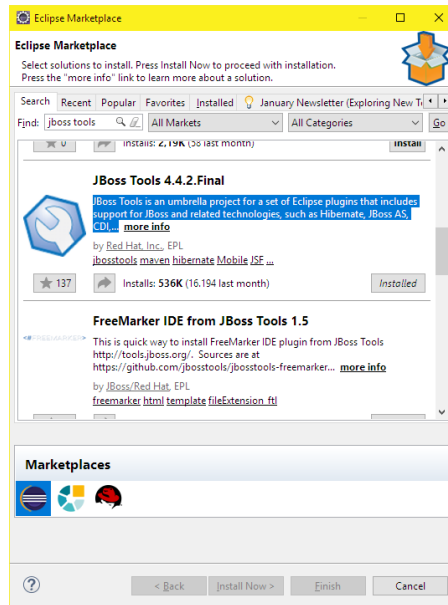
*Irudia 87 - Eclipse instalatzailea*

[Eclipse deskargatzeko link -a](#)

Programa ordenagailuko erro karpetan kokatuko dugu.

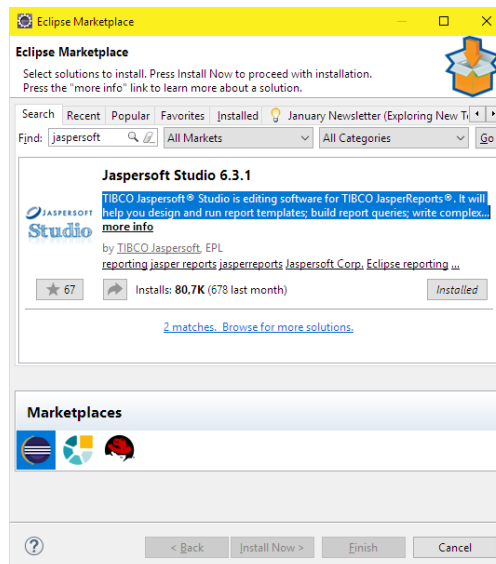
Plugin –ak

Garapenerako Eclipse –n bi *plugin* beharrezkoak izango dira. Horretarako Eclipse –k duen Marketplace –ra joko dugu (Help > Eclipse Marketplace). Bertatik JBoss Tools instalatuko dugu.



*Irudia 88 - JBoss Tools instalazioa MarketPlace-tik*

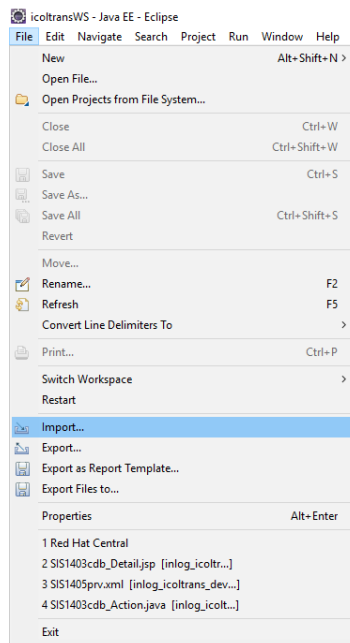
Ondoren txostenak editatzeko edo eratzeko erabiliko dugun erraminta deskargatuko dugu baita ere. Horretarako Jaspersoft Studio beharko dugu.



*Irudia 89 - Jaspersoft instalazioa MarketPlace-tik*

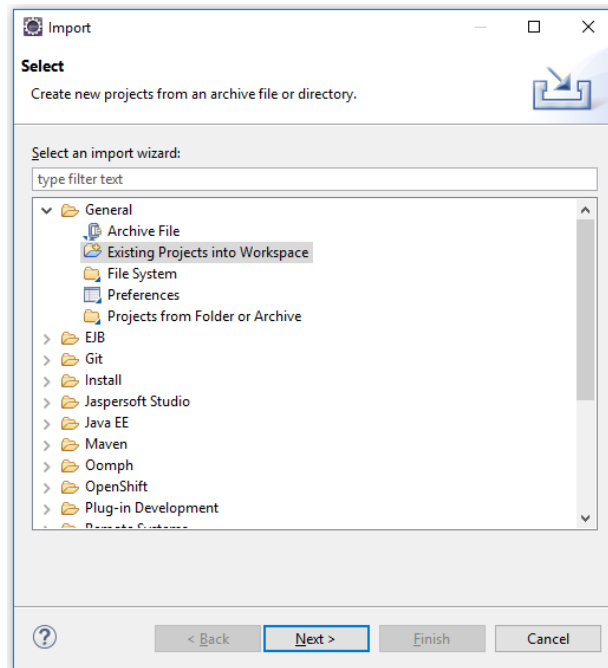
### Proiektua inportatu

Proiektu bat inportatzeko Git errepositoriotik klonatu beharko dugu. Klonatu ostean *File* > *Import* atalera joango gara.



Irudia 90 - Proiektua inportatu

*General* atala zabaldu eta bertan agertuko den *Existing Projects into Workspace* aukeratuko dugu.

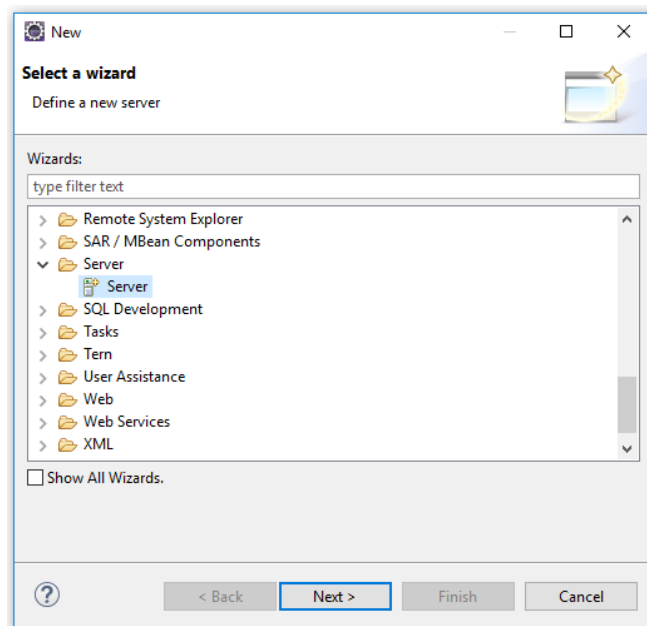


Irudia 91 - Existing Projects aukera

Git errepositorioak klonatu ditugun path –a idatziko dugu bertan.

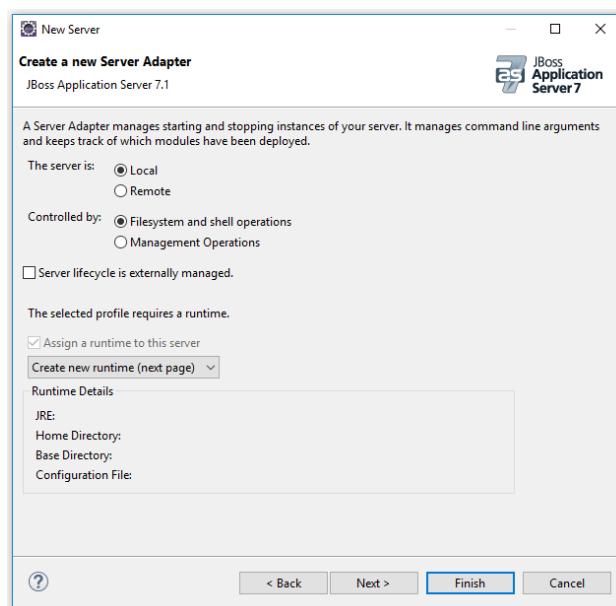
### Zerbitzariaren instalazioa

Jboss aplikazioen zerbitzaria instalatzeko *File > New > Other* atalera joko dugu eta bertan *Server* aukera hautatu.



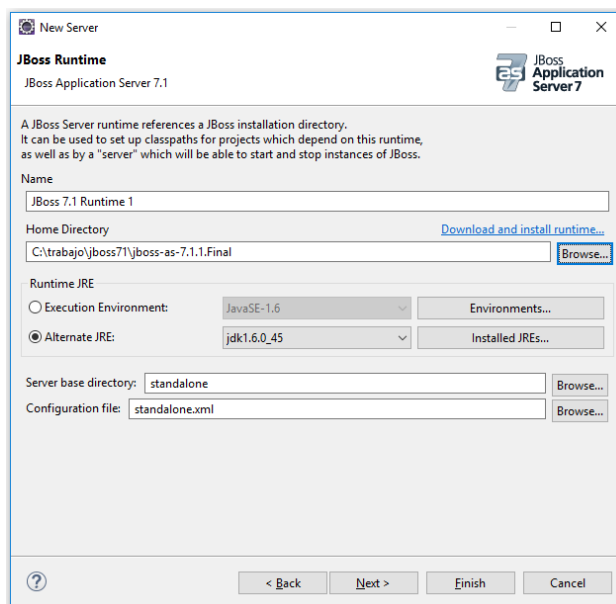
Irudia 92 - Zerbitzaria instalatzen

Jboss 7.1 aukeratuko dugu zerbitzari berria eratzekoan. Ondorengo pausuan *Create new Runtime* aukeratuko dugu eta “Siguiete” sakatu.



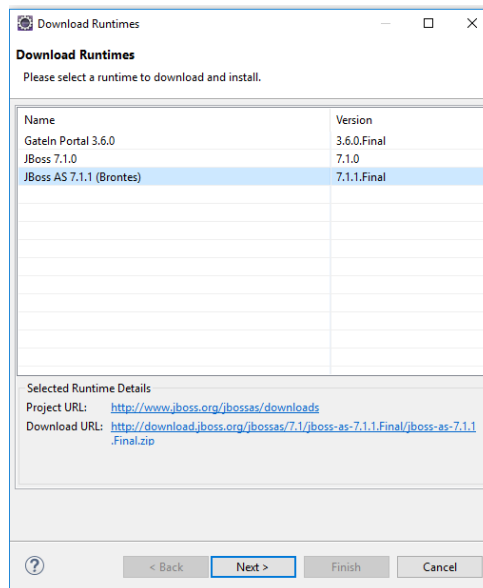
*Irudia 93 - Create new runtime*

Ondorengo pausuan zerbitzariaren izena eta kokatuta dagoen bidea adierazi beharko dugu. Alternate JRE atlean JDK bertsioa aukeratuko dugu baita ere (gure kasuan 1.6.0\_45).

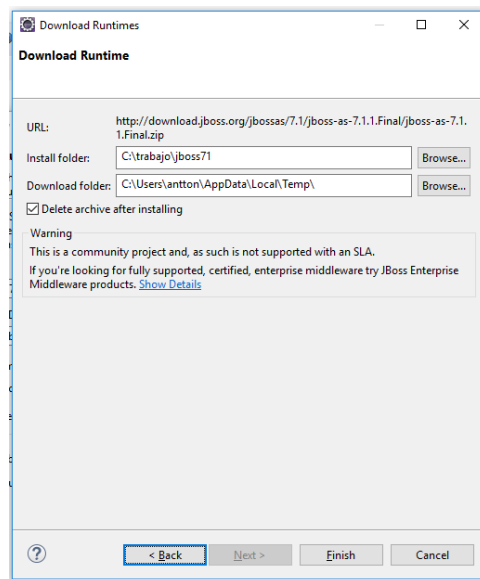


*Irudia 94 - JBoss eta JDK adierazten*

Jarraitu ahal izateko *runtime* –a deskargatu eta instalatu beharko dugu. Horretarako *Download and install runtime* –n sakatuko dugu (95. Irudia).

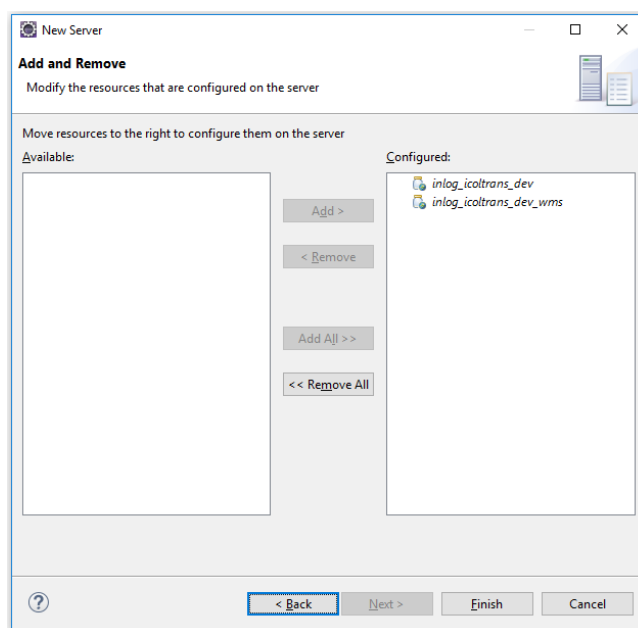


*Irudia 95- Runtime deskarga*



*Irudia 96 - Runtime deskarga*

Amaitzeko inportatuko proiektuak zerbitzarira gehituko ditugu.



Irudia 97 - Proiektuak zerbitzarian gehitu

### Zerbitzaria konfiguratu

Behin zerbitzaria instalatu dugula, datasource –ak erabiliko duten Oracle JDBC driver –a instalatzea beharrezkoa izango da. Horretarako JBOSS konfiguratzea derrigorrezkoa izango da.

Beharko dugun lehenengo gauza ojdbc6.jar Oracle driver –a izango da. Driver –a honako link hau jarraituz deskargatu daiteke: [JBOSS konfiguratzeko Oracle driver -a](#) .

Deskargatu ondoren JBOSS –en erro direktorioan dagoen *modules* karpetan kokatuko gara. oracle > jdbc direktorioak sortuko ditugu:

```
d:\trabajo\jboss71\jboss-as-7.1.0.Final\modules\oracle\jdbc\main\
```

Bertan ojdbc6.jar driver –a kopia eta *module.xml* fitxategi bat srotuko dugu jarraian dagoen edukiarekin:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="oracle.jdbc">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```



Hurrengo pausua modulu berri hau *standalone.xml* fitxategian kargatu beharko dugu. Fitxategi hau JBOSS –en *standalone > configuration* –en dago. Horretarako ondorengo edukia `<drivers>` atalean idatziko dugu:

```
<driver name="OracleJDBCDriver" module="oracle.jdbc">
<xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
</driver>
```

Ondoren *datasource* –rako erreferentzia gehitu beharko dugu `<datasources>` atalean.

```
<datasource jta="false" jndi-name="java:/jdbc/DS_Spia_Dev_Almacen" pool-name="DS_Spia_Dev_Almacen"
enabled="true" use-ccm="false">
<connection-url>jdbc:oracle:thin:@192.168.1.218:1521:INLOG</connection-url>
<driver-class>oracle.jdbc.OracleDriver</driver-class>
<driver>OracleJDBCDriver</driver>
<security>
<user-name>SPIA_DEV_SGA</user-name>
<password>SPIA_DEV_SGA</password>
</security>
<validation>
<validate-on-match>>false</validate-on-match>
<background-validation>>false</background-validation>
<background-validation-millis>0</background-validation-millis>
</validation>
<statement>
<prepared-statement-cache-size>0</prepared-statement-cache-size>
<share-prepared-statements>>false</share-prepared-statements>
</statement>
</datasource>
<datasource jta="false" jndi-name="java:/jdbc/DS_Spia_Dev_Seguridad" pool-name="DS_Spia_Dev_Seguridad"
enabled="true" use-ccm="false">
<connection-url>jdbc:oracle:thin:@192.168.1.218:1521:INLOG</connection-url>
<driver-class>oracle.jdbc.OracleDriver</driver-class>
<driver>ojdbc6.jar</driver>
<security>
<user-name>SPIA_DEV_SEG</user-name>
<password>SPIA_DEV_SEG</password>
</security>
<validation>
<validate-on-match>>false</validate-on-match>
<background-validation>>false</background-validation>
<background-validation-millis>0</background-validation-millis>
</validation>
<statement>
<prepared-statement-cache-size>0</prepared-statement-cache-size>
<share-prepared-statements>>false</share-prepared-statements>
</statement>
</datasource>
<datasource jta="false" jndi-name="java:/jdbc/DS_Spia_Dev_Fact" pool-name="DS_Spia_Dev_Fact"
enabled="true" use-ccm="false">
<connection-url>jdbc:oracle:thin:@192.168.1.218:1521:INLOG</connection-url>
<driver-class>oracle.jdbc.OracleDriver</driver-class>
<driver>OracleJDBCDriver</driver>
```

```
<security>
  <user-name>SPIA_FACT_SGA</user-name>
  <password>SPIA_FACT_SGA</password>
</security>
<validation>
  <validate-on-match>>false</validate-on-match>
  <background-validation>>false</background-validation>
  <background-validation-millis>0</background-validation-millis>
</validation>
<statement>
  <prepared-statement-cache-size>0</prepared-statement-cache-size>
  <share-prepared-statements>>false</share-prepared-statements>
</statement>
</datasource>
```