

SISTEMA DE GESTIÓ
D'UNA INTRANET PER A
INSTAL·LACIONS RIBA SL

MEMÒRIA DEL TFC-J2EE

Xavier Fernández Soler

ETIS

Professor: Jose Juan Rodríguez Aula TFC-J2EE: 4

14/01/2009

1. Dedicatòria i agraïments

Ja fa uns quants anys que em vaig inscriure a la UOC, per a poder trobar una sortida als meus estudis. Vaig deixar de cursar una carrera convencional, per realitzar-ne una de virtual. Tothom em deia que estava perdent els millors anys de la vida d'un jove a una Universitat presencial, que si campanes, festes, amics... Jo em resignava a pensar que estava perdent els anys, però mica en mica he anat observant que gràcies a aquesta metodologia d'estudi, he pogut avançar, madurar i aprendre molt d'aquesta vida.

Des d'un bon principi em vaig marcar unes pautes que si fa no fa, he anat seguint durant aquests sis llargs anys a la UOC. He pogut fer un munt d'amics i animar a d'altres a entrar-hi. Tot això m'ha enriquit molt, cosa que no pensava o creia que succeiria en el passat.

M'agradaria donar les gràcies als meus pares qui sempre han estat al meu costat animant-me a no decaure en els moments difícils.

Fa cosa de tres anys i mig, que vaig conèixer a la persona que m'ha estat acompanyant durant aquests últims anys d'estudi i que m'ha ajudat a ser més constant i meticulós en tot el que he estat fent. Això s'ha vist reflectit en les notes finals, fent que no baixes del Notable i ratllant la Matricula d'Honor. Aquesta TFC¹ va per tu Marta, gràcies per fer-me costat en els moments més difícils, gràcies per ser com ets.

No voldria deixar aquesta línia sense donar les gràcies al meu tutor, Pepe, del treball final de carrera, qui m'ha guiat perfectament en totes les entregues que s'han hagut de realitzar i m'ha donat "canya" en els moments d'acomodament durant el curs.

A per cert, i abans que se m'oblidi donar les gràcies als meus companys, socis d'empresa, Oriol, Àlex, Toni i Jordi, qui han entès i valorat la meva situació.

Gràcies a tots.

2. Resum

El treball que es presenta a continuació, és la culminació d'uns estudis i inici d'un esperançat futur. Intentant recopilar tots els coneixements adquirits en el procés d'aprenentatge en ETIS², Enginyeria Tècnica en Informàtica de Sistemes, així com també l'aplicació pràctica. Per tant, els que ens podem trobar d'aquí en endavant és un projecte informàtic, relacionat amb el món del J2ee³, passant per totes les etapes d'un projecte, i obviant fases, com pot ser el cost econòmic.

Aquest treball final de carrera, constà de tres parts ben diferenciades, que són el pla de treball, on podem trobar una manera més detallada les fases a desenvolupar ajustat al període de temps que es disposa per a complementar-lo. Un cop tenim definit el pla de treball, s'ha de començar a treballar sobre els requeriments per a construir el disseny, definint cada cas d'ús per l'usuari. Aquests, a alt nivell, serviran de base per a desenvolupar el codi i processos de baix nivell.

Deixant enrere la fase de disseny, ens trobem amb la fase d'implementació, on donarem a vida tot el que s'ha planificat anteriorment. Serà una etapa dura, ja que s'hauran de posar sobre la taula coneixements adquirits durant aquests últims anys, i adquirir-ne de nous, per a poder portar a bon terme aquest TFC.

Per finalitzar, haurem de passar un test unitari bàsic, que ens ajudarà a comprovar errors i possibles fases d'implementació a modificar. Part d'aquest test unitari, serà realitzat amb el client, per poder acabar d'enllestir el producte correctament.

En última instància, serà el tribunal el que valorarà el projecte i realitzarà el UAT⁴, Test d'Acceptació d'Usuari amb la documentació i *scripts*⁵ aportats.

Índex

1. Dedicatòria i agraïments	2
2. Resum.....	4
Índex.....	5
3. Memòria	8
3.1. Capítol 1, Introducció	8
3.1.1. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.	8
3.1.2. Objectius del TFC	9
3.1.2.1. Objectius de la assignatura	9
3.1.2.2. Objectius generals.....	9
3.1.2.3. Objectius específics	10
3.1.3. Enfoc metodològic	10
3.1.4. Planificació del projecte.....	11
3.1.4.1. Descomposició en fases, activitats i entregues.....	11
3.1.4.1.1. Descripció de les fases	12
3.1.4.2. Cronograma	13
3.1.5. Productes obtinguts.....	13
3.1.6. Breu descripció dels altres capítols de la memòria.	16
3.2. Capítol 2, Anàlisi i especificació de requeriments	16
3.2.1. Introducció.....	16
3.2.2. Requeriments funcionals del projecte	17
3.2.2.1. Actors del sistema.....	17
3.2.2.1.1. Administrador de l'aplicació	17
3.2.2.1.2. Gestors de l'aplicació.....	18
3.2.2.1.3. Operaris.....	18
3.2.2.2. Casos d'ús	19
3.2.3. Requeriments no funcionals del projecte.....	21
3.2.3.1. Rendiment.....	22
3.2.3.1.1 Volum de peticions	22
3.2.3.2. Usabilitat	23
3.2.3.2.1. Microcontinguts.....	23
3.2.3.2.2. Formularis	24
3.2.3.2.3. Nombre de Passos	24
3.2.3.2.4. Temps d'Espera	24
3.2.3.2.5. Look&Feel	24
3.2.3.3. Seguretat.....	25
3.2.3.3.1. Política de Seguretat	25
3.2.3.4. Fiabilitat.....	25
3.2.4. Disseny de l'Arquitectura de l'aplicació JEE.....	26
3.2.4.1. Introducció	26
3.2.4.2. Disseny de l'arquitectura de la plataforma	26
3.2.4.2.1. Patró de disseny	26
3.2.4.2.2. Arquitectura de la plataforma	27

3.2.4.2.2.1. Diagrama de classes	28
3.2.4.2.2.2. Diagrama de seqüència	28
3.2.4.3. Accés a dades.....	29
3.2.4.3.1. Base de Dades	30
3.2.4.3.2. Repositori de Fitxers	30
3.2.4.4. Model de dades.....	30
3.2.4.4.1. Model relacional.....	30
3.2.4.4.2. Diagrama ER	31
3.2.4.4.2.1. Diagrama	31
3.2.4.5. Altres aspectes de disseny	32
3.2.4.5.1. Logs.....	32
3.2.4.5.2. Configuració	33
3.2.4.5.3. Robustesa	34
3.2.5. Eleccions tecnològiques.....	34
3.2.5.1. Introducció	34
3.2.5.2. Framework de desenvolupament	34
3.2.5.2.1. Struts	35
3.2.5.3. DBMS.....	35
3.2.5.3.1. Requeriments.....	35
3.2.5.3.2. PostGreSQL	36
3.2.5.4. Servidor	36
3.2.5.4.1. Contenedor de Servlets.....	36
3.2.5.4.1.1. Apache Tomcat	36
3.2.5.5. Altres eleccions	37
3.2.5.5.1. Logs.....	37
3.2.5.5.2. Apache Tiles	37
3.2.5.5.3. DisplayTag	38
3.2.5.5.4. Fop.....	38
3.2.5.5.5. Configuració	38
3.2.5.5.5.1. Configurabilitat de l'Interfaç d'usuari	38
3.2.6. Interfície d'Usuari	39
3.2.6.1. Aparència	39
3.3. Capítol 3, Implementació i proves	40
3.3.1. Funcionament.....	40
3.3.1.1. Capa Web	40
3.3.1.2. Capa de Negoci.....	41
3.3.2. Requeriments de Software.....	41
3.3.3. Estat de l'aplicació	42
3.3.4. Decisions de Disseny i d'Implementació	43
3.3.5. Consideracions necessàries per a l'Instal·lació.....	44
3.3.5.1. Procés d'instal·lació	44
3.3.5.2. Base de Dades	45
3.3.5.2.1. Script de la Base de Dades	47
3.3.5.3. Configuració de l'aplicació	48
3.3.5.4. Estructura de Directoris	49
3.3.5.4.1. Logs.....	49
3.3.5.4.2. Full Feina.....	50

3.3.5.4.3. Factura	50
3.3.5.4.4. Imatges	50
3.3.6. Jocs de Prova	51
3.3.6.1. Usuari	51
3.3.6.2.1. Insets Clients	51
3.3.6.2.2. Insets Proveïdors	51
3.3.6.2.3. Insets Usuaris	52
3.4. Capítol 4, Valoració econòmica	52
3.5. Capítol 5, Conclusions	52
4. Glossari	55
5. Bibliografia	56
6. Annexes	57
6.1. Annex I: Tecnologies Avaluades	57
6.1.1. Arquitectura de les aplicacions web	57
6.1.1.1. Model 1	57
6.1.1.2. Model 2	58
6.1.1.3. Model 1 vs. Model 2	59
6.1.2. Frameworks de desenvolupament d'aplicacions web	59
6.1.2.1. Struts	59
6.1.2.1.1. Components	60
6.1.2.2. Java Server Faces	61
6.1.2.3. Struts vs. JSF	62
6.1.3. DBMS	62
6.1.3.1. PostgreSQL	62
6.1.3.2. MySQL	63
6.1.3.3. Elecció	63
6.2. Annex II: Informació ampliada dels casos d'ús del sistema	64
6.2.1. Casos d'ús	64
6.2.1.1. Log in i Log out	64
6.2.1.2. Client	65
6.2.1.3. Proveïdor	67
6.2.1.4. Factures	69
6.2.1.5. Full Feina	71
6.2.1.6. Fitxers de Log	74

3. Memòria

3.1. Capítol 1, Introducció

3.1.1. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.

Viure atrapats en el passat, o actualitzar-se i navegar a tota vela pel present. Avui en dia, qui més qui menys coneix Internet, moltes empreses s'han anat adequant al present sempre mirant al futur. Aquest és el cas de *Instal·lacions Riba S.L.*, que han cregut oportú relançar la seva imatge en un mar anomenat Internet.

Aquesta situació és idònia per a crear un portal on puguin estar present les millors qualitats d'aquesta empresa, i intentar fer-se un forat al món virtual, tal i com s'ha anat fent durant aquest anys en el món laboral, i aconseguir d'aquesta manera deixar petjada en el segle XXI.

La finalitat principal d'aquest portal, serà donar una sèrie d'eines pels treballadors, perquè puguin desenvolupar la seva feina d'una manera més ordenada i senzilla, i intentant focalitzar i centralitzar tota la informació.

Aquest tipus d'empresa treballa amb proveïdors i clients de tota mena, fent feines a gran o petita escala, i lògicament això reclama una gestió de continguts que permeti a l'empresa actualitzar còmodament les dades de cada client o proveïdor .

Un altre punt bàsic d'aquest portal és la comunicació entre els treballadors i gestors, és a dir, cada dia es duen a terme feines que han de quedar registrades, per conseqüentment realitzar-ne la facturació. El portal disposarà d'un apartat per als treballadors on, diàriament o setmanalment , ompliran amb les tasques realitzades durant la jornada de treball, i facilitat així, la feina a la part gestora.

Aquest requeriments fan absolutament imprescindible una gestió d'usuaris que, segons el perfil, puguin accedir en major o menor grau a la gestió de continguts.

3.1.2. Objectius del TFC

3.1.2.1. Objectius de la assignatura

Els objectius de l'assignatura són:

- L'estudiant realitzi un treball de síntesi dels coneixements apresos en les assignatures de la carrera i posar-los en pràctica.
- L'estudiant realitzi un treball teòric-pràctic i vinculat a la informàtica.
- L'estudiant profunditzi en el coneixement de la plataforma J2ee.

3.1.2.2. Objectius generals

L'abast del nostre projecte és el desenvolupament d'un sistema d'informació, és a dir, d'una aplicació Java empresarial, que s'adapti a les necessitats de l'empresa sol·licitant.

L'empresa es dedica al món de l'Electricitat – Lampisteria – Climatització, i ens ha sol·licitat ajuda per a poder gestionar més còmodament l'empresa.

Els punts que volem oferir a l'empresa són:

- Emmagatzemament de clients
 - Crear un client.
 - Modificar un client.
 - Eliminar un client.
 - Llistar un client o clients.

Poder emmagatzemar els clients de l'empresa, poder disposar de les dades més importants per a poder gestionar la facturació amb més facilitat, poder crear llistats en format pdf o excel, poder actualitzar les dades fàcilment...

- Emmagatzemament de proveïdors
 - Crear un proveïdor.
 - Modificar un proveïdor.
 - Eliminar un proveïdor.
 - Llistar un proveïdor o proveïdors.

Poder emmagatzemar els proveïdors de l'empresa, poder crear llistats en format pdf o excel, poder actualitzar les dades fàcilment...

- Emmagatzemament dels fulls de feines
 - Crear un full de feina.
 - Modificar un full de feina.
 - Eliminar un full de feina.
 - Llistar un full de feina.

Poder estar al dia amb els treballadors, disposant al cap del dia del full de feina realitzat; facilitat a l'hora de crear/modificar/eliminar qualsevol full de feina; llistar la feina realitzada a casa d'un client, per consegüentment realitzar-ne la facturació.

- Facturació
 - Crear una factura.
 - Modificar una factura.
 - Eliminar una factura.
 - Llistar una factura o més factures.

Realitzar les factures amb més comoditat, i poder-les modificar en qualsevol moment, sense necessitat de crear-ne una de nova; llistar les factures realitzades a un client al cap de l'any.

Tots aquests punts els explicarem més àmpliament en la secció dels casos d'ús.

3.1.2.3. Objectius específics

El nostre TFC serà el resultat de treballar les següents tasques:

- Configurar el servidor d'aplicacions J2ee (Tomcat⁶).
- Crear i configurar el servidor de base de dades, així com una base de dades relacional pertinent (PostgreSQL⁷).
- Configurar el nostre entorn de desenvolupament (Eclipse⁸).
- Ús de les API's⁹ necessàries per al desenvolupament del projecte.
- Ús de l'arquitectura o framework Struts¹⁰ i així aprofundir en el patró MVC¹¹ (Model Vista Controlador).
- Ús de l'API JDBC¹², per a connectar amb la BD¹³.
- Ús de llibreries de tercers per a generar documents en diferents formats.
- Control d'errors amb la biblioteca *log4j*¹⁴.
- Multidioma (català i castellà).
- Data Access Object(DAO¹⁵), interfaç comú entre l'aplicació i la base de dades.

Tota l'aplicació estarà realitzada amb el llenguatge de programació Java¹⁶, realitzant un anàlisi i disseny orientat a objectes.

3.1.3. Enfoc metodològic

Donat que és un projecte de desenvolupament d'un sistema per a un client i que ens podem trobar que algun dels requeriments ens impliqui aplicar més temps del que tenim estipulat en el TFC, hem cregut oportú utilitzar un cicle de vida clàssic, o també anomenat en "cascada".

Durant la fase d'anàlisi, més concretament en els requeriments funcionals, tindrem un coneixement més exacte del que el client ens ha demanat pel seu sistema i així aconseguir realitzar una estimació més ajustada i poder decidir quins dels requeriments romandran dins del projecte desenvolupat en el TFC, i quins es quedaran fora de l'abast de projecte, de cara a futures ampliacions.

Considerem que la tria d'aquest cicle de vida, ens permetrà poder assolir les nostres fites i les exigències de l'assignatura.

3.1.4. Planificació del projecte

Com a conseqüència que el projecte segueix un cicle de vida en cascada, les activitats s'agruparan en fases, que utilitzaran com a entrada els productes generats durant la fase anterior i proporcionaran com a sortida nous productes que seran utilitzats en les posteriors fases.

En hem basat en el mètode de l'avaluació continuada de l'assignatura, per a fixar-nos unes fites (entregues) que ens marcaran el ritme de treball.

Donat que no disposem de tot el temps que ens agradaria, hem trobat oportú, crear un pla de treball continu durant tot el trimestre, i lògicament existiran dies en els quals

es treballarà més que d'altres (caps de setmana i festius), per tant, no ens marcarem cap quantitat d'hores dedicades a cada tasca.

Per a realitzar la planificació, hem utilitzat *Microsoft Project 2007*¹⁷, amb el qual realitzarem el control i seguiment necessari.

3.1.4.1. Descomposició en fases, activitats i entregues

	Nombre de tarea	Duración	Comienzo	Fin
1	<input type="checkbox"/> Fase arrencada del projecte	10 días?	dom 21/09/08	mar 30/09/08
2	Descripció del TFC	5 días?	dom 21/09/08	jue 25/09/08
3	Objectius generals i específics	3 días?	vie 26/09/08	dom 28/09/08
4	Planificació prevista amb fites i temporalització	2 días?	lun 29/09/08	mar 30/09/08
5	Entrega Pac1	1 día?	mié 01/10/08	mié 01/10/08
6	<input type="checkbox"/> Fase d'Anàlisi	15 días?	mié 01/10/08	vie 17/10/08
7	Requeriments Funcionals	4 días?	mié 01/10/08	sáb 04/10/08
8	Requeriments no Funcionals	3 días?	dom 05/10/08	mar 07/10/08
9	Diagrama dels Casos d'ús	8 días?	mié 08/10/08	vie 17/10/08
10	<input type="checkbox"/> Fase del Disseny	16 días?	sáb 18/10/08	mar 04/11/08
11	Estudi i disseny de l'Arquitectura de l'aplicació JEE	5 días?	sáb 18/10/08	jue 23/10/08
12	Disseny de l'Entitat-Relació	4 días?	vie 24/10/08	lun 27/10/08
13	Disseny del Diagrama de Classes	4 días?	mar 28/10/08	vie 31/10/08
14	Disseny de l'interfície del portal	3 días?	sáb 01/11/08	mar 04/11/08
15	Entrega Pac2	1 día?	mié 05/11/08	mié 05/11/08
16	<input type="checkbox"/> Fase d'Implementació	30 días?	mié 05/11/08	mar 16/12/08
17	Preparació de l'entorn de construcció i reproducció	2 días?	mié 05/11/08	jue 06/11/08
18	Generació del pla d'implementació	2 días	vie 07/11/08	lun 10/11/08
19	Implementació del codi	23 días?	mar 11/11/08	jue 11/12/08
20	Proves	3 días?	vie 12/12/08	mar 16/12/08
21	Entrega Pac3	1 día?	mié 17/12/08	mié 17/12/08
22	<input type="checkbox"/> Fase de la Memòria	22 días?	mié 17/12/08	mar 13/01/09
23	Finalització de la Memòria	9 días?	mié 17/12/08	sáb 27/12/08
24	Elaboració de la presentació	9 días?	dom 28/12/08	mié 07/01/09
25	Presentació Virtual	4 días?	jue 08/01/09	mar 13/01/09
26	Entrega Memòria Final	1 día?	mié 14/01/09	mié 14/01/09

3.1.4.1.1. Descripció de les fases

Fase d'arrencada del projecte

Aquesta fase ens ajudarà a delimitar i definir el projecte en quant a àmbit, abast, objectiu, infraestructura, organització i planificació.

Com a resultat d'aquesta fase, extraurem l'entrega de la pac1, és a dir, el nostre pla de treball.

Fase d'anàlisi del projecte

Aquesta fase persegueix l'intenció d'extreure l'especificació detallada del sistema, és a dir, els requeriments necessaris que ens serviran com a base per la següent fase.

Fase de disseny del projecte

Aquesta fase persegueix la definició de l'arquitectura i l'entorn tecnològic del sistema.

Com a resultat d'aquestes dues fases, extraurem l'entrega de la pac2.

Fase d'implementació del projecte

Aquesta fase persegueix la finalitat de preparar, codificar i provar l'implementació del portal conforme el disseny i així poder donar peu a l'última fase del projecte

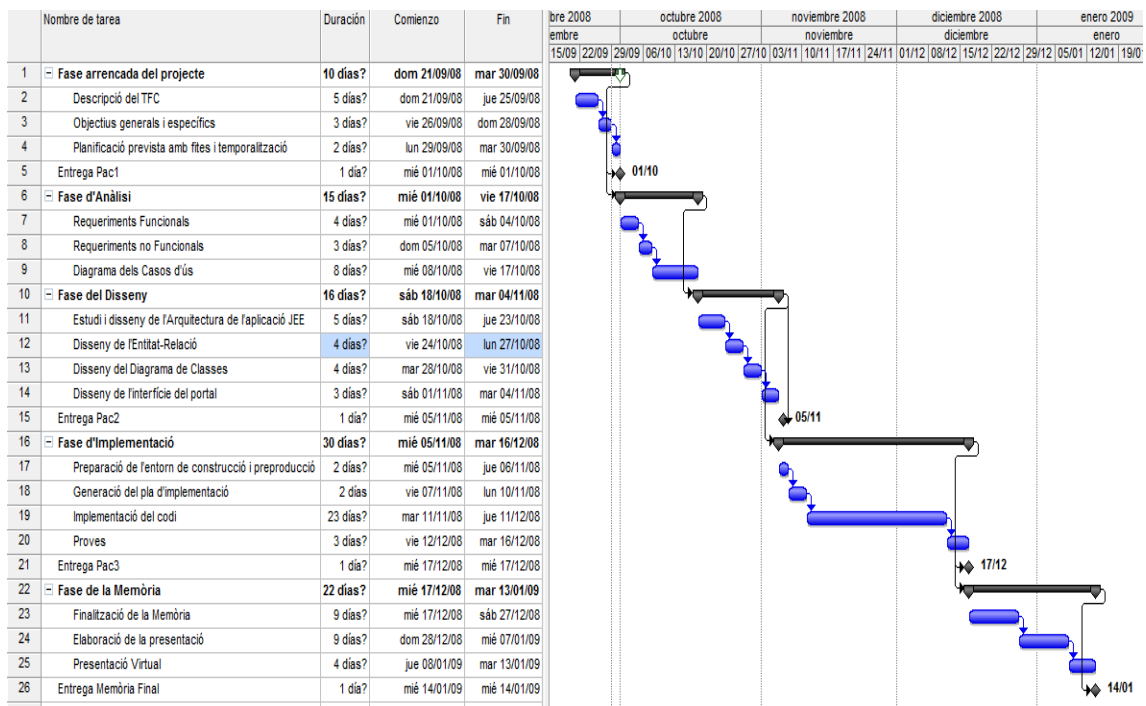
Com a resultat d'aquesta fase, extraurem l'entrega de la pac3.

Fase de la Memòria

Aquesta fase persegueix realitzar les últimes pinzellades del projecte, realitzar les conclusions, la memòria final del projecte, l'empaquetat i entrega del projecte.

Com a resultat d'aquesta fase, extraurem l'entrega final del TFC.

3.1.4.2. Cronograma



3.1.5. Productes obtinguts

A continuació es detalla el producte i com es pot procedir a la seva instal·lació.

En primera instància hem de disposar d'una BBDD¹⁸ per emmagatzemar les dades i un contenidor de servlets on deployarem¹⁹ el nostre fitxer war²⁰ resultant. Un cop tinguem la base de dades i el servidor d'aplicacions funcionals, simplement haurem d'obrir el nostre navegador preferit i teclejar la ruta on tenim deployat el producte.

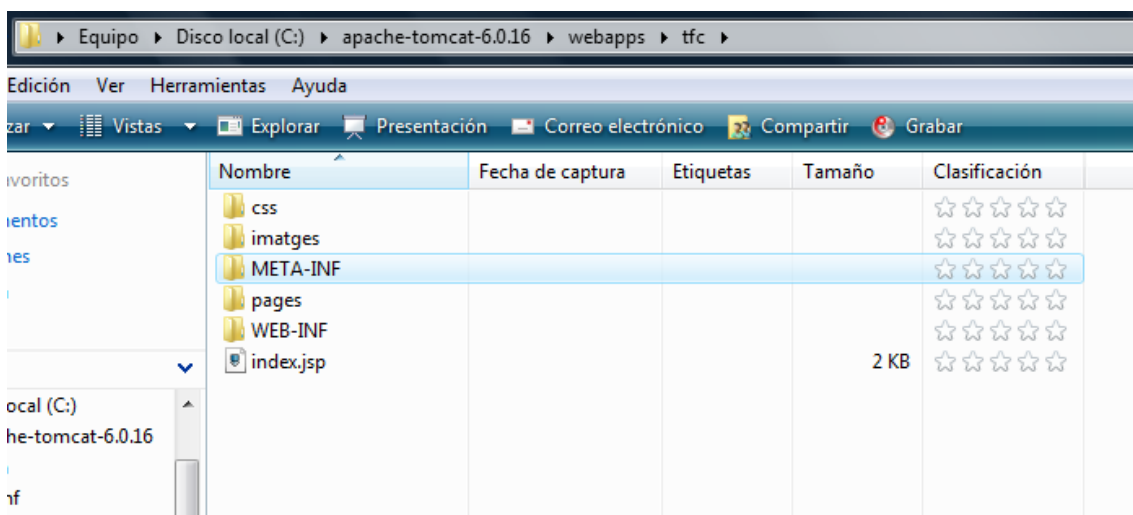
Com a punt de partida, el servidor d'aplicacions estarà instal·lat en la màquina local del client, i a la llarga s'intentarà poder disposar un servidor on poder migrar el producte.

A conseqüència de la manca de temps, de moment el producte només constarà d'un usuari, que serà l'administrador, amb el qual l'usuari gestor podrà realitzar totes les tasques necessàries. En properes modificacions, es podran crear més usuaris, i cadascú d'ells podrà realitzar les seves tasques, tal i com havíem especificat anteriorment.

Després d'aquest incís i reprenent el fil de paràgrafs anteriors, l'aplicació serà un fitxer amb extensió war, el qual estarà allotjat a la següent ruta, en local de moment, de la màquina del client:

“C:\apache-tomcat-6.0.16\webapps\tfc\”

on a dins podem observar la següent estructura de directoris:



Seguidament, detallarem que podem trobar dins de cada carpeta o directori.

Dins del directori `\css\` trobarem un fitxer amb extensió “css²¹” on podem trobar l'estil que disposarà el producte generat, és a dir, l'estil de lletra, de caixes de text, de fons de pantalla, de les pestanyes...

Dins del directori `\imatges\` podem trobar, totes les imatges que s'han fet servir per a generar l'aplicació empresarial del nostre client.

Dins del directori `\META-INF` com a fitxer més important trobarem el “*context.xml*”, que és el fitxer que haurem de modificar si volem canviar els paràmetres de la BD, ja sigui usuari, contrasenya o la direcció on apunta aquesta.

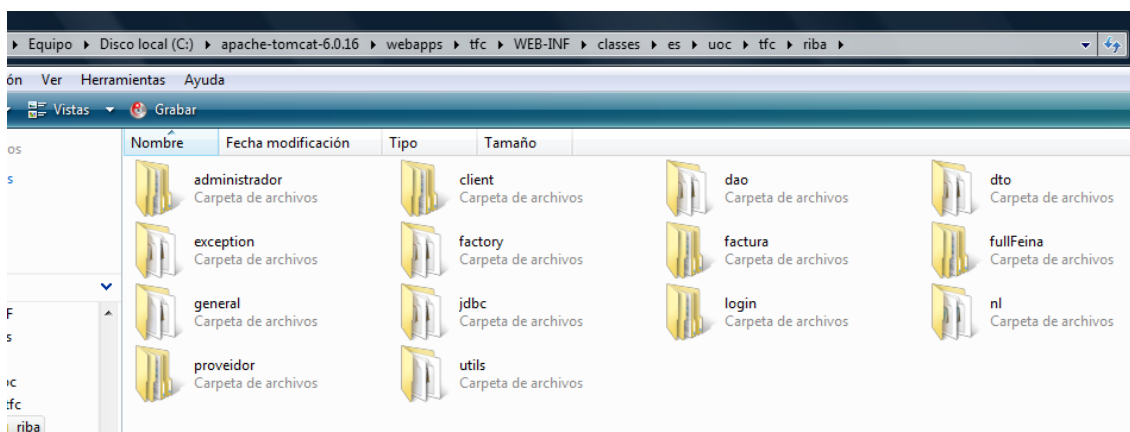
Dins del directori `\pages` disposarem de totes les pàgines amb extensió “*jsp*”²² que té el portal dissenyat.

L'últim directori que podem apreciar, `\WEB-INF`, és pot ser el més important de tots, ja que en ell trobarem, per exemple, els fitxers de configuració:

- `struts-config.xml`.
- `tiles-defs.xml`.
- `web.xml`.
- `displaytag_ca.properties` a `\WEB-INF\classes`.
- `displaytag_es.properties` a `\WEB-INF\classes`.

A més a més, a `\WEB-INF\lib`, trobarem totes les llibreries externes que fa ús la nostre aplicació.

I finalment a `\WEB-INF\classes\es\uoc\tfc\riba`, trobarem la següent estructura amb les classes del codi java generat:



Podem observar els diferents mòduls:

- *Administrador.*
- *Client.*
- *Factura.*
- *Full Feina.*
- *Login.*
- *Proveedor.*

Que tindran la següent estructura de directoris:

action	17/12/2008 18:41	Carpeta de archivos
beans	17/12/2008 18:41	Carpeta de archivos
bussinesObject	17/12/2008 18:41	Carpeta de archivos
forms	17/12/2008 18:41	Carpeta de archivos
modelObject	17/12/2008 18:41	Carpeta de archivos

A `\\WEB-INF\classes\es\uoc\tfc\riba\general\`, podrem trobar el fitxer de constants del portal.

A `\\WEB-INF\classes\es\uoc\tfc\riba\i\`, el fitxer de propietats amb els dos idiomes predefinits (espanyol i català).

A `\\WEB-INF\classes\es\uoc\tfc\riba\utils\`, les funcions java més usades del producte.

I la resta de directoris, estan relacionats amb els fitxers DAO, per a donar connexió entre el codi java i la BD.

3.1.6. Breu descripció dels altres capítols de la memòria.

La resta de capítols que manquen per veure són:

- **Fase d'anàlisi i disseny.**
On podrem trobar els requeriments funcionals i no funcionals, el casos d'ús, disseny d'entitat-relació i el disseny del diagrama de classes, d'entre altres.
- **Fase d'implementació i proves.**
On trobarem la lògica de negoci, requeriments de software, decisions de disseny, estat de l'aplicació, consideracions per a l'instal·lació i el joc de proves.

3.2. Capítol 2, Anàlisi i especificació de requeriments

3.2.1. Introducció

El primer pas que hem de donar abans de començar a dissenyar un sistema és definir i acotar quines funcionalitats ha de realitzar. En el projecte que ens ocupa implementarem una eina de gestió, de cara a digitalitzar i optimitzar el sistema existent.

Per això, hem de conèixer tot els processos que haurà de gestionar la nostre aplicació, quins actors els intervindran i com haurem de respondre en tot moment i en front a possibles eventualitats. També haurem d'aclarar un altre tipus de característiques com per exemple quant tardarem en realitzar una determinada operació o quants usuaris podran treballar amb l'aplicació de forma simultània.

Abans per això, s'ha partit de la base que el nostre projecte ha passat per l'etapa de l'estudi de viabilitat, i que aquesta ha estat aprovada, i per tant, és viable pel seu conseqüent desenvolupament.

3.2.2. Requeriments funcionals del projecte

Arrel de les reunions tècniques que hem concertat amb el client, hem pogut extreure una sèrie de necessitats, que seguidament desglossarem:

- ❖ El portal ha de disposar d'una plana d'inici per a treballadors de l'empresa, on amb un usuari i contrasenya tindran accés.
- ❖ Els gestors han de poder donar d'alta un client, modificar les seves dades, eliminar-lo si és necessari i poder extreure llistats dels clients.
- ❖ Els gestors han de poder donar d'alta un proveïdor, modificar les seves dades, eliminar-lo si és necessari i poder extreure llistats dels proveïdors.
- ❖ Els gestors han de poder crear una factura per un client, modificar les dades d'una factura, eliminar-la si és necessari i poder llistar una o totes les factures per client o per tots els clients.
- ❖ Els gestors han de tenir accés a tots els fulls de feines creats pels treballadors, i poder modificar o eliminar-los.
- ❖ Els treballadors han de poder crear, modificar i llistar els fulls de feines creats.
- ❖ L'administrador, ha de poder accedir a tots els casos d'ús del sistema i a més a més disposar d'una secció on podrà observar al fitxers de log les traces generades.

3.2.2.1. Actors del sistema

No tots els usuaris que accediran al sistema ho faran per a realitzar les mateixes operacions ni tindran els mateixos permisos. Seguidament, es detallen les funcions a desenvolupar per cada usuari en el context de l'aplicació:

3.2.2.1.1. Administrador de l'aplicació

Serà un superusuari encarregat del manteniment del portal, supervisant tot el moviment que hi ha en el portal mitjançant els fitxers de log generats dinàmicament. En futures actualitzacions, s'encarregarà, també, de donar d'alta i baixa els correus electrònics de cada usuari.

3.2.2.1.2. Gestors de l'aplicació

Són els encarregats de crear, modifica, eliminar i llistar clients, proveïdors, fulls de feina i factures. Disposaran dels privilegis necessaris per a gestionar la seva feina correctament.

3.2.2.1.3. Operaris

Són els encarregats de crear o modificar el full de feines. Tindran l'accés restringit a la resta de zones de l'aplicació.

Per a futures actualitzacions del sistema, es crearà un portal web amb accés des d'Internet, d'on qualsevol usuari no registrat o visitant podrà navegar pel portal en busca d'informació o realitzar un contacte amb l'empresa o si es tracta d'un treballador de l'empresa, podrà accedir a gestionar els fulls d'empresa i no haver de fer-ho localment des de l'empresa.

3.2.2.2. Casos d'ús

Administrador:

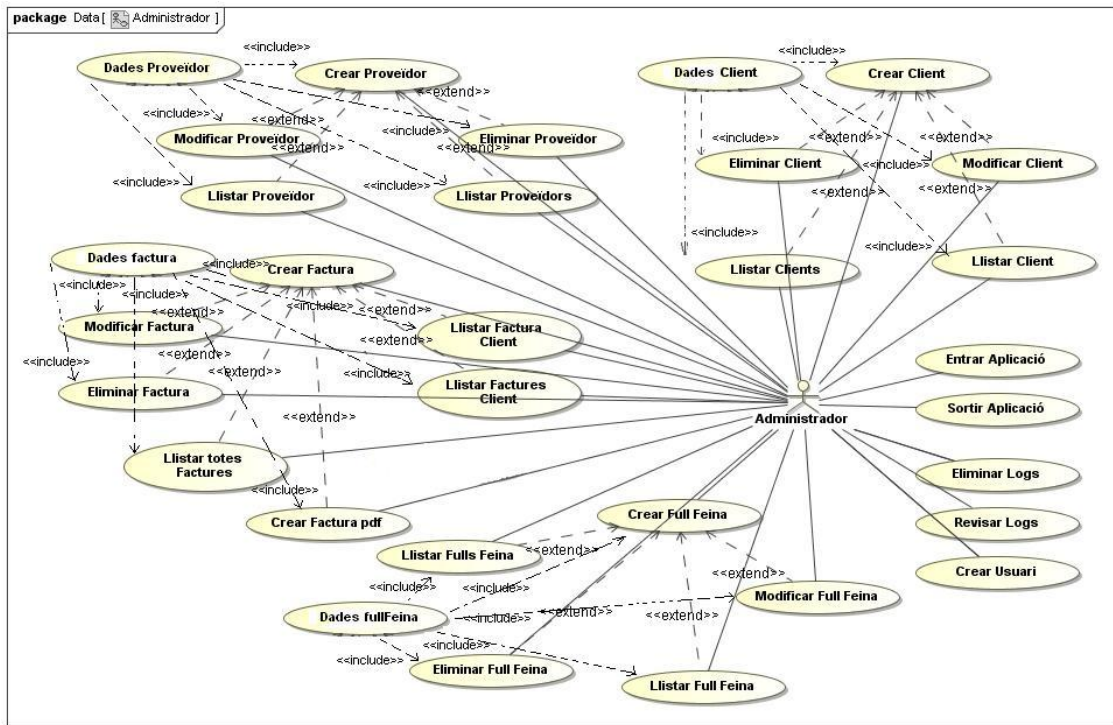


Fig 2.1 Casos d'ús de l'usuari Administrador

Gestor:

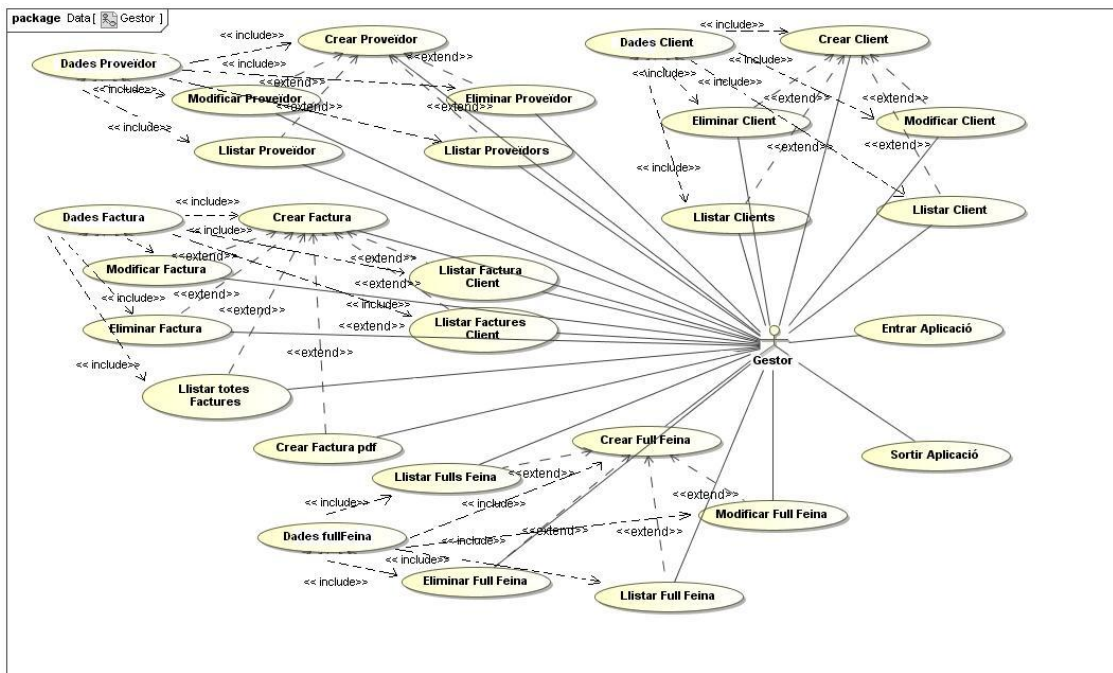


Fig 2.2 Casos d'ús de l'usuari Gestor

Operari:

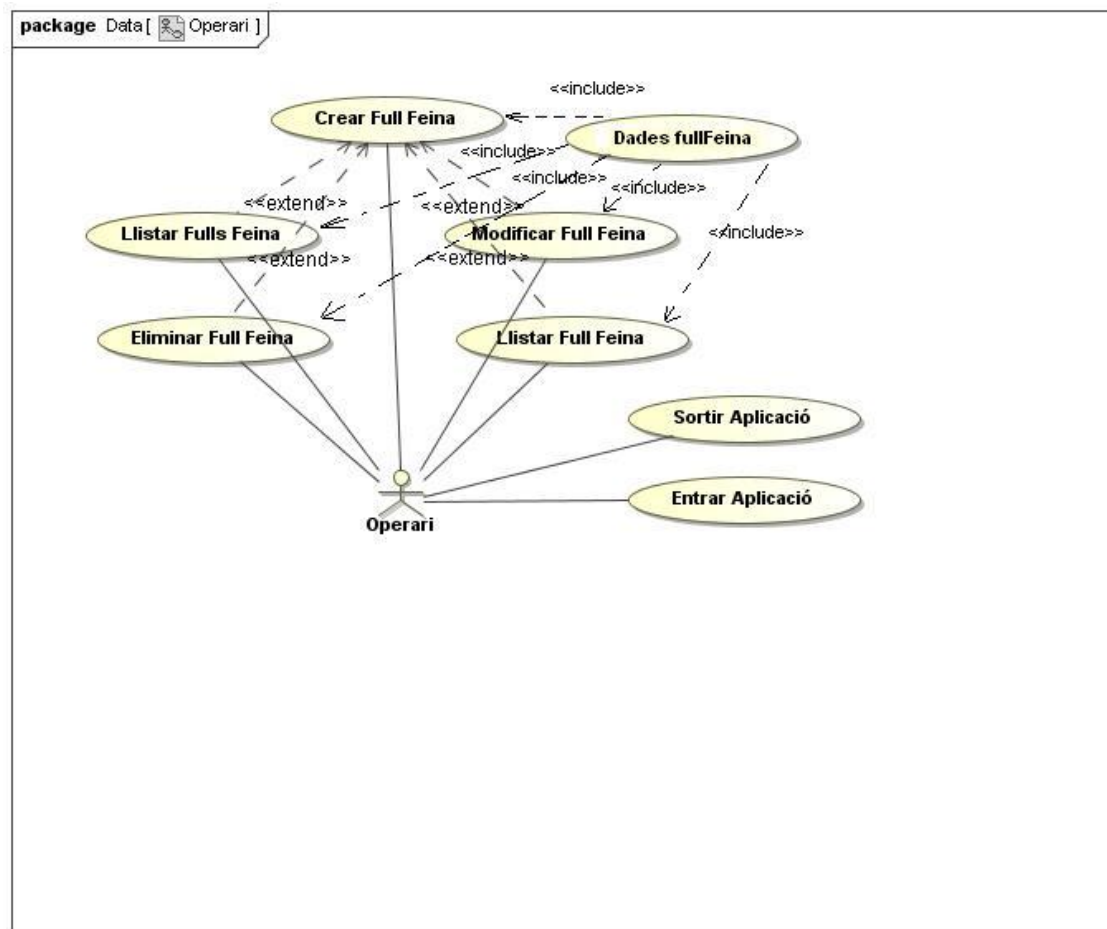


Fig 2.3 Casos d'ús de l'usuari Operari

Observem un parell d'exemples de la descripció d'aquest casos d'ús, que podrem trobar amb més detall en els annexes d'aquesta memòria:

- **Cas d'ús Crear Client**

Descripció: L'usuari gestor, principalment, crea un nou client i l'emmagatzema a la base de dades.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El client encara no està enregistrat a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya clients.
2. El sistema mostra la pàgina amb les dades a omplir del client
3. El gestor omple els camps necessaris per a crear el client.
4. El gestor prem el botó de crear, acceptant l'operació.
5. Si les dades són correctes:

- 5.1. Si el client no existeix
 - 5.1.1. Es crea el registre a la taula CLIENT.
- 5.2. Si el client existeix
 - 5.2.1. Es mostra per pantalla un missatge d'error per pantalla.
6. Si les dades introduïdes no són correctes, es mostra un missatge d'error per pantalla.
7. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El client s'ha emmagatzemat a la base de dades.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

- **Cas d'ús** *CrearFacturaPdf*

Descripció: L'usuari gestor, principalment, crear un arxiu pdf d'una factura.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir la factura en el sistema.

Flux principal:

1. El gestor selecciona la pestanya factures.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El gestor busca la factura.
4. El gestor prem el botó de crear pdf.
5. Si l'acció falla:
 - 5.1. Es mostra el missatge pertinent per pantalla.
6. Si l'acció es realitza correctament:
 - 6.1. Es crea el pdf en la ruta pertinent del servidor
 - 6.2. Es mostrarà el PDF generat per pantalla.
7. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat de totes les factures del sistema.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

3.2.3. Requeriments no funcionals del projecte

A part dels requeriments purament funcionals que desglossàvem en el anterior apartat existeixen altres requeriments, d'una naturalesa diferent, que també s'han de tenir en compte a l'hora de realitzar el disseny del sistema.

Els requeriments no-funcionals són requeriments que no estan relacionats amb la funcionalitat del sistema. Tracten aspectes com la seguretat o la disponibilitat, i poden

arribar a considerar-se restriccions o limitacions sobre el sistema, doncs han de respectar-se a l'hora d'establir els requeriments funcionals.

Podem classificar-los segons la seva naturalesa en requeriments de producte, de procés o externs, tot i així en ocasions pot no quedar suficientment clara la seva pertinença a un o altre grup. La manera de discriminar-los seria fixant-nos en les següents consideracions:

- Els requeriments de producte especifiquen característiques que el sistema ha de posseir, com la disponibilitat o la quantitat de peticions per segon que pot processar el sistema.
- Els requeriments de procés fan referència a les normes que s'han de seguir en el desenvolupament d'un sistema. Per exemple es poden definir quina documentació s'ha de generar durant l'implementació o la metodologia a seguir durant l'elaboració del projecte.
- Els requeriments externs tenen a veure amb el entorn en el que es desenvolupa el projecte, com per exemple l'entorn d'explotació en el que s'ha d'integrar el sistema.

El bloc de requeriments que més ens afecta en aquest cas és el primer. A continuació passarem a enumerar-los en funció de quins aspectes defineixen.

3.2.3.1. Rendiment

Dins de l'apartat de rendiment hem d'especificar quins paràmetres de rendiment s'hauran d'assolir per a que el sistema puguin dur a terme la seva labor d'efectivitat. Per exemple, el sistema ha de poder crear un client, però si tarda cinc minuts en fer-ho podria ser frustrant per l'usuari.

3.2.3.1.1 Volum de peticions

És important definir la càrrega de treball que serà capaç de processar el sistema. El número de peticions que processarà el sistema depèn bàsicament de dos factors:

- El nombre d'usuaris que accediran al sistema.
- El perfil d'aquest usuari i per tant l'utilització que aquest faci del sistema.

Observem que no només s'ha de tenir en compte el nombre d'usuaris sinó l'ús que aquest li donaran al sistema. No es el mateix un usuari que es limita a consultar les dades d'un client que un altre usuari que freqüentment altera el contingut de les taules.

Actualment i classificats segons el rol que ocuparan, podem dir que el nombre d'usuaris serà:

- Gestors del sistema: 2
- Operaris: 4
- Administradors: 1

Perfil d'usuari: Tot i que una intranet està basada en el funcionament d'Internet, no podem dir que el tipus d'ús que fan els usuaris sigui el mateix. Un usuari d'Internet es comporta segons el següent patró a l'hora de visionar pàgines web:

- Rep la pàgina web sol·licitada.
- Passa una estona llegint-la o buscant nous links que l'interessin.
- Sol·licita una pàgina web. La seqüència torna a començar.

No tots els usuaris de la nostre intranet es comportaran de la mateixa manera.

En un futur, el usuari no registrat serà l'usuari amb el perfil més semblant. Però un altre tipus d'usuari, com per exemple l'usuari Operari es comportarà segons la següent seqüència:

- Sol·licita la pàgina que desitja.
- Omple el full de feina.
- Envia informació al sistema, és a dir, els formulari del full de feina.
- Passada una estona tornem a l'inici de la seqüència.

El tràfic generat per l'usuari de l'intranet segurament estarà format per ràfegues de peticions amb un temps d'inactivitat entre elles. En canvi, el tràfic generat per un usuari web, aquestes ràfegues no seran tant llargues, sinó que el tràfic estarà constituït per peticions més espaciades.

Degut a que de moment, el client només vol disposar d'un sistema Intranet local, aquest estudi exposat anteriorment, es veurà realitzat en futures actualitzacions del sistema proposat.

3.2.3.2. Usabilitat

En aquest projecte l'usuari executarà l'aplicació a través d'una interfàç web. Existeixen nombrosos estudis sobre usabilitat referents a aquest àmbit i que podem extrapolar al nostre domini. Tot i que els perfils d'ús no són exactament iguals si tindran molta similitud.

Els problemes d'usabilitat en una Intranet corporativa repercuteixen en una disminució de la productivitat del empleats i per tant en una pèrdua de diner.

3.2.3.2.1. Microcontinguts

És important que els títols de les pàgines, notificacions o qualsevol titular compleixin les següents premisses:

- Que siguin una breu descripció del macro contingut associat.
- Llenguatge pla.
- Procurar que la primera paraula sigui la més important.
- Que els títols de totes les pàgines no comencin amb la mateixa paraula.

3.2.3.2.2. Formularis

A l'hora de presentar formularis a l'usuari per a que aquest els ompli, s'ha de tenir en compte les següents característiques:

- Complexitat de la informació sol·licitada: la informació sol·licitada a l'usuari ha de permetre'l omplir-la sense esforç. Procurar que la informació no sigui més complexa i en qualsevol cas oferir-li el material o documentació necessari.
- Nombre de passos: Que el nombre de passos necessaris per a completar l'operació sigui reduït.
- Que els camps sempre siguin els mateixos i existeixi una única seqüència o fluxe d'operacions.
- Que la seqüència de passos sigui lineal.
- Ometre tots aquells camps que no siguin necessaris i facin que el formulari sigui massa extens.

3.2.3.2.3. Nombre de Passos

El nombre de passos en el que un usuari pot realitzar qualsevol operació hauria d'estar acotat. Pel nostre cas considerarem adequat que un usuari pugui accedir a qualsevol pàgina amb només tres salts(un cop identificat en el sistema).

3.2.3.2.4. Temps d'Espera

En els primers estudis sobre els temps d'espera i usabilitat de pàgines web, es considerava que el màxim temps que l'usuari podia esperar mentre es carregava una pàgina sense desviar la seva atenció era de 10 segons. Actualment l'usuari ja s'ha acostumat a temps d'espera inferiors.

La naturalesa de la nostre aplicació fa que hagi de comportar-se com una aplicació corrent de forma local en un equip, per tant, els temps d'espera també han de ser inferiors. Considerem com a vàlids temps d'espera en que el màxim no superi els 4 segons i la mitjana es situï al voltant de 1 segon.

3.2.3.2.5. Look&Feel

S'han de complir les següents premisses:

- Totes les pantalles de l'aplicació han de mostrar l'emblema identificatiu de l'empresa.
- Es convenient la presència del colors corporatius, sempre que no xoquin amb les requeriments d'usabilitat de l'aplicació.

3.2.3.3. Seguretat

L'aplicació ha de garantir l'integritat de les dades que utilitza. També ha d'assegurar que aquestes dades seran accessibles per aquells usuaris que hi hagin estat autoritats.

Els principals aspectes de seguretat que s'han de garantir per considerar una aplicació com a segura són la confidencialitat de les dades, la integritat i el no repudi.

En el nostre cas hem d'assolir un compromís entre seguretat i funcionalitat: una aplicació que constantment sol·licités a l'usuari contrasenyes, claus de seguretat o certificats a través de protocols xifrats podria ser molt segura, per no ho seria funcionalment.

El sistema ha de garantir que:

- Només els usuaris autoritzats podran accedir al sistema.
- Només aquest usuaris podran accedir a l'informació gestionada pel sistema i només a la que els hi estigui permès accedir.

3.2.3.3.1. Política de Seguretat

Els usuaris hauran de complir amb la política de Seguretat dissenyada per la plataforma.

Les polítiques de seguretat indiquen i obliguen als usuaris com han d'interactuar amb el sistema de forma correcta, de manera que es redueixin les possibilitats que es produeixin problemes de seguretat.

Algunes clàusules habituals en aquest documents són per exemple:

- Obligació d'utilitzar contrasenyes complexes i segures.
- Prohibició de comunicar aquestes contrasenyes a companys.
- Obligació de finalitzar la sessió en el sistema quan s'abandoni el lloc de treball.

Com observem regulen aspectes trivials que podrien comprometre la seguretat d'un sistema.

3.2.3.4. Fiabilitat

L'aplicació a implementar serà l'eina de gestió d'una empresa. Un grup de persones treballaran recolzant-se en aquesta eina i els errors o el no funcionament d'aquesta repercutiran directament en la productivitat del grup, amb el seu corresponent efecte econòmic.

Per tant, és important assegurar:

- Alta disponibilitat: El sistema ha d'estar operatiu i funcionant en tot moment.
- Integritat de la informació: Hem de garantir que no perdrem informació.

De cara a assegurar el requeriment d'alta disponibilitat depenem bàsicament de tres factors: La estabilitat de la plataforma desenvolupada, el funcionament dels equips on s'allotja i la fiabilitat de la xarxa.

L'integritat de la informació dependrà de la disponibilitat del sistema i que l'accés a les dades es realitzi de forma ordenada evitant problemes de concurrència.

3.2.4. Disseny de l'Arquitectura de l'aplicació JEE

3.2.4.1. Introducció

Una vegada especificats els requeriments tant funcionals com no funcionals hem de definir quines peces formaran part del nostre sistema i quin serà el mètode en que aquestes peces s'uniran. D'aquestes decisions depèn en gran manera que el sistema desenvolupat compleixi amb els requeriments especificats inicialment.

La plataforma que volem desenvolupar està orientada a oferir un servei als usuaris de l'intranet empresarial. La forma en que aquesta proporcionarà el servei als usuaris és la típica de les aplicacions web: seguint un model client-servidor i a través d'una interfície d'usuari html accessible des de qualsevol navegador.

Els avantatges d'aquest model entre d'altres són:

- Accessibilitat des de qualsevol màquina connectada a la Intranet empresarial.
- Absència de necessitat d'instal·lar aplicacions en les màquines dels usuaris per accedir al sistema.
- Aplicació en xarxa i en temps real: s'assegura la disponibilitat i integritat de les dades.

L'implementació d'aquest model fa que hàgim de tenir en compte diversos aspectes per a garantir el correcte funcionament del sistema:

- Assegurar l'accés concurrent als recursos del sistema.
- Implementar un control d'accés fiable i funcional.
- Protegir el sistema en front a possibles errors de manera que aquests no afectin a la disponibilitat del mateix.

3.2.4.2. Disseny de l'arquitectura de la plataforma

3.2.4.2.1. Patró de disseny

Com apuntàvem en l'apartat anterior, l'esquema de funcionament de la plataforma serà el model client-servidor, típic de les aplicacions web. El desenvolupament

d'aplicacions web és un camp molt madur i en la majoria de projectes és habitual utilitzar patrons de disseny fiables i de funcionament contrastat.

El model escollit com a patró de disseny és el Model 2 o Model Vista-Controlador:

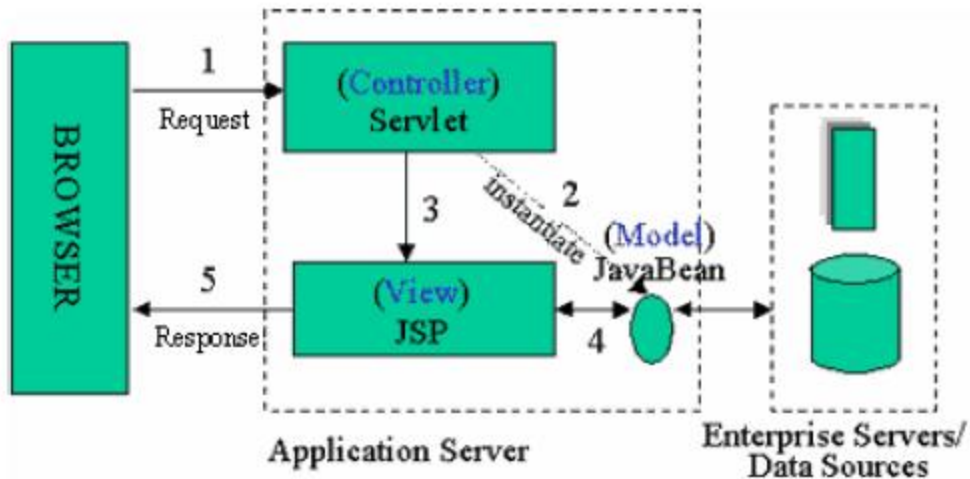


Fig 3.1 Funcionament del Model 2

3.2.4.2.2. Arquitectura de la plataforma

Seguint aquest patró de disseny podem fer una separació en capes dels diferents blocs lògics que conformarien l'aplicació.

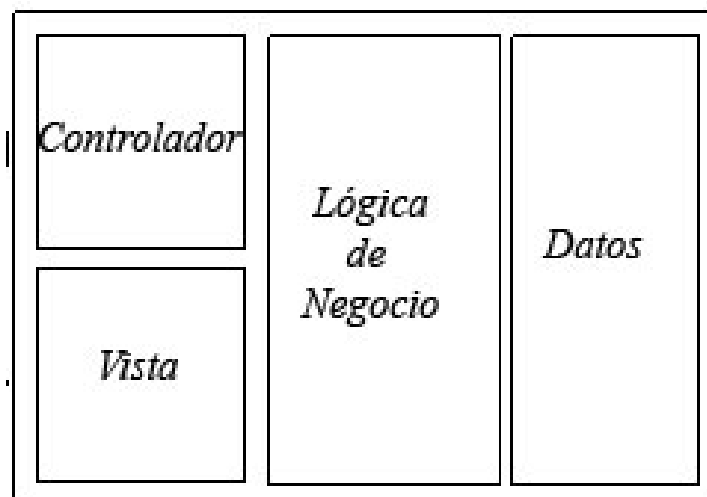


Fig 3.2 Capes que formen el sistema

- Capa de presentació
 - Vista: genera el codi html que se li mostra a l'usuari.

- Controlador: rep les peticions de l'usuari i decideix quin element de Vista contestarà a la petició.
- Lògica de negoci: implementació dels casos d'ús especificats en els requeriments.
- Capa de dades: implementa les comunicacions amb la base de dades i el repositori de fitxers.

3.2.4.2.2.1. Diagrama de classes

En el diagrama de classe següent podem observar l'estructura que formen part de les diferents capes. Es tracta d'una versió simplificada del diagrama.

- Controlador: crearClientAction
- Lògica de Negoci: MOClient
- Accés a dades: DaoFactory

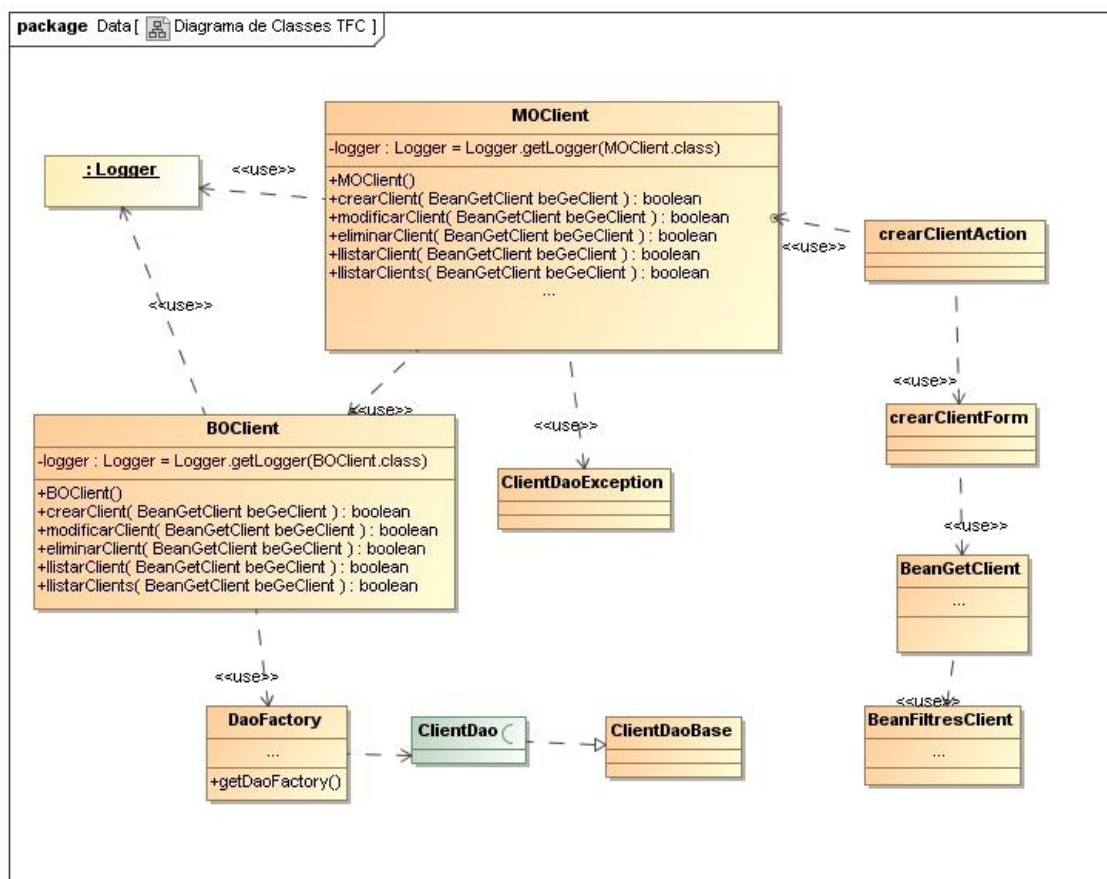


Fig 3.2 Diagrama de Classes

3.2.4.2.2.2. Diagrama de seqüència

A través del següent diagrama de seqüència podem fer-nos una idea de quin serà el comportament del sistema. El diagrama mostra de forma resumida el fluxe normal del cas d'ús "Crear Client":

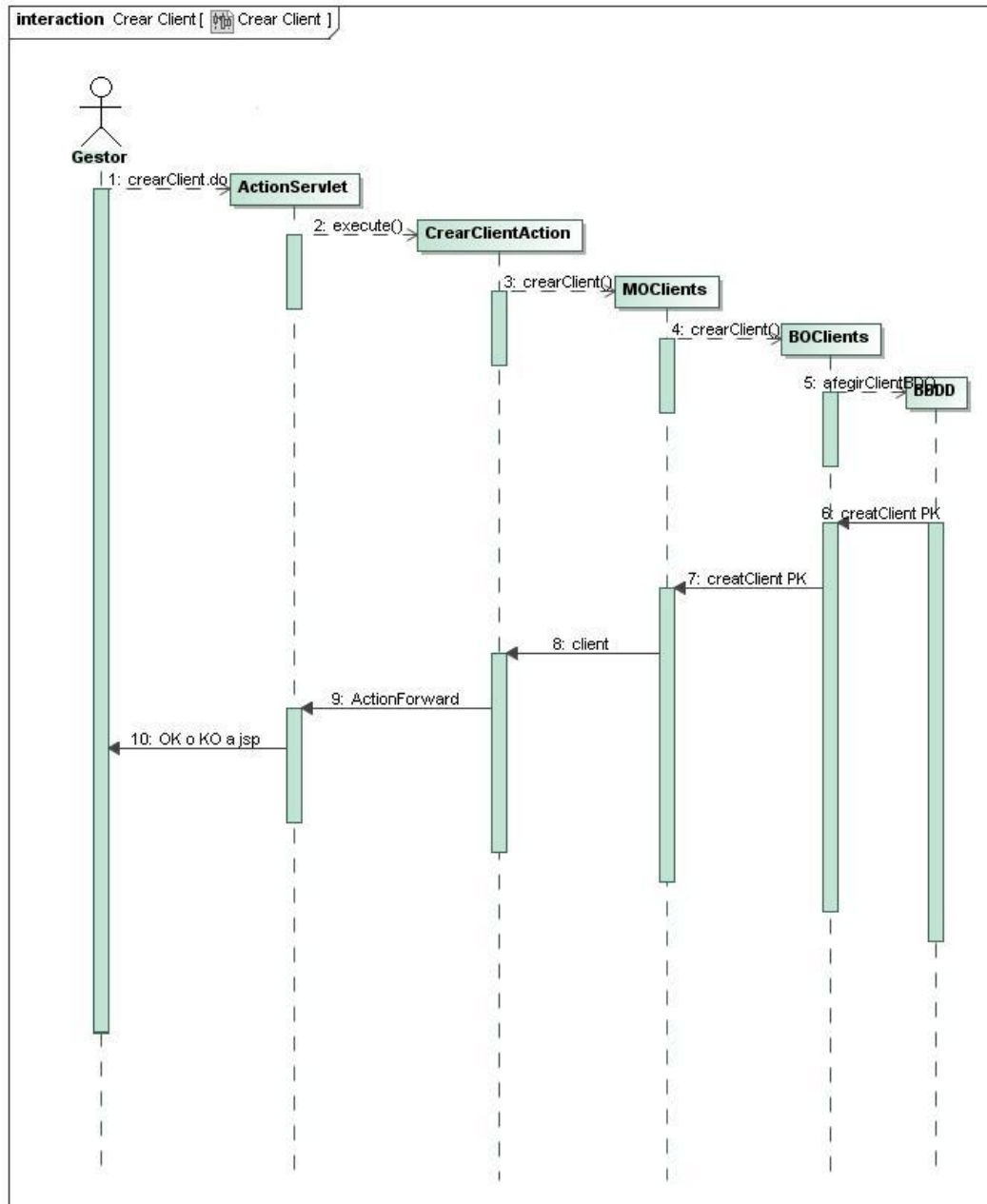


Fig 3.4 Diagrama de Seqüència de crearClient()

3.2.4.3. Accés a dades

La plataforma haurà de gestionar una gran quantitat d'informació. De partida haurem d'emmagatzemar les dades relatives a les factures de clients i els seus respectius fulls de feines. Del mateix mode haurem d'emmagatzemar dades relatives al funcionament intern de la plataforma: usuaris, permisos, clients, proveïdors, etc.

Per emmagatzemar les dades d'una forma adequada haurem d'utilitzar una base de dades i un repositori de fitxers.

3.2.4.3.1. Base de Dades

Per treballar amb una gran quantitat de dades en temps real i de forma distribuïda és necessari l'ús d'un Sistema de Gestió de Bases de Dades (DBMS²³), o el que és el mateix, un servidor de bases de dades que s'encarregui de gestionar les peticions dels usuaris i modificar les dades assegurant l'integritat i coherència dels mateixos.

La manera en que el "Model" es comunica amb la base de dades és crítica pel rendiment global del sistema. Per reduir el temps que tarda en realitzar-se una consulta hi ha dues vies: la primera és l'elaboració de "queries²⁴" o consultes adequades, que veurem en el "model de dades", i l'altre via és la correcta gestió de les connexions al DBMS.

Hem de procurar utilitzar les connexions recurrentment per a realitzar peticions, ja que, el cost d'obrir-les i tancar-les constantment és elevat i compromet el rendiment de l'aplicació. Per això es comú l'ús d'un "Pool de connexions²⁵".

Un "Pool de connexions" funciona com un pool o depòsit: mantenim varies connexions actives constantment. En l'instant en que la nostre aplicació necessita realitzar alguna consulta al sistema, agafem aquesta connexió del dipòsit i no hem d'esperar el temps d'establiment de la connexió. Quan finalitzem la consulta, la connexió s'allibera i torna al pool fins que no hagi de tornar a utilitzar-se. D'aquesta manera ens estalviem els temps de connexió i el cost computacional que suposaria obrir i tancar una connexió per a cada consulta, tant per al que realitza la consulta com per al DBMS.

3.2.4.3.2. Repositori de Fitxers

Utilitzarem un repositori de fitxers per emmagatzemar documents com per exemple, les factures dels clients, llistat de clients, full de feines.

Ha de permetre crear, eliminar o consultar documents, sense problemes de concurrència al treballar amb ells múltiples usuaris.

Per a treballar de forma distribuïda, de cara a garantir l'escalabilitat del sistema, el repositori ha de ser accessible a través de la xarxa. D'aquesta manera podrà replicar-se el nucli de la plataforma de cara a millorar el rendiment del sistema.

3.2.4.4. Model de dades

3.2.4.4.1. Model relacional

El model més utilitzat actualment per a la gestió de dades és el model relacional.

En el model relacional totes les dades es representant mitjançant relacions N-aries matemàtiques, poden tractar la informació mitjançant expressions que retornaran valors booleans True o False.

El model de dades relacional permet crear una representació lògica i consistent de la informació.

3.2.4.4.2. Diagrama ER

Els diagrames ER són un llenguatge gràfic formal que ens permet, mitjançant dibuixos, descriure la informació amb la que treballarà el nostre sistema i la manera en que aquest ho farà.

Mitjançant aquests diagrames, podem representar el model de dades del sistema. Aquest és el primer pas per a poder dissenyar la nostre base de dades.

Primer hem de diferenciar els diferents elements que representarem en el nostre diagrama:

- **Entitats:** Una entitat és qualsevol objecte o element sobre el qual tenim informació. Es representa mitjançant un rectangle o caixa etiqueta amb el seu nom. Per exemple: Usuari, Client o Factura. Cada unitat o exemplar d'aquesta Entitat és una Instància.
- **Relacions:** Una relació és la representació de qualsevol interdependència entre dos entitats. Es representa mitjançant un rombe etiquetat en el seu interior amb un verb (que identifica la relació). Aquest rombe s'uneix amb línies a les entitats que relaciona.
- **Atributs:** Son propietats relatives a una Entitat. Es representa mitjançant un cercle o el·lipse etiquetat en l'interior d'una entitat. Solen omitir-se en el diagrames per motius de llegibilitat i es descriuen a part, per exemple en l'especificació de les taules.

3.2.4.4.2.1. Diagrama

A continuació presentem el diagrama entitat-relació de la Base de dades, generat amb la eina Microsoft Visio 2007.

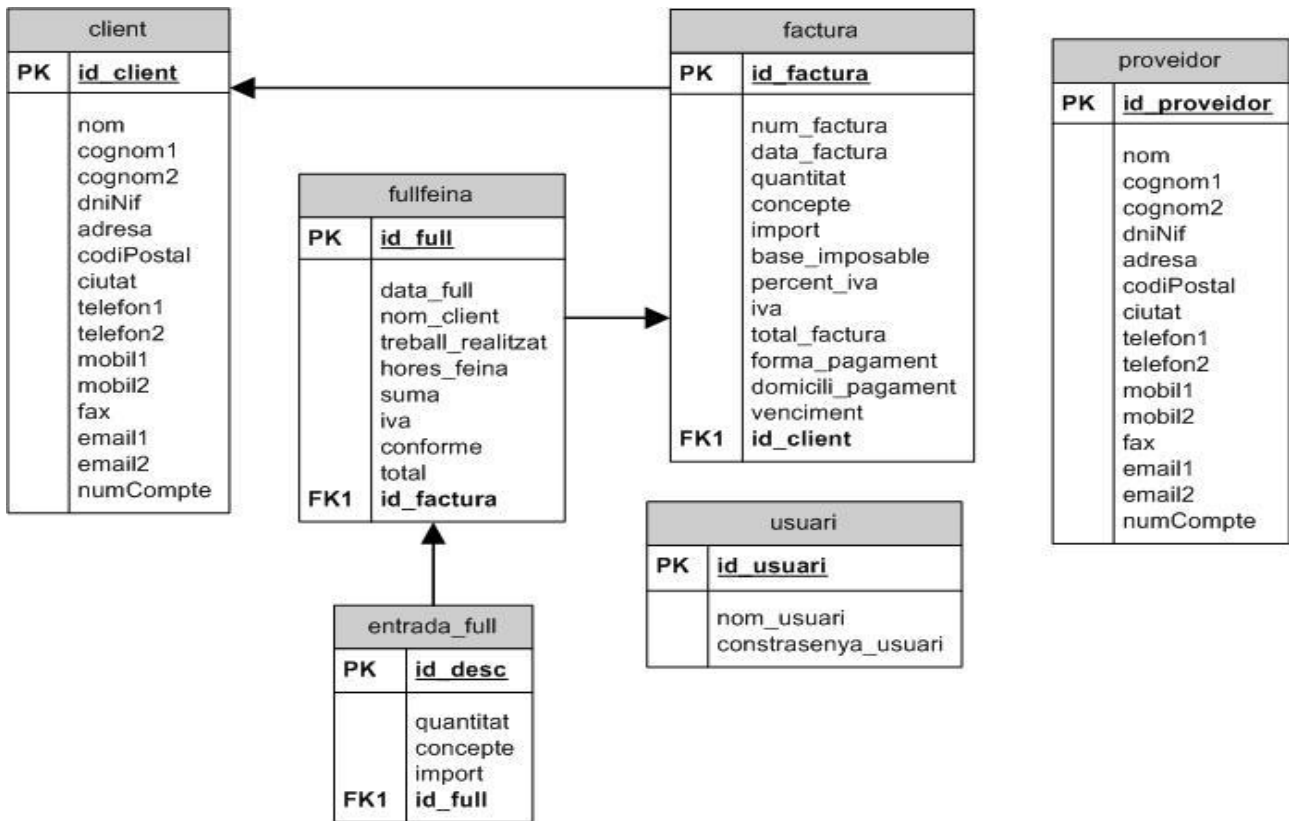


Fig 3.5 Diagrama d'entitat-relació de la Base de Dades

3.2.4.5. Altres aspectes de disseny

3.2.4.5.1. Logs

És molt important que puguem obtenir en temps real dades relacionades amb el funcionament del sistema. Tant en les fases inicials d'implantació com en estat més madurs, poder obtenir i analitzar informació detallada sobre el comportament de l'aplicació ens ajudarà a poder detectar l'origen de possibles errors que d'una altre manera suposaria un terrible mal de cap.

Un sistema que generi una gran quantitat de traces veurà afectat el seu rendiment, a conseqüència del consum de recursos que això implica. Per una altre banda, emmagatzemar dades de manera insuficient no ens ajudarà en cas que necessitem revisar la informació.

Aquest problema pot solucionar-se de manera que puguem escollir en cada moment el nivell de detall amb el que volem emmagatzemar la informació.

Els nivells de detall seran:

- **Debug:** nivell de detall molt elevat. Recomanable utilitzar tant sols en fases de proves i d'implantació on necessitem revisar a consciència el funcionament de l'aplicació.

- **Info:** nivell de detall mig, en el que podem seguir la seqüència de execució d'una acció però no dades com valors de les variables, etc.
- **Warn:** nivell de detall baix. En aquest cas el sistema guarda detall de els errors provocats per un ús indegut però controlat del sistema.
Exemple: la connexió amb la base de dades es tanca i impedeix realitzar un "commit".

També és important la forma en que emmagatzemem les dades. Si ho fem en un fitxer en el disc de la nostra màquina, convindrà limitar la longitud total del fitxer, ja que treballar amb fitxers d'un tamany molt extens disminueix l'eficiència del sistema. En el nostre cas, el fitxer en el que s'emmagatzemen les dades anirà canviant diàriament, de manera que per cada fitxer de log tindrem una distinció segons la data en que s'ha generat.

No s'emmagatzemaran en fitxers de log informació sensible com poden ser contrasenyes. Per una altra banda, procurarem emmagatzemar sempre els fitxers en un directori amb els permisos adequats per a que no puguin ser accessibles per usuaris no desitjats.

3.2.4.5.2. Configuració

L'utilització de mecanisme de configuració aporta certa flexibilitat a una aplicació, ja que podem variar certs paràmetres de comportament de la mateixa sense necessitat de modificar el codi de l'aplicació.

Realitzar una modificació en el codi font, per senzilla que sigui, pot suposar una gran inversió en temps, i per tant de diner. La trivial modificació del valor d'una constant suposaria:

- Modificar el codi font.
- Compilar el codi.
- Empaquetar el codi.
- Reinstal·lar l'aplicació

Pot no semblar tant complicat, però quan la persona que realitza la modificació no és la mateixa que ha creat l'aplicació, o quan ha passat un temps des de que es va dur a terme l'implementació fins que es modifica, es fa més probable que qualsevol petit detall (absència d'una llibreria necessària per a la compilació, etc) ens faci perdre una gran quantitat de hores.

En canvi la modificació d'aquesta constant en un fitxer de configuració ens portarà escassos minuts.

Per aquest motiu hem de procurar que es puguin configurar paràmetres com:

- Rutes de fitxers de configuració, repositori de fitxers.
- URL de la base de dades, així com el nom d'usuari i contrasenya.
- Configuració del pool de connexions.
- Constants del sistema.

Un altre aspecte que ha de ser fins a cert punt configurable és el disseny de l'interfaç de usuari. Haurem d'utilitzar les facilitats que ens ofereixen tecnologies com CSS per a aconseguir certa configurabilitat en el codi html.

3.2.4.5.3. Robustesa

Un altre factor que haurem de tenir en compte a l'hora de dissenyar la plataforma és com respondrà el sistema en front a les possibles dificultats que apareixen. Les dificultats o errades a les que ens enfrontem poden provenir de la mateixa plataforma o dels elements externs (BD i Repositori). La manera en que el sistema reaccionarà a un d'aquests errors serà la següent:

- *Reacció davant de problemes interns.*
Al detectar-se un mal funcionament o un error intern de la plataforma es cancel·larà l'operació que s'està realitzant en aquell precís instant i es guardaran els detalls del mal funcionament en un fitxer de log.
- *Reacció en front problemes amb la Base de Dades.*
Si al realitzar una operació es produeix algun tipus d'error a l'intentar accedir a la Base de dades (es perd la connectivitat, etc) es procedirà de la següent manera:
 - S'anul·larà l'operació que s'estava realitzant.
 - Es notificarà l'error retornant el codi de l'error que pertoqui.
 - Es guardarà constància de l'error en el log pertinent.

3.2.5. Eleccions tecnològiques.

3.2.5.1. Introducció

Una vegada definida l'arquitectura del sistema haurem d'escollir les tecnologies i eines necessàries per a implementar-la.

L'elecció d'una o altre opció sempre anirà condicionada pels requeriments a complir, a més a més d'un altre tipus de factors com poden ser la preferència o coneixement de les diferents tecnologies per part del desenvolupador. S'ha escollit, preferiblement, eines de software lliure degut al nul cost d'adquisició d'aquests.

A continuació descriurem quines eines utilitzarem per implementar cadascun dels elements de l'arquitectura del sistema.

3.2.5.2. Framework de desenvolupament

En el mercat existeixen diferents implementacions del Model Vista-Controlador. Existeix molt bones opcions que ens ajudaran en el desenvolupament de la plataforma. L'opció que hem escollit és el framework Apache Struts.

3.2.5.2.1. Struts

Per implementar el patró de disseny escollit per a l'arquitectura de l'aplicació hem decidit utilitzar el framework de codi obert Apache Struts.

Struts és una implementació en Java del model Vista-Controlador. Es tracta d'una tecnologia madura i eficient àmpliament utilitzada.

En els annexes s'inclou una descripció de les principals característiques de aquesta implementació, el seu funcionament i una breu comparació amb altres opcions que haguessin pogut ser escollides.

3.2.5.3. DBMS

Una de les decisions que hem hagut de prendre és l'elecció del Gestor de Bases de dades o Data Base Manager Server (DBMS). EL DBMS és una aplicació que gestiona l'accés al suport d'on es troben emmagatzemats les dades amb les que treballaran una o varies aplicacions. Proporcionen una interfaç de comunicació entre aplicacions i el sistema operatiu, amb l'objectiu que l'accés a dades es realitzi de forma més eficient, senzilla i segura.

3.2.5.3.1. Requeriments

Les principals característiques que busquen en un DBMS són les següents:

- *Persistència de les dades*: Que les dades emmagatzemades romanguin en el suport escollit fins que siguin explícitament mogudes o eliminades.
- *Capacitat de consulta*: El sistema ha de facilitar a l'usuari la possibilitat de realitzar consultes sobre les dades emmagatzemades en funció de diversos paràmetres.
- *Concurrència*: Varis usuaris han de poder consultar i modificar les mateixes dades a la vegada d'acord amb certes normes que ens assegurin la consistència de la informació.
- *Replicable*: El DBMS ha de permetre que es realitzin còpies de seguretat o Backups de la informació que emmagatzema. En cas contrari un error en els suports o en el funcionament del DBMS podria resultar en la eliminació d'informació important.
- *Establiment de normes*: El sistema ha de permetre l'establiment de normes sobre la coherència de les dades introduïdes en les taules. Per exemple no possibilitar l'eliminació de dades amb dependències.
- *Seguretat*: És important que el sistema només permeti accés a les dades als usuaris autoritzats per aquesta finalitat.
- *Capacitat d'operació sobre les dades*: Que el sistema ofereixi possibilitat de realitzar operacions tals com contar registres, ordenar-los, agrupar-los...

3.2.5.3.2. PostGreSQL

En els annexos es presenten dos Gestors de Bases de Dades que podríem utilitzar en el nostre projecte i es realitza una breu comparació dels seves característiques.

El DBMS escollit és PostGreSQL, que s'ajusta a les necessitats llistades en l'apartat anterior. La resta de factors que han fet que ens decanem per aquesta opció es detalla junt a la comparativa, en la secció d'annexos.

3.2.5.4. Servidor

3.2.5.4.1. Contenedor de Servlets

En el mercat hi ha nombrosos servidors o contenidors de servlets que podrien servir-nos com a suport per a la plataforma. Hem d'escollir quin d'ells s'ajusta millor a les nostres necessitats.

Si limitem l'elecció a les opcions gratuïtes el ventall de possibilitats es redueix. En aquest cas les més àmpliament utilitzades són les següents:

- Apache Tomcat.
- JBoss.

Apache Tomcat és el Contenedor de Servlets de referència. Ens ofereix una sèrie de funcionalitats addicionals com la possibilitat de configurar pools de connexions amb la base de dades, mecanismes de balanceig de càrrega i facilitats en quant a configuració, logs i utilitats de gestió de les aplicacions web desplegades.

Aquestes característiques, sumades a la contrastada estabilitat del servidor, fan que ens decanem per Tomcat com a contenedor de servlets per al nostre projecte.

3.2.5.4.1.1. Apache Tomcat

Apache Tomcat és un contenedor de servlets de codi obert desenvolupat per la *Apache Software Foundation*. Tomcat es distribueix sota la llicència Apache 2.0.

Existeixen diferents versions de Tomcat. Per al nostre projecte s'ha escollit l'última versió estable: Apache Tomcat 6.0.16.

Aquesta versió implementa l'especificació 2.5 de Servlets i la 2.1 de JSP.

Tomcat, a més a més de la gestió de *threads*²⁶, ens ofereix altres eines que ens seran útils en el desenvolupament de la nostra plataforma:

- *Data Base Connection Pool*
Tomcat implementa un pool de connexions que permet establir connexions de forma àgil per a comunicar-se amb la Base de dades.
- *Logging*

Tomcat 6 utilitza la llibreria Commons-logging que permet treballar amb APIs com log4j. Log4j ens permet gestionar el emmagatzemament de logs podent establir nivell d'informació o rotació diària de fitxers de log de forma independent al sistema.

- *Deployer*
Tomcat ens ofereix la possibilitat de “Desplegar”, parar i recarregar les nostres aplicacions sense haver de reiniciar el servidor. D'aquesta manera i en el cas d'allotjar varies aplicacions en un únic servidor no hauríem de parar totes les aplicacions per actualitzar una d'elles.
- *Configuration*
Podem utilitzar el sistema de fitxers de configuració dels contextos per configurar les nostres aplicacions.

3.2.5.5. Altres eleccions

3.2.5.5.1. Logs

Una de les opcions més utilitzades per a gestionar els logs en aplicacions Java és la Api Log4j.

Log4j forma part del projecte Apache i permet generar logs d'una manera flexible i configurable, podent modificar el nivell de detall de les traces i el format de sortida de les dades sense necessitat d'haver de modificar el codi de l'aplicació.

Aquesta API ens permet emmagatzemar logs tant en fitxer locals com de forma distribuïda en una base de dades. En el cas d'utilitzar l'emmagatzemament d'informació en fitxer locals podem regular paràmetres tals com el tamany màxim, o el període temporal que utilitzarem un fitxer de log abans de passar a emmagatzemar les dades en el següent.

Per una altra banda, Log4j funciona perfectament en combinació amb la API commons-logging de Tomcat, amb el que es pot integrar sense problemes en l'aplicació web sense generar conflictes amb el sistema de logs del servidor.

3.2.5.5.2. Apache Tiles

Apache Tiles és un framework construït per facilitar el desenvolupament d'interfaç d'usuari d'aplicacions web. Tiles intenta evitar la duplicació de contingut de llenguatge de marcat dins d'una aplicació amb respecte al *look&Feel* d'un portal. A més a més redueix el tamany de codi redundant en una aplicació i separa el contingut de la visualització del mateix de manera eficient.

Per tant, si la nostra aplicació ha de tenir similitud en totes les pàgines, Tiles ens permet crear una plantilla on és configuraran l'aspecte que mostraran les nostres pàgines, per exemple: disposar del mateixa capçalera i menú esquerre en totes les pàgines html.

3.2.5.5.3. DisplayTag

És una llibreria de tags jsp de codi obert que serveix de gran utilitat quan es vol mostrar una col·lecció de dades en una taula, permetem-nos, a més a més, entre d'altres coses, agregar estils, decoradors de les dades mostrades, exportar a diferents formats, utilitzar la paginació i ordenació.

3.2.5.5.4. Fop

En anglès s'anomena "*Formatting Objects Processor*", és un processador de fitxer XSL-FO multiplataforma escrit en java que ens permet convertir arxius XSL-FO²⁷ a format PDF²⁸ o a arxius directament imprimibles.

Originalment, va ser desenvolupat per "James Tauber", qui més tard va donar el projecte a la Fundació Apache en 1999.

3.2.5.5.5. Configuració

Per implementar les funcionalitats de configurabilitat de la plataforma ens ajudarem de les pròpies facilitats que ens ofereix el servidor Tomcat.

En tota aplicació web de Tomcat s'utilitza l'arxiu "*web.xml*" per a definir els servlets que s'executaran en el context de l'aplicació. En aquest mateix fitxer es poden indicar paràmetres d'inicialització als que podran accedir els servlets al crear-se.

En aquest arxiu es definiran alguns dels paràmetres configurables i s'indicaran les rutes de fitxers de configuració a utilitzar per altres APIs integrades en la plataforma:

- *Context.xml*: dades de configuració del pool de connexions a la base de dades.
- *Log4j.properties*: dades de configuració per al sistema de logs.
- *Struts_config.xml*: fitxer que consulta l'ApplicationServlet de Struts per a resoldre i encaminar les peticions rebudes.

3.2.5.5.5.1. Configurabilitat de l'Interfaç d'usuari

Mitjançant l'utilització de Cascade Style Sheets (CSS) s'aconsegueix una certa configurabilitat de l'aparença de l'interfaç d'usuari.

En un fitxer es defineixen certes propietats dels elements que apareixen en el codi html de l'interfaç d'usuari, com per exemple la font utilitzada pels diferents títols, per la capçalera de les taules o el color de fons de les pestanyes.

Aquesta configurabilitat fa possible que es pugui alterar l'aparença global de l'interfaç d'usuari a través d'aquest fitxer de configuració, sense necessitat de modificar el codi de tots els jsp que generen el codi html.

3.2.6. Interfície d'Usuari

L'interfície d'usuari de la plataforma estarà formada per pàgines jsp, encarregades de generar el codi html que se li presentarà a l'usuari.

A continuació podem veure l'aparença de la IU²⁹.

3.2.6.1. Aparença

The screenshot shows a Mozilla Firefox browser window with the address bar set to `http://localhost/TFC/client.do`. The page title is 'Portal Administratiu Riba' and the user is identified as 'Xavi Avui'. The application has a navigation menu with tabs: 'Clients', 'Proveïdors', 'Fulls de Feina', 'Factures', and 'Administrador'. Below the navigation is a search section titled 'Filtres Recerca' with input fields for 'IDClient', 'Nom', 'Cognoms', and 'Adreça', and buttons for 'Buscar' and 'Netejar'. The main content area is titled 'Llistat Clients' and displays a table with 14 records. The table columns are 'IdClient', 'Nom', 'Cognom1', 'Adresa', and 'Drinif'. The records show a list of clients with IDs ranging from 4300001 to 4300010, all with the surname 'fernandez' and address 'C/Mossen Cinto Verdaguer n75'.

IdClient	Nom	Cognom1	Adresa	Drinif
4300001	xavi	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300002	jesus	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300003	marta	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300004	asasa	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300005	assdsd	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300006	adffdfdf	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300007	adfdfdfdff	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300008	mnjumjuju	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300009	hgthgttg	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q
4300010	popopo	fernandez	C/Mossen Cinto Verdaguer n75	46816148Q

3.3. Capítol 3, Implementació i proves

L'implementació ve condicionada pel patró de disseny que hem triat, és a dir, Struts, i per les quantes llibreries que s'han adjuntat al projecte, per poder dur a terme les nostres perspectives exposades a l'inici del projecte.

3.3.1. Funcionament

L'aplicació es divideix en dos grans capes, la capa web que correspon a la vista, al controlador, més el patró Business Delegate que ens servirà de vincle d'unió entre aquesta capa i la capa que conté la lògica de negoci i de la base de dades (capa de negoci).

L'arquitectura de l'aplicació i la comunicació entre les capes serà:

Pàgina JSP → ActionForms (Struts) → Actions (Struts) → Model Object → Business Object → Base de Dades (DAO).

3.3.1.1. Capa Web

Aquesta capa és la porta d'entrada a l'aplicació, ja sigui a través de les vistes, o del propi controlador. Cada vegada que el servidor rep una petició, el servlet de l'aplicació la captura i realitza el necessari segons el fitxer de configuració d'Struts.

Un cop capturada la petició, es comprova si existeix algun mapeig en el fitxer de configuració que mapegi la petició amb l'instància **Action** apropiada. Aquesta classe **Action** serveix per reconduir la petició del client d'una operació de negoci.

Si la petició prové d'un "form bean" es fan servir els objectes **ActionForm** d'Struts per passar les dades d'entrada el client entre l'usuari i la capa de negoci. Struts recull automàticament l'entrada de dades de la petició i els passa a l'**Action** adequat fent ús d'un "form bean".

La classe Action corresponent delegarà a el patró Business Delegate de l'usuari, que pren el control i realitza les operacions necessàries.

Per a fer l'interface d'usuari s'ha fet servir el framework d'Apache Tiles, com una implementació del patró Composite View i JSP junt amb tags d'HTML.

Tiles està configurat mitjançant un arxiu anomenat "tiles-defs.xml".

S'ha definit una plantilla que seguiran totes les pàgines JSP del portal, anomenada, tilesGeneral.jsp, aquí s'ha definit quina serà l'organització de la pàgina, que estarà formada per:

- **areaCapçalera:** fixe en totes les pàgines.
- **navegació:** depenent els rols d'usuari, es veuran unes pestanyes o unes altres.

- **areaCos:** depenent quina acció es triï, aquesta part de la pàgina variarà en conseqüència.

Totes les pàgines JSP extendran de tilesGeneral, d'aquesta manera ens assegurem mostrar el contingut que nosaltres vulguem sense haver de repetir codi.

3.3.1.2. Capa de Negoci

La lògica de negoci la componen les operacions que poden realitzar els tipus d'usuaris del portal que s'ha creat, és a dir, Administrador, Gestors i Treballadors i és l'eix fonamental sobre el que funciona l'aplicació.

Les operacions contra la Base de Dades de l'aplicació són encarregades al patró DAO.

3.3.2. Requeriments de Software

La descripció dels recursos necessaris pel desenvolupament del projecte i per a la seva implantació a l'empresa serà:

- **Recursos de Hardware**
 - **Servidor de l'aplicació**
 - **Hardware de tipus escalable recomanat:**
CPU: *CPU XEON Irw 2,8 Ghz 2MB*
RAM: *2Gb DDR2- 400 ECC*
HDD: *6xHDD SCSI 10K 146GB ST3146707L*
LAN: *Dual Intel Pro/1000 82541PI*
DVD: *DVD Rom LG GDR 8164B*
Teclat y ratolí òptic
3 anys de garantia "in situ" 4 h.
 - **Software:**
 - Windows 2003 Server Standard Edition R2.
 - **Comunicació:**
 - Configurada per permetre la connexió des dels altres equips del sistema a la direcció IP fixa del servidor.

Com podem observar aquesta seria una possible configuració de cara a un futur, quan es volgués expandir i donar accés als usuaris mitjançant la xarxa, és a dir, disposar d'una pàgina web, on l'usuari podrà interactuar amb la web depenent les seves necessitats.

En el nostre cas, de moment s'instal·larà l'aplicació a dos ordinadors, que seran els encarregats de fer de BD i contenidor de servlets; i en futures ampliacions, es realitzarà un estudi a mida de les possibles necessitats que disposi en aquell moment l'empresa.

La configuració de les màquines serà:

- **Recursos de Hardware**
 - **Servidor de l'aplicació i BD**
 - **Hardware de tipus escalable recomanat:**
CPU: *CPU Intel Core 2 Duo 6600 2,4GHZ*
RAM: *2Gb DDR2- 400 ECC*
HDD: *2xHDD SATA 7600Rpm, configurat en format mirall, raid 1*
LAN: *Dual Intel Pro/1000 82541PI*
DVD: *DVD Rom LG GDR 8164B*
Teclat y ratolí òptic
3 anys de garantia "in situ" 4 h.
 - **Software:**
 - Windows XP Professional SP2.
 - **Comunicació:**
 - Configurada per permetre la connexió des dels altres equips del sistema a la direcció IP fixe del servidor.

3.3.3. Estat de l'aplicació

Quan vam realitzar el pla de treball, i ens vam marcar les fites a seguir, vam calcular que en un període de 23 dies disposaríem de l'aplicació acabada. A conseqüència del volum de feina que hem hagut de suportar aquest últims mesos, ha produït que l'aplicació no estigui, a hores d'ara, al 100% acabada. En canvi, si que podem dir que resta al 98% i casi bé totes les funcionalitats o casos d'ús que vam esmentar, s'han anat codificant a temps.

Si ho dividim per mòduls:

- **Mòdul Client:**

Tots els casos d'ús han estat realitzats.
- **Mòdul Proveïdor:**

Tots els casos d'ús han estat realitzats.
- **Mòdul Full de Feina:**

Tots els casos d'ús han estat realitzats.
- **Mòdul Factura:**

Tots els casos d'ús han estat realitzats.
- **Mòdul Administrador:**

Tots els casos d'ús han estat realitzats.

- **Mòdul de Log in:**

Tots els casos d'ús han estat realitzats.

- **Mòdul de Log out:**

Resta la realització del cas d'ús.

En entregues anteriors, vam comentar que el nostre projecte estava al 90%, i disposava de certes mancances que havia detectat el client. Actualment, podem observar que el projecte roman al 98% i que resten, dos o tres punts per acabar de tancar. Aquestes actualitzacions s'hauran de realitzar fora del termini que es disposa.

Els punts que resten per realitzar són:

- Log Out de l'usuari de sessió.
- Retocar el codi, per a que tots els tipus d'usuaris del portal i tinguin accés. Recordem que actualment, només està habilitat l'usuari Administrador.
- Ajustar el control d'errors del portal.

3.3.4. Decisions de Disseny i d'Implementació

Com era d'esperar, durant l'implementació del portal hem tingut moments els quals hem vist oportú modificar les nostres previsions i perspectives en l'elaboració dels mòduls, com a conseqüència de possibles millores en eficiència o d'entrebancs relacionats amb el framework escollit.

- Un dels entrebancs que ens vam trobar, que ens va fer modificar part de l'estructura dels mòduls de Full de Feines i de Factura, va ser una "*OptionCollections*³⁰" d'Struts, que ens donava un error en fer el "submit" o validació de la pàgina JSP de factures. Aquest era un "*div*³¹" el qual ens mostrava un llistat dels clients del sistema, i que havíem programat de tal manera que, si seleccionàvem un client del llistat (fent doble clic), les seves dades es veien reflectides a uns camps pertinents, per a la futura creació de la Factura i Full de Feina.

Això ho vam solucionar, optant, per la solució més senzilla i eficaç de mostrar una llistat dels clients prèviament a la factura, que es composava de totes les dades necessàries per a la creació de la Factura i el Full de Feina. D'aquesta manera, vam evitar l'ús excessiu del "*javascript*" i vam reutilitzar codi que ja havíem fet anteriorment.

- Un altre entrebanc, que pot ser ens ha fet donar més voltes del projecte, ha estat a l'hora de realitzar el pdf de Factures. Una factura pot estar composta de "*n*" fulls de feina, cada full de feina pot disposar de "*m*" entrades de full. Per

tant, com podem observar podem disposar de “n” objectes **Fullfeina**[], i de “m” objecte **EntradaFull**[]. Posem un exemple:

Per a generar una factura d'un client que disposa de tres fulls de feina, els quals cada full de feina té sis línies **entradaFull**(composades dels atributs quantitat, concepte i import, mirar BD), tindrem tres objectes **Fullfeina** [3] i divuit objectes **EntradaFull**[18].

El llistat factura generat, es divideix de tal manera que cada full de feina es veu reflectit al llistat encapçalat per la data del full de feina i seguit de les línies objecte d'**EntradaFull**. Per a poder generar el llistat hem hagut de crear un objecte **BeanFacturaPDF** que conte dos objectes associats, és a dir, **Fullfeina**[], **EntradaFull**[] i d'aquesta manera a l'hora de generar el xml, per més tard utilitzar-lo en la generació del fitxer “xs” per a poder extreure el llistat en format “pdf”. Podem observar el mètode a la classe **FacturaAction.java**.

Aquests parell de problemes, pot ser han estat els més complicats de resoldre, tot i així també podríem esmentar entrebancs de més baix nivell com:

- “*Connection pool*” de la BD amb el portal.
- L'ús de la llibreria Tiles.
- Configuració del Log4j.
- Ús de la llibreria Fop i els seu jar's pertinents.

La solució a aquest dubtes els hem solventat, amb paciència i llegint multitud de pàgines web que hem anat trobant per la xarxa, que les podrem trobar en els annexes de la futura entrega final.

3.3.5. Consideracions necessàries per a l'Instal·lació.

3.3.5.1. Procés d'instal·lació

Per a instal·lar l'aplicació, primer hem de descarregar el programari que necessitem per a executar-l'ho.

Les llibreries que hem fet servir:

- Struts-Tiles 1.3.8.
- Log4j.
- JDom.
- Fop.
- DisplayTag.
- Connector PostGreSQL. (Adjunta en l'entrega)

no seran necessàries la seva descarrega, ja que venen en el fitxer *.war que subministrem en l'entrega.

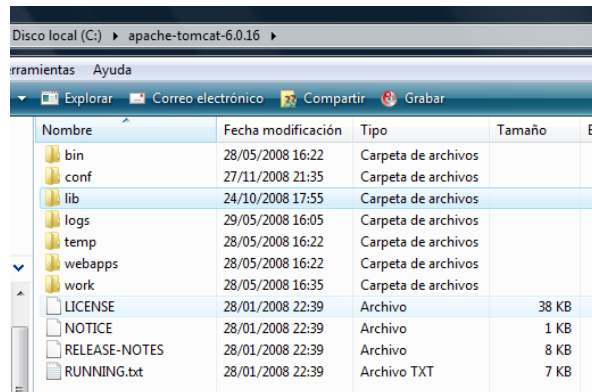
Seguidament haurem d'instal·lar el programari Apache-Tomcat-6.0.16, la versió 8.1 del PostGreSQL i el jdk-1.5.0.15 de Java o superior.

Un cop tenim instal·lat el Tomcat, anem al directori arrel de Tomcat i a la carpeta “lib”, posarem la llibreria que s'adjunta en el correu d'entrega, relacionada amb el connector PostgreSQL.

Un cop tenim les llibreries al seu lloc, passem a instal·lar l'aplicació.

La distribució es realitzarà a través d'un arxiu .war. Aquest arxiu s'ha de copiar en el directori adequat del servidor instal·lat prèviament per desplegar les aplicacions. Depenent de quina versió tinguem del servidor, ens serà necessari reiniciar-lo.

L'estructura de directoris del servidor Tomcat, és la següent:



Nombre	Fecha modificación	Tipo	Tamaño	Et
bin	28/05/2008 16:22	Carpeta de archivos		
conf	27/11/2008 21:35	Carpeta de archivos		
lib	24/10/2008 17:55	Carpeta de archivos		
logs	29/05/2008 16:05	Carpeta de archivos		
temp	28/05/2008 16:22	Carpeta de archivos		
webapps	28/05/2008 16:22	Carpeta de archivos		
work	28/05/2008 16:35	Carpeta de archivos		
LICENSE	28/01/2008 22:39	Archivo	38 KB	
NOTICE	28/01/2008 22:39	Archivo	1 KB	
RELEASE-NOTES	28/01/2008 22:39	Archivo	8 KB	
RUNNING.txt	28/01/2008 22:39	Archivo TXT	7 KB	

L'arxiu war, haurà d'anar col·locat a la carpeta “webapps” i les llibreries a la carpeta “lib” com dèiem anteriorment.

A l'estructura anterior hem d'assignar-la a un “context path” coincident amb el nom amb el que es voldrà accedir a l'aplicació mitjançant la URL. Cada aplicació web estarà associada a un context i tots els components existiran en relació a aquest context.

3.3.5.2. Base de Dades.

Per a connectar-se a la base de dades s'ha d'usar un nom i una clau. Aquests poden variar depenent com s'ha instal·lat el servidor. El valor de l'usuari i de la clau venen establerts en l'arxiu de configuració de l'aplicació “web.xml”, que apunta a “context.xml” que és on realment trobarem l'usuari i contrasenya. Observem-ho:

web.xml:

```
<resource-ref>
  <description>Connexio amb la BD</description>
  <res-ref-name>jdbc/TFC</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

context.xml:

```
<Resource name="jdbc/TFC" auth="Container" type="javax.sql.DataSource"
driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://192.168.1.28:5432/TFC" username="xavi"
password="j196mmbg" maxActive="20" maxIdle="10" maxWait="-1"/>
```

```
removeAbandoned="true"  
logAbandoned="true"/>
```

```
removeAbandonedTimeout="60"
```

El fitxer context.xml el podem trobar dins del war, a la següent ruta: **/TFC/WebContent/META-INF/** i un cop desplegat el producte dins del Tomcat, el fitxer a modificar estarà a **C:\apache-tomcat-6.0.16\webapps\tfc\META-INF**

3.3.5.2.1. Script de la Base de Dades.

```
CREATE TABLE client
(
  id_client serial NOT NULL,
  nom character varying(30),
  cognom1 character varying(30),
  cognom2 character varying(30),
  dniNif character varying(9),
  adresa character varying(100),
  codiPostal character varying(5),
  ciutat character varying(50),
  telefon1 character varying(9),
  telefon2 character varying(9),
  mobil1 character varying(9),
  mobil2 character varying(9),
  fax character varying(9),
  email1 character varying(40),
  email2 character varying(40),
  numCompte character varying(20),

  CONSTRAINT client_pk PRIMARY KEY (id_client)
)

CREATE TABLE proveidor
(
  id_proveidor serial NOT NULL,
  nom character varying(30),
  cognom1 character varying(30),
  cognom2 character varying(30),
  dniNif character varying(9),
  adresa character varying(100),
  codiPostal character varying(5),
  ciutat character varying(50),
  telefon1 character varying(9),
  telefon2 character varying(9),
  mobil1 character varying(9),
  mobil2 character varying(9),
  fax character varying(9),
  email1 character varying(40),
  email2 character varying(40),
  numCompte character varying(20),

  CONSTRAINT proveidor_pk PRIMARY KEY (id_proveidor)
)

CREATE TABLE factura
(
  id_factura serial NOT NULL,
  id_client integer NOT NULL,
  num_factura integer NOT NULL,
  data_factura date NOT NULL,
  base_imposable character varying(10),
  percent_iva character varying(2),
  iva character varying(10),
  total_factura character varying(13),
  forma_pagament character varying(20),
  domicili_pagament character varying(30),
  venciment date NOT NULL,
  nom_client varchar(30),
```

```
CONSTRAINT factura_pk PRIMARY KEY (id_factura),
CONSTRAINT factura_fk FOREIGN KEY (id_client)
REFERENCES client (id_client) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

CREATE TABLE fullFeina
(
  id_full serial NOT NULL,
  id_factura integer NOT NULL,
  data_full date NOT NULL,
  nom_client character varying(30),
  treball_realitzat character varying(500),
  hores_feina character varying(3),
  suma character varying(10),
  iva character varying(10),
  conforme character varying(10),
  total character varying(10),

  CONSTRAINT fullFeina_pk PRIMARY KEY (id_full),
  CONSTRAINT fullFeina_fk FOREIGN KEY (id_factura)
REFERENCES factura (id_factura) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

CREATE TABLE entrada_full
(
  id_desc serial NOT NULL,
  id_full integer NOT NULL,
  quantitat character varying(3),
  concepte character varying(200),
  import character varying(8),

  CONSTRAINT entrada_full_pk PRIMARY KEY (id_desc),
  CONSTRAINT entrada_full_fk FOREIGN KEY (id_full)
REFERENCES fullfeina (id_full) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

CREATE TABLE usuari
(
  id_usuari serial NOT NULL,
  nom_usuari character varying(20),
  contrasenya_usuari character varying(20),

  CONSTRAINT usuari_pk PRIMARY KEY (id_usuari)
)
```

3.3.5.3. Configuració de l'aplicació

L'aplicació disposa de varis fitxers de configuració, com:

- **web.xml fitxer descriptor de desplegament:** el contingut d'aquest fitxer és fonamental per el desplegament de l'aplicació, ja que descriu al contenidor web els seus elements i el mode en que s'accedeix a aquests. A més a més, defineix aspectes de seguretat, fitxers de benvinguda, paràmetres inicials, paràmetres de context... Quan Tomcat s'inicia, llegeix aquest fitxer i si conté algun error, es llancen excepcions que indiquen que el servidor no s'ha iniciat correctament.
- **struts-config.xml arxiu de configuració de Struts:** Arxiu de configuració de Struts. En ell, es mapegen les accions que realitzarà l'aplicació en cada

Action, es poden configurar paràmetres d'inicialització del servlet, es defineixen els form-beans de cada formulari, s'indica on s'ubica els fitxers de missatges. Tiles tot i tenir un fitxer de configuració, també s'ha de definir en el fitxer de configuració d'Struts...

3.3.5.4. Estructura de Directoris

El nostre sistema crearà i farà ús d'uns directoris on el sistema deixarà caure els fitxers de log, els fitxers xml creats, els fitxers pdf generats... i a més a més usarà una sèrie de fitxers *.xsl i *.xml i *.jpg, per a poder portar a bon port la creació dels llistats demanats.

Tot i que el sistema, es capaç de generar els directoris si no han estat creats, és del tot recomanable crear aquests directoris, i deixar els fitxers necessaris. Adjunt a l'entrega, trobarem les carpetes amb els fitxers necessaris per a que tot funcioni a la perfecció.

3.3.5.4.1. Logs

Els fitxers de log, es generen automàticament, i es van omplint a mesura que es va fent servir l'aplicació, i va emmagatzemant, tot el moviment del portal.

Allà es veuran reflectits tots els possibles errors no controlats amb anterioritat per part nostre.

El directori on podem trobar aquests fitxer és: "**C:\InstalacionsRiba_logsVogs**" on hauríem de veure:

- administrador.log.
- client.log
- proveidor.log
- factura.log
- feullFeina.log
- login.log

Aquests fitxers tindran com a màxim un pes de 1Mb, i un cop arribin al seu límit, es generaran fills, amb el mateix límit, i així progressivament. L'Administrador, tindrà accés a aquests arxius des de l'aplicació.

Un cop tinguem desplegat el producte dins del Tomcat, podrem trobar el fitxer amb la configuració del log4j a **C:\apache-tomcat-6.0.16\webapps\tfcVog4j** amb el nom de **log4j.properties**.

Observem un tros del fitxer de configuració:

```
log4j.logger.es.uoc.tfc.riba.login=DEBUG, login
log4j.logger.es.uoc.tfc.riba.client=DEBUG, client
log4j.logger.es.uoc.tfc.riba.proveidor=DEBUG, proveidor
log4j.logger.es.uoc.tfc.riba.fullFeina=DEBUG, fullFeina
log4j.logger.es.uoc.tfc.riba.factura=DEBUG, factura
log4j.logger.es.uoc.tfc.riba.administrador=DEBUG, administrador

log4j.appender.login=org.apache.log4j.RollingFileAppender
```



```
log4j.appender.login.File=/InstalacionsRiba_logs/logs/login.log
log4j.appender.login.MaxFileSize=1Mb
log4j.appender.login.MaxBackupIndex=5
log4j.appender.login.layout=org.apache.log4j.PatternLayout
log4j.appender.login.layout.ConversionPattern=%d{ISO8601} - %m%n
```

3.3.5.4.2. Full Feina

Per a la creació de fulls de feina, és necessiten quatre fitxers:

- FullFeina.xml --> creat des de l'aplicació tindrà un nom supeditat per la data i el client.
- FullFeina.xsl
- FullFeina_es_ES.xml
- FullFeina_ca_ES.xml

La llibreria Fop, utilitzarà aquests fitxers per a poder generar el llistat en format pdf.

Els pdf's els podrem trobar a "**C:\InstalacionsRiba\FitxersPDF\FullFeinaPdfs**" i la resta de fitxers a "**C:\InstalacionsRiba\FitxersXML\FullFeina**".

3.3.5.4.3. Factura

Per a la creació de factures, és necessiten quatre fitxers:

- Factura.xml --> creat des de l'aplicació, tindrà un nom supeditat per la data i el client.
- Factura.xsl
- Factura_es_ES.xml
- Factura_ca_ES.xml

La llibreria Fop, utilitzarà aquests fitxers per a poder generar el llistat en format pdf.

Els pdf's els podrem trobar a "**C:\InstalacionsRiba\FitxersPDF\FacturaPdfs**" i la resta de fitxers a "**C:\InstalacionsRiba\FitxersXML\Factura**".

3.3.5.4.4. Imatges

Els llistats mostraran imatges de l'entitat, i aniran a buscar-les a "**C:\InstalacionsRiba\Imatges**".

3.3.6. Jocs de Prova

3.3.6.1. Usuari

Com que de moment l'aplicació no està al 100%, l'usuari que es podrà fer servir per a fer proves serà l'usuari administrador. L'usuari Gestor i Empleat, es realitzaran en futures actualitzacions i entregues.

Usuari: xavi
Clau: xavi

L'usuari administrador pot fer servir totes les accions del portal, per tant podrà:

- Crear/Modificar/Eliminar/Llistar Client.
- Crear/Modificar/Eliminar/Llistar Proveïdor.
- Crear/Modificar/Eliminar/Llistar/CrearPDF Full de Feina.
- Crear/Modificar/Eliminar/Llistar/CrearPDF Factura.
- Llistar Fixers de Log.

Es podrà crear qualsevol usuari a la BD, per a poder fer proves.

3.3.6.2. Inserts a la BD

Seguidament, passem una sèrie de dades que s'hauran d'introduir a la BD, per poder realitzar les proves necessàries, a no ser que es vulguin introduir manualment des de l'aplicació.

3.3.6.2.1. Inserts Clients

```
insert into client(nom, cognom1, cognom2, dninif, adresa, codipostal, ciutat, telefon1, telefon2, mobil1, mobil2, fax, email1, email2, numcompte) values ('xavier', 'fernandez', 'soler', '46816148q', 'c/Mossen Cinto Verdaguer', '08980', 'Sant Feliu de Llobregat', '936666666', '936666667', '666666666', '666666667', '936666666', 'xfernandezs@uoc.edu', 'xfernandezs@uoc.edu', '0000-00-0000000');
```

```
insert into client(nom, cognom1, cognom2, dninif, adresa, codipostal, ciutat, telefon1, telefon2, mobil1, mobil2, fax, email1, email2, numcompte) values ('jesus', 'fernandez', 'soler', '46816148q', 'c/Mossen Cinto Verdaguer', '08980', 'Sant Feliu de Llobregat', '936666666', '936666667', '666666668', '666666669', '936666666', 'jfernandezs@uoc.edu', 'jfernandezs@uoc.edu', '0000-00-0000001');
```

3.3.6.2.2. Inserts Proveïdors

```
insert into proveïdor(nom, cognom1, cognom2, dninif, adresa, codipostal, ciutat, telefon1, telefon2, mobil1, mobil2, fax, email1, email2, numcompte) values ('dgalix', 'solutions', 'SL', '46816148q', 'c/Joan Maragall', '08980', 'Sant Feliu de Llobregat',
```

```
'936666632', '936666631', '666666615', '666666616', '936666632',  
'admin@digalix.com', 'xmen@digalix.com', '0000-00-0000321');
```

```
insert into proveidor(nom, cognom1, cognom2, dninif, adresa, codipostal, ciutat,  
telefon1, telefon2, mobil1, mobil2, fax, email1, email2, numcompte) values ('radical',  
'media', 'SL', '46816148q', 'c/Josep Tarradelles', '08230', 'Barcelona', '936661234',  
'936666543', '666666456', '666666232', '936666144', 'admin@radical-media.com',  
'info@radical-media.com', '0000-00-0000321');
```

3.3.6.2.3. *Inserts Usuaris*

Per poder accedir a l'aplicació s'haurà d'introduir el següent usuari a la BD, també podríem crear qualsevol usuari, i passar la fase de log in. Però això ja és decisió del professor.

```
insert into usuari (nom_usuari, contrasenya_usuari) values ('xavi', 'xavi');
```

3.4. Capítol 4, Valoració econòmica

Aquest projecte no té valoració econòmica per l'àmbit al que està destinat. Si s'hagués de comptabilitzar econòmicament s'hauria d'haver establert el projecte a preu tancat, donat que l'estimació del pla de projecte, disseny, implementació i proves han estat exhaustius i això comportaria que el preu si es comptabilitzés en hores es dispararia.

Tot i així, hem de tenir en ment que el producte final i la documentació tenen un nivell molt complet per un projecte tan petit, per tant és aquesta tasca de oficialitat la que implicaria un augment del cost més significatiu.

3.5. Capítol 5, Conclusions

Han estat sis anys ben llargs els que han passat, i per fi he arribat al final de túnel.

Aquest ha estat un treball final de carrera que m'ha fet canviar, m'ha fet créixer, en definitiva m'ha fet madurar com a persona. Ha estat un repte que s'ha assolit amb escreix, i ha fet fugir les pors que des d'un bon inici el meu cervell produïen.

Poder realitzar un projecte passant per totes les seves fases i fet per un mateix, sense haver de tenir ningú per sobre que et mani com i quan s'han de fer les coses, és una tasca que recomanaria a qualsevol persona. Ha estat un plaer poder veure créixer un projecte, dissenyat, implementat, depurat i documentat des de zero. És clar que el producte final no compleix mai les primeres expectatives, ni es pot arribar a deixar com un voldria, ja que són molts els afers (feina, compromisos personals...) que ens retallen el temps que voldríem o necessitaríem, per poder tancar el TFC a la perfecció. Tot i així, crec que això forma part del TFC, és una síntesi, no només de coneixements adquirits, sinó també de responsabilitat personal i administració del temps.

Per la part tècnica, el món del J2ee està molt vinculat a la feina que tinc actualment, per tant, mai està de més aprendre noves tecnologies o refrescar coneixements o parts del projecte que per la situació laboral vas descuidant.

Ha estat interessant, muntar aquest projecte des de zero i no dedicar-se només a la part d'implementació, i d'aquesta manera, poder valorar totes les fases d'un projecte i fent més amena la seva elaboració.

Concloent, aquesta ha estat una etapa molt interessant formativament parlant, que espero que tanqui un cicle de la meua vida de molt esforç, compaginant estudi i treball. Tant bo és el regust de boca que m'han deixat aquests anys d'estudi, que del més segur és que el curs vinent comenci una nova etapa cursant la superior a la UOC.

4. Glossari

1.	TFC	-	Treball Final de Carrera.
2.	ETIS	-	Enginyeria tècnica d'informàtica de Sistemes.
3.	J2ee	-	Java 2 Platform, Enterprise Edition.
4.	UAT	-	Test d'Acceptació d'Usuari.
5.	Scripts	-	Codi font que executa un procés.
6.	Tomcat	-	Contenidor de servlets de la fundació Apache.
7.	PostgreSQL	-	Servidor de Bases de Dades.
8.	Eclipse	-	Programari utilitzar per a programar el producte.
9.	API's	-	Interfície de Programació d'Aplicacions.
10.	Struts	-	Eina de desenvolupament d'aplicacions web.
11.	MVC	-	Patró de disseny.
12.	JDBC	-	Connectivitat java a bases de dades.
13.	BD	-	Bases de Dades.
14.	Log4j	-	Llibreria java, generadora de missatges de log.
15.	DAO	-	Data Access Objects.
16.	Java	-	Llenguatge de programació.
17.	Microsoft Project 07-	-	Eina de desenvolupament de diagrames.
18.	BBDD	-	Bases de dades.
19.	Deployar	-	Desplegar.
20.	War	-	Fitxer web comprimit.
21.	CSS	-	Cascading Style Sheets, Fulls d'estil.
22.	JSP	-	Pàgines web programades en java.
23.	DBMS	-	DataBase Management System.
24.	Queries	-	Sentències sql.
25.	Pool de connexions	-	Col·lecció de connexions establertes amb la BD.
26.	Threads	-	Fil d'execució.
27.	XSL-FO	-	Document XML, per a generar pdf.
28.	PDF	-	Portable Document Format.
29.	IU	-	Interfície d'usuari.
30.	OptionCollections	-	Col·lecció de radio buttons.
31.	Div	-	Tag d'HTML.

5. Bibliografia

Recursos lèxics:

Diccionari de l'"Institut d'Estudis Catalans":

<http://pdl.iec.es/>

Alguns recursos del "Consorti per a la Normalització Lingüística":

<http://www.cpnl.org/recursos/rec015.htm>.

Alguns recursos de l'Arquitectura Struts & Tiles usada:

<http://faq.javaranch.com/java/StrutsAndEclipse>.

http://www.javahispano.org/contenidos/es/tiles_en_struts/.

<http://struts.apache.org/1.x/struts-tiles/installation.html>.

http://www.myjavaserver.com/~aldosolis/tutorial_struts/struts_tutorial.html.

<http://www.laliluna.de/first-steps-with-struts-free-tools-en.html>.

Alguns recursos de llibreries externes:

Log4j:

<http://snippets.dzone.com/posts/show/3248>.

http://www.chuidiang.com/chuwiki/index.php?title=Ejemplo_con_log4j.

<http://www.proactiva-calidad.com/java/herramientas/log4j/index.html>.

http://www.geocities.com/neos_software/articles/j2se/log4j.html.

DAO:

http://www.chuidiang.com/chuwiki/index.php?title=Patrón_DAO.

<http://mdaog.sourceforge.net/demo/>.

JDOM:

<http://www.javaworld.com/javaworld/jw-07-2000/jw-0728-jdom2.html?page=3>.

http://www.programacion.com/articulo.php?id=joa_pdf.

<http://msdn.microsoft.com/es-es/library/ms256058.aspx>.

<http://www.w3.org/TR/xsl/>.

DisplayTag:

<http://displaytag.sourceforge.net/11/install.html>.

Alguns recursos BBDD:

<http://www.postgresql.org/docs/8.2/interactive/ddl-constraints.html>

6. Annexes

6.1. Annex I: Tecnologies Avaluades

I.1. Arquitectura de les aplicacions web

Les aplicacions web segueixen dos patrons de disseny o arquitectures. El primer i tradicional és el denominat Model 1.

I.1.1. Model 1

És el que tradicionalment s'ha utilitzat per a construir aplicacions web

Funciona de la següent manera:

Un usuari fa una petició a un recurs en un servidor web. Aquest recurs gestiona les dades rebudes, les valida, executa la "lògica de negoci" i retorni la resposta generada.

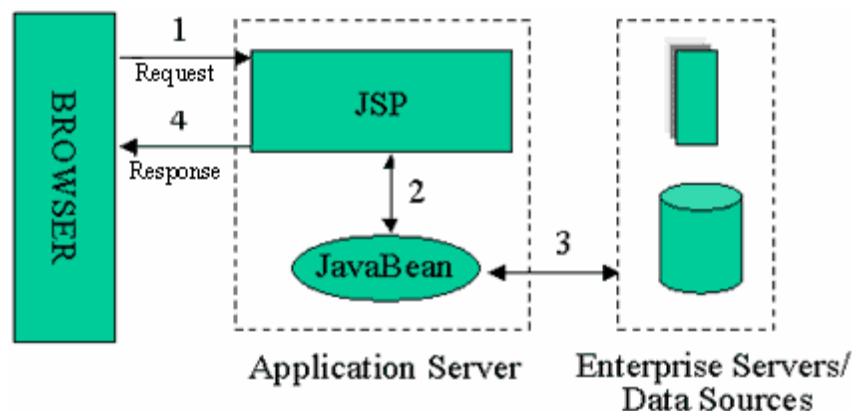


Fig I.1 Model 1

Aquest model pot ser d'una, dues o tres capes:

- Una capa: un únic element realitza les funcions de presentació de dades, processament i emmagatzemament.
- Dos capes: un element s'encarrega de la presentació i processament de la informació, mentre que la gestió de les dades la realitza una altra "capa".
- Tres capes: és la més habitual i la més convenient en quant a disseny. La capa de presentació s'encarrega de "presentar" les dades a l'usuari, la capa de lògica de negoci s'encarrega de processar les dades, i la capa d'accés a dades s'encarrega de gestionar la informació emmagatzemada en una base de dades o repositori de fitxers.

I.1.2. Model 2

El model 2 per les aplicacions web va sorgir per solucionar certes limitacions del primer model. Parteix de la base del disseny Model-Vista-Controlador, que separa la capa de presentació en dos elements: Vista i Controlador.

El funcionament del Model 2 és el següent:

- Un usuari fa una petició al servidor i aquesta és gestionada pel Controlador.
- El controlador decideix quin recurs contestarà la petició i l'habilita.
- El recurs seleccionat genera la resposta HTML i la retorna.

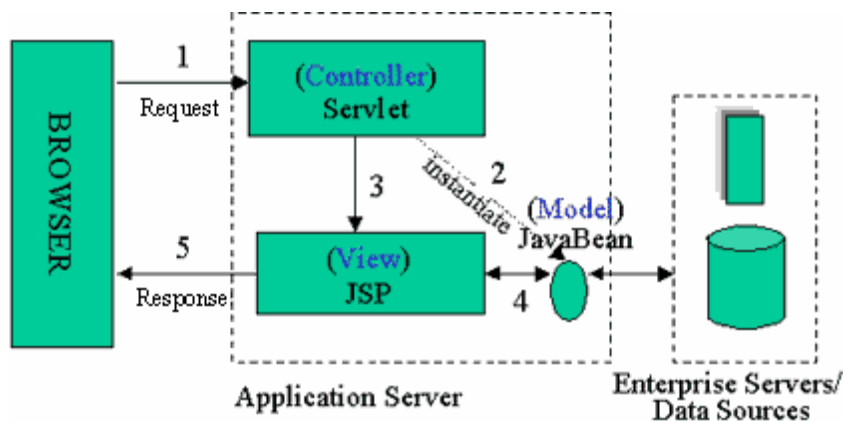


Fig 1.2 Model 2

Simplificant podem dir que el patró MVC es compon de varis blocs diferenciats:

- **Model:**
És el bloc lògic del sistema. S'encarrega de dur a terme les operacions que se li sol·liciten, processant les dades d'entrada i generant les respostes a retornar al client.
- **Vista:**
S'encarrega de recollir les dades generades pel model i presentar-les d'una forma adequada per a que l'usuari treballi amb elles. En el nostre cas seria el codi que genera el html que se li entregarà a l'usuari.
- **Controlador:**
Rep les peticions dels usuaris, sol·licita les accions a dur pel Model i selecciona quin element del bloc Vista respondrà a la petició. El sistema pot constar també d'una Capa de dades que s'encarregui d'emmagatzemar la informació que hagi de mantenir-se pel funcionament del sistema.

I.1.3. Model 1 vs. Model 2

Hem d'analitzar quin d'aquests dos models s'ajusta més a les nostres necessitats.

El model 1 és un sistema que funciona perfectament per a sistemes senzills. Tant funcionalment com en rendiment, podríem implementar el nostre sistema d'acord a aquest patró. En canvi, el model 2 ens ofereix certs avantatges:

- Al separar vista i control, provoca que els elements siguin més senzills i més fàcils de mantenir. Això en una aplicació amb funcionalitats no molt nombroses pot no ser crític. En canvi el sistema a desenvolupar ha de permetre una gran varietat d'operacions, a més a més de poder ser ampliable en un futur. Per tant, l'escalabilitat és un factor crític que hem de tenir en compte.

El bloc controlador permet escollir el recurs que respondrà a la petició, a més a més de validar les dades i altres operacions.

D'acord amb el model 1, el recurs que rep la petició (per exemple una pàgina JSP o un servlet) ha de generar també la resposta, sigui quina sigui aquesta.

Segons el model 2, la lògica en el bloc de la Vista es redueix. Per exemple: quan es produeixi un error serà una única pàgina JSP la que el presenti a l'usuari en comptes de gestionar l'error en totes les pàgines o redirigir la petició.

I.2. Frameworks de desenvolupament d'aplicacions web

I.2.1. Struts

La meta del projecte Apache Struts és afavorir el desenvolupament d'aplicacions basades en el "Model 2", aproximació del paradigma de disseny MVC. Per això han desenvolupat un framework de codi obert que implementa aquest model i permet facilitar el desenvolupament d'aplicacions basades en ell.

Struts està implementat en java i va sorgir com l'evolució dels servlets.

Els servlets de java van ser la primera revolució de java en les aplicacions web. Van suposar una millora en front a les aplicacions CGI, en quant a rendiment, portabilitat i flexibilitat.

En canvi la generació del codi html a través de servlets era incòmoda, i per a solucionar aquesta característica van sorgir les Java Server pages, que permetien barrejar codi html i java per facilitar el disseny d'interfícies.

El problema de les JSP's és que al contenir massa codi java es tornen pesades i difícils de mantenir, perdent les avantatges en quant a presentació.

Struts pretén solucionar aquest problema mitjançant la combinació d'ambdues tecnologies: servlets i jsp, aprofitant els punts forts de cadascun d'ells. Sota el model

MVC un servlet gestiona l'execució de la lògica de negoci mentre que la presentació recau principalment en les "Server pages" jsp.

I.2.1.1. Components

Seguint el patró de disseny MVC podem fer una diferenciació entre tres capes o mòduls d'Struts.

- **The Model**

Són les classes que contenen la "lògica de negoci" de l'aplicació. Dins d'aquest bloc podem englobar també les classes destinades a l'accés a dades (bases de dades o repositori).

És preferible no barrejar la lògica de negoci amb la de control.

- **The View**

Aquesta capa és la destinada a generar l'interfície d'usuari. El més habitual és utilitzar JSP i les etiquetes proporcionades per la Java Server Tag Library, JSTL. També podem barrejar codi en java però no és massa recomanable en quan a que dificulta el manteniment i comprensió de la pàgina.

A partir del JSP generarem el codi HTML i també les funcions en javascript que ens permetin interactuar amb les dades des de la banda del client.

- **The Controller: ActionServlet i ActionMapping**

El bloc de control s'encarrega de recollir les peticions entrants, decidir quin mètode de la lògica de negoci ha de ser executat i posteriorment decidir quin element de la Vista (quina JSP) ha de generar el codi HTML de resposta.

Els ActionMappings marquen el camí que ha de seguir una petició, és a dir, quin mètode de la lògica de negoci li correspon a una petició i quina jsp generarà la vista en funció de quin hagi estat el resultat obtingut a l'executar la lògica de negoci.

```
<action-mappings>
  <!-- Forwards to tilesGeneral.jsp -->
  <action      path="/client"
              scope="request"
              type="es.uoc.tfc.riba.client.action.ClientAction"
              name="clientForm">
    <forward name="success" path=".clients/client" />
  </action>
  .
  .
</action-mappings>
```

Fig I.3 Fragment de struts-config.xml

Els ActionServlets és el servlet que gestiona totes les peticions. Quan es sol·licita un recurs amb l'extensió .do el ActionServlet consulta en els ActionMappings el camí per a aquesta petició i la redirigeix per a que sigui gestionada pel sistema.

```
<!-- Standard Action Servlet Configuration -->
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
  <init-param>
    <param-name>application</param-name>
    <param-value>es.uoc.tfc.riba.messages.Messages</param-value>
  </init-param>
  <init-param>
    <param-name>chainConfig</param-name>
    <param-value>org/apache/struts/tiles/chain-config.xml</param-value>
  </init-param>
  <load-on-startup>2</load-on-startup>
</servlet>
```

Fig 1.4 Fragment de web.xml

I.2.2. Java Server Faces

Un altre framework java basat en el patró MCV és Java Server Faces (JSF). JSF és una combinació conceptual d'Struts i Swing.

Es sol dir que Struts està orientat a peticions, mentre que JSF està orientat a components. Podem veure la diferència en la manera de processar les peticions en la figura I.3.

Encara que el diagrama és més complexa que en el cas del Struts, podem veure a simple vista que els blocs segueixen complint tres funcions diferents: Model, Vista i Control, d'acord amb l'implementació del Model 2.

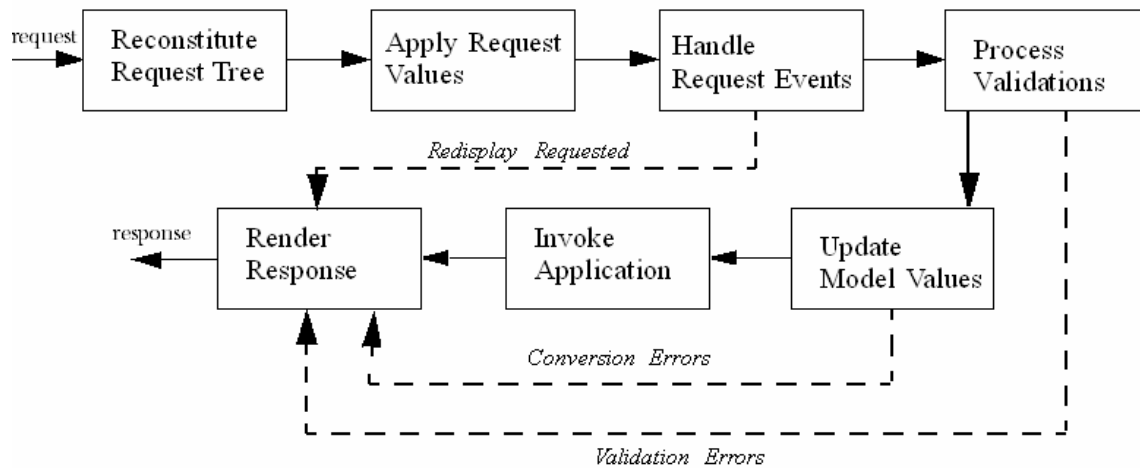


Fig 1.5 Cicle de vida de JSF

I.2.3. Struts vs. JSF

Podem dir que JSF és un framework molt més flexible que Struts, donat que va sorgir per evolucionar algunes limitacions d'aquest. Tot i que avui en dia hi han molts frameworks que s'han imposat a Struts, la maduresa i la major disponibilitat de recursos i suport fan que Struts, sigui una opció molt vàlida pel desenvolupament d'una eina de les característiques de la nostra plataforma.

I.3. DBMS

I.3.1 PostGreSQL

PostGreSQL és un gestor de bases de dades relacional. Com altres gestors, és un sistema prou madur i potent, que funciona en la majoria de plataformes que existeixen avui en dia.

PostGreSQL, és de codi obert i està distribuït sota la llicència BSD.

En quant a funcionalitats i prestacions, aquest DBMS s'ajusta als requeriments el projecte. A la xarxa podem trobar-lo de forma gratuïta amb una gran quantitat d'informació i eines, com pots ser el "PGAdmin", i a més a més del programari necessari per comunicar-se amb la BD.

I.3.2. MySQL

MySQL és un DBMS OpenSource desenvolupat, distribuït i mantingut per l'empresa MySQLLab. La versió Community Edition de MySQL es distribueix sota la llicència GPL de GNU.

Es tracta d'un gestor madur amb orígens en els anys 80 d'ús molt extens. S'ajusta als requeriments funcionals i de rendiment de la nostre aplicació i igual que en el cas anterior, disposem de forma gratuïta de quantitat de recursos en la xarxa que ens facilitarien el desenvolupament i manteniment de l'aplicació.

I.3.3. Elecció

Les dues alternatives exposades s'ajusten en quant a funcionament a les necessitats de l'aplicació. L'elecció d'una o altre dependrà d'altres factors.

Primer de tot, hem d'avaluar si la llicència sota la que es distribueixen els DBMS limita o afecta en alguna cosa l'ús que volem donar a l'aplicació desenvolupada. La nostra aplicació no serà distribuïda sinó que està destinada a l'ús intern de la companyia. Per tant, no resulta important la llicència sota la que es distribueix el DBMS.

De totes maneres, el disseny de la plataforma no ens limita en quant a l'ús d'un DBMS en concret, podent canviar en un futur, d'acord al caràcter estàndard de SQL.

Escollirem PostGreSQL, donat que ja tenim experiència prèvia amb aquesta tecnologia, cosa que no hem fet anteriorment amb MySQL. D'aquesta manera reduïm el risc de trobar-nos amb problemes durant l'implementació, com el cost d'aprenentatge.

6.2. Annex II: Informació ampliada dels casos d'ús del sistema

II.1. Casos d'ús.

II.1.1. Log in i Log out

- **Entrar al Portal**

Descripció: L'usuari introdueix el usuari i contrasenya, el sistema valida les dades i permet l'entrada o no.

Actors principals: Administrador, Gestors, Treballadors.

Actors secundaris: Cap.

Condicions prèvies: Cap.

Flux principal:

1. El sistema mostra la pantalla de benvinguda o entrada al sistema.
2. L'usuari introdueix les dades per a validar-se.
3. Si l'usuari existeix:
 - 3.1. Es redirigeix a la pàgina d'inici.
4. Si l'usuari no existeix:
 - 4.1. Es mostra per pantalla un missatge conforme l'usuari introduït no pertany al sistema.
 - 4.2. Es genera una traça, informant de l'acció, en el fitxer de log.

Condicions posteriors: L'usuari s'introdueix al sistema

Flux alternatiu: Cap.

- **Sortir del Portal**

Descripció: L'usuari selecciona l'opció de sortir del portal.

Actors principals: Administrador, Gestors, Treballadors.

Actors secundaris: Cap.

Condicions prèvies: L'usuari vol sortir realment del portal.

Flux principal:

1. L'usuari selecciona el botó de sortir del portal.
2. Es mostra un missatge conforme l'usuari està d'acord en sortir del sistema.
3. Es tanquen totes les connexions obertes i es buida la sessió, en cas que tingues informació.
4. Es torna a la pàgina d'inici de sessió.

Condicions posteriors: L'usuari es desvincula del portal.

Flux alternatiu: Cap.

II.1.2. Client

• Afegir Client

Descripció: L'usuari gestor, principalment, crea un nou client i l'emmagatzema a la base de dades.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El client encara no està enregistrat a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya clients.
2. El sistema mostra la pàgina amb les dades a omplir del client
3. El gestor omple els camps necessaris per a crear el client.
4. El gestor prem el botó de crear, acceptant l'operació.
5. Si les dades són correctes:
 - a. Si el client no existeix
 - i. Es crea el registre a la taula CLIENT.
 - b. Si el client existeix
 - i. Es mostra per pantalla un missatge d'error per pantalla.
6. Si les dades introduïdes no són correctes, es mostra un missatge d'error per pantalla.
7. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El client s'ha emmagatzemat a la base de dades.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Modificar Client

Descripció: L'usuari gestor, principalment, modifica les dades d'un client, ja emmagatzemat a la base de dades.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El client existeix a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya clients.
2. El gestor busca al client amb els filtres pertinents.
3. El sistema mostra una pàgina amb un llistat del client o clients depenent el filtre.
4. El gestor selecciona el botó de modificar el client.
5. El sistema mostra una pàgina amb les dades del client.
6. El gestor modifica les dades del client.
7. El gestor accepta els canvis realitzats.
8. Si els canvis realitzats són correctes:
 - 8.1. S'actualitzen els atributs de la taula CLIENT.
9. Si les dades no són correctes:
 - 9.1. Es mostra per pantalla un missatge d'error corresponent.
10. S'emmagatzema al fitxer de log l'acció realitzada.

Condicions posteriors: Les dades del client han estat modificades.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Eliminar Client

Descripció: L'usuari gestor, principalment, elimina un client del sistema.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El client existeix a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya clients.
2. El gestor busca al client amb els filtres pertinents.
3. El sistema mostra una pàgina amb un llistat del client o clients depenent el filtre.
4. El gestor prem el botó d'eliminar el client.
5. El sistema mostra una pàgina amb les dades del client.
6. El gestor prem el botó d'eliminar el client del sistema.
7. Si el client té factures obertes:
 - 7.1. El sistema mostra l'error pertinent per pantalla.
8. Si el client no té factures obertes:
 - 8.1. El sistema mostra per pantalla que el client ha estat eliminat del sistema.
9. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El client ha estat esborrat del sistema.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Llistar Client

Descripció: L'usuari gestor, principalment, llista el client del sistema.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El client existeix a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya clients.
2. El gestor omple els camps dels filtres per a realitzar la recerca.
3. El gestor prem el botó de cercar.
4. Si el client existeix en el sistema:
 - 4.1. El sistema mostra en una llista, el client cercat.
5. Si el client no existeix en el sistema:
 - 5.1. Es mostra per pantalla un missatge d'error pertinent.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El client es mostrat per pantalla.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Llistar Clients

Descripció: L'usuari gestor, principalment, llista els clients del sistema.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Que existeixin clients en el sistema.

Flux principal:

1. El gestor selecciona la pestanya clients.
2. El sistema mostra una pagina amb uns filtres i els botons pertinents.
3. El gestor prem el botó de cercar tots els clients.
4. El sistema mostra una llista amb tots els clients del sistema.
5. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat de tots els clients del sistema

Flux alternatiu: Error al realitzar l'operació a la base de dades.

II.1.3. Proveïdor

• Afegir Proveïdor

Descripció: L'usuari Gestor, principalment, crea un nou proveïdor i l'emmagatzema a la base de dades.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El proveïdor encara no està enregistrat a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya proveïdors.
2. El sistema mostra la pàgina amb les dades a omplir del proveïdor.
3. El gestor omple els camps necessaris per a crear el proveïdor.
4. El gestor prem el botó de crear, acceptant l'operació.
5. Si les dades són correctes:
 - 5.1. Si el proveïdor no existeix
 - 5.1.1. Es crea el registre a la taula PROVEIDOR.
 - 5.2. Si el proveïdor existeix
 - 5.2.1. Es mostra per pantalla un missatge d'error per pantalla.
6. Si les dades introduïdes no són correctes, es mostra un missatge d'error per pantalla.
7. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El proveïdor s'ha emmagatzemat a la base de dades.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Modificar Proveïdor

Descripció: L'usuari gestor, principalment, modifica les dades d'un proveïdor, ja emmagatzemat a la base de dades.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El proveïdor existeixi a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya proveïdors.
2. El gestor busca al proveïdor amb els filtres pertinents.
3. El sistema mostra una pàgina amb les dades del proveïdor.
4. El gestor modifica les dades del proveïdor.
5. El gestor accepta els canvis realitzats.
6. Si els canvis realitzats són correctes:
 - 6.1. S'actualitzen els atributs de la taula PROVEIDOR.
7. Si les dades no són correctes:
 - 7.1. Es mostra per pantalla un missatge d'error corresponent.
8. S'emmagatzema al fitxer de log l'acció realitzada.

Condicions posteriors: Les dades del proveïdor han estat modificades.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Eliminar Proveïdor

Descripció: L'usuari gestor, principalment, eliminar un proveïdor del sistema.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El proveïdor existeixi a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya proveïdors.
2. El gestor busca al proveïdor amb els filtres pertinents.
3. El sistema mostra una pàgina amb un llistat del proveïdor o proveïdors depenent el filtre.
4. El gestor prem el boto d'eliminar el proveïdor.
5. El sistema mostra una pàgina amb les dades del proveïdor.
6. El gestor prem el botó d'eliminar el proveïdor del sistema.
7. El sistema mostra per pàgina que el client ha estat eliminat del sistema.
8. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El proveïdor ha estat esborrat del sistema.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Llistar Proveïdor

Descripció: L'usuari gestor, principalment, llista el proveïdor del sistema.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: El proveïdor existeixi a la base de dades.

Flux principal:

1. El gestor selecciona la pestanya proveïdors.
2. El gestor omple els camps dels filtres per a realitzar la recerca.
3. El gestor prem el botó de cercar.
4. Si el proveïdor existeix en el sistema:
 - 4.1. El sistema mostra en una llista, el proveïdor cercat.
5. Si el proveïdor no existeix en el sistema:
 - 5.1. Es mostra per pantalla un missatge d'error pertinent.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El proveïdor es mostrat per pantalla.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Llistar Proveïdors

Descripció: L'usuari gestor, principalment, llista els proveïdors del sistema.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Que existeixin proveïdors en el sistema.

Flux principal:

1. El gestor selecciona la pestanya proveïdors.
2. El sistema mostra una pàgina amb els filtres i botons pertinents
3. El gestor prem el botó de cercar tots els proveïdors.
4. El sistema mostra una llista amb tots els proveïdors del sistema.
5. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat de tots els proveïdors del sistema

Flux alternatiu: Error al realitzar l'operació a la base de dades.

II.1.4. Factures

• Crear Factura Client

Descripció: L'usuari gestor, principalment, crea una factura per a un client.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Que el client existeixi en el sistema.

Flux principal:

1. El gestor selecciona la pestanya facturació.
2. El sistema mostra una pàgina amb els camps necessaris per a la facturació.
3. El gestor selecciona el client a qui vol crear la factura.
4. El gestor omple les dades necessàries per a crear la factura a un client.
 - 4.1. Si les dades són correctes:
 - 4.1.1. Es crea la factura pel client i s'emmagatzema.
 - 4.2. Si les dades no són correctes:
 - 4.2.1. Es mostra l'error pertinent per pantalla.
5. Es mostra un missatge per pantalla informant que la factura s'ha creat correctament.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: S'ha creat i emmagatzemat a la base de dades una factura per a un client.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Modificar Factura Client

Descripció: L'usuari gestor, principalment, modifica una factura d'un client.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Que el client i factura existeixen en el sistema.

Flux principal:

1. El gestor selecciona la pestanya facturació.
2. El sistema mostra la pàgina de facturació.
3. El gestor mitjançant els filtres, busca la factura del client.
4. El sistema mostra les factures del client, si en té.
5. El gestor selecciona la factura a modificar.
6. El sistema mostra una pàgina amb els camps de la factura.
7. El gestor modifica la factura.
8. Si les dades introduïdes són correctes:
 - 8.1. La factura és modifica en el sistema.
 - 8.2. Es mostra un missatge per pantalla informant de la modificació.
9. Si les dades no són introduïdes correctament:
 - 9.1. Es mostra un error per pantalla.
10. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: S'ha modificat les dades d'una factura per a un client en la base de dades.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• **Llistar Factura Client**

Descripció: L'usuari gestor, principalment, llista la factura d'un client.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir la factura en el sistema.

Flux principal:

1. El gestor selecciona la pestanya factures.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El gestor introdueix els camps filtres i prem el botó buscar.
4. Si les dades són correctes:
 - 4.1. El sistema mostra un llistat amb la factura del client.
5. Si les dades són errònies:
 - 5.1. El sistema mostra l'error pertinent per pantalla.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat de les factures del client

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• **Llistar Factures Client**

Descripció: L'usuari gestor, principalment, llista les factures d'un client.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir el client i les factures en el sistema.

Flux principal:

1. El gestor selecciona la pestanya factures.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El gestor introdueix els camps filtres i prem el botó buscar.
4. Si les dades són correctes:
 - a. El sistema mostra un llistat amb les factures del client.
5. Si les dades són errònies:
 - a. El sistema mostra l'error pertinent per pantalla.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat de les factures d'un client.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Crear Factura Pdf

Descripció: L'usuari gestor, principalment, crear un arxiu pdf d'una factura.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir la factura en el sistema.

Flux principal:

1. El gestor selecciona la pestanya factures.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El gestor busca la factura.
4. El gestor prem el botó de crear pdf.
5. Si l'acció falla:
 - a. Es mostra el missatge pertinent per pantalla.
6. Si l'acció es realitza correctament:
 - a. Es crea el pdf en la ruta pertinent del servidor
7. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra el fitxer pdf per pantalla i s'emmagatzema una còpia en el servidor.

Flux alternatiu: Error al realitzar l'operació.

II.1.5. Full Feina

• Crear Full Feina

Descripció: L'usuari treballador, principalment, crea un full de feina realitzat durant el dia.

Actors principals: Administrador, Gestors i treballadors.

Actors secundaris: Cap.

Condicions prèvies: Cap.

Flux principal:

1. El treballador selecciona la pestanya factures.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El treballador selecciona el botó de crear full en un client.
4. El sistema mostra una pàgina amb el camps a omplir.
5. El treballador omple els camps i prem el botó guardar.
6. Si les dades són correctes:

- 6.1. El full de feina és creat al sistema.
7. Si les dades són errònies:
 - 7.1. Es mostra el missatge d'error pertinent per pantalla.
8. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es crea un full de feina per un dia estipulat.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• **Modificar Full Feina**

Descripció: L'usuari treballador, principalment, modifica un full de feina.

Actors principals: Administrador, Gestors i treballadors.

Actors secundaris: Cap.

Condicions prèvies: El full de feina ha d'estar creat prèviament.

Flux principal:

1. El treballador selecciona la pestanya full de feines.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El treballador omple els filtres amb les dades necessàries.
4. El sistema mostra un llistat amb full seleccionat.
5. El treballador selecciona el botó modificar del full de feina.
6. El sistema mostra una pàgina amb el camps del full de feina.
7. El treballador modifica els camps i prem el botó modificar.
8. Si les dades són correctes:
 - a. El full de feina es modifica.
9. Si les dades són errònies:
 - a. Es mostra el missatge d'error pertinent per pantalla.
10. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El full de feina és modificat.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• **Eliminar Full Feina**

Descripció: L'usuari treballador, principalment, elimina un full de feina.

Actors principals: Administrador, Gestors i treballadors.

Actors secundaris: Cap.

Condicions prèvies: El full de feina ha d'estar creat prèviament.

Flux principal:

1. El treballador selecciona la pestanya full de feines.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El treballador omple els filtres amb les dades necessàries.
4. El sistema mostra un llistat amb full seleccionat.
5. El treballador selecciona del llistat el botó eliminar.
6. El sistema mostra una pàgina amb el camps del full de feina.
7. El treballador prem el botó eliminar.
8. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: El full de feina és eliminat del sistema.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Llistar Full Feina

Descripció: L'usuari treballador, principalment, llista un full de feina del sistema.

Actors principals: Administrador, Gestors i treballadors.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir el full de feina en el sistema.

Flux principal:

1. El treballador selecciona la pestanya full de feines.
2. El sistema mostra una pàgina amb uns camps filtres i els pertinents botons.
3. El treballador omple el camp filtres amb les dades pertinents i prem cercar.
4. Si les dades són correctes:
 - 4.1. El sistema mostra un llistat amb el full de feina buscat.
5. Si les dades són errònies:
 - 5.1. El sistema mostra l'error pertinent per pantalla.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat amb el full de feina sol·licitat.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Llistar Fulls Feina

Descripció: L'usuari treballador, principalment, llista els fulls de feina realitzats a un client.

Actors principals: Administrador, Gestors i treballadors.

Actors secundaris: Cap.

Condicions prèvies: Han d'existir els fulls de feina en el sistema.

Flux principal:

1. El treballador selecciona la pestanya full de feines.
2. El sistema mostra una pàgina amb uns camps filtres i els pertinents botons.
3. El treballador omple el camp filtres amb les dades pertinents i prem cercar tots.
4. Si les dades són correctes:
 - 4.1. El sistema mostra un llistat amb els fulls de feina del client.
5. Si les dades són errònies:
 - 5.1. El sistema mostra l'error pertinent per pantalla.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat amb els fulls de feina sol·licitats.

Flux alternatiu: Error al realitzar l'operació a la base de dades.

• Crear Full Feina Pdf

Descripció: L'usuari gestor, principalment, crear un arxiu pdf d'un full de feina.

Actors principals: Administrador, Gestors.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir el full de feina en el sistema.

Flux principal:

1. El gestor selecciona la pestanya full feines.
2. El sistema mostra una pàgina amb els filtres i botons pertinents.
3. El gestor busca el full de feina.
4. El gestor prem el botó de crear pdf.
5. Si l'acció falla:
 - a. Es mostra el missatge pertinent per pantalla.
6. Si l'acció es realitza correctament:
 - a. Es crea el pdf en la ruta pertinent del servidor
7. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra el fitxer pdf per pantalla i s'emmagatzema una còpia en el servidor.

Flux alternatiu: Error al realitzar l'operació.

II.1.6. Fitxers de Log

- Llistar Fitxers de Log

Descripció: L'usuari administrador, llista els fitxers de log generats pel sistema.

Actors principals: Administrador.

Actors secundaris: Cap.

Condicions prèvies: Han d'existir els fitxers de log en el sistema.

Flux principal:

1. L'Administrador selecciona la pestanya administrador.
2. El sistema mostra una pàgina amb uns camps filtres i els pertinents botons.
3. L'Administrador omple el camp filtres amb les dades pertinents i prem cercar tots.
4. Si les dades són correctes:
 - a. El sistema mostra un llistat amb els fitxers de log del sistema.
5. Si les dades són errònies:
 - a. El sistema mostra l'error pertinent per pantalla.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat amb els fitxers de log del sistema sol·licitats.

Flux alternatiu: Error al realitzar l'operació pertinent.

- Llistar Fitxer de Log

Descripció: L'usuari administrador, llista el fitxer de log generat pel sistema.

Actors principals: Administrador.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir el fitxer de log en el sistema.

Flux principal:

1. L'Administrador selecciona la pestanya administrador.
2. El sistema mostra una pàgina amb uns camps filtres i els pertinents botons.
3. L'Administrador omple el camp filtres amb les dades pertinents i prem cercar.
4. Si les dades són correctes:

- a. El sistema mostra un llistat amb el fitxer de log del sistema.
5. Si les dades són errònies:
 - a. El sistema mostra l'error pertinent per pantalla.
6. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra un llistat amb el fitxer de log del sistema sol·licitat.

Flux alternatiu: Error al realitzar l'operació pertinent.

- Veure Dades Fitxer de Log

Descripció: L'usuari administrador, observar les dades del fitxer de log generat pel sistema.

Actors principals: Administrador.

Actors secundaris: Cap.

Condicions prèvies: Ha d'existir el fitxer de log en el sistema.

Flux principal:

1. L'Administrador selecciona la pestanya administrador.
2. El sistema mostra una pàgina amb uns camps filtres i els pertinents botons.
3. L'Administrador omple el camp filtres amb les dades pertinents i prem cercar.
4. Si les dades són correctes:
 - a. El sistema mostra un llistat amb el fitxer de log del sistema.
5. Si les dades són errònies:
 - a. El sistema mostra l'error pertinent per pantalla.
6. Seguidament, el sistema mostra el contingut del fitxer de log.
7. S'emmagatzema l'acció realitzada al fitxer de log.

Condicions posteriors: Es mostra el contingut d'un fitxer de log del sistema sol·licitat.

Flux alternatiu: Error al realitzar l'operació pertinent.