

Aquest treball es distribueix sota llicència Creative Commons CC BY-SA
3.0
La llicència completa es pot consultar a
<http://creativecommons.org/licenses/by-sa/3.0/deed.ca>

TREBALL FINAL DE CARRERA: J2EE

GESTIÓ BANC SOLIDARI (GBS)



Miguel Salinas Vivancos
ETIG / ETIS

Salvador Campos Mazarico
13 de gener de 2014

Dedicatòria i agraïments

A l'Ana per demostrar-me com es podia, a la Dra. Isabel per ensenyar-me que podia, i en especial als mes pares, en Luis i la M^a Jose, per tots els anys de suport i constant confiança.

"Mostly, when you see programmers, they aren't doing anything. One of the attractive things about programmers is that you cannot tell whether or not they are working simply by looking at them. Very often they're sitting there seemingly drinking coffee and gossiping, or just staring into space. What the programmer is trying to do is get a handle on all the individual and unrelated ideas that are scampering around in his head."

- Charles M Strauss

Índex

Dedicatòria i agraïments	1
Índex.....	2
INTRODUCCIO.....	4
Resum	4
Justificació i context	5
Objectius.....	6
Abast del projecte	6
Enfocament i mètode seguit	7
Planificació.....	9
RELACIÓ D'ACTIVITATS	9
DIAGRAMA DE GANTT	10
ESPECIFICACIO DEL PROGRAMARI	11
ACTORS.....	11
CASOS D'ÚS.....	11
Subsistema Entrada/Sortida.....	12
Subsistema Administració	14
Subsistema Voluntaris.....	16
Subsistema Serveis	18
Subsistema Sol·licitud.....	21
Subsistema Informes.....	21
ARQUITECTURA DE L'APLICACIÓ	24
J2EE.....	24
Model MVC.....	25
FRAMEWORKS	26
Struts 2	26
Spring.....	27
Hibernate.....	27
Altres eines.....	28
ENTORN DE DESENVOLUPAMENT I PROGRAMARI	29

Eclipse.....	29
MySQL	29
JBoss	30
Diagrama de Classes.....	31
Model Relacional.....	32
Pantalles de l'usuari	33
Escalabilitat i futures millores	35
Conclusions.....	36
Objectius Assolits	36
Conclusions Personals	36
Bibliografia.....	38
Annex 1: Sentències DDL.....	39
Annex 2: Sentències DML.....	44
Annex 3: Manual d'instal·lació.	47

INTRODUCCIO

Resum

El projecte GBS es concep com la base d'una solució a una petició real formulada per una associació sense ànim de lucre, el Banc Solidari del barri de Les Corts de Barcelona.

Aquesta associació s'encarrega de posar en contacte a persones que tinguin la necessitat d'algun servei durant un espai curt de temps; això podria ser un acompanyament al metge, canviar un endoll, fer la declaració de la renda, recollir als nens de l'escola, etc. Així, apareix la figura del voluntari/a, que desinteressadament ofereix el seu temps i coneixements per realitzar aquestes tasques.

El Banc Solidari realitza aquesta feina de gestió amb unes eines ofimàtiques que, per la seva complexitat i en ser d'ús general, suposen un esforç per uns usuaris que no tenen experiència en aplicacions informàtiques.

Els voluntaris i els sol·licitants de serveis han de omplir formularis en paper, que posteriorment han de ser introduïts manualment en fulls de càlcul o una base de dades rudimentària per tal poder fer cerques.

Doncs bé, el projecte GBS pretén oferir l'inici d'una plataforma única des de la qual realitzar les principals operacions de l'entitat d'una manera simple i intuïtiva.

La solució escollida ha sigut una aplicació distribuïda, basada en la Web, un servidor que hostatjarà els fitxers de l'aplicació i una base de dades per emmagatzemar la informació ja existent i la que es generarà en el futur.

Justificació i context

La decisió de realitzar aquest projecte va sorgir ja fa molts anys per una estreta vinculació amb aquesta associació (el Banc Solidari). En els inicis de la meua etapa acadèmica se'm va demanar consell sobre com millorar la base de dades actual que disposaven i la realització d'una nova, però sempre partint del MS Access com a programari base.

A mesura que anava adquirint nous coneixements de les diferents assignatures, anava veient clar que una BD com el MS Access no podia oferir totes les funcionalitats que demanaven i que actualment desenvolupant a l'associació, així que la petició va quedar aparcada.

Arribats al TFC, era el moment d'escollir l'àrea i el tema.

L'arquitectura Java és present en milers de milions de dispositius i ha demostrat la seva robustesa i fiabilitat.

I per fi tenim l'oportunitat de construir una aplicació útil, funcional i fàcil d'utilitzar per als usuaris del Banc Solidari.

Aquesta aplicació permetrà reduir dràsticament la quantitat de paper ja que eliminarà les fitxes dels voluntaris, els llistats dels serveis oferts, informes, noves sol·licituds, etc.

També eliminarà un dels passos de omplir primer les dades en paper i després introduir-les a un programa. El temps es pot reduir a la meitat.

I finalment però no menys important, la centralització de les dades: amb una sola aplicació que contingui la informació de l'associació s'eviten duplicacions, pèrdues i consum innecessari de recursos.

Objectius

En aquest TFC es pretén realitzar la investigació de l'arquitectura J2EE: mitjançant un conjunt de capes on resideixen diferents tecnologies, podem desenvolupar aplicacions per gairebé qualsevol necessitat.

En el cas concret del projecte Gestió Banc Solidari farem ús d'aquesta arquitectura J2EE per coordinar les diferents tecnologies que faran falta.

L'objectiu és proposar una solució funcional i escalable per a les necessitats del Banc Solidari, realitzar el seu anàlisi de requeriments, l'especificació i finalment una implementació del que es podria anomenar "prototipus" o "esquelet".

Abast del projecte

Per definir l'abast del projecte s'ha de entendre que l'objectiu final és aconseguir una aplicació totalment funcional per a una associació real.

La quantitat de processos que es realitzen en el dia i a dia, els suggeriments de millores i la necessitat d'una adaptació gradual del programari en l'organització, ja avançaven que aquest projecte s'emmarcaria en un tipus prototipal.

Així, en aquest projecte es pretén:

- Establir el model relacional de la base de dades, no només pels objectes utilitzats en el projecte sinó per l'aplicació final.
- Estructurar l'arquitectura J2EE amb la integració de les diferents tecnologies, no totes s'utilitzen en el prototipus però seran importants en les futures versions de l'aplicació.
- Un sistema d'introducció de voluntaris, substituït de les fitxes en paper que s'omplen de manera presencial pel voluntari o per telèfon pel administrador.
- Un sistema d'introducció, consulta i modificació dels serveis que ofereixen els voluntaris. Encara que inicialment és un conjunt estàtic de valors (uns 100), la introducció d'aquesta aplicació obrirà noves possibilitats i una major personalització de l'atenció.
- Un conjunt d'informes definits prèviament per l'Equip Directiu de l'associació. S'inclouen una mostra i s'utilitza una tecnologia que permetrà l'exportació en altres formats diferents al web, com són el PDF, l'XLS o l'XML.
- Administració dels usuaris encarregats de gestionar l'aplicació i únics que tenen accés a les dades dels voluntaris, sol·licitants i altres informacions sensibles.

Enfocament i mètode seguit

Seguint la guia del Salvador, el tutor, i amb els coneixements adquirits en assignatures com Enginyeria del Programari, el desenvolupament del projecte s'ha dividit en fases que anaven aportant cos i entitat al resultat final.

Utilitzant el cicle de vida en cascada, el projecte ha passat per 6 etapes obtenint diversos productes en cadascuna d'elles:

Recollida de requeriments: aquesta primera etapa va consistir en entrevistes presencials amb membres de l'Equip encarregat del funcionament de l'associació. Varen expressar les seves necessitats i van proporcionar el material, tant físic com digital, amb el que treballen actualment.

Anàlisi i Especificació: s'analitza l'actual base de dades, el domini de l'aplicació, les funcions que realitzen els usuaris i es tenen en compte dues alternatives:

1. Mantenir la bases de dades en el MS Access, ja que els usuaris estan més acostumats a treballar amb aquest, i mirar d'incorporar les millores que demanen (incorporant taules, procediments).
2. Crear una aplicació web (utilitzant l'arquitectura J2EE) des de zero i així corregir errors inicials de disseny, eficiència, escalabilitat, etc.

S'opta per aquesta segona opció per augmentar les possibilitats d'un entorn multiusuari, més robust i menys dependent de la màquina.

Es creen els diferents casos d'ús; en el projecte s'inclouen en detall els que apareixen en aquesta versió de l'aplicació i es mencionen les de futures versions.

Investigació: s'ha de fer una especial menció en aquesta etapa per la seva rellevància. Encara que en l'abast d'aquest projecte el nombre destinat a la investigació quedi per sota a les hores d'implementació, és la meva opinió que en el conjunt del desenvolupament de tota una aplicació, aquesta fase és crítica. No només perquè posa les bases de la següent etapa, sinó perquè dona els coneixements per ara i per al futur.

Implementació: la fase d'implementació tradueix en codi d'un llenguatge concret l'especificació resultant de les etapes anteriors. En aquest projecte s'ha seguit el següent enfocament i ordre:

1. Integració de totes les tecnologies implicades.
2. Creació d'un cas d'ús (Login).
3. Desenvolupament d'un subsistema sencer (Serveis).
4. Replicació i adaptació dels diferents subsistemes.

Els tres primers passos han necessitat solapar-se amb l'anterior etapa d'Investigació, però com s'ha comentat anteriorment, una vegada arribat a cert punt, el codi era estable, els coneixements eren sòlids i només quedava realitzar el pas 4: replicar i adaptar.

En les futures versions, que no entren en l'abast d'aquest projecte s'inclouran jocs de proves, validacions, etc.

[Memòria i Presentació](#): finalment, es redacten els documents (incloent aquest) que expliquen tant a usuaris no informàtics com a altres desenvolupadors, una descripció del projecte. També s'inclou el codi d'aquesta versió inicial, o prototipus, de l'aplicació.

Planificació

A continuació es presenta la planificació del projecte, corresponent als mesos que dura el TFC.

RELACIÓ D'ACTIVITATS

En aquest apartat es mostra la descomposició d'activitats utilitzant la metodologia del desenvolupament en cascada, que consisteix en realitzar les diferents fases del projecte una darrere de l'altre.

Les activitats o tasques seran les següents:

Id	EDT	Nombre de tarea	Duración	Comienzo	Fin
1	1	TFC-J2EE	62,25 días	lun 23/09/13	lun 13/01/14
2	1.1	Pla de Treball	4,94 días	lun 23/09/13	mié 02/10/13
3	1.1.1	Redacció Esborrany	3,5 días	lun 23/09/13	vie 27/09/13
4	1.1.1.1	Planificació	10 horas	lun 23/09/13	mar 24/09/13
5	1.1.1.2	Objectius i Requisits	8 horas	jue 26/09/13	vie 27/09/13
6	1.1.1.3	Revisió Esborrany	2 horas	vie 27/09/13	vie 27/09/13
7	1.1.2	Correcció i Maquetació	3 horas	lun 30/09/13	lun 30/09/13
8	1.1.3	Entrega PAC1	0 días	mié 02/10/13	mié 02/10/13
9	1.2	Investigació i aprenentatge	23,5 días	lun 07/10/13	sáb 16/11/13
10	1.2.1	Investigació JSF	40 horas	lun 07/10/13	mar 15/10/13
11	1.2.2	Investigació Struts	50 horas	mar 15/10/13	vie 25/10/13
12	1.2.3	Investigació Spring	60 horas	sáb 26/10/13	vie 08/11/13
13	1.2.4	Investigació Hibernate	40 horas	vie 08/11/13	sáb 16/11/13
14	1.3	Anàlisi i Especificació de Requisits	18 días	lun 07/10/13	jue 07/11/13
15	1.3.1	Anàlisi funcionalitats	8 horas	lun 07/10/13	mar 08/10/13
16	1.3.2	Disseny Casos d'Ús	20 horas	mar 08/10/13	vie 11/10/13
17	1.3.3	Disseny Diagrama de Classes	10 horas	vie 11/10/13	lun 14/10/13
18	1.3.4	Disseny BBDD	10 horas	lun 14/10/13	mié 16/10/13
19	1.3.5	Disseny Diagrama de Seqüència	8 horas	mié 16/10/13	vie 18/10/13
20	1.3.6	Arquitectura de l'Aplicació	14 horas	vie 18/10/13	lun 21/10/13
21	1.3.7	Prototipus entorn Web	10 horas	mar 22/10/13	mié 23/10/13
22	1.3.8	Revisió i Correcció Document	4 horas	mié 23/10/13	jue 24/10/13
23	1.3.9	Entrega PAC2	0 días	jue 07/11/13	jue 07/11/13
24	1.4	Implementació	36 días	mié 23/10/13	vie 27/12/13
25	1.4.1	Instal·lació IDE+servidor+SGBD	8 horas	mié 23/10/13	jue 24/10/13
26	1.4.2	Creació Base de Dades	6 horas	jue 24/10/13	vie 25/10/13
27	1.4.3	1er Cas d'ús: Accés	4 días	vie 25/10/13	vie 01/11/13
28	1.4.3.1	Desenvolupament	18 horas	vie 25/10/13	mié 30/10/13
29	1.4.3.2	Usabilitat i Gestió d'errors	8 horas	mié 30/10/13	jue 31/10/13
30	1.4.3.3	Proves	6 horas	jue 31/10/13	vie 01/11/13
31	1.4.4	2on Cas d'ús: Alta	3,25 días	vie 01/11/13	jue 07/11/13
32	1.4.4.1	Desenvolupament	16 horas	vie 01/11/13	mar 05/11/13
33	1.4.4.2	Usabilitat i Gestió d'errors	6 horas	mar 05/11/13	mié 06/11/13
34	1.4.4.3	Proves	4 horas	mié 06/11/13	jue 07/11/13
35	1.4.5	3er Cas d'ús: Informe	2,75 días	jue 07/11/13	mar 12/11/13
36	1.4.5.1	Desenvolupament	12 horas	jue 07/11/13	lun 11/11/13
37	1.4.5.2	Usabilitat i Gestió d'errors	6 horas	lun 11/11/13	lun 11/11/13
38	1.4.5.3	Proves	4 horas	lun 11/11/13	mar 12/11/13
39	1.4.6	Desenvolupament resta casos d'ús	70 horas	mar 12/11/13	mié 27/11/13
40	1.4.7	Prova Casos d'ús	12 horas	mié 27/11/13	mar 03/12/13
41	1.4.8	Finalització Capa Web	16 horas	mar 03/12/13	vie 06/12/13
42	1.4.9	Poblar base de dades	6 horas	vie 06/12/13	sáb 07/12/13
43	1.4.10	Proves i Test	12 horas	sáb 07/12/13	mar 10/12/13
44	1.4.11	Correccions	6 horas	mar 10/12/13	mié 11/12/13
45	1.4.12	Entrega PAC3	0 días	lun 16/12/13	lun 16/12/13
46	1.5	Documentació	15,19 días	mié 18/12/13	lun 13/01/14
47	1.5.1	Memòria	10,75 días	mié 18/12/13	lun 06/01/14
48	1.5.1.1	Index i revisió de les PAC	3 horas	mié 18/12/13	mié 18/12/13
49	1.5.1.2	Modificacions dels continguts	8 horas	mié 18/12/13	jue 19/12/13
50	1.5.1.3	Valoració TFC i Propostes futur	6 horas	vie 20/12/13	vie 20/12/13
51	1.5.1.4	Correcció i Maquetació	6 horas	vie 20/12/13	lun 23/12/13
52	1.5.1.5	Manual d'ús	16 horas	jue 02/01/14	lun 06/01/14
53	1.5.2	Presentació	4,5 días	lun 06/01/14	lun 13/01/14
54	1.5.2.1	Elecció del format i l'aplicació	6 horas	lun 06/01/14	lun 06/01/14
55	1.5.2.2	Guió i continguts	8 horas	lun 06/01/14	mié 08/01/14
56	1.5.2.3	Desenvolupament i correcció	16 horas	mié 08/01/14	sáb 11/01/14
57	1.5.3	Entrega Final	0 días	lun 13/01/14	lun 13/01/14

FIGURA 1: ACTIVITATS DEL PROJECTE

DIAGRAMA DE GANTT

El diagrama de Gantt serveix per visualitzar de manera ràpida i concisa la temporització de les tasques i la quantitat càrrega de treball en les diferents setmanes. Aquesta és la planificació inicial que es va realitzar durant l'etapa de Recollida de requeriments, en les conclusions es farà menció de les variacions sofertes finalment.

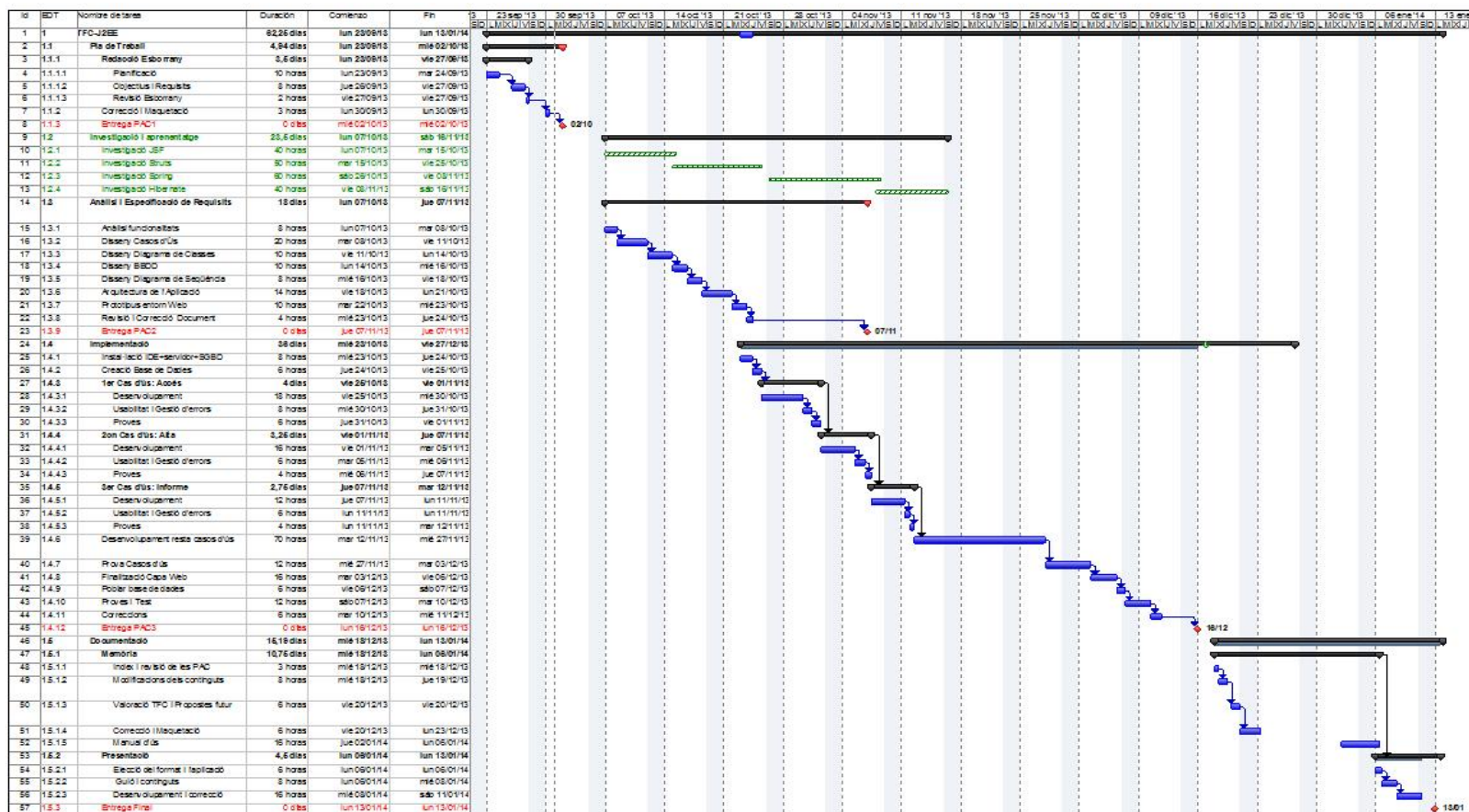


FIGURA 2: DIAGRAMA DE GANTT

ESPECIFICACIO DEL PROGRAMARI

ACTORS

L'aplicació Gestió Banc Solidari només disposa d'un actor: l'**Administrador**. Això és degut a la pròpia mida de l'associació i el fet que totes les persones encarregades del funcionament del sistema (que entre ells s'anomenen Grup Motor) realitzen la majoria dels processos.

És cert que dintre del mencionat Grup Motor es podria establir certa jerarquia, i que algun dels membres pren més protagonisme en la utilització de les TIC (pel fet de tenir més coneixements informàtics); però s'ha decidit implantar un sol actor ja que la pròpia dinàmica de les associacions facilita la renovació continua del personal, i es preferible simplificar el sistema amb "*un usuari-tots els permisos*".

Un suggeriment per part del tutor es la d'ampliar el nombre d'actors per tal de tenir en compte altres possibilitats, com invitats que puguin veure com funciona l'aplicació o *superusuaris* o administrador d'administradors.

En aquest projecte es manté un sol actor però es possible que en el continu desenvolupament de l'aplicació s'inclouï aquest rol de *superusuari*, per el supòsit que tots els administradors es donin de baixa a si mateixos i s'hagin de crear de nous.

L'actor invitat es descarta per el moment ja que l'associació treballa amb dades personals i l'encaix de la Llei de Protecció de Dades s'hauria de revisar amb l'Equip Directiu.

CASOS D'ÚS

Amb la descripció dels casos d'ús es pretén mostrar la interacció entre el programari i els actors, que en el nostre cas és l'Administrador.

En l'aplicació GBS dividirem els diferents CdU en **sis subsistemes** responnent als diversos objectes/processos del domini. Aquests subsistemes seran:

1. Entrada/Sortida
2. Administració
3. Voluntaris
4. Serveis
5. Sol·licitud
6. Informes

A continuació es farà la descripció dels corresponents casos d'ús per cadascun dels subsistemes. Aquí s'inclouen en detall els subsistemes de l'abast del projecte i es fa menció d'aquells que l'aplicació ja té en compte, tant en el seu model relacional, com en la arquitectura J2EE (encara que siguin funcionals).

Subsistema Entrada/Sortida

Aquest subsistema s'encarregarà de gestionar la vida dels usuaris en l'aplicació. Mitjançant uns valors assignats a cadascun d'ells i validats a la base de dades, es podrà accedir al programa i operar amb normalitat.

Una senzilla tècnica que ofereixen les aplicacions web per realitzar aquesta gestió es emmagatzemar les "credencials" de l'usuari en **la sessió de l'aplicació**. Així, si es volgués, en cadascuna de les pàgines que es visitessin es podria realitzar una comprovació de que la sessió està activa i l'usuari té els permisos, per exemple.

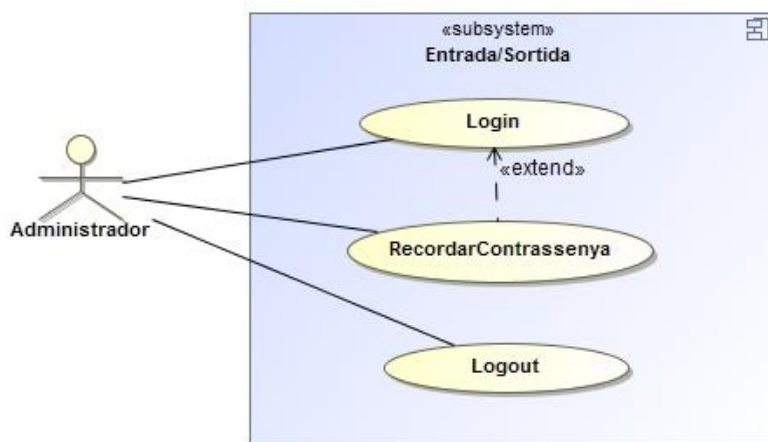


FIGURA 3: SUBSISTEMA ENTRADA/SORTIDA

Cas d'ús número 1: "Login"	
<i>Resum de la funcionalitat:</i>	Permet a un Administrador l'entrada a l'aplicació.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	L' Administrador ha de estar donat d'alta.
<i>Postcondició:</i>	L' Administrador ha accedit a l'aplicació.
<i>Procés normal:</i>	L'Administrador introdueix el seu nom d'usuari i la seva contrasenya. Aquestes dades són validades amb el sistema (juntament amb la data d'alta) i si són correctes

	es permet l'accés.
<i>Alternatives de procés:</i>	Si les dades proporcionades no són correctes es torna a demanar la introducció de les dues dades (usuari i contrasenya).

Cas d'ús número 2: "Logout"	
<i>Resum de la funcionalitat:</i>	Permet a l'usuari abandonar l'aplicació d'una manera segura.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	L' Administrador ha de estar autènticat.
<i>Postcondició:</i>	L' Administrador queda fora de l'aplicació i la sessió es tanca.
<i>Procés normal:</i>	L'Administrador demana sortir de l'aplicació, el sistema comprova que aquest té una sessió oberta i la tanca.
<i>Alternatives de procés:</i>	-

Subsistema Administració

El següent subsistema inclourà els procediments de gestió dels actors, es a dir, els administradors del sistema. Qualsevol administrador podrà donar d'alta nous administradors que tindran els permisos necessaris per realitzar totes les funcions de l'aplicació (encara que per l'abast d'aquest projecte, de moment no es fa el control dels permisos).

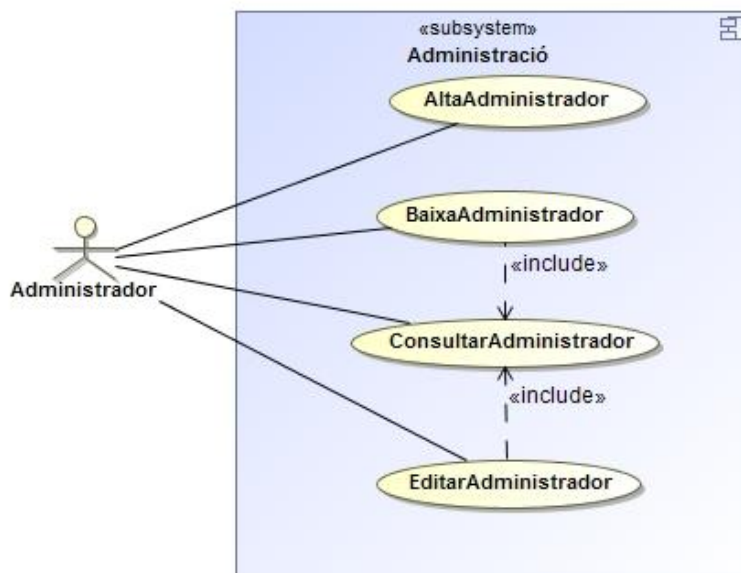


FIGURA 4: SUBSISTEMA ADMINISTRACIÓ

Cas d'ús número 3: "AltaAdministrador"	
<i>Resum de la funcionalitat:</i>	Permet donar d'alta un nou Administrador en el sistema.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	El nou usuari no pot existir en el sistema.
<i>Postcondició:</i>	S'ha creat un nou Administrador .
<i>Procés normal:</i>	L' Administrador introdueix el nom, els cognoms, un nom d'usuari, una adreça electrònica, un DNI i una contrasenya. El sistema comprova que les dades tenen el format correcte i que no existeix cap Administrador que coincideixi amb aquest nou usuari. Si tot es correcte, es guarden les dades en el sistema i es surt del procés.
<i>Alternatives de procés:</i>	Si alguna de les dades no compleix el format corresponent (com el DNI), es demana una rectificació a

	l'usuari.
--	-----------

Cas d'ús número 4: "BaixaAdministrador"

<i>Resum de la funcionalitat:</i>	Permet donar de baixa un Administrador .
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	<u>ConsultarAdministrador</u>
<i>Precondició:</i>	El Administrador que està realitzant el procés no pot ser el mateix que serà donat de baixa. Aquest ha de estar en alta en el moment actual.
<i>Postcondició:</i>	L' Administrador que s'ha donat de baixa té una nova data de baixa.
<i>Procés normal:</i>	L' Administrador crida al CdU <u>ConsultarAdministrador</u> per trobar a l'usuari corresponent. Quan l'ha trobat, el selecciona i torna al CdU <u>BaixaAdministrador</u> on introdueix la data de baixa (per defecte és l'actual). El sistema demana confirmació i si l' Administrador accepta es guarda la nova data de baixa.
<i>Alternatives de procés:</i>	

Cas d'ús número 5: "ConsultarAdministrador"

<i>Resum de la funcionalitat:</i>	Permet consultar les dades dels Administradors .
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició</i>	-
<i>Postcondició:</i>	-
<i>Procés normal:</i>	El sistema retorna un llistat amb els noms de tots els administradors. L' Administrador selecciona un i el sistema mostra totes les dades d'aquest, excepte la contrasenya.
<i>Alternatives de procés:</i>	

Cas d'ús número 6: "EditarAdministrador"

<i>Resum de la funcionalitat:</i>	Permet modificar les dades dels Administradors .
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	<u>ConsultarAdministrador</u>
<i>Precondició</i>	-

<i>Postcondició:</i>	No es permet donar de baixa en aquest CdU.
<i>Procés normal:</i>	<p>L'Administrador crida al CdU <u>ConsultarAdministrador</u> per trobar a l'usuari corresponent.</p> <p>Quan l'ha trobat, el selecciona i torna al CdU <u>EditarAdministrador</u> on modifica els camps que siguin necessaris.</p> <p>El sistema demana confirmació i si l'Administrador accepta es guarda les noves dades.</p>
<i>Alternatives de procés:</i>	

Subsistema Voluntaris

El subsistema Voluntaris serà una de les principals eines de l'aplicació i d'ús més exhaustiu, ja que per la pròpia casuística de l'associació, la renovació dels voluntaris es constant i es necessiten realitzar operacions d'alta, baixa i modificació totes les setmanes. Establint aquests casos d'ús com a base s'haurà d'esperar a les primeres proves amb els usuaris per fer modificacions segons la seva usabilitat.

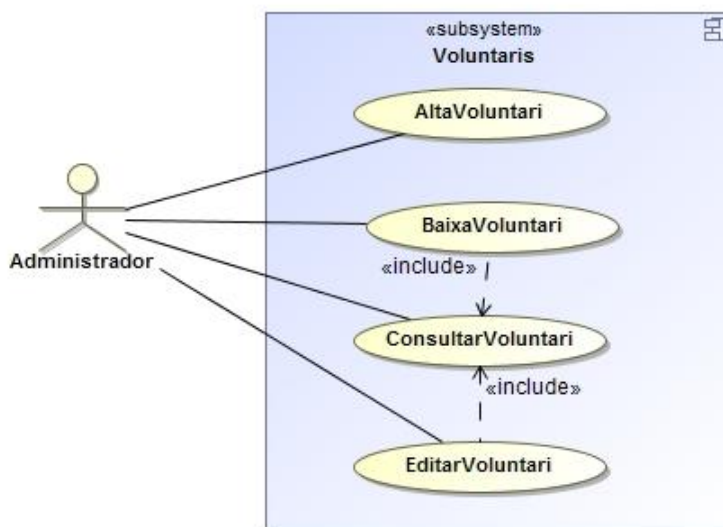


FIGURA 5: SUBSISTEMA VOLUNTARIS

Cas d'ús número 7: "AltaVoluntari"	
<i>Resum de la funcionalitat:</i>	Permet donar d'alta un nou Voluntari en l'aplicació.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	-

<i>Postcondició:</i>	No hi ha dos Voluntaris amb el mateix DNI/NIF.
<i>Procés normal:</i>	<p>L'Administrador introdueix les dades del nou usuari, es realitza una primera comprovació de format en camps com el del DNI o els cognoms que no poden estar buits. Posteriorment es realitza una comparació amb els registres del sistema per saber si l'usuari ja existeix, i si no és així, se li demana confirmació a l'Administrador per donar d'alta el nou Voluntari.</p> <p>El sistema introdueix en el registre informació addicional com la data i l'usuari administrador.</p>
<i>Alternatives de procés:</i>	Si el procés principal detecta que l'usuari que s'està introduint pot existir actualment en el sistema, s'ofereix la opció a l'Administrador de visualitzar la concordança.

Cas d'ús número 8: "BaixaVoluntari"

<i>Resum de la funcionalitat:</i>	Permet donar de baixa un Voluntari per tal que no realitzi més serveis.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	<u>ConsultarVoluntari</u>
<i>Precondició:</i>	El Voluntari ha d'existir i estar d'alta actualment.
<i>Postcondició:</i>	El Voluntari té una data de baixa superior a la data d'alta.
<i>Procés normal:</i>	<p>L'Administrador crida al CdU <u>ConsultarVoluntari</u> per trobar a l'usuari corresponent. El sistema els retornarà tots.</p> <p>Quan l'ha trobat el selecciona però el sistema només li permetrà fer-ho si el Voluntari encara està en estat d'alta. Si és així, torna al CdU <u>BaixaVoluntari</u> on introdueix una data de baixa i un motiu. També ha d'especificar si la baixa serà permanent o temporal.</p> <p>El sistema demana confirmació i si l'Administrador accepta es guarden les noves dades.</p>
<i>Alternatives de procés:</i>	-

Cas d'ús número 9: "ConsultarVoluntari"

<i>Resum de la funcionalitat:</i>	Permet visualitzar la informació d'un Voluntari segons un
-----------------------------------	---

	o més criteris.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	-
<i>Postcondició:</i>	-
<i>Procés normal:</i>	L'Administrador introdueix algun criteri en els camps de cerca i el sistema retorna tots aquells Voluntaris que compleixen aquelles condicions.
<i>Alternatives de procés:</i>	-

Cas d'ús número 10: "EditarVoluntari"	
<i>Resum de la funcionalitat:</i>	Permet modificar les dades d'un Voluntari.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	ConsultarVoluntari
<i>Precondició:</i>	-
<i>Postcondició:</i>	El camp de la data de baixa no és pot modificar.
<i>Procés normal:</i>	<p>L'Administrador crida al CdU <u>ConsultarVoluntari</u> i fa la cerca del Voluntari que vol modificar.</p> <p>Una vegada l'ha trobat, el selecciona i retorna al CdU <u>EditarVoluntari</u>. En aquest modifica els camps que siguin necessaris i envia els canvis al sistema. Es realitzen les comprovacions pertinents com ara la duplicació de DNI, camps buits, etc. i si tot és correcte es demana confirmació a l'Administrador.</p> <p>Quan aquest el dona, els canvis queden guardats en el sistema.</p>
<i>Alternatives de procés:</i>	-

Subsistema Serveis

El principal objectiu del Banc Solidari es posar en contacte a persones que tenen una necessitat i voluntaris que estan disposats a realitzar aquests serveis. Aquest subsistema s'encarregarà de mantenir la informació de tots els tipus de serveis que s'estan oferint des de l'associació.

Actualment, amb el sistema antic, el nombre de serveis és fixe, però quan l'aplicació GBS s'implanti ja no només quedaran accessibles els serveis que per defecte "mostra" l'associació, si no també aquells nous que van sorgint de les noves sol·licituds.

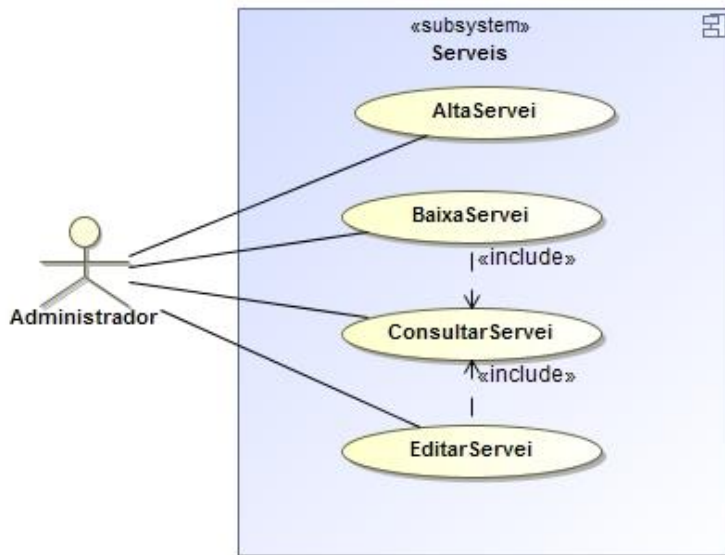


FIGURA 6: SUBSISTEMA SERVEIS

Cas d'ús número 11: "AltaServei"	
<i>Resum de la funcionalitat:</i>	Permet donar d'alta un nou Servei.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	El servei no està donat d'alta i no existeix en el sistema.
<i>Postcondició:</i>	-
<i>Procés normal:</i>	<p>L'Administrador introdueix un nom i una descripció per al nou Servei. El sistema comprova que aquest servei no existeixi actualment i demana una categoria.</p> <p>L'Administrador la introdueix i el sistema demana una confirmació final. Si és acceptada, les dades queden enregistrades.</p>
<i>Alternatives de procés:</i>	-

Cas d'ús número 12: "BaixaServei"	
<i>Resum de la funcionalitat:</i>	Permet donar de baixa un Servei existent.
<i>Actors:</i>	Administrador

<i>Casos d'ús relacionats:</i>	<u>ConsultarServei</u>
<i>Precondició:</i>	El Servei existeix i està en estat d'alta.
<i>Postcondició:</i>	La data de baixa és posterior a la data d'alta.
<i>Procés normal:</i>	L'Administrador crida el CdU <u>ConsultarServei</u> i mitjançant aquest troba el Servei que vol donar de baixa. Introdueix una data de baixa i el sistema li demana confirmació.
<i>Alternatives de procés:</i>	-

Cas d'ús número 13: "ConsultarServei"

<i>Resum de la funcionalitat:</i>	Permet consultar la informació sobre un Servei determinat.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	-
<i>Postcondició:</i>	-
<i>Procés normal:</i>	L'Administrador introdueix uns criteris de cerca per un o més camps del formulari retornat pel CdU. El sistema mostrarà els Serveis que compleixin tots aquests criteris.
<i>Alternatives de procés:</i>	-

Cas d'ús número 14: "EditarServei"

<i>Resum de la funcionalitat:</i>	Permet modificar les dades d'un Servei.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	<u>ConsultarServei</u>
<i>Precondició:</i>	El Servei existeix.
<i>Postcondició:</i>	El nom del Servei i la seva categoria no han canviat.
<i>Procés normal:</i>	L'Administrador crida al CdU <u>ConsultarServei</u> i cerca el Servei que vol modificar. Una vegada que l'ha trobat pot modificar alguns camps però el nom i la categoria són fixes per mantenir la coherència. El sistema demana confirmació i una vegada l'Administrador ho accepta, es guarden els canvis.

<i>Alternatives de procés:</i>	-
--------------------------------	---

Subsistema Sol·licitud

Si a l'aplicació GBS un dels inputs humans són els voluntaris, l'altra cara de la moneda són els sol·licitants, aquells que requereixen d'alguna ajuda.

Aquest subsistema contindrà els CdU que emmagatzemen no només una nova sol·licitud d'un servei, si no també la creació d'aquest sol·licitant i el seguiment i valoració una vegada s'hagi realitzat la tasca.

Com s'ha comentat en l'apartat de l'Abast del Projecte, la implementació d'aquest subsistema queda per següents etapes, però s'ha aprofitat aquests mesos per fer l'anàlisi, especificació i incloure aquestes funcions en l'estructura de la base de dades i l'arquitectura J2EE.

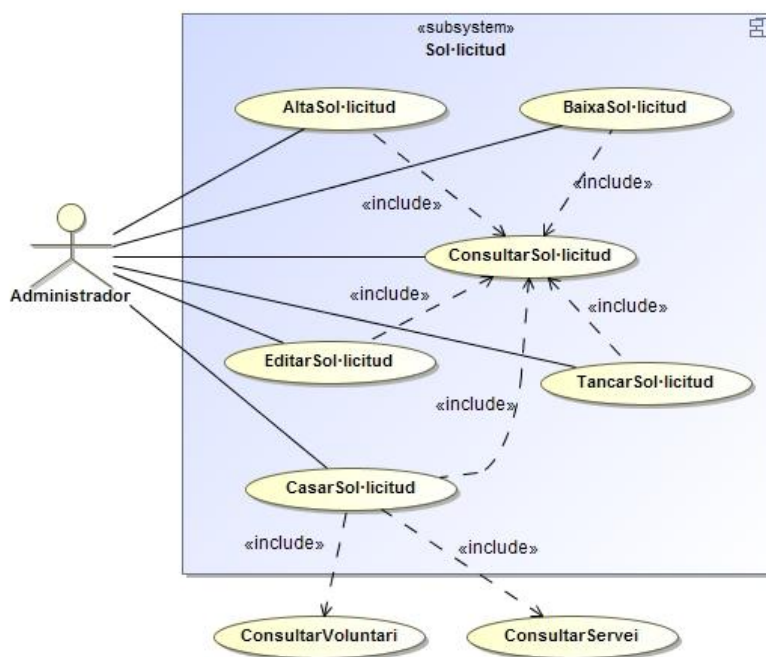


FIGURA 7: SUBSISTEMA SOL·LICITUD

Subsistema Informes

Aquest subsistema beu directament de l'aplicació que actualment utilitzen al Banc Solidari.

Consistirà en un seguit d'informes preestablerts i consensuats per l'Equip Directiu i que donaran informació en temps real segons les dades existents a la bases de dades de l'aplicació.

En un primer moment s'havia pensat d'implementar un sistema de creació "a la carta" d'informes, on l'actor només hagués d'anar introduint criteris, de quines taules volia consultar i anar guardant els informes si estava satisfet amb el resultat. Però un anàlisi inicial del possible control d'errors que s'hauria de estar realitzant per cada input de l'usuari va descartar la idea, per afavorir un posada en marxa més propera.

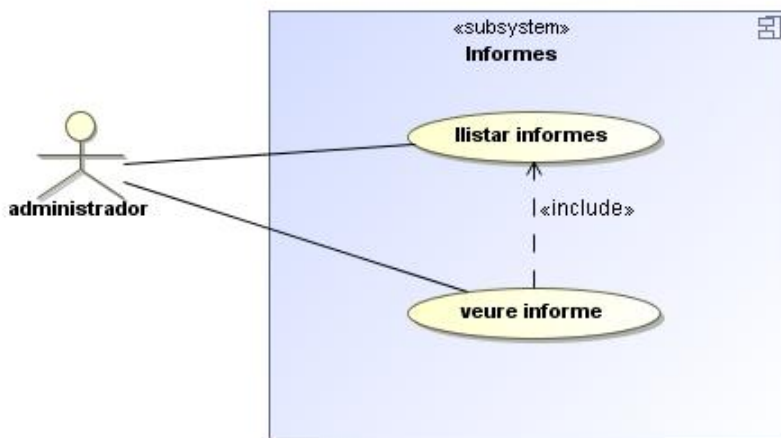


FIGURA 8: SUBSISTEMA INFORMES

Cas d'ús número 22: "LlistarInformes"	
<i>Resum de la funcionalitat:</i>	Permet veure els Informes establerts en el sistema.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	VeureInforme
<i>Precondició:</i>	-
<i>Postcondició:</i>	-
<i>Procés normal:</i>	El sistema mostra un llistat amb tots els Informes actius en el sistema. L'Administrador pot veure les dades bàsiques en la taula de dades, accedir a un detall més extens o cridar al CdU <u>VeureInforme</u> .
<i>Alternatives de procés:</i>	-

Cas d'ús número 23: "VeureInforme"	
<i>Resum de la funcionalitat:</i>	Es mostren les dades que retorna l'Informe en qüestió.
<i>Actors:</i>	Administrador
<i>Casos d'ús relacionats:</i>	-
<i>Precondició:</i>	-
<i>Postcondició:</i>	-
<i>Procés normal:</i>	El sistema mostra les dades seguint el criteri establert en l'Informe.
<i>Alternatives de procés:</i>	L'Administrador podrà imprimir aquest Informe en PDF.

ARQUITECTURA DE L'APLICACIÓ

J2EE

Per tal de facilitar la lectura d'aquesta documentació a una major quantitat de desenvolupadors, encara es fa referència a l'arquitectura com a J2EE tot i que ja fa temps que el seu nom va canviar i en el moment de la redacció d'aquest document, s'anomena **Java EE** en la seva versió 7.

El Projecte del GBS es realitzarà utilitzant una arquitectura Java EE per la seva reusabilitat, robustesa i escalabilitat.

L'arquitectura Java EE és caracteritzada per una separació de l'aplicació en parts anomenades capes; aquestes capes reuneixen unes funcionalitats comuns i són diferents de les altres capes.

A cadascuna de les capes s'incorporen una o més tecnologies o frameworks, permetent canvis en una capa sense haver de modificar les altres facilitant el treball en equip i el manteniment de l'aplicació.

Les capes més importants que s'implementen són quatre:

1. **Capa Client:** consisteixen en aplicacions que resideixen fora del servidor i que li envien peticions. Pot ser un navegador, una aplicació web o un altre servidor.
2. **Capa Web:** és la capa que s'encarrega de gestionar les transaccions entre la capa Client i la capa de Negoci. Controla la informació que es presenta al client, els inputs d'aquest, manté l'estat de la sessió, etc.
3. **Capa Lògica de Negoci:** comprèn tot el codi que descriu la funcionalitat de l'aplicació, ja sigui una web de comerç electrònic o la gestió d'un servei de taxis.
4. **Capa de Dades:** manté la informació del sistema en sistemes de bases de dades i per seguretat, normalment es situa en una màquina diferent a la del servidor d'aplicacions.

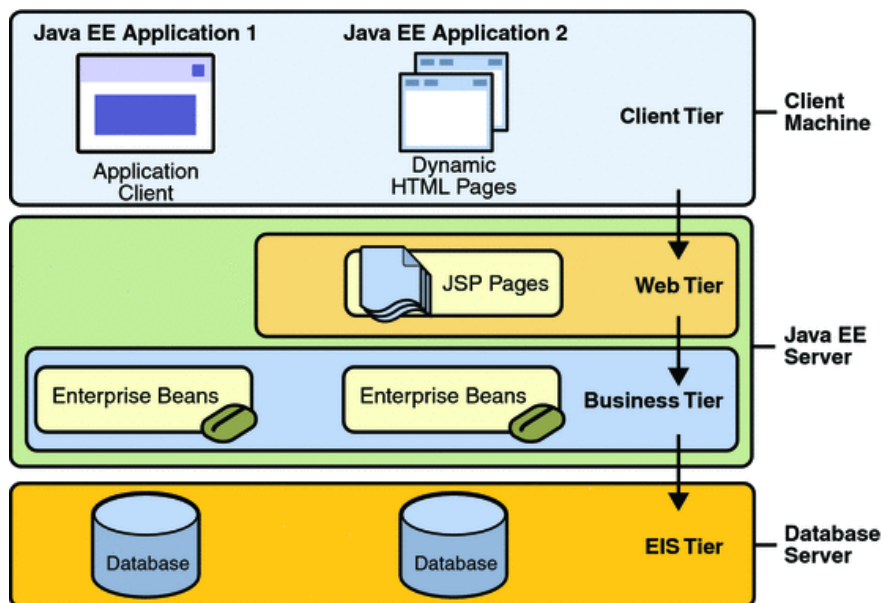


FIGURA 9: CAPES DE L'ARQUITECTURA J2EE

Font: <http://docs.oracle.com/javaee/5/tutorial/doc/bnaay.html>

Model MVC

El patró Model-Vista-Controlador de l'arquitectura del programari separa la comunicació entre l'usuari i l'aplicació, i el tractament de les dades, en tres representacions diferents.

1. **Model:** són les dades que representen l'estat de la realitat; l'objecte que representa i treballa directament amb les dades del programa.
2. **Vista:** és l'objecte que maneja la presentació visual de les dades gestionades pel Model.
3. **Controlador:** és l'objecte que proporciona significat a les ordres de l'usuari, actuant sobre les dades representades pel model. Entra en acció quan es realitza una operació, ja sigui un canvi d'informació en el Model o una interacció sobre la Vista.

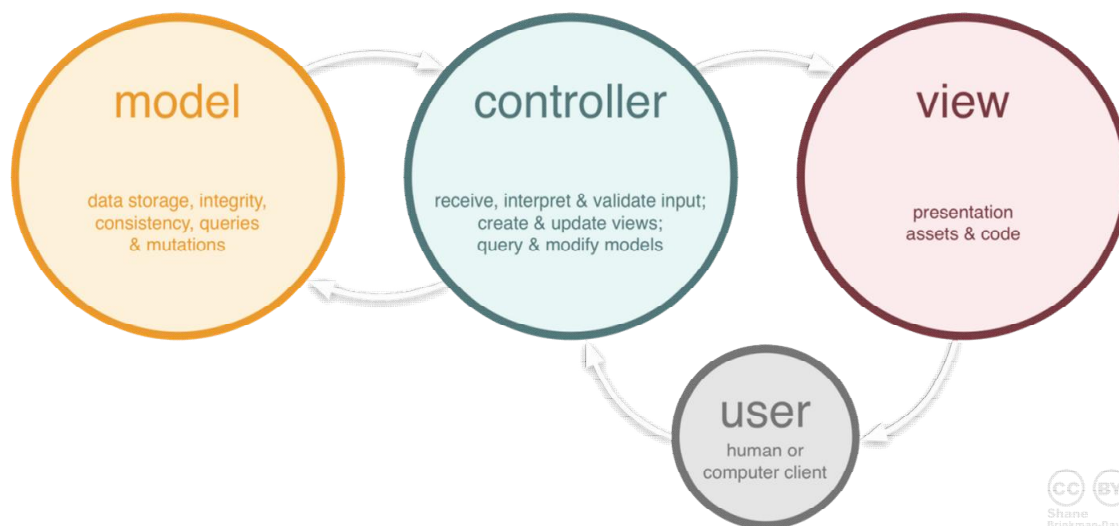


FIGURA 10: MODEL MVC

Font: <http://yanjin.info/tech/2013/06/24/mvc-stopwatch.html>

FRAMEWORKS

A continuació es descriuen els principals frameworks a utilitzar en l'aplicació GBS.

Struts 2

Struts 2 és un framework dirigit a la capa del Controlador i la seva principal característica és que funciona mitjançant Actions. Struts 2 s'encarrega de les peticions que l'usuari realitza sobre la Vista i pot aplicar (amb Interceptors) validacions de camps, població de dades o conversions de tipus.

Una vegada les peticions hagin estat processades al costat del servidor, es retornaran uns Results que modificaran la capa Vista.

La decisió d'escollir entre Struts 2 i JSF és una de les més recurrents en el disseny d'aplicacions web en l'actualitat. La versatilitat de les dues tecnologies permet decantar-se per una amb la confiança de satisfer gairebé qualsevol demanda.

Hi va haver un suggeriment inicial per part del tutor de provar d'incloure les dues tecnologies alhora (Struts2 i JSF).

Es va realitzar la investigació, existeix un plugin anomenat Struts2 JSF Plugin (<http://struts.apache.org/release/2.1.x/docs/jsf-plugin.html>) que ha de permetre la coexistència dels dos frameworks. Així, l'Struts2 s'encarregaria només de la part del

Controlador i faria sobretot ús dels *interceptors* i els *actions*, mentre que el JSF faria tota la feina de la capa Vista, la part més propera al client.

Després de diverses proves i com de moment no s'ha trobat una documentació suficient sobre la integració d'aquestes dues tecnologies, s'ha decidit utilitzar únicament Struts2, que permet gestionar tant les peticions del client com la posterior presentació de les respostes del servidor.

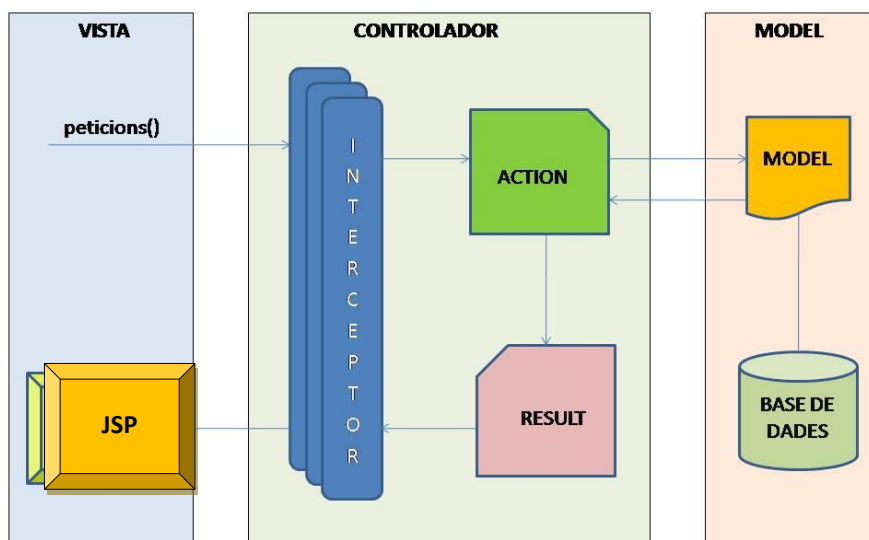


FIGURA 11: CICLE DE REQUEST/RESULT DEL STRUTS2

Spring

L'Spring Framework s'encarregarà principalment de mantenir l'encaix de la resta de frameworks en l'arquitectura de l'aplicació. Situat de manera transversal per les diferents capes, permet el conegut *dependency-injection*, que redueix la necessitat de crear objectes d'aquelles classes de les que volem fer algun ús. La utilització d'interfícies i de constructors amb arguments es converteixen en el cavall de batalla més potent.

Spring Framework dóna suport al model MVC així com a la gran majoria de frameworks utilitzats en el desenvolupament de programari.

Hibernate

Encara que el projecte Hibernate actualment abasta més funcionalitats, hom fa la relació immediata amb la persistència de les dades. Establint un mapeig entre les classes (o

POJOS) i les diferents taules de la base de dades podem assegurar que l'execució de l'aplicació mantindrà les dades en un estat consistent.

Integrat en l'Eclipse el seu funcionament és molt senzill:

- Es crea una connexió amb una base de dades ja existent.
- Es rep la informació de les taules que conformen el model.
- S'indica la transformació de les dades en el format Java que es necessita.
- El propi Hibernate s'encarrega de crear els POJO corresponents.

Disposa d'un llenguatge propi HQL molt semblant al SQL per realitzar les comunicacions amb la base de dades.

Altres eines

Encara que ja s'han descrit els 4 pilars bàsics a utilitzar en l'arquitectura, es pretén fer ús d'altres eines que complementen els frameworks anteriors. Donen suport a les capes de Vista, a la organització de les dependències o el logging de les activitats.

- Log4J
- Maven
- Spring DAO
- Jetty (Servidor d'aplicacions lleuger per realitzar les proves inicials del CdU Login).

A continuació es mostra el model MVC amb els frameworks a utilitzar en el projecte GBS.

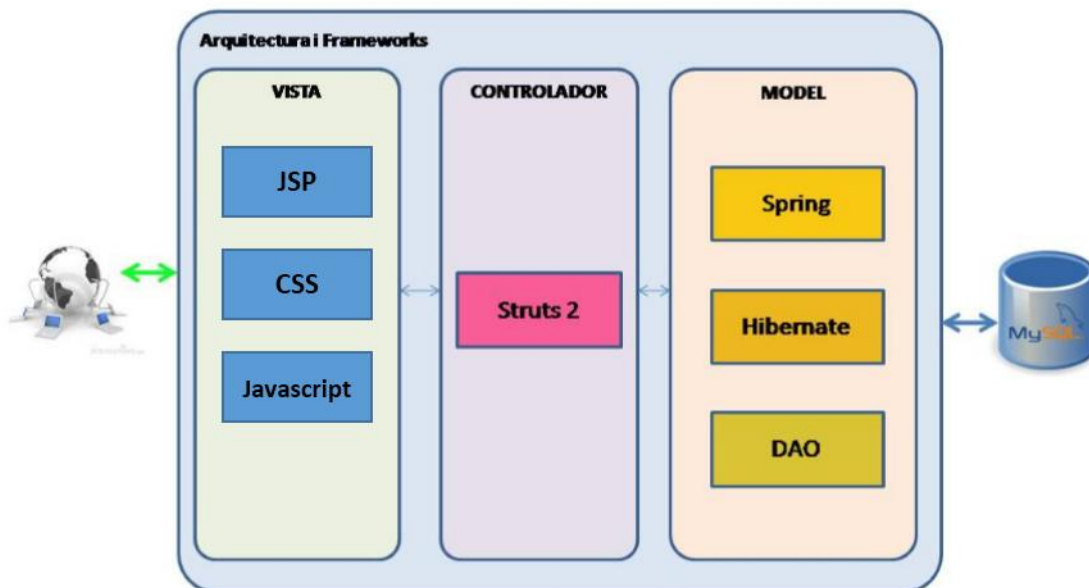


FIGURA 12: TECNOLOGIES I FRAMEWORKS DE L'APLICACIÓ

ENTORN DE DESENVOLUPAMENT I PROGRAMARI

Eclipse

Eclipse és un entorn de desenvolupament (IDE) dissenyat inicialment per IBM però mantinguda a continuació per una comunitat lliure d'usuaris.

Té la característica de ésser escrita pràcticament tota en Java i disposa de diversos components que permeten ampliar les possibilitats de l'usuari, com el Workbench o el SWT.

Les diferents versions han anat agafant noms de cossos estel·lars, satèl·lits, etc. Pel desenvolupament d'aquest projecte s'utilitzarà l'Eclipse Juno.

MySQL

MySQL és un sistema gestor de bases de dades (SGBD) disponible en versió gratuïta i amb una gran penetració en el món actual, tant a nivell particular com en l'empresarial. Això és degut a la seva robustesa, una gran velocitat de processament i senzillesa en el manteniment de les bases de dades.

Després de molts anys amb el suport dels desenvolupadors i la compra per part d'Oracle, MySQL incorpora un seguit d'eines que facilita el disseny, creació i manteniment de la BD.

El MySQL Workbench és una de les eines més útils ja que permet realitzar el modelatge de les diferents classes, realitzar el traspàs en taules i tot de manera simple i neta.

JBoss

Per poder executar el codi es necessita un servidor d'aplicacions i l'escollit ha sigut JBoss. Encara que existeix tot un entorn instalable, el JBoss Developer Studio, que permet un gran control sobre els processos del servidor, simulacions, execucions en dispositius mòbils, etc. existeix una versió integrada sobre l'IDE Eclipse que facilita el desenvolupament del projecte.

JBoss s'encarregarà d'interpretar el codi Java dels diferents POJOS que conformen l'aplicació, controlarà la prioritat de les execucions i retornarà al client la resposta adequada.

Durant la redacció d'aquest document aquesta aplicació ha canviat el seu nom a Wildfly (<http://www.wildfly.org/>).

Model Relacional

El Model Relacional es la transformació des de el diagrama de classes a la base de dades que s'hagi seleccionat per la nostra aplicació. En aquest procés, que comporta diversos passos i s'apliquen diverses tècniques com la normalització, el diagrama de classes sofreix certs canvis com la modificació de noms, la fusió de taules redundants o l'addició de taules auxiliars que estableixen les relacions entre les classes.

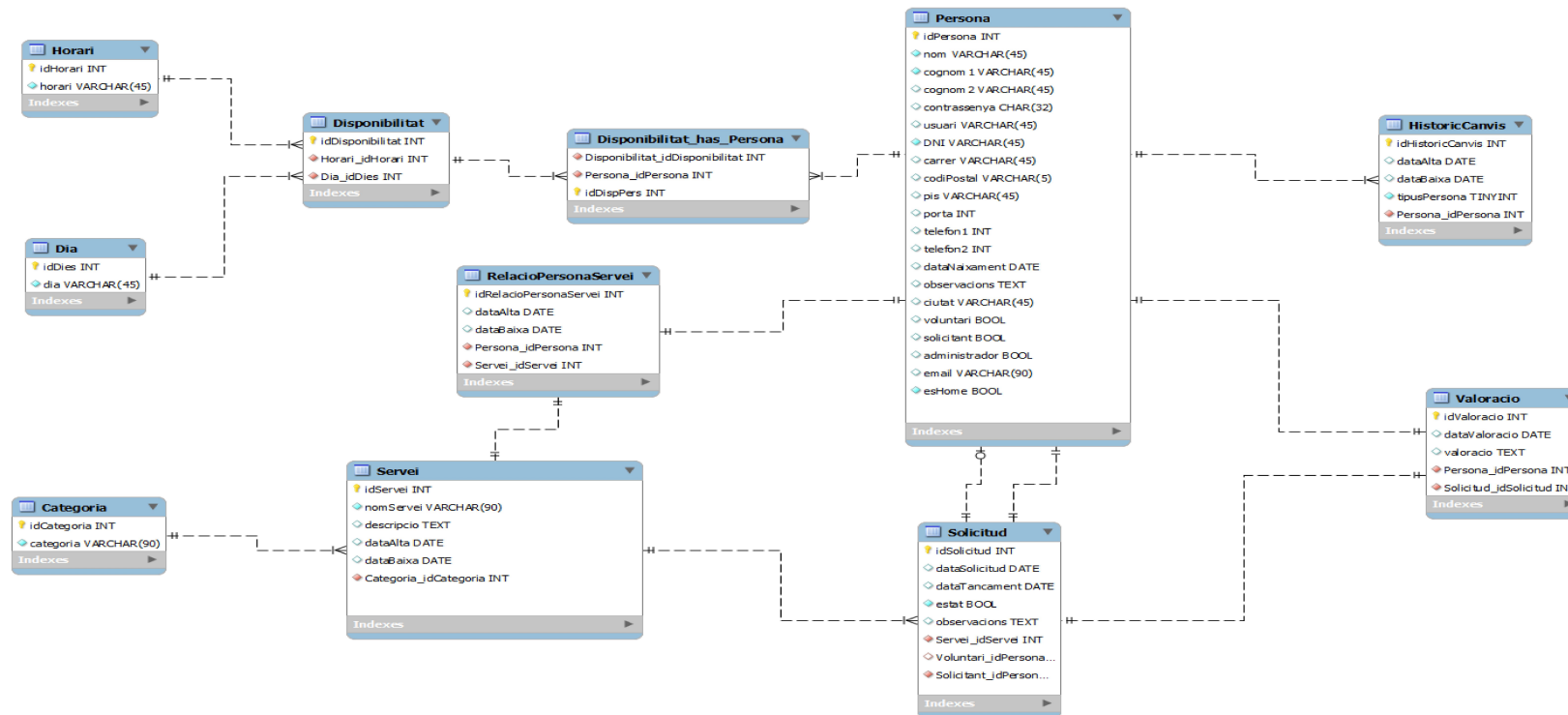


FIGURA 14: MODEL RELACIONAL

Pantalles de l'usuari

A continuació es mostren diverses pantalles de l'actual estat del projecte. Son casos d'ús que estan totalment operatius, se'ls hi ha aplicat el CSS i permeten realitzar les operacions CRUD estàndard.



FIGURA 15: LOGIN

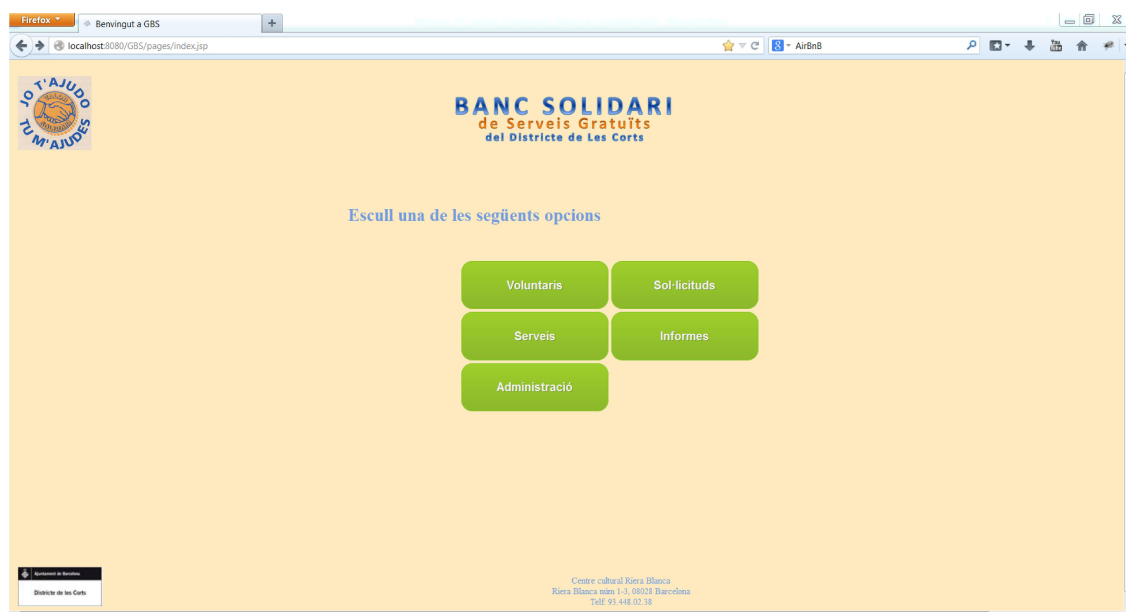


FIGURA 16: MENÚ PRINCIPAL



FIGURA 17: CONSULTA DE SERVEIS

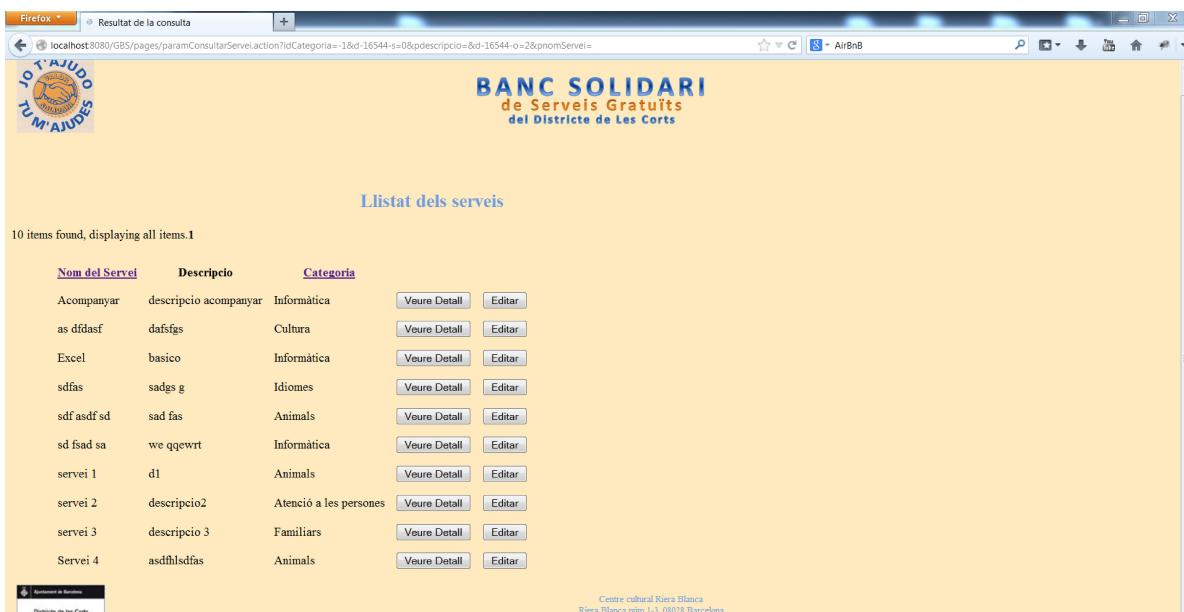


FIGURA 18: EDICIÓ DE SERVEI

Escalabilitat i futures millores

Durant la fase d'anàlisi i especificació s'ha realitzat un esforç per preparar la aplicació per que sigui escalable en nombre de casos d'ús així com en els atributs particulars de cada classe definida en el domini.

Dintre de les futures millores incloure en primer lloc els apartats restants per posar en marxa l'aplicatiu, que són:

- Finalitzar el subsistema de sol·licituds.
- Assegurar la navegabilitat entre totes les vistes de l'aplicació.
- Unificar l'estil CSS en totes les pàgines.
- Realitzar les validacions pertinents en els inputs dels usuaris, en equilibri amb la usabilitat que demanin els administradors, a la fi serà qui utilitzaran l'aplicació.

Per continuar ampliant coneixements, oferir noves funcionalitats i donar valor afegit a l'aplicació es proposen les següents millores.

- Enviament de correus electrònics de manera automàtica en esdeveniments programats (aniversaris, alarmes, etc.)
- Creació de diferents rols amb nivells diferents de permisos.
- Exportació del informes en altres formats (els canvis a realitzar seran mínims per la elecció de la tecnologia)
- Incloure un sistema de pujada de documents, investigant els requisits de la base de dades, els límits físics, tipus d'arxius, etc.

Conclusions

Objectius Assolits

En el moment d'iniciar aquest projecte hi havia l'objectiu clar d'acabar implantant una aplicació molt concreta en un entorn real. El nombre d'hores que s'havien de destinar al TFC ja feien preveure que no totes les funcions estarien disponibles, sent només una persona, havent de realitzar la investigació de les tecnologies.

Tot i això, encara que el projecte ha quedat en un estat de prototipus, és ja la base per l'aplicació final a implantar.

S'han aconseguit els coneixements necessaris no només per realitzar el GBS, sinó gairebé qualsevol aplicació web amb una arquitectura J2EE.

També, uns dels objectius assolits més importants és el d'haver aconseguit coordinar diverses tecnologies de fabricants diferents en un únic entorn i formar una base sòlida pel desenvolupament de software.

Conclusions Personals

Aquí acaben uns mesos d'esforç amb moltes hores d'investigació, aprenentatge, a vegades frustració i també moments de curiositat i motivació. Ha sigut la oportunitat d'aprendre com enfrontar-se a un projecte ja de certa magnitud i durant un temps considerable.

Hi ha hagut errors durant aquests mesos, com dedicar setmanes a la investigació purament teòrica, quan es millor combinar-la amb les primeres proves de la implementació. Crec que l'aprenentatge és molt més ràpid: pots llegir-te 100 pàgines d'un manual i en la primera funció adonar-te que no necessites res d'allò o que la nova versió ha canviat l'especificació.

Treballar sol no és fàcil i indubtablement Internet és l'eina de la qual se'n pot treure més profit pel desenvolupament del software. Les documentacions oficials de les tecnologies et donaran la base per entendre què fa i perquè ho fa, però serà l'experiència d'altres usuaris la que t'ajudarà en els possibles obstacles que trobaràs en el camí; i sobretot si estàs integrant tecnologies de diferents fabricants, opensource, etc.

Trobar un mètode ordenat de treballar és fonamental i en aquest projecte no s'aconsegueix fins gairebé al final, quan totes les peces encaixen i llavors sí, realment, s'està fent una traducció gairebé directa de l'especificació al codi. Abans d'això, és un procés d'aprenentatge, de prova i error, i de intentar mantenir l'objectiu final en el punt de mira, encara que costi.

Aquest projecte suposa a més moltes coses a nivell personal. És el final d'una carrera que per diverses raons ha durat massa, aquells que em coneixen o s'han trobat en una situació semblant ja saben de que parlo.

He de dir que estic satisfet pels esforços invertits i les assignatures que m'han donat coneixement però sobretot la capacitat d'afrontar els problemes amb una mirada crítica i un pensament analític. Crec que he après a aprendre.

Bibliografia

Beneito Montagut, Roser (2007) "*Presentació de documents i elaboració de presentacions.*". UOC

Campderrich Falgueras, Benet (11/08/2012) "*La recollida i documentació de requisits.*". Universitat Rovira i Virgili

Campderrich Falgueras, Benet "*UML (II): el model dinàmic i d'implementació.*". Universitat Oberta de Catalunya

González Morro, Samuel (16/03/2011) "*TFC-J2EE: Web Training Diary. Anàlisis*". UOC

Keogh, Jim (2003) "*J2EE. Manual de Referència*" McGraw Hill

La resolució de problemes no hagués sigut possible sense l'ajuda dels milers d'usuaris desinteressats que col·laboren en els fòrums.

<http://www.tutorialspoint.com>

<http://www.mkyong.com/>

<http://stackoverflow.com/>

<http://www.coderanch.com>

<http://www.javatutoriales.com>

Annex 1: Sentències DDL

```
CREATE DATABASE IF NOT EXISTS `gbs_schema` USE `gbs_schema`;
```

```
CREATE TABLE categoria (  
  idCategoria int(11) NOT NULL,  
  nomCategoria varchar(90) NOT NULL,  
  PRIMARY KEY (idCategoria),  
  UNIQUE KEY categoria_UNIQUE (nomCategoria));
```

```
DROP TABLE IF EXISTS dia;
```

```
CREATE TABLE dia (  
  idDies int(11) NOT NULL,  
  dia varchar(45) NOT NULL,  
  PRIMARY KEY (idDies),  
  UNIQUE KEY dies_UNIQUE (dia));
```

```
DROP TABLE IF EXISTS disponibilitat;
```

```
CREATE TABLE disponibilitat (  
  idDisponibilitat int(11) NOT NULL,  
  Horari_idHorari int(11) NOT NULL,  
  Dia_idDies int(11) NOT NULL,  
  PRIMARY KEY (idDisponibilitat),  
  KEY fk_Disponibilitat_Horari1_idx (Horari_idHorari),  
  KEY fk_Disponibilitat_Dia1_idx (Dia_idDies),  
  CONSTRAINT fk_Disponibilitat_Dia1 FOREIGN KEY (Dia_idDies) REFERENCES dia (idDies) ON  
  DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT fk_Disponibilitat_Horari1 FOREIGN KEY (Horari_idHorari) REFERENCES horari  
  (idHorari) ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

```
DROP TABLE IF EXISTS disponibilitat_has_persona;
```

```
CREATE TABLE disponibilitat_has_persona (  
  Disponibilitat_idDisponibilitat int(11) NOT NULL,
```

```
Persona_idPersona int(11) NOT NULL,  
idDispPers int(11) NOT NULL,  
PRIMARY KEY (idDispPers),  
KEY fk_Disponibilitat_has_Persona_Persona1_idx (Persona_idPersona),  
KEY fk_Disponibilitat_has_Persona_Disponibilitat1_idx (Disponibilitat_idDisponibilitat),  
CONSTRAINT fk_Disponibilitat_has_persona_Dispinibilitat1 FOREIGN KEY  
(Disponibilitat_idDisponibilitat) REFERENCES disponibilitat (idDisponibilitat) ON DELETE  
CASCADE ON UPDATE CASCADE,  
CONSTRAINT fk_Disponibilitat_has_persona_Persona1 FOREIGN KEY (Persona_idPersona)  
REFERENCES persona (idPersona) ON DELETE CASCADE ON UPDATE CASCADE);
```

```
DROP TABLE IF EXISTS historiccanvis;
```

```
CREATE TABLE historiccanvis (
```

```
idHistoricCanvis int(11) NOT NULL,  
dataAlta date DEFAULT NULL,  
dataBaixa date DEFAULT '2099-12-31',  
tipusPersona tinyint(4) NOT NULL,  
Persona_idPersona int(11) NOT NULL,  
PRIMARY KEY (idHistoricCanvis),  
KEY fk_HistoricCanvis_Persona1_idx (Persona_idPersona),  
CONSTRAINT fk_HistoricCanvis_Persona1 FOREIGN KEY (Persona_idPersona) REFERENCES  
persona (idPersona) ON DELETE NO ACTION ON UPDATE NO ACTION);
```

```
DROP TABLE IF EXISTS horari;
```

```
CREATE TABLE horari (
```

```
idHorari int(11) NOT NULL,  
horari varchar(45) NOT NULL,  
PRIMARY KEY (idHorari),  
UNIQUE KEY horaris_UNIQUE (horari));
```

```
DROP TABLE IF EXISTS persona;
```

```
CREATE TABLE persona (
```

```
idPersona int(11) NOT NULL,  
nom varchar(45) NOT NULL,
```

```
cognom1 varchar(45) NOT NULL,  
cognom2 varchar(45) DEFAULT NULL,  
contrassenya char(32) DEFAULT NULL,  
usuari varchar(45) DEFAULT NULL,  
DNI varchar(45) DEFAULT NULL,  
carrer varchar(45) DEFAULT NULL,  
codiPostal varchar(5) DEFAULT NULL,  
pis varchar(45) DEFAULT NULL,  
porta int(11) DEFAULT NULL,  
telefon1 int(11) DEFAULT NULL,  
telefon2 int(11) DEFAULT NULL,  
dataNaixament date DEFAULT NULL,  
observacions text,  
ciutat varchar(45) DEFAULT NULL,  
voluntari tinyint(1) unsigned zerofill NOT NULL DEFAULT '0',  
solicitant tinyint(1) unsigned zerofill NOT NULL DEFAULT '0',  
administrador tinyint(1) unsigned zerofill NOT NULL DEFAULT '0',  
email varchar(90) DEFAULT NULL,  
esHome tinyint(1) DEFAULT '1',  
PRIMARY KEY (idPersona),  
UNIQUE KEY DNI_UNIQUE (DNI),  
UNIQUE KEY usuari_UNIQUE (usuari));
```

```
DROP TABLE IF EXISTS relaciopersonaservei;
```

```
CREATE TABLE relaciopersonaservei (  
idRelacioPersonaServei int(11) NOT NULL,  
dataAlta date DEFAULT NULL,  
dataBaixa date DEFAULT '2099-12-31',  
Persona_idPersona int(11) NOT NULL,  
Servei_idServei int(11) NOT NULL,  
PRIMARY KEY (idRelacioPersonaServei),  
KEY fk_RelacioPersonaServei_Persona1_idx (Persona_idPersona),  
KEY fk_RelacioPersonaServei_Servei1_idx (Servei_idServei),
```

```
CONSTRAINT fk_RelacioPersonaServei_Persona1 FOREIGN KEY (Persona_idPersona)
REFERENCES persona (idPersona) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT fk_RelacioPersonaServei_Servei1 FOREIGN KEY (Servei_idServei)
REFERENCES servei (idServei) ON DELETE CASCADE ON UPDATE CASCADE);
```

```
DROP TABLE IF EXISTS servei;
```

```
CREATE TABLE servei (
```

```
idServei int(11) NOT NULL,
```

```
nomServei varchar(90) NOT NULL,
```

```
descripcio text,
```

```
dataAlta date DEFAULT NULL,
```

```
dataBaixa date DEFAULT '2099-12-31',
```

```
Categoria_idCategoria int(11) NOT NULL,
```

```
PRIMARY KEY (idServei),
```

```
UNIQUE KEY nomServei_UNIQUE (nomServei),
```

```
KEY fk_Servei_Categoria_idx (Categoria_idCategoria),
```

```
CONSTRAINT fk_Servei_Categoria FOREIGN KEY (Categoria_idCategoria) REFERENCES
categoria (idCategoria) ON DELETE NO ACTION ON UPDATE NO ACTION);
```

```
DROP TABLE IF EXISTS solitud;
```

```
CREATE TABLE solitud (
```

```
idSolitud int(11) NOT NULL,
```

```
dataSolitud date DEFAULT NULL,
```

```
dataTancament date DEFAULT '2099-12-31',
```

```
estat tinyint(1) NOT NULL DEFAULT '1',
```

```
observacions text,
```

```
Servei_idServei int(11) NOT NULL,
```

```
Voluntari_idPersona int(11) DEFAULT NULL,
```

```
Solicitant_idPersona1 int(11) NOT NULL,
```

```
PRIMARY KEY (idSolitud),
```

```
KEY fk_Solitud_Servei1_idx (Servei_idServei),
```

```
KEY fk_Solitud_Persona1_idx (Voluntari_idPersona),
```

```
KEY fk_Solitud_Persona2_idx (Solicitant_idPersona1),
```

```
CONSTRAINT fk_Solicitud_Persona1 FOREIGN KEY (Voluntari_idPersona) REFERENCES
persona (idPersona) ON DELETE NO ACTION ON UPDATE NO ACTION,
```

```
CONSTRAINT fk_Solicitud_Persona2 FOREIGN KEY (Solicitant_idPersona1) REFERENCES
persona (idPersona) ON DELETE NO ACTION ON UPDATE NO ACTION,
```

```
CONSTRAINT fk_Solicitud_Servei1 FOREIGN KEY (Servei_idServei) REFERENCES servei
(idServei) ON DELETE NO ACTION ON UPDATE NO ACTION);
```

```
DROP TABLE IF EXISTS valoracio;
```

```
CREATE TABLE valoracio (
```

```
idValoracio int(11) NOT NULL,
```

```
dataValoracio date DEFAULT NULL,
```

```
valoracio text,
```

```
Persona_idPersona int(11) NOT NULL,
```

```
Solicitud_idSolicitud int(11) NOT NULL,
```

```
PRIMARY KEY (idValoracio),
```

```
KEY fk_Valoracio_Persona1_idx (Persona_idPersona),
```

```
KEY fk_Valoracio_Solicitud1_idx (Solicitud_idSolicitud),
```

```
CONSTRAINT fk_Valoracio_Persona1 FOREIGN KEY (Persona_idPersona) REFERENCES
persona (idPersona) ON DELETE NO ACTION ON UPDATE NO ACTION,
```

```
CONSTRAINT fk_Valoracio_Solicitud1 FOREIGN KEY (Solicitud_idSolicitud) REFERENCES
solicitud (idSolicitud) ON DELETE NO ACTION ON UPDATE NO ACTION);
```

Annex 2: Sentències DML

Aquestes sentències són necessàries per tal de que l'aplicació tingui uns valors inicials. Pertanyen a les taules "auxiliars" com els dies de la setmana, les disponibilitats horàries i les categories.

```
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (1,'Animals');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (2,'AtenciÃ³ a les persones');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (3,'Cultura');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (4,'DecoraciÃ³ i petit bricolatge');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (5,'Familiars');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (6,'Idiomes');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (7,'InformÃtica');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (8,'Salut i estÃtica');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (9,'Tasques administratives');
INSERT INTO categoria (idCategoria, nomCategoria) VALUES (10,'Xerrades i assessorament');
```

```
INSERT INTO dia (idDies, dia) VALUES (4,'Dijous');
INSERT INTO dia (idDies, dia) VALUES (1,'Dilluns');
INSERT INTO dia (idDies, dia) VALUES (2,'Dimarts');
INSERT INTO dia (idDies, dia) VALUES (3,'Dimecres');
INSERT INTO dia (idDies, dia) VALUES (6,'Dissabte');
INSERT INTO dia (idDies, dia) VALUES (7,'Diumenge');
INSERT INTO dia (idDies, dia) VALUES (5,'Divendres');
```

```
INSERT INTO horari (idHorari, horari) VALUES (1,'MatÃ');
INSERT INTO horari (idHorari, horari) VALUES (3,'Migdia');
INSERT INTO horari (idHorari, horari) VALUES (4,'Nit');
INSERT INTO horari (idHorari, horari) VALUES (2,'Tarda');
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (15,39,1);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (16,39,2);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (9,43,3);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (11,43,4);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (4,44,5);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (11,44,6);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (18,44,7);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (24,44,8);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (8,45,9);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (21,45,10);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (2,46,11);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (16,46,12);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (21,47,13);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (21,48,14);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (1,49,15);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (15,49,16);
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (9,50,17);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (11,50,18);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (16,50,19);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (18,50,20);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (22,50,21);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (24,50,22);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (1,51,23);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (16,51,24);
```

```
INSERT INTO disponibilitat_has_persona (Disponibilitat_idDisponibilitat, Persona_idPersona, idDispPers) VALUES (21,52,25);
```


Annex 3: Manual d'instal·lació.

S'inclou una informació addicional referent al hardware i al software utilitzat pel desenvolupament del projecte. Hi han algunes explicacions per la instal·lació de l'entorn i la posada en marxa, encara que d'una manera superficial.

La màquina utilitzada per la realització d'aquest projecte constava de:

AMD Phenom II X6 1035T Processor 2.60 GHz

4 GB RAM

NVIDIA GeForce GT 320 1024 MB

S.O: Windows 7 Professional 64 bits

Pel desenvolupament del codi s'ha necessitat:

El **Java Development Kit d'Oracle**, la versió jdk1.7.0_45. Una vegada descarregat, s'inicialitzen les variables d'entorn per tal de que el sistema reconegui l'executable.

Descarregar l'**Eclipse Java EE IDE for Web Developers**, versió Juno Service Release 2

El **MySQL**, en la seva versió 5.6, incloent el MySQL WorkBench per la realització del model relacional.

El **conector MySQL-Java**, en la seva versió 5.1.25, per establir la comunicació entre l'Eclipse i el MySQL.

Des de el propi Eclipse es fa ús de l'Eclipse Marketplace per descarregar les tecnologies més importants com son el JBoss, l'Hibernate i l'Spring. Són processos automàtics i creen llibreries que s'afegeixen en el path del nostre projecte.

Per la gestió dels .jars (codis comprimits que necessiten les aplicacions per funcionar), utilitzem l'eina **Maven**, que s'instal·la directament a l'Eclipse. Accedint a la pàgina <http://mvnrepository.com/> podem trobar totes les tecnologies que necessitem per la nostra aplicació i que afegirem en un arxiu anomenat pom.xml situat en l'arrel del projecte.

Per donar d'alta un servidor, en el nostre cas **JBoss 7.0.1**, anem a la pestanya de Server de l'Eclipse i amb el botó dret diem New Server. Aquí apareixen diferents tipus de servidors que Eclipse té emmagatzemats a la seva base de dades (es poden descarregar més).

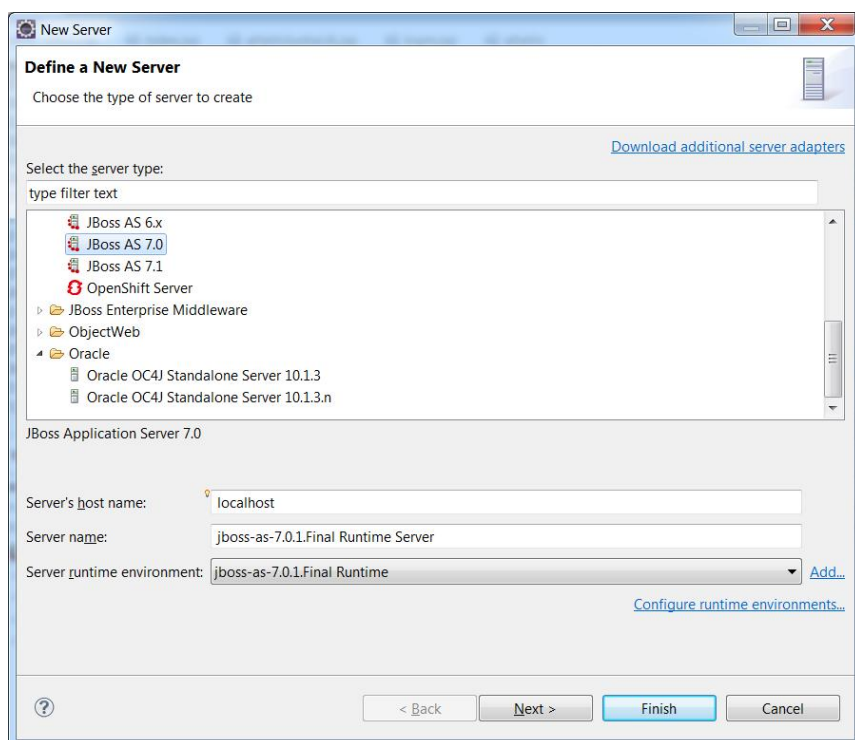


FIGURA 19: DEFINICIÓ D'UN NOU SERVIDOR

Una vegada seleccionat el desitjat, editarem les opcions de l'entorn d'execució per tal de que seleccioni el JDK que havíem descarregat prèviament.

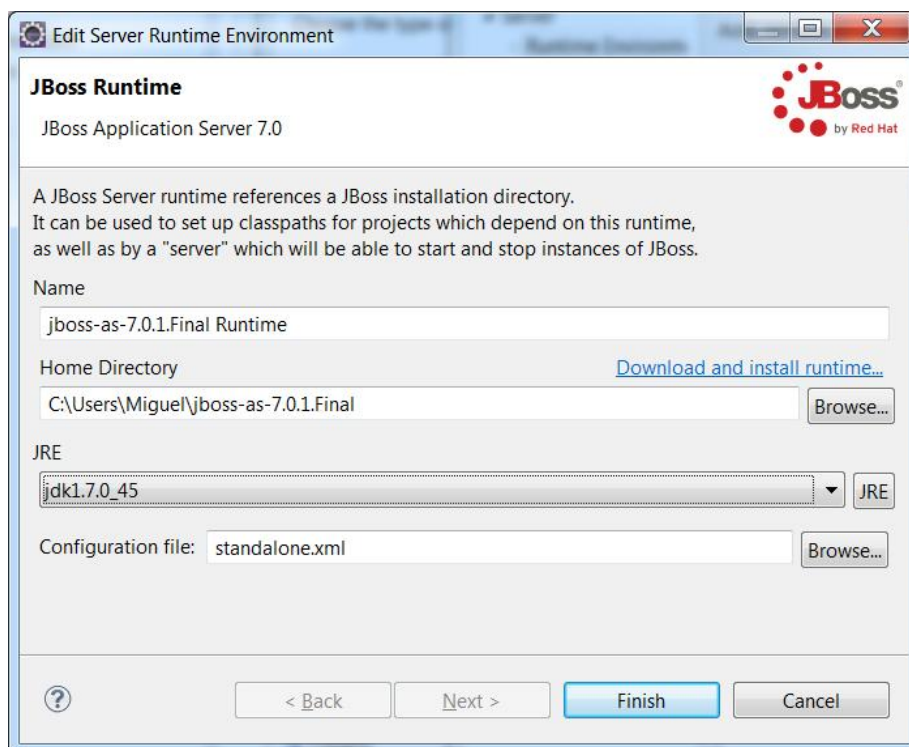


FIGURA 20: SELECCIÓ DEL RUNTIME ENVIRONMENT

Seguint els passos que l'aplicació ens demana i afegint el projecte que estem realitzant, serem capaços de tenir en l'Eclipse el nostre servidor d'aplicacions JBoss funcionant, i l'aplicació GBS també executant-se. Serà des de aquí que realitzarem les operacions d'aturada, debugueig, reiniciar, etc. Cada vegada que sigui necessari.

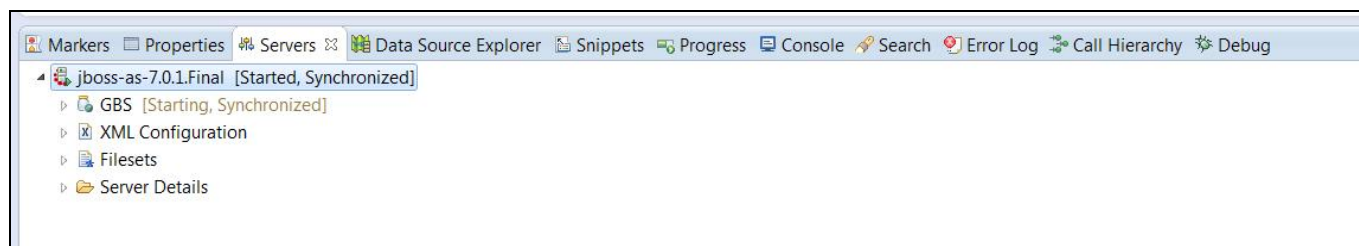


FIGURA 21: SERVIDOR JBOSS INSTAL·LAT I FUNCIONANT

Una eina molt interessant que podem obtenir a l'Eclipse és l'Hibernate Tools. Aquesta incorpora un seguit de funcions que ens permet establir de manera senzilla la connexió amb la base de dades, importar el model relacional i fins i tot crear les primeres classes del model i les de persistència.

Seguint els wizards d'aquesta eina crearem una consola de configuració per l'Hibernate on es poden definir els paràmetres de connexió amb el MySQL com el tipus de base de dades, el nom d'usuari, a quin esquema, el port, etc.

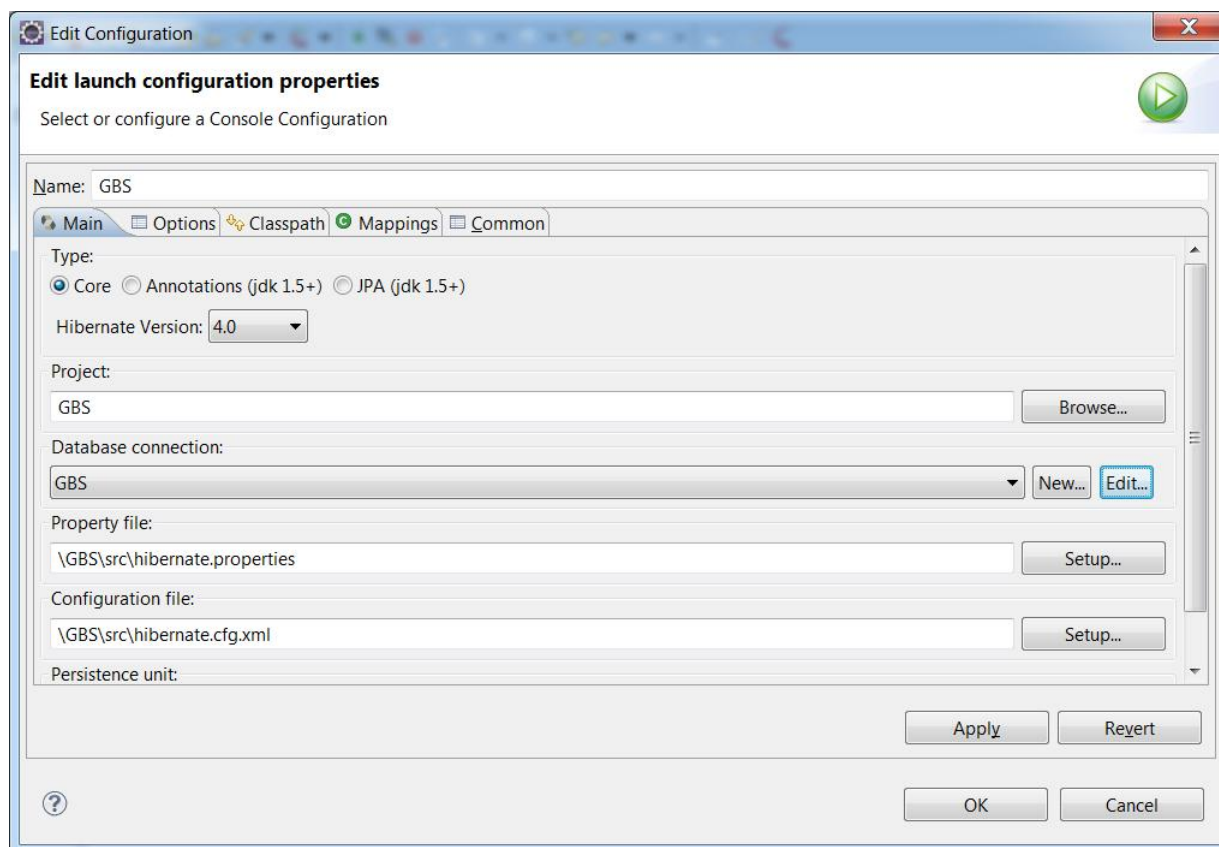


FIGURA 22: CONSOLA DE CONFIGURACIÓ DE L'HIBERNATE TOOLS

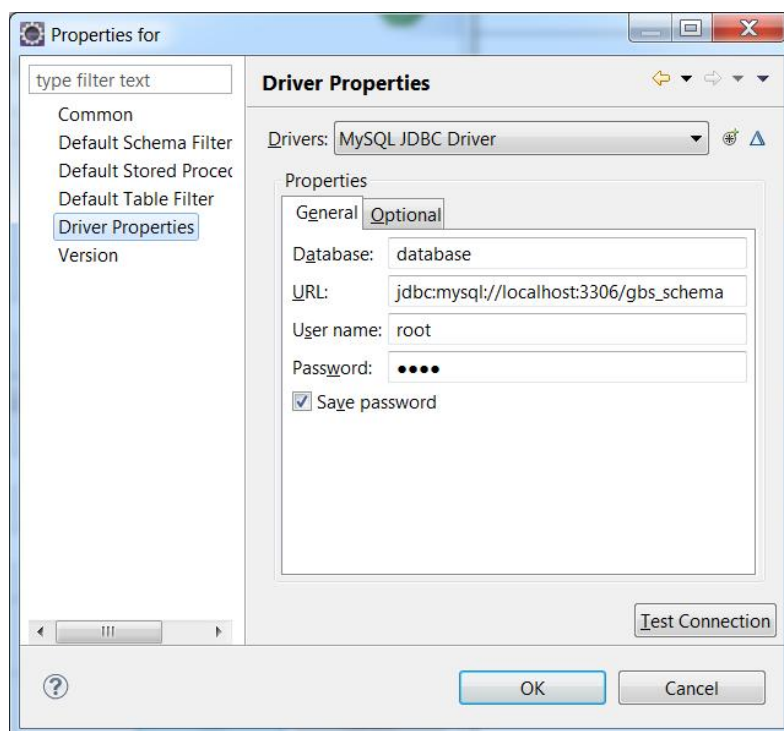


FIGURA 23: CONFIGURACIÓ DEL DRIVER DE CONNEXIÓ AMB LA BD