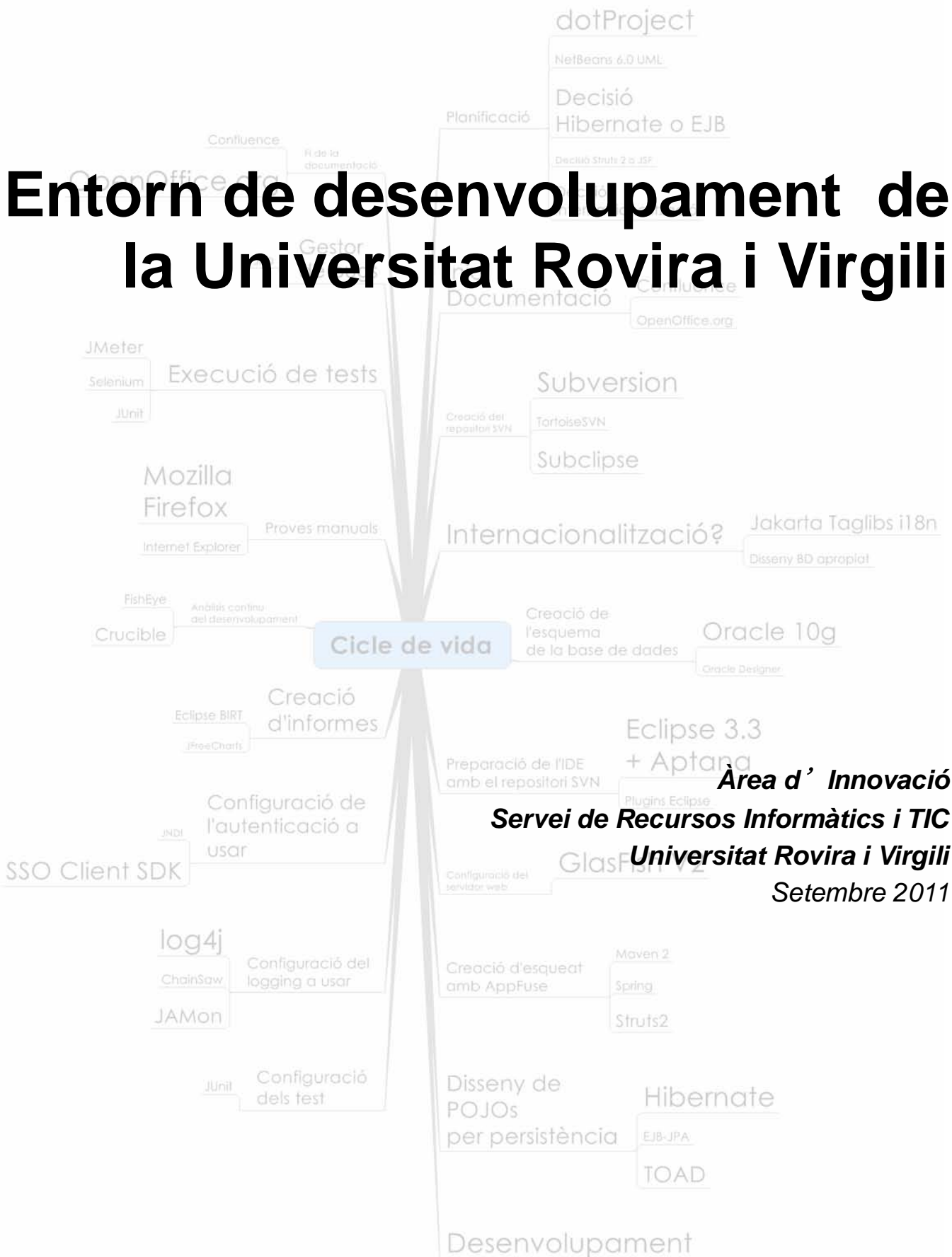


Entorn de desenvolupament de la Universitat Rovira i Virgili



Àrea d' Innovació
Servei de Recursos Informàtics i TIC
Universitat Rovira i Virgili
 Setembre 2011

Índex

Copyright © 2008 Tomàs Reverter Morelló / Universitat Rovira i Virgili

Es garanteix el permís per copiar, distribuir i modificar aquest document segons els termes de la llicència Creative Commons "Reconeixement-Compartir amb la mateixa llicència 2.5 Espanya" i en tot cas, amb l'atribució dels drets d'explotació de l'obra original a la URV. Es disposa d'una còpia de la llicència al capítol "Llicència".

ÍNDEX.....	1
ÍNDEX DE TAULES.....	9
ÍNDEX DE FIGURES.....	10
CONVENCIONS DE FORMAT DEL DOCUMENT.....	10
DESCÀRREC.....	11
1. OBJECTIUS DEL PROJECTE.....	1
La URV i les aplicacions informàtiques.....	2
El Servei de Recursos Informàtics i TIC.....	2
Objectius generals.....	3
Objectius formatius.....	3
2. ESPECIFICACIONS.....	4
Requisits tecnològics del SER.RI-TIC.....	5
Destinatari del projecte.....	5
Aproximació al problema.....	6
Planificació i gestió de projectes.....	6
Entorn Integrat de Desenvolupament (IDE).....	6
Generadors d'esquelets de codi.....	7
Control de versions o Software Configuration Management (SCM).....	7
Eines comunicatives.....	7
Registre d'aplicacions (logging).....	8
Internacionalització (i18n) i localització (l10n).....	8

Eines de building.....	8
Testing.....	9
Gestió de bugs.....	9
Escriure la documentació.....	9

3. DISSENY.....10

Mètode de desenvolupament del projecte.....	11
Recerca inicial d'informació.....	11
Instal·lació d'un cas real.....	12
Desenvolupament del projecte.....	12
Dedicació temporal.....	13
Eines de suport al desenvolupament del projecte.....	13
Arbre de disseny.....	13
Arquitectura centralitzada – Solaris.....	15
Base de dades.....	15
Oracle 10g.....	15
Control de versions (SCM).....	15
Subversion.....	17
FishEye.....	19
Crucible.....	20
Avaluació.....	20
Gestor de bugs i issues.....	21
Jira.....	21
Trac.....	22
Mindquarry.....	23
Avaluació.....	24
Gestor de projectes.....	25
dotProject.....	25
Servidor d'aplicacions.....	25
Sun Java System Application Server.....	25
Wiki.....	26
MediaWiki.....	26
Confluence.....	27
Avaluació.....	28
Desenvolupador – Windows i/o GNU/Linux.....	29
Client Subversion.....	29
Subclipse.....	29
TortoiseSVN.....	29
Avaluació.....	29

Creació i anàlisi de consultes SQL.....	30
TOAD for Oracle.....	30
TOra.....	30
Disseny de bases de dades.....	30
Oracle Designer.....	30
DBDesigner.....	31
Disseny UML.....	31
NetBeans UML Project.....	31
ArgoUML.....	32
Altres eines no lliures per Eclipse.....	32
Avaluació.....	32
IDE.....	33
Eclipse Europa 3.3.....	33
EasysEclipse.....	34
NetBeans 6.0.....	35
Aptana.....	36
Avaluació.....	36
Navegadors.....	38
Mozilla Firefox.....	38
Internet Explorer.....	39
Safari.....	39
Avaluació.....	39
Servidor d'aplicacions per desenvolupament.....	40
Sun Java System Application Server.....	40
GlassFish.....	41
Tomcat.....	41
Sun Java System Web Server.....	42
Avaluació.....	42
Utilitats.....	43
Tecnologies.....	45
Accés a dades - persistència.....	45
Hibernate.....	46
Enterprise JavaBean (EJB).....	47
Avaluació.....	48
Accés a LDAP.....	49
Java Naming and Directory Interface (JNDI).....	49
Eines de build.....	50
Ant.....	52
Maven 2.....	53
Avaluació.....	54
Eines de generació d'informes.....	55
JReport.....	55
Eclipse BIRT.....	56
JasperReports.....	58
OpenReports.....	58
Avaluació.....	59
Esquelet de codi.....	60
AppFuse 2.....	60
Framework de model.....	63
Spring Framework.....	63
Framework de presentació.....	65

Struts 2.....	65
Java Server Faces (JSF).....	67
Avaluació.....	68
Gràfics estadístics.....	69
JReport.....	69
JFreeChart.....	69
Avaluació.....	69
Internacionalització (i18n) i localització (l10n).....	70
Jakarta Taglibs i18n.....	71
Framework presentació.....	71
Avaluació.....	71
Logging.....	72
log4j.....	72
Chainsaw v2.....	73
JAMon.....	73
Avaluació.....	73
Single Sign-On (SSO).....	74
Sun Java System Access Manager 7 - Client SDK.....	74
Tests unitaris.....	75
JUnit.....	75
TestNG.....	76
Avaluació.....	77
Tests de càrrega.....	78
JMeter.....	78
JUnitPerf.....	78
The Grinder.....	79
Avaluació.....	79
Tests de funcionalitat del sistema i de compatibilitat del navegador.....	80
Selenium.....	80
Convencions dels noms dels paquets Java.....	81
Ús de formats estàndards lliures.....	82
Què és un format?.....	82
Formats lliures i formats propietaris.....	82
Format obert o lliure.....	82
Format propietari o privatiu.....	82
Quatre motius per no utilitzar formats propietaris.....	82
Quatre motius per adoptar formats oberts.....	82
Equivalències de formats.....	83

4. DESENVOLUPAMENT.....84

Procediments d'instal·lació.....	85
Instal·lació de Subversion.....	85
Instal·lació d'Eclipse, EasyEclipse i Aptana.....	87
Eclipse flavours.....	87

Instal·lació.....	87
El workspace.....	88
EasyEclipse.....	88
Aptana.....	88
Instal·lació de Netbeans.....	89
Instal·lació de FishEye.....	89
Requisits.....	89
Instal·lació.....	89
Instal·lació d'Oracle 10g Express.....	90
Instal·lació de GlassFish V2.....	91
Configuració del driver JDBC d'Oracle.....	92
Instal·lació d'Eclipse BIRT.....	93
BIRT Report Designer.....	93
BIRT Runtime.....	93
Instal·lació del runtime.....	93
Instal·lació dels drivers JDBC.....	93
Instal·lació de la dependència Apache Commons Logging.....	94
Prova de funcionament.....	94
Instal·lació d'OpenReports.....	95
Instal·lació dels drivers JDBC.....	95
Configuració de la base de dades.....	95
Creació de l'esquema i les taules.....	95
Creació de l'usuari administrador.....	95
Configuració d'Hibernate i de l'scheduler.....	96
Configuració del directori contenidor dels .properties.....	97
Creació del fitxer empaquetat .war.....	97
Configuració del directori d'informes amb l'administrador web.....	98
Procediments de configuració i ús.....	99
Guia ràpida de GlassFish V2.....	99
Consola d'administració web.....	99
Logs del servidor.....	99
Directorio d'autodeploy.....	100
Desplegament amb Eclipse WTP.....	100
Desplegament amb NetBeans i Sun Java Studio Creator.....	100
Desplegament de prova.....	100
Creació i configuració d'un dipòsit.....	101
Creació del dipòsit.....	101
Planificació de l'organització d'un dipòsit.....	102
Importació de dades.....	102
Altres aspectes.....	103
Creació d'un esquelet de projecte amb AppFuse.....	104

Requeriments.....	104
Creació de l'esquelet.....	104
Oracleitzant l'AppFuse.....	105
Preparant l'execució de l'aplicació.....	107
Provant l'aplicació sobre el servidor d'aplicacions Jetty.....	107
Provant l'aplicació sobre el servidor d'aplicacions GlassFish.....	107
LDAPitzant AppFuse.....	108
El mode de codi complet d'AppFuse.....	110
Preparant el projecte per Eclipse i NetBeans.....	111
Eclipse.....	112
Integració amb Eclipse WTP.....	113
Netbeans 6.0.....	115
Configuració de log4j per aplicacions del SER.RI-TIC.....	116
Format de la sortida.....	116
Invocació de l'objecte Logger en aplicacions Java i J2EE.....	116
Invocació de l'objecte Logger en pàgines JSP.....	116
Logging d'Exceptions.....	117
Estructura de carpetes.....	117
Fitxer de configuració log4j.properties.....	117
Classe de reinicialització del sistema de logging.....	118
Desenvolupaments amb les eines.....	120
Mètode de prova i enunciat de l'aplicació a desenvolupar.....	120
Desenvolupament amb Eclipse 3.3 Europa.....	121
Desenvolupament amb NetBeans 6.0.....	137
Desenvolupament amb eines de generació d'informes.....	150
Configuració del driver JDBC d'Oracle i l'esquema HR.....	151
Creació d'un informe d'exemple.....	151

5. CONCLUSIONS.....155

Definició de la solució escollida.....	156
Avantatges de la solució.....	159
Vigència de la solució.....	159
Cicle de vida del desenvolupament d'un projecte.....	160
Requeriments de maquinari dels desenvolupadors.....	161
Proposta de curs de formació del SER.RI-TIC.....	162

6. RECURSOS UTILITZATS.....164

Bibliografia i pàgines web.....	165
Programari.....	169
Maquinari.....	169
7. AGRAÏMENTS.....	170
8. LLICÈNCIA.....	172
Llicència.....	173
Text complet de la llicència.....	174
ÍNDIX ALFABÈTIC.....	179
MANUALS.....	184
Manual del desenvolupador del SER.RI-TIC.....	185
Manual del desenvolupador extern al SER.RI-TIC.....	185
ANNEX I: ENTRADES DEL BLOG.....	186

Índex de taules

Taula 1: Requisits tecnològics del SER.RI-TIC.....	5
Taula 2: Conceptes sobre el control de versions.....	16
Taula 3: Comparació sobre les eines de gestió de bugs i issues.....	24
Taula 4: Comparació sobre wikis.....	28
Taula 5: Comparació sobre eines UML.....	32
Taula 6: Comparació de les característiques dels IDEs.....	36
Taula 7: Comparació d'altres característiques dels IDEs.....	37
Taula 8: Comparació de les característiques dels servidors d'aplicacions.....	42
Taula 9: Utilitats lliures diverses.....	43
Taula 10: Comparació de tecnologies de persistència.....	48
Taula 11: Conseqüències del canvi de codi font per l'eina de muntatge.....	50
Taula 12: Comparació sobre eines de construcció de projectes.....	54
Taula 13: Comparació sobre eines de reporting.....	59
Taula 14: Comparació dels frameworks de presentació.....	68
Taula 15: Exemple de configuració de fitxers de propietats en castellà i anglès.....	70
Taula 16: Tipus de tests de software.....	75
Taula 17: Comparació sobre eines de testing unitari.....	77
Taula 18: Comparació sobre eines de tests de càrrega.....	79
Taula 19: Equivalències de formats lliures i propietaris segons l'àmbit.....	83
Taula 20: Planificació de l'organització de carpetes d'un dipòsit.....	102
Taula 21: Cadena de configuració del ConversionPattern de log4j.....	116
Taula 22: Bateria de proves del desenvolupament.....	120
Taula 23: Definició de l'entorn de desenvolupament.....	157
Taula 24: Requeriments de maquinari dels desenvolupadors.....	161

Índex de figures

Figura 1: Esquema de treball de versions centralitzat.....	15
Figura 2: Arquitectura de Subversion.....	18
Figura 3: Comparació de les tasques d'Eclipse i Maven 2.....	54
Figura 4: Arquitectura del desenvolupament d'informes amb Eclipse BIRT.....	56
Figura 5: Arquitectura Model-Vista-Control d'AppFuse.....	62
Figura 6: Arquitectura dels frameworks d'Spring.....	64
Figura 7: Cicle de vida de les peticions amb Struts 2.....	66
Figura 8: Disseny de l'esquema d'exemple HR.....	150
Figura 9: Disseny del layout per la generació de l'informe.....	153
Figura 10: Resultat de l'informe de BIRT generat amb PDF.....	153
Figura 11: Mapa conceptual de la solució escollida per l'entorn de desenvolupament	158
Figura 12: Mapa conceptual del cicle de vida del desenvolupament d'un projecte.....	160
Figura 13: Mapa conceptual de la proposta de formació.....	162

Convencions de format del document

Per la redacció d'aquest document s'ha utilitzat les següents convencions:

- El text en *cursiva* per marcar les paraules angleses.
- El text entremig de cometes per fer referència a noms de carpetes, noms de fitxers o accions de menús.
- El text amb fons groc, per exemple,

Exemple de codi font

per indicar que el contingut és una peça de codi font.

- El text amb fons morat, per exemple,

Exemple de comanda

per indicar que el contingut és una comanda a executar a la línia de comandes, una ruta absoluta d'una carpeta o fitxer o una URL.

Descàrrec

Aquest projecte s'ha realitzat a l'àrea d'Aplicacions del Servei de Recursos Informàtics i TIC de la Universitat Rovira i Virgili, on l'autor del document actualment està treballant, en la creació d'aplicacions web com a tècnic informàtic.

Dintre del servei s'ha comptat amb una cotutora que ha guiat el desenvolupament del projecte, la tècnica d'e-serveis Maria Teresa Bordas. A més, com a supervisor dels resultats del treball, ha participat el director del servei, Lluís Alfons Ariño.

1. Objectius del projecte

To err is human--and to blame it on a computer is even more so.

Robert Orben

La URV i les aplicacions informàtiques

La Universitat Rovira i Virgili té tres missions principals:

- Recerca
- Docència
- Transferència de coneixement a la societat

Dintre de cada missió hi ha unes necessitats informàtiques inherents a l'època actual de la comunicació. Els usuaris volen tota la informació amb accés immediat i des de qualsevol lloc. I a més, volen que aquesta experiència sigui el més rica i versàtil possible.

Per tant, és necessari proveir d'unes eines polivalents i funcionals a tots els implicats, els alumnes, els docents i el personal d'administració i serveis (PAS), per aconseguir una alta productivitat i un acontentament general de la comunitat universitària.

El Servei de Recursos Informàtics i TIC

El Servei de Recursos Informàtics i TIC, s'encarrega de la planificació i explotació dels sistemes d'informació de la Universitat Rovira i Virgili.

Amb competències TIC sobre els àmbits de la Gestió, la Recerca i la Docència, col·labora en la planificació estratègica per donar resposta i avançar-se als diferents requeriments de la comunitat universitària.

Per dur a terme aquests objectius s'encarrega des de la capa i elements d'infraestructura que possibiliten els serveis més bàsics, fins al capa de planificació estratègica.

Compresa dintre la planificació estratègica està la planificació de solucions de programari a mida. El servei compta amb un equip de desenvolupadors per la creació d'aplicacions. Però, quan els requisits així ho determinen, els desenvolupaments s'encarreguen a terceres empreses. Per aquest motiu la definició d'un entorn de desenvolupament no només afecta als desenvolupadors del SER.RI-TIC, sinó que també afecta als desenvolupadors d'aquestes empreses.

Objectius generals

El principal objectiu del projecte és la definició d'un entorn de desenvolupament pel SER.RI-TIC. Aquest entorn es definirà utilitzant eines i tecnologies lliures i intentant seguir i ampliar els criteris tecnològics ja adoptats pel servei.

S'entén per entorn de desenvolupament:

1. El conjunt de serveis centralitzats que el propi SER.RI-TIC prestarà als desenvolupadors per possibilitar i millorar el procés de desenvolupament de les aplicacions.
2. El conjunt d'eines informàtiques lliures que els desenvolupadors utilitzaran per aconseguir un producte final pels usuaris.
3. El conjunt de tecnologies lliures, *frameworks* i llibreries a emprar durant l'etapa de disseny i implementació de les aplicacions.

Donada la participació de terceres empreses en les aplicacions destinades als membres de la Universitat, també es fa necessària la creació d'una guia de criteris tecnològics. Aquest criteris tecnològics seran, com a norma general, d'indispensable compliment pel desenvolupament d'aplicacions externes que s'hagin d'instal·lar en l'entorn de producció del SER.RI-TIC.

És clar que aquesta guia de criteris serà més laxa que el conjunt de criteris de l'entorn de desenvolupament, però farà èmfasi sobretot amb el conjunt de tecnologies a emprar.

Objectius formatius

A part de la definició raonada de l'entorn de desenvolupament, també s'espera aconseguir tot un seguit de documentació tècnica pels desenvolupadors, en forma de manuals, per disminuir la pendent de la corba d'aprenentatge global de tot l'entorn.

2. Especificaciones

¿Por qué esta magnífica tecnología [científica], que ahorra trabajo y nos hace la vida mas fácil, nos aporta tan poca felicidad?

La respuesta es esta, simplemente: porque aún no hemos aprendido a usarla con tino

Albert Einstein

Requisits tecnològics del SER.RI-TIC

Els actuals mètodes de desenvolupament d'aplicacions web del SER.RI-TIC són els requisits tecnològics sobre els que ha partit el projecte. Aquests criteris són els resumits a la taula 1:

Taula 1: Requisits tecnològics del SER.RI-TIC.

ÀMBIT	REQUISIT	LLICÈNCIA
Base de dades	Oracle 10g	Privativa
Desenvolupament SQL	TOAD for Oracle	Privativa
Gestió de projectes	dotProject	GPL v2
Identificació	LDAP, Sun Java System Acces Manager	Privativa
Plataforma	Java 2 Enterprise Edition (J2EE)	GPL v2
Servidor d'aplicacions	Sun Java System Application Server	CDDL ¹ i GPL v2 ²

L'acompliment d'aquests requisits té com a principal objectiu aconseguir una compatibilitat cap enrere amb les aplicacions actualment existents.

Destinatari del projecte

Els destinataris principals del projecte són tots els membres del SER.RI-TIC, amb especial incidència els membres de l'àrea d'Aplicacions. Actualment aquesta àrea consta de 16 membres, entre tècnics d'aplicacions, tècnics de suport i programadors. Per tant, es podria concretar l'objectiu en implantar l'entorn de desenvolupament principalment en un conjunt de 15-20 persones.

Pel desenvolupament del projecte, es considerarà que el destinatari final té següents coneixements:

- Coneixements d'IDEs
- HTML i CSS
- Interès tecnològic
- Java
- SQL i PL/SQL

Per últim, els desenvolupadors d'empreses externes també són destinataris d'aquest projecte, ja que es definiran les tecnologies a utilitzar. Queda fora de l'ambició del projecte definir si una empresa externa ha de demostrar que els seus desenvolupadors i programadors tenen coneixements dels criteris tecnològics.

¹ *Common Development and Distribution License*, basada en la *Mozilla Public License* (MPL)

² Es desenvolupa amb llicència lliure com a GlassFish. Té una part privativa, la Classpath Exception

Aproximació al problema

Es necessiten solucions pràctiques per facilitar el desenvolupament d'aplicacions web. Què significa “pràctiques” per a les eines que s'utilitzen en entorns de desenvolupament?

- Accessibles.
- De cost apropiat, però idealment lliures i gratuïtes.
- D'instal·lació i configuració usable, per fer el desenvolupament possible, i que l'eina no falli regularment ni tingui molts de *bugs*.
- Que la documentació de l'eina sigui adequada.

Seguidament es farà una revisió de les parts que ha de tenir l'entorn de desenvolupament a dissenyar. Juntament amb la definició de cada part, es realitzarà una breu explicació. A més a més, es donaran alguns exemples del món Java per facilitar-ne la comprensió.

Les següents seccions d'aquest capítol estan ordenades, aproximadament, segons la cronologia del seu ús durant desenvolupament d'una aplicació.

Planificació i gestió de projectes

Tot projecte ha de començar per una planificació temporal i dels recursos humans que té previst utilitzar. Hi ha eines específiques que permeten facilitar i comunicar a tot l'equip de treball la planificació actual, així com fer seguiments del treball a realitzar per cada membre de l'equip.

Actualment al SER.RI-TIC ja es disposa d'una eina de gestió de projectes, el dotProject.

Entorn Integrat de Desenvolupament (IDE)

Un IDE és una aplicació integrada d'un gran nombre d'eines per un programador. Integra des del compilador fins a un depurador, passant per editors de text, eines d'automatització, control de versions, disseny de GUIs... És per tant, una eina que s'ha de considerar bàsica per qualsevol equip de treball.

Actualment els IDEs més utilitzats són l'Eclipse i el NetBeans.

Generadors d'esquelets de codi

La generació automàtica de codi és una eina sovint oblidada, però que facilita en gran mesura el desenvolupament d'aplicacions web, sobretot en les etapes inicials dels projectes. El motiu és que els desenvolupadors, a l'igual que els escriptors, sofreixen el mal del "pànic a la pàgina en blanc".

La majoria d'IDEs porten incorporats potents assistents o *wizards* per superar el petit tràngol inicial del projecte en blanc. Existeixen però, *frameworks* específics per cobrir de forma més eficient aquestes necessitats, afegint contingut útil i no creant únicament l'esquelet.

El generador d'esquelets més àmpliament utilitzat per Java és l'AppFuse.

Control de versions o *Software Configuration Management* (SCM)

Els SCMs ens permeten mantenir diferents versions del codi font que s'utilitza per crear productes de *software* dintre d'un depòsit (*repository*). Encara que només sigui una persona la que canviï aquest codi, pot ser interessant veure com un projecte evoluciona al llarg del temps.

També és una bona forma de compartir col·laborativament i de forma efectiva el codi d'un projecte, ja que permet controlar qui pot accedir i modificar els fitxers.

Lligat als SCM hi ha altres conceptes com ara l'explotació de la informació emmagatzemada als dipòsits i la utilització d'eines específiques per la revisió de codi desenvolupat.

Els més utilitzats són el CVS i el Subversion, tot i que aquest últim s'està convertint en l'actual *estàndard de facto*, sobretot per projectes Java.

Eines comunicatives

Com s'ha comentat anteriorment, els destinataris són entre 15-20 desenvolupadors, per tant, els equips de treball seran d'unes 5 persones.

Independentment del nombre de membres de l'equip de treball, és beneficiós per la fluïdesa correcta d'informació, proveir d'eines als desenvolupadors per comunicar-se. L'eina ideal hauria de ser centralitzada, amb gestió de permisos i de fàcil ús. Actualment l'eina que s'està imposant per complir aquesta tasca és el *wiki*.

Els *wikis* més utilitzats són el MediaWiki i el Confluence.

Registre d'aplicacions (*logging*)

Durant el desenvolupament i, també, a l'entorn de producció és necessari comptar amb eines que informin del mal funcionament de les aplicacions desenvolupades. L'objectiu és anticipar-se, en la mesura que sigui possible, a la detecció d'errors per part dels usuaris finals. A més, una vegada detectat l'error també necessitarem concretar la posició exacta d'aquest error dintre del codi font de l'aplicació implicada.

Les eines per dur a terme aquesta tasca són les eines de registre (*logging*), que ens permetran seguir el flux d'execució de les aplicacions itjançant diversos mitjans: fitxers de registre, consola del sistema, correus electrònics...

L'eina més utilitzada és, sense cap dubte, el log4j.

Internacionalització (i18n) i localització (l10n)

Un altre aspecte dels entorns de producció és la internacionalització (i18n³) i la localització (l10n⁴).

I18n és dissenyar i implementar un producte que podrà ser localitzat, és a dir, tenir en compte des dels inicis de l'aplicació que aquesta tindrà una interfície en diversos idiomes.

L10n és el treball de crear una versió del producte per una configuració regional. Això implica des de traduccions fins a canvis en l'ordenació de cadenes, diferents combinacions de tecles, canvis al format de dates, números i monedes i, fins i tot, canvis al disposició de la interfície de l'usuari.

Eines de *building*

Les aplicacions es construeixen a partir dels seus fitxers font, el codi font, que hauria de ser una col·lecció de fitxers dintre d'un dipòsit SCM. L'eina de construcció segueix una sèrie de regles configurades adequadament per construir l'aplicació.

Els més habituals són l'Ant i el Maven.

3 NdR: s'ha escollit l'acrònim i18n perquè a la paraula anglesa *internationalization* hi ha 18 lletres entre la i inicial i la n final

4 NdR: de forma semblant, s'ha escollit l10n per la paraula *localization*

Testing

Les eines de *testing* permeten fer créixer l'àmbit de proves dels programadors, des de petites parts de codi (*unit testing*) fins a tota l'aplicació sencera (*integration testing*).

És important tenir una eina pràctica de *testing*, ja que sinó els programadors s'abandonaran a entregar aplicacions no depurades totalment. És important també mentalitzar-los de la necessitat de testar les classes desenvolupades.

Les eines de testing més habituals són el Junit, el TestNG, el JMeter i el Selenium.

Gestió de *bugs*

Els tests d'aplicacions proporcionen informació sobre quines parts funcionen correctament i quines no. Aquesta informació ha de ser accessible per als programadors, els testers, els mànagers i per a qualsevol que intervingui en la decisió de publicar una versió d'una aplicació. Facilitar aquesta accessibilitat és la tasca que realitza l'eina de gestió de *bugs*.

Un *bug* o *issue* és una petició de modificació, defecte o incident.

Les eines més utilitzades són el JIRA i el Trac.

Escriure la documentació

El producte ideal és tant "transparent" que no necessita documentació, però a l'hora de la veritat sempre s'ha de documentar.

Per documentació s'entén tant la orientada al client final com la tècnica, ja sigui la de disseny, desenvolupament o la d'una API per altres programadors.

Els formats més comuns per escriure documentacions són text bàsic, XML en DocBook, OpenOffice i Microsoft Word.

3. Disseny

Web 2.0 applications may often need such highly specialized experts, but more than anything else, they need developers who are perhaps not experts in all these areas, but understand enough of each technology to get the big picture, and who understand the division of roles between them and the trade-offs that will be made.

“Professional Web 2.0 Programming”
Eric van der Vlist et al.

Mètode de desenvolupament del projecte

Per dur a terme el projecte inicialment es van planificar tres grans etapes:

- recerca inicial d'informació.
- instal·lació de tots els elements a avaluar dintre el projecte.
- desenvolupament per comprovar la efectivitat dels elements escollits.

Recerca inicial d'informació

Qualsevol tipus de projecte innovador ha de començar amb una intensa recerca d'informació per aconseguir conèixer quin és l'*state-of-the-art* sobre el camp investigat.

Aquesta tasca inicial de recerca d'informació va començar a partir del llibre *Practical Development Environment* [1]. Al ser un llibre del 2005 però, molts conceptes no eren plenament vigents en l'actualitat. De totes formes, ha servit per aconseguir una àmplia visió de quin havia de ser el producte final del projecte.

D'altra banda, la recerca d'informació va continuar per la vessant electrònica, mitjançant tots els mitjans d'informació que dona Internet: blogs, wikipedia, pàgines de projectes, llistes de correu... Especialment importants han estat les comunitats de desenvolupament com *The Server Side* [2], *JavaWorld* [3], *developerWorks* [4] o *J2EE Programming with Passion!* [5]

Aquesta fase de recerca d'informació va donar lloc a la creació d'un document intern del Servei [6], on s'explicaven les eines que hauria de tenir l'entorn de desenvolupament i es feia una explicació sobre les alternatives possibles. Aquest document posteriorment ha estat introduït a la documentació tècnica d'aquest projecte.

Una part important, i que també es pot entendre com recerca d'informació, ha estat el contacte amb els membres del SER.RI-TIC, per conèixer les seves inquietuds i les seves necessitats. El motiu era aconseguir planificar un acoblament progressiu entre l'entorn de desenvolupament resultant del projecte i el treball del dia a dia dels desenvolupadors. Aquesta és, potser, la part més important, ja que sense una adopció positiva i oberta per part dels destinataris, el projecte està destinat al fracàs.

Instal·lació d'un cas real

Un cop les especificacions eren clares, els requisits inicials havien estat fixats pel SER.RI-TIC i s'havia fet la recerca d'informació el següent pas previst era el treball de camp. Es va planificar una instal·lació per simular l'entorn real amb que es troben els programadors i fer totes les proves necessàries en aquell entorn.

A l'ordinador portàtil, que el SER.RI-TIC va cedir per la realització del projecte, es va instal·lar aquest entorn de simulació. A grans trets, aquest entorn es basa en:

- Sistema operatiu Windows XP Professional SP2. Aquest és el sistema operatiu instal·lat nadiwament al portàtil. Actua com a sistema operatiu del desenvolupador.
- Aplicació de virtualització VMWare. Mitjançant de la virtualització es simulen els recursos centralitzats oferts pel SER.RI-TIC, com per exemple el servidor de control de versions Subversion. Per fer-ho el més real possible, s'ha instal·lat el sistema operatiu Solaris 10 (per plataforma x86), que és el sistema operatiu utilitzat majoritàriament a l'àrea d'infraestructures. A sobre de Solaris s'han instal·lat tots els serveis necessaris per a la simulació [7].

A partir d'aquest punt s'ha treballat amb totes les proves d'integració d'aplicacions que s'explicaran posteriorment al capítol de desenvolupament.

Desenvolupament del projecte

El desenvolupament de codi font s'ha realitzat sobretot a la recta final del projecte, on ja es tenia gairebé clar quin seria l'escenari final. Ha servit per veure que l'escenari era factible, compatible i exempt de qualsevol problemàtica tècnica.

Hi ha dos casos especials, els IDEs i les eines de generació d'informes, en que era necessari fer proves d'ús per acabar de decidir l'eina més adequada per cobrir les necessitats del servei.

Dedicació temporal

El projecte s'ha desenvolupat majoritàriament en hores no laborals. El SER.RI-TIC però, ha permès utilitzar hores laborals de l'autor per la seva realització.

Eines de suport al desenvolupament del projecte

Per gestionar la realització del projecte s'han utilitzat dues eines col·laboratives, amb l'objectiu de facilitar la comunicació amb els tutors i de dur a terme un control sobre la feina feta.

Aquestes dues eines han estat:

- Un *blog* [8] que ha estat utilitzat com a quadern de laboratori, és a dir, amb totes les proves, experiències, processos d'instal·lació, idees... qualsevol aspecte ha quedat enregistrat al *blog*. D'aquesta forma s'ha facilitat la comunicació i la posterior redacció de la documentació tècnica. A petició del director del SER.RI-TIC el *blog* és privat i per accedir-hi es necessita un usuari.
- El gestor de projectes dotProject. Aquest gestor s'utilitza regularment al SER.RI-TIC pel treball diari. Aprofitant la familiaritat amb l'eina es va proposar anar elaborant la planificació del projecte i anotar el número d'hores de treball involucrades al projecte.

Al final d'aquest document es presentaran un annex amb els articles publicats al *blog* (Annex I).

Des de la data d'entrega de la documentació, 17 de gener de 2007, fins la data de l'últim dia de tribunal, 31 de gener de 2007, hi haurà activat l'usuari "entregapfc", amb contrasenya "entregapfc", per poder entrar al blog. La direcció del blog és <http://www.lotomas.net/pfcblog/>

Arbre de disseny

Aquesta secció de la documentació s'utilitzarà per mostrar extensivament totes les solucions que s'han estudiat i que tenien possibilitats d'entrar dintre l'entorn de desenvolupament final.

Per algunes de les necessitats s'explicarà només una solució, ja sigui perquè no existeixen alternatives al mercat, perquè és un requeriment del SER.RI-TIC o perquè aquesta solució és un estàndard de facto. En d'altres, es comentaran diverses solucions per complir una mateixa tasca, a mode comparatiu.

S'han estructurat les solucions en tres apartats:

- Arquitectura centralitzada – Solaris: dintre d'aquest apartat s'estudiaran les solucions sobre les que hauria de donar un servei centralitzat el SER.RI-TIC.
- Desenvolupador – Windows i/o GNU/Linux: dintre d'aquest apartat s'estudiaran les solucions específiques amb que treballarà el desenvolupador, les eines de treball pròpiament dites.
- Tecnologies: s'estudiaran les solucions tecnològiques, en forma de *frameworks*, llibreries o APIs a utilitzar dintre dels desenvolupaments d'aplicacions web al servei.

Dintre de cada eina o tecnologia, que tingui diversos components entre els que elegir i que no sigui un requisit del SER.RI-TIC, s'inclourà un apartat anomenat "Avaluació". Aquest apartat inclourà els criteris que han fet decidir-se per la solució elegida.

Segons l'esquema de documentació dels projectes de final de carrera, l'avaluació va en un capítol separat, després del de desenvolupament. El canvi d'organització és per motius pràctics. L'objectiu és facilitar la lectura i la interpretació de la informació.

Arquitectura centralitzada – Solaris

Base de dades

Oracle 10g

L'ús de la base de dades Oracle és un requisit imposat pel SER.RI-TIC, com ja s'ha comentat a la taula 1.

Actualment tots els desenvolupaments que han de treballar amb una base de dades estan preparats per fer-ho amb una base de dades Oracle. L'objectiu és la integració de dades sota una mateixa tecnologia, per tal de facilitar la comunicació de dades entre aplicatius.

Per la realització del projecte, la base de dades s'ha instal·lat sobre Windows enlloc d'instal·lar sobre Solaris 10, perquè s'ha utilitzat la versió gratuïta de la base de dades, l'Oracle 10g Express, i només hi ha disponibles versions per Windows i GNU/Linux. Aquest petit canvi, però, no afecta en cap moment la simulació de l'entorn real.

Control de versions (SCM)

Els SCMs més utilitzats són els d'arquitectura centralitzada, com el que es mostra a la figura 1:

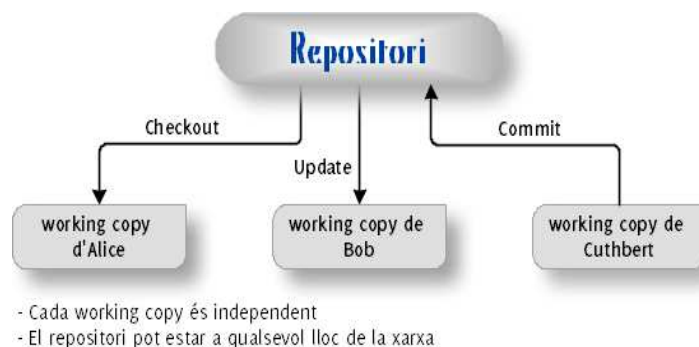


Figura 1: Esquema de treball de versions centralitzat

S'ha de reconèixer que és una de les eines que necessita més coneixements teòrics, ja que l'esquema de treball difereix substancialment de l'habitual "copia de carpetes entre desenvolupadors".

Sobre el control de versions és necessari conèixer un conjunt de conceptes, mostrats a la taula 2, ja que s'utilitzaran en gran mesura a partir de la implantació de l'entorn:

Taula 2: Conceptes sobre el control de versions.

CONCEPTE	EXPLICACIÓ
<i>Branch</i>	És una còpia paral·lela d'un conjunt de fitxers, i amb un cert nom
<i>Changelog</i>	És una llista dels canvis als fitxers, ordenada per temps
<i>Changeset</i>	És un grup de canvis relacionats amb un conjunt de fitxers, que o s'apliquen a tots a l'hora o no s'apliquen
<i>Checkout</i>	Acció d'aconseguir el <i>working copy</i> . Només es fa el primer cop
<i>Commit</i>	Acció de publicar els canvis realitzats a la <i>working copy</i> cap al dipòsit comú
<i>Diffs</i>	Canvis entre dues versions d'un fitxer
<i>Merging</i>	És quan es copien els canvis fets en una branca cap una altra branca. No sempre és fàcil ni possible
dipòsit	Col·lecció de fitxers que manega un SCM. Dintre d'un SCM poden haver-hi el control de versions de diversos projectes
<i>Tag, label</i>	Etiquetar un conjunt de fitxers amb un nom, com per exemple RELEASE_1_0. Un <i>tag</i> marca un instant en el temps, amb la qual cosa no es pot canviar els fitxers sense moure el <i>tag</i>
<i>Update</i>	Acció d'actualitzar la nostra <i>working copy</i> amb els canvis que altres membres de l'equip han realitzat als fitxers del dipòsit
<i>Working copy</i>	Conjunt de fitxers amb que el desenvolupador està treballant

La majoria de conceptes no s'han traduït, ja que dintre d'aquest àmbit, són d'ús ordinari en la forma anglesa.

Un altre aspecte que s'ha de conèixer és el tipus de model de concurrència durant l'edició de fitxers. Hi ha dos possibles tractaments:

- *Locking model*: s'evita l'edició de fitxers per part de terceres persones mentre la persona que ha sol·licitat el *lock* està editant el fitxer.
- *Merging model*: l'eina d'SCM s'encarrega de combinar els canvis produïts a un mateix fitxer, en la mesura que sigui possible. Sinó, el fitxer és marcat com a conflictiu perquè sigui l'usuari qui el resolgui.

A la Wikipedia podem trobar una llista completa de solucions SCM [9] i també una comparativa d'aquestes [10]. De totes formes, hi ha poc marge de decisió sobre aquest aspecte, ja que a menys que els desenvolupaments siguin en C i C++ (on encara regna el CVS), l'estàndard de facto és Subversion [11]. Fins i tot serveis de desenvolupament gratuïts que inicialment donaven servei CVS (SourceForge [12], java.net) estan migrant a Subversion.

Subversion

Està dissenyat com una aproximació moderna al CVS, ja que, tot i que tenen moltes diferències, és l'evolució natural d'aquest. Conserva el mateix model client/servidor utilitzat per CVS.

NOM: Subversion
DESENVOLUPADOR: CollabNet, Inc.
ÚLTIMA VERSIÓ: 1.4.6 - 20/12/2007
LLICÈNCIA: Subversion license
COST LLICÈNCIA: Gratuïta
PÀGINA WEB: <http://subversion.tigris.org/>
COMPATIBILITAT S0: Multiplataforma
COMPATIBILITAT BD: Alta

Característiques a destacar:

- S'ha passat a una base de dades Berkeley DB (ara propietat d'Oracle). Té accés per sistema de fitxer, webdav/deltav, ssh o protocol svn.
- Els directoris també es versionen, d'aquesta forma es pot canviar el nom a un directori i no perdre la seva història de *commits*.
- Operacions de *commit* atòmiques.
- Inclusió de metadata amb les versions.
- Ple suport per fitxers binaris.
- Optimització dels *tags* i branques, és molt menys costós en nivell de servidor.
- Molt bona documentació i suport per part de la comunitat lliure.

Per entendre millor com funciona, i per veure reflectides algunes de les característiques comentades, a la figura 2 es mostra un esquema de l'arquitectura de Subversion.

La documentació oficial del projecte Subversion [13] és excel·lent i està traduïda a diversos idiomes, entre ells el castellà. A més, hi ha publicats una gran quantitat de llibres de suport [14].

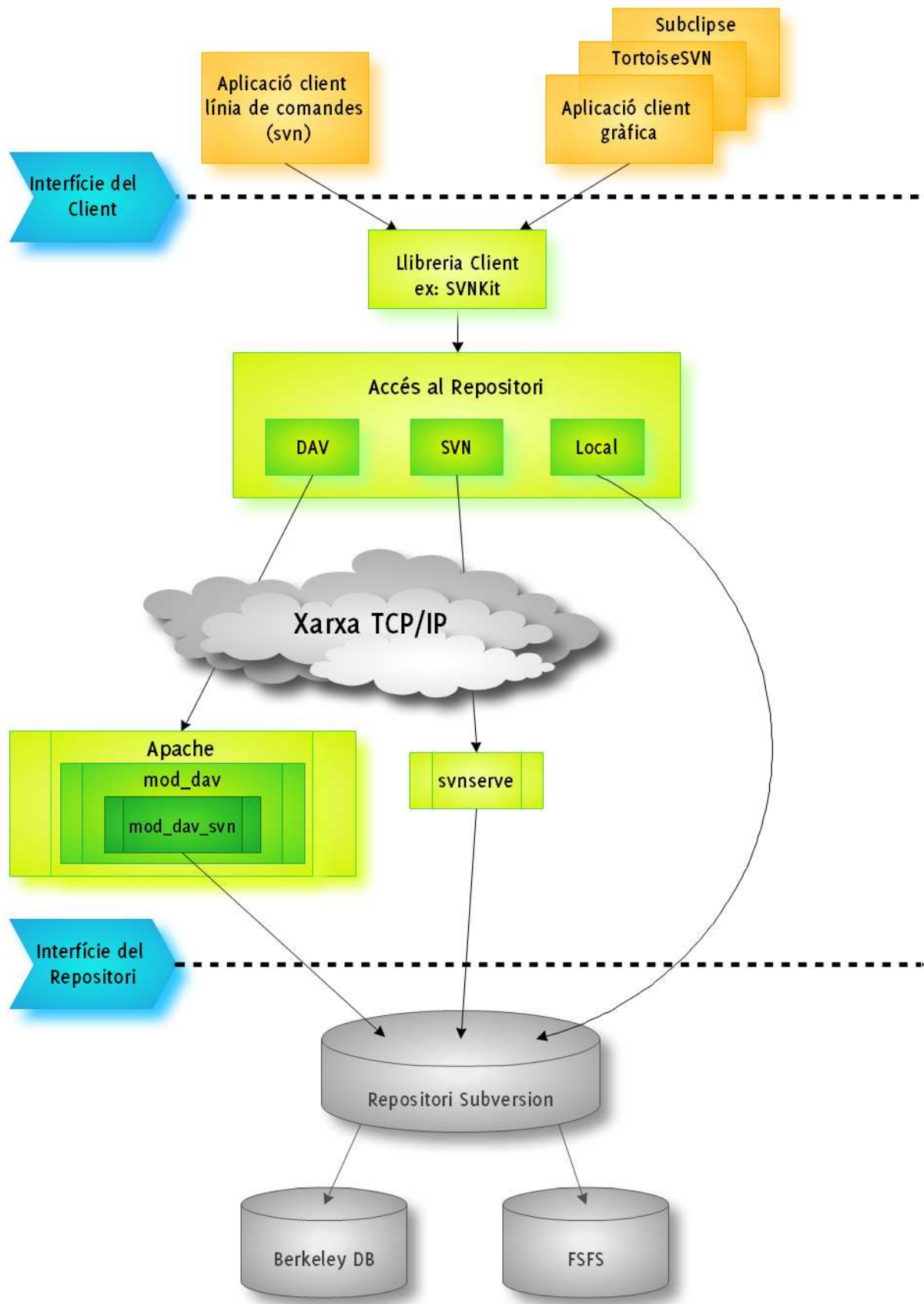


Figura 2: Arquitectura de Subversion.

FishEye

És una utilitat que s'integra amb els dipòsits de Subversion per aconseguir extreure informació no visible a simple cop d'ull amb les eines estàndards.

NOM:	FishEye
DESENVOLUPADOR:	Atlassian Software Systems Pty Ltd
ÚLTIMA VERSIÓ:	1,4,1 - 19/12/2007
LLICÈNCIA:	Semi-Privativa
COST LLICÈNCIA:	Acadèmica 750,00€/25 usuaris
PÀGINA WEB:	http://www.atlassian.com/fisheye/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

FishEye permet obrir el dipòsit per ajudar a entendre com evoluciona el codi font, facilitant la lectura i la comprensió d'aquesta informació. Tot aquest potencial informatiu no està destinat al creador de codi, el programador o el desenvolupador, sinó que està orientat a figures més altes, com ara els caps de projecte o arquitectes de *software*. El principal avantatge és l'abstracció de l'excés de detall que s'obté llegint directament la informació dels dipòsits.

Les principals característiques que aporta FishEye són:

- Monitorització: permet fer un seguiment precís dels canvis del codi font, així com configurar alertes i avisos mitjançant RSS i de correu electrònic.
- Cerca avançada: permet cercar transversalment per comentaris, rutes de fitxers, codi font, autors... fins i tot integra un llenguatge relacional, l'EyeQL.
- Anàlisis estadístics: permet veure la informació gràficament.

Es poden veure exemples en línia d'altres projectes, com ara MySQL [15] i el dipòsit públic de projectes lliures Codehaus [16].

La llicència és un pèl especial, ja que és privativa però t'entreguen el codi font de la part de la presentació per poder-la personalitzar segons les pròpies necessitats. Tenen llicències especials per universitats. Inclou l'ús indefinit de l'aplicació i manteniment durant 1 any.

No hi ha alternatives a aquest producte (ni lliures ni privatives), ja que està en un nínxol molt específic.

La documentació és pública i molt completa.

Crucible

És una eina potent complementària a FishEye. Permet fer revisions del codi que hi ha dintre els dipòsits, d'una forma eficient i distribuïda.

NOM:	Crucible
DESENVOLUPADOR:	Atlassian Software Systems Pty Ltd
ÚLTIMA VERSIÓ:	1.2.1 - 19/12/2007
LLICÈNCIA:	Semi-Privativa
COST LLICÈNCIA:	Acadèmica 750,00€/25 usuaris
PÀGINA WEB:	http://www.atlassian.com/crucible/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

La realització de revisions de codi té diferents objectius, però tal i com expliquen Wiegers i Moore [17], els principals motius són la millora de la qualitat del codi i l'educació dels membres de l'equip en noves tecnologies, mitjançant codi real.

Un aspecte que també destaquen a l'article és els beneficis aportats pel *peer reviewing*, on la revisió de codi no la realitza el propi autor del codi, sinó que la realitza una altra persona externa al desenvolupament. Sobretot ha de quedar clar que no s'està avaluant el treball del programador, només el codi en si.

Els costos d'aquesta metodologia són justificables, ja que eviten la propagació d'errors, tant de disseny com d'implementació, cap a etapes finals del producte, on el cost de correcció pot ser molt alt.

Per a l'equip de treball també té un benefici molt gran en l'educació, ja que permet la transmissió de coneixement entre companys, tot i que no treballin dintre del mateix projecte.

L'últim benefici a comentar és la petita pressió que rep un programador quan sap que el seu codi serà visualitzat per terceres persones. Un dels arguments esgrimits pels partidaris del programari lliure és la qualitat del codi generat, ja que tot desenvolupador és plenament conscient que qualsevol persona podrà veure el codi font, i per tant, sacrificarà part de la seva llibertat d'escriptura per millorar el codi, per exemple, posant més comentaris.

Per acabar d'entendre el concepte és recomanable veure el vídeo de demostració que hi ha a la pàgina del producte [18], així com el *tour* de funcionalitats.

A l'igual que amb FishEye, no hi ha cap solució alternativa, i també comparteix els mateixos termes de la llicència. La documentació és pública i de molt bona qualitat..

Avaluació

Amb les eines de control de versions no hi ha discussió possible, l'eina escollida serà Subversion. En la mesura que sigui possible, també s'aconsella la instal·lació de les dues eines explicades: FishEye i Crucible. Les tres eines són compatibles, no es necessària cap elecció.

Gestor de *bugs* i *issues*

Els sistemes de seguiment d'errors són eines col·laboratives que ajuden a treballar a un equip de persones sobre un conjunt de *bugs* o mal funcionaments. Han de tenir connexió amb el sistema d'SCM per realitzar el seguiment de canvis provocats al codi font.

Jira

És una eina dedicada a facilitar el seguiment de *bugs*, i de les peticions de millores que realitzen els usuaris. Permet, a més, la gestió de projectes dintre d'un equip de desenvolupament.

NOM:	JIRA
DESENVOLUPADOR:	Atlassian Software Systems Pty Ltd
ÚLTIMA VERSIÓ:	3.12.1 - 24/12/2007
LLICÈNCIA:	Semi-Privativa
COST LLICÈNCIA:	Acadèmica Enterprise 1.600,00€
PÀGINA WEB:	http://www.atlassian.com/jira/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Multi base de dades

Està orientada a treballar utilitzant les tasques com a unitat principal. Per aquest motiu, és un bon complement a un planificador de projectes com el dotProject.

Mitjançant una interfície senzilla, usable per personal no tècnic, es poden crear les tasques, els informes de nous *bugs* i les peticions de millores. Aquesta informació s'emmagatzema a la base de dades per realitzar-ne el seguiment i, una vegada finalitzat el cicle de vida, crear els informes de resolució d'incidents.

Es recomana veure el vídeo d'exemple de la seva pàgina web, així com donar un cop d'ull al *tour* de funcionalitats del producte. Entre aquestes funcionalitats en destaquem:

- Notificació d'informació mitjançant canals RSS i correu electrònic.
- Integració d'usuaris amb LDAP.
- Suport per comunicació mitjançant Web Services, SOAP o XML-RPC.

Per facilitar l'aprenentatge de l'eina existeix un webinari on s'explica com utilitzar aquesta eina [19]. La documentació és de molt bona qualitat.

La llicència en aquest cas varia respecte FishEye i Crucible, ja que amb la llicència d'ús s'entreguen el codi font complet de l'aplicació. L'aplicació està creada amb tecnologia J2EE.

Trac

És un *software* de desenvolupament de projectes que consta d'un *wiki* millorat i un sistema de seguiment d'incidències. Utilitza un entorn minimalista basat en una web per gestionar els projectes. Proporciona una interfície cap a Subversion, permetent treballar amb ell com a part integrant del sistema de gestió d'incidències.

NOM:	Trac
DESENVOLUPADOR:	Edgewall Software
ÚLTIMA VERSIÓ:	0.10.4 - 20/04/2007
LLICÈNCIA:	Tipus BSD
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://trac.edgewall.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	SQLite, PostgreSQL i MySQL (experim)

Les principals característiques són:

- Possibilitat d'escriure les descripcions d'incidències i els missatges de *commit* amb edició d'estil *wiki*.
- Creació de links i referències entre *bugs*, tasques, *changesets*, fitxers i pàgines del *wiki* de forma senzilla.
- Permet veure l'evolució temporal dels events d'un projecte.
- Publicació d'informació via canals RSS.
- Treballa amb les bases de dades SQLite, PostgreSQL i MySQL (experimental). S'està treballant pel suport d'Oracle, però de moment no en té.
- Està programat amb Python.

Té alguns inconvenients importants per instal·lar-lo en un entorn de producció:

- Actualment l'última versió és la 0.10.4, molt nova.
- Dependències de Python i Clearsilver que poden complicar o impossibilitar la instal·lació i en dificulten el manteniment.
- El servidor de Subversion ha d'estar a la mateixa màquina que Trac.

Mindquarry

És una eina col·laborativa lliure, destinada a grups de treball de mida petita i mitjana. Intenta ser una eina equilibrada, entre la facilitat d'ús i les funcionalitats.

NOM:	Mindquarry
DESENVOLUPADOR:	Mindquarry GmbH
ÚLTIMA VERSIÓ:	1.2beta - 24/10/2007
LLICÈNCIA:	Mozilla Public License 1.1
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.mindquarry.com/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Està dissenyada en dos components:

- El servidor és qui realitza tota la part pesada del treball. Utilitza Apache, Java i Subversion
- El client, una aplicació d'escriptori que actua com a pont entre la màquina de l'usuari i el servidor. El client està disponible per Windows, Linux i MacOSX.

Després de les proves realitzades [20] les conclusions han estat:

- El look&feel de l'aplicació i la facilitat de treball són excepcionals. La seva utilització comporta tot un gir en l'organització dels equips de treball, sobretot si fins ara no s'utilitzava cap eina d'aquest estil. Les opcions de comunicació per RSS són molt grans, així com la facilitat de creació d'informes PDF segons el contingut de la pàgina.
- D'altra banda, però, la versió de l'aplicació és només la 1.1 estable. Aquesta, tot i ser totalment funcional, encara li falta certa maduració.
- Si es pogués integrar amb un LDAP pels usuaris i amb un dotProject per les tasques, seria perfecte. Sinó, es solapa amb la gestió de tasques, perdent part del significat de l'eina.

Avaluació

A la taula 3 podem veure les característiques principals d'aquest tipus d'eines:

Taula 3: Comparació sobre les eines de gestió de bugs i issues.

	JIRA	TRAC	MINDQUARRY
LLICÈNCIA LLIURE	✗	✓	✓
GRATUÏTA	✗	✓	✓
LLENGUATGE JAVA	✓	✗	✓
BASE DE DADES ORACLE	✓	✗	✗
INTEGRACIÓ AMB SUBVERSION	✓	✓	✗
AUTENTIFICACIÓ LDAP	✓	✗	✗
SUPORT UNICODE	✓	✓	✓
CONTINUOUS INTEGRATION	✓	✗	✗
GENERACIÓ DE DOCUMENTACIÓ DINÀMICA	✓	✓	✓
COMUNICACIÓ PER CANALS RSS	✓	✓	✗
SUPORT PER SOA I WEB SERVICES	✓	✗	✗

L'eina més completa, a nivell de funcionalitat, és sense cap dubte Jira. La llicència no és lliure, però s'entrega el codi font de l'aplicació a l'adquirir-la.

Si el cost de la llicència no és motiu de dissuasió, l'eina a escollir és Jira. Si els costos no són assumibles, l'eina a escollir serà Trac.

Gestor de projectes

dotProject

L'ús del gestor de projectes dotProject és un requisit imposat pel SER.RI-TIC, com ja s'ha enumerat a la taula 1.

NOM:	dotProject
DESENVOLUPADOR:	Voluntaris
ÚLTIMA VERSIÓ:	2.1.1 - 14/11/2007
LLICÈNCIA:	BSD
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.dotproject.net/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	MySQL (Oracle no comprovat)

L'eina dotProject fa 2 anys que està implantada al SER.RI-TIC, i tots els membres (tant de l'àrea d'infraestructures com d'aplicacions) l'utilitzen regularment per informar sobre les tasques en que treballen. El seguiment de tasques té dos objectius principals:

- El seguiment per part del director i dels coordinadors del Servei, de la feina efectiva de les persones que estan sota el seu càrrec.
- La planificació dels projectes assumibles pel SER.RI-TIC.

Mensualment, es retornen informes a tots els membres com a retroacció de la informació introduïda.

Destacar que és una eina molt potent i completa, així com la seva llicència lliure.

Servidor d'aplicacions

Sun Java System Application Server

El servidor d'aplicacions que està previst utilitzar a partir d'una propera actualització és el Sun Java System Application Server. És un requisit imposat pel SER.RI-TIC, com ja s'ha enumerat a la taula 1.

NOM:	Sun Java System Application Server
DESENVOLUPADOR:	Sun Microsystems, Inc.
ÚLTIMA VERSIÓ:	9.1
LLICÈNCIA:	Privativa
PÀGINA WEB:	http://www.sun.com/software/products/appsrvr/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Segons el criteri de l'autor, cada desenvolupador necessita tenir instal·lat un servidor d'aplicacions a la seva màquina. D'aquesta forma s'aconsegueix independitzar el desenvolupament propi del de qualsevol altra persona (tècnics de sistemes, tècnics d'e-serveis...), i per tant, accelerar el temps de desenvolupament. Més endavant es comentarà aquesta necessitat pels desenvolupadors.

Wiki

És un tipus de *software* col·laboratiu que permet la creació i l'edició de pàgines web mitjançant el navegador. Ha tingut un gran èxit gràcies a la seva facilitat d'ús i a la fama aconseguida per la Wikipedia.

En l'entorn del SER.RI-TIC l'ús que se li donaria seria el d'un dipòsit de gestió del coneixement. Els desenvolupadors podrien escriure les seves experiències desenvolupant, perquè posteriorment qualsevol que es trobés en aquella mateixa situació pogués aprendre del primer. Les experiències poden ser de qualsevol tipus: instal·lacions, configuracions, trossos de codi genèric, problemes i trobar la solució...

Hi ha moltes eines de *wiki*, la majoria utilitzant PHP, tot i que també n'hi ha que utilitzen Java [21]. Tot seguit analitzarem tres solucions *wiki* orientades al mercat professional.

MediaWiki

És l'eina *wiki* més famosa, ja que va ser creada per poder fer realitat el projecte de la Wikipedia, i actualment s'utilitza per donar el seu servei.

NOM:	MediaWiki
DESENVOLUPADOR:	Wikimedia Foundation
ÚLTIMA VERSIÓ:	1.11 - 10/09/2007
LLICÈNCIA:	GPL
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.mediawiki.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	MySQL o PostgreSQL

Està escrita en PHP i únicament pot funcionar sobre bases de dades MySQL i PostgreSQL.

Les seves principals característiques són:

- Està completament preparat per treballar amb qualsevol llengua (+70).
- Alta escalabilitat (per donar servei a totes les peticions de la Wikipedia).
- No utilitza sintaxi CamelCase pels hiperenllaços.
- Edició de subseccions de pàgines.
- Existeixen complements per augmentar la funcionalitat.
- Notificació per RSS i correu electrònic.
- Integració amb LDAP.

Hi ha una versió desenvolupada en Java anomenada JAMWiki.

Confluence

És una eina col·laborativa de gestió del coneixement. El seu objectiu és facilitar la col·laboració dintre dels equips de desenvolupament i afavorir la compartició del coneixement.

NOM:	Confluence
DESENVOLUPADOR:	Atlassian Software Systems Pty Ltd
ÚLTIMA VERSIÓ:	2.7 - 12/12/2007
LLICÈNCIA:	Semi-Privativa
COST LLICÈNCIA:	Acadèmica 400,00€/25 usuaris
PÀGINA WEB:	http://www.atlassian.com/confluence/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Multi base de dades

Les seves principals característiques són:

- Possibilitat d'integrar l'autenticació amb LDAP.
- Senzilla instal·lació i manteniment, és una aplicació empaquetada WAR.
- Interfície d'usuari agradable amb editor WYSIWYG.
- Exportació de contingut a PDF.
- Molts complements per augmentar la seva funcionalitat.
- Notificació d'events mitjançant correu electrònic i RSS.
- Es poden penjar fitxers.
- Es poden deixar comentaris a les pàgines, com si fossin *blogs*.
- API per poder treballar amb ell des d'altres programes.

Totes aquestes característiques fan possible utilitzar-lo de diverses formes [22]:

- Base de dades col·laborativa.
- Base de coneixements.
- Creació de documentació final, si durant l'etapa de desenvolupament s'ha anat afegint contingut.
- Gestor de fitxers compartits, com a substitució del BSCW.
- Medi alternatiu al correu electrònic per comunicar-se amb l'equip.

La llicència en aquest cas és com la de JIRA. Amb la llicència t'entreguen el codi font complet de l'aplicació, que està creada amb tecnologia J2EE. Altres components que utilitza són Velocity, WebWork, SiteMesh, Hibernate i Lucene.

Per facilitar l'aprenentatge de l'eina tenen disponible un webminari [23] on s'explica com fer-la servir. La documentació és de molt bona qualitat.

Avaluació

A la taula 4 podem veure les característiques principals d'aquest tipus d'eines:

Taula 4: Comparació sobre wikis.

	CONFLUENCE	MEDIAWIKI
LLICÈNCIA LLIURE	✗	✓
GRATUÏTA	✗	✓
LLENGUATGE JAVA	✓	✗
BASE DE DADES ORACLE	✓	✗
CÀRREGA DE FITXERS	✓	✓
AUTENTIFICACIÓ LDAP	✓	✓
RENOMBRAT CORRECTE DE DOCUMENTS	✓	✓
EDITOR WYSIWYG	✓	✗
GENERACIÓ DE DOCUMENTACIÓ DINÀMICA	✓	✗
COMUNICACIÓ PER CANALS RSS	✓	✓
SUPORT PER SOA I WEB SERVICES	✓	✗

L'eina més completa a nivell de funcionalitat és Confluence. La llicència no és lliure, però s'entrega el codi font de l'aplicació a l'adquirir-la.

Si el cost de la llicència no és motiu de dissuasió, l'eina a escollir és Confluence. Si els costos no són assumibles, l'eina a escollir serà MediaWiki.

Desenvolupador – Windows i/o GNU/Linux

Client Subversion

Per treballar amb els dipòsits de codi Subversion tenim tres opcions:

- Utilitzar l'eina de línia de comandes (svn).
- Utilitzar una eina d'integració amb l'IDE de desenvolupament.
- Utilitzar una eina d'integració del sistema operatiu.

Repassem les eines que ens brinden les dues últimes opcions. Fer notar que són complementàries, no excloents.

Subclipse

És un *plugin* per l'IDE Eclipse que permet treballar amb dipòsits Subversion de forma integrada.

Eclipse per defecte només té suport per dipòsits CVS, de forma que amb aquest complement tindrem les mateixes opcions que amb els dipòsits Subversion.

NOM:	Subclipse
DESENVOLUPADOR:	Comunitat
ÚLTIMA VERSIÓ:	1.0.6 - 27/08/2007
LLICÈNCIA:	Eclipse Public License 1.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://subclipse.tigris.org
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Implementa totes les operacions que es poden realitzar mitjançant línia de comanda: *checkout*, *commit*, *update*, comparació gràfica de versions, *revert*, *ignore*...

TortoiseSVN

És una eina que s'integra completament amb l'explorador de fitxers de Windows.

Una vegada instal·lat apareix un nou menú de propietats de les carpetes, que permet realitzar totes les operacions típiques sobre un dipòsit.

NOM:	TortoiseSVN
DESENVOLUPADOR:	Comunitat
ÚLTIMA VERSIÓ:	1.4.6 - 21/12/2007
LLICÈNCIA:	GPL v2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://tortoisesvn.net/
COMPATIBILITAT SO:	Windows
COMPATIBILITAT BD:	Independent

És una eina molt útil, tot i que també tinguem la integració a nivell de d'IDE.

Avaluació

Les dues eines explicades són compatibles, i es recomana la instal·lació de totes dues. Com a nota final comentar que Netbeans du integrat el client de Subversion.

Creació i anàlisi de consultes SQL

TOAD for Oracle

És una eina d'administració de bases de dades i desenvolupament de codi SQL. Gairebé es podria dir que és l'eina més utilitzada a nivell professional, fins i tot per damunt de les pròpies eines d'Oracle.

NOM: TOAD
 DESENVOLUPADOR: Quest Software
 ÚLTIMA VERSIÓ: 9.5.0.31 - 15/11/2007
 LLICÈNCIA: Privativa
 COST LLICÈNCIA: Indeterminat
 PÀGINA WEB: <http://www.quest.com/toad-for-oracle/>
 COMPATIBILITAT SO: Windows
 COMPATIBILITAT BD: Oracle

Forma part dels requisits del SER.RI-TIC, ja que és una eina àmpliament utilitzada i amb la que s'han fet cursos extensius a tot el servei.

TOra

És una alternativa lliure creada a imatge i semblança de TOAD, i a més, multiplataforma. Fins i tot té el suport de Quest Software, l'empresa responsable de TOAD.

NOM: TOra
 DESENVOLUPADOR: Nathan Neulinger
 ÚLTIMA VERSIÓ: 1.3.22 - 18/09/2007
 LLICÈNCIA: GPL
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://tora.sourceforge.net/>
 COMPATIBILITAT SO: GNU/Linux
 COMPATIBILITAT BD: Oracle

L'apunt d'aquesta eina es realitza només com a informació opcional, donat que l'objectiu del projecte és oferir un entorn completament multiplataforma.

Disseny de bases de dades

Oracle Designer

Proporciona suport pel modelatge de processos de negoci, anàlisi de sistemes i disseny de bases de dades.

Està totalment integrada dintre l'Oracle Developer Suite.

NOM: Oracle Designer
 DESENVOLUPADOR: Oracle Inc.
 ÚLTIMA VERSIÓ: 10.1.2.3 - 30/04/2007
 LLICÈNCIA: Privativa
 COST LLICÈNCIA: Indeterminada
 PÀGINA WEB: <http://www.oracle.com/technology/products/designer>
 COMPATIBILITAT SO: Windows
 COMPATIBILITAT BD: Oracle

És un requisit de les especificacions donades pel SER.RI-TIC.

DBDesigner

És un sistema de disseny visual de bases de dades que integra en un únic entorn el disseny, el modelatge, la creació i el manteniment.

NOM: DBDesigner
 DESENVOLUPADOR: fabFORCE.net team
 ÚLTIMA VERSIÓ: 4.0.5.6
 LLICÈNCIA: GPL
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://fabforce.net/dbdesigner4>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Multi base de dades

És compatible amb Oracle i té versions per Windows i GNU/Linux. Disposa de *plugins* desenvolupats per la comunitat amb el que augmenta les seves funcionalitats.

En un futur aquesta aplicació passarà a anomenar-se MySQL Workbench.

L'apunt d'aquesta eina es realitza només com a informació opcional, donat que l'objectiu del projecte és oferir un entorn completament multiplataforma.

Disseny UML

NetBeans UML Project

És un editor potent d'UML integrat a l'IDE NetBeans.

NOM: NetBeans UML Project
 DESENVOLUPADOR: Sun Microsystems
 ÚLTIMA VERSIÓ: Amb NetBeans 6.0
 LLICÈNCIA: CDDL
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://www.netbeans.org/features/uml/>
 COMPATIBILITAT SO: Multiplataforma

Disposa de generació de codi font a partir de diagrames UML i també permet

actualitzar els diagrames a partir de canvis al codi font, mitjançant enginyeria inversa.

Dóna suport per la creació dels següents tipus de diagrames UML:

- Diagrama d'activitat
- Diagrama de classe
- Diagrama de col·laboració
- Diagrama de components
- Diagrama de desplegament
- Diagrama de seqüències
- Diagrama d'estats
- Diagrama de casos d'ús

Té un conjunt predefinit de patrons de disseny d'EJB 1.1, EJB 2.0 i GoF [24].

ArgoUML

És una aplicació de disseny de diagrames UML escrita en Java.

La versió actual és la 0.24. És una aplicació relativament recent i amb alguns punts febles (estabilitat de la versió, importació/exportació cap a Java, no compleix totalment les especificacions UML...).

NOM: ArgoUML
 DESENVOLUPADOR: Comunitat
 ÚLTIMA VERSIÓ: 0.24 - 12/02/2007
 LLICÈNCIA: BSD License
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://argouml.tigris.org/>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Independent

Està disponible en castellà i és desenvolupada pel mateix equip que desenvolupa Subversion. Disposa d'una versió Java Web Start, una versió que es pot executar directament per Internet sense necessitat d'instal·lació.

Altres eines no lliures per Eclipse

Les eines lliures de disseny per UML no estan a l'altura de les eines comercials, encara els manquen moltes funcionalitats per implementar. Les eines comercials més destacades són:

- Omondo UML
- Borland Together
- eUML2 Studio (amb versió gratuïta)
- IBM Rational Software Modeler

Avaluació

A la taula 5 podem veure les característiques principals d'aquest tipus d'eines:

Taula 5: Comparació sobre eines UML.

	NETBEANS UML	ARGO UML
LLICÈNCIA LLIURE	✓	✓
GRATUÏTA	✓	✓
GENERACIÓ DE CODI FONT	✓	✗
ENGINYERIA INVERSA	✓	✗
PATRONS DE DISSENY	✓	✗
EXPORTACIÓ A FORMATS GRÀFICS	✓	✓

L'eina més completa, a nivell de funcionalitat, és el mòdul UML de Netbeans. En cas de voler una solució encara més completa s'hauria de recórrer a solucions comercials.

IDE

Eclipse Europa 3.3

És una plataforma de *software* de codi obert amb projectes enfocats a la construcció d'eines de desenvolupament mitjançant l'ús de *frameworks* extensibles, eines i *runtimes* per construir, desplegar i manegar

NOM:	Eclipse
DESENVOLUPADOR:	Eclipse Foundation
ÚLTIMA VERSIÓ:	3.3 - 29/06/2007
LLICÈNCIA:	Eclipse Public License
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.eclipse.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

aplicacions al llarg de tot el seu cicle de vida. Inicialment va ser desenvolupat per IBM però posteriorment va ser alliberat a la comunitat amb la creació de l'Eclipse Foundation al 2004.

Aquesta primera definició pot semblar ambigua, però ve a dir que Eclipse no és únicament l'IDE de desenvolupament per Java (concretament anomenat JDT, *Java Development Toolkit*), sinó que és molt més. La plataforma està orientada al desenvolupament d'RCA (Rich Client Application). Un exemple d'RCA és l'IDE, però tenim altres exemples, com el client per BitTorrent Azureus.

Dels projectes Eclipse n'hi ha de recomanats per la programació en Java d'aplicacions webs:

- Visual Editor (VE) – És un GUI visual per disseny d'interfícies Swing/JFC i SWT/RCP.
- Java EE and Web Tools (WTP) – Extèn la plataforma eclipse amb utilitats per desenvolupar J2EE i aplicacions web.
- Test and Performance Tools (TPTP) – Proporciona una plataforma oberta amb potents *frameworks* i serveis que permeten als desenvolupadors de *software* construir eines de test i rendiment i que poden ser fàcilment integrades amb altres eines.
- Reporting Tools (BIRT) – És un sistema de generació de reports que s'integra amb les aplicacions per produir documents en diversos formats, com ara excels o pdf. Aquesta eina s'explica més detalladament a la secció d'eines de *reporting*.
- Data Tools (DTP) – Proporciona eines per accedir, veure i manipular bases de dades.

Des del 2006, la política de versions de la fundació Eclipse ha coordinat tots els seus projectes per fer un llançament de versions simultani. Cada versió inclou la plataforma Eclipse, així com un gran número de projectes. El propòsit d'aquesta mesura és distribuir el *software* amb unes versions i funcionalitats estàtiques, per evitar problemes de compatibilitats i per simplificar el manteniment dels sistemes corporatius. Des d'aquesta organització de versions s'han alliberat:

- Callisto, consistent en 10 projectes de la fundació, alliberat el 30 de Juny del 2006.
- Europa, consistent en 21 projectes de la fundació, alliberat el 29 de Juny del 2007.
- Ganymede, prevista pel 29 o 30 de Juny del 2008.

EasysEclipse

És una distribució empaquetada de la plataforma Eclipse juntament amb una selecció de *plugins*. Amb aquest paquet es pot començar a desenvolupar directament, sense necessitat de descarregar cap altre fitxer ni *plugin*.

NOM:	EasyEclipse
DESENVOLUPADOR:	nexB
ÚLTIMA VERSIÓ:	1.3 - 13/09/2007
LLICÈNCIA:	CDL i EPL
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.easyeclipse.org
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

El principal objectiu és tenir un paquet, amb la plataforma i una sèrie de *plugins*, redistribuïble i fàcilment instal·lable en qualsevol entorn.

EasyEclipse proporciona un paquet per a cadascuna de les àrees amb les que treballa Eclipse:

- Distribucions per Java: EasyEclipse Expert Java, EasyEclipse Desktop Java, EasyEclipse Server Java EasyEclipse Mobile Java i EasyEclipse for Plugins and RCP Apps.
- Distribucions per llenguatges dinàmics: EasyEclipse per LAMP, EasyEclipse per PHP, EasyEclipse per Ruby and Rails, EasyEclipse per Python, EasyEclipse per C i C++.

Per l'entorn de desenvolupament, i donat els requeriments del SER.RI-TIC, la més adequada a les necessitats és la EasyEclipse Server Java, orientada al desenvolupament d'aplicacions Java server-side, JSP's, EJB's i Web Services. Les principals característiques són:

- Inclou un entorn Java JDK 1.5 preinstal·lat.
- Control integrat amb Eclipse per diferents servidors d'aplicacions: Tomcat, JBoss i WebLogic.
- Eines específiques per *frameworks*: Hibernate Tools, Spring IDE, J2EE Tools.
- Eines per treballar amb bases de dades: Eclipse Data Tools, QuantumDB.
- Editor de fitxers amb reconeixement de tot tipus de llenguatges.
- Integració amb control de versions: Subclipse, Eclipse CVS client.

Per concloure, és un sistema molt bo per aconseguir la plataforma instal·lada homogèniament per tot un equip de treball. Proporciona una instal·lació neta i completa.

NetBeans 6.0

És una plataforma lliure, basada en Java, de desenvolupament d'aplicacions d'escriptori, amb la qual es construeix l'entorn de desenvolupament NetBeans IDE. La relació entre l'IDE i la plataforma és la mateixa que la que existeix entre l'Eclipse Platform i l'Eclipse IDE, aquí igualment es poden desenvolupar aplicacions RCP.

NOM:	NetBeans
DESENVOLUPADOR:	Sun Microsystems
ÚLTIMA VERSIÓ:	6.0 - 03/12/2007
LLICÈNCIA:	CDDL i GPL v2
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.netbeans.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Va néixer com un projecte d'un estudiant de Praga i posteriorment va ser comprat per Sun, qui al cap de poc temps (Juny del 2000) li va donar una llicència lliure per fer créixer una comunitat al seu voltant. Ara està integrat en eines completes de Sun, com ara el Sun Java Studio Enterprise o el Sun Java Studio Creator.

Hi ha un projecte o paquet principal, el NetBeans IDE, però s'hi poden afegir els següents paquets de desenvolupament:

- Mobility Pack for CLDC/MIDP o per CDC, per construir aplicacions mòbils amb eines visuals.
- Visual Web & Java EE Development Pack, creació d'aplicacions web i web services mitjançant eines *drag-and-drop*. Està basat en JSF, i permet desenvolupar amb les tecnologies AJAX, CSS, JSP, JavaScript, JSF, Struts 1 i EJBs. També inclou un explorador de bases de dades.
- UML Modeling, eines per modelar i desenvolupar aplicacions SOA. S'ha comentat anteriorment a la secció d'UML.
- C/C++ Pack, creació d'aplicacions en C o C++.
- Ruby Pack, creació d'aplicacions en Ruby, JRuby i Rails.

Algunes característiques destacades de la versió actual són:

- Potent *profiler* per analitzar el rendiment de les aplicacions desenvolupades.
- Sistema de projectes basat en Ant. El *plugin* Mevenide permet Maven 2.
- Integració amb sistemes de control de versions (CVS i Subversion).
- Ajuda a la generació de codi "*refactoring*".
- Eines per treballar amb XML.

Aptana

És un entorn de desenvolupament lliure dirigit a les aplicacions web escrites amb AJAX/JavaScript.

Les seves principals característiques són:

NOM: Aptana
 DESENVOLUPADOR: Aptana Inc.
 ÚLTIMA VERSIÓ: 1.0 - 25/10/2007
 LLICÈNCIA: Aptana Public License 1.0 i GPL
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://www.aptana.com/>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Independent

- Ajuda en l'edició de JavaScript, HTML i CSS.
- *Debugger* de JavaScript, on podem establir *breakpoints*.
- Suporta un gran nombre de llibreries AJAX: Prototype, Scriptaculous, Dojo, Mochikit, Yahoo UI, Aflax, JQuery i Rico.
- Està basat completament sobre Eclipse, és un conjunt de *plugins*.

Està mantingut per un gurú de la web 2.0, Paul Colton, amb un petit equip de desenvolupament, dintre d'una companyia fundada al 2005.

Gràcies al sistema de perspectives d'Eclipse és molt fàcil integrar-lo a la nostra instal·lació, complementant les eines de que ja disposem. Té una gran col·lecció d'*screencasts* a la pàgina d'aptana.tv [25], que facilitaràn l'adaptació a l'eina.

Avaluació

A la taula 6 podem veure les característiques principals d'aquest tipus d'eines. A la columna Eclipse s'està representant l'Eclipse 3.3 amb l'Aptana.

Taula 6: Comparació de les característiques dels IDEs.

	ECLIPSE 3.3	NETBEANS 6.0
LLICÈNCIA LLIURE	✓	✓
GRATUÏTA	✓	✓
GENERACIÓ DE CODI FONT	✓	✓
MONITOR DE CONEXIONS	✓	✓
EDITOR HTML, CSS, XML, JAVASCRIPT	✓	✓
UTILITATS XML/DTD/XSD	✓	✓
EDITORS GRÀFICS WSDL/XSD	✓	✓
EXPLORADOR DE WEB SERVICES	✓	✓
EXECUCIÓ SOBRE SERVIDORS	✓	✓

	ECLIPSE 3.3	NETBEANS 6.0
DEBUG SOBRE SERVIDORS	✓	✓
PROFILING SOBRE SERVIDORS	✓	✓
SUPORT HIBERNATE, EJB, JPA	✓	✓
SUPORT JSP, SERVLET	✓	✓
SUPORT PER SPRING, STRUTS 2, JSF	✓	✓
SUPORT PER AJAX	✓	✓
EINES PER ORIGENS DE DADES	✓	✓
DEBUG DE JAVASCRIPT	✓	✓
SUPORT PER SUBVERSION	✓	✓
SUPORT PER ANT I MAVEN	✓	✓

L'elecció entre els IDEs és la més difícil de realitzar. No hi ha característiques funcionals que facin decantar la balança cap a un o altre. Hi ha aspectes amb que algun destaca, però no són suficients com per fer-lo declarar vencedor.

A la taula 7 es mostra una selecció de criteris per ajudar en l'elecció:

Taula 7: Comparació d'altres característiques dels IDEs.

	ECLIPSE 3.3	NETBEANS
FACILITAT D'ÚS		✓
POTÈNCIA I PERSONALITZACIÓ AVANÇADA	✓	
SUPORT DE LA COMUNITAT	✓	
NOMBRE DE PLUGINS	✓	
ÚS MAJORITARI EN ENTORNS PROFESSIONALS	✓	
DESENVOLUPAMENT D'APLICACIONS D'ESCRITORI		✓

Amb aquests criteris, l'eina elegida per l'entorn de desenvolupament és l'Eclipse.

Navegadors

Un dels problemes dels desenvolupadors d'avui dia és la comprovació de les aplicacions web en tots els navegadors que els usuaris poden tenir instal·lats. Un problema en especial és el no compliment dels estàndards CSS de l'Internet Explorer, motiu pel que en ocasions s'ha de programar específicament el comportament de cada navegador.

Mozilla Firefox

El Firefox, juntament amb Linux i OpenOffice, és el buc insígnia del programari lliure. És el segon navegador més utilitzat, amb una quota de gairebé el 17% [26], per darrera d'Internet Explorer.

NOM:	Firefox
DESENVOLUPADOR:	Mozilla Foundation
ÚLTIMA VERSIÓ:	2.0.0.11 - 30/11/2007
LLICÈNCIA:	Mozilla EULA i GPL
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.mozilla.com/firefox/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Quan es parla del Firefox s'ha de parlar dels complements desenvolupats per tercers, més que explicar les seves funcionalitats. Hi ha complements per gairebé totes les tasques possibles, però on té especial rellevància és als destinats al desenvolupament. En destacarem els següents:

- Firebug: permet *debuggar*, editar i modificar en temps real els elements CSS, HTML, DOM i JavaScript, veure en temps real l'activitat de la xarxa i *logging* de funcions JavaScript, entre d'altres. Un *must-have*.
- Web Developer Toolbar: permet incorporar una nova barra d'eines amb multitud de millores per ajudar el desenvolupament d'aplicacions web.
- Colorzilla: ajuda a treballar amb colors, permetent llegir valors exactes de color, mesurar distàncies i elecció de colors, entre d'altres.
- LinkChecker: permet veure mitjançant colors l'estat dels hipervincles d'una pàgina web, per saber en un cop d'ull s'hi n'hi ha algun incorrecte.
- LiveHTTPHeaders: permet veure en temps real les capçaleres HTTP i *cookies* enviades pels servidors. Pot ser molt útil per *debuggar*.
- MeasureIt: dibuixa un regle al voltant dels elements seleccionats, per mesurar les mides i els espais.
- View formatted source: permet veure el codi font de la pàgina i directament el codi font dels estils CSS de forma clara i ordenada.

Internet Explorer

És el navegador més utilitzat des de 1999, amb una quota màxima del 95% d'ús al 2002. Des de llavors el Firefox està sent el seu gran competidor.

NOM:	Internet Explorer
DESENVOLUPADOR:	Microsoft
ÚLTIMA VERSIÓ:	7.0.5730.13 - 2007
LLICÈNCIA:	Propietària
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://microsoft.com/ie/
COMPATIBILITAT SO:	Windows
COMPATIBILITAT BD:	Independent

Està integrat amb el sistema operatiu Windows i ve instal·lat per defecte, per la qual cosa tots els usuaris Windows el tenen instal·lat. No hi ha versions suportades per cap altre sistema operatiu.

Com s'ha comentat anteriorment, té poca consideració pels estàndards i molts problemes de seguretat. Tot i això és el més emprat per la majoria d'usuaris, per tant és necessari realitzar les proves pertinents per validar el seu correcte funcionament.

Cal destacar l'eina de debug HttpWatch, una eina que permet veure i analitzar el tràfic HTTP i HTTPS de les webs que s'estan mostrant mitjançant l'Explorer. Té llicència privativa, però hi ha disponible una versió gratuïta.

Safari

És el navegador instal·lat per defecte als sistemes operatius Mac OS X, i és el tercer més utilitzat.

NOM:	Safari
DESENVOLUPADOR:	Apple Inc.
ÚLTIMA VERSIÓ:	3.0.4 - 14/11/2007
LLICÈNCIA:	Propietària
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.apple.com/safari/
COMPATIBILITAT SO:	Windows i Mac OS X
COMPATIBILITAT BD:	Independent

Al 2007 es va treure una versió per Windows, cosa que ha facilitat la verificació de la compatibilitat per part dels desenvolupadors.

Avaluació

Per comprovar el funcionament de les aplicacions web es recomana la instal·lació de tots els navegadors, i fer un número mínim de proves manuals.

Per realitzar el desenvolupament de les aplicacions i *debuggar*-les (CSS, JavaScript, HTML...) es recomana vivament la instal·lació del Firefox i dels complementos explicats.

Servidor d'aplicacions per desenvolupament

Les aplicacions han de córrer damunt d'un servidor d'aplicacions, un contenidor que pugui executar el codi desenvolupat dintre d'un entorn controlat i segur. Per millorar el rendiment de treball dels desenvolupadors és necessari que aquests tinguin instal·lat un servidor propi a la seva màquina per no dependre de terceres persones que facin alentir els desenvolupaments.

Per evitar confusions s'ha de distingir entre dos conceptes:

- Servidor web: és una aplicació responsable d'acceptar peticions HTTP des d'altres clients, els navegadors web, i respondre a les peticions amb respostes HTTP, transmetent les dades sol·licitades. Les dades poden ser de dos tipus: estàtiques, llegides directament del sistema de fitxers, o dinàmiques, generades mitjançant algun llenguatge d'*scripting* o API cridada pel servidor web.
- Servidor d'aplicacions J2EE: és un motor de *software* que proporciona serveis d'aplicació a altres clients. Generalment gestiona la major part de lògica de negoci i d'accés a dades de l'aplicació. A la plataforma J2EE, el servidor d'aplicacions fa de contenidor dels components que constitueixen les aplicacions. Aquests components són els Servlets, els JSP i els EJBs, que permeten dissenyar aplicacions multi-capa (capa d'aplicació, UI, lògica de negoci, gestió de sessions i bases de dades).

Sun Java System Application Server

És un servidor d'aplicacions que fa de plataforma per l'execució de totes les tecnologies J2EE. S'anomena també contenidor d'EJBs.

NOM:	Sun Java System Application Server
DESENVOLUPADOR:	Sun Microsystems
ÚLTIMA VERSIÓ:	9.1
LLICÈNCIA:	Propietària
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.sun.com/software/products/appsrvr/
COMPATIBILITAT SO:	Multiplataforma

A partir de la versió 9, el SJSAS és la versió comercial del projecte lliure GlassFish.

GlassFish

És un servidor d'aplicacions J2EE (JSP, Servlets i EJBs) dissenyat per Sun (qui alhora el ven amb el nom de SJSAS).

El seu origen es troba en una donació de

codi de Sun i d'Oracle Corporation. El contenidor de Servlets està basat en Tomcat.

Compleix amb l'últim estàndard J2EE 5, i està suportat per treballar amb la gran majoria d'IDEs. Té una compatibilitat gairebé del 100% amb el SJSAS, cosa que el fa ideal per la instal·lació a l'entorn dels desenvolupadors.

Les diferències concretes entre SJSAS i GlassFish [27] són:

- GlassFish no té instal·lador gràfic, s'instal·la a partir d'un JAR.
- GlassFish no té suport oficial.
- A diferència de GlassFish, SJSAS disposa de:
 - Magatzem de certificats NSS de Mozilla
 - Drivers JDBC Data Direct
 - HADB (High Availability Database)

Tomcat

És un servidor web o contenidor web desenvolupat a partir de les especificacions de Sun. Inicialment, era un producte intern d'aquesta empresa, però a partir de la versió 3.0 (la primera en sortir públicament) se li va donar nom propi i una llicència lliure.

Té suport per executar certes parts de l'estàndard J2EE, els JSPs i els Servlets, però no és un contenidor d'aplicacions J2EE complet, ja que no conté tots els components necessaris per executar aplicacions EJB. Incorpora el seu propi servidor HTTP per respondre a les peticions.

NOM:	GlassFish
DESENVOLUPADOR:	Sun Microsystems
ÚLTIMA VERSIÓ:	V2 - 17/09/2007
LLICÈNCIA:	CDDL i GPL
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://glassfish.java.net/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

NOM:	Apache Tomcat
DESENVOLUPADOR:	Apache Software Foundation
ÚLTIMA VERSIÓ:	6.0.14 - 13/08/2007
LLICÈNCIA:	Apache 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://tomcat.apache.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Sun Java System Web Server

És un servidor web semblant al Tomcat, i que a l'igual que aquest, dóna suport per l'execució de JSPs i Servlets. Tampoc té suport per l'execució d'EJBs, ni RMI.

NOM: Sun Java System Web Server
 DESENVOLUPADOR: Sun Microsystems
 ÚLTIMA VERSIÓ: 7.0 - 22/01/2007
 LLICÈNCIA: Propietària
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://www.sun.com/webserver>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Independent

És la nova versió del servidor Sun One Web Server.

Avaluació

A la taula 8 podem veure les característiques principals d'aquest tipus d'eines:

Taula 8: Comparació de les característiques dels servidors d'aplicacions.

	TOMCAT	SJSWS	SJSAS	GLASSFISH V2
SERVIDOR WEB	✓	✓	✓	✓
SERVLETS+JSPS	✓	✓	✓	✓
J2EE (EJB, RMI, ...)	✗	✗	✓	✓
SUPORT NETBEANS	✓	✓	✓	✓
SUPORT ECLIPSE	✓	✗	✗	✓
GRATUÏT	✓	✓	✓	✓
LLICÈNCIA LLIURE	✓	✗	✗	✓
SUN-COMPATIBLE	✗	✓	✓	✓

El millor servidor per a les necessitats dels desenvolupadors del SER.RI-TIC és GlassFish V2. Amb ell s'assegura la compatibilitat a la fase de producció i és possible facilitar el desenvolupament amb qualsevol IDE.

Utilitats

Les eines explicades fins ara són les que s'utilitzaran al dia a dia del SER.RI-TIC. El conjunt d'eines que es comentaran a continuació són d'un ús esporàdic, però són utilitats interessants de conèixer.

No se'n farà una extensa explicació, només es comentarà per sobre la seva utilitat, ja que l'objectiu no és introduir-les dintre de la solució d'entorn de desenvolupament escollit. Totes tenen alguna llicència lliure i si no s'indica el contrari, són multiplataforma.

Taula 9: Utilitats lliures diverses.

APLICACIÓ	DESCRIPCIÓ I PÀGINA WEB
Analitzador de xarxa Wireshark (Ethereal)	http://www.wireshark.org És una aplicació per analitzar i filtrar els paquets a nivell de xarxa. Pot ser útil per trobar errors invisibles a nivell d'aplicació, quan el problema és a nivell de protocol. També s'utilitza com a <i>sniffer</i> .
Creació de PDF PDFCreator	http://www.pdfforge.org/products/pdfcreator És una aplicació per convertir documents al format PDF. Quan s'instal·la crea una nova impressora, tots els documents enviats són transformats. Només per Windows.
Millores consola Console	http://sourceforge.net/projects/console/ Eina lliure amb una consola totalment diferent de l'usual.
Microsoft PowerShell	http://www.microsoft.com/technet/scriptcenter/hubs/msh.mspx Coneguda com a Monad, la consola dels futurs sistemes operatius.
puttycyg	http://web.gccaz.edu/~medgar/puttycyg/ La millor opció és instal·lar Cygwin sobre Windos i connectar-se via el terminal de PuTTY.
Editor de CSS Simple CSS	http://www.hostm.com/css/ És un GUI per editar els fitxers CSS de les pàgines web.
Editor d'HTML senzill NVU, CompoZer	http://www.nvu.com/ http://www.kompozer.net/ NVU és el projecte original, i CompoZer és un fork d'NVU. Permeten crear pàgines web de forma semblant a DreamWeaver o FrontPage.
Editor d'imatges senzill Paint.NET	http://www.getpaint.net/ És un editor gràfic ideat per substituir l'MS Paint que ve per defecte amb Windows. Té suport per capes, <i>blending</i> , transparències i <i>plugins</i> . Només disponible per aquesta plataforma.
Editor d'imatges potent GIMP	http://www.gimp.org/ És l'editor gràfic lliure més potent, competint directament amb l'Adobe Photoshop.
Editor de PDF Scribus	http://www.scribus.net/ És una eina de publicació, com d'Adobe PageMaker o QuarkXPress.

APLICACIÓ	DESCRIPCIÓ I PÀGINA WEB
Editor de text/codi Notepad++	http://notepad-plus.sourceforge.net/ És un molt bon substitut per l'UltraEdit, ja que també té suport per un gran nombre de llenguatges, expressions regulars, divisió de pantalla, pestanyes i un <i>plugin</i> per FTP.
gVim	http://www.vim.org/ El millor editor de text, sense cap dubte. Díficil d'aprendre a utilitzar.
Escriptoris virtuals Virtual Dimension	http://virt-dimension.sourceforge.net Permet tenir diversos escriptoris sobre plataforma Windows.
FTP, SFTP i SCP WinSCP	http://winscp.net/ Utilitat per transferir fitxers de forma segura mitjançant protocol SCP o SFTP, tot i que també suporta FTP. Només per Windows.
GNU/Linux a Windows Cygwin	http://www.cygwin.com/ Conjunt d'eines lliures que permeten treballar amb entorns Windows de forma semblant a com treballen els entorns UNIX. Obviament, només per Windows.
UnxUtils	http://sourceforge.net/projects/unxutils És un port d'eines GNU d'Unix a plataforma Windows, però només a nivell d'executable.
Missatgeria instantània Pidgin	http://pidgin.im/ Originalment anomenat Gaim, és una aplicació de missatgeria instantània multi-protocol. La comunicació instantània també beneficia els grups de treball, evitant reunions innecessàries.
Notes adhesives WikidPad	http://www.jhorman.org/wikidPad/ És una eina de notes adhesives d'escriptori que funcionen com si fos un <i>wiki</i> , podent enllaçar notes entre elles.
SSH PuTTY	http://www.chiark.greenend.org.uk/~sgtatham/putty/ És un emulador de terminal que pot ser utilitzat com a client SSH, telnet i rlogin. Té suport per autenticació mitjançant SmartCard.
Suite ofimàtica OpenOffice.org	http://www.openoffice.org/ Suite ofimàtica per substituir l'Office.
Múltiples ordinadors Synergy	http://synergy2.sourceforge.net/ És una aplicació per controlar diversos ordinadors amb únicament un teclat i ratolí, com un KVM però sense vídeo. Es configuren les posicions dels ordinadors i a l'arrastrar el ratolí fora dels límits de la pantalla automàticament passa a l'altre ordinador. Per ser segur necessita treballar creant un túnel SSH.
Visor de logs Tail for Win32	http://tailforwin32.sourceforge.net/ És una versió gràfica de l'eina tail de UNIX. Actualitza contínuament el visor per reflexar els canvis que s'han realitzar als fitxers. Sobretot útil per veure <i>logs</i> de servidors en temps real.

Tecnologies

Accés a dades - persistència

La forma més senzilla de realitzar l'accés a la base de dades des de Java és utilitzant el *driver* JDBC per fer crides SQL directament a la base de dades. Aquest, però, no és la millor tàctica, ja que té associats alguns de problemes:

- És fàcil que modificacions de la base de dades donin error en temps d'execució, ja que les consultes no s'hauran actualitzat.
- Generalment no s'utilitza el patró DAO [28] [29] i es fan crides directament.

En qualsevol cas, si és necessari utilitzar JDBC és recomanable llegir documents sobre com crear eficientment les consultes [30].

El més adequat és buscar una llibreria ORM (Object Relational Mapping). Aquestes llibreries permeten comunicar informació entre sistemes de dades incompatibles, al nostre escenari, l'orientació a objectes de Java amb bases de dades relacionals. Els avantatges d'utilitzar una llibreria d'aquest tipus són:

- Independència de la base de dades. Les llibreries internament utilitzen els drivers JDBC adequats segons la configuració.
- Permeten una programació totalment orientada a objectes, fins i tot amb les dades.
- Facilita l'etapa de desenvolupament, ja que el codi es simplifica. Per exemple, per crear un nou registre a la taula n'hi ha prou en crear un objecte.
- Facilita la lectura del codi, ja que no és necessari conèixer l'estructura interna de la base de dades per començar a treballar-hi.

El canvi conceptual és substancial [31], però una vegada superat el canvi a la nova metodologia s'aprecien les millores de codi i el rendiment de les aplicacions.

És necessari també fer un apunt sobre els *Plain Old Java Objects* (POJO). Quan es parla de POJOs es refereix a classes senzilles de Java, que no implementen cap interfície ni hereten de les classes internes d'un *framework*. És un terme que va sorgir per contrarestar la complexitat dels EJB, anteriors a la versió 3.0, i per recordar que com més senzill és el disseny, millor.

Hibernate

És l'ORM més utilitzat a l'entorn Java, gairebé sent un *estàndard de facto*.

La seva principal característica és el mapatge des de classes Java a taules de la

NOM:	Hibernate
DESENVOLUPADOR:	Red Hat
ÚLTIMA VERSIÓ:	3.2.5.ga - 31/07/2007
LLICÈNCIA:	LGPL
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.hibernate.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Multi base de dades

base de dades (i des de tipus de dades Java a tipus de dades SQL). També proporciona facilitats a l'hora de fer consultes i recuperar dades. Genera crides SQL i allibera al programador del tractament manual dels *resultsets* i de les conversions a objectes, mantenint l'aplicació portable a qualsevol base de dades SQL.

Proporciona un sistema de persistència transparent pels *Plain Old Java Objects* (POJO). També proporciona un sistema de comprovació de dades (*dirty checking*) que permet controlar quan es fan les actualitzacions de dades per evitar escriptures innecessàries.

La llibreria està dividida en diferents mòduls, dels quals se n'han de destacar:

- Annotations: dóna suport per definir les metadates del mapatge de dades directament sobre les classes POJO enlloc de fer-ho en fitxers externs XML.
- Validator: validació amb anotacions de restriccions de bases de dades.
- Search: permet utilitzar cerques mitjançant el motor Apache Lucene sobre les entitats persistents manegades per Hibernate.

Per definir els mapatges de persistència de les dades, existeixen 4 opcions:

- XML metadata: és la forma més popular de crear els mappings. Cada classe (amb les seves propietats) està representada dintre del fitxer XML de forma correcta segons el corresponent esquema DTD.
- Annotations metadata: utilitzant les Java Annotations podem definir el mapatge directament al codi font de les classes amb persistència, els POJO. S'afegeixen a la configuració d'Hibernate en temps d'execució.
- XDoclet metadata: és un intent d'aconseguir els mateixos resultats que amb les annotations abans de Java 5, mitjançant tasques Ant.
- Manipulació de metadata mitjançant codi: Hibernate proporciona una API per manipular els mapatges abans de la instanciació del SessionFactory.

Enterprise JavaBean (EJB)

És un component de l'arquitectura de servidor que s'utilitza per construir aplicacions corporatives. És una de les moltes APIs de J2EE (Java Platform Enterprise Edition).

NOM:	Enterprise JavaBean (EJB)
DESENVOLUPADOR:	Sun Microsystems
ÚLTIMA VERSIÓ:	3.0 - 02/05/2006
LLICÈNCIA:	Privativa
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://java.sun.com/products/ejb
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Multi base de dades

Intenta proporcionar una forma estàndard d'implementar la lògica de negoci que típicament s'ha de desenvolupar per a les aplicacions corporatives. Ha estat dissenyada per treballar amb conceptes avançats com ara persistència, integritat transaccional, comunicació remota d'objectes, seguretat, concurrència, events (usant JMS), serveis de directori (JNDI), RMI-IIOP i temes de Web Services, entre d'altres.

Han sorgit molts detractors d'aquesta tecnologia, ja que la complexitat de desenvolupament i el baix rendiment han marcat els seus inicis. Posteriorment s'ha actualitzat per no tenir un aprenentatge contra-intuïtiu, tot i que la corba d'aprenentatge continua tenint molta pendent. Ha creat un moviment reivindicatiu que ha generat tecnologies lleugeres: Hibernate per la persistència i l'Spring Framework per la lògica de negoci.

La versió 3.0 ha estat un gir radical respecte les versions anteriors, seguint els paradigmes de simplicitat d'Hibernate i Spring. Ha passat a utilitzar POJOs (com Spring) i també ha canviat la persistència basada en Entity Beans de la 2.x per la persistència de Java Persistence API [32], amb ús d'annotations (com Hibernate).

Hi ha tres tipus de *beans*:

- Session Beans:
 - Stateful: són objectes distribuïts que tenen un estat conegut. Tenen persistència.
 - Stateless: són objectes distribuïts que no tenen estat, permetent accessos concurrents al *bean*. Necessiten menys recursos que els Stateful.
- Message Driven Beans: introduïts a partir de la versió 2.0, representen la integració de Java Message Services (JMS) amb els EJBs, per crear un nou tipus de *bean* que pot gestionar missatges JMS asíncrons. S'utilitzen per implementar funcionalitats que no requereixen una resposta immediata.
- Entity Beans: existeixen a l'especificació EJB 3.0 només per compatibilitat, ja que han estat substituïts per la Java Persistence API. N'hi ha dos tipus, el CMP (Container Managed Persistence) i el BMP (Bean Managed Persistence).

Avaluació

Com a introducció, explicar que Hibernate i EJB no són directament competidors. Hibernate és una tecnologia de persistència, i EJB és una tecnologia per estructurar la capa de negoci. Estrictament parlant, Hibernate necessita Spring per competir amb EJB i JPA.

De totes formes, la comparativa entre EJB i Hibernate és insalvable, ja que EJB va ser la primera tecnologia en implementar un cert tipus d'ORM, amb els seus Entity Beans.

A la taula 10 podem veure les característiques principals d'aquest tipus d'eines:

Taula 10: Comparació de tecnologies de persistència.

	EJB	HIBERNATE
LLICÈNCIA LLIURE	✗	✓
GRATUÏTA	✓	✓
ESTÀNDARD OBERT	✓	✗
MAPATGE AMB ANNOTACIONS ALS POJOS	✓	✓
MOTOR DE CERCA	✗	✓
INVOCACIÓ D'OBJECTES REMOTS	✓	✓

Ens podem trobar en dos escenaris diferents:

- Per aplicacions senzilles sense necessitat de capacitats avançades l'elecció és Spring+Hibernate. El motiu és senzill, en aquest cas no es necessita tota la potència d'EJB. Aquesta elecció també serà prioritària per aplicacions web que vulguin utilitzar el motor de cerca Lucene amb Hibernate.
- Per aplicacions que necessitin una arquitectura complexa, com ara l'enllaç a un ESB⁵, l'elecció serà els EJB+JPA.

⁵ *Enterprise Service Bus*, un tipus d'arquitectura complexa amb comunicació asíncrona.

Accés a LDAP

Java Naming and Directory Interface (JNDI)

És una API per treballar amb serveis de directori, permetent als clients descobrir i buscar informació i objectes mitjançant un nom.

NOM: JNDI
DESENVOLUPADOR: Sun Microsystems
ÚLTIMA VERSIÓ: 1.2
LICÈNCIA: GPL v2
COST LICÈNCIA: Gratuïta
PÀGINA WEB: <http://java.sun.com/products/jndi/>
COMPATIBILITAT SO: Multiplataforma
COMPATIBILITAT BD: Independent

Forma part del Java 2 Platform Standard Edition (J2SE), tot i ser una API orientada a serveis corporatius i que s'utilitza a les APIs d'RMI i J2EE per buscar objectes a la xarxa.

L'API de JNDI proporciona:

- Un mecanisme per enllaçar un objecte a un nom.
- Una interfície de cerca als directoris que permet realitzar consultes.
- Una interfície d'events que permet als clients determinar quan les entrades del directori han estat modificades.
- Extensions LDAP per usar les capacitats addicionals del servei LDAP.

Permet connectar-se pràcticament a qualsevol tipus de servei de directori o de noms:

- LDAP
- DNS
- NIS
- RMI
- Servei de noms CORBA

Actualment, al SER.RI-TIC s'utilitza per realitzar les autenticacions dels usuaris a les diverses aplicacions web que validen amb l'LDAP.

Eines de *build*

Les eines encarregades d'aquesta tasca són les responsables de transformar el codi font en una aplicació executable usable per tots els usuaris. En el cas concret de les aplicacions web, el procés de muntatge genera un fitxer compactat per desplegar l'aplicació al servidor d'aplicacions, sovint amb extensió .war (Web Archive) o .ear (Enterprise Archive).

Desenvolupar el codi font és relativament directe, fins que la mida del codi font comença a créixer. En aquest punt, comencen a sorgir dependències entre paquets o diferents projectes i la tasca manual de muntatge es complica considerablement. Per poder gestionar aquestes dependències, que poden ser explícites o implícites (dependències de les dependències), necessitem una eina que ens ajudi.

Les tasques necessàries que ha de realitzar l'eina de muntatge segons el canvi dels fitxers del codi font són les que es presenten a la taula 11:

Taula 11: Conseqüències del canvi de codi font per l'eina de muntatge.

TIPUS DE CANVI	QUÈ HA DE FER L'EINA DE MUNTATGE
Afegir nous fitxers	El programador ha d'actualitzar el fitxer de descripció de muntatge.
Modificació de fitxers	Reconstruir el fitxer.
Modificació d'un fitxer sobre el que es depèn	Reconstruir tots els fitxers afectats.
Nova dependència d'un fitxer existent	Reconstruir l'arbre de dependències i comprovar que la dependència sigui accessible.
Nova dependència d'un nou fitxer	Reconstruir l'arbre de dependències i analitzar si la nova dependència és d'un fitxer existent o és sobre un fitxer que es crearà durant el procés de muntatge.
Abandonar una dependència	Reconstruir l'arbre de dependències, verificar si l'antic fitxer causa problemes i si ha de ser eliminat.
Nova dependència d'un fitxer generat	Assegurar que l'arbre de dependències genera el fitxer abans que sigui necessari.
Nova dependència d'un fitxer generat però amb canvis al codi font	Regenerar tots els fitxers necessaris.

Les diferents etapes d'un procés de muntatge d'una aplicació són:

- Definició dels objectius: què es desitja construir? Es pot diferenciar entre muntatges complets o parcials.

- Lectura del fitxer de muntatge: aconseguir informació explícita sobre el procés a realitzar. Aquesta informació estarà emmagatzemada al fitxer de configuració del projecte.
- Configuració: l'eina de muntatge aconsegueix informació sobre l'entorn d'execució, podent modificar el procés segons si s'executa, per exemple, sobre plataforma Windows o GNU/Linux.
- Càlcul de les dependències: escanejar els fitxers de configuració del muntatge i els fitxers amb codi font per saber quines parts depenen de quines altres. També realitza una detecció de dependències circulars.
- Determinar què s'ha de muntar: anotar quins fitxers necessiten ser actualitzats o muntats, segons les dependències calculades. En aquest punt es detecten els errors sobre fitxers necessaris i no existents.
- Generar les comandes per realitzar el muntatge: construir la comanda adequada per actualitzar les parts que ho necessitin.
- Executar les comandes per crear el muntatge: Inici del procés de muntatge segons la informació recopilada anteriorment. També es retornen a l'usuari els errors que s'han detectat en les diferents etapes.

Aquestes etapes poden repetir-se diverses vegades en un procés de muntatge.

Ant

És una eina de muntatge lliure. Forma part de la comunitat de projectes lliures d'Apache. Originàriament va ser dissenyada com una extensió per *make*, però ràpidament va passar a ser l'eina preferida per projectes Java, de forma que qualsevol entorn de desenvolupament té suport per Ant.

NOM:	Ant
DESENVOLUPADOR:	Apache Software Foundation
ÚLTIMA VERSIÓ:	1.7.0 - 19/12/2006
LLICÈNCIA:	Apache Public License 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://ant.apache.org/
COMPATIBILITAT SO:	Multiplataforma basat en Java
COMPATIBILITAT BD:	Independent

Ve preconfigurada amb un gran nombre de tasques llestes per utilitzar. Aquestes quasi 80 tasques donen suport per compilació de Java, comandes de sistema, correu electrònic i comandes CVS entre altres. Conté 60 tasques opcionals més, amb suport per altres eines SCM, *unit testing*, generació de codi font, manteniment de pàgines web i automatització de moltes tasques. Cada tasca permet a l'usuari definir una part del procés de muntatge. A més, un usuari pot definir les seves pròpies tasques.

Els fitxers de muntatge per Ant estan escrits en XML, i generalment s'anomenen *build.xml*. Aquest funcionament sobre XML permet una fàcil transformació de la informació per aconseguir una representació gràfica amb eines com Vizant o AntGraph.

Està construït amb Java, per això pot córrer sobre totes les plataformes que el suporten. Té una bona i extensa documentació [33].

Punts febles:

- Limitacions de l'XML: difícil de mantenir grans fitxers.
- Cadenes de dependència: no gestiona bé dependències encadenades.
- No es poden fer *dry-runs* (una simulació de l'execució del procés de muntatge per verificar que no dona cap error).
- Arrancada lenta, ja que utilitza la JVM.

Ant també disposa d'algunes eines complementàries:

- Antelope (<http://antelope.tigris.org/>) UI per creació i debug de fitxers XML.

Ant també s'utilitza com a base d'entorns d'automatització, que permeten realitzar automàticament el procés de *checkout-build-test-deploy*, amb CruiseControl.

Maven 2

És una eina de l'Apache Foundation, i és l'evolució natural d'Ant. Va sorgir a partir de la necessitat de facilitar el complex sistema de muntatge basat en fitxers d'Ant del projecte (Apache Turbine), i va passar a ser un projecte independent.

NOM:	Maven 2
DESENVOLUPADOR:	Apache Software Foundation
ÚLTIMA VERSIÓ:	2.0.8 - 27/11/2007
LLICÈNCIA:	Apache Public License 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://maven.apache.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Maven 2 treballa a partir d'una unitat d'informació anomenada Project Object Modeler (POM). Els POMs s'utilitzen per descriure els projectes de *software* que han de ser contruïts, les seves dependències i altres components externs. Ve predefinit amb una sèrie d'objectius comuns de construcció per dur a terme tasques definides com ara compilacions de codi i empaquetament de projectes.

Fer notar que la filosofia de treball [34] és a nivell de projecte, no mitjançant la definició d'un conjunt de fitxers. El tractament de les dependències també es manega d'aquesta forma, inclús amb la capacitat de descarregar dependències via xarxa, mitjançant la definició de dipòsits públics o locals.

Els principals objectius que té són:

- Facilitar el procés de muntatge dels projectes. No abstreu completament de conèixer internament el projecte, però sí de conèixer-ne tots els detalls.
- Proporcionar un sistema uniforme de muntatge. Tots els projectes seran gestionats de la mateixa forma i amb la mateixa estructura.
- Proporcionar informació de qualitat sobre el projecte, podent relacionar-se amb altres eines com ara SCMs i tests unitaris.
- Proporcionar una guia de consulta per desenvolupadors, un *best practices*. Utilitzar el cicle de vida dels projectes de Maven 2 té el benefici d'incorporar el flux correcte de treball als projectes, sobretot pels tests unitaris.
- Permet una migració automàtica a noves funcionalitats de futures versions.

Disposa de multitud de complements per integrar-se amb més eines de desenvolupament, com per exemple, servidors d'aplicacions o IDEs. Hi ha disponible una llista de complements [35].

La potència d'aquesta eina fa que funcionalment es solapi amb les eines que porten incorporades els IDEs. L'encaix amb Eclipse és el que es mostra a la figura 3 [36]:

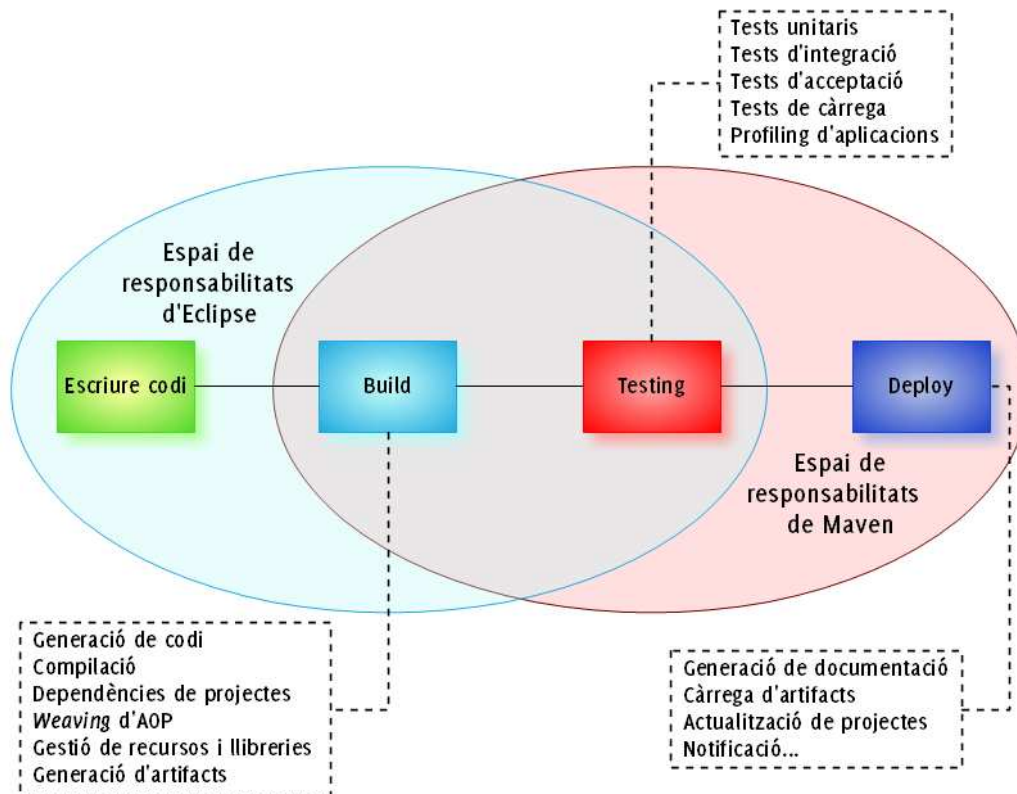


Figura 3: Comparació de les tasques d'Eclipse i Maven 2.

Avaluació

A la taula 12 podem veure les característiques principals d'aquest tipus d'eines:

Taula 12: Comparació sobre eines de construcció de projectes.

	ANT	MAVEN
LLICÈNCIA LLIURE	✓	✓
GRATUÏTA	✓	✓
INTEGRACIÓ AMB IDES	✓	✓
GENERACIÓ DE DOCUMENTACIÓ	✗	✓
SCRIPTS SENZILLS	✗	✓
ORIENTACIÓ A PROJECTES	✗	✓

L'eina més completa a nivell de funcionalitat és Maven. Al ser una evolució d'Ant corregeix molts dels errors d'aquest. Ant tenia un sistema d'scripts de construcció poc eficient i que tendia a créixer desmesuradament. Per aquest motiu Maven es va concebre per modelar un projecte. Per altra banda, la generació de documentació automàtica també és un punt a favor.

Eines de generació d'informes

JReport

És una eina per a la generació d'informes personalitzats a partir de bases de dades relacionals.

Les seves principals característiques són:

NOM: JReport
DESENVOLUPADOR: Jinfonet Software, Inc.
ÚLTIMA VERSIÓ: 8.2 - 19/12/2007
LLICÈNCIA: Privativa
COST LLICÈNCIA: Indeterminada
PÀGINA WEB: <http://www.jinfonet.com/products>
COMPATIBILITAT SO: Multiplataforma
COMPATIBILITAT BD: Multi base de dades

- Arquitectura 100% Java.
- Integració a les aplicacions de forma transparent.
- Model fàcilment integrable dintre dels processos corporatius.
- Permet treballar en múltiples formats: DHTML, HTML, PDF, Excel, XML i RTE.

Està dividit en dos components:

- JReport Designer: és una aplicació 100% Java, basada en components Swing. Permet desenvolupar visualment els informes de forma ràpida. Proporciona components reutilitzables, un centenar d'objectes, taules i gràfics preconstruïts i accés a qualsevol font de dades (inclús múltiples simultàniament dintre d'un mateix informe).
- JReport Enterprise Server: és un servidor central sobre el que es despleguen els informes creats amb el Designer. Una vegada desplegats poden utilitzar-se directament o fer crides remotament des d'aplicacions web J2EE per fer-ne un ús remot.

Va ser la primera eina d'aquest tipus, i gràcies a això va fer-se un gran nínxol de clients. Actualment al SER.RI-TIC s'utilitza la versió 7.2 per generar informes de diverses aplicacions web.

Eclipse BIRT

El projecte *Business Intelligence and Reporting Tools* (BIRT) està integrat dintre la plataforma Eclipse. Proporciona eines de *reporting* i capacitats de *business intelligence* (BI) per aplicacions web de contingut ric, especialment aquelles basades en Java i J2EE.

NOM:	Eclipse BIRT
DESENVOLUPADOR:	Eclipse Foundation
ÚLTIMA VERSIÓ:	2.2.1.1 - 01/11/2007
LLICÈNCIA:	Eclipse Public License
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.eclipse.org/birt
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Multi base de dades

La seva arquitectura té diversos components principals:

- Un dissenyador d'informes visual integrat dintre d'Eclipse.
- Un nucli per generar els informes que pot ser desplegat en un entorn Java.
- Un motor de diagrames.

Aquesta arquitectura es representa a l'esquema de la figura 4:

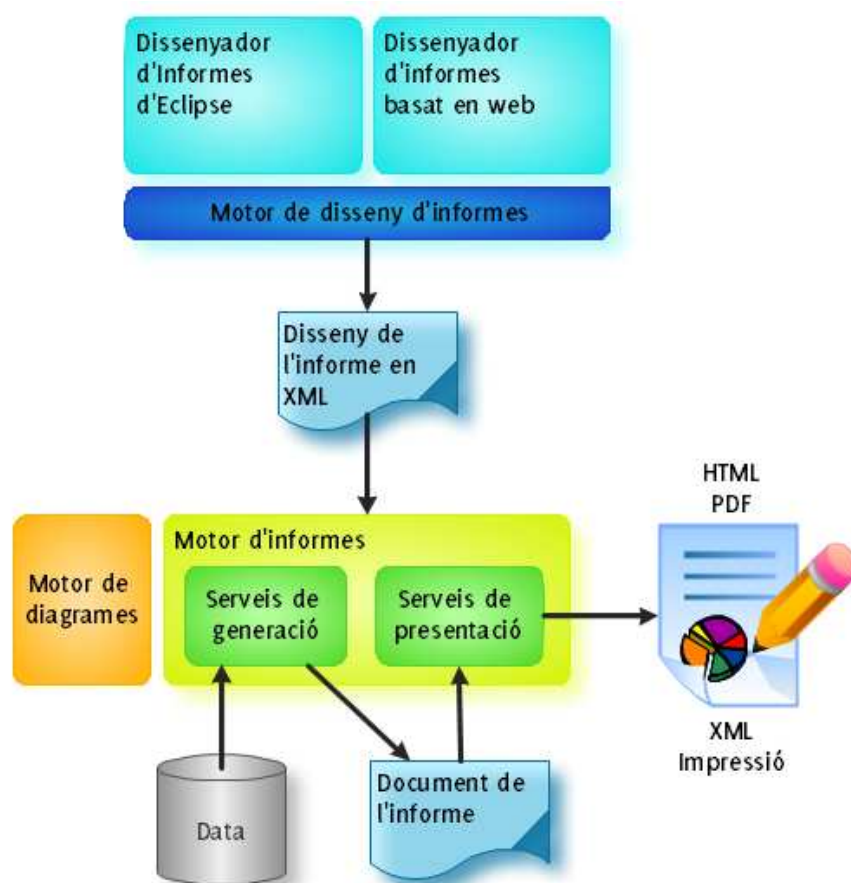


Figura 4: Arquitectura del desenvolupament d'informes amb Eclipse BIRT

Pot crear els següents tipus d'informes:

- Llistes: el tipus més senzill d'informes. Es poden fer agrupacions de dades i càlculs amb les dades numèriques.
- Gràfics: per representar dades numèriques gràficament. Proporciona gràfics de barres, línies, pastís... Poden ser renderitzats amb SVG.
- Taules creuades: per representar la informació en dues dimensions.
- Cartes i documents: automatització de comunicacions.
- Informes composts: es poden combinar els tipus d'informes.

Un informe està format per quatre parts:

- Dades: bases de dades (per SQL o JDBC), Web Services, objectes Java... hi ha moltes fonts de dades per alimentar els informes.
- Transformació de dades: ordenacions, operacions matemàtiques de resum, filtres, percentatges... tot a mida segons les necessitats dels usuaris.
- Lògica de negoci: de vegades no tot es pot solucionar amb ordenacions i es necessari aplicar una lògica concreta, mitjançant JavaScript o classes Java.
- Presentació: un cop les dades estan preparades es poden utilitzar segons les necessitats, amb taules, llistes, documents...

Està desenvolupat bàsicament per empreses [37] (IBM, Innovent Solutions, Actuate...) mitjançant un comitè d'Eclipse. S'ha creat una gran comunitat darrera d'ell [38], ja que està basat en tecnologies lliures.

JasperReports

És una eina de generació d'informes per Java i aplicacions web J2EE. Permet la generació dinàmica de contingut. Compta a un dissenyador gràfic d'informes l'iReport i amb un servidor web interactiu d'execució d'informes, el JasperServer.

NOM: JasperReports
 DESENVOLUPADOR: JasperSoft
 ÚLTIMA VERSIÓ: 2.0.2 - 9/10/2007
 LLICÈNCIA: LGPL
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://jasperforge.org/jasperreports>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Multi base de dades

Les seves principals característiques són:

- Format de sortida a PDF, HTML, Excel, RTF, ODT, CSV i XML
- Pot operar simultàniament informació de múltiples orígens de dades. Les dades poden ser adquirides per mitjà de JDBC, JavaBeans, EJBQL, XML, Hibernate, fitxers de text amb valors separats per comes i, amb una extensió, procediments PL/SQL.
- Possibilitat de creació d'*scriptlets*, utilitzant Java, per manegar les dades.
- Creació de *subreports*.
- El JasperServer té capacitats avançades, com ara la programació horària de reports i la gestió d'usuaris amb permisos sobre els reports.
- El format amb que s'emmagatzemen els informes és XML.

Hi ha disponible suport comercial.

OpenReports

És un entorn d'administració web per l'execució d'informes de diverses tecnologies lliures, com ara JasperReports o Eclipse BIRT.

NOM: OpenReports
 DESENVOLUPADOR: Open Source Software Solutions
 ÚLTIMA VERSIÓ: 3.0.1 - 30/12/2007
 LLICÈNCIA: GPL
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://oreports.com>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Multi base de dades

Les seves principals característiques són:

- Gestió web d'usuaris, grups, informes, paràmetres dels informes i orígens de dades.
- Programació horària d'execució d'informes.
- Múltiples formats de sortida.
- Integració amb aplicacions externes.

Avaluació

A la taula 13 podem veure les característiques principals d'aquest tipus d'eines:

Taula 13: Comparació sobre eines de reporting.

	JREPORT	ECLIPSE BIRT	JASPERREPORTS	OPENREPORTS
LLICÈNCIA LLIURE	✗	✓	✓	✓
GRATUÏTA	✗	✓	✓	✓
COMPONENTS REUSABLES	✓	✓	✓	✗
SORTIDA AMB PDF I EXCEL	✓	✓	✓	✓
EXPLOTACIÓ D'INFORMES DES D'APLICACIONS WEB	✓	✓	✓	✓
EINA DE DISSENY GRÀFIC	✓	✓	✓	✗
CONNEXIÓ A BASE DE DADES ORACLE	✓	✓	✗	✓
CONNEXIÓ A HIBERNATE	✗	✗	✓	✗
CONSOLA D'ADMINISTRACIÓ WEB AMB USUARIS	✓	✗	✓	✓
FORMAT D'EMMAGATZEMAMENT DELS INFORMES OBERT	✗	✓	✓	✗
INTERNACIONALITZACIÓ AMB RESOURCEBUNDLES	✗	✓	✓	✗
INTEGRACIÓ AMB STRUTS 2	✗	✗	✓	✗

Les funcionalitats ofertes són gairebé idèntiques. JReport disposa d'una eina d'administració web, des d'on cada usuari pot executar els seus informes. Eclipse BIRT combinat amb OpenReports també proporciona aquesta funcionalitat, i a més, proporciona independència del proveïdor, gràcies al seu format obert basat en XML. JasperReports ofereix totes aquestes funcionalitats i encara més, com ara integració amb Struts2 o connexió a orígens de dades Hibernate, però no està preparada utilitzar la base de dades d'Oracle.

El treball amb JReport, dintre del SER.RI-TIC, ha estat marcat pels problemes i les dificultats durant el desenvolupament i l'explotació d'informes. D'altra banda, la combinació de funcionalitats de d'Eclipse BIRT i OpenReports, dóna tota la potència i versatilitat del JReport. Per aquest motiu la elecció és Eclipse BIRT.

Esquelet de codi

AppFuse 2

És un projecte lliure que utilitza altres eines lliures sobre la plataforma Java, i que ajuda al desenvolupament ràpid i eficient d'aplicacions web. Segons la seva web, el seu objectiu principal és eliminar el malbaratament de temps que es produeix cada vegada que s'ha de construir una nova aplicació per un client.

NOM: AppFuse
DESENVOLUPADOR: Matt Raible i comunitat
ÚLTIMA VERSIÓ: 2.0.1 - 26/11/2007
LLICÈNCIA: Apache 2.0
COST LLICÈNCIA: Gratuïta
PÀGINA WEB: <http://appfuse.org/>
COMPATIBILITAT SO: Multiplataforma
COMPATIBILITAT BD: Multi base de dades

Eminentment, AppFuse és un *core*, un nucli que fa d'esquelet per nous projectes, de forma similar als esquelets que generen els IDEs més populars, com ara Eclipse, NetBeans o IDEA, mitjançant els assistents per crear un nou projecte web. Però a diferència d'aquestes eines, incorpora un gran nombre de fitxers implementats (concretament, implementa la part de gestió d'usuaris) per continuar el desenvolupament a mode d'exemple. L'esquelet aconseguit genera el que es coneix com una aplicació de tipus "CRUD" ("Create, Read, Update and Delete"). Un exemple d'aplicació CRUD és una aplicació de gestió d'estocs d'un magatzem amb usuaris.

Les principals característiques, segons el patró MVC, són:

- Aspectes del model:
 - Independència de la base de dades.
 - Suporta JPA (balancejat de càrrega per clústers).
 - Té suport integrat per DbUnit, una extensió de JUnit que permet automatitzar la importació i exportació de les dades d'una base de dades cap a un fitxer XML, evitant problemes de corrupció de la base de dades durant el desenvolupament.
- Aspectes de la interfície d'usuari:
 - L'esquelet té ple suport per l'ús de log4j i i18n, portant una pregeneració ja configurada de ResourceBundles.
 - Gestiona la càrrega de fitxers (*upload*) al servidor.
 - AJAX per la UI (Script.aculo.us, SiteMesh, Wufoo Forms).
 - Utilitza un *framework* CSS mitjançant el qual es poden dissenyar temes.
 - Pot afegir automàticament CAPTCHAs als formularis de l'aplicació web.

- Aspectes del control:
 - Disposa de Web Filters configurables per realitzar diverses tasques (urlRewrite...).
 - Suport per autenticació d'usuaris LDAP.
 - Suport per SOA, Web Services i crides remotes amb XFire i Apache CXF.
 - Utilitza AOP per ajudar a mantenir net el codi font.
- Altres aspectes:
 - Suporta JDK5, JSP 2.0 i Servlet 2.4.
 - Suport total pels IDEs Eclipse, IDEA i NetBeans, mitjançant Maven.
 - Combinat amb *plugins* de Maven, pot utilitzar-se com una útil funcionalitat d'inici ràpid, on es poden fer proves de l'aplicació web que s'està desenvolupament sense perdre temps per fer cap *deploy* al servidor. Aquesta funcionalitat únicament és pel servidor d'aplicacions Jetty.
 - Suporta GlassFish.
 - Possibilitat de *testing* en múltiples servidors d'aplicacions i bases de dades, gràcies a Cargo i l'ús de múltiples perfils. Sense suport per GlassFish.
 - Al ser una aplicació lliure és possible agafar el codi font del nucli i modificar-lo per adaptar-lo a les nostres necessitats.
 - Unit Testing i Integration Testing, a nivell de DAO, UI i Unit.

AppFuse et dona tot l'esquelet de l'aplicació web, de forma segmentada, seguint de forma exquisita el patró MVC representat a la figura 5:

- El *core*, que vindria a ser el control, basat en Spring.
- El model, en forma d'*entities* i DAOs.
- La vista, integrant de forma automàtica diversos *frameworks* web:
 - JSF
 - Spring MVC
 - Struts 2
 - Tapestry

Gràcies a aquest disseny les possibilitats d'adaptació i configuració específica per projectes en concret és enorme.

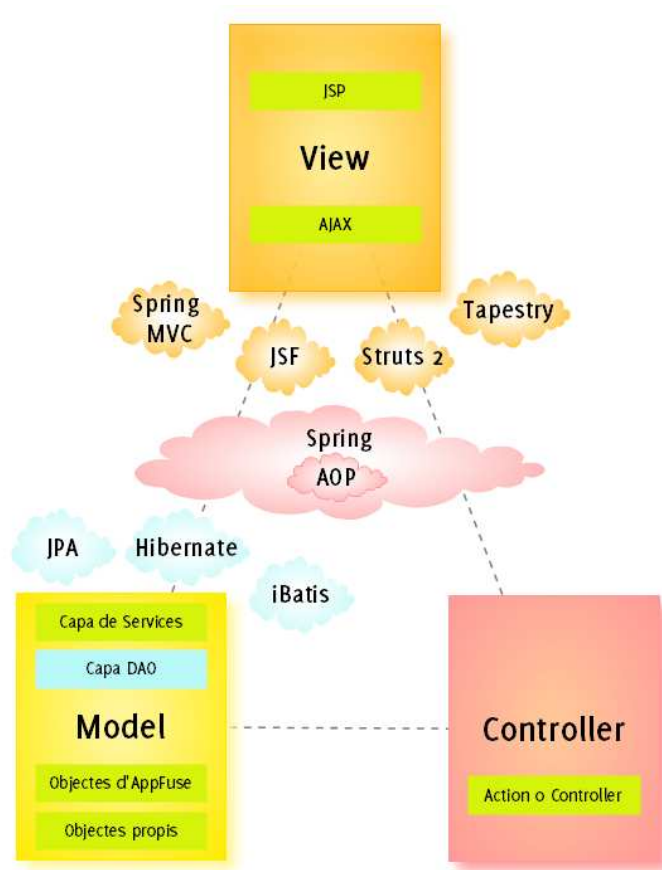


Figura 5: Arquitectura Model-Vista-Control d'AppFuse.

Quan es comença a treballar en AppFuse s'ha de fer una tria inicial del *flavour*, el sabor, el *framework* web que voldrem utilitzar. Segons la nomenclatura Maven 2 s'anomena *archetypes*, però vindrien a ser els diferents tipus de projectes inicials de que disposem per escollir. Hi ha dos tipus d'*archetypes*:

- Basic: és ideal per a la creació de projectes que seran només aplicacions web.
- Modular: és ideal per a la creació de projectes que tindran una lògica de negoci reutilitzable, per exemple, aprofitar el desenvolupament per una aplicació web i un Web Services.

Framework de model

Com s'ha comentat a la secció d'accés a dades – persistència, la unió de Spring i Hibernate és el gran competidor dels EJBs, l'estàndard de lògica de negoci per aplicacions J2EE. En aquesta secció s'explicarà l'altre component del grup, l'Spring Framework.

Spring Framework

Aquest *framework* va néixer al 2001, amb l'objectiu de simplificar el complex model de desenvolupament dels EJBs. El problema era que fins i tot les aplicacions més senzilles, que no necessitaven objectes

NOM:	Spring Framework
DESENVOLUPADOR:	SpringSource
ÚLTIMA VERSIÓ:	2.5.0 - 19/11/2007
LLICÈNCIA:	Apache Public License 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.springframework.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Multi base de dades

distribuïts ni transaccions remotes, havien de desenvolupar-se de forma semblant a com es desenvolupaven les grans arquitectures que si que ho necessitaven.

Els principis amb els que es va crear Spring són quatre [39]:

- J2EE ha de ser senzill d'utilitzar.
- Un bon disseny és més important que la tecnologia d'implementació.
- Els JavaBeans dèbilment acoblats mitjançant interfícies són un bon model.
- Ha de ser fàcil testar el codi font generat, mitjançant POJOs.

Aquests quatre principis són totalment oposats a la filosofia EJB: és difícil d'utilitzar, es basa en la tecnologia i entrega de serveis, els EJB estan fortament acoblats i per testar el codi font has de tenir el contenidor J2EE executant-se.

Spring pot definir-se com un *framework* de contenidor lleuger amb inversió de control⁶ i amb orientació a aspectes⁷:

- La tècnica d'inversió de control utilitzada s'anomena “injecció de dependències”⁸. El contenidor d'Spring injecta les dependències als components, al contrari que passa amb l'EJB, que els components demanen les dependències al contenidor J2EE mitjançant el protocol JNDI.
- L'ús d'AOP permet separar efectivament la capa de negoci dels serveis del sistema, com ara auditoria o gestió de transaccions o *logging*.

⁶ *Inversion Of Control (IoC)*, un patró de disseny que permet alterar el flux d'execució d'una plataforma

⁷ *Aspect-oriented programming (AOP)*, és una tècnica que permet aplicar la separació de conceptes i facilitar la modularització de codi.

⁸ *Dependency Injection (DI)*, un tipus de patró IOC, proporciona les dependències externes a un component en temps d'execució.

Altres característiques d'Spring són:

- Una factoria⁹ abstracta¹⁰ de beans accessible globalment.
- Una capa d'abstracció genèrica per la gestió de transaccions amb bases de dades.
- Suport per la *Java Transaction API* (JTA).
- Integració amb *frameworks* de persistència: Hibernate, iBatis, JDO i JPA.

Internament, Spring està compost per una col·lecció de frameworks més petits que poden treballar de forma independent:

- Contenedor d'IoC, gestiona el cicle de vida de les aplicacions.
- *Framework* AoP, gestiona l'ús d'aspectes.
- *Framework* d'accés a dades, utilitza JDBC i ORM per l'adquisició de dades.
- *Framework* de gestió de transaccions
- *Framework* MVC, per utilitzar a la capa de presentació.
- *Framework* d'accés remot, dona suport per l'ús d'invocació remota d'objectes, com ara RMI, CORBA, Apache Axis, Web Services i Apache CXF (SOAP).
- *Framework* d'autenticació, gestió de l'accés als components. S'anomena Acegi.
- *Framework* de gestió remota, implementació del les JMX (*Java Manager Extensions*).
- *Framework* de missatgeria asíncrona, mitjançant JMS (*Java Messaging Service*)
- Framework de tests, permet l'execució de tests unitaris i d'integració.

A la figura 6 podem veure l'arquitectura dels frameworks d'Spring:

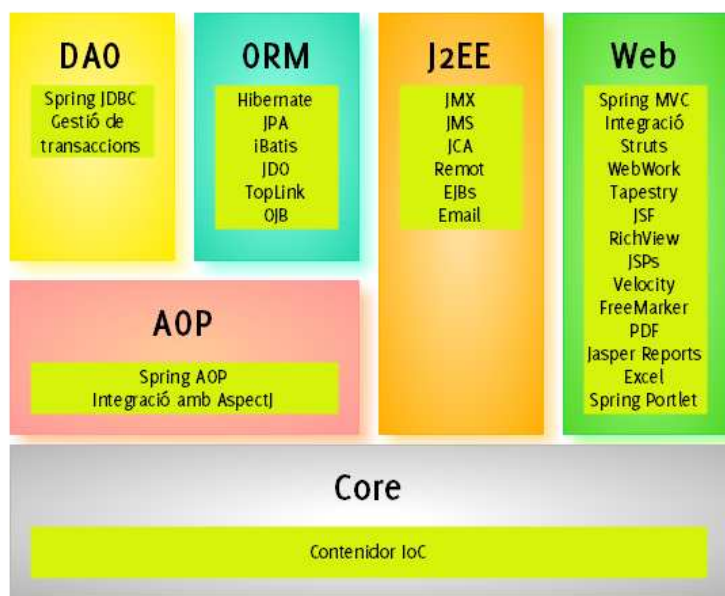


Figura 6: Arquitectura dels frameworks d'Spring.

⁹ *Factory Method*, és un patró de disseny que permet l'abstracció del mètode de creació d'un objecte, ja que no es necessari indicar la classe exacta d'objecte a crear.

¹⁰ *Abstract Factory*, és un patró de disseny que permet l'encapsulació d'un grup de factories individuals que tenen un comportament semblant.

Framework de presentació

Struts 2

És un *framework* extensible per la creació d'aplicacions web corporatives. Està dissenyat per optimitzar el cicle de desenvolupament, des del *building* fins al manteniment d'aplicacions.

NOM:	Struts 2
DESENVOLUPADOR:	Apache Foundation
ÚLTIMA VERSIÓ:	2.0.11 - 23/10/2007
LLICÈNCIA:	Apache Public License 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://struts.apache.org/2.x/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Multi base de dades

La branca 2.x d'Struts és el resultat de la unió de dos frameworks: Struts 1.x i WebWork. Amb aquesta unió de forces s'ha aconseguit simplificar el model de treball d'Struts 1.x.

Les seves principals característiques són:

- Arquitectura basada en *plugins*, que permet personalitzar accions fàcilment.
- Regles de validació desacoblades del codi dels Actions.
- Suport per i18n.
- Conversió de tipus automàtica i transparent entre HTTP i Java.
- Utilitza Spring Core per manejar el cicle de vida amb injecció de dependències.
- Configuració amb Java 5 Annotations.
- Llenguatge OGNL compatible amb JSTL, que permet accedir a les propietats de diversos objectes com si fos un únic JavaBean.
- Facilita els tests dels Actions, sense necessitat d'objectes Mock.
- Resultats en diversos formats: JSP, FreeMarker, Velocity, PDF i JasperReports.
- AJAX, mitjançant el *framework* Dojo¹¹.
- Integració amb Hibernate, Spring, SiteMesh i JSTL.
- Integració amb Maven.
- Eines de debug integrades.

El cicle de vida de les peticions dels clients amb Struts 2 és el que es detalla a continuació:

1. L'usuari envia una petició d'algun recurs al servidor.
2. El FilterDispatcher determina quina és l'Action apropiada segons la petició.
3. S'apliquen Interceptors, encarregats d'executar comportaments (validacions, càrrega de fitxers, *workflow*...) automàtics.
4. Execució de l'Action, executant les operacions necessàries amb la base de dades.
5. Generació del resultat, en base a les dades recuperades.

¹¹ The Dojo Toolkit <http://www.dojotoolkit.org/>

6. Retorn de la petició, a través dels Interceptors i seguint el camí invers al de la petició. En aquest moment també s'aprofita per netejar recursos.
7. Mostrar el resultat a l'usuari, mitjançant el contenidor de Servlets del servidor.

A la figura 7 podem veure un esquema d'aquest comportament.

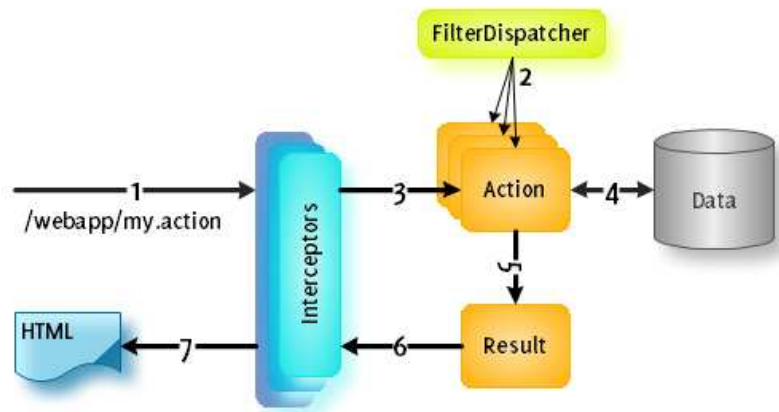


Figura 7: Cicle de vida de les peticions amb Struts 2.

Java Server Faces (JSF)

És un *framework* Java per aplicacions web que simplifica el desenvolupament d'interfícies d'usuari per aplicacions J2EE.

És un estàndard obert desenvolupat per la *Java Comunity Process* (JCP).

NOM:	Java Server Faces (JSF)
DESENVOLUPADOR:	Sun Microsystems
ÚLTIMA VERSIÓ:	1.2_04 P01 - 20/03/2007
LLICÈNCIA:	CDDL i GPL
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://java.sun.com/javaee/javaserverfaces/
COMPATIBILITAT SO:	Multiplataforma

Utilitza un model de desenvolupament enfocat als components i events, a diferència dels tradicionals frameworks MVC que utilitzen un programació orientada a les accions i peticions HTTP. L'ús de components permet al desenvolupador abstraure's del flux de comunicacions de les peticions HTTP, ell només veurà un event, per exemple, (un clic) a un botó. Amb els frameworks de components es pot fer un símil amb el tipus de desenvolupament de Visual Basic o de la plataforma .NET.

Una de les conseqüències d'aquesta abstracció és la sobrecàrrega de controls que ha de tenir aquesta plataforma, aspecte que provoca que sigui pesada respecte altres *frameworks* web.

Les seves principals característiques són:

- Suport integrat d'AJAX.
- Integració amb els *taglibs* JSTL.
- Gestió de la navegació dels usuaris per l'aplicació web. S'encarrega del correcte funcionament dels botons de navegació, la coherència amb diverses finestres obertes, el problema de la duplicació d'events per clics successius...
- Ús de ResourceBundles per la internacionalització de les aplicacions.
- Integració amb *portlets*.
- Arquitectura extensible mitjançant plugins.

Tant NetBeans com Eclipse tenen suport integrat pel desenvolupament d'aplicacions amb aquesta tecnologia.

La documentació de JSF és abundant, des de tutorials [40] fins a llibres [41][42].

Avaluació

A la taula 14 podem veure les característiques principals d'aquest tipus d'eines:

Taula 14: Comparació dels frameworks de presentació.

	STRUTS	JSF
LLICÈNCIA LLIURE	✓	✓
GRATUÏTA	✓	✓
INTEGRACIÓ AMB IDES	✓	✓
INTEGRACIÓ AMB SPRING	✓	✓
FACILITAT D'APRENTATGE	✗	✓
VERSATILITAT	✓	✗
LLEUGERA	✓	✗
SUPORT D'IDES	✓	✓

Les dues eines són semblants a nivell de funcionalitat. Les separen funcions molt específiques i un mètode de desenvolupament totalment diferent.

L'elecció en aquest cas dependrà del tipus d'aplicació web a desenvolupar [43]. Segons el criteri del desenvolupador i del cap de projecte s'haurà de decidir entre l'orientació a accions d'Struts i l'orientació a components de JSF.

Gràfics estadístics

JReport

Aquesta eina, que s'ha comentat de forma detallada a la secció d'eines de generació d'informes, està instal·lada actualment al SER.RI-TIC. Segons les proves realitzades, no té la funcionalitat de creació de gràfics sense haver de crear tot un informe sencer.

JFreeChart

És una llibreria per Java que permet la creació de gràfics estadístics complexos de forma senzilla.

Les seves principals característiques són:

- API establerta i ben documentada, amb suport per molts tipus de gràfics.
- Disseny flexible, que en permet l'ús tant per desenvolupaments d'escriptori com per desenvolupaments sobre servidors.
- Suport per múltiples formats de sortida, inclòs components Swing.

La qualitat dels gràfics resultants és molt alta [44].

Té un *plugin* per integrar-se amb diversos *frameworks*, com l'Struts 2.

Avaluació

Donada l'absència d'aquesta funcionalitat al JReport, i les qualitats de JFreeChart, la llibreria elegida és aquesta última, JFreeChart.

NOM:	JFreeChart
DESENVOLUPADOR:	Object Refinery Limited
ÚLTIMA VERSIÓ:	1.0.8 - 23/11/2007
LLICÈNCIA:	LGPL
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.jfree.org/jfreechart
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Internacionalització (i18n) i localització (l10n)

A Internet no hi ha barreres ni fronteres. Per satisfer els clients, les aplicacions web van començar un procés de traducció a múltiples idiomes.

Tal i com s'ha explicat al capítol d'especificacions, no és el mateix internacionalitzar una aplicació que localitzar una aplicació. Les tècniques a aplicar en un i altre cas varien, però sobretot els resultats, ja que és molt més difícil localitzar una aplicació ja creada.

Tant i18n com l10n es centren únicament en la part de la interfície de l'usuari, deixant a mans del desenvolupador les decisions de disseny per internacionalitzar el contingut de les aplicacions web.

La internacionalització en Java s'implementa mitjançant la classe `ResourceBundle` i dels fitxers de traduccions `.properties` (un per cada *locale* o idioma que es vulgui). Per exemple, la següent configuració, que es mostra a la taula 15, permetria tenir un error de validació en dos idiomes:

Taula 15: Exemple de configuració de fitxers de *properties* en castellà i anglès.

	CASTELLÀ	ANGLÈS
NOM DEL FITXER	aplicacio_es_ES.properties	aplicacio_en_US.properties
CONTINGUT	error_message=Imposible validar ok_message=Validación correcta	error_message=Can't validate ok_message=Validation OK

Fer notar que s'han de complir els següents formats:

- Nom del fitxer: `nomAplicacio_xx_YY.properties`, on: `xx` és el codi de llengua i `YY` és el codi de país. Per exemple, per català el *locale* seria "ca_ES".
- Contingut: són un número il·limitat de línies amb format `clau=valor`. Les claus seran comunes a tots els fitxers i el valor contindrà la traducció.

Podrem obtenir els valors adequats mitjançant el mètode `getString(String key)` de la classe `ResourceBundle`, segons la clau que li passem per paràmetre.

Hi ha eines i *plugins* d'Eclipse que ajuden a crear els `ResourceBundle` necessaris:

- Eclipse `ResourceBundle Editor`: permet editar paral·lelament els fitxers i comprovar que totes les claus estan a tots els fitxers.
- `ETranslator`: *plugin* per Eclipse que permet traduir automàticament fitxers `.properties` utilitzant el servei de traducció de Google.

Jakarta Taglibs i18n

El projecte Jakarta Taglibs és un dipòsit lliure de llibreries de *JSP custom tags*.

Dintre del projecte hi ha multitud d'eines per controlar tasques no implementades a l'especificació inicial JSP. És una eina d'imprescindible revisió per tot desenvolupador de JSPs, sobretot les JSP Standard Tag Library (JSTL).

NOM:	Jakarta Taglibs
DESENVOLUPADOR:	Apache Software Foundation
ÚLTIMA VERSIÓ:	1.1
LLICÈNCIA:	Apache License 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://jakarta.apache.org/taglibs/doc/i18n-doc/intro.html
COMPATIBILITAT SO:	Multiplataforma

Els *tags* que ens ocupen en aquesta secció són els i18N, que formen part de l'especificació JSP 1.1 Tag Libs. Aquesta llibreria ens permet manegar de forma pràctica la complexitat d'implementar una aplicació web amb diverses llengües. Bàsicament permet utilitzar els ResourceBundles sense haver d'utilitzar codi Java incrustat. Per recuperar el valor adequat s'ha d'utilitzar la instrucció

```
<fmt:message key="value" />
```

Framework presentació

Fins ara s'ha explicat la internacionalització tant de codi Java com de fitxers JSPs, però les aplicacions web han crescut i actualment s'utilitzen *frameworks* especialitzats en la capa de presentació.

Dels *frameworks* que s'han presentat en podem destacar les següents característiques:

- JSF utilitza un únic ResourceBundle per *locale*. A més, s'ha de declarar el ResourceBundle a cada pàgina.
- Spring MVC utilitza, també, un únic ResourceBundle per *locale*.
- Struts 2 aposta per separar els fitxers segons les pàgines i/o accions definides.

Segons el desenvolupador i l'aplicació a crear serà més còmode un mètode o un altre. Per a grans aplicacions, amb molts de textos a traduir, serà més còmode tenir diversos fitxers, mentre que per aplicacions petites serà més còmode fer-ho amb un fitxer únic.

Avaluació

Les eines són complementàries, però per utilitzar els mecanismes d'internacionalització dels *frameworks* es necessita conèixer el funcionament dels Taglibs i18n.

Logging

El *logging* d'aplicacions web és una tasca molt important pels desenvolupadors. Durant el desenvolupament és possible utilitzar mètodes pocs ortodoxos de *logging* (com per exemple, els "System.out.println(...);"), però en l'entorn de producció aquest mètode només servirà per generar brossa als logs del servidor.

Mitjançant l'ús d'una llibreria específica per *logging* aconseguirem crear *logs* de qualitat, que ajudin a solucionar els problemes que es donin en l'entorn de producció.

log4j

És una utilitat de *logging* per Java. Es considera un *estàndard de facto*, el seu ús és gairebé universal.

Les seves principals característiques són:

NOM:	log4j
DESENVOLUPADOR:	Apache Software Foundation
ÚLTIMA VERSIÓ:	1.2.15 - 29/08/2007
LLICÈNCIA:	Apache Public License 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://logging.apache.org/log4j
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

- Gran rendiment amb el sistema de *log* desactivat. Abans d'utilitzar les sentències de *debug* es pot comprovar l'estat del sistema, activat o desactivat, amb la instrucció *isDebugEnabled()*. D'aquesta forma es pot millorar el rendiment, ja que amb el *debug* desactivat es pot escollir no avaluar les expressions de depuració.
- Diferents nivells de *log*, segons les necessitats (OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE, ALL).
- Facilitat de configuració per diferents formats de sortida (text, text amb *rolling file*, xml, tcp, udp, html, syslog, jdbc...).
- Instal·lació com una llibreria a l'aplicació web.
- Possibilitat d'activar i desactivar el sistema de *log* "en calent", sense necessitat de reiniciar el servidor ni fer una relectura del fitxer de configuració del servidor.

Actualment, al SER.RI-TIC s'està utilitzant pel desenvolupament d'aplicacions web. Tenint en compte aquest aspecte, juntament amb el fet que sigui un estàndard de facto, serà la tecnologia de *logging* elegida.

Tot seguit s'explicaran eines per treballar amb log4j.

Chainsaw v2

És un visor de *logs* generats amb log4j. Està desenvolupat pels mateixos membres de la comunitat log4j.

Les seves principals característiques són:

- Visió d'events remots, mitjançant la configuració de Receivers de log4j.
- Control de resposta, per no saturar la GUI amb massa events.
- Visió de diversos *logs* mitjançant pestanyes.
- Colorejat d'events.
- Filtre dinàmic per facilitar la cerca d'informació concreta.

NOM: Chainsaw
 DESENVOLUPADOR: Apache Software Foundation
 ÚLTIMA VERSIÓ: 2.0 - 02/03/2006
 LLICÈNCIA: Apache Public License 2.0
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://logging.apache.org/chainsaw>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Independent

JAMon

El Java Application Monitor (JAMon) és una eina que permet als desenvolupadors monitoritzar les aplicacions web J2EE en entorn de producció.

Permet fer un seguiment d'aplicacions mitjançant:

- JDBC/SQL, mitjançant un *proxy* que incorpora.
- log4j, mitjançant un nou Appender.
- HTTP, mitjançant un mòdul del contenidor web.
- EJB, mitjançant interceptors.

Aquesta eina solucionarà un dels problemes que tenen actualment els desenvolupadors del SER.RI-TIC, el poder accedir als *logs* del servidor sense dependre de cap altra persona.

Avaluació

L'ús del log4j és imprescindible. Pel que fa a les utilitats, el seu ús es recomanable, segons la funcionalitat que vulgui el desenvolupador.

NOM: JAMon
 DESENVOLUPADOR: Steve Souza
 ÚLTIMA VERSIÓ: 2.7 - 20/09/2007
 LLICÈNCIA: Tipus BSD
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://jamonapi.sourceforge.net/>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Independent

Single Sign-On (SSO)

Sun Java System Access Manager 7 - Client SDK

Actualment la protecció de l'accés a les aplicacions web es realitza mitjançant crides a un dipòsit LDAP. Cada aplicació gestiona la seva pròpia autenticació, per tant, els usuaris s'han de validar per a cada una de les aplicacions a les que vulguin accedir.

NOM: SJSAM Client SDK
DESENVOLUPADOR: Sun Microsystems, Inc.
ÚLTIMA VERSIÓ: 7.1
LLICÈNCIA: Propietària
COST LLICÈNCIA: Indeterminada
PÀGINA WEB: http://www.sun.com/software/products/access_mgr
COMPATIBILITAT SO: Multiplataforma

L'objectiu a assolir és que la validació tingui un punt únic d'entrada (Single Sign-On), i a partir d'aquest punt es pugui accedir a qualsevol recurs electrònic de la URV.

El mecanisme escollit pel SER.RI-TIC, i que consta com a requisit del projecte, és el producte Sun Java System Access Manager 7. Actualment està en procés d'instal·lació.

Des del punt de vista del desenvolupador, s'ha d'instal·lar un agent al servidor i un agent a les pàgines que hagin de fer les peticions de validació. S'entén per agent un paquet que permet l'accés a tots els usuaris que compleixin la política d'accés atorgada per l'Access Manager, definit al domini del servidor d'aplicacions. Hi ha dos possibles formes de realitzar la validació amb l'Access Manager:

- Amb un agent web. És un *servlet* que controla l'accés a les pàgines interceptant les peticions rebudes pel servidor web i comprovant-les amb una política de l'Access Manager. Si l'usuari que ha realitzat la petició no s'ha autenticat abans amb l'Access Manager (i per tant, no té una *cookie SSO*), és redirigit per l'agent a la pàgina d'autenticació. Si l'usuari ja s'havia autenticat, per tant té la *cookie SSO*, l'agent comprovarà si l'usuari té permisos per accedir al recurs electrònic sol·licitat. Les crides es fan mitjançant el Client SDK i seguint les instruccions de la documentació del desenvolupador [45].
- Amb un agent de servidor d'aplicacions. És similar al web, però només actua a nivell de domini d'autenticació i no conté llistes dels recursos a protegir.

Tests unitaris

Hi ha diferents tipus de tests [46], no només els unitaris. La taula 16 presenta els més rellevants:

Taula 16: Tipus de tests de software.

TIPUS	DESCRIPCIÓ
Unit Tests	Estan dissenyats per verificar petites parts d'un producte, idealment una única part individual per unit test. Per part individual s'entén una funcionalitat sencera, una classe o només un mètode o funció. Posteriorment pot servir com a font de documentació sobre com utilitzar aquell producte. Estan estretament lligats al codi al que es fa el test.
Regression Tests	Són verificacions que estan dissenyades per assegurar que no s'han introduït <i>bugs</i> a una nova versió, és a dir, que un codi produeix els mateixos resultats que anteriorment.
System Tests	Estan dissenyats per verificar tota l'aplicació, i no tenen perquè estar dissenyats pels desenvolupadors de l'aplicació. Inclouen els tests de càrrega, els tests en diferents plataformes i els tests en navegadors.
Customer Tests	Estan dissenyats per utilitzar l'aplicació de la mateixa forma que ho faria el client. Això inclou la instal·lació, actualització, usabilitat...
Interoperability Tests	Estan dissenyats per verificar com de bé treballen diverses aplicacions les unes amb les altres, normalment per comprovar que totes compleixen una especificació o protocol.

Dintre d'aquesta secció s'explicaran les eines per realitzar tests unitaris, però en successives seccions es mostraran eines per altres tipus de tests.

JUnit

És l'eina de verificació per Java més utilitzada, i actualment està integrada amb Ant, Maven i la majoria d'IDEs.

Permet crear noves proves a partir de

l'herència de la classe `TestCase`, i configurar-lo mitjançant alguns mètodes:

- Implementació dels mètodes `setUp` i `tearDown`: configuració i preparació del test per al mètode `setUp`, i finalització i neteja de recursos per `tearDown`.
- Implementació de mètodes de prova: implementació de mètodes que utilitzen mètodes `assert` per comprovar el correcte funcionament.
- Múltiples tests poden ser combinats en una classe `TestSuite`.
- Cada classe de test és una petita part de l'aplicació, una *unit*.

Hi ha disponibles GUIs pel control, disseny i execució de proves.

NOM:	JUnit
DESENVOLUPADOR:	Comunitat i Object Mentor
ÚLTIMA VERSIÓ:	4.4 - 18/07/2007
LLICÈNCIA:	Common Public License
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.junit.org
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

TestNG

És un entorn de verificació inspirat en Junit, però introduint noves funcionalitats.

Va ser iniciat per un usuari frustrat de Junit, ja que va trobar diversos problemes i no li

van oferir cap solució. Junit està pensat per fer proves a petites parts, objectes senzills, mentre que TestNG va néixer directament amb aspiracions de comprovar grans peces de codi, tasca sobre la que supera Junit.

NOM:	TestNG
DESENVOLUPADOR:	Cédric Beust
ÚLTIMA VERSIÓ:	08/11/2007
LLICÈNCIA:	Apache Software License
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://testng.org/
COMPATIBILITAT SO:	Multiplataforma
COMPATIBILITAT BD:	Independent

Característiques:

- Suport de JDK 5 Annotations (JDK suportat amb JavaDoc Annotations).
- Configuració flexible de proves.
- Suport per proves amb diferents proveïdors de dades (*data-driven*).
- Suport per paràmetres als tests.
- Potent model d'execució (sense l'herència de la classe TestSuite com JUnit).
- Suport per eines de desenvolupament (Eclipse, IDEA, Maven...).
- Utilitza les funcions estàndards de JDK per l'execució i el *logging*, evitant les dependències de terceres llibreries.
- Està preparat per ser automatitzat amb Ant i Maven.

A diferència de Junit, no està orientat únicament als tests unitaris, sinó que està dissenyat per gairebé qualsevol tipus de test: funcional, *end-to-end*, d'integració...

Avaluació

A la taula 17 podem veure les característiques principals d'aquest tipus d'eines:

Taula 17: Comparació sobre eines de testing unitari.

	JUNIT	TESTNG
LLICÈNCIA LLIURE	✓	✓
GRATUÏTA	✓	✓
INTEGRACIÓ AMB IDES	✓	✓
INTEGRACIÓ AMB SPRING	✓	✗
TEST NO BASAT EN HERÈNCIA	✗	✓
ANNOTATIONS PER CONFIGURAR ELS TESTS	✗	✓

L'eina més completa a nivell de funcionalitat és TestNG. De totes formes, el punt negatiu que te és molt important, ja que la integració amb Spring (que utilitza AppFuse) és un punt clau. El problema no és irresoluble [47], però si que complica el desenvolupament. A causa d'aquest fet l'elecció és JUnit.

Tests de càrrega

JMeter

És una aplicació d'escriptori dissenyada per provar el comportament d'aplicacions amb molta càrrega.

NOM: JMeter
 DESENVOLUPADOR: Apache Foundation
 ÚLTIMA VERSIÓ: 2.3 - 24/09/2007
 LLICÈNCIA: Apache License 2.0
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://jakarta.apache.org/jmeter/>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Independent

Pot ser utilitzada per comprovar el rendiment tant de recursos estàtics com dinàmics (fitxers, Servlets, *scripts* Perl, objectes Java, bases de dades i consultes...). Pot simular una gran càrrega al servidor, a la xarxa o a certs objectes per comprovar la resistència i analitzar el rendiment general de tota l'aplicació.

Característiques:

- Proves de càrrega i rendiment sobre HTTP i FTP, així com consultes a bases de dades (via JDBC).
- Pot treballar concurrentment amb *multithreading*.
- Guarda en *cache* els resultats per fer un posterior anàlisi (o *replay*) *offline*.
- Extensible mitjançant *plugins*.

JUnitPerf

Aquesta eina és un conjunt de *decorators*¹² utilitzats per mesurar el rendiment i l'escalabilitat de tests JUnit ja existents.

NOM: JunitPerf
 DESENVOLUPADOR: Clarkware Consulting, Inc.
 ÚLTIMA VERSIÓ: 1.9
 LLICÈNCIA: BSD
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://clarkware.com/software/JUnitPerf.html>
 COMPATIBILITAT SO: Multiplataforma

Incorpora dos *decorators*, un per temps (TimedTest) i un per tests de càrrega (LoadTest).

Característiques:

- La granularitat inclou els mètodes `setUp()`, `testXXX()` i `tearDown()`.
- No està dissenyada per ser una eina de *profiling* intens, només té l'ambició d'augmentar fàcilment la potència de tests més petits.

¹² NdR: Un *decorator* és un patró de disseny que permet afegir dinàmicament un nou comportament a un objecte.

The Grinder

És un *framework* lliure de test de càrrega per Java. Facilita la coordinació d'*scripts* de test en diversos processos, per paral·lelitzar mitjançant diverses màquines, utilitzant una aplicació de consola gràfica. Aquest tests són utilitzats a través de *plugins* Java, reutilitzables i proporcionats amb el paquet, tot i que es poden personalitzar. En l'última versió s'utilitza el llenguatge d'script Jython, que permet fer tests de codi Java sense haver d'utilitzar *plugins*.

NOM: The Grinder
 DESENVOLUPADOR: Paco Gómez
 ÚLTIMA VERSIÓ: 3.0beta - 17/03/2007
 LLICÈNCIA: GPL v2
 COST LLICÈNCIA: Gratuïta
 PÀGINA WEB: <http://grinder.sourceforge.net/>
 COMPATIBILITAT SO: Multiplataforma
 COMPATIBILITAT BD: Independent

Té un punt feble important, ja que s'ha d'aprendre com funciona el Jython (tot i que té una corba d'aprenentatge molt poc pronunciada).

Avaluació

A la taula 18 podem veure les característiques principals d'aquest tipus d'eines:

Taula 18: Comparació sobre eines de tests de càrrega.

	JMETER	JUNITPERF	THE GRINDER
LLICÈNCIA LLIURE	✓	✓	✓
GRATUÏTA	✓	✓	✓
TESTS SOBRE MÚLTIPLES FONTS	✓	✗	✗
PLUGINS	✓	✗	✓
TESTS ESCRITS AMB JAVA	✓	✓	✗
TESTS DE WEB SERVICES	✓	✗	✗

L'eina més completa a nivell de funcionalitat és l'Apache JMeter. Pot sobrecarregar múltiples fonts i de diverses formes, i a més, els tests s'escriuen en Java.

Tests de funcionalitat del sistema i de compatibilitat del navegador

Selenium

És una aplicació per realitzar tests d'aplicacions web. Els tests s'escriuen com taules HTML i poden ser executats directament a la majoria de navegadors web.

NOM:	Selenium
DESENVOLUPADOR:	ThoughtWorks
ÚLTIMA VERSIÓ:	0.8.3 - 20/09/2007
LLICÈNCIA:	Apache Public License 2.0
COST LLICÈNCIA:	Gratuïta
PÀGINA WEB:	http://www.openqa.org/selenium/
COMPATIBILITAT SO:	Windows, MacOSX i GNU/Linux
COMPATIBILITAT BD:	Independent

Consta de tres components diferenciats:

- Selenium IDE: és un entorn de desenvolupament de tests que permet grabar, editar i *debuggar* tests. Està implementat com una extensió de Firefox [48].
- Selenium Core: és el component que executa els tests directament sobre els navegadors. Està escrit amb JavaScript i DHTML. Té suport per:
 - Windows:
 - Internet Explorer 6.0 i 7.0
 - Firefox 0.8 fins 2.0
 - Mozilla Suite 1.6+, 1.7+
 - Opera 8 & 9
 - Mac OS X:
 - Safari 2.0.4+
 - Firefox 0.8 fins 2.0
 - Camino 1.0a1
 - Mozilla Suite 1.6+, 1.7+
 - Seamonkey 1.0
 - Linux:
 - Firefox 0.8 fins 2.0
 - Mozilla Suite 1.6+, 1.7+
 - Konqueror
 - Opera 8 & 9
- Selenium RC: és un servidor opcional que permet automatitzar els tests.

Convencions dels noms dels paquets Java

És important tenir un espai de noms de paquets Java coherent i respectuós per facilitar el desenvolupament i el reaprofitament de codi:

- El nom del paquet tindrà el format: `cat.urv.nomaplicacio`
- Estructura de paquets dintre del projecte:
 - `cat.urv.nomaplicacio.model` -> Entitats i *beans*, és a dir, la definició dels objectes de la capa de negoci.
 - `cat.urv.nomaplicacio.dao` -> Interfícies d'accés a dades, sigui quin sigui el mètode elegit per accedir a les dades (Hibernate...).
 - `cat.urv.nomaplicacio.dao.hibernate` -> Implementacions d'Hibernate.
 - `cat.urv.nomaplicacio.service` -> Interfícies de servei. Faciliten l'accés als objectes de la capa de negoci, de forma que podem utilitzar els mateixos components tant per una aplicació web, com per a un Web Service o per una aplicació d'escriptori.
 - `cat.urv.nomaplicacio.service.impl` -> Implementació dels serveis.
 - `cat.urv.nomaplicacio.webapp.action` -> Struts 2 Actions

Ús de formats estàndards lliures

Què és un format?

Un format és el que permet a una aplicació interpretar les dades crues d'un arxiu. En altres paraules, un format és una forma de representació de dades. Moltes vegades, els formats d'arxiu estan marcats la extensió, que identifica visiblement el format (pdf, html, png, gif...)

Formats lliures i formats propietaris

Per poder treballar en arxius es necessiten aplicacions que permetin la lectura, edició i emmagatzematge de les dades d'aquest arxiu. Per tant, necessita interpretar el format.

Format obert o lliure

Direm que un format és obert si la forma de representació de les seves dades és transparent i/o la seva especificació està disponible públicament. Tot i que normalment hi ha una autoritat pública o una institució internacional darrera, no sempre és el cas.

Format propietari o privatiu

Direm que un format és propietari si la forma de representació de les seves dades és opaca i la seva especificació no està disponible públicament. Sempre estan desenvolupats per empreses de software, per provocar que només les seves aplicacions puguin llegir completament el format.

Quatre motius per no utilitzar formats propietaris

1. Al compartir-ho hi ha el risc que el destinatari no pugui llegir l'arxiu.
2. Hi ha el risc de difusió d'informació confidencial, ja que no és possible comprovar el contingut real del fitxer.
3. Mantenir monopolis de software existents al món informàtic.
4. Evitar la dependència tecnològica d'un únic proveïdor.

Quatre motius per adoptar formats oberts

1. Garantir l'accessibilitat i perpetuïtat a les dades, amb independència del proveïdor.
2. Garantir una completa transparència del contingut dels arxius.
3. Limitar la propagació de virus, els formats oberts redueixen dràsticament el perill.
4. Promocionar la diversitat i la interoperabilitat.

Equivalències de formats

La taula 19 mostra l'equivalència de formats, segons diversos àmbits informàtics.

Taula 19: Equivalències de formats lliures i propietaris segons l'àmbit.

ÀMBIT	FORMAT PROPIETARI	FORMAT LLIURE
DOCUMENTS DE TEXT	DOC	HTML, DVI, OpenDocument ¹³ , PDF
FULLS DE CÀLCUL	XLS, XLW	OpenDocument ¹³
PRESENTACIONS	PPS, PPT	OpenDocument ¹³
IMATGES	GIF, BMP, TIF	JPEG, PNG
GRÀFICS VECTORIALS	WMF	SVG
AUDIO	MP3, WMA	OGG, MPC, FLAC

Existeixen llistes més extensives sobre formats lliures [49].

¹³ OpenDocument és un estàndard ISO, promogut per aplicacions com OpenOffice.org i Koffice.

4. Desenvolvimento

If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost \$100, get a million miles per gallon, and explode once a year, killing everyone inside.

Robert X. Cringely
revista InfoWorld

Procediments d'instal·lació

En aquesta secció es descriuran els processos d'instal·lació dels principals components de l'entorn de desenvolupament.

Instal·lació de Subversion

Per aquesta instal·lació es va utilitzar un paquet compilat per Solaris x86, aconseguida del dipòsit públic d'aplicacions SunFreeware [50]. No és una instal·lació senzilla, ja que té moltes dependències. Hi ha una pàgina concreta d'ajuda a la instal·lació de Subversion [51] on s'expliquen els passos a realitzar, així com totes aquestes dependències.

Com a introducció, fer notar que la instal·lació de paquets a Solaris es realitza mitjançant l'eina `pkgadd`:

```
$ pkgadd -d paquet
```

Les dependències de Subversion, seguint l'ordre d'instal·lació, són:

- swig-1.3.29
- openssl-0.9.8c
- db-4.2.52.NC
- expat-1.95.5
- gdbm.3
- libiconv-1.9
- libxml2-2.6.26
- zlib-1.2.3
- apache-2.0.59
 - libgcc o gcc
- neon-0.25.5

Després d'instal·lar totes les dependències de Subversion s'ha de modificar el *PATH* de les llibreries del sistema. Els directoris a afegir, i les comandes per fer-ho, en aquesta instal·lació en concret, són:

```
$ crle -u -l /usr/local/lib
$ crle -u -l /usr/sfw/lib
$ crle -u -l /usr/local/ssl/lib
```

També s'ha de modificar el fitxer `httpd.conf` de la configuració d'Apache per ajustar dos aspectes:

- Els paràmetres `User` i `Group`:

```
User nobody  
Group nobody
```

- Carregar els mòduls per permetre l'accés mitjançant webDAV:

```
LoadModule dav_svn_module modules/mod_dav_svn.so  
LoadModule authz_svn_module modules/mod_authz_svn.so
```

Arribat a aquest punt tindrem una instància de Subversion funcionant localment i mitjançant el servidor web Apache.

Instal·lació d'Eclipse, EasyEclipse i Aptana

Eclipse 3.3 Europa és l'últim llançament simultani de projectes de l'Eclipse Foundation, tal i com s'ha comentat al capítol de disseny. Aquesta versió és la que seguidament s'instal·larà.

Eclipse *flavours*

El primer pas per la instal·lació és descarregar el paquet de *software* a instal·lar. Podem realitzar dos tipus de descàrregues:

- Paquets d'Eclipse Europa. Dintre d'aquest tipus hi ha diverses opcions [52]:
 - Eclipse IDE for Java Developers. Eines per desenvolupar Java.
 - Eclipse IDE for Java EE Developers. Eines per desenvolupar aplicacions web amb tecnologia J2EE.
 - Eclipse for RCP/Plug-in Developers. Eines per desenvolupar *plugins* d'Eclipse o *Rich Client Applications* (RCA).
 - Eclipse Classic. La plataforma Eclipse amb els components bàsics.
- Distribucions de terceres empreses o *third-party distros*. Hi ha empreses que preparen paquets amb configuracions especials preparades amb el seu *flavour* especial:
 - BEA Logic
 - Obeo
 - IBM
 - Red Hat, amb el Red Hat Developer Studio
 - nexB, amb l'EasyEclipse

Instal·lació

Per instal·lar Eclipse s'han de seguir els següents passos:

- Descarregar el fitxer comprimit escollit, l'Eclipse IDE for Java EE Developers, i descomprimir-lo.
- Moure la carpeta "eclipse" a c:\eclipse. Aquesta carpeta conté tot el que necessitem per treballar amb l'IDE.

Dintre de la carpeta "eclipse" s'han de destacar dues carpetes, "plugins" i "features". Contenen tots els projectes que integren el paquet de l'IDE. La versió que s'ha escollit no inclou el codi font dels *plugins*. En cas que vulguem disposar d'ell el podem descarregar de la pàgina del projecte, o bé escollir un altre paquet complementari, com l'Eclipse WTP All-in-one SDK.

El *workspace*

Quan iniciem Eclipse per primera vegada ens pregunta el *workspace* amb que volem treballar. El *workspace* és la carpeta que utilitza Eclipse per emmagatzemar els projectes dels usuaris. Podem tenir diferents *workspaces*, cadascun amb els seus projectes, però normalment treballarem amb un únic *workspace* amb diversos projectes dintre seu.

Arribats a aquest punt ja estarà l'Eclipse instal·lat, sense necessitat d'haver de fer cap més configuració perquè sigui operatiu.

EasyEclipse

És una distribució empaquetada realitzada per l'empresa nexB. Conté la plataforma Eclipse juntament amb una selecció de *plugins*. Tot i ser un producte creat per una empresa, s'allibera sota llicència lliure.

El principal objectiu d'aquesta distribució és aconseguir un paquet, amb la plataforma Eclipse i una sèrie de *plugins*, redistribuïble i fàcilment instal·lable a qualsevol plataforma (Windows, GNU/Linux i MacOSX).

D'entre les versions disponibles, s'ha escollit la Server Java, orientada al desenvolupament d'aplicacions *server-side*. Aquesta versió du instal·lats els següents components:

- Inclou un entorn Java JDK 1.5 preinstal·lat.
- Control integrat amb Eclipse per diferents servidors d'aplicacions: Tomcat, Jboss, ...
- Eines específiques per *frameworks*: Hibernate Tools, Spring IDE, J2EE Tools.
- Eines per treballar amb bases de dades: Eclipse Data Tools, QuantumDB.
- Editor de fitxers amb reconeixement de tot tipus de llenguatges.
- Integració amb control de versions: Subclipse, Eclipse CVS client.

La instal·lació és senzilla, s'ha de descarregar el fitxer per la plataforma escollida i executar-lo.

Encara que finalment no s'utilitzi aquesta, és interessant donar-li un cop d'ull per comprovar quins dels *plugins* que du integrats poden ser d'utilitat.

Aptana

L'IDE Aptana està basat en Eclipse i pot instal·lar-se de dues formes diferents:

- Com una aplicació completa, amb la seva pròpia versió de la plataforma Eclipse.
- Com un *plugin* d'Eclipse, per instal·lar damunt d'una plataforma ja instal·lada.

L'opció recomanada és la instal·lació en forma de *plugin*, per tenir tot l'entorn integrat.

Instal·lació de Netbeans

Netbeans també compta amb diferents *flavours*. Els dedicats al desenvolupament de Java són:

- Web & Java EE: desenvolupament d'aplicacions web J2EE.
- Mobility: desenvolupament d'aplicacions per dispositius mòbils utilitzant J2ME.
- Java SE: desenvolupament d'aplicacions d'escriptori.
- All: tots els components de la plataforma.

Es recomana escollir l'últim, l'All, ja que com es pot veure a la comparativa de components [53], és l'únic que instal·la el pack UML.

Un cop descarregat el fitxer executable, la instal·lació és molt senzilla. En cas d'escollir la instal·lació del servidor web Tomcat, o del servidor d'aplicacions GlassFish V2, s'ha de tenir en compte el recordar l'usuari, el password i els ports de servei que s'especifiquin durant el procés.

Instal·lació de FishEye

Com ja s'ha explicat al capítol de disseny, FishEye és una aplicació *server-side* que permet realitzar l'explotació d'informació provinent d'un dipòsit CVS o Subversion. És una eina que ajuda a visualitzar el ritme de treball i estadístiques basant-se en les versions que reporta un equip de treball.

Requisits

Al ser una aplicació que s'instal·larà sobre un servidor centralitzat té una sèrie de requisits:

- Com a mínim és necessari un JRE 1.5.
- Subversion 1.1.
- Un client SVN pel servidor (per exemple, l'SVNKit Client).

Instal·lació

A la pàgina de FishEye hi ha una versió de prova que és pot descarregar per analitzar les funcionalitats del producte. La llicència és vàlida durant un mes.

Una vegada complerts els requisits la instal·lació és molt senzilla, ja que conté el seu propi servidor web.

Els passos a realitzar són:

1. Descarregar l'aplicació web comprimida de la pàgina web de FishEye [54].

2. Una vegada tenim el fitxer l'hem de descomprimir dintre del servidor (s'aconsella utilitzar la carpeta `/usr/local/`). Dintre la carpeta `bin` hi ha l'*script* per engegar l'aplicació, el fitxer `run.sh`. A l'executar-lo ens ha sortir un missatge informant que el servidor no troba el fitxer de llicència. No hi ha cap problema, accedint a l'aplicació web mitjançant la direcció `http://127.0.0.1:8080` es pot introduir la que s'ha generat a la web d'Atlassian.

Arribats a aquest punt l'eina està preparada per l'explotació de dades.

Instal·lació d'Oracle 10g Express

Es vol fer especial incís en que aquesta instal·lació és a nivell d'aquest projecte i que en cap cas és per a la instal·lació a les màquines dels desenvolupadors.

Per tal de realitzar les proves de desenvolupament ha estat necessari comptar amb una base de dades per les aplicacions web. Seguint els requisits del SER.RI-TIC, explicats a la taula 1, el gestor de bases de dades que s'utilitzarà és Oracle.

Oracle té una versió del seu gestor de base de dades orientat als entorns de desenvolupament, que es pot descarregar de forma gratuïta. Aquesta versió s'anomena Oracle Database 10g Express Edition. Només està disponible per Windows i Linux.

El procés d'instal·lació és senzill, no hi ha cap opció de personalització a destacar. Només fer notar que una vegada instal·lada tindrem la consola d'administració web escoltant al port 8080. Aquesta configuració és conflictiva amb la configuració de la majoria de servidors d'aplicacions. Per canviar el port per defecte hem d'executar aquest procediment com a usuari DBA o SYSTEM:

```
begin
  dbms_xdb.sethttpport('8090');
end;
```

Una configuració a modificar, pel correcte funcionament dels tests d'AppFuse, és pel número de processos oberts. S'ha d'executar la següent comanda SQL i reiniciar la base de dades:

```
ALTER SYSTEM SET PROCESSES=150 SCOPE=SPFILE
```

Com a últim pas complementari, s'ha d'aconseguir el *driver* JDBC d'Oracle per connectar-se des de Java a la base de dades. Per aquesta versió es necessita l'*Oracle Database 10g Release 2 (10.2.0.3) JDBC Drivers*, concretament el fitxer `ojdbc14.jar`, la versió per JDK 1.4 i 1.5.

Instal·lació de GlassFish V2

GlassFish V2 és el servidor d'aplicacions elegit per instal·lar a les màquines dels desenvolupadors, per fer totes les proves de desenvolupament.

La instal·lació del servidor no està basada en un executable o msi, sinó que és un fitxer .jar. En properes versions del servidor està previst que s'adopti l'instal·lador del SJSAS.

Els passos per instal·lar-lo són:

1. Descarregar l'última versió de GlassFish V2. Al moment d'escriure aquesta entrada, l'última versió disponible és la "V2 Update Release 1". Hi ha una versió multilingüe amb castellà. El fitxer descarregat el situarem a l'arrel del disc dur, ja que el procés d'instal·lació crea la carpeta "glassfish" a partir d'on està el fitxer pare.
2. Suposant que tenim instal·lat el JDK 5 o 6 i correctament configurat el JAVA_HOME amb aquest JDK (i no amb un JRE), executarem la següent comanda:

```
$ java -Xmx256m -jar glassfish-installer-*.jar
```

Després de llegir l'acord de la llicència es començarà a desempaquetar el servidor, creant la carpeta C:\glassfish.

3. Una vegada desempaquetat, procedirem a executar la configuració, que en aquest cas s'executa mitjançant Ant:

```
$ cd glassfish
$ lib\ant\bin\ant -f setup.xml
```

Tot seguit començarà la configuració del servidor, així com la instal·lació dintre dels serveis del sistema operatiu. Un dels punts més rellevants és la configuració de ports, que es realitzarà tal com segueix:

- El port 4848 para Admin.
- El port 8080 para HTTP Instance.
- El port 7676 para JMS.
- El port 3700 para IIOP.
- El port 8181 para HTTP_SSL.
- El port 3820 para IIOP_SSL.
- El port 3920 para IIOP_MUTUALAUTH.
- El port 8686 para JMX_ADMIN.

4. Afegir la carpeta C:\glassfish\bin\ a la variable PATH del sistema, perquè puguem arrancar el servidor des de qualsevol directori.
5. Arrancar el servidor:

```
$ asadmin start-domain domain1
```

Després d'una breu espera ens informarà dels serveis i ports donats d'alta.

6. Per últim, comprovar que ha arrancat correctament, utilitzant el navegador web amb la direcció <http://localhost:8080>

Arribats a aquest punt el servidor d'aplicacions està preparat per desplegar aplicacions web.

Configuració del driver JDBC d'Oracle

El driver JDBC d'Oracle s'ha d'instal·lar a la carpeta de llibreries del servidor d'aplicacions, perquè les aplicacions puguin utilitzar connexions JDBC cap a bases de dades Oracle. S'ha de copiar el fitxer del driver, descarregat a la secció de la instal·lació d'Oracle d'aquest capítol, a la carpeta

```
C:\glassfish\lib
```

I reiniciar el servidor GlassFish amb les següents comandes (o través de la consola web):

```
$ asadmin stop-domain domain1  
$ asadmin start-domain domain1
```


Instal·lació d'Eclipse BIRT

Tant pel BIRT Report Designer com pel BIRT Runtime hi ha disponible un pack de llengües amb el castellà.

BIRT Report Designer

El dissenyador d'informes d'Eclipse BIRT es pot descarregar de tres formes diferents:

- All-in-one: conté una instal·lació de l'Eclipse SDK amb tots els *plugins* necessaris.
- Framework: només els plugins de BIRT, sense les dependències (SDK, DTP, GEF, EMF, WTP), per una instal·lació ja existent d'Eclipse.
- RCP Designer: el dissenyador en forma d'RCP (*Rich Client Application*).

L'opció més senzilla i estable és la instal·lació de l'aplicació RCP. D'aquesta forma tindrem l'eina aïllada de la resta d'entorn, facilitant el desenvolupament en paral·lel.

La instal·lació és molt senzilla, només s'ha de descarregar el fitxer de la web i descomprimir-lo.

BIRT Runtime

Instal·lació del runtime

Per desplegar els informes dintre d'aplicacions web es necessita instal·lar un component, una aplicació war, sobre el servidor d'aplicacions.

S'ha de descarregar el fitxer de la web, descomprimir-lo i instal·lar el fitxer birt.war. La instal·lació és realitzada amb qualsevol dels mètodes explicats a la guia d'administració de GlassFish d'aquest capítol.

Instal·lació dels drivers JDBC

Per fer funcionar el BIRT amb una base de dades s'ha de copiar el driver JDBC, el que hem descarregat a la secció d'instal·lació de l'Oracle 10g Express, dintre de la carpeta:

```
C:\glassfish\domains\domain1\applications\j2ee-modules\  
birt\WEB-INF\platform\plugins\org.eclipse.birt.report.data.oda.jdbc\drivers
```

Instal·lació de la dependència Apache Commons Logging

Hi ha una dependència de BIRT, Apache Commons Logging [55], que s'haurà d'instal·lar manualment. S'ha de descarregar la versió binària i copiar els fitxers .jar a la carpeta

```
C:\glassfish\lib
```

I reiniciar el servidor GlassFish amb les següents comandes (o través de la consola web):

```
$ asadmin stop-domain domain1
$ asadmin start-domain domain1
```

Un cop instal·lat estarà disponible a la direcció

```
http://localhost:8080/birt/
```

i si la instal·lació ha estat correcta hauríem de veure un missatge de confirmació.

Prova de funcionament

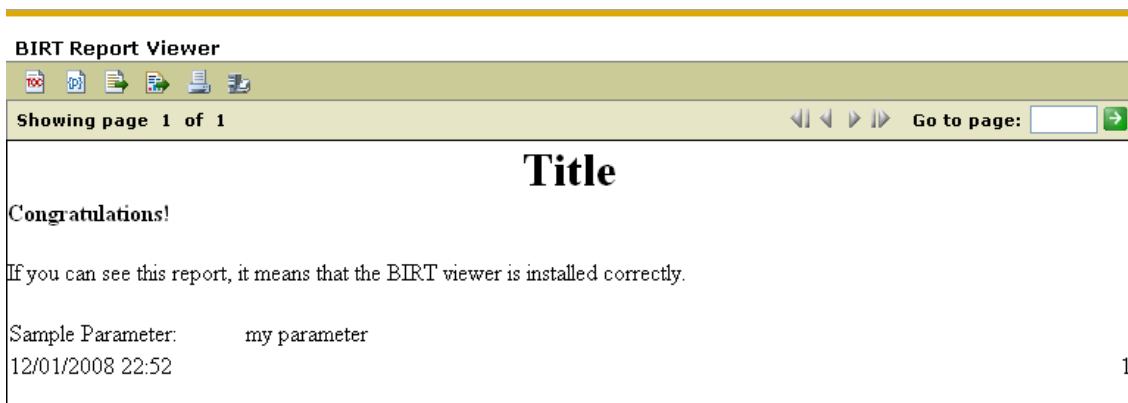
Per fer una prova del correcte funcionament podem executar l'informe de prova bàsic. Hi ha dos modes, el de visor de documents o el d'execució d'informes. mitjançant la següent direcció entrarem dintre del visor, amb el que tindrem una barra de navegació o d'eines per veure l'informe:

```
http://localhost:8080/birt/frameset?__report=test.rptdesign&sample=my+parameter
```

i amb aquesta entrarem dintre la d'execució, on només veurem l'informe

```
http://localhost:8080/birt/run?__report=test.rptdesign&sample=my+parameter
```

A la següent imatge podem veure l'aspecte del visor d'informes web.



Instal·lació d'OpenReports

El primer pas per començar la instal·lació d'OpenReports és descarregar el *software* des de la seva web. El fitxer amb que s'instal·larà aquest exemple té el nom "or-3.0.1.zip". Una vegada descomprimit comença el procés de configuració de la instal·lació.

Instal·lació dels *drivers* JDBC

Per fer funcionar el BIRT amb una base de dades s'ha de copiar el driver JDBC, el que hem descarregat a la secció d'instal·lació de l'Oracle 10g Express, dintre de la carpeta:

```
.\WebRoot\WEB-INF\lib
```

Configuració de la base de dades

La configuració de la base de dades es realitza en 2 passos, primer la creació de l'esquema a la base de dades i segon amb la inserció de l'usuari administrador a aquestes taules.

Creació de l'esquema i les taules

Crearem un nou esquema dintre la base de dades, amb el nom OpenReports. Després s'ha d'executar l'script següent sobre aquest esquema:

```
.\database\schema\or_ddl_oracle.sql
```

Si a l'executar dóna error és perquè falten ; al final de les sentències SQL.

També executarem l'script de creació de les taules de l'scheduler:

```
.\database\schema\quartz\tables_oracle.sql
```

Creació de l'usuari administrador

Per crear l'usuari administrador s'ha d'executar la següent comanda:

```
INSERT INTO REPORT_USER (REPORTUSER_ID, NAME, PASSWORD, PDF_EXPORT_TYPE)
VALUES (hibernate_sequence.nextval, 'admin', 'password', 0)
```

I per donar-li els permisos d'administració la següent comanda:

```
INSERT INTO USER_SECURITY (USER_ID, ROLE_NAME) VALUES (1, 'ROOT_ADMIN_ROLE')
```

Configuració d'Hibernate i de l'scheduler

OpenReports es connecta a la base de dades mitjançant Hibernate, per tant, també el configurarem adequadament. S'ha d'editar el fitxer

```
.\src\openreports.properties
```

Quedarà de la següent forma

```
# properties used by the Spring configuration

hibernate.dialect=org.hibernate.dialect.OracleDialect

hibernate.jdbc.driver=oracle.jdbc.OracleDriver
hibernate.jdbc.url=jdbc:oracle:thin:@localhost:1521:XE
hibernate.jdbc.username=openreports
hibernate.jdbc.password=openreports
hibernate.jdbc.maxIdle=2
hibernate.jdbc.maxActive=5

quartz.jobStore.driverDelegateClass =
    org.quartz.impl.jdbcjobstore.oracle.OracleDelegate

# use the following line for HSQLDB demo Quartz databases
# quartz.jobStore.selectWithLockSQL = SELECT * FROM {0}LOCKS UPDLOCK WHERE
LOCK_NAME = ?

# use this for all other Quartz databases
quartz.jobStore.selectWithLockSQL = SELECT * FROM {0}LOCKS WHERE LOCK_NAME = ?
FOR UPDATE

quartz.jdbc.driver = oracle.jdbc.OracleDriver
quartz.jdbc.url = jdbc:oracle:thin:@localhost:1521:XE
quartz.jdbc.username = openreports
quartz.jdbc.password = openreports
quartz.jdbc.maxIdle=2
quartz.jdbc.maxActive=5

#number of threads that are available for concurrent execution of jobs
org.quartz.threadPool.threadCount = 5
```

Configuració del directori contenidor dels .properties

Per fer-lo funcionar sobre GlassFish és necessari canviar el el següent codi situat al principi del fitxer `./WebRoot/WEB-INF/applicationContext.xml`:

```
<bean id="environment"
      class="org.springframework.beans.factory.config.PropertyPlaceholderCo
nfigurer"
      lazy-init="false">
  <property name="ignoreResourceNotFound" value="false" />
  <property name="locations">
    <list>
      <value>
        /WEB-INF/openreports.properties
      </value>
    </list>
  </property>
</bean>
```

Creació del fitxer empaquetat .war

Una vegada estigui tot configurat ja podem crear el war executant la següent comanda des de la carpeta arrel d'OpenReports:

```
C:\glassfish\lib\ant\bin\ant.exe war
```

Aquest comanda generarà el fitxer:

```
.\deploy\openreports.war
```

Aquest fitxer és el que s'ha de desplegar sobre el servidor GlassFish, seguint algun dels mètodes explicats a la guia d'administració de GlassFish d'aquest capítol.

Configuració del directori d'informes amb l'administrador web

L'últim pas serà comprovar la correcta instal·lació entrant dintre de l'aplicació i configurar el directori d'on el servidor llegirà els informes.

Per accedir a l'administrador web entrarem dintre la URL:

```
http://localhost:8080/openreports
```

Entrarem amb l'usuari administrador creat anteriorment i anirem a la secció *Administration* -> *General Settings* i configurarem els següents paràmetres com a mínim:

- Base Directory, amb el valor

```
C:\glassfish\domains\domain1\applications\j2ee-modules\openreports\reports\
```

- Report Generation Directory, amb el valor

```
C:\glassfish\domains\domain1\applications\j2ee-  
modules\openreports\reports\generated\
```

- Temp Directory, amb el valor

```
C:\glassfish\domains\domain1\applications\j2ee-  
modules\openreports\reports\temp\
```

- Query Report Max Rows, amb el valor 1000
- Mail Host, amb stmp.urv.cat

Les carpetes indicades s'han de crear abans de configurar el servidor.

Arribats a aquest punt, l'aplicació web està llesta per ser utilitzada.

Procediments de configuració i ús

En aquesta secció es descriuran els processos d'instal·lació dels principals components de l'entorn de desenvolupament.

Guía ràpida de GlassFish V2

Consola d'administració web

La consola d'administració és una eina de gestió del servidor mitjançant una interfície web. Simplifica les tasques de manteniment i configuració. Les tasques més usuals són:

- Fer el desplegament i replegament d'aplicacions.
- Activar i desactivar aplicacions que ja tinguem desplegades.
- Identificar quines aplicacions hi ha actualment funcionant.
- Configurar recursos JDBC i altres aspectes del servidor.
- Visualitzar fitxers de configuració.

Per accedir a la consola d'administració web haurem d'accedir a la URL amb el següent usuari i password per defecte:

- URL: `http://localhost:4848`
- Usuari: `admin`
- Password: `adminadmin`

Si hem instal·lat la versió multilingüe el llenguatge per defecte hauria de ser el castellà.

Logs del servidor

Mitjançant la consola d'administració podem visualitzar els *logs*, però en certes ocasions ens serà més convenient accedir directament mitjançant el sistema d'arxius. Els *logs* estan situats a la carpeta:

```
$ C:\glassfish\domains\domain1\logs
```

Director d'autodeploy

Cada domini de Glassfish conté un directori anomenat "autodeploy" que fa un desplegament automàtic de qualsevol aplicació web empaquetada (WAR, EAR o JAR) que es copii dintre d'ella. El directori és:

```
$ C:\glassfish\domains\domain1\autodeploy
```

Aquesta carpeta ens pot servir per fer proves ràpides de desplegament d'aplicacions web que no tinguem configurades en l'entorn Eclipse o NetBeans.

Un truc útil, sobre plataforma Windows, és tenir un enllaç directe a aquesta carpeta a l'escriptori o al menú "Enviar a", per poder enviar els fitxers fàcilment.

Desplegament amb Eclipse WTP

Eclipse WTP no ve configurat per defecte per treballar amb servidor GlassFish o SJSAS. Per sort, hi ha un projecte lliure que s'encarrega de desenvolupar un *plugin* d'Eclipse per poder aprofitar tota la potència d'automatització de l'IDE. A *GlassFish Plugins* [56] podem trobar el *plugin*, anomenat GlassFish Eclipse Plugin.

La instal·lació és senzilla, s'ha de copiar el contingut del fitxer zip dintre de la carpeta *plugins* de la instal·lació local d'Eclipse. Després de reiniciar l'Eclipse tindrem el servidor GlassFish a la secció de configuracions dels servidors, al menú "Preferences -> Servers -> Installed Runtimes".

Desplegament amb NetBeans i Sun Java Studio Creator

NetBeans i SJSJC venen configurats per defecte per treballar amb servidors GlassFish o SJSAS, de forma que es podran realitzar els *deploys* d'aplicacions normalment i sense haver d'instal·lar cap component addicional.

Desplegament de prova

Podem descarregar una aplicació HelloWorld, empaquetada amb el format WAR, per fer les primeres proves de *deploy* amb el servidor. Aquest fitxer el podem trobar a la direcció:

```
http://glassfish.dev.java.net/downloads/quickstart/hello.war
```

Una vegada desplegada per qualsevol dels mètodes comentats, podrem comprovar el seu funcionament mitjançant la URL:

```
http://localhost:8080/hello
```


Creació i configuració d'un dipòsit

Creació del dipòsit

La primera tasca a realitzar és la creació del propi dipòsit. Alguns aspectes que s'han de tenir en compte són:

- Quines dades s'espera que hi hagi al dipòsit o dipòsits i com estaran organitzades.
- A quin servidor estarà el dipòsit i com s'hi accedirà.
- Quins tipus de control d'accés i notificació d'events es necessitaran.
- Quin tipus d'emmagatzemament es voldrà utilitzar.

Aquests aspectes són d'administració, per tant, es necessitarà la col·laboració de l'administrador del sistema, si és una tercera persona.

Per crear un dipòsit seguirem el següent procediment a la consola de Solaris:

```
$ svnadmin create /carpeta_del_dipòsit
$ chown -R nobody:nobody /carpeta_del_dipòsit
```

Des d'aquest moment ja funcionarà localment. Exemple d'un *checkout* local:

```
$ svn co file:///dipòsit wc
```

El següent pas és configurar una Location, al fitxer httpd.conf de l'Apache, per accedir al dipòsit mitjançant el servidor web:

```
<Location /carpeta_de_apache>
  DAV svn
  SVNPath /carpeta_del_dipòsit
</Location>
```

Al reiniciar el servidor, el dipòsit ja serà accessible per màquines externes:

```
$ svn co http://192.168.1.5 wc
```

Planificació de l'organització d'un dipòsit

Es recomana [57] que cada dipòsit tingui la estructura que es mostra a la taula 20:

Taula 20: Planificació de l'organització de carpetes d'un dipòsit.

DIRECTORI	EXPLICACIÓ
projecte1/	Arrel del dipòsit. Aquesta carpeta només ha de contenir les carpetes “trunk”, “tags” i “branches”.
trunk/	És el directori sota del qual es realitzen les principals tasques de desenvolupament.
tags/	És el directori on es poden crear les diferents branques d'un projecte principal.
branches/	És el directori que conté <i>snapshots</i> d'una versió concreta, normalment versions estables i entregades als clients.

Abans d'executar el primer import del dipòsit s'hauran de crear aquests directoris i situar dintre de “trunk” el contingut inicial a importar. La resta de carpetes inicialment estaran buides, i s'aniran emplenant a mesura que l'equip generi dades.

Importació de dades

Una vegada creada l'estructura executarem la següent comanda per importar el directori “projecte1” al dipòsit:

```
$ svn import projecte1 http://servidor/dipòsit/projecte1 -m "Importació inicial"
```

Fer notar que és imprescindible incloure missatges d'informació a gairebé totes les operacions de Subversion. Com més concisos siguin, més útil ens serà posteriorment.

Altres aspectes

Hi ha detalls que s'han de conèixer quan es treballa amb Subversion:

- Les funcionalitats del control de versions únicament serveixen per fitxers de text. És a dir, pels fitxers binaris únicament comprova que s'hagin modificat, no quines són les modificacions realitzades.
- Hi ha un control dels diferents tipus finals de línia de les plataformes Windows (CRLF) i Unix (LF). Si estem treballant manualment amb un mateix codi en dues plataformes diferents podem veure canvis que realment no ho són. Subversion soluciona aquest problema, transparentment, amb la configuració de la propietat *svn:eol-style* a *native*. D'aquesta forma els canvis de línia són gestionats automàticament segons la plataforma on s'executi el client svn.
- Ens pot interessar no versionar certs fitxers, és a dir, ignorar-los. Per exemple, documents i arxius temporals, fitxers intermedis dels compiladors, directoris amb *backups* manuals... Hi ha dues formes diferents d'aconseguir-ho:
 - Configuració global del Subversion, mitjançant la propietat *global-ignores*. S'aplicarà a qualsevol ítem que encara no hagi estat versionat. És a dir, si s'activa un *global-ignores* però el fitxer ja està al dipòsit, es continuarà versionant. Aquest tipus de filtre s'utilitza, per exemple, per no versionar còpies de seguretat dels editors de text.
 - Configuració específica d'un directori versionat. S'aplicarà a tots els usuaris que versionen aquella carpeta. Per activar-ho s'ha d'executar la següent comanda:

```
svn propedit svn:ignore nom_directori patró
```

Creació d'un esquelet de projecte amb AppFuse

El procediment que seguidament es presenta és el primer pas per la generació de l'esquelet d'una aplicació web amb AppFuse. Aquest pas és independent de l'IDE amb que es desenvolupi.

Requeriments

Els requeriments d'AppFuse són:

- JDK 1.5+.
- MySQL 5.x, tot i que modificarem la configuració perquè treballi amb l'Oracle que ja hem instal·lat anteriorment.
- De forma opcional, un servidor SMTP per enviar correus.
- Maven 2.0.7

Només fa falta instal·lar Maven2 per complir els requeriments. Pels IDEs amb que treballem necessitarem un *plugin* per poder-lo utilitzar directament:

- Maven Integration for Eclipse.
- Mevenide2 for Netbeans

Es recomana crear els projectes per entorn Windows des de Cygwin, per estalviar problemes amb els caràcters de *path* (/ i \). Amb una instal·lació bàsica n'hi hauria prou perquè el Maven funcioni, i realment és molt pràctic ja que et dóna tota la potència de *bash* amb Windows. A l'apartat Utilitats del capítol de disseny d'aquest document es pot trobar més informació sobre Cygwin.

Creació de l'esquelet

Aquesta instal·lació tindrà les següents característiques:

- El tipus d'*archetype* serà el bàsic.
- *Framework* de presentació amb Struts 2 i una altra versió per provar JSF.
- Configuració de la base de dades Oracle.
- Configuració de validació d'usuaris LDAP, tot i que no s'utilitzarà durant el desenvolupament. El motiu és que s'ha de crear un mapatge entre un camp del directori LDAP i la gestió de rols d'AppFuse, i aquest mapatge queda fora de l'objectiu del projecte.
- Funcional amb els servidors d'aplicacions GlassFish i Jetty.
- Preparació per treballar amb els IDEs Eclipse i NetBeans.

Per crear el projecte i descarregar, automàticament, tots els fitxers i dependències introduïrem les següents comandes:

```
$ mkdir workspace_temp
$ cd workspace_temp
$ mvn archetype:create -DarchetypeGroupId=org.appfuse.archetypes
-DarchetypeArtifactId=appfuse-basic-struts
-DremoteRepositories=http://static.appfuse.org/releases
-DarchetypeVersion=2.0.1 -DgroupId=cat.urv.pfcei.demol
-DartifactId=1_AppFuseStruts2Basic
```

Hi ha dos paràmetres que han de ser personalitzats:

- `-DgroupId`, és el nom del paquet Java amb el que es crearà l'esquelet. Al capítol de disseny hi ha un apartat amb més informació sobre les convencions dels noms dels paquets Java a utilitzar.
- `-DartifactId`, és el nom que tindrà el projecte.

Amb aquesta comanda el que ordenem a Maven és que descarregui tot el projecte d'AppFuse des de la seva web i ens generi el codi coherentment segons les indicacions de paquet i de nom de l'aplicació.

Una vegada acabat el procés, si mirem dintre el directori, veurem que tenim una nova carpeta "src", amb tot el codi font de l'esquelet.

Oracleitzant l'AppFuse

Si tinguéssim instal·lada una base de dades MySQL ara ja podríem executar l'aplicació web, però en aquest cas, com tenim una base de dades Oracle, haurem de fer un pas més, configurar AppFuse, i consegüentment, tota la seva capa d'accés a dades. Aquí veiem clarament un dels avantatges de treballar amb un ORM, la independència del proveïdor de dades.

El primer pas és preparar la base de dades. Si no està creat, crear l'*schema* i usuari sobre el qual treballarà AppFuse. Per aquesta instal·lació s'ha escollit l'usuari i password afuser/afuser.

Obrim el fitxer `1_AppFuseStruts2Basic/pom.xml` i buscarem les següents línies:

```
<profile>
  <id>oracle</id>
  <properties>
    <dbunit.dataTypeFactoryName>
      org.dbunit.ext.oracle.OracleDataTypeFactory
    </dbunit.dataTypeFactoryName>
    <dbunit.schema>SYSTEM</dbunit.schema>
```

```

        <!-- Make sure to capitalize the schema name -->
<hibernate.dialect>
    org.hibernate.dialect.Oracle9Dialect
</hibernate.dialect>
<jdbc.groupId>com.oracle</jdbc.groupId>
<jdbc.artifactId>ojdbc14</jdbc.artifactId>
<jdbc.version>10.2.0.2.0</jdbc.version>
<jdbc.driverClassName>oracle.jdbc.OracleDriver</jdbc.driverClassName>
<jdbc.url><![CDATA[ jdbc:oracle:thin:@localhost:1521:XE ]]></jdbc.url>
<jdbc.username>system</jdbc.username>
<jdbc.password>system</jdbc.password>
</properties>
</profile>

```

Ho modificarem per configurar-ho segons la nostra instal·lació: l'*schema* (sobretot posar-ho en majúscules), l'*hibernate.dialect*, la versió, la url jdbc per fer-la coincidir amb l'entrada adequada de *tnsnames.ora*, la màquina on ens connectem, l'usuari i el password:

```

<profile>
  <id>oracle</id>
  <properties>
    <dbunit.dataTypeFactoryName>
      org.dbunit.ext.oracle.OracleDataTypeFactory
    </dbunit.dataTypeFactoryName>
    <dbunit.schema>AFUSER</dbunit.schema>
    <!-- Make sure to capitalize the schema name -->
    <hibernate.dialect>
      org.hibernate.dialect.Oracle10gDialect
    </hibernate.dialect>
    <jdbc.groupId>com.oracle</jdbc.groupId>
    <jdbc.artifactId>ojdbc14</jdbc.artifactId>
    <jdbc.version>10.2.0.1.0</jdbc.version>
    <jdbc.driverClassName>oracle.jdbc.OracleDriver</jdbc.driverClassName>
    <jdbc.url><![CDATA[ jdbc:oracle:thin:@localhost:1521:XE ]]></jdbc.url>
    <jdbc.username>afuser</jdbc.username>
    <jdbc.password>afuser</jdbc.password>
  </properties>
</profile>

```

Seguidament reconstruïrem el projecte en funció dels nous canvis activats a la configuració:

```
$ mvn integration-test -Poracle
```

Aquesta comanda ens donarà un error, perquè falta el fitxer *ojdbc14.jar*, el driver JDBC d'Oracle. A causa de la llicència d'aquest fitxer no es pot distribuir amb AppFuse, i per tant, l'haurèm de descarregar manualment. A la secció d'instal·lació de la base de dades Oracle 10g Express Edition s'explica com aconseguir el *driver*.

Aquest fitxer el copiarem dintre la carpeta `1_AppFuseStruts2Basic/lib`. Per finalitzar la instal·lació executarem la següent comanda:

```
$ mvn install:install-file -DgroupId=com.oracle -DartifactId=ojdbc14  
-Dversion=10.2.0.1.0 -Dpackaging=jar -Dfile=./lib/ojdbc14.jar
```

Preparant l'execució de l'aplicació

En aquest moment sorgeix el dubte sobre per què no s'ha executat cap *script* de creació de taules, inserció de dades... Tota aquesta feina es realitza també automàticament mitjançant Maven 2, amb Hibernate i dbunit. Per realitzar aquesta tasca hem de fer el primer *deploy* de l'aplicació sobre el servidor Jetty que porta integrat AppFuse.

```
$ cd 1_AppFuseStruts2Basic  
$ mvn -Poracle
```

Aquesta última comanda trigarà força estona, ja que executa totes les tasques Maven del projecte. Si durant la instal·lació hi ha algun problema podem afegir un `-e` a la crida de `mvn` per veure les traces d'error.

Provant l'aplicació sobre el servidor d'aplicacions Jetty

Finalment, només queda fer el *deploy* de l'aplicació al servidor. Primer ho provarem amb el servidor Jetty que porta integrat:

```
$ mvn jetty:run-war -Poracle
```

En aquest moment podem anar a la URL `http://localhost:8080` per veure l'aplicació funcionant. Els passwords per defecte són `admin/admin` pel rol d'administrador i `user/user` pel rol d'usuari.

Provant l'aplicació sobre el servidor d'aplicacions GlassFish

Per fer un *deploy* manual sobre el servidor GlassFish (mitjançant la consola d'administració web o mitjançant la carpeta `autodeploy`) hem d'agafar el fitxer `.war` que hi ha a la carpeta `"1_AppFuseStruts2Basic/target/"` i desplegar-lo amb el mètode escollit.

Per automatitzar el deploy mitjançant Maven 2 hem de seguir el següent procediment:

- Utilitzar l'archetype `exec:exec`, tal i com s'explica en l'article *Deploying to GlassFish using Maven2* [58]. D'aquest article també és interessant l'explicació sobre com fer *deploys* al servidor GlassFish des de la línia de comandes amb la comanda *asadmin*. Per incorporar-ho al nostre projecte hem de modificar el fitxer `pom.xml` i afegir el següent codi a la secció de "plugins":

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <executions>
    <execution>
      <phase>deploy_glassfish</phase>
      <goals>
        <goal>exec</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <executable>C:\glassfish\bin\asadmin.bat</executable>
    <arguments>
      <argument>deploy</argument>
      <argument>--user=admin</argument>
      <argument>--passwordfile=C:\glassfish_pass.txt</argument>
      <argument>--host=localhost</argument>
      <argument>--port=4848</argument>
      <argument>target/${artifactId}-${version}.${packaging}</argument>
    </arguments>
  </configuration>
</plugin>
```

Una vegada configurat, per fer un *deploy* del war cap al servidor definit a la configuració, s'haurà d'executar la següent comanda:

```
$ mvn exec:exec
```

LDAPitzant AppFuse

Al SER.RI-TIC hi ha directori LDAP per gestionar la identificació d'usuaris, tant per l'entrada a la Intranet i correu electrònic com per a la majoria d'aplicacions web. AppFuse pot incorporar automàticament LDAP com a sistema de validació per l'aplicació web que genera.

Com s'ha comentat a la introducció, aquest mètode s'explica com a demostració de la capacitat de treballar amb LDAP, tot i que no s'utilitzi per a les proves.

Per fer-ho s'han de realitzar els següents canvis:

- Si realment vulguéssim que funcionés la validació d'usuaris mitjançant LDAP i no la taula d'usuaris d'AppFuse, s'hauria d'editar el fitxer `src/main/webapp/WEB-INF/security.xml` i canviar la línia de la secció `authenticationManager`:

```
<ref bean="daoAuthenticationProvider"/>
```

per la línia:

```
<ref bean="ldapProvider"/>
```

- Al mateix fitxer, afegir una secció per l'identificador `LdapProvider`:

```
<bean id="ldapProvider"
  class="org.acegisecurity.providers.ldap.LdapAuthenticationProvider">
  <constructor-arg>
    <bean class="org.acegisecurity.providers.ldap.
      authenticator.BindAuthenticator">
      <constructor-arg ref="initialDirContextFactory" />
      <property name="userDnPatterns">
        <list>
          <value>uid={0}</value>
        </list>
      </property>
      <property name="userSearch" ref="userSearch" />
      <property name="userDetailsMapper" ref="ldapUserDetailsMapper" />
    </bean>
  </constructor-arg>
  <constructor-arg>
    <bean class="org.acegisecurity.providers.ldap.
      populator.DefaultLdapAuthoritiesPopulator">
      <constructor-arg ref="initialDirContextFactory" />
      <constructor-arg value="" />
      <property name="groupRoleAttribute" value="cn" />
      <property name="groupSearchFilter"
        value="(&!(objectclass=groupOfUniqueNames) (uniqueMember={0}))"
      />
      <property name="searchSubtree" value="true" />
      <property name="rolePrefix" value="" />
      <property name="convertToUpperCase" value="false" />
    </bean>
  </constructor-arg>
</bean>

<bean id="initialDirContextFactory"
```

```

class="org.acegisecurity.ldap.DefaultInitialDirContextFactory">
<constructor-arg value="\${ldap.url}/${ldap.base}" />
<property name="managerDn" value="\${ldap.username}" />
<property name="managerPassword" value="\${ldap.password}" />
</bean>

<bean id="userSearch"
class="org.acegisecurity.ldap.search.FilterBasedLdapUserSearch">
<constructor-arg index="0" value="" />
<constructor-arg index="1" value="(uid={0})" />
<constructor-arg index="2" ref="initialDirContextFactory" />
<property name="searchSubtree" value="true" />
</bean>

<bean id="ldapUserDetailsMapper"
class="org.acegisecurity.userdetails.ldap.LdapUserDetailsMapper">
<property name="rolePrefix" value="" />
</bean>

```

- Descomentar la següent línia:

```

<bean id="passwordEncoder"
class="org.acegisecurity.providers.encoding.ShaPasswordEncoder" />

```

- Crear el fitxer `src/main/resources/ldap.properties` i emplenar-lo amb la configuració LDAP, per exemple:

```

ldap.url=ldap://servidor_ldap:389
ldap.base=dc=es
ldap.username=uid=usuariLDAP
ldap.password=passwordLDAP

```

Arribats a aquest punt l'AppFuse està configurat per treballar amb LDAP.

El procediment de configuració del mapatge correcte es pot aconseguir dels següents recursos:

- [UserDetails from LDAP and DB \[59\]](#).
- [LDAP-Authentication with Active Directory \[60\]](#).

El mode de codi complet d'AppFuse

En l'estat actual de l'aplicació generada per AppFuse no veiem cap fitxer del nucli de l'aplicació. A partir d'aquest punt tenim dues opcions:

1. Ampliar l'aplicació inicial, afegint *actions* i *beans*. S'anomena *embedded-mode*.
2. Descarregar el codi complet i fer canvis al nucli de l'aplicació per adaptar-la a les nostres necessitats. Amb nomenclatura d'AppFuse es diu mode *full-source*.

En aquesta secció explicarem com aconseguir l'estat d'aplicació del segon mode, perquè en algun projecte pot ser necessari.

Per descarregar tot el codi s'ha d'executar la comanda:

```
$ mvn appfuse:full-source
```

I esperar una estona a que descarregui totes les llibreries restants.

En entorns Windows hi ha un *bug* que pot provocar un error durant la instal·lació. L'arrel del problema és la ruta del dipòsit local de Maven2. Si per defecte està a una ruta amb espais, donarà error a causa d'aquest *bug*. Per solucionar-ho s'ha de canviar de lloc la carpeta que conté el dipòsit local (per defecte a C:\Documents and Settings\nom usuari\.m2) i indicar manualment la nova ubicació a Maven.

Per fer-ho, primer copiarem la carpeta a una ruta nova que no contingui cap espai al seu nom, i seguidament crearem el fitxer C:\Documents and Settings\nom usuari\.m2\settings.xml i escriurem dintre:

```
<settings>
  <localRepository>Nova_ruta_completa_i_sense_espais</localRepository>
</settings>
```

A partir de llavors, cada vegada que cridem la comanda mvn, aquesta llegirà correctament el dipòsit.

Quan acabi la generació tindrem a l'arrel del nostre projecte noves carpetes amb un gran nombre de fitxers i carpetes dintre, que és el codi real de l'aplicació, i que podem modificar al nostre gust.

Per últim, comentar que a conseqüència d'haver sortit de l'*embeded-mode* cap a *full-source*, perdrem la capacitat d'actualitzar automàticament noves versions d'AppFuse, ja que els fitxers del nucli no seran coherents.

Preparant el projecte per Eclipse i NetBeans

Mitjançant Maven es poden generar els fitxers de configuració d'AppFuse pels IDEs:

- Eclipse
- NetBeans
- IDEA

Eclipse

Per un projecte d'*archetype basic* executarem la següent comanda:

```
$ mvn eclipse:eclipse
```

o aquesta altra si el tipus de projecte és d'*archetype modular*:

```
$ mvn install eclipse:eclipse
```

Seguidament, hem d'importar el projecte mitjançant Eclipse, anant al menú File -> Import -> Existing projects into workspace.

També necessitarem configurar la variable `M2_REPO` al *classpath*, ja que Eclipse necessita saber la ruta del dipòsit de Maven per resoldre les dependències. Per fer-ho tenim dues formes:

1. Executar la comanda:

```
$ mvn -Declipse.workspace="ruta_eclipse_home":add-maven-repo
```

2. Anar a Window > Preferences, Seleccionar Java > Build Path > Classpath Variables i configurar la variable `M2_REPO` igual a `~/m2/repository`

Una cop configurat ja serem capaços de compilar i executar els tests des d'Eclipse. Algunes coses a saber són:

- Per tests que modifiquin les dades s'ha d'executar periòdicament:

```
$ mvn dbunit:operation
```

per repoblar la base de dades amb informació original.

- Per fer el *deploy* de noves versions tenim alternatives:
 - Utilitzar Glassfish:

```
$ mvn exec:exec
```

- Utilitzar Jetty:

```
$ mvn jetty:run-war
```

- Utilitzar Cargo:

```
$ mvn cargo:start -Dcargo.wait=true
```

- Utilitzar l'integració amb Eclipse WTP, cas en que s'han de fer més passos. La següent secció està dedicada a aquesta integració.

Alguns aspectes que s'han de depurar a causa d'algun *bug* d'Eclipse:

- L'error "*Invalid character constant*", és degut a un *bug* d'Eclipse. No té solució, però no interfereix en la correcta creació del paquet.
- Un error amb el mètode `findAnnotations()`, causat per una incorrecta resolució de les llibreries d'Spring. El problema és que agafa les d'Spring 2.0.5 enlloc de les 2.5. La solució és modificar el fitxer `.classpath` de l'arrel del projecte i reemplaçar els literals de les versions d'Spring.
- Hi ha 6 errors deguts a altres *bugs* que no es poden solucionar de moment, però que no provoquen problemes al construir l'aplicació.
- L'extensió Maven 2 d'EasyEclipse (versió 0.10) conté un *bug* pel que no pot llegir el dipòsit si no està a la ruta per defecte. S'ha de canviar a l'última versió, la 0.12.

Amb tot aquest procediment tindrem preparada una copia per funcionar en Eclipse.

Per millorar el desenvolupament, s'aconsella instal·lar el *plugin* Spring IDE, que permet veure gràficament els *beans* d'Spring.

Integració amb Eclipse WTP

Per aconseguir una perfecta integració entre el projecte AppFuse i l'Eclipse WTP s'han de modificar certes parts de la configuració. Els canvis són els següents:

1. Utilitzar algun dels *archetypes* d'AppFuse per crear el projecte. Aquest pas haurà generat una carpeta "projecte". Si el tipus és basic hem d'anar al pas 7.
2. Dintre de la carpeta "projecte" crear una nova carpeta "pom". Copiar el fitxer "projecte/pom.xml" a dintre la nova carpeta, per tindre "projecte/pom/pom.xml".
3. Modificar el fitxer "projecte/pom.xml" i canviar l'"artifactId" a "projecte-pom".
4. Modificar el fitxer "projecte/core/pom.xml" i el fitxer "projecte/web/pom.xml" per canviar el "parent pom artifactId" i eliminar l'element `relativePath`.
5. Modificar el fitxer "projecte/pom.xml" i afegir un mòdul "pom" a la secció de mòduls, juntament amb els "core" i "web" existents (a la versió modular).
6. Modificar el fitxer "projecte/pom/pom.xml" i eliminar els mòduls "core" i "web".
7. En cas de tenir un projecte modular, executar:

```
$ mvn install eclipse:eclipse
```

o en cas de tenir un projecte bàsic:

```
$ mvn eclipse:eclipse
```

8. Editar el fitxer “projecte/web/.settings/org.eclipse.wst.common.component”:

- Eliminar la línia:

```
<wb-resource deploy-path="/" source-path="src/main/webapp"/>
```

- Afegir la línia següent, canviant “projecte” pel nom del propi projecte:

```
<wb-resource deploy-path="/" source-path="target/projecte-webapp-1.0-SNAPSHOT"/>
```

- Eliminar la línia:

```
<wb-resource deploy-path="/WEB-INF/classes" source-path="src/main/resources"/>
```

- Editar el fitxer “projecte/web/.classpath”:

- Eliminar la línia:

```
<classpathentry kind="src" path="src/main/resources"
  excluding="ApplicationResources_zh*.properties ... **/*.java"/>
```

- Afegir la línia:

```
<classpathentry kind="con"
  path="org.eclipse.jst.server.core.container/com.sun.enterprise.jst.server
  .runtimeTarget/GlassFish V2 Java EE 5"/>
```

- Afegir la línia:

```
<classpathentry kind="con"
  path="org.eclipse.jst.j2ee.internal.web.container"/>
```

9. Anar a “Preferences -> Java -> Installed JREs” i assegurar-se que el JRE configurat per defecte és un JDK i no un JRE.

10. Anar a “Preferences -> Server -> Installed Runtimes”. Dintre, seleccionar un “Apache -> Tomcat 5.5” i configurar-lo a la instal·lació local.

11. Anar a “File -> Import -> Existing Projects into Workspace” i seleccionar la carpeta del projecte. Pels projectes d'*archetype* modular, importar els projectes “core” i “web”.

12. Obrir la perspectiva J2EE. Anar a “Run -> Run On Server”.

13. En cas d'un projecte modular, copiar la configuració de l'arrancada del servidor i afegir-la a les propietats definides al fitxer “projecte/pom.xml”

Aquesta tècnica funciona (es genera el war, s'arranca el servidor i fa el *deploy*) però per un *bug* de GlassFish V2 dóna error en temps de *deploy*. En pròximes versions estarà solucionat.

Netbeans 6.0

NetBeans és capaç d'obrir automàticament el projecte, sense necessitat de crear més fitxers de configuració. El que sí necessitarem és la instal·lació del *plugin* Mevenide 2 per Maven 2.

Per instal·lar Mevenide 2, mitjançant Netbeans, seguirem els següents passos:

- Dintre de Netbeans anirem al menú Tools->Plugins.
- Una vegada dintre anirem a la pestanya "Avaible Plugins" i buscarem el Maven.
- El seleccionarem i l'instal·larem.
- Per comprovar la correcta instal·lació anirem a File->New Project i hauríem de tenir una nova categoria, Maven.

En aquest moment ja podem anar a File-> Open Project i seleccionar la carpeta on tenim el projecte d'AppFuse. A l'obrir-lo ens demanarà dos *plugins* més, i aprofitarem per afegir el suport per Spring:

- Visual Web JSF Backwards Compatibility Kit.
- JAX-RPC Web Services.
- Spring Framework Support.

Després de reiniciar el Netbeans tindrem el projecte preparat per desenvolupar.

Configuració de log4j per aplicacions del SER.RI-TIC

La configuració que seguidament es detalla està extreta d'un document intern del SER.RI-TIC (l'autor va col·laborar a la seva redacció) on explica quina és la configuració per les aplicacions web a desplegar al servidor de producció.

Format de la sortida

El format del ConversionPattern a establir pels diferents Appenders és:

```
%d{dd/MM/yyyy:HH:mm:ss.SSS} %-5p %C: %M(%L): %m\n
```

L'explicació d'aquesta configuració es pot veure a la taula 21:

Taula 21: Cadena de configuració del ConversionPattern de log4j.

CARÀCTERS DE CONVERSIÓ	SIGNIFICAT
%d{ dd/MM/yyyy:HH:mm:ss.SSS}	Timestamp
%-5p	Nivell de l'event
%C	Classe de l'invocador
%M	Mètode on s'ha invocat
%L	Número de línia
%m	Missatge emès a la crida de la comanda

Invocació de l'objecte Logger en aplicacions Java i J2EE

En fitxers Java, que no estiguin integrats en cap *framework*, el *logging* es realitzarà invocant la classe Logger sobre una propietat de classe estàtica anomenada "logger":

```
private static Logger logger =
    Logger.getLogger(((new Throwable()).getStackTrace()[0].getClassName()));
```

Per un *framework* concret s'utilitzaran els mecanismes proporcionats per aquest per instanciar l'objecte Logger.

Invocació de l'objecte Logger en pàgines JSP

Als fitxers JSP, la crida a les funcions de *logging* de log4j es realitzarà mitjançant Taglibs, mai mitjançant codi Java incrustat. Concretament, mitjançant la JSTL Log Tag Library [61].

Logging d'Exceptions

En cas d'Exception s'ha d'afegir com a segon paràmetre dels mètodes de *logging* la referència de la classe Throwable que ha provocat l'Exception, perquè al *log* també surti la traça de la pila d'execució amb l'Exception.

Estructura de carpetes

Dintre la carpeta WEB-INF de les aplicacions web crearem la següent estructura:

- WEB-INF
 - conf: Carpeta on situarem el fitxer log4j.properties.
 - logs: Carpeta on es guardaran els fitxers de *log*.

Fitxer de configuració log4j.properties

Aquest fitxer s'encarregarà del control dels Appenders, les sortides del *log*, especificant el destí dels fitxers, el format de les línies d'informació i el nivell de *logging*.

```
# Configura el nivell del root logger i addiciona appenders.
log4j.rootLogger=ERROR, fitxer
# Perquè no faci logs de debug les classes de apache commons
#log4j.logger.org.apache.commons=INFO
# consola és configurat per utilitzar ConsoleAppender.
#log4j.appender.consola=org.apache.log4j.ConsoleAppender
#log4j.appender.consola.layout=org.apache.log4j.PatternLayout
#log4j.appender.consola.layout.ConversionPattern
    =%d{dd/MM/yyyy:HH:mm:ss.SSS} %-5p %C: %M(%L): %m\n

# fitxer és configurat per utilitzar RollingFileAppender.
log4j.appender.fitxer=org.apache.log4j.DailyRollingFileAppender
log4j.appender.fitxer.File =/directori/[nom_aplicacio]/\
    WEB-INF/logs/[nom_aplicacio].log
log4j.appender.fitxer.DatePattern='.'yyyy-MM-dd
log4j.appender.fitxer.layout=org.apache.log4j.PatternLayout
log4j.appender.fitxer.layout.ConversionPattern
    =%d{dd/MM/yyyy:HH:mm:ss.SSS} %-5p %C: %M(%L): %m\n

# correu és configurat per utilitzar SMTPAppender.
#log4j.appender.correu=org.apache.log4j.net.SMTPAppender
#log4j.appender.correu.To=[nom_aplicacio]@urv.cat
#log4j.appender.correu.From=[nom_aplicacio]@urv.cat
#log4j.appender.correu.SMTPHost=smtplib.org
#log4j.appender.correu.Subject=ERROR a l'aplicació[nom_aplicacio]
#log4j.appender.correu.layout=org.apache.log4j.PatternLayout
#log4j.appender.correu.layout.ConversionPattern
    =%d{dd/MM/yyyy:HH:mm:ss.SSS} %-5p %C: %M(%L): %m\n
```

Classe de reinicialització del sistema de *logging*

L'aplicació web haurà de comptar amb una classe (un Servlet) que permeti executar una relectura del fitxer de configuració `log4j.properties`. Aquesta funcionalitat es necessària per poder canviar el nivell de *logging* en temps real, i sense necessitat de la col·laboració dels administradors del sistema.

Exemple de classe:

```
package cat.urv.logging;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.log4j.PropertyConfigurator;

public class Log4jInitServlet extends HttpServlet {

    String log4jConfigPath = null;

    public Log4jInitServlet() {
        super();
    }

    public void init() throws ServletException {
        String prefix = getServletContext().getRealPath("/");
        String file = getInitParameter("log4j-config");
        if (file != null) {
            System.out.println(this.getClass().getName() + ": 'log4j-config'
servlet init-param: " + file);
            this.log4jConfigPath = prefix + file;
            configure();
        }
        else {
            System.out.println(this.getClass().getName() + ": ERROR: 'log4j-
config' servlet init-param es null");
        }
    }

    /**
     * Configura el sistema de logging a partir de l'init-param 'log4j-config'
     * (al web.xml)
     * que conté la direcció de l'arxiu de configuració.
     */
    protected void configure() {
```

```
        if (this.log4jConfigPath != null) {
            PropertyConfigurator.configure(this.log4jConfigPath);
        }
        else {
            System.out.println(this.getClass().getName() + ": ERROR:
this.log4jConfigPath es null");
        }
    }

    public void destroy() {
        super.destroy();
    }

    public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        processRequest(request, response);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        processRequest(request, response);
    }

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        // reconfigurem el sistema de logging
        configure();

        out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01
Transitional//EN\">");
        out.println("<HTML>");
        out.println("  <HEAD></HEAD>");
        out.println("  <BODY>");
        out.print("    <b>Sistema de logging reconfigurat.</b>");
        out.println("  </BODY>");
        out.println("</HTML>");

        out.flush();
        out.close();
    }
}
```

Desenvolupaments amb les eines

Mètode de prova i enunciat de l'aplicació a desenvolupar

Per provar les eines, mitjançant un desenvolupament, s'han ideat un petit enunciat a resoldre diverses tecnologies explicades al capítol de disseny. Les proves no seran extensives per totes les tecnologies (queda fora de l'abast d'aquest projecte un comprovació tant extensiva), sinó que s'han elegit un conjunt representatiu del que és la solució final escollida.

La tasca a realitzar és ampliar el *framework* AppFuse amb una nova secció de gestió. Aquesta secció s'encarregarà de permetre treballar amb persones: afegir, eliminar, actualitzar, llistar... en resum, totes les operacions d'una eina CRUD.

El mètode de solució s'ha organitzat en diferents bateries de proves, per tal de comprovar les principals eines implicades al desenvolupament: els IDEs i els *frameworks* de presentació. La taula 22 resumeix aquestes bateries de proves:

Taula 22: Bateries de proves del desenvolupament.

	<i>PRESENTACIÓ</i>	<i>IDE</i>
<i>BATERIA 1</i>	Struts 2	Eclipse
<i>BATERIA 2</i>	JSF	NetBeans

Les següents característiques són comunes a les dues bateries:

- Esquelet generat amb AppFuse, cada bateria amb el respectiu *framework* de presentació. Per la generació s'utilitza un *archetype* bàsic.
- Base de dades Oracle 10g Express.
- Ús de Subversion, mitjançant els IDEs i el sistema operatiu, per emmagatzemar el codi del projecte.
- Ús del servidor d'aplicacions GlassFish.
- Accés a dades amb Hibernate, a través de Spring integrat a AppFuse.
- Logging.
- Internacionalització mitjançant el *framework* de presentació.
- Tests amb JUnit.

Desenvolupament amb Eclipse 3.3 Europa

El procés de preparació del projecte, per desenvolupar amb l'eina Eclipse, ha seguit els següents passos d'instal·lació i configuració:

- Instal·lació de Subversion.
- Creació del dipòsit Subversion.
- Instal·lació del servidor d'aplicacions GlassFish.
- Instal·lació de l'IDE, Eclipse 3.3 WTP All-in-one.
- Configuració de l'IDE per treballar amb GlassFish i Subversion.
- Creació de l'esquelet d'AppFuse amb el *framework* de presentació Struts 2. El codi d'AppFuse es deixarà en *embedded-mode*.
- Personalització d'AppFuse per l'ús d'Oracle.
- Personalització d'AppFuse per convertir-lo en un projecte Eclipse.
- Personalització d'AppFuse per adaptar-lo a un projecte Eclipse WTP.
- Importació del projecte a l'Eclipse WTP.

Tots aquests passos estan descrits en seccions anteriors d'aquest capítol.

El mètode de desenvolupament que s'utilitzarà és *test-driven*, orientat als tests. Primer es desenvoluparà el test que validi la classe que volem implementar i després s'implementarà la classe a testar. Amb aquest mètode s'aconsegueix tenir molt clar quines tasques han de realitzar aquestes proves.

El procés s'ha estructurat amb el següent ordre:

1. Creació dels POJOs utilitzant anotacions JPA.
2. Creació de la taula de la base de dades utilitzant el POJO definit i Maven.
3. Creació del *bean* DAO genèric d'Hibernate lligat al POJO creat.
4. Creació del test i la implementació d'una Business Facade.
5. Creació dels tests i accions d'Struts 2.
6. Creació dels JSPs d'entrada i modificació de dades i de llistat.
7. Internacionalització dels JSPs.
8. Configuració de la validació de formularis dels JSPs.
9. Mostrar als usuaris d'AppFuse un nou menú.

S'explicarà pas a pas la personalització del *framework*, i posteriorment es mostraran captures de pantalla del resultat obtingut: llista de persones, el formulari i la validació.

Creació dels POJOs utilitzant anotacions JPA

El primer pas a realitzar per ampliar l'esquelet d'AppFuse és la creació de l'objecte que s'ha de persistir a la base de dades. Aquest objecte el crearem com un POJO, i li afegirem les JPA Annotations necessàries per indicar la configuració que li volem donar.

S'ha de crear el fitxer “\src\main\java\cat\urv\pfcei\demo2\model\Person.java”. Tot seguit es mostra el fitxer, intercalant-se explicacions sobre el seu contingut.

```
package cat.urv.pfcei.demo2.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

import org.appfuse.model.BaseObject;
import org.apache.commons.lang.builder.CompareToBuilder;
import org.apache.commons.lang.builder.EqualsBuilder;
import org.apache.commons.lang.builder.HashCodeBuilder;
import org.apache.commons.lang.builder.ToStringBuilder;
```

Amb les JPA Annotations indicarem quins són els mapatges entre objecte i taula i entre propietats de classe i columnes de la taula de la base de dades.

La primera anotació que s'utilitza és @Entity, que indica que el següent objecte està relacionat amb una taula de la base de dades. En aquest cas, s'indica que la classe Person està relacionada amb la taula PERSON de la base de dades.

```
@Entity
public class Person extends BaseObject {
    private Long id;
    private String firstName;
    private String lastName;
```

La classe té tres camps: identificador, nom i cognom. Enlloc d'escriure les anotacions a la declaració de la propietat, s'escriuran als mètodes gets.

```
    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }
```

```
public void setId(Long id) {
    this.id = id;
}
```

A l'anotació del mètode get s'indica que aquesta propietat és la clau primària (@Id) i que el seu valor es generat per la base de dades (@GeneratedValue(strategy = GenerationType.AUTO)).

```
@Column(name="first_name", length=50)
public String getFirstName() {
    return firstName;
}
public void setFirstName(String firstName) {
    this.firstName = firstName;
}
@Column(name="last_name", length=50)
public String getLastName() {
    return lastName;
}
public void setLastName(String lastName) {
    this.lastName = lastName;
}
```

Per la resta de columnes no és necessari indicar cap anotació, serien persistides automàticament a la base de dades. Hi ha dos motius per afegir-los-hi anotacions:

- Canviar el nom de la columna i/o especificar el tipus de dades, amb @Column
- Indicar que no es vol persistència sobre aquest camp, amb @Transient.

```
public int compareTo(Object object) {
    Person myClass = (Person) object;
    return new CompareToBuilder().append(this.id, myClass.id).append(
        this.lastName,
        myClass.lastName).append(this.firstName,
        myClass.firstName).toComparison();
}
public boolean equals(Object object) {
    if (!(object instanceof Person)) {
        return false;
    }
    Person rhs = (Person) object;
    return new EqualsBuilder().append(this.id, rhs.id).append(
        this.lastName, rhs.lastName).append(this.firstName,
        rhs.firstName).isEquals();
}
public int hashCode() {
    return new HashCodeBuilder(1396090399, 1679115747).append(this.id)
        .append(this.lastName).append(this.firstName).toHashCode();
}
```

```

public String toString() {
    return new ToStringBuilder(this).append("firstName", this.firstName)
        .append("lastName", this.lastName).append("id", this.id).toString();
}
}

```

En aquest punt tenim el POJO construït i preparat per persistir. Però com ja hi ha indicada la configuració per a la base de dades es procedirà a crear-la a través de Maven.

Creació de la taula de la base de dades utilitzant el POJO definit i Maven

Primer s'ha d'indicar a Hibernate que s'ha creat un nou POJO que es vol persistir. S'ha d'obrir el fitxer "src/main/resources/hibernate.cfg.xml" i afegir la següent línia de codi:

```
<mapping class="cat.urv.pfcei.demo2.model.Person"/>
```

Després de guardar el fitxer s'ha d'executar la següent comanda de Maven:

```
$ mvn test-compile hibernate3:hbm2ddl -Poracle
```

Aquesta comanda es pot executar a través d'Eclipse, fent clic amb el botó dret al projecte -> "Run -> Maven build...", i escrivint "test-compile hibernate3:hbm2ddl" al camp Goals, i "Poracle" a Profile.

Al seguir la traça d'execució de Maven es veurà la següent línia:

```
create table person (id bigint not null auto_increment, first_name
    varchar(50), primary key (id))
```

que indica la creació de la taula dintre la base de dades.

Aprofitant que s'està treballant amb la base de dades també s'afegiran els registres d'exemple que crea dbUnit per executar els tests. S'ha d'editar el fitxer "src/test/resources/sample-data.xml" i afegir el següent codi XML:

```

<table name='person'>
  <column>id</column>
  <column>first_name</column>
  <column>last_name</column>
  <row>
    <value>1</value>
    <value>Tomas</value>
    <value>Reverter</value>
  </row>
  <row>

```



```
<value>2</value>
<value>Otaku</value>
<value>Kaworu</value>
</row>
</table>
```

Es crearan dos registres, un per fer proves amb les modificacions i l'altre per fer proves amb les eliminacions.

Creació del *bean* DAO genèric d'Hibernate lligat al POJO creat

El següent pas és crear el DAO¹⁴ que permetrà recuperar les dades de la base de dades.

AppFuse permet la creació de dos tipus de DAOs, genèrics o universals. Els genèrics permeten utilitzar els POJOs com a DAOs sense haver programat cap *bean* d'Spring. Per contra, els universals permeten personalitzar aquest *bean* segons les necessitats concretes. Al cas desenvolupat s'utilitzarà un DAO genèric.

Per indicar que es vol utilitzar un DAO genèric s'ha d'afegir el següent codi al fitxer "src/main/webapp/WEB-INF/applicationContext.xml":

```
<bean id="personDao" class="org.appfuse.dao.hibernate.GenericDaoHibernate">
    <constructor-arg value="cat.urv.pfcei.demo2.model.Person"/>
    <property name="sessionFactory" ref="sessionFactory"/>
</bean>
```

Amb aquesta declaració del bean ja tenim accessible un DAO pel POJO Person definit.

Creació del test i la implementació d'una Business Facade

Un Business Facade és similar a un Session Facade¹⁵, però sense utilitzar EJBs. Aquesta façada s'utilitza per comunicar la capa de presentació, en aquest cas l'aplicació web, amb la capa DAO. Al context d'AppFuse, aquest component s'anomena Manager.

¹⁴ *Data Access Object* (DAO), un patró de disseny que permet independitzar la lògica de l'aplicació de la implementació de la persistència de dades.

¹⁵ *Session Facade*, és un patró de disseny que defineix un component de negoci de nivell superior que centralitza les interaccions dels components de nivells inferiors.

De forma semblant als DAOs, per treballar amb operacions CRUD estàndard es pot utilitzar un `GenericManager`. Per crear-lo s'ha de substituir l'anterior codi per aquest:

```
<bean id="personManager" class="org.appfuse.service.impl.GenericManagerImpl">
  <constructor-arg>
    <bean class="org.appfuse.dao.hibernate.GenericDaoHibernate"
      autowire="byType">
      <constructor-arg value="cat.urv.pfcei.demo2.model.Person" />
    </bean>
  </constructor-arg>
</bean>
```

El codi anterior no és necessari substituir-lo, però d'aquesta forma s'assegura l'ús de la façada per utilitzar la lògica de negoci.

Creació dels tests i accions d'Struts 2

Amb nomenclatura Struts 2 les pàgines de llistat i d'entrada de dades s'anomenen *master* i *detail* respectivament. Amb la configuració d'AppFuse, les pàgines *master* tenen l'habilitat d'ordenar columnes i mostrar 25 registres per pàgina. Per altra banda, les pàgines *detail* tenen el formulari d'entrada/modificació de dades i validacions de camps *server-side*.

Com s'ha comentat a la introducció, s'utilitza el mètode *test-driven* pel desenvolupament. Això significa que es començarà amb la creació dels tests que validaran la correcta execució de les classes.

Per comprovar l'Action es realitzaran els següents tests:

- Un test de cerca de tots els usuaris.
- Un test d'edició de dades.
- Un test d'emmagatzemament de dades, utilitzant una objecte `Mock`¹⁶.
- Un test d'eliminació de dades.

Es crearà el fitxer "src\test\java\cat\urv\pfcei\demo2\webapp\PersonActionTest.java":

```
package cat.urv.pfcei.demo2.webapp;
import org.appfuse.webapp.action.BaseActionTestCase;
import com.opensymphony.xwork2.ActionSupport;
import org.apache.struts2.ServletActionContext;
import org.appfuse.service.GenericManager;
import cat.urv.pfcei.demo2.model.Person;
import cat.urv.pfcei.demo2.webapp.action.PersonAction;
import org.springframework.mock.web.MockHttpServletRequest;
```

¹⁶ *Mock Object*, és un objecte que simula el comportament d'objectes reals, però de forma controlada. S'utilitza per simular les peticions a través de Servlets.

```

public class PersonActionTest extends BaseActionTestCase {
    private PersonAction action;

    @Override
    protected void setUpBeforeTransaction() throws Exception {
        super.setUpBeforeTransaction();
        action = new PersonAction();
        GenericManager personManager = (GenericManager)
            applicationContext.getBean("personManager");
        action.setPersonManager(personManager);
    }
    public void testSearch() throws Exception {
        assertEquals(action.list(), ActionSupport.SUCCESS);
        assertTrue(action.getPersons().size() >= 1);
    }
    public void testEdit() throws Exception {
        log.debug("test d'edició...");
        action.setId(1L);
        assertNull(action.getPerson());
        assertEquals("success", action.edit());
        assertNotNull(action.getPerson());
        assertFalse(action.hasActionErrors());
    }
}

```

Al mètode testEdit() es pot veure com s'utilitza la funció de *logging*, mitjançant la variable log de la classe BaseActionTestCase, la classe de qui estén la principal.

```

public void testSave() throws Exception {
    MockHttpServletRequest request = new MockHttpServletRequest();
    ServletActionContext.setRequest(request);
    action.setId(1L);
    assertEquals("success", action.edit());
    assertNotNull(action.getPerson());
}

```

Al mètode testSave() es pot veure el funcionament d'un objecte Mock.

```

// update last name and save
action.getPerson().setLastName("Updated Last Name");
assertEquals("input", action.save());
assertEquals("Updated Last Name", action.getPerson().getLastName());
assertFalse(action.hasActionErrors());
assertFalse(action.hasFieldErrors());
assertNotNull(request.getSession().getAttribute("messages"));
}

public void testRemove() throws Exception {
    MockHttpServletRequest request = new MockHttpServletRequest();
    ServletActionContext.setRequest(request);
    action.setDelete("");
}

```

```
        Person person = new Person();
        person.setId(2L);
        action.setPerson(person);
        assertEquals("success", action.delete());
        assertNotNull(request.getSession().getAttribute("messages"));
    }
}
```

Al crear aquesta classe no compilarà, ja que falta la classe `PersonAction`, però ara ja és clara la funcionalitat que ha de tenir l'acció.

El fitxer “`src/main/java/cat/urv/pfcei/demo2/webapp/action/PersonAction.java`” contindrà l'acció principal d'Struts. Aquest fitxer conté:

```
package cat.urv.pfcei.demo2.webapp.action;

import org.appfuse.webapp.action.BaseAction;
import org.appfuse.webapp.action.BaseAction;
import cat.urv.pfcei.demo2.model.Person;
import org.appfuse.service.GenericManager;
import java.util.List;

public class PersonAction extends BaseAction {
    private GenericManager<Person, Long> personManager;
    private List persons;
    private Person person;
    private Long id;

    public void setPersonManager(GenericManager<Person, Long> personManager) {
        this.personManager = personManager;
    }
    public List getPersons() {
        return persons;
    }
    public String list() {
        persons = personManager.getAll();
        return SUCCESS;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public Person getPerson() {
        return person;
    }
    public void setPerson(Person person) {
        this.person = person;
    }
    public String delete() {
        personManager.remove(person.getId());
    }
}
```

```
        saveMessage(getText("person.deleted"));
        return SUCCESS;
    }
    public String edit() {
        if (id != null) { person = personManager.get(id);
        } else { person = new Person(); }
        return SUCCESS;
    }
    public String save() throws Exception {
        if (cancel != null) { return "cancel"; }
        if (delete != null) { return delete(); }

        boolean isNew = (person.getId() == null);
        person = personManager.save(person);

        String key = (isNew) ? "person.added" : "person.updated";
        saveMessage(getText(key));

        if (!isNew) { return INPUT;
        } else { return SUCCESS;}
    }
}
```

Aquesta classe implementa la funcionalitat a que respondrà l'Action segons les peticions que l'usuari pot fer a través d'un formulari. I a més, el disseny compleix els tests dissenyats.

Per comprovar que realment compleix els tests s'ha d'executar la següent comanda Maven:

```
$ mvn test -Dtest=PersonActionTest -Poracle
```

o, a través d'Eclipse a "Run->Maven build...", escrivint "test -Dtest=PersonActionTest" a Goals i "oracle" a Profile. Amb l'execució del test es veu que s'executen els tests, i si tot va correcte mostrarà un BUILD SUCCESSFUL al final de l'execució.

Creació dels JSPs d'entrada i modificació de dades i de llistat

Una vegada creada la capa DAO i la capa de negoci (amb la façada inclosa), és hora de desenvolupar la capa de presentació.

La presentació tindrà dos parts principals: un JSP per llistar les persones i un JSP amb un formulari per l'edició.

El fitxer per llistar persones és “src/main/webapp/WEB-INF/pages/personList.jsp”, i el seu contingut és:

```
<%@ include file="/common/taglibs.jsp"%>

<head>
  <title><fmt:message key="personList.title"/></title>
  <meta name="heading" content="<fmt:message key='personList.heading' />"/>
</head>

<c:set var="buttons">
  <input type="button" style="margin-right: 5px"
    onclick="location.href='<c:url value="/editPerson.html"/>'"
    value="<fmt:message key="button.add"/>"/>
  <input type="button" onclick="location.href='<c:url
    value="/mainMenu.html"/>'"
    value="<fmt:message key="button.done"/>"/>
</c:set>

<c:out value="\${buttons}" escapeXml="false" />

<s:set name="persons" value="persons" scope="request"/>
<display:table name="persons" class="table" requestURI="" id="personList"
  export="true" pagesize="25">
  <display:column property="id" sortable="true" href="editPerson.html"
    paramId="id" paramProperty="id" titleKey="person.id"/>
  <display:column property="firstName" sortable="true"
    titleKey="person.firstName"/>
  <display:column property="lastName" sortable="true"
    titleKey="person.lastName"/>

  <display:setProperty name="paging.banner.item_name" value="person"/>
  <display:setProperty name="paging.banner.items_name" value="people"/>

  <display:setProperty name="export.excel.filename" value="Person
  List.xls"/>
  <display:setProperty name="export.csv.filename" value="Person List.csv"/>
  <display:setProperty name="export.pdf.filename" value="Person List.pdf"/>
</display:table>

<c:out value="\${buttons}" escapeXml="false" />
<script type="text/javascript">
  highlightTableRows("personList");
</script>
```

Aquest JSP utilitza diverses llibreries (Display, JSTL, XLST). S'ha de destacar la crida que realitza al PersonAction, mitjançant la línia:

```
<s:set name="persons" value="persons" scope="request"/>
```

El nom del fitxer que crea el formulari per les insercions, modificacions i eliminacions, un únic fitxer, és "src/main/webapp/WEB-INF/pages/personForm.jsp", i el seu contingut és:

```
<%@ include file="/common/taglibs.jsp"%>

<head>
  <title><fmt:message key="personDetail.title"/></title>
  <meta name="heading" content="<fmt:message key='personDetail.heading' />"/>
  >
</head>

<s:form id="personForm" action="savePerson" method="post" validate="true">
<s:hidden name="person.id" value="%{person.id}"/>

  <s:textfield key="person.firstName" required="true" cssClass="text
medium"/>
  <s:textfield key="person.lastName" required="true" cssClass="text
medium"/>

  <li class="buttonBar bottom">
    <s:submit cssClass="button" method="save" key="button.save"
theme="simple"/>
    <c:if test="${not empty person.id}">
      <s:submit cssClass="button" method="delete" key="button.delete"
onclick="return confirmDelete('person');" theme="simple"/>
    </c:if>
    <s:submit cssClass="button" method="cancel" key="button.cancel"
theme="simple"/>
  </li>
</s:form>

<script type="text/javascript">
  Form.focusFirstElement($("#personForm"));
</script>
```

Com es pot veure, Struts 2 redueix considerablement la quantitat d'HTML que s'ha d'utilitzar per crear un formulari amb el tag `<s:form>`.

Amb els fitxers JSP creats, s'ha d'indicar a Struts 2 les noves accions implementades amb el JSP. Per fer-ho, s'ha d'editar el fitxer "src/main/resources/struts.xml" i afegir el següent contingut:

```
<action name="persons" class="cat.urv.pfcei.demo2.webapp.action.PersonAction"
method="list">
  <result>/WEB-INF/pages/personList.jsp</result>
</action>

<action name="editPerson"
class="cat.urv.pfcei.demo2.webapp.action.PersonAction" method="edit">
```

```

<result>/WEB-INF/pages/personForm.jsp</result>
<result name="error">/WEB-INF/pages/personList.jsp</result>
</action>

<action name="savePerson"
class="cat.urv.pfcei.demo2.webapp.action.PersonAction" method="save">
<result name="input">/WEB-INF/pages/personForm.jsp</result>
<result name="cancel" type="redirect">persons.html</result>
<result name="delete" type="redirect">persons.html</result>
<result name="success" type="redirect">persons.html</result>
</action>

```

Amb aquest codi es crea l'enllaç entre la capa de presentació i la façada dels Actions.

Internacionalització dels JSPs

Als JSPs s'ha utilitzat comandes com, per exemple:

```
<fmt:message key="personDetail.title"/>
```

Aquest codi anirà a buscar al ResourceBundle configurat la clau personDetail.title i el mostrarà a l'usuari. Un pas restant és afegir les noves entrades dels JSPs als ResourceBundle desitjats. Per l'idioma anglès el fitxer és "src/main/resources/ApplicationResources.properties" i el nou contingut a afegir és:

```

# -- person form --
person.id=Id
person.firstName=Nom
person.lastName=Cognom

person.added=Person has been added successfully.
person.updated=Person has been updated successfully.
person.deleted=Person has been deleted successfully.

# -- person list page --
personList.title=Person List
personList.heading=Persons

# -- person detail page --
personDetail.title=Person Detail
personDetail.heading=Person Information

# -- menu title --
menu.viewPeople=View People

```

Per altres idiomes s'haurien d'afegir les mateixes entrades, però amb el valor traduït.

Configuració de la validació de formularis dels JSPs

Per tenir completada la capa de presentació només queda la validació dels camps, de forma automàtica, i per part del servidor.

El fitxer "src\main\resources\cat\urv\pfcei\demo2\model\Person-validation.xml" és el que realitza la validació en sí. El codi és:

```
<!DOCTYPE validators PUBLIC "-//OpenSymphony Group//XWork Validator
1.0.2//EN"
"http://www.opensymphony.com/xwork/xwork-validator-1.0.2.dtd">
<validators>
  <field name="person.firstName">
    <field-validator type="requiredstring">
      <message key="errors.required"/>
    </field-validator>
  </field>
  <field name="person.lastName">
    <field-validator type="requiredstring">
      <message key="errors.required"/>
    </field-validator>
  </field>
</validators>
```

Notar que els missatges d'error també estan internacionalitzats.

També es necessita indicar al PersonAction que ha de realitzar una validació per visitants:

```
<!DOCTYPE validators PUBLIC "-//OpenSymphony Group//XWork Validator
1.0.2//EN"
"http://www.opensymphony.com/xwork/xwork-validator-1.0.2.dtd">
<validators>
  <field name="person">
    <field-validator type="visitor">
      <param name="appendPrefix">false</param>
      <message/>
    </field-validator>
  </field>
</validators>
```

Mostrar als usuaris d'AppFuse un nou menú

L'últim pas abans de comprovar el correcte funcionament, és afegir una pestanya al menú dels usuaris, juntament amb els existents fins al moment.

El fitxer a editar és "src/main/webapp/WEB-INF/menu-config.xml" i afegir una nova entrada pel menú de persones. El contingut a afegir dintre d'aquest fixer és:

```
<Menu name="PeopleMenu" title="menu.viewPeople" page="/persons.html"/>
```

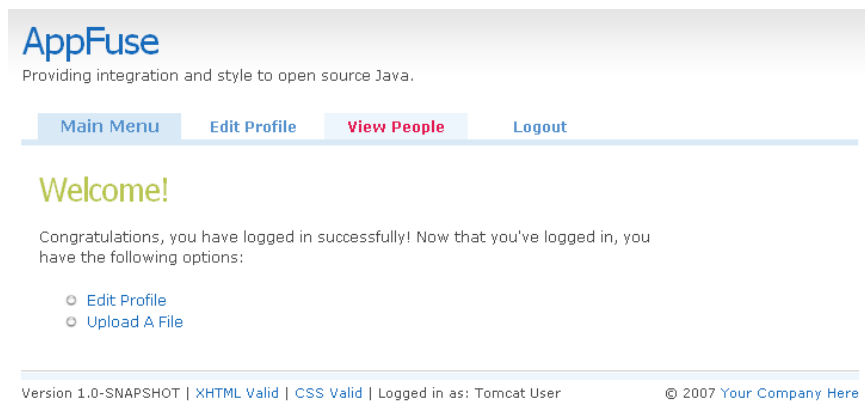
Una vegada definit el menú només quedarà mostrar-lo per pantalla modificant el fitxer on actualment es troben els menus, "src/main/webapp/common/menu.jsp" amb la línia:

```
<menu:displayMenu name="PeopleMenu" />
```

Per comprovar el correcte funcionament dels canvis desplegarem l'aplicació des d'Eclipse amb l'opció "Run -> Run on Server", acció que empaquetarà l'aplicació web en un .war, i una vegada creat, el desplegarà sobre GlassFish.

Captures del resultat

A la següent imatge es pot veure el menú "View People" afegit.



A la següent imatge podem veure el llistat de persones introduïdes amb dbUnit:

The screenshot shows the AppFuse web application interface. At the top, there is a navigation menu with links for 'Main Menu', 'Edit Profile', 'View People', and 'Logout'. Below the menu, the page title is 'Persons'. There are two buttons: 'Add' and 'Done'. A message states '2 people encontrados/as, mostrando todos/as los/as people.'. Below this is a table with three columns: 'Id', 'Nom', and 'Cognom'. The table contains two rows of data. Below the table, there are export options: 'CSV', 'Excel', 'XML', and 'PDF'. There are also 'Add' and 'Done' buttons at the bottom of the table area. The footer includes version information, validation links, and user information.

Id	Nom	Cognom
1	Tomas	Reverter
2	Otaku	Kaworu

Algunes funcionalitats a destacar d'aquesta pantalla són:

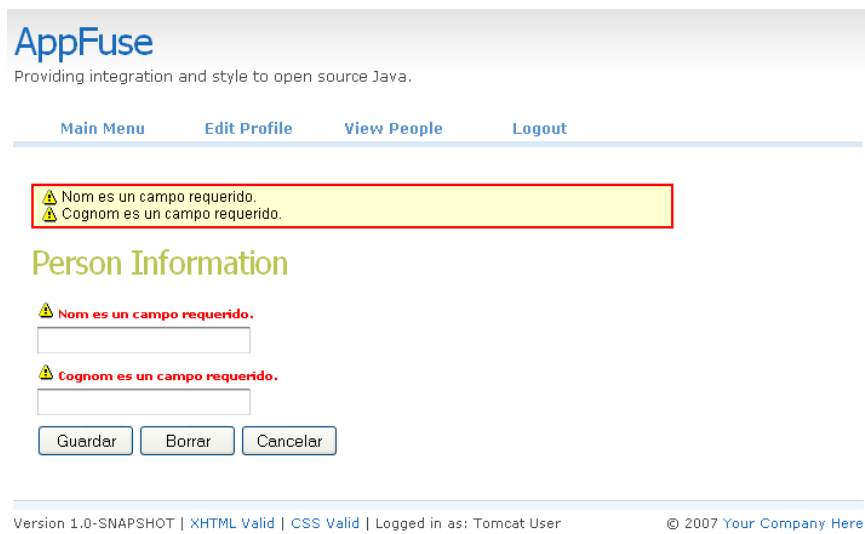
- Els elements de la capçalera de la taula són hipervíncles, amb la funcionalitat d'ordenar per aquella columna.
- La taula automàticament es pot exportar a CSV, Excel, XML i PDF.
- Els identificadors són hipervíncles, per editar la informació de l'usuari.
- El botó Add permet introduir nous registres.

La següent captura mostra el formulari d'edició de dades:

The screenshot shows the 'Person Information' form in the AppFuse web application. The form has two input fields: 'Nom *' with the value 'Tomas' and 'Cognom *' with the value 'Reverter'. Below the fields are three buttons: 'Guardar', 'Borrar', and 'Cancelar'. The footer includes version information, validation links, and user information.

Fer notar que els camps surten amb l'asterisc perquè els hem marcat com a requerits.

La següent pantalla mostra l'error de validació al intentar guardar un registre sense valors:



AppFuse
Providing integration and style to open source Java.

[Main Menu](#) [Edit Profile](#) [View People](#) [Logout](#)

⚠ Nom es un campo requerido.
⚠ Cognom es un campo requerido.

Person Information

⚠ Nom es un campo requerido.

⚠ Cognom es un campo requerido.

Version 1.0-SNAPSHOT | [XHTML Valid](#) | [CSS Valid](#) | Logged in as: Tomcat User | © 2007 [Your Company Here](#)

Amb aquesta captura es pot observar l'error que es mostra a l'usuari quan la validació de dades genera un error.

Desenvolupament amb NetBeans 6.0

El procés de preparació del projecte, per desenvolupar amb l'eina NetBeans, ha seguit els següents passos d'instal·lació i configuració:

- Instal·lació de Subversion.
- Creació del dipòsit Subversion.
- Instal·lació del servidor d'aplicacions GlassFish.
- Instal·lació de l'IDE, Eclipse 3.3 WTP All-in-one.
- Configuració de l'IDE per treballar amb GlassFish i Subversion.
- Creació de l'esquelet d'AppFuse amb el *framework* de presentació JSF. El codi d'AppFuse es deixarà en *embedded-mode*.
- Personalització d'AppFuse per l'ús d'Oracle.
- Importació del projecte al NetBeans.

Tots aquests passos estan descrits en seccions anteriors d'aquest capítol.

L'etapa inicial del desenvolupament és la mateixa que a la secció de desenvolupament en Eclipse. Les següents parts del procés són exactes:

1. Creació dels POJOs utilitzant anotacions JPA.
2. Creació de la taula de la base de dades utilitzant el POJO definit i Maven.
3. Creació del *bean* DAO genèric d'Hibernate lligat al POJO creat.
4. Creació del test i la implementació d'una Business Facade.

La resta del procés s'ha estructurat amb el següent ordre:

1. Creació dels tests i llistes de JSF.
2. Creació del fitxer XHTML que mostrarà el formulari i el llistat de resultats.
3. Internacionalització dels XHTML.
4. Configuració de la validació de formularis dels XHTML.
5. Mostrar als usuaris d'AppFuse un nou menú.

S'explicarà pas a pas la personalització del *framework*, i posteriorment es mostraran captures de pantalla del resultat obtingut: llista de persones, el formulari i la validació.

Creació dels tests i llistes de JSF

A l'igual que amb el desenvolupament amb Eclipse, les proves amb NetBeans seran *test-driven*, amb el desenvolupament dels tests abans de les implementacions.

S'ha de crear el fitxer "src\test\java\cat\urv\pfcei\demo3\webapp\action\PersonListTest.java", amb el següent contingut:

```
package cat.urv.pfcei.demo3.webapp.action;

import org.appfuse.webapp.action.BasePageTestCase;
import org.appfuse.service.GenericManager;
import org.appfuse.tutorial.model.Person;

public class PersonListTest extends BasePageTestCase {
    private PersonList bean;
    private GenericManager<Person, Long> personManager;

    public void setPersonManager(GenericManager<Person, Long> personManager)
    {
        this.personManager = personManager;
    }

    @Override @SuppressWarnings("unchecked")
    protected void setUp() throws Exception {
        super.setUp();
        bean = new PersonList();
        bean.setPersonManager(personManager);

        // add a test person to the database
        Person person = new Person();
        person.setFirstName("Abbie Loo");
        person.setLastName("Raible");
        personManager.save(person);
    }

    @Override
    protected void tearDown() throws Exception {
        super.tearDown();
        bean = null;
    }

    public void testSearch() throws Exception {
        assertTrue(bean.getPersons().size() >= 1);
        assertFalse(bean.hasErrors());
    }
}
```

Encara no compilarà, ja que falta la classe PersonList.

Amb els objectius clars, ja es pot crear la implementació. En aquest cas, el fitxer que s'ha de crear és “src/main/java/cat/urv/pfcei/demo3/webapp/action/PersonList.java”, amb:

```
package cat.urv.pfcei.demo3.webapp.action;

import java.io.Serializable;
import java.util.List;
import org.appfuse.webapp.action.BasePage;
import org.appfuse.service.GenericManager;

public class PersonList extends BasePage implements Serializable {
    private GenericManager personManager;

    public void setPersonManager(GenericManager manager) {
        this.personManager = manager;
    }
    public PersonList() {
        setSortColumn("id"); // sets the default sort column
    }
    public List getPersons() {
        return sort(personManager.getAll());
    }
}
```

El mètode sort() del mètode getPersons() crida a la implemetació de la classe BasePage. D'aquesta forma s'aconsegueix poder ordenar les columnes a través de la interfície d'usuari.

El següent pas és afegir la definició del *managed bean*¹⁷ per la classe PersonList. Per fer-ho s'ha d'afegir el següent codi al fitxer “src/main/webapp/WEB-INF/faces-config.xml”:

```
<managed-bean>
  <managed-bean-name>personList</managed-bean-name>
  <managed-bean-
class>org.appfuse.tutorial.webapp.action.PersonList</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
  <managed-property>
    <property-name>personManager</property-name>
    <value>#{personManager}</value>
  </managed-property>
</managed-bean>
```

Amb el codi creat fins ara ja es pot executar la validació de la classe:

```
$ mvn test -Dtest=PersonListTest
```

¹⁷ Un Managed Bean en JSF segueix el patró de disseny Command Pattern. Aquest patró crea una encapsulació de les accions amb l'objecte que representa

Creació del fitxer XHTML que mostrarà el formulari i el llistat de resultats

Crear el fitxer “src/main/webapp/persons.xhtml” amb el següent contingut:

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:c="http://java.sun.com/jstl/
  core"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:t="http://myfaces.apache.org/tomahawk">

  <f:view>
  <f:loadBundle var="text" basename="#{personList.bundleName}"/>
  <head>
    <title>#{text['personList.title']}</title>
    <meta name="heading" content="#{text['personList.heading']}/>
    <meta name="menu" content="PersonMenu"/>
  </head>
  <body id="personList">
  <h:form id="editPerson">

  <h:commandButton value="#{text['button.add']}" action="add" id="add"
    immediate="true" styleClass="button"/>
  <h:commandButton value="#{text['button.done']}" action="mainMenu" id="cancel"
    immediate="true" styleClass="button" style="margin-left: 5px"/>

  <!-- Error from this table is caused by http://issues.apache.org/jira/browse/
    TOMAHAWK-466 -->
  <t:dataTable id="persons" var="person" style="margin-top: 10px"
    value="#{personList.persons}" rows="25"
    sortColumn="#{personList.sortColumn}"
    sortAscending="#{personList.sortAscending}" styleClass="scrollerTable table"
    headerClass="standardTable_Header"
    rowClasses="standardTable_Row1,standardTable_Row2"
    columnClasses="standardTable_Column,standardTable_Column,standardTable_Co
    lumn,standardTable_Column,standardTable_ColumnCentered">

  <t:column>
    <f:facet name="header">
      <t:commandSortHeader columnName="id" arrow="true">
        <h:outputText value="#{text['person.id']}" />
      </t:commandSortHeader>
    </f:facet>
    <h:commandLink action="#{personForm.edit}" value="#{person.id}">
      <f:param name="id" value="#{person.id}"/>
      <f:param name="from" value="list"/>
    </h:commandLink>
  </t:column>
```



```

<t:column>
  <f:facet name="header">
    <t:commandSortHeader columnName="firstName" arrow="true">
      <h:outputText value="#{text['person.firstName']}" />
    </t:commandSortHeader>
  </f:facet>
  <h:outputText value="#{person.firstName}" escape="true"/>
</t:column>

<t:column>
  <f:facet name="header">
    <t:commandSortHeader columnName="lastName" arrow="true">
      <h:outputText value="#{text['person.lastName']}" />
    </t:commandSortHeader>
  </f:facet>
  <h:outputText value="#{person.lastName}" escape="true"/>
</t:column>

</t:dataTable>

<ui:include src="/common/tableFooter.xhtml">
  <ui:param name="tableName" value="persons"/>
</ui:include>

<script type="text/javascript">
  highlightTableRows("editPerson:persons");
</script>

</h:form>
</body>
</f:view>
</html>

```

Aquest codi mostrarà per pantalla la llista de persones, en possibilitat d'ordenar.

Ara s'ha d'afegir una regla de navegació al fitxer "src/main/webapp/WEB-INF/faces-config.xml":

```

<navigation-rule>
  <from-view-id>/persons.xhtml</from-view-id>
  <navigation-case>
    <from-outcome>add</from-outcome>
    <to-view-id>/personForm.xhtml</to-view-id>
  </navigation-case>
</navigation-rule>

```

En aquest punt ja està llesta la pantalla de llistat d'usuaris, ara es crearà el text del formulari, al fitxer "src\test\java\cat\urv\pfcei\demo3\webapp\action\PersonFormTest.java":

```
package cat.urv.pfcei.demo3.webapp.action;

import cat.urv.pfcei.demo3.model.Person;
import org.appfuse.webapp.action.BasePageTestCase;
import org.appfuse.service.GenericManager;

public class PersonFormTest extends BasePageTestCase {
    private PersonForm bean;
    private GenericManager<Person, Long> personManager;

    public void setPersonManager(GenericManager<Person, Long> personManager)
    {
        this.personManager = personManager;
    }

    @Override
    protected void setUp() throws Exception {
        super.setUp();
        bean = new PersonForm();
        bean.setPersonManager(personManager);
    }

    @Override
    protected void tearDown() throws Exception {
        super.tearDown();
        bean = null;
    }

    public void testAdd() throws Exception {
        Person person = new Person();
        // set required fields
        person.setFirstName("firstName");
        person.setLastName("lastName");
        bean.setPerson(person);

        assertEquals("list", bean.save());
        assertFalse(bean.hasErrors());
    }

    public void testEdit() throws Exception {
        log.debug("testing edit...");
        bean.setId(1L);

        assertEquals("edit", bean.edit());
        assertNotNull(bean.getPerson());
        assertFalse(bean.hasErrors());
    }
}
```

```

    /*
    public void testResourceBundle() {
        ResourceBundle bundle =
        ResourceBundle.getBundle(Constants.BUNDLE_KEY, Locale.ENGLISH,
        this.getClass().getClassLoader());
        log.debug("bundle: " + bundle.getString("person.added"));

        assertNotNull(bean.getText("person.added"));
        log.debug("bean.added: " + bean.getText("person.added"));
    }*/

    public void testSave() {
        bean.setId(1L);

        assertEquals("edit", bean.edit());
        assertNotNull(bean.getPerson());
        Person person = bean.getPerson();

        // update fields
        person.setFirstName("firstName");
        person.setLastName("lastName");
        bean.setPerson(person);

        assertEquals("edit", bean.save());
        assertFalse(bean.hasErrors());
    }

    public void testRemove() throws Exception {
        Person person = new Person();
        person.setId(2L);
        bean.setPerson(person);

        assertEquals("list", bean.delete());
        assertFalse(bean.hasErrors());
    }
}

```

Aquest test encara no compilarà, falta la implementació. El fitxer que s'ha de crear perquè compili és "src/main/java/cat/urv/pfcei/demo3/webapp/action/":

```

package cat.urv.pfcei.demo3.webapp.action;

import java.io.Serializable;
import cat.urv.pfcei.demo3.model.Person;
import org.appfuse.webapp.action.BasePage;
import org.appfuse.service.GenericManager;

public class PersonForm extends BasePage implements Serializable {

```

```
private GenericManager<Person, Long> personManager;
private Person person = new Person();
private Long id;

public void setPersonManager(GenericManager<Person, Long> manager) {
    this.personManager = manager;
}

public Person getPerson() {
    return person;
}

public void setPerson(Person person) {
    this.person = person;
}

public void setId(Long id) {
    this.id = id;
}

public String delete() {
    personManager.remove(person.getId());
    addMessage("person.deleted");

    return "list";
}

public String edit() {
    if (id != null) {person = personManager.get(id);
    } else {person = new Person();}
    return "edit";
}

public String save() {
    boolean isNew = (person.getId() == null);
    personManager.save(person);

    String key = (isNew) ? "person.added" : "person.updated";
    addMessage(key);

    if (isNew) {
        return "list";
    } else {
        return "edit";
    }
}
}
```

Ara només queda indicar la creació d'aquest *bean*, als fitxers abans editats. Dintre de "src/main/webapp/WEB-INF/faces-config.xml" la següent definició:

```
<managed-bean>
  <managed-bean-name>personForm</managed-bean-name>
  <managed-bean-
class>cat.urv.pfcei.demo3.webapp.action.PersonForm</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
  <managed-property>
    <property-name>id</property-name>
    <value>#{param.id}</value>
  </managed-property>
  <managed-property>
    <property-name>personManager</property-name>
    <value>#{personManager}</value>
  </managed-property>
</managed-bean>
```

I les següents regles de validació:

```
<navigation-rule>
  <from-view-id>/personForm.xhtml</from-view-id>
  <navigation-case>
    <from-outcome>cancel</from-outcome>
    <to-view-id>/persons.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
  <navigation-case>
    <from-outcome>list</from-outcome>
    <to-view-id>/persons.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
```

L'últim pas d'aquest apartat és crear el fitxer XHTML que mostrarà el formulari:

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:c="http://java.sun.com/jstl/
core"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:t="http://myfaces.apache.org/tomahawk"
  xmlns:v="http://corejsf.com/validator">

  <f:view>
  <f:loadBundle var="text" basename="#{personForm.bundleName}"/>
  <head>
```

```

        <title>#{text['personDetail.title']}</title>
        <meta name="heading" content="#{text['personDetail.heading']}" />
        <meta name="menu" content="PersonMenu" />
    </head>
    <body id="personForm">

    <h:form id="personForm">
    <h:inputHidden value="#{personForm.person.id}" id="id" />
    <h:panelGrid columns="3">

        <h:outputLabel styleClass="desc" for="firstName"
        value="#{text['person.firstName']}" />
        <h:inputText styleClass="text medium" id="firstName"
        value="#{personForm.person.firstName}" />
        <t:message for="firstName" styleClass="fieldError" />

        <h:outputLabel styleClass="desc" for="lastName"
        value="#{text['person.lastName']}" />

        <h:inputText styleClass="text medium" id="lastName"
        value="#{personForm.person.lastName}" />
        <t:message for="lastName" styleClass="fieldError" />

        <h:panelGroup styleClass="buttonBar bottom">
            <h:commandButton value="#{text['button.save']}"
            action="#{personForm.save}" id="save" styleClass="button" />

            <c:if test="${not empty personForm.person.id}">
                <h:commandButton value="#{text['button.delete']}"
                action="#{personForm.delete}"
                id="delete" styleClass="button" />
            </c:if>

            <h:commandButton value="#{text['button.cancel']}" action="cancel"
            immediate="true"
            id="cancel" styleClass="button" />
        </h:panelGroup>
        <h:outputText /><h:outputText />
    </h:panelGrid>
    </h:form>

    <script type="text/javascript">
        Form.focusFirstElement($('personForm'));
        highlightFormElements();
    </script>

    </body>
</f:view>
</html>

```

Els fixers de codi ja estan preparats.

Internacionalització dels XHTML

Per internacionalitzar aquesta versió s'ha d'incloure les noves claus a fitxer de recursos, en concret, per l'anglès és "src/main/resources/ApplicationResources.properties" amb el contingut:

```
# -- person form --
person.id=Id
person.firstName=Nom
person.lastName=Cognom
person.added=Person has been added successfully.
person.updated=Person has been updated successfully.
person.deleted=Person has been deleted successfully.

# -- person list page --
personList.title=Person List
personList.heading=Persons

# -- person detail page --
personDetail.title=Person Detail
personDetail.heading=Person Information

# -- menu name --
menu.viewPeople=View People
```

Configuració de la validació de formularis dels XHTML

La validació en JSF es crea directament sobre el codi XHTML. S'haurà de substituir el codi anterior pel següent:

```
<h:form id="personForm" onsubmit="return validatePersonForm(this)">
<h:inputHidden value="#{personForm.person.id}" id="id"/>

<h:panelGrid columns="3">
  <h:outputLabel styleClass="desc" for="firstName"
  value="#{text['person.firstName']}" />
  <h:inputText styleClass="text medium" id="firstName"
  value="#{personForm.person.firstName}" required="true">
    <v:commonsValidator type="required"
    arg="#{text['person.firstName']}" />
  </h:inputText>
  <t:message for="firstName" styleClass="fieldError" />

  <h:outputLabel styleClass="desc" for="lastName"
  value="#{text['person.lastName']}" />
  <h:inputText styleClass="text medium" id="lastName"
  value="#{personForm.person.lastName}" required="true">
    <v:commonsValidator type="required" arg="#{text['person.lastName']}" /
  >
```

```

</h:inputText>
<t:message for="lastName" styleClass="fieldError"/>

<h:panelGroup styleClass="buttonBar bottom">
    <h:commandButton value="#{text['button.save']}"
action="#{personForm.save}" id="save" styleClass="button"/>

    <c:if test="${not empty personForm.person.id}">
    <h:commandButton value="#{text['button.delete']}"
action="#{personForm.delete}"
id="delete" styleClass="button" onclick="bCancel=true; return
confirmDelete('Person')"/>
    </c:if>

    <h:commandButton value="#{text['button.cancel']}" action="cancel"
immediate="true"
id="cancel" styleClass="button" onclick="bCancel=true"/>
</h:panelGroup>
<h:outputText/><h:outputText/>
</h:panelGrid>
</h:form>

```

Mostrar als usuaris d'AppFuse un nou menú

Aquest pas és com el d'Eclipse, s'han d'editar els mateixos fitxers amb els mateixos valors, ja que basa la seva construcció amb la direcció URL.

Captures del resultat

A la següent imatge es pot veure el menú "View People" afegit i la llista de persones:

AppFuse
Providing integration and style to open source Java.

[Main Menu](#) [Edit Profile](#) [View People](#) [Logout](#)

Persons

Id	First Name	Last Name
<u>2</u>	Otaku	Kaworu
<u>1</u>	Tomas	Reverter

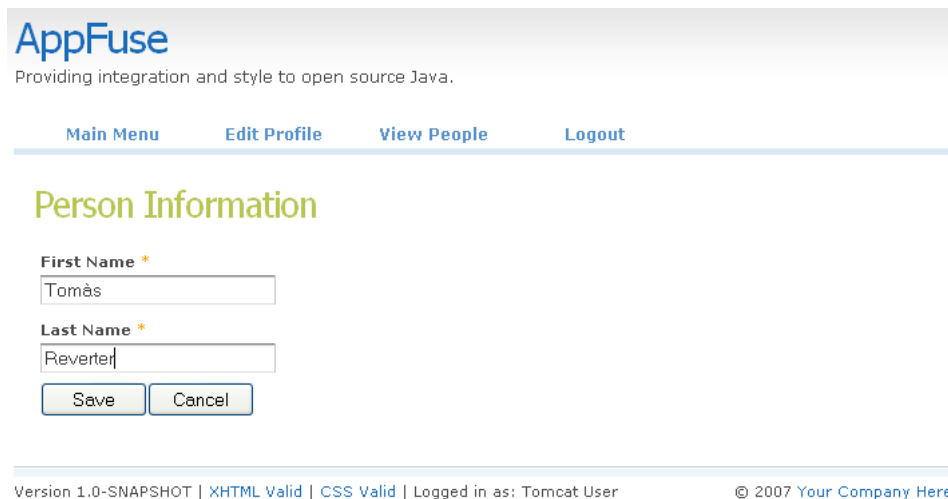
⏪ ⏩ ⏴ ⏵

Version 1.0-SNAPSHOT | [XHTML Valid](#) | [CSS Valid](#) | Logged in as: Tomcat User © 2007 [Your Company Here](#)

Algunes funcionalitats a destacar d'aquesta pantalla són:

- Els elements de la capçalera de la taula són hipervíncles, amb la funcionalitat d'ordenar per aquella columna.
- Els identificadors són hipervíncles, per editar la informació de l'usuari.
- El botó Add permet afegir nous registres.
- Els botons blaus i vermells permeten recórrer les pàgines de registres.

La següent captura mostra el formulari d'edició de dades:



The screenshot shows a web application interface for AppFuse. At the top, there is a navigation bar with links for 'Main Menu', 'Edit Profile', 'View People', and 'Logout'. Below this, the page title is 'Person Information'. The form contains two text input fields: 'First Name' with the value 'Tomàs' and 'Last Name' with the value 'Reverter'. Below the fields are 'Save' and 'Cancel' buttons. At the bottom of the page, there is a footer with the text 'Version 1.0-SNAPSHOT | XHTML Valid | CSS Valid | Logged in as: Tomcat User' and '© 2007 Your Company Here'.

I per últim, la validació dels camps, que es mostra a través d'un missatge del navegador:



Desenvolupament amb eines de generació d'informes

A l'apartat d'eines de generació d'informes del capítol de disseny s'ha escollit entre dues solucions: JReport o Eclipse BIRT. En aquest apartat de desenvolupament només es faran proves amb Eclipse BIRT. El motiu, JReport és una eina que ja ha estat provada, amb escriu, al SER.RITIC i l'autor del document té suficients coneixements d'ella per comparar-la i avaluar-la amb qualsevol altra eina.

S'utilitzarà l'esquema d'exemple HR d'Oracle [62] per tal crear informes amb dades reals. Aquest esquema ve instal·lat per defecte la versió de la base de dades que s'ha creat. Per utilitzar l'esquema s'ha de desbloquejar l'usuari HR i assignar-li una contrasenya. A la figura 8 podem veure el disseny d'aquest esquema.

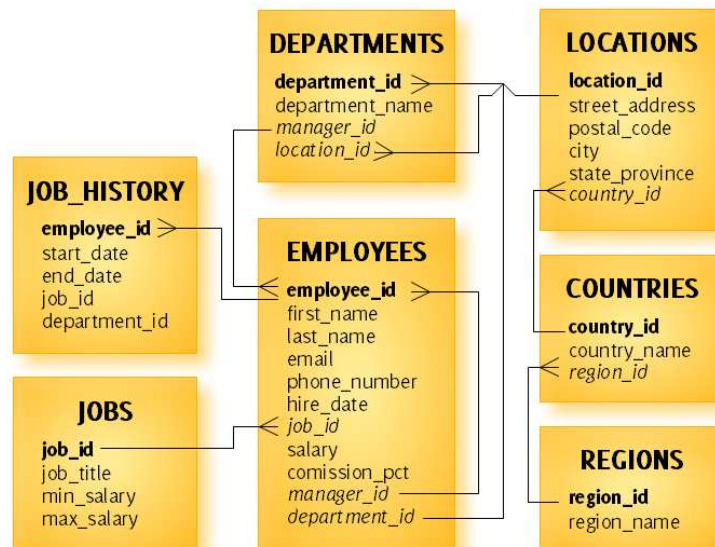


Figura 8: Disseny de l'esquema d'exemple HR

Per provar l'eina de disseny de BIRT es realitzarà un informe dels treballadors (EMPLOYEES) i dels seus departaments (DEPARTMENTS), i a més, que treballin a un cert país (COUNTRIES), agrupat per la ciutat on està la seu (LOCATIONS).

Configuració del driver JDBC d'Oracle i l'esquema HR

Per tal de fer les proves amb bases de dades Oracle necessitem instal·lar el driver a dissenyador també. A part del driver també configurarem un origen de dades amb l'esquema d'exemple HR. El procediment és el següent:

1. Dintre del BIRT Report Designer, fer click al primer botó de la barra d'eines (New), desplegar *Connection Profiles* i seleccionar *Connection Profile*. Fer clic a *Next*.
2. Seleccionar *BIRT JDBC Data Source*. Fer clic a *Next*.
3. Escriure un nom i una descripció. Fer clic a *Next*.
4. Fer clic al botó *Manage Drivers*.
5. Fer clic al botó *Add...* i seleccionar el fitxer *.jar* del driver d'Oracle, *ojdbc14.jar*. Fer clic a *Ok*. Per comprovar que s'han instal·lat correctament els drivers anirem a la pestanya *Drivers* i mirarem per l'existència de les entrades d'Oracle. Fer clic a *Ok*.
6. Escriurem la següent configuració:
 - Driver Class: `oracle.jdbc.OracleDriver (v10.2)`
 - Database URL: `jdbc:oracle:thin:@localhost:1521:XE`
 - User Name: HR
 - Password: el que s'hagi posat a l'anterior pas

Per assegurar que la connexió funciona farem clic al botó *Test Connection*. Si tot ha anat bé sortirà una finestra de configuració confirmant la correcta connexió. Farem clic a *Ok* i *Next*.

7. A l'última finestra surt una pantalla de resum de l'origen de dades. Clic a *Finish*.

Arribats a aquest punt ja tindrem preparat l'origen de dades per utilitzar-lo des del el BIRT Report Designer.

Pot ser que aquest pas el toqui repetir per algun informes, però enlloc de crear un *Connection Profile* mitjançant la creació d'un *Data Source* al panell *Data Explorer*.

Creació d'un informe d'exemple

Un informes de BIRT és un fitxer del disc dur, en format XML, que emmagatzema totes les característiques que li indiquem amb el dissenyador gràfic d'informes.

Per crear un informe seguirem el següent procediment:

1. Anar a *File -> New -> New Report...*
2. Donar-li un nom a l'informe, com per exemple, `employees.rptdesign`. Si volem canviar el directori per defecte aquí tenim l'oportunitat. Fer clic a *Next*.

3. En aquesta finestra veiem les plantilles amb que ve per defecte BIRT. Són:
 1. *My First Report*, inicia una plantilla que et dóna una passeig mitjançant el tutorial de BIRT. És recomana cursar-lo per veure'n més possibilitats.
 2. *Blank Report*, crea un informe en blanc sense cap contingut predefinit.
 3. *Simple Listing*, un llistat senzill sense agrupació de dades.
 4. *Grouped Listing*, un llistat amb agrupació de dades i possibilitat de subtotals.
 5. *Dual Column Listing*, un llistat a doble columna per dues agrupacions diferents.
 6. *Chart & Listing*, un llistat senzill i un gràfic estadístic al damunt.
 7. *Dual Column Chart & Listing*, un llistat a doble columna amb gràfics.
 8. *Side by Side Chart & Listing*, un llistat a doble fila amb gràfics.
 Per aquest cas escollirem *Chart & Listing*. Fer clic a *Finish*.

En aquest moment tindrem l'informe creat, i l'editor obert, preparat pel desenvolupament. Es recomana mirar el *Brief UI Tour* [63] del tutorial de BIRT per entendre cada vista que tenim oberta a la finestra.

En aquesta secció es reproduiran la majoria de passos del tutorial esmentat, també disponible *offline* al BIRT Report Designer, per aquest motiu s'obvia la part d'explicació.

La consulta utilitzada per la generació del DataSet és:

```

SELECT e.first_name e_first_name
, e.last_name e_last_name
, e.salary
, e2.first_name m_first_name
, e2.last_name m_last_name
, d.department_name, l.city
, c.country_name
FROM EMPLOYEES e, DEPARTMENTS d
, COUNTRIES c, LOCATIONS l,
EMPLOYEES e2
WHERE
e.department_id = d.department_id
AND d.location_id = l.location_id
AND l.country_id = c.country_id
AND e.manager_id = e2.employee_id

```

A l'exemple elaborat s'han provat filters, groups, gràfics, agrupacions i paràmetres en temps d'execució. També combinació d'informes i incrustació d'uns dintre dels altres.

Un aspecte negatiu que s'ha trobat és amb les fonts. No es possible personalitzar el tipus de font dels gràfics, només agafa els tipus de fonts del sistema.

La disposició de l'informe des de l'editor a quedat de la forma mostrada a la figura 9:

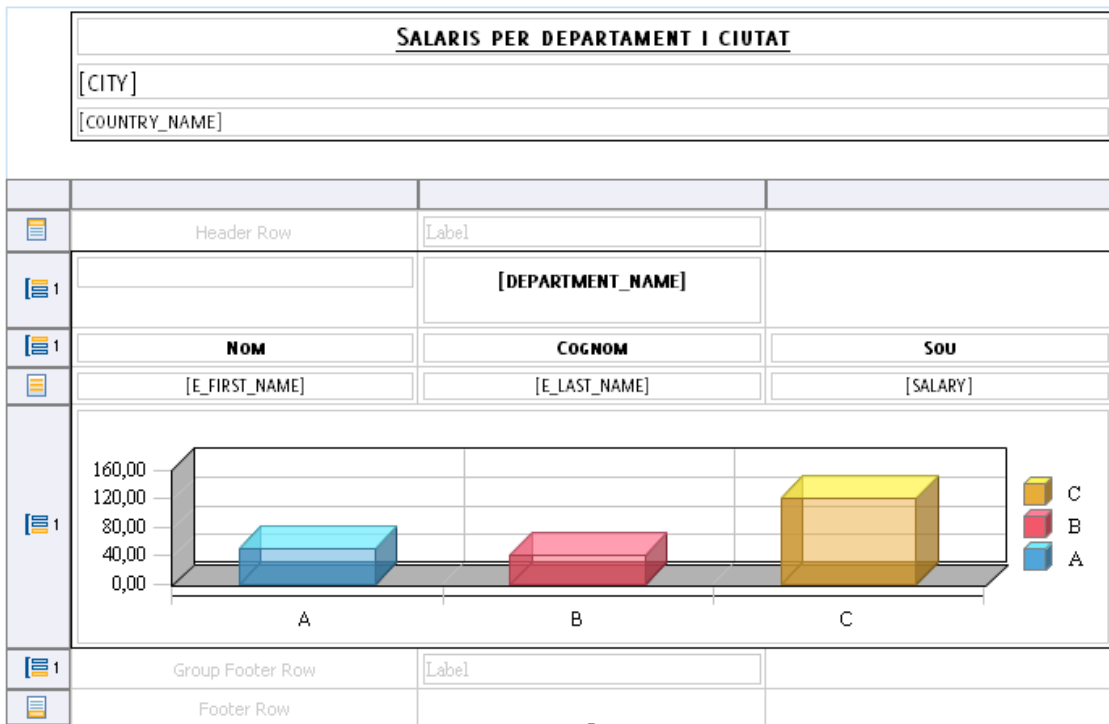


Figura 9: Disseny del layout per la generació de l'informe

I l'informe resultat de l'execució a la figura 10, amb els paràmetres CITY='Seattle' AND DEPARTMENT_NAME='Finance':

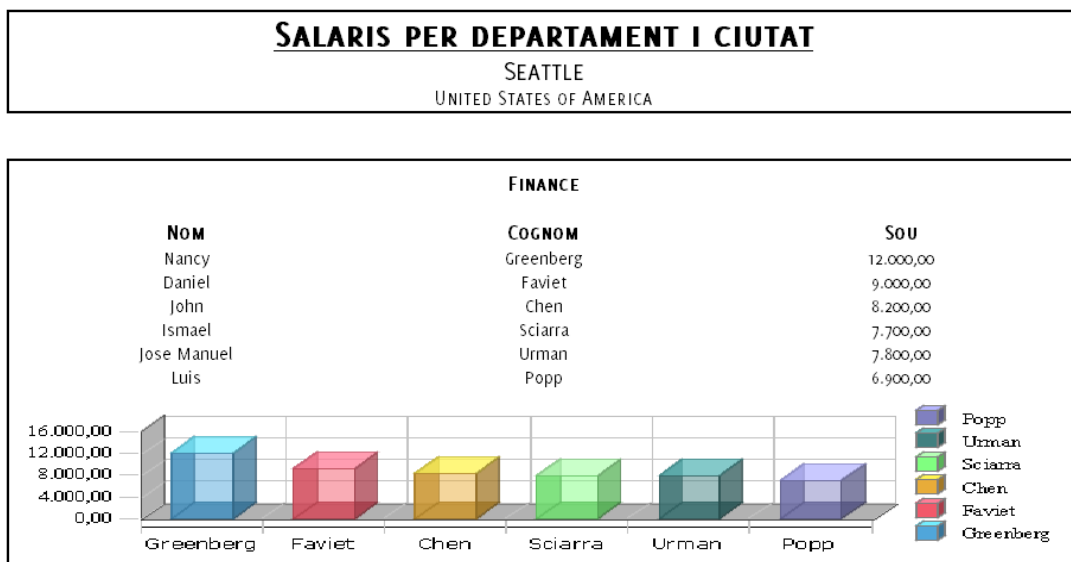


Figura 10: Resultat de l'informe de BIRT generat amb PDF

Els formats de sortida dels informes, en la versió 2.2.1 són:

- DOC
- PostScript
- PDF
- HTML
- XLS
- PPT

Destacar el bon funcionament en fitxers XLS, ja que les columnes són adaptades correctament, i el format es conserva. Els gràfics no es generen, però es copien les dades en diferents fulles per si després es vol generar manualment.

El fitxer creat durant l'exemple es pot trobar als medis de suport digital acompanyats de la documentació, amb el nom *employees221.rptdesign*.

5. Conclusions

Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning.

Rick Cook
The Wizardry Compiled

Definició de la solució escollida

Durant l'etapa de disseny s'ha estudiat quines eren les aplicacions disponibles dintre de cada categoria enumerada a l'apartat d'especificacions. Després de l'explicació de cada eina i tecnologia se les ha avaluat comparativament, per comprovar quina era superior en funcionalitats d'ús.

Per ajudar a avaluar les eines també s'ha realitzat tot un seguit d'instal·lacions, configuracions i desenvolupaments, d'aquesta forma s'ha provat les eines i tecnologies sobre el camp de treball.

Amb aquesta informació, la de disseny i la de desenvolupament, s'ha extret una conclusió, una definició formal de les eines escollides perquè siguin l'entorn de desenvolupament del Servei d'Informàtica de la URV. Una vegada realitzada la selecció s'ha assegurat que l'entorn definit està alineat amb la planificació estratègica del SER.RI-TIC.

Inclusa amb la conclusió, però independent de les eines i llibreries, hi ha una consideració metodològica. L'autor del projecte anima a utilitzar tecnologies de desenvolupament XP¹⁸ dintre del SER.RI-TIC per tres motius:

- Afavorir la transferència de coneixements entre els membres de l'equip de treball.
- Als projectes que l'acoblament amb el client, normalment un altre servei o unitat de la URV, sigui proper, intentar verificar les especificacions inicials mitjançant la construcció de prototipus amb funcionalitats molt bàsiques.
- Facilita la transició a un entorn de treball totalment diferent al que ara s'utilitzava. Compartint les inquietuds i problemes tècnics amb els companys ajuda a veure positivament els canvis.

Les conclusions es mostraran de dues formes diferents:

- La taula 23 és el resum analític de les conclusions.
- La figura 11, on es pot veure gràficament amb un cop d'ull l'entorn de desenvolupament dissenyat.

¹⁸ *Extreme Programming (XP)*, és una metodologia de desenvolupament basada en dos principis. De forma simplificada, aquest són: trobar errades en fases inicials del procés és beneficiós i desenvolupar un codi dues persones davant la mateixa pantalla.

Taula 23: Definició de l'entorn de desenvolupament.

	ÀMBIT	ELECCIÓ	COMPLEMENTARI	COMENTARI
EINES CENTRALITZADES	Base de dades	Oracle 10g		Requisit del SER.RI-TIC
	Control de versions (SCM)	Subversion		Estàndard de facto
	Explotació informació de dipòsits	FishEye		Sense alternatives
	Gestor de bugs i issues	Jira		
	Peer reviewing	Crucible		Sense alternatives
	Gestió de projectes	dotProject		Requisit del SER.RI-TIC
	Servidor d'aplicacions	SISAS		Requisit del SER.RI-TIC
	Wiki	Confluence		
EINES CLIENT	Client Subversion	Subclipse	TortoiseSVN	Complementaris
	Creació i anàlisis de consultes	TOAD		
	Disseny de bases de dades	Oracle Designer		Requisit del SER.RI-TIC
	Disseny UML	NetBeans 6.0		
	IDE	Eclipse 3.3+Aptana		
	Navegadors	Mozilla Firefox	Internet Explorer	Complementaris
	Servidor d'aplicacions	GlassFish		
	Utilitats	Diversos		
TECNOLOGIES	Accés a dades - persistència	Hibernate	EJB	Complementaris
	Accés a LDAP	JNDI		Requisit del SER.RI-TIC
	Eines de build	Maven 2		
	Eines de generació d'informes	Eclipse BIRT	OpenReports	
	Esquelet de codi	AppFuse		Sense alternatives
	Framework de presentació	Struts 2	JSF	
	Framework de model	Spring		
	Gràfics estadístics	JFreeChart		
	I18n i l10n	Jakarta Taglibs	Framework vista	
	Logging	log4j	ChainSaw, JAMon	Estàndard de facto
	Single Sign-On	JSAMPA Client SDK		Requisit del SER.RI-TIC
	Tests de càrrega	IMeter		
	Tests de navegador	Selenium		
	Tests unitaris	JUnit		

Al mapa conceptual es pot veure d'un cop d'ull la solució elegida:

- El requisits del SER.RI-TIC, marcats amb la bandera roja
- Les eines complementàries que s'utilitzen simultàniament. Estan connectades amb una línia blava i cercles a les puntes.
- Les tecnologies a elegir, és a dir, aquelles en que segons el tipus de projecte s'escollirà una o l'altra. Aquesta elecció formarà part de les decisions de disseny. Les podem veure connectades amb una línia verda i romboides a les puntes.

Els mapes conceptuals es llegeixen en el sentit de les agulles del rellotge.

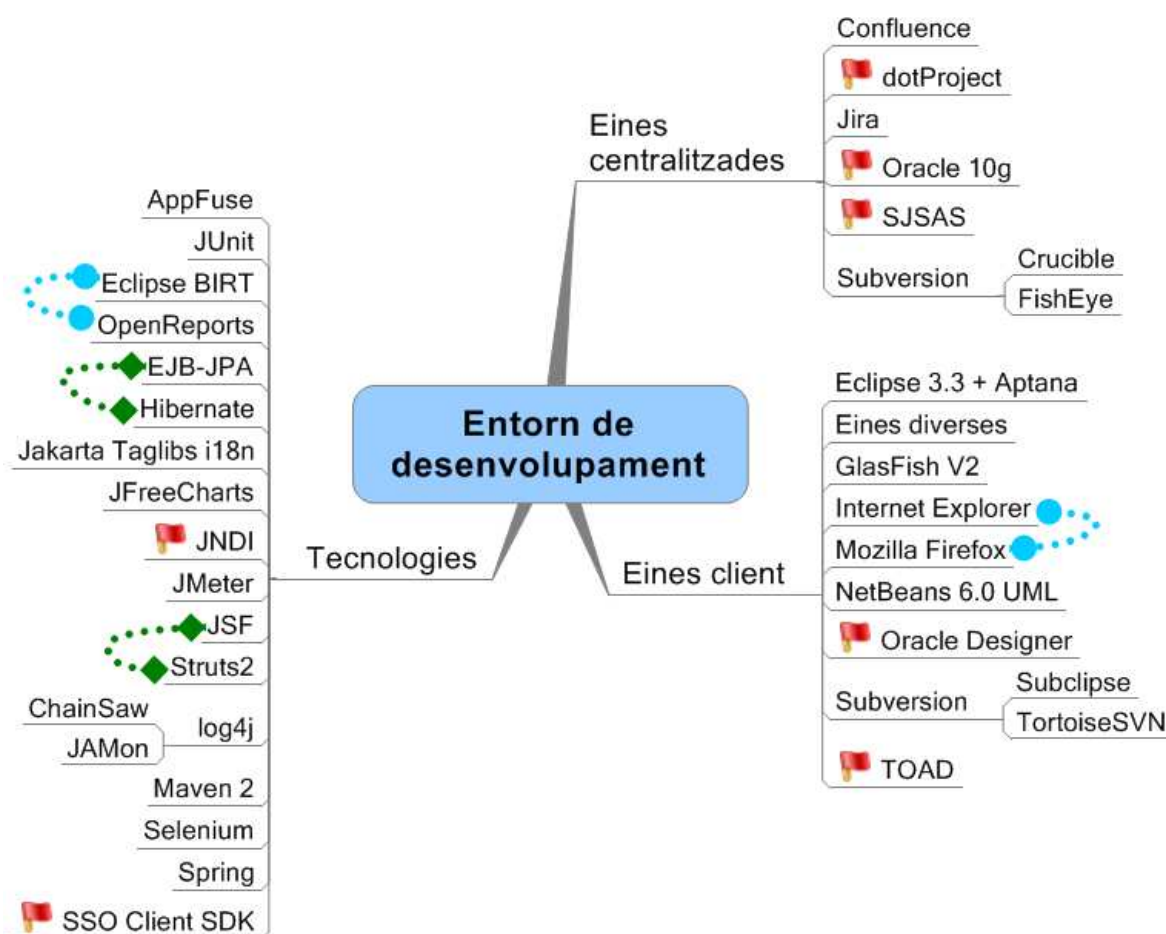


Figura 11: Mapa conceptual de la solució escollida per l'entorn de desenvolupament

Avantatges de la solució

La solució dissenyada aporta els següents avantatges:

- Entorn de desenvolupament coherent i unificat. Tots els membres treballaran amb les mateixes eines i tecnologies.
- Facilitar la integració de nous membres interns a un equip de treball, a conseqüència de tenir un entorn integrat que tots els del servei coneixeran.
- Facilitar l'entrada de nous programadors i desenvolupadors. Hi haurà un temari amb un manual, que proporcionarà una introducció a les eines i tecnologies utilitzades.
- Donar un clar marc tecnològic a les terceres empreses.
- Definició de procediments estàndards d'instal·lació i configuració de l'entorn de treball. Un desenvolupador podrà utilitzar l'ordinador d'un company i disposar de les mateixes eines.
- Ús de metodologies de desenvolupament modernes, per desenvolupar amb el paradigma d'aplicació web. L'ús de patrons de disseny, encara que sigui de forma inconscient, facilitarà les tasques de desenvolupament.
- Automatització de gran nombre de parts de l'aplicació: creació de DAOs, tests unitaris, tests d'integració, tests de navegador, internacionalització, logging, construcció de projectes...
- Centralització del codi font creat pels usuaris al dipòsit de codi. Té un doble avantatge, com a copia de seguretat i com a estudi de la feina realitzada.
- Centralització de la documentació de les etapes de desenvolupament amb el wiki. Informació molt accessible i amb facilitat per crear i editar contingut.
- Millora de la comunicació dintre d'un equip, utilitzant els canals RSS que proporcionen les diverses eines de l'entorn.
- Incorporació dintre del cicle de vida dels projectes una eina de gestió de bugs, per optimitzar aquest cicle.
- Definició d'un pla de formació, que informa dels coneixements necessaris a aconseguir per tal d'utilitzar satisfactòriament l'entorn.

Vigència de la solució

La vigència de la solució és indefinida, no té una data de final d'ús. L'avanç de les tecnologies ho decidirà.

Cicle de vida del desenvolupament d'un projecte

A la figura 12 podem veure un mapa conceptual sobre el cicle de vida del desenvolupament d'un projecte, i les eines de l'entorn de desenvolupament implicades en cada etapa.

Les línies taronja amb un quadre a l'inici indiquen les tecnologies afectades per decisions del cicle de vida del projecte, i la línia verda amb un cercle a l'inici indica la iteració d'aquests passos.. Recordar que aquest mapa es llegeix amb el sentit de les agulles del rellotge.

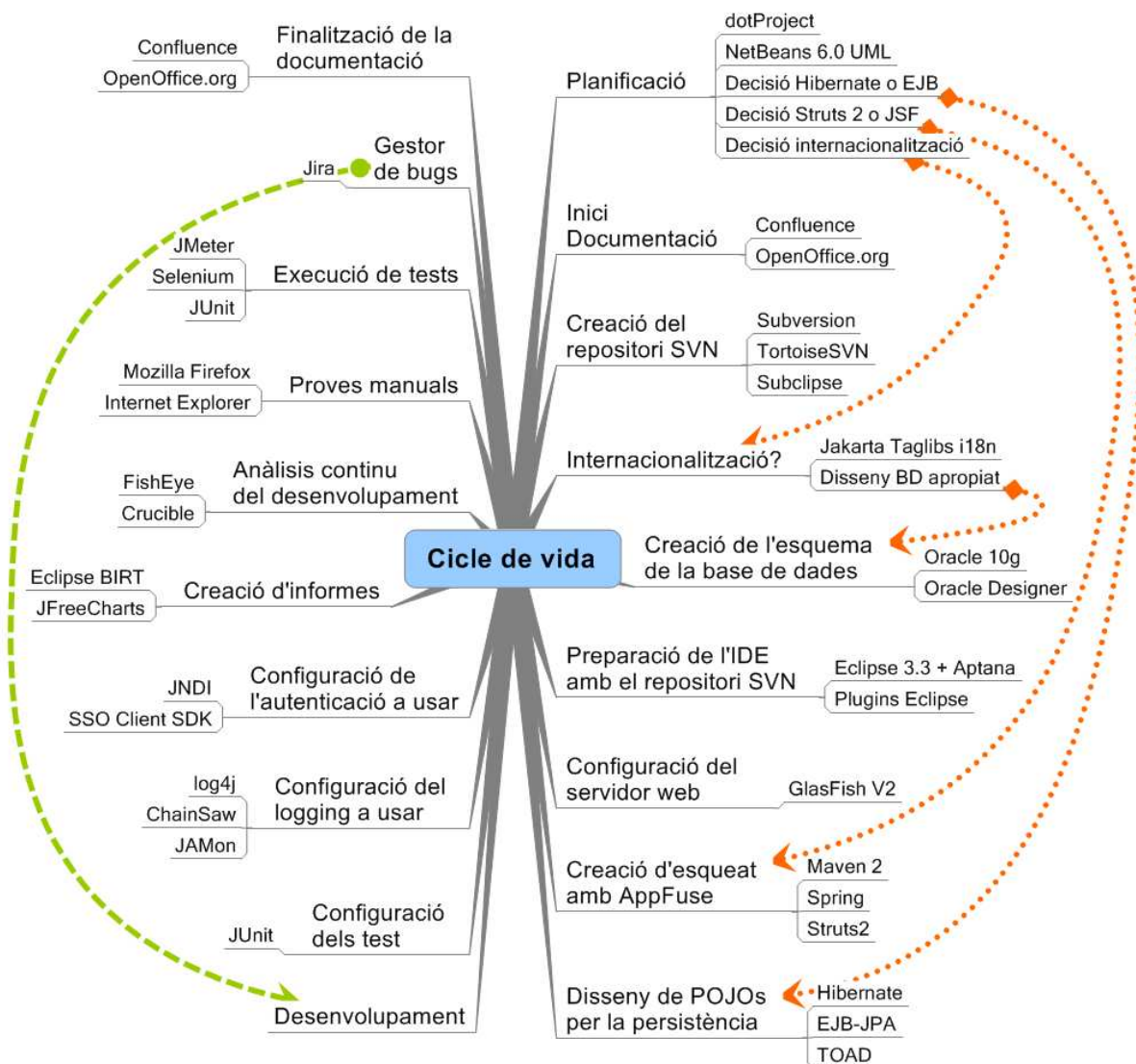


Figura 12: Mapa conceptual del cicle de vida del desenvolupament d'un projecte

Requeriments de maquinari dels desenvolupadors

Per les característiques de les eines a utilitzar de forma simultània, sorgeixen els requeriments de maquinari descrits a la taula 24.

Taula 24: Requeriments de maquinari dels desenvolupadors.

<i>ELEMENT</i>	<i>REQUISIT</i>
<i>PROCESSADOR</i>	Core2 Duo 2.0 Ghz
<i>MEMÒRIA</i>	2GB
<i>DISC DUR</i>	60 Gb
<i>MONITOR</i>	2 x 17" o 22"

Proposta de curs de formació del SER.RI-TIC

A la figura 13 podem veure un mapa conceptual sobre la proposta de curs. Recordar que aquest mapa es llegeix amb el sentit de les agulles del rellotge.

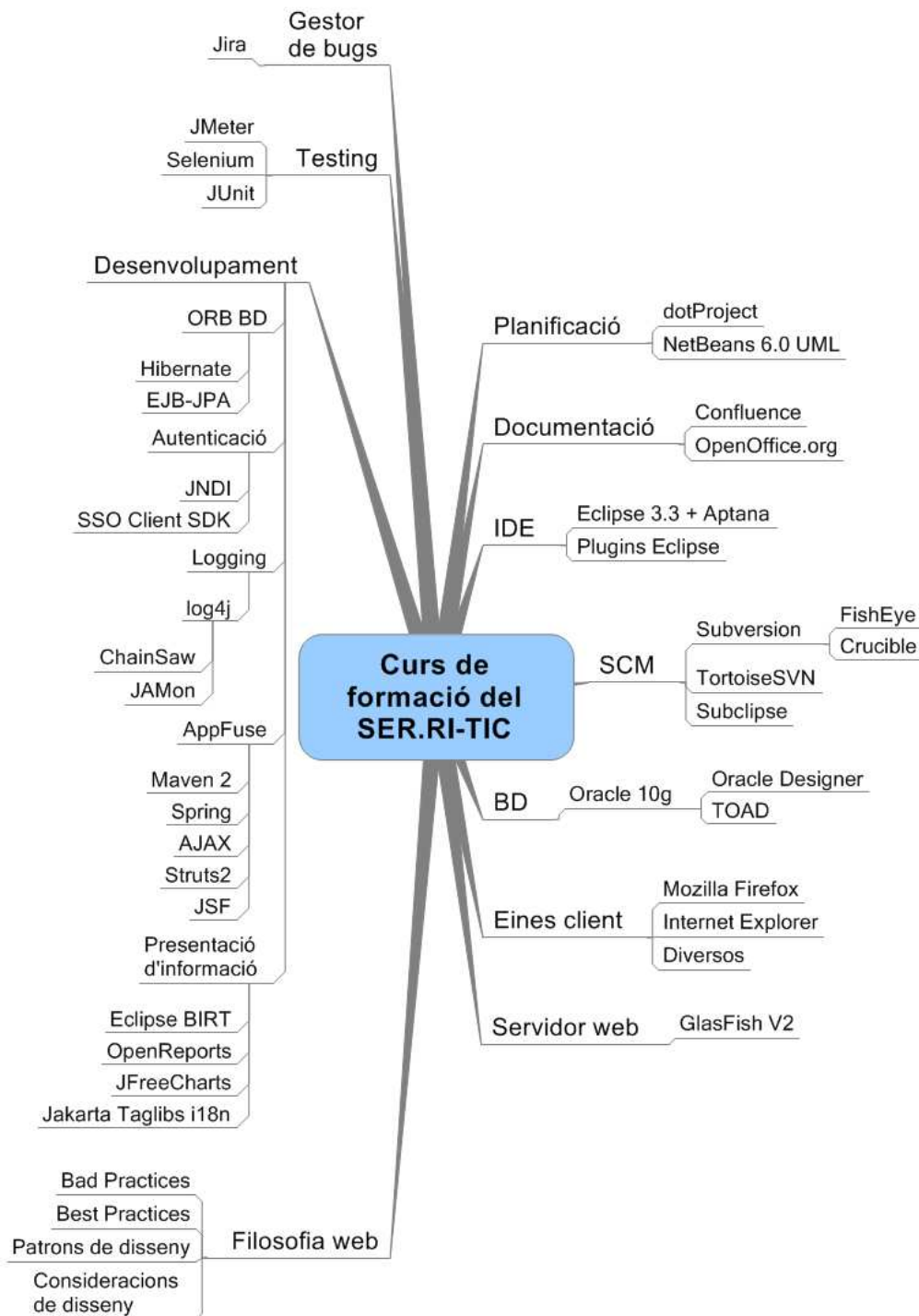


Figura 13: Mapa conceptual de la proposta de formació

La bibliografia recomanada per impartir el curs és:

- *Subversion in Action* [14].
- *Design Patterns: Elements of Reusable Object-Oriented Software* (GoF) [24].
- *Art of Java Web development* [64].
- *Spring Live* [65].
- *Appfuse Primer* (propera publicació).
- *Web Tools Platform: Developing Java Web Applications* [66].
- *The Unified Modeling Language Reference Manual (2d ed)* [67].

6. Recursos utilizats

Reading a lot of books does not guarantee good developers, but not reading any book at all (not including university' required readings) does guarantee bad developers.

"When you learn new things, learn from books"
Buu Nguyen

Bibliografia i pàgines web

- [1] Doar, Matthew B. ,Practical Development Environments, 2005
- [2] The Server Side.com, , <http://www.theserverside.com/>
- [3] JavaWorld, , <http://www.javaworld.com/>
- [4] developerWorks: IBM's resource for developers, , <http://www.ibm.com/developerworks/>
- [5] J2EE Programming (with Passion!), , <http://www.javapassion.com/j2ee/>
- [6] Reverter, Tomàs, Documentació Entorn Desenvolupament URV, 2007
- [7] Freeware for Solaris, , <http://www.sunfreeware.com/indexintel10.html>
- [8] freeURV! Disseny d'un entorn de desenvolupament lliure per la URV, , <http://www.lotomas.net/pfcblog/>
- [9] List of revision control software, , http://en.wikipedia.org/wiki/List_of_revision_control_software
- [10] Comparison of revision control software, , http://en.wikipedia.org/wiki/Comparison_of_revision_control_software
- [11] Forrester Research, Inc., The Forrester Wave™: Software Change and Configuration Management, Q2 2007,
- [12] SourceForge.net News: Subversion General Availability, , http://sourceforge.net/forum/forum.php?forum_id=544830
- [13] Next-Generation Open Source Version Control, , <http://svnbook.red-bean.com/>
- [14] Machols, J.,Subversion in Action,
- [15] FishEye at MySQL Project, , <http://svn.mysql.com/fisheye/>
- [16] FishEye at Codehaus, , <http://fisheye.codehaus.org/>
- [17] Wieggers, Karl; Moore, Peter, Lightweight Tool Support For Effective Code Reviews, 2005
- [18] Crucible - Overview demo, , <http://www.atlassian.com/software/crucible/screencast.jsp>
- [19] JIRA - Interactive webinar demo, , <http://confluence.atlassian.com/display/DEMOS>
- [20] Mindquarry, presa de contacte, , <http://www.lotomas.net/pfcblog/?p=26>
- [21] Wikipedia: Comparison of wiki software, , http://en.wikipedia.org/wiki/Comparison_of_wiki_software
- [22] Confluence: Solutions, , <http://www.atlassian.com/software/confluence/solutions/>
- [23] Confluence: Webinar, , <http://www.atlassian.com/software/confluence/webinar.jsp>
- [24] ,Design Patterns: Elements of Reusable Object-Oriented Software (GoF),
- [25] aptana.tv, , <http://aptana.tv/>
- [26] Net Applications: Top Browser Share Trend for February, 2007 to December, 2007 , , <http://marketshare.hitslink.com/report.aspx?qprid=3>

- [27] How does GlassFish V2 differ from the Sun Java System Application Server 9.1?, , <http://wiki.glassfish.java.net/Wiki.jsp?page=FaqGlassFishV2vsSJSAS91>
- [28] Core J2EE Patterns: Data Access Object Pattern, , <http://www.corej2eepatterns.com/Patterns2ndEd/DataAccessObject.htm>
- [29] Core J2EE Patterns - Data Access Objects, , <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>
- [30] JDBC Performance Best Practices, , <http://www.javalobby.org/eps/jdbc-performance/>
- [31] Transparence persistence in object-relational mapping, , http://www.service-architecture.com/object-relational-mapping/articles/transparent_persistence.html
- [32] Sun Microsystems: Java Persistence API, , <http://java.sun.com/javaee/technologies/persistence.jsp>
- [33] Loughran, Steve;Hatcher, Erik ,Ant in Action, Second Edition,
- [34] Wikipedia: Apache Maven, , http://en.wikipedia.org/wiki/Apache_Maven
- [35] Maven: Available Plugins, , <http://maven.apache.org/plugins/>
- [36] developerWorks: Exploiting Maven in Eclipse, , <http://www-128.ibm.com/developerworks/opensource/library/os-maven/index.html>
- [37] Eclipse BIRT Project Contributors, , <http://www.eclipse.org/birt/phoenix/project/contributors.php>
- [38] BIRT Exchange Community, , <http://www.birt-exchange.com/>
- [39] Walls C.; Breidenbach R.,Spring in Action, 2005
- [40] JavaServer Faces (JSF) Tutorial Net, , <http://www.jsftutorials.net/>
- [41] Geary, D.; Horstmann, C.,Core JavaServer Faces, 2004
- [42] Mann, K.,JavaServer Faces in Action, 2005
- [43] What are the fundamental differences between Struts and JSF, , <http://struts.apache.org/2.x/docs/what-are-the-fundamental-differences-between-struts-and-jsf.html>
- [44] JFreeChart Samples, , <http://www.jfree.org/jfreechart/index.html>
- [45] Sun Microsystems Documentation: Sun Java System Access Manager 7 2005Q4 Developer's Guide, , <http://docs.sun.com/app/docs/doc/819-2139>
- [46] Open Source Software Testing Tools, , <http://opensource-testing.org/>
- [47] AbstractTransactionalSpringContextTests meets TestNG, , <http://centuryminds.wordpress.com/category/testing/>
- [48] Firefox Extensions: Selenium IDE, , <https://addons.mozilla.org/es-ES/firefox/addon/2079>
- [49] Google Directory: Open Standards, , http://directory.google.com/Top/Computers/Data_Formats/Open_Standards/
- [50] SunFreeWare, , <http://www.sunfreeware.com/>
- [51] Installation Steps for Subversion 1.4.x, , <http://www.sunfreeware.com/subversion.html>

- [52] Compare Eclipse Packages, ,
<http://www.eclipse.org/downloads/moreinfo/compare.php>
- [53] NetBeans IDE 6.0 Download, , <http://download.netbeans.org/netbeans/6.0/final/>
- [54] FishEye Download Center, ,
<http://www.atlassian.com/software/fisheye/FishEyeDownloadCenter.jspa>
- [55] Apache Commons Logging, , <http://commons.apache.org/logging/>
- [56] GlassFish Eclipse Plugin, ,
<https://glassfishplugins.dev.java.net/specifications/introduction.html>
- [57] Planning Your Repository Organization, , <http://svnbook.red-bean.com/nightly/en/svn.reposadmin.planning.html#svn.reposadmin.projects.chooselayout>
- [58] Deploying to GlassFish using Maven2, , <http://technology.amis.nl/blog/?p=2495>
- [59] UserDetails from LDAP and DB, , <http://toubesen.de/appfuse/ldap-db>
- [60] LDAP-Authentication with Active Directory, , <http://toubesen.de/appfuse/ldap>
- [61] Jakarta Taglibs: JSTL Log Tag Library, , <http://jakarta.apache.org/taglibs/doc/log-doc/intro.html>
- [62] Installing the Sample Schemas and Establishing a Database Connection, ,
<http://www.oracle.com/technology/obe/obe1013jdev/common/obeconnection.htm#t1s1>
- [63] Brief UI Tour, , <http://www.eclipse.org/birt/phoenix/tutorial/basic/basic03.php>
- [64] Ford N., Art of Java Web Development, 2003
- [65] Raible M., Spring Live, 2004
- [66] Dai N., Mandel L., Ryman A., Eclipse Web Tools Platform: Developing Java Web Applications, 2007
- [67] Booch, G.; Rumbaugh, J.; Jacobson, I., Unified Modeling Language User Guide, The, 2005

Programari

El programari utilitzat per la realització del projecte ha estat:

- VMWare Server 1.0.3, per la virtualització de Solaris.
- OpenOffice 2.3 Multilingual, per la realització de la documentació.
- Mindjet MindManager Pro Trial Version, per la creació dels mapes conceptuals.
- ConceptDraw Trial Version, per la creació dels diagrames d'arquitectures.
- WordPress 2.2, per la realització del blog privat.
- DotProject, per la planificació del projecte.
- Totes les eines explicades al capítol de disseny d'aquest document.

Maquinari

Per la realització del projecte s'ha utilitzat la següent infraestructura:

- Portàtil HP Compaq nc6220, amb processador 1,7GHz i 2Gb de memòria. Aquest portàtil ha estat cedit pel SER.RI-TIC per realitzar totes les proves d'instal·lació.
- Portàtil Apple MacBook Pro, amb processador 2,33Ghz i 2Gb de memòria. Aquest portàtil es va utilitzar inicialment per les proves amb Solaris. Posteriorment s'ha utilitzat com a equip de suport.
- Servidor lotomas.net, on està instal·lat el blog privat del projecte.

7. Agraïments

El treball en equip és essencial, perquè et permet donar la culpa a l'altre.

*Llei de Murphy
Regla de Finagle*

Primer de tot vull donar el més profund agraïment a Cristina, per aconseguir aguantar-me durant la difícil època de projecte. No és senzill tenir un novio informàtic i geek...

Aquest projecte ha estat “des de” i “per al” Servei. S'ha fet pensant en tots els tècnics, perdó, amics, a qui els he d'agrair que m'ensenyin des de la seva experiència. I és que no tot en aquesta vida són APIs, jars i excels, i m'agrada tenir gent al voltant que m'ho recordi. Per això vull donar molts molts agraïments al Juanvi(!!!), Paula, Ivan, Àngeles, David, Mariú, Rafa, Teresa, Jacint, Núria, Lluís, Toni, Marcos, Jordi, Xavi... i tots els que em deixo el camí.

Per altra banda, l'objectiu final del projecte és millorar la vida de les meves “usuàries avançades”, les que tot i no ser informàtiques cada dia s'esforcen per aprendre alguna cosa nova. Així que també vull agraïr a la Blanca, Laura, Cinta, Dolors, Eli... l'ajuda moral que m'han donat per animar-me a continuar.

I per últim, però no per això menys important, al que m'ha omplert hores i hores d'explicacions sobre les fàbriques de software, al gran amic Dani. No seria el que és sense tu.

8. Llicència

Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer.

Richard Stallman
The Free Software Definition

Llicència

L'article 135 de l'estatut de la Universitat Rovira i Virgili estableix:

“l'autoria dels treballs docents realitzats per l'estudiant que siguin objecte d'avaluació en el marc dels seus estudis a la Universitat correspon exclusivament a l'estudiant, excepte en el cas de treballs dirigits o coordinats pels professors de la URV, en que serà compartida.” “En ambdós casos, la totalitat dels drets d'explotació del treball correspon a la Universitat, sense perjudici de la participació de l'autor en els beneficis eventuals.”

Tenint en compte aquestes consideracions, l'obra aquí publicada respon als drets de Llicència Creative Commons següidament descrita.



Reconeixement-Compartir amb la mateixa llicència 2.5 Espanya

Sou lliure de:

	copiar, distribuir i comunicar públicament l'obra
	fer-ne obres derivades

Amb les condicions següents:

	Reconeixement. Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).
	Compartir amb la mateixa llicència. Si altereu o transformeu aquesta obra, o en genereu obres derivades, només podeu distribuir l'obra generada amb una llicència idèntica a aquesta.

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.
- No hi ha res en aquesta llicència que menyscabi o restringeixi els drets morals de l'autor.

<http://creativecommons.org/licenses/by-sa/2.5/es/deed.ca>

Text complet de la llicència

Reconeixement-CompartirIgual 2.5 Espanya

Llicència

L'OBRA, TAL COM ES DEFINEIX MÉS ENDAVANT, S'OFEREIX SEGONS ELS TERMES D'AQUESTA LLICÈNCIA PÚBLICA DE CREATIVE COMMONS (D'ARA ENDAVANT "CCPL" O "LLICÈNCIA"). L'OBRA ESTÀ PROTEGIDA PER LA LLEI ESPANYOLA DE PROPIETAT INTEL·LECTUAL O QUALSEVOL ALTRA NORMA APLICABLE. QUEDA PROHIBIT QUALSEVOL ÚS DE L'OBRA DIFERENT DEL QUE AUTORITZI AQUESTA LLICÈNCIA O DISPOSIN LES LLEIS DE PROPIETAT INTEL·LECTUAL.

L'EXERCICI DE QUALSEVOL DRET SOBRE L'OBRA COMPORTA QUE ACCEPTEU LES LIMITACIONS I OBLIGACIONS D'AQUESTA LLICÈNCIA I HI CONSENTIU. EL LLICENCIADOR US CEDEIX ELS DRETS CONTINGUTS EN AQUESTA LLICÈNCIA, SEMPRE QUE N'ACCEPTEU LES CONDICIONS QUE HI SÓN PRESENTS.

1. Definicions

- a. L'"obra" és la creació literària, artística o científica oferta en els termes d'aquesta llicència.
- b. L'"autor" és la persona o l'entitat creadora de l'obra.
- c. Una "obra conjunta" és aquella obra susceptible de ser inclosa en alguna de les categories següents:
 - i. "Obra en col·laboració": obra creada com a resultat unitari de la col·laboració de diversos autors.
 - ii. "Obra col·lectiva": obra creada per la iniciativa i sota la coordinació d'una persona física o jurídica, que l'edita i la difon amb el seu nom, i que està constituïda per aportacions de diversos autors, la contribució personal dels quals es fon en una creació única i autònoma, per a la qual ha estat concebuda, sense que sigui possible atribuir-los separatament un dret sobre el conjunt de l'obra realitzada.
 - iii. "Obra composta i independent": obra nova que incorpora una obra preexistent sense la col·laboració de l'autor d'aquesta última.
- d. Les "obres derivades" són aquelles que es basen en una obra o en unes obres preexistents -com per exemple: les traduccions i adaptacions; les revisions, actualitzacions i anotacions; els compendis, resums i extractes; els arranjaments musicals i, en general, qualsevol transformació d'una obra literària, artística o científica-, sempre que l'obra resultant no tingui el caràcter d'obra conjunta. Per exemple, si l'obra consisteix en una composició musical o en un enregistrament de sons, la sincronització temporal de l'obra amb una imatge en moviment (*synching*) serà considerada com una obra derivada als efectes d'aquesta llicència.
- e. Es consideraran "obres audiovisuals" les creacions expressades mitjançant una sèrie d'imatges associades, amb o sense sonorització incorporada, així com les composicions musicals, que estiguin essencialment destinades a ser difoses a través d'aparells de projecció o per qualsevol altre mitjà de comunicació pública de la imatge i del so, independentment de la naturalesa dels suports materials d'aquestes obres.
- f. El "licenciador" és la persona o l'entitat que ofereix l'obra subjecta als termes d'aquesta llicència i que en cedeix els drets d'explotació conforme al que s'hi disposa.
- g. "Vós" sou la persona o l'entitat que exerceix els drets cedits mitjançant aquesta llicència i que prèviament no hagi violat els termes d'aquesta llicència respecte a l'obra, o que hagi rebut un permís exprés del licenciador per exercir els drets cedits mitjançant aquesta llicència a pesar d'una violació anterior.

- h. La "**transformació**" d'una obra comprèn la traducció, l'adaptació i qualsevol altra modificació de la qual derivi una obra diferent. Quan es tracti d'una base de dades, tal com es defineix més endavant, reordenar-la també es considerarà una transformació. La creació resultant de la transformació d'una obra tindrà la consideració d'obra derivada.
- i. S'entén per "**reproducció**" la fixació de l'obra en un mitjà que en permeti la comunicació i l'obtenció de còpies de tota l'obra o d'una part.
- j. S'entén per "**distribució**" el fet de posar a disposició del públic l'original o còpies de l'obra mitjançant la venda, el lloguer, el préstec o qualsevol altre sistema.
- k. S'entén per "**comunicació pública**" qualsevol acte mitjançant el qual una pluralitat de persones pot tenir accés a l'obra sense que se'ls hagi fet una distribució prèvia d'exemplars. No es considerarà pública, aquella comunicació que es faci dins d'un àmbit estrictament domèstic que no estigui integrat o connectat a una xarxa de difusió de qualsevol tipus. Als efectes d'aquesta llicència es considerarà comunicació pública el fet de posar l'obra a disposició del públic, ja sigui per procediments electrònics amb fil o sense fil. S'hi inclou també el fet de posar l'obra a disposició del públic de manera que qualsevol persona pugui accedir-hi des d'on vulgui i quan vulgui.
- l. L"**explotació**" de l'obra comprèn la reproducció, la distribució, la comunicació pública i la transformació.
- m. Es consideraran "**bases de dades**" les col·leccions d'obres alienes, de dades o d'altres elements independents -com ara les antologies i les bases de dades pròpiament dites- que, per la selecció o disposició dels seus continguts, constitueixin creacions intel·lectuals, sense perjudici, en aquest cas, dels drets que puguin subsistir sobre aquests continguts.
- n. Els "**elements de la llicència**" són les característiques principals de la llicència segons la selecció feta pel llicenciador i indicades anteriorment en el títol: reconeixement d'autoria (Reconeixement) i obra derivada compartida de manera igual (CompartirIgual).

2. Límits i ús legítim dels drets. No hi ha res en aquesta llicència que pretengui reduir o restringir qualsevol dels límits legals dels drets exclusius del titular dels drets de propietat intel·lectual d'acord amb la Llei de propietat intel·lectual o qualsevol altra normativa aplicable, ja siguin drets derivats d'usos legítims -com ara el dret de còpia privada o el dret de cita- o altres limitacions -com la que deriva de la primera venda d'exemplars.

3. Concessió de llicència. Conforme a les condicions d'aquesta llicència, el llicenciador concedeix (durant tota la vigència dels drets de propietat intel·lectual) una llicència d'àmbit mundial, sense dret de remuneració, no exclusiva i indefinida, que inclou la cessió dels drets següents:

- a. Dret de reproducció, distribució i comunicació pública de l'obra.
- b. Dret a incorporar l'obra en una o més obres conjuntes o bases de dades, i dret a reproduir-la incorporada en aquestes obres conjuntes o bases de dades.
- c. Dret a efectuar qualsevol transformació de l'obra, i crear i reproduir obres derivades.
- d. Dret de distribució i comunicació pública de còpies o d'enregistraments de l'obra, incorporada en obres conjuntes o bases de dades.
- e. Dret de distribució i comunicació pública de còpies o d'enregistraments de l'obra, mitjançant una obra derivada.
- f. Per evitar dubtes, sense perjudici de l'autorització preceptiva del llicenciador, i especialment quan l'obra sigui una obra audiovisual, el llicenciador renuncia al dret exclusiu de percebre, tant individualment com mitjançant una entitat de gestió de drets, o diverses (per exemple: SGAE, Dama, VEGAP), els drets d'explotació de l'obra, així com els derivats d'obres derivades, conjuntes o bases de dades, si aquesta explotació té com a objectiu l'obtenció d'un benefici mercantil o una remuneració monetària privada.

Aquests drets es poden exercir en tots els mitjans i formats, tangibles o intangibles, coneguts o per conèixer. Els drets esmentats inclouen el dret a efectuar les modificacions que siguin tècnicament necessàries per exercir els drets en altres mitjans i formats. Tots aquells drets no cedits expressament pel llicenciador queden reservats.

4. Restriccions. La cessió de drets que suposa aquesta llicència està subjecta i limitada a les restriccions següents:

- a. Podeu reproduir, distribuir o comunicar públicament l'obra només en els termes d'aquesta llicència i n'heu d'incloure una còpia, o el corresponent identificador uniforme de recursos (URI), en cada còpia o enregistrament de l'obra que reproduïu, distribuïu o comuniqueu públicament. No podeu oferir o imposar cap terme sobre l'obra que alteri o restringeixi els termes d'aquesta llicència o l'exercici dels drets per part dels seus cessionaris. No podeu sublllicenciar l'obra. Heu de mantenir intactes tots els avisos referents a aquesta llicència i a l'absència de garanties. No podeu reproduir, distribuir o comunicar públicament l'obra amb mesures tecnològiques que controlin l'accés o l'ús de l'obra d'una manera contrària als termes d'aquesta llicència. Això últim s'aplica a una obra incorporada en una obra conjunta o base de dades, però no implica que aquestes, al marge de l'obra subjecta a aquesta llicència, hagin d'estar subjectes als mateixos termes. Si creeu una obra conjunta o base de dades, amb l'avís previ del llicenciador, haureu de treure de l'obra conjunta o base de dades qualsevol crèdit requerit en l'apartat 4c, segons el que es requereixi i fins on sigui possible. Si creeu una obra derivada, amb l'avís previ del llicenciador, haureu de treure de l'obra derivada qualsevol crèdit requerit en l'apartat 4c, segons el que es requereixi i fins on sigui possible.
- b. Podeu reproduir, distribuir o comunicar públicament una obra derivada només sota els termes d'aquesta llicència, o d'una versió posterior d'aquesta llicència amb els mateixos elements principals, o d'una llicència iCommons de Creative Commons que contingui els mateixos elements principals que aquesta llicència (per exemple, Reconeixement-Compartir Igual 2.5 Japó). Heu d'incloure una còpia d'aquesta llicència o d'alguna de les que s'han esmentat anteriorment, o bé el corresponent identificador uniforme de recursos (URI), amb cada còpia o enregistrament de l'obra que reproduïu, distribuïu o comuniqueu públicament. No podeu oferir o imposar cap terme respecte les obres derivades o a les seves transformacions que alteri o restringeixi els termes d'aquesta llicència o l'exercici dels drets per part dels seus cessionaris. Heu de mantenir intactes tots els avisos que es refereixin a aquesta llicència i a l'absència de garanties. No podeu reproduir, distribuir o comunicar públicament l'obra derivada amb mesures tecnològiques que controlin l'accés o l'ús de l'obra d'una manera contrària als termes d'aquesta llicència. Això últim s'aplica a una obra derivada incorporada en una obra conjunta o base de dades, però no implica que aquestes, al marge de l'obra subjecta a aquesta llicència, hagin d'estar subjectes als mateixos termes.
- c. Si reproduïu, distribuïu o comuniqueu públicament l'obra o qualsevol obra derivada, conjunta o base de dades que la incorpori, heu de mantenir intactes tots els avisos sobre la propietat intel·lectual de l'obra i reconèixer-ne l'autor original, de manera raonable conforme al mitjà o als mitjans que estigueu utilitzant, indicant-hi el nom (o el pseudònim, si és el cas) de l'autor original, si s'ha facilitat, i/o reconèixer-ne aquelles parts (per exemple: institució, publicació, revista) que l'autor original i/o llicenciador designin per ser reconeguts en l'avís legal, les condicions d'ús, o de qualsevol manera raonable; el títol de l'obra, si s'ha facilitat; en la mesura que sigui possible i si n'hi ha, l'identificador uniforme de recursos (URI) vinculat a l'obra que el llicenciador hagi especificat, llevat que aquest URI no es refereixi a l'avís sobre propietat intel·lectual o a la informació sobre la llicència de l'obra. En el cas d'una obra derivada, caldrà que hi feu constar un avís que identifiqui l'ús de l'obra en l'obra derivada (per exemple, "traducció catalana de l'obra de l'autor original" o "guió basat en l'obra original de l'autor original"). Aquest avís es pot desenvolupar de qualsevol manera raonable a fi que, en el cas d'una obra derivada, conjunta o base de dades, hi aparegui de forma comparable als avisos corresponents a altres autors.
- d. En el cas de la inclusió de l'obra en alguna base de dades o recopilació, el propietari o el gestor de la base de dades haurà de renunciar a qualsevol dret relacionat amb aquesta inclusió i als usos de l'obra un cop extreta de la base de dades, ja sigui de manera individual o conjuntament amb altres materials.

5. Exoneració de responsabilitat

LLEVAT QUE S'ACORDI MÚTUAMENT ENTRE LES PARTS, EL LLICENCIADOR OFEREIX L'OBRA TAL COM ÉS (ON AN "AS-IS" BASIS) I NO CONFEREIX GARANTIA DE CAP MENA RESPECTE L'OBRA O LA PRESÈNCIA O ABSÈNCIA D'ERRORS QUE ES PUGUIN, O NO, DETECTAR. ALGUNES JURISDICCIONS NO PERMETEN L'EXCLUSIÓ D'AQUESTES GARANTIES, DE MANERA QUE AQUESTA EXCLUSIÓ POT NO SER APLICABLE EN EL VOSTRE CAS.

6. Limitació de responsabilitat.

LLEVAT QUE LA LLEI APLICABLE HO DISPOSI EXPRESSAMENT I IMPERATIVAMENT, EN CAP CAS EL LLICENCIADOR SERÀ RESPONSABLE DAVANT VOSTRE PER QUALSEVOL TEORIA LEGAL, DE QUALSEVOL DANY QUE EN PUGUI RESULTAR, TANT SI ÉS GENERAL COM ESPECIAL (INCLOSOS EL DANY EMERGENT I EL LUCRE CESSANT), FORTUÏT O CAUSAL, DIRECTE O INDIRECTE, PRODUÏT PER L'ÚS DE L'OBRA DURANT EL PERÍODE EN QUÈ S'EXTÈN AQUESTA LLICÈNCIA, ENCARA QUE EL LLICENCIADOR HAGUÉS ESTAT INFORMAT DE LA POSSIBILITAT D'AQUESTS DANYS.

7. Finalització de la llicència

- a. Aquesta llicència i la cessió dels drets que conté acabaran automàticament en cas que se n'incompleixi qualsevol dels termes. No obstant això, no s'anul·larà la llicència d'aquelles persones o entitats que hagin rebut obres derivades, conjuntes o bases de dades vostres amb aquesta llicència, sempre que aquestes persones o entitats es mantinguin en el compliment íntegre d'aquesta llicència. Les seccions 1, 2, 5, 6, 7, i 8 continuaran vigents malgrat que finalitzi aquesta llicència.
- b. Conforme a les condicions anteriors, la cessió de drets d'aquesta llicència és perpètua (durant tota la vigència dels drets de propietat intel·lectual aplicables a l'obra). Malgrat això, el llicenciador es reserva el dret a divulgar o publicar l'obra en condicions diferents a les presents, o a retirar l'obra en qualsevol moment. Tanmateix, això no suposarà donar per conclosa aquesta llicència (o qualsevol altra llicència que hagi estat concedida, o calgui concedir, en els termes d'aquesta llicència), sinó que continuarà vigent i amb efectes complets, a menys que no hagi finalitzat conforme allò establert anteriorment.

8. Miscel·lània

- a. Cada vegada que exploteu d'alguna manera l'obra, l'obra conjunta o la base de dades que la incorpori, el llicenciador original ofereix als tercers i successius llicenciataris la cessió de drets sobre l'obra, en les mateixes condicions que la llicència que us ha estat concedida.
- b. Cada vegada que exploteu d'alguna manera una obra derivada, el llicenciador original ofereix als tercers i successius llicenciataris la cessió de drets sobre l'obra original, en les mateixes condicions i termes que la llicència que us ha estat concedida.
- c. Si alguna disposició d'aquesta llicència resulta invàlida o inaplicable segons la llei vigent, això no afectarà la validesa o l'aplicabilitat de la resta dels termes d'aquesta llicència i, sense cap acció addicional per a qualsevol de les parts d'aquest acord, aquesta disposició s'entendrà reformada en allò estrictament necessari per fer que aquesta disposició sigui vàlida i executiva.
- d. No s'entendrà que existeix renúncia pel que fa a algun terme o disposició d'aquesta llicència, ni que se'n consent cap violació, a menys que aquesta renúncia o consentiment figuri per escrit i sigui firmada per la part que renuncia o que consent.
- e. Aquesta llicència constitueix l'acord ple entre les parts respecte l'objecte de la llicència. No hi ha interpretacions, acords o condicions respecte l'obra que no s'especifiquin expressament en aquesta llicència. El llicenciador no estarà obligat per cap disposició complementària que pugui aparèixer en qualsevol comunicació vostra. Aquesta llicència no es pot modificar sense que ho acordeu per escrit amb el llicenciador.

Creative Commons no és part d'aquesta llicència, i no ofereix cap garantia en relació amb l'obra. Creative Commons no serà responsable davant vostre -o qualsevol part- i per qualsevol teoria legal, de danys de qualsevol tipus, tant si són generals o especials (inclosos el dany emergent i el lucre cessant), com fortuïts

o causals, que estiguin relacionats amb aquesta llicència. A pesar de las dues oracions anteriors, si Creative Commons s'identifica expressament com a llicenciador, tindrà tots els drets i obligacions del llicenciador.

Llevat del propòsit limitat d'indicar al públic que l'obra està subjecta a una llicència CCPL, cap part utilitzarà la marca registrada Creative Commons o qualsevol altra marca registrada o insígnia relacionada amb Creative Commons sense el seu consentiment per escrit. Qualsevol ús permès es farà conforme a les pautes vigents en cada moment sobre l'ús de la marca registrada per Creative Commons, publicades en el seu espai web o proporcionades a petició prèvia.

Podeu contactar amb Creative Commons a: <http://creativecommons.org/>.

Índex alfabètic

A

AJAX.....	35, 36, 37, 60, 65, 67
Annotations.....	46, 47, 65, 76, 77, 122
Ant.....	8, 35, 37, 46, 52, 53, 54, 75, 76, 91
Appfuse...7, 60, 61, 62, 77, 90, 104, 105, 106, 107, 108, 109, 110, 111, 113, 115, 120, 121, 122, 125, 126, 128, 134, 137, 138, 139, 142, 143, 148, 157, 163	
Aptana.....	36, 87, 88, 157
ArgoUML.....	32
Autenticació.....	27, 44, 64, 74

B

Bases de dades.....	22, 26, 30, 31, 33, 34, 35, 40, 45, 46, 55, 57, 61, 64, 78, 88, 90, 92, 151, 157, 175, 177
Beans.....	47, 48, 64, 81, 97, 110, 113
BIRT.....	33, 56, 58, 59, 93, 94, 95, 150, 151, 152, 157
Blog.....	13, 169, 186
Bug.....	9, 111, 113, 114
Build.....	50, 52, 112, 124, 129, 157

C

CDDL.....	5
Commit.....	16, 17, 22, 29
Comunitat.....	2, 17, 31, 33, 35, 37, 52, 57, 73
Confluence.....	7, 27, 28, 157
Control de versions.....	6, 7, 12, 15, 16, 20, 34, 35, 88, 103, 157
Crucible.....	20, 21, 157
CRUD.....	60, 120, 126
CSS.....	5, 35, 36, 38, 39, 43, 60
CVS.....	7, 16, 17, 29, 34, 35, 52, 88, 89
ChainSaw.....	73, 157
Checkout.....	16, 29, 52, 101

D

DAO.....	45, 61, 81, 121, 125, 126, 129, 137
Dipòsit.....	8, 16, 19, 26, 29, 71, 74, 85, 89, 101, 102, 103, 111, 112, 113
DotProject.....	5, 6, 13, 21, 23, 25, 157, 169

E

EasyEclipse.....	34, 87, 88, 113
Eclipse...6, 29, 32, 33, 34, 35, 36, 37, 42, 53, 56, 57, 58, 59, 60, 61, 67, 70, 76, 87, 88, 93, 100, 104, 111, 112, 113, 114, 120, 121, 124, 129, 134, 137, 138, 148, 150, 157	
Eclipse WTP.....	87, 100, 112, 113, 121
EJB.....	31, 34, 35, 37, 40, 41, 42, 45, 47, 48, 58, 63, 73, 125, 157

F

Firefox.....38, 39, 80, 157
 FishEye.....19, 20, 21, 89, 157

G

Gestió de projectes.....5, 6, 21, 157
 GlassFish...40, 41, 42, 61, 89, 91, 92, 93, 94, 97, 98, 99, 100, 104, 107, 108, 112, 114,
 120, 121, 134, 137, 157
 GPL.....5

H

Hibernate27, 34, 37, 46, 47, 48, 58, 59, 63, 64, 65, 81, 88, 95, 96, 106, 107, 120, 121,
 124, 125, 126, 137, 157
 HTML.....5, 36, 38, 39, 43, 55, 58, 80, 83, 119, 131, 154

I

I18n.....8, 60, 65, 70, 71, 157
 IDE.....6, 29, 31, 33, 34, 35, 42, 80, 87, 88, 100, 104, 113, 120, 121, 137, 157
 Integration Testing.....9, 61
 IoC.....64
 Issue.....9
 Issues.....21, 24, 140, 157

J

J2EE.1, 5, 11, 21, 27, 33, 34, 40, 41, 42, 47, 49, 55, 56, 58, 63, 67, 73, 87, 88, 89, 114,
 116
 JAMon.....73, 157
 Java.....5, 6, 7, 23, 24, 25, 26, 28, 32, 33, 34, 35, 40, 42, 45, 46, 47, 49, 52, 55, 56, 57,
 58, 60, 64, 65, 67, 69, 70, 71, 72, 73, 74, 75, 78, 79, 81, 87, 88, 89, 90, 100, 105,
 112, 114, 116, 163
 Javascript.....35, 36, 37, 38, 39, 57, 80, 130, 131, 141, 146
 JDBC.....41, 45, 57, 58, 64, 72, 73, 78, 90, 92, 93, 95, 96, 99, 106, 151, 185
 JFreeChart.....69, 157
 JIRA.....9, 24, 27
 JMeter.....9, 78, 79, 157
 JNDI.....47, 49, 63, 157
 JPA.....37, 48, 60, 64, 121, 122, 137
 JReport.....55, 59, 69, 150
 JSF.....35, 37, 61, 67, 68, 71, 104, 115, 120, 137, 138, 140, 145, 147, 157
 Jsp.....34, 35, 37, 40, 41, 61, 65, 71, 116, 129, 130, 131, 132, 134
 JUnit.....9, 60, 75, 76, 77, 78, 120, 157
 JUnitPerf.....78, 79

L

L10n.....8, 70, 157
 LDAP.....5, 21, 23, 24, 26, 27, 28, 49, 61, 74, 104, 108, 109, 110, 157

Locale.....	70, 71, 143
Log4j.....	8, 60, 72, 73, 116, 117, 118, 157
Logging.....	8, 38, 63, 72, 76, 94, 116, 117, 118, 119, 120, 127, 157, 159
Lucene.....	27, 46, 48

M

Maven 2.....	35, 53, 62, 104, 107, 108, 113, 115, 157
MediaWiki.....	7, 26, 28
Muntatge.....	50, 51, 52, 53

N

NetBeans.....	6, 29, 31, 32, 35, 36, 37, 42, 60, 61, 67, 89, 100, 104, 111, 115, 120, 137, 138, 157
---------------	---

O

Openoffice.org.....	44
OpenReports.....	58, 59, 95, 96, 97, 157
Oracle.....	5, 15, 17, 22, 24, 28, 30, 31, 41, 59, 90, 92, 93, 95, 96, 104, 105, 106, 107, 120, 121, 129, 137, 150, 151, 157, 185
ORM.....	45, 46, 48, 64, 105

P

Patr3.....	45, 60, 61, 103
PL/SQL.....	5, 58
POJO.....	45, 46, 121, 122, 124, 125, 137

R

ResourceBundle.....	59, 60, 67, 70, 71, 132, 143
RSS.....	19, 21, 22, 23, 24, 26, 27, 28, 159

S

SCM.....	7, 8, 15, 16, 21, 52, 53, 157
Selenium.....	9, 80, 157
SER.RI-TIC...2, 3, 5, 6, 11, 12, 13, 14, 15, 25, 26, 30, 34, 42, 43, 49, 55, 59, 69, 72, 73, 74, 90, 108, 116, 150, 156, 157, 158, 162, 169, 185	
SER.RI-TIC.....	69
Servidor d'aplicacions.....	5, 25, 40, 50, 74, 89, 91, 92, 93, 120, 121, 137, 157
Servlet.....	37, 40, 41, 42, 61, 66, 74, 78, 118, 119, 126, 127
SJSAS.....	40, 41, 42, 91, 100, 157
Solaris.....	12, 14, 15, 85, 101, 169
Spring...34, 37, 47, 48, 61, 63, 64, 65, 68, 71, 77, 88, 96, 113, 115, 120, 125, 157, 163	
Sql.....	5, 30, 45, 46, 57, 58, 73, 90, 95
SSO.....	74
Struts.....	35, 37, 59, 61, 65, 68, 69, 71, 81, 104, 105, 120, 121, 126, 128, 131, 157
Subclipse.....	29, 34, 88, 157
Subversion. .7, 12, 16, 17, 19, 20, 22, 23, 24, 29, 32, 35, 37, 85, 86, 89, 102, 103, 120,	

121, 137, 157, 163	
Sun Java System Access Manager.....	74
Sun Java System Application Server.....	5, 25, 40

T

Taglibs.....	67, 71, 116, 130, 131, 157
Testing.....	9, 52, 61, 77, 142
TestNG.....	9, 76, 77
The Grinder.....	79
TOAD.....	5, 30, 157
Tomcat.....	34, 41, 42, 88, 89, 114
TortoiseSVN.....	29, 157
Trac.....	9, 22, 24

U

UML.....	31, 32, 35, 89, 157
UML integrat a l'IDE NetBeans.....	31
Unit Testing.....	9, 52, 61
Update.....	16, 29, 60, 91, 96, 127, 143

W

Web Services.....	21, 24, 28, 34, 35, 36, 47, 57, 61, 62, 64, 79, 115
Wiki.....	7, 22, 26, 44, 159
Working copy.....	16

Manuals

Manual del desarrollador del SER.RI-TIC

Aquest manual està format pels següents capítols d'aquesta documentació:

- Especificacions.
- Disseny. Només les seccions de les eines o tecnologies escollides. Les avaluacions de les eines no són necessàries tampoc.
- Desenvolupament, Tot menys la instal·lació de la base de dades d'Oracle, però conservant la descàrrega del driver JDBC d'Oracle.
- Conclusions, per tenir els mapes conceptuals.
- Bibliografia.
- Llicència.
- Índex alfabètic.

Manual del desarrollador extern al SER.RI-TIC

Aquest manual està format pels següents capítols d'aquesta documentació:

- Especificacions.
- Conclusions, només les eines escollides i el mapa conceptual d'aquestes.
- Llicència.