



Personalització i desenvolupament sobre Liferay

Prototipus base d'aplicació web

Bartolomé Martínez Román
Grau en Enginyeria Informàtica

Albert Grau Perisé

Juny 2013



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	Personalització i desenvolupament sobre Liferay. Prototips d'aplicació web.
Nom de l'autor:	Bartolomé Martínez Román
Nom del consultor:	Albert Grau Perisé
Data de lliurament (mm/aaaa):	06/2013
Àrea del Treball Final:	Java EE
Titulació:	Grau en Enginyeria Informàtica
Resum del Treball (màxim 250 paraules):	
<p>Aquest Treball Final de Grau es centra en l'estudi del programari de portal <i>Liferay</i>, i en la creació d'un prototipus sobre aquest programari 100% Java.</p> <p>Liferay ofereix infinitat de funcionalitats i opcions: gestió de continguts, gestió d'usuaris, portlets, temes, etcètera. A partir d'un llistat d'objectius, el treball ha aprofundit en diferents tecnologies, generant diversos mòduls:</p> <p>S'ha treballat la instal·lació i configuració d'un entorn de desenvolupament integral, format per un servidor d'aplicacions i per un IDE de desenvolupament amb totes les utilitats i llibreries pertinents.</p> <p>S'ha generat un tema base, a partir de l'entorn anterior, per a poder aprendre i documentar els aspectes visuals del portal.</p> <p>S'han implementat també una sèrie de plantilles per al portal, que s'integren amb el tema i donen estructura a les pàgines.</p> <p>També s'han desenvolupat una seguit de portlets, utilitzant Plugins SDK i el Service Builder de Liferay, aquesta última una eina que treballa amb Hibernate i Spring per a facilitar el desenvolupament sobre Liferay.</p> <p>Tot aquest treball de desenvolupament s'ha documentat, amb la intenció de poder generar nous projectes a partir del coneixement generat, plasmat als Annexos.</p>	

Abstract (in English, 250 words or less):

This final project focuses on the study of the Liferay portal software and on the creation of a prototype of this software 100% Java.

Liferay offers many features and options: content management, user management, portlets, themes, etc. Starting from a predefined list of objectives to study and document, the project has generated several products with different technologies:

First of all, a complete development environment has been installed and configured. It includes an application server and a development IDE with all the necessary libraries and tools.

Moreover, a base theme has been developed too, and all the process has been documented. Two templates have been also done from scratch, to show correctly the portlets in the theme.

A set of portlets has been developed using the Plugins SDK and Liferay Service Builder. Service Builder is a Liferay tool that works with Hibernate and Spring to make easier the development of some functionalities.

All this work has been documented and brought together in different annexes at the end of this document.

Paraules clau (entre 4 i 8):

Liferay, Portlets, Java, JSR 286, Portal

Índex

1. Introducció.....	8
1.1 Context i justificació del Treball	8
1.2 Objectius del Treball.....	9
1.3 Enfocament i mètode seguit.....	10
1.4 Planificació del projecte.....	11
1.5 Breu resumari de productes obtinguts.....	13
1.6 Breu descripció dels altres capítols de la memòria	13
2. Anàlisi del Prototipus.....	15
2.1 Rols	15
2.2 Casos d'ús.....	15
2.3 Apartats / Pàgines	17
2.4 Pantalles.....	18
2.5 Diagrama de classes.....	27
2.6 Diagrames d'estats.....	27
2.7 Components utilitzats	28
3. Arquitectura.....	30
3.1 Liferay, visió global.....	30
3.2 Entorn de desenvolupament.....	31
3.3 Arquitectura dels components.....	32
3.4 Tecnologies i programari utilitzat.....	33
4. Muntatge de l'organització des de zero.....	35
4.1 Planejament inicial	35
4.2 Creació de la organització i del lloc web.....	35
4.3 Creació de les pàgines.....	36
4.4 Continguts. Estructures i plantilles web.....	37
4.5 Muntatge dels portlets	38
4.6 Usuaris i rols.....	38
4.7 Altres	38
5. Conclusions.....	40
6. Glossari	41
7. Bibliografia.....	43
Annex A. Liferay: conceptes bàsics.....	44
A.1 Portal.....	44
A.2 Administració (Tauler de Control)	44
A.3 Membres: Usuaris, Rols i Organitzacions	45
A.4 Gestor de continguts (CMS).....	48
A.5 Continguts Web, Estructures web i Plantilles web.	49
A.6 Importació i exportació de continguts.....	51
A.7 Mètodes de desenvolupament: Plugins SDK, Liferay IDE i Ext	51
A.8 Service Builder.....	53
Annex B. Instal·lació i configuració d'un entorn de desenvolupament.....	54
B.1 Repositoris	54
B.2 Instal·lació JDK	54
B.3 MySQL Server	54
B.4 Driver MySQL	55
B.5 Client MySQL.....	55

B.6 Liferay Portal.....	55
B.7 Plugins SDK.....	56
B.8 Eclipse	57
B.9 Configuració Tomcat-Liferay	57
B.10 Configuració Plugins SDK.....	57
B.11 Provar l'entorn.....	58
Annex C. Desenvolupament d'un tema a mida	60
C.1 Creació del tema base.....	60
C.2 Generació d'un projecte Eclipse associat	60
C.3 Directori <i>_diffs</i>	62
C.4 Estudi de la maqueta HTML	63
C.5 Creació de les estructures base. Carpeta <i>templates</i>	63
C.6 Altres : Codi estàtic	64
C.7 Desplegament del tema	64
Annex D. Desenvolupament d'una plantilla a mida	66
D.1 Passos per a generar una plantilla	66
D.2 El fitxer <i>.tpl</i>	67
Annex E. Especificació de portlets	69
E.1 Que és un portlet?	69
E.2 Fases	69
E.3 Cicle de vida d'un portlet.....	71
E.4 Modes d'un portlet	71
Annex F. Desenvolupament de portlets	73
F.1 Creació del projecte e integració a Eclipse	73
F.2 Els fitxers <i>portlet.xml</i> i <i>liferay-portlet.xml</i>	75
F.3 Implementació. Mètodes a implementar. <i>javax.portlet.GenericPortlet</i>	75
F.4 Portlets MVC	76
F.5 Capa de Servei	76
F.6 Portlets de tercers	78

1. Introducció

1.1 Context i justificació del Treball

Dins del ampli ventall d'eines que es poden utilitzar per a fer realitat un projecte, existeix tota una família de tecnologies construïdes al voltant del llenguatge de programació Java. Aquesta família de tecnologies és la *Plataforma Java*.

Dins de la Plataforma Java es pot trobar la coneguda com a *Plataforma Java EE*, orientada a desenvolupar aplicacions empresarials distribuïdes. Existeixen infinitat de bastiments de tercers, llibreries e implementacions d'estàndards dissenyats per a la plataforma Java EE, que ha anat evolucionant al llarg dels anys.

Els diferents estàndards que la Plataforma Java implementa són a priori especificats per la *Java Community Process* –coneguda per les seves sigles, JCP- en els seus documents *Java Specification Request*. Dins dels diferents estàndards que aquest organisme defineix es pot trobar el *JSR 168*, que especifica la versió 1 de portlets, i el *JSR 286*, que especifica la versió 2 de portlets. Aquests components són molt importants dins d'aquest Treball Final de Grau.

Un portlet és un component modular, una mini aplicació web, que s'executa sobre una plataforma –anomenada generalment portal- de forma integrada. Això és així gràcies a que el portal té un contenidor de portlets, una eina que gestiona i visualitza cada portlet tal i com marca l'especificació. Però el desenvolupament d'aplicacions sobre Java no és -tot i la existència d'estàndards ben documentats com s'ha vist- una tasca fàcil. Tota la plataforma és una eina molt potent i complexa, i es requereix un esforç important per a poder desenvolupar projectes sobre ella de forma competent.

L'objectiu d'aquest Projecte de Final de Grau és generar un model d'aplicació –un prototipus- que serveixi com a base per a desenvolupaments d'aplicacions webs sobre eines de portal. No tots els projectes basats en Java es realitzen sobre aquest tipus de plataformes, però si que representen una part dels projectes que es demanden actualment, i freqüentment tenen una càrrega de complexitat important.

Gràcies al programari sobre el que s'ha muntat el prototipus (Liferay, un programari líder en portals Java), es compta amb tota una base sobre la que treballar, i sobre la que es poden desenvolupar noves funcionalitats –utilitzant portlets-, o modificar els ja existents.

Per què un prototipus sobre un programari de portal?

Aquest mateix projecte es podria haver plantejat sobre la plataforma Java EE utilitzant un bastiment (*Spring*, *Struts*, etcètera), sense utilitzar cap programari de portal.

Però el programari de portal –en aquest cas Liferay–, tot i afegir una certa complexitat al model, aporta tot un seguit de funcionalitats pròpies que seran molt útils. A més a més, Liferay permetrà reutilitzar fàcilment qualsevol component que es desenvolupi.

En definitiva, el que s'ha buscat ha estat generar coneixement i una base per a futurs projectes sobre un programari de portal, i no desenvolupar un producte amb una finalitat en si mateix.

1.2 Objectius del Treball

Tal i com es comenta en el punt anterior, l'objectiu d'aquest Treball Final de Grau es generar un prototipus sobre Liferay que faciliti el futur desenvolupament d'aplicacions web sobre aquesta plataforma.

Liferay és actualment un estàndard *de facto* quan es parla de programaris de portal dins de Java. És per això que el treball es planteja directament sobre aquest programari, sense tenir en compte altres possibles candidats.

Les diferents tasques que s'han desenvolupat dins d'aquest Treball Final de Grau es poden classificar en quatre grups diferents. Cadascun dels grups s'encarrega d'una sèrie de tasques dins del TFG.

La realització de les diferents tasques de cadascun dels grups dona com a resultat un prototipus complet i útil com a base per a futurs projectes.

Instal·lació, configuració i anàlisi de funcionalitats del portal

Primerament, ha estat necessari instal·lar i configurar un servidor de desenvolupament correctament. Aquesta tasca inclou la instal·lació d'un servidor i –el que es més important– la instal·lació i configuració d'un Liferay correctament. Més endavant es pot trobar més informació sobre el programari base necessari.

Una vegada instal·lada l'eina de portal, s'han d'estudiar els diferents components que té Liferay, com la gestió d'usuaris, la gestió de continguts o la biblioteca de documents i d'imatges.

Tema base

En segon lloc s'ha de generar un tema per al prototip. Un tema encapsula l'aspecte visual d'una aplicació web Liferay en un únic component.

Així doncs, Liferay compta amb tot un seguit d'eines que permeten muntar diferents tipus de components –no només portlets- per al portal.

En aquest segon apartat s'ha treballat la creació d'un tema bàsic, a partir del qual es poden realitzar variacions i/o prendre'l com a model per a nous temes.

Per l'apartat visual, a més a més del tema s'han creat dues plantilles. Aquests components donen estructura al contingut i es combinen amb el tema per a generar l'aspecte visual final de cada pàgina. Més endavant veurem doncs com tema + plantilles treballen juntes per a donar un aspecte determinat.

Generació de noves funcionalitats per al portal

Tal i com s'ha comentat anteriorment, Liferay aporta tot un seguit de funcionalitats que ens ajuden a resoldre moltes de les demandes d'un projecte. Tot i això, hi haurà certs casos en els que s'ha de desenvolupar o modificar un component en concret.

El prototip pot generar noves funcionalitats per al portal, mitjançant portlets. Els portlets que s'han generat a dia d'avui serveixen per a la gestió integral de productes, i per a la gestió de contactes per a una web d'una botiga, en aquest cas, de mobiliari.

En pròxims apartats es veurà més en detall cadascun dels portlets desenvolupats.

Documentació

Per a poder afrontar correctament futurs projectes sobre aquesta plataforma s'han documentat els diferents processos / objectius en aquest mateix document. Aquesta documentació són un seguit d'annexos sobre diferents temes:

- Annex A: Liferay, conceptes bàsics
- Annex B: Instal·lació i configuració d'un entorn de desenvolupament
- Annex C: Desenvolupament d'un tema a mida
- Annex D: Desenvolupament d'una plantilla a mida
- Annex E: Especificació de portlets
- Annex F: Desenvolupament de portlets

1.3 Enfocament i mètode seguit

El Treball Final de Grau plantejat és bàsicament un treball d'investigació i desenvolupament. L'anàlisi del prototipus, que es veurà més endavant, és l'anàlisi d'una aplicació destinada a testejar diferents parts del portal.

Així doncs, a partir d'un anàlisi inicial s'ha realitzat un treball de formació pròpia i d'investigació en diverses tecnologies que s'utilitzen a l'univers Java. A partir

dels coneixements obtinguts s'han realitzat el prototipus, seguint l'anàlisi abans mencionat.

1.4 Planificació del projecte

Tal i com s'ha pogut veure al punt 1.2, existeixen diferents conjunts d'objectius, que s'han d'anar resolent de forma seqüencial, repartits en tres fases o entregues .

També va haver un treball previ, en el qual es va discutir el projecte i la seva viabilitat.

Fase I: Anàlisi i disseny

La primera fase del projecte s'ha encarregat de l'anàlisi i del disseny del prototip.

Primerament es va realitzar un anàlisi previ amb tots els punts que es volien tractar en el projecte. A continuació, es va realitzar la instal·lació de l'entorn de treball inicial, i la configuració de la màquina virtual utilitzada (un Ubuntu 12.04) i de totes les eines associades (Java, Eclipse, Ant, MySQL, etcètera) i del portal.

També es van realitzar les maquetes HTML del prototipus, seguint les indicacions de l'anàlisi.

Per últim, es va plasmar tot aquest treball en un document (PAC2).

Durant tot el procés es va aprofitar per anar indagant sobre com desenvolupar portlets, temes i altres elements utilitzant Liferay i l'eina associada Plugins SDK. També sobre programació mitjançant Hibernate i Spring, tecnologies utilitzades per Liferay.

Fase II: Desenvolupament

La segona fase del projecte, que s'ha encarregat del desenvolupament dels diferents mòduls, es pot desglossar en els següents elements:

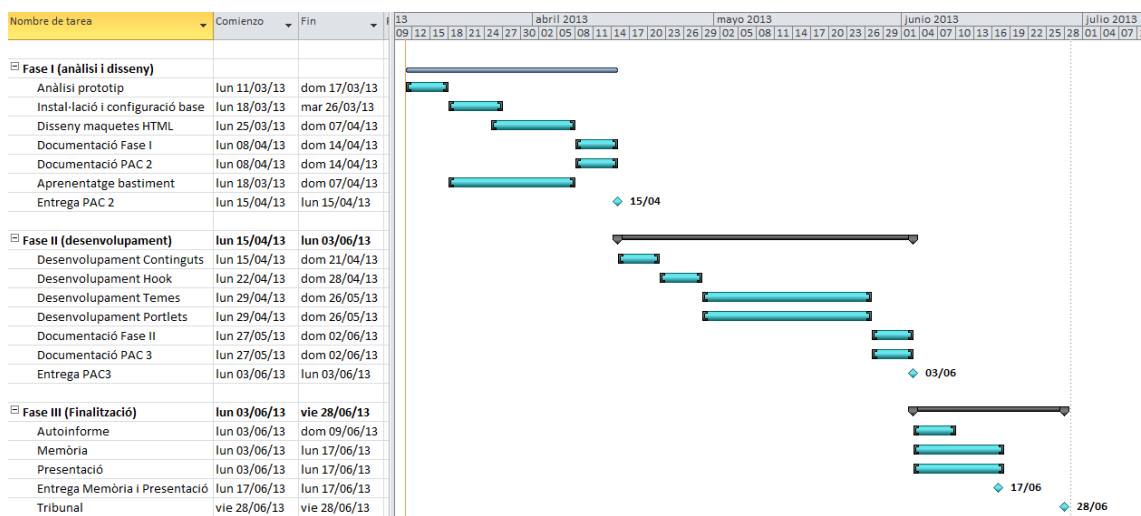
- *Generació d'un tema a partir de la maqueta:* Utilitzant l'eina de Plugins SDK s'ha generat un tema base per al prototipus basat en les maquetes HTML de la primera fase. En un principi, la idea era desenvolupar també un segon tema per a dispositius mòbils, però no ha estat possible per manca de temps.
- *Generació de dues plantilles (layouts) a partir de la maqueta:* Al igual que s'ha generat un tema, ha estat necessari generat dues plantilles per a poder estructurar on es volen posar els diferents portlets a cada pàgina.

- *Creació de portlets*: S'han generat tot un seguit de portlets per a la gestió de productes i de contactes. En el següent punt (1.5) es detallen tots els portlets generats.
- Documentació: També s'ha documentat tot el procés de desenvolupament, en diferents apartats d'aquest document.

Fase III: Finalització

La fase III inclou tot el procés de documentació final: creació de la memòria i de la presentació final, a més a més d'un document d'autoavaluació.

A continuació es pot veure la planificació temporal del TFG. Tal i com es pot observar, existeixen tres fases clarament diferenciades:



Problemes trobats

Durant la realització del Prototipus no s'ha produït cap problema important que hagi obligat a modificar substancialment els objectius del projecte.

Tot i això, el rendiment de la màquina virtual una vegada es va instal·lar tot l'entorn i es va començar a desenvolupar era molt pobre.

Així doncs, es va optar - durant la Fase II – per tornar a instal·lar tot l'entorn de desenvolupament directament sobre Windows 7, a la màquina on s'executava la màquina virtual Ubuntu.

El rendiment en aquest segon cas ha estat molt millor, i ha permès que es compleixin les dates d'entrega, inassolibles amb la primera instal·lació sobre una màquina virtual amb Ubuntu.

1.5 Breu sumari de productes obtinguts

Aquest projecte genera tot un seguit de components que, en conjunt formen un prototipus sobre Liferay:

- Un servidor d'aplicacions GlassFish, versió 3.1.2, amb un Liferay 6.1.1 operatiu, apuntant a una base de dades MySQL.
- Un entorn de desenvolupament Plugins SDK, mitjançant el qual es poden generar diferents tipus de mòduls (temes, portlets, plantilles i altres) per a Liferay.
- Un tema base (Prototipus-Theme), que encapsula tot l'aspecte visual del prototipus.
- Una plantilla 1-70-30-1. Això vol dir que la informació es mostra en tres files:
 - una primera al 100% d'amplada
 - una segona amb dues columnes, una al 70% i una al 30%
 - una tercera al 100% d'amplada
- Una plantilla 70-30-1. Això vol dir que la informació es mostra en dues files:
 - una primera amb dues columnes, una al 70% i una al 30%
 - una segona al 100% d'amplada
- Un conjunt de portlets, que engloben tot un seguit de funcionalitats al voltant de productes i contactes:
 - Portlet de Gestió de Producte
 - Portlet de Gestió de Contacte
 - Portlet de Llistat/Cerca de Producte
 - Portlet de Detall de Producte
 - Portlet de Minidetall de Producte
 - Portlet de Contacte.
- Aquest document, on es detallen conceptes i processos bàsics per al desenvolupament sobre Liferay:
- Una màquina virtual, amb el prototipus totalment operatiu.

Altres productes

Tal i com es pot observar al llistat anterior, un dels productes és una màquina virtual amb un prototipus operatiu.

A diferència de l'entorn de la presentació que utilitza un servidor d'aplicacions GlassFish, aquesta màquina virtual utilitza un servidor Tomcat, degut al rendiment, que no feia possible treballar sobre GlassFish amb una màquina virtual sobre els equips disponibles per a desenvolupar aquest TFG.

1.6 Breu descripció dels altres capítols de la memòria

Tal i com ja s'ha comentat, el prototipus ha de servir com a base per a futurs desenvolupaments. El comptar amb un entorn operatiu de Liferay permet solucionar dubtes i provar de realitzar segons quines accions al portal.

És important però comptar també amb una documentació que serveixi de guia, com a material de consulta als possibles dubtes que es generin.

És per aquesta raó que als següents capítols es tracten diferents aspectes i conceptes del desenvolupament sobre Liferay.

El capítol 2, *Anàlisi Prototipus*, inclou un anàlisi del prototipus des d'un punt de vista funcional. També detalla els components que s'han creat per a fer del prototipus una realitat.

El capítol 3, *Arquitectura*, detalla l'arquitectura del portal, dona informació sobre components y detalla les tecnologies que s'han utilitzat.

En el capítol 4, *Muntatge de l'organització des de zero* es pot veure tot el procés de construcció d'una web sobre Liferay com la que s'ha fet pel prototipus.

Al capítol 5, *Conclusions*, es plasmen les conclusions finals de tot el treball realitzat aquestes darreres setmanes.

Al capítol 6, *Glossari*, es pot veure un llistat de diferents termes i conceptes utilitzats al treball.

Al capítol 7, *Bibliografia*, es pot trobar un llistat de fonts d'informació per a poder aprofundir i entendre millor aquest Treball Final de Grau.

L'annex A, *Liferay: conceptes bàsics*, parla del diferents conceptes que són importants per a poder treballar ordenadament amb el programari de portal. Hi ha molts conceptes a Liferay, i tot i que els seus noms són molt descriptius (*usuari, organització, rol*), és important tenir clar tot l'entorn abans de treballar sobre ell.

L'annex B, *Instal·lació i configuració d'un entorn de desenvolupament*, detalla tot el procés per a instal·lar un entorn de desenvolupament totalment operatiu.

L'annex C, *Desenvolupament d'un tema a mida*, serveix com a guia per a la creació d'un tema.

L'annex D, *Desenvolupament d'una plantilla a mida*, serveix com a guia per a la creació d'una plantilla.

L'annex E, *Especificació de portlets*, serveix com a punt d'entrada per a desenvolupar aquests components, explicant les seves característiques essencials.

Per últim l'annex F, *Desenvolupament de portlets*, serveix com a guia per a la creació d'un portlet sobre Plugins SDK.

2. Anàlisi del Prototipus

En aquest apartat es detallen diferents aspectes funcionals del prototipus. Primerament es detallen els rols i casos d'ús associats al prototipus, i es veuen les diferents pantalles associades.

També es detallen alguns punts importants del projecte mitjançant diagrames de classes i d'estat.

Finalment es poden veure els diferents components que s'han de desenvolupat.

El prototipus, tal i com s'ha comentat, ha de servir per a testejar a fons diferents tecnologies al voltant del programari de portal. Així, tot i que no és important la finalitat de la web, s'ha decidit prendre com a model una botiga de mobles, de forma que tots els apartats del prototipus (centres, productes, contacte, etcètera) utilitzin aquest tema com a *excusa* per a testejar diferents tecnologies.

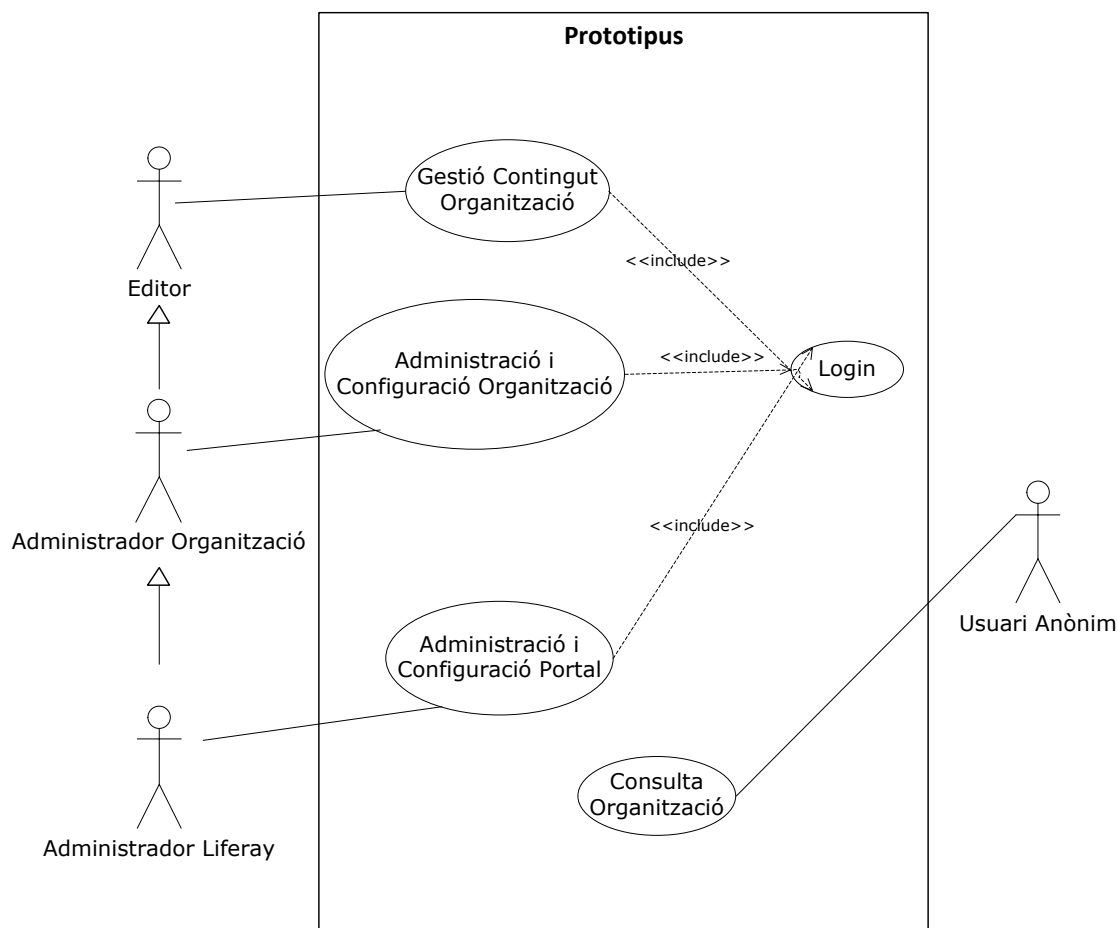
2.1 Rols

El portal compta amb quatre rols/actors que cobriran diferents perfils del prototipus:

- Administrador Liferay: Superusuari del portal, els usuaris amb aquest rol podran realitzar tasques de configuració i de gestió tant d'organitzacions com del seu contingut, pàgines, usuaris, etcètera.
- Administrador Web: Superusuari d'una organització, en el nostre cas del prototipus, pot gestionar de forma completa l'organització de la que és administrador i el seu contingut.
- Editor (o Gestor): Rol per a usuaris que s'encarreguen de l'edició d'una organització. No pot realitzar tasques de gestió de pàgines o de les metadades de l'organització, però sí que pot gestionar el seu contingut: textos, imatges, documents, etc. També pot configurar algunes de les funcions dels diferents portlets que es troben a les pàgines de la seva organització. També és anomenat en alguns punts *Gestor*.
- Usuari Anònim: Rol pels usuaris sense identificar, pot consultar dades als diferents apartats de l'organització i realitzar tasques d'enviament d'informació, a través dels formularis disponibles.

2.2 Casos d'ús

A continuació es pot veure el diagrama de casos d'ús per als diferents actors que s'han detallat a l'apartat anterior. Cada cas d'ús és molt general, englobant diferents tasques.



En les següents taules es pot trobar detallat cadascun dels casos d'ús del diagrama anterior:

Nom	Login (Identificació)
Descripció	Permet a l'usuari identificar-se al sistema. Una vegada identificat, obtindrà permisos sobre diferents accions en funció del seu perfil.

Nom	Gestió Contingut Organització
Descripció	Gestió completa dels diferents continguts de l'organització, des dels textos dels diferents apartats, imatges i documents associats fins a les dades de cada producte. Accés a les dades de contacte.

Nom	Administració i Configuració Organització
Descripció	Control total sobre l'organització, les seves pàgines i format. Possibilitat de crear noves pàgines, esborrar les ja creades, modificar meta informació de l'organització o canviar el tema, ja sigui d'una pàgina o de la totalitat d'elles.

Nom	Administració i Configuració Portal
Descripció	Control sobre el portal i totes les seves organitzacions. Possibilitat de modificar paràmetres bàsics, usuaris, carregar

	components (portlets, temes i altres).
--	--

Nom	Consulta Organització
Descripció	Possibilitat de consultar la informació pública de l'organització i de les seves pàgines. Possibilitat d'utilitzar els diferents formularis, i realitzar cerques sobre producte.

2.3 Apartats / Pàgines

El prototipus simula una web presencial d'una organització / empresa, i consta dels apartats (pàgines) que es poden veure en la següent taula:

PÀGINA	DESCRIPCIÓ
Inici	Pàgina d'inici de la web. Dona informació general sobre el web i accés als diferents apartats. També té un apartat amb productes destacats. I un rotador de promocions (rotador d'imatges amb un link associat).
Empresa	Apartat amb contingut de l'empresa (història, dades interessants, etcètera). Apartat de contingut. Té un apartat amb productes destacats.
Centres	Informació sobre els diferents centres de l'organització/empresa. Té un apartat amb productes destacats.
Catàleg de Productes / Cercador	Apartat principal de producte. Compta amb un cercador de producte, que es pot personalitzar, com es podrà veure més endavant.
Detall de Producte	Detall d'un producte en particular. Compta amb un formulari per a demanar més informació sobre el producte en concret.
Contacte	Formulari de contacte general.
Mapa Web	Pàgina amb els diferents apartats del web. Contingut.
Dades Legals	Dades legals de l'organització. Contingut.
Administració – Llistat Producte	Llistat dels diferents productes donats d'alta al sistema. Accés a modificació, alta, i eliminació de productes.
Administració – Alta Producte	Fitxa per a donar d'alta nous productes al sistema.
Administració – Modificació Producte	Fitxa de modificació de productes.
Administració – Llistat Contactes	Llistat dels diferents contactes donats d'alta a la organització actual. Accés a la informació detallada de cada

	contacte.
Administració – Detall Contacte	Fitxa de detall per a un contacte en particular.
Pàgina Login (Administració)	Pàgina per a poder accedir al sistema amb algun dels usuaris amb privilegis. Aquesta pàgina és pública però es troba oculta, no surt al llistat general.

Apart d'aquests apartats, Liferay compta amb un Tauler de Control mitjançant el qual es poden gestionar, segons els rols de que es disposi, diferents àrees: gestió d'usuaris, gestió d'organitzacions, permisos, continguts, etcètera.

2.4 Pantalles

A continuació es poden veure una sèrie de captures de cadascun dels apartats abans comentat, amb una petita descripció dels punts més importants de cada apartat. Per mantenir un format agradable, algunes de les imatges tenen una mida més reduïda que la resta.

Inici

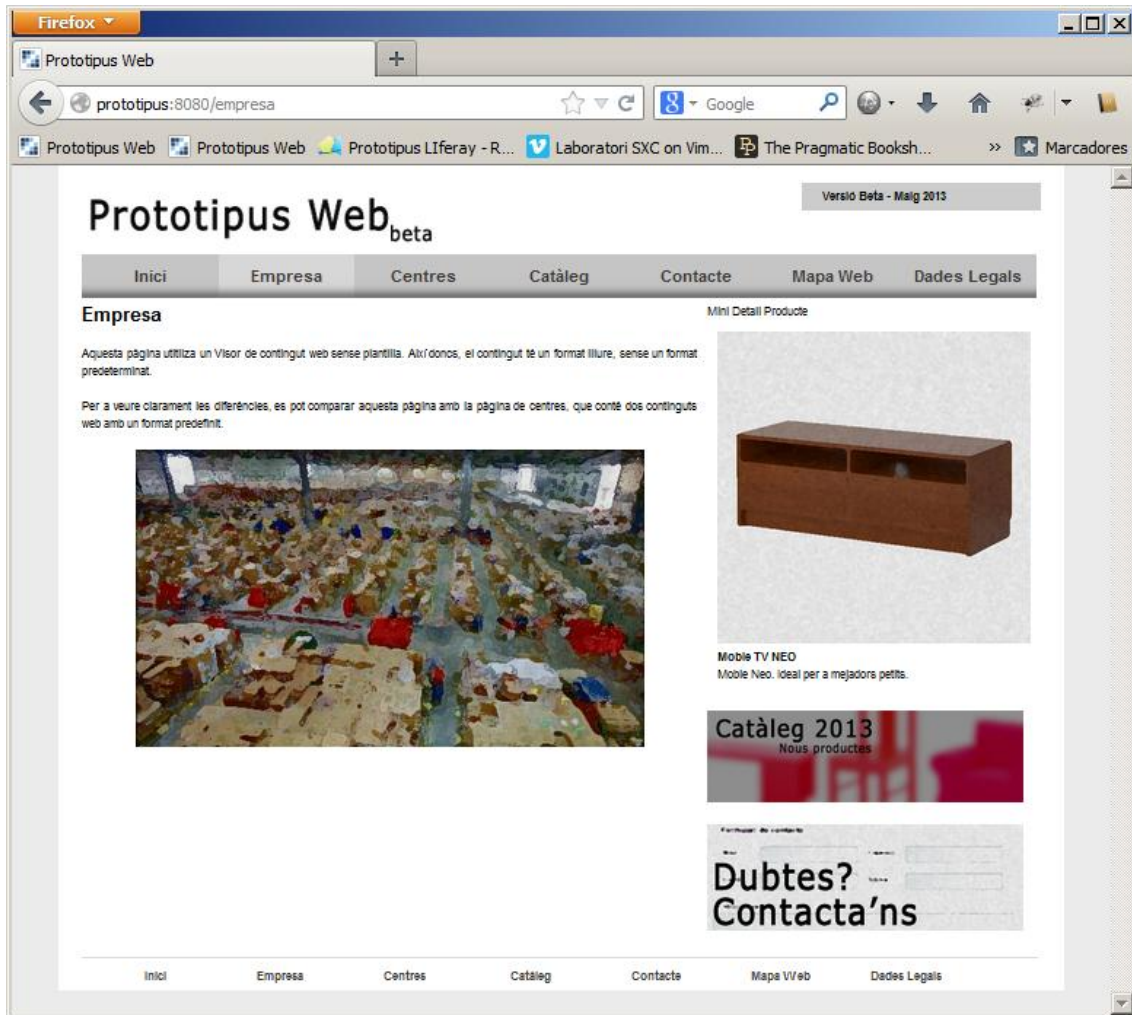


La pantalla de inici té cinc parts clarament diferenciades:

- Una capçalera, que s'anirà repetint a totes les pàgines públiques, amb el logo i una botonera amb accés a les diferents seccions del web.
- Un *rotador d'imatges*, que s'ha implementat utilitzant un portlet anomenat *Carousel*, del que es parlarà més endavant.

- Un cos, amb contingut genèric, és a dir, sense un format determinat.
- Un lateral, que conté, per una banda un contingut genèric igual que el cos i per l'altra un parell de portlets que ens donen un petit detall d'un producte en concret (portlet Minidetall de producte).
- Un peu, que al igual que la capçalera es va repetint a totes les pàgines públiques.

Empresa



L'apartat d'empresa és un apartat molt genèric. De fet, és molt semblant a la pàgina d'inici, sense rotador.

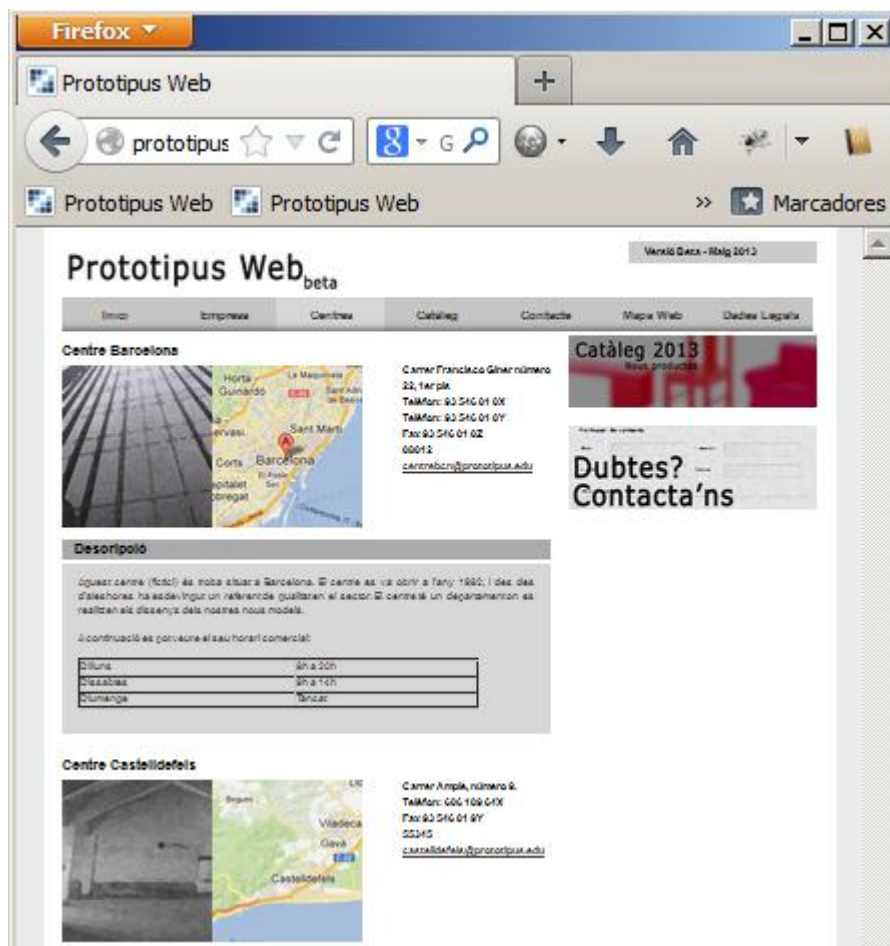
Mentre que a la pàgina d'inici s'utilitza la plantilla 1-70-30-1, en aquesta s'utilitza la plantilla 70-30-1.

Centres

La pàgina de centres utilitza, al igual que les dues vistes fins ara, un portlet anomenat Visor de Contingut Web (*Web Content Display* en anglès) per a

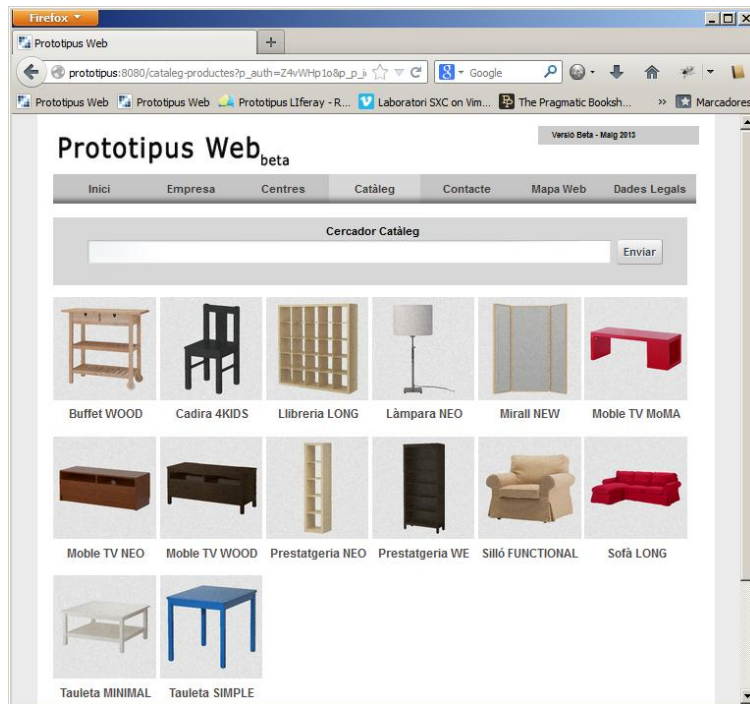
mostrar els diferents mòduls de contingut que conté la pàgina (al centre i al lateral).

Alguns d'aquests continguts (els centres per a ser més exactes) no tenen un format estàndard, i utilitzen una estructura i una plantilla associada per a obtenir un format concret a partir de les dades del contingut. Això es veurà més endavant a l'annex A, al punt A.5 *Continguts Web, Estructures Web i Plantilles web*.



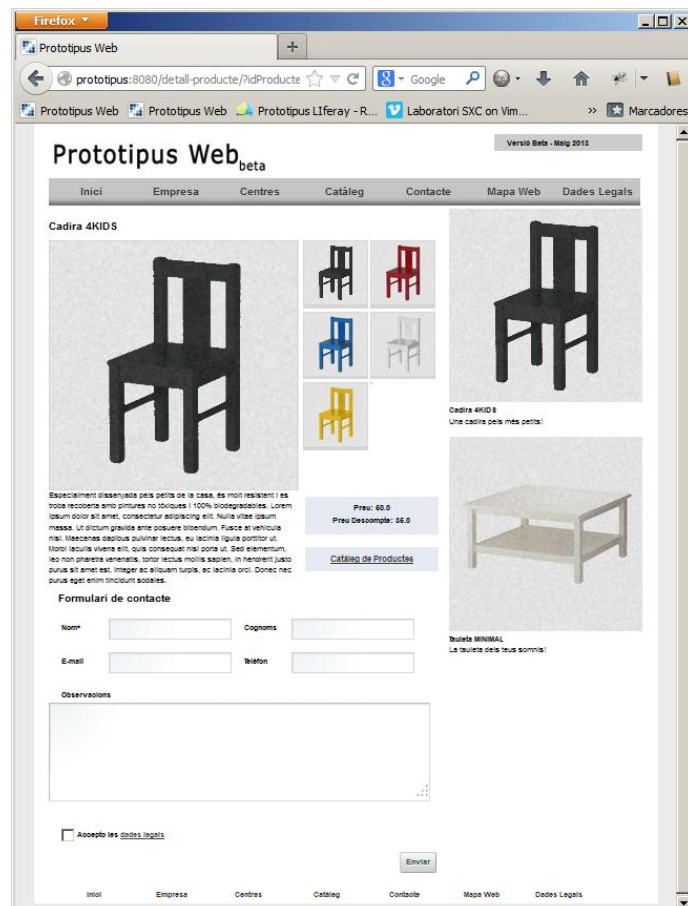
Catàleg de Productes / Cercador

El catàleg de producte compta a la part central amb un portlet que permet cercar per continguts sobre tots els productes de la web. El portlet també es pot configurar per a mostrar un llistat amb tots els productes enlloc del cercador.



Detall de Producte

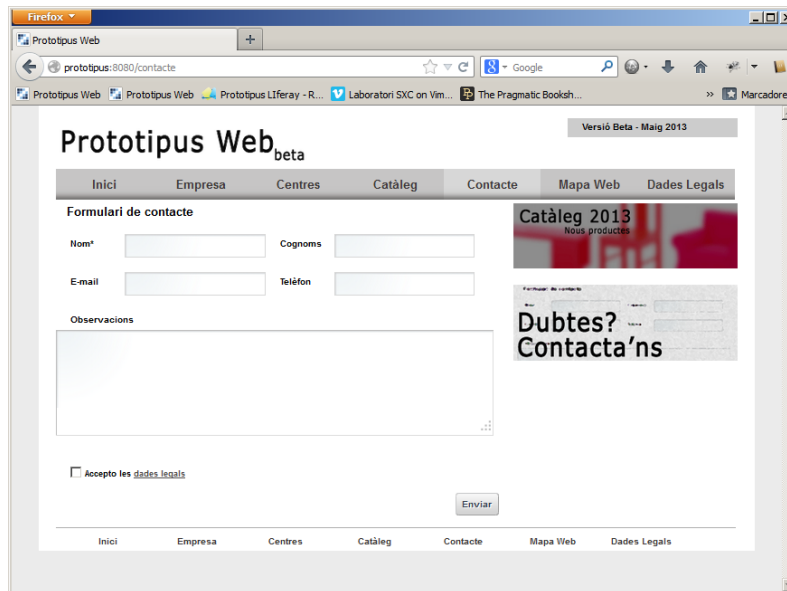
La fitxa de detall de producte compta en la seva part central amb un cercador amb les dades del producte, seguit d'un formulari mitjançant el qual un usuari pot donar d'alta una petició de contacte.



Tots els formularis del prototipus implementen validació de dades.

Al lateral (al igual que a la resta de pàgines públiques) es poden afegir portlets de minidetall de producte, per a promocionar un producte en concret.

Contacte

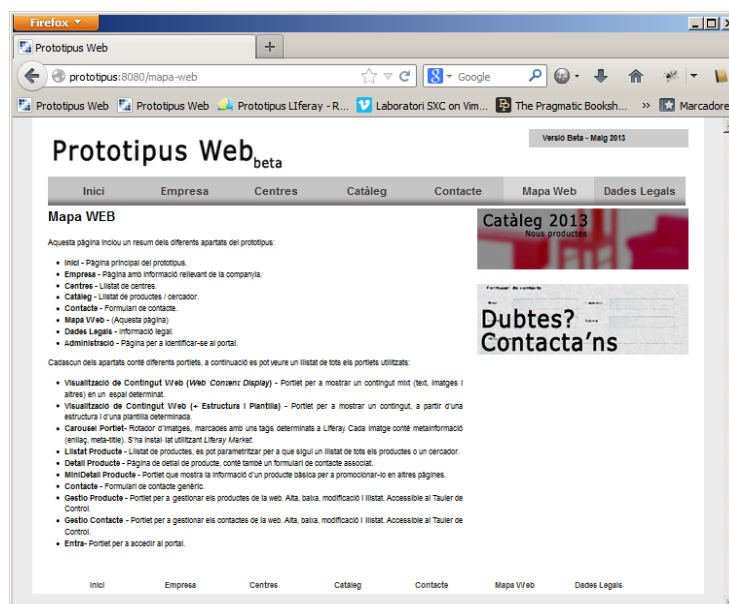


El formulari de contacte dona d'alta un contacte, només que en aquest cas de tipus genèric, sense informació associada.

Així doncs, més endavant, a l'apartat de llistat de contacte, es podrà veure com hi ha entrades de contactes vinculades a un producte i també genèriques.

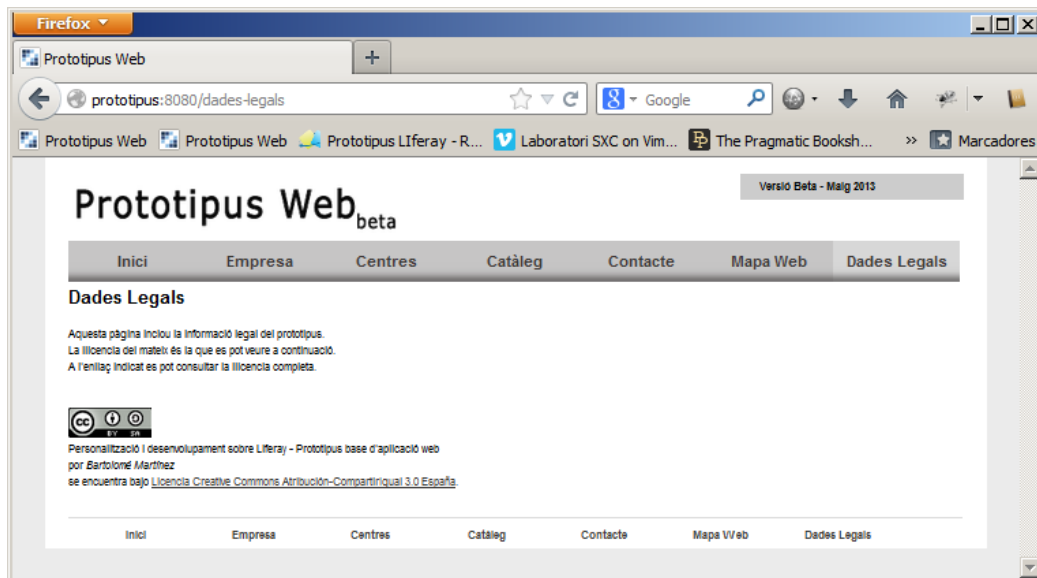
Mapa Web

Pàgina amb contingut genèric, no aporta noves funcionalitats al prototipus.



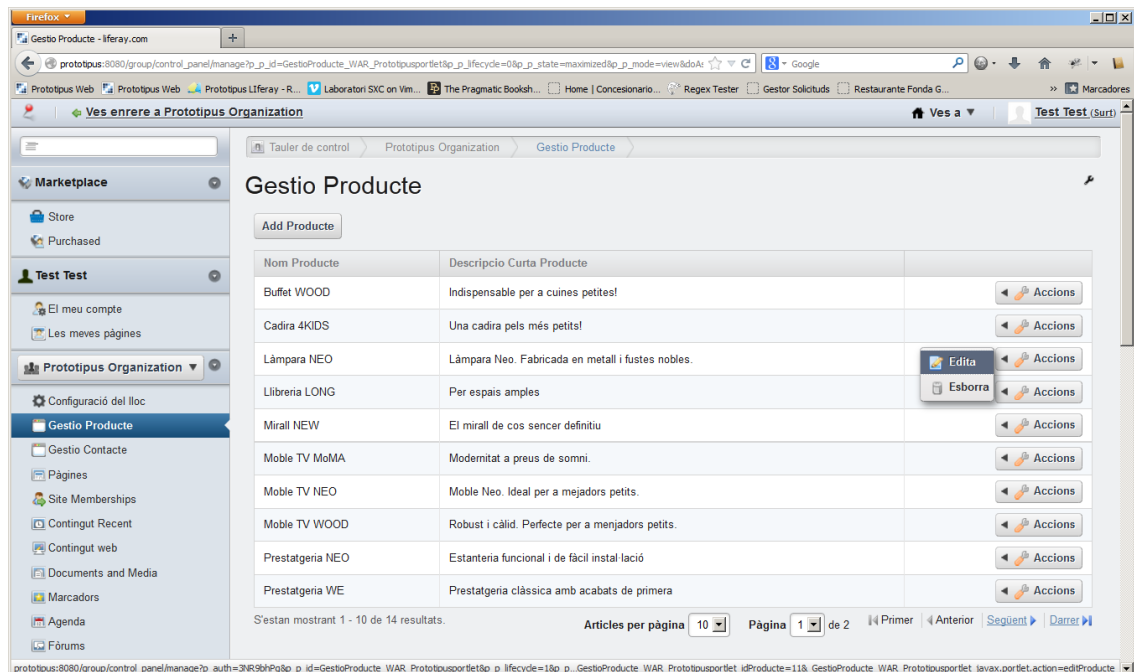
Dades Legals

Pàgina amb contingut genèric, no aporta noves funcionalitats al prototipus.



Administració – Llistat Producte

Una vegada identificats al sistema, es pot accedir a l'apartat de llistat de producte. En aquest apartat es accedeix a l'alta de producte, a la modificació i també es pot eliminar un producte concret.



Administració - Alta Producte & Modificació Producte

Els apartats d'alta i de modificació son molt semblants (de fet, a nivell de codi comparteixen inclús fitxer *jsp*).

Des d'aquesta pantalla es pot donar d'alta (o modificar) un producte, amb totes les seves dades adjuntes: nom, descripcions, preus, imatges...

Aquest és un dels apartats on per a futures versions del prototip es podrien seguir ampliant funcionalitats. Per exemple, la càrrega d'imatges es realitza de forma síncrona amb l'alta modificació, i es troba limitada a sis imatges. Es podria realitzar una càrrega mitjançant crides AJAX de les imatges, de forma que podríem arribar a tenir un llistat dinàmic d'imatges.

També es podrien realitzar altres treballs sobre les imatges, com per exemple redimensionament a les diferents mides d'imatges per cadascun dels apartats o utilització d'un servidor extern per a les imatges (ara es guarden al mateix servidor i es serveixen a través d'un domini diferent).

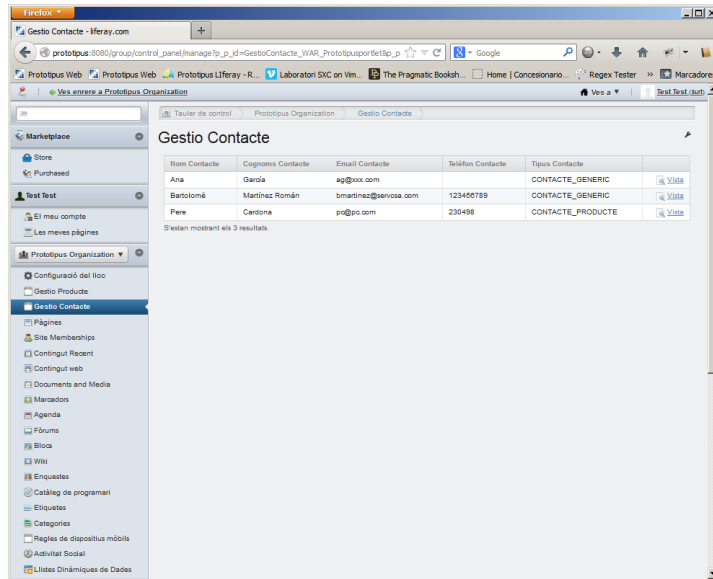
Aquests formularis compten amb una validació per assegurar que les dades seran vàlides per al sistema.

The screenshot shows a web browser window with the URL 'gestio-producte-liferay.com'. The page title is 'Gestio Producte'. The form contains the following fields and elements:

- Detalls Producte**
 - Nom*: Llàmpara NEO
 - Preu*: 300.0
 - Preu Descompte*: 200.0
 - Descripcio Curta*: Llàmpara Neo. Fabricada en metall i fustes nobles.
 - Descripcio Llarga: Llàmpara Neo. Fabricada en metall i fustes nobles.
- Image Uploads**
 - Imatge 1: [input checked="checked" type="checkbox"/>] Esborrar [Examinar...]
 - Imatge 2: [input checked="checked" type="checkbox"/>] Esborrar [Examinar...]
 - Imatge 3: [input type="checkbox"/>] Esborrar [Examinar...]
 - Imatge 4: [input type="checkbox"/>] Esborrar [Examinar...]
 - Imatge 5: [input type="checkbox"/>] Esborrar [Examinar...]
 - Imatge 6: [input type="checkbox"/>] Esborrar [Examinar...]
- Buttons: Submit, Cancel

Administració - Llistat Contactes

La pàgina de llistat de contactes s'assembla molt a la pàgina de llista de productes, però amb menys opcions. Així, aquesta pàgina permet visualitzar un contacte, però no gestionar-lo (CRUD).

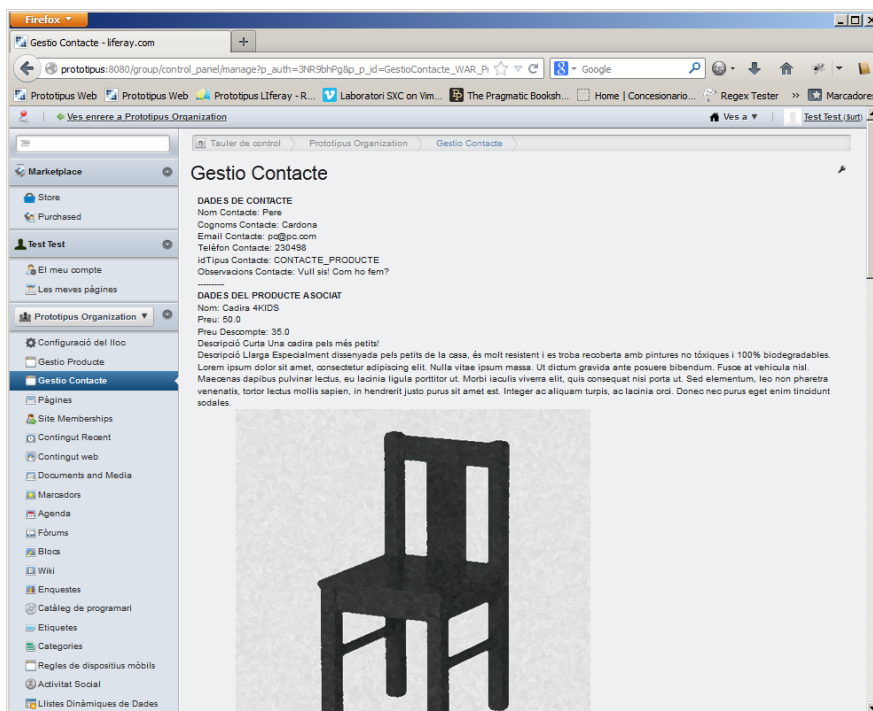


Administració – Detall Contacte

La pàgina de detall de contacte mostra tota la informació disponible d'un contacte en particular. Això implica, per una banda, la informació del contacte pròpiament dit, i per l'altra, en cas de que el contacte tingui un producte associat, la de producte.

A la següent imatge es pot veure un detall de contacte amb la informació de contacte (nom i cognoms, correu electrònic, etcètera) i la del seu producte associat.

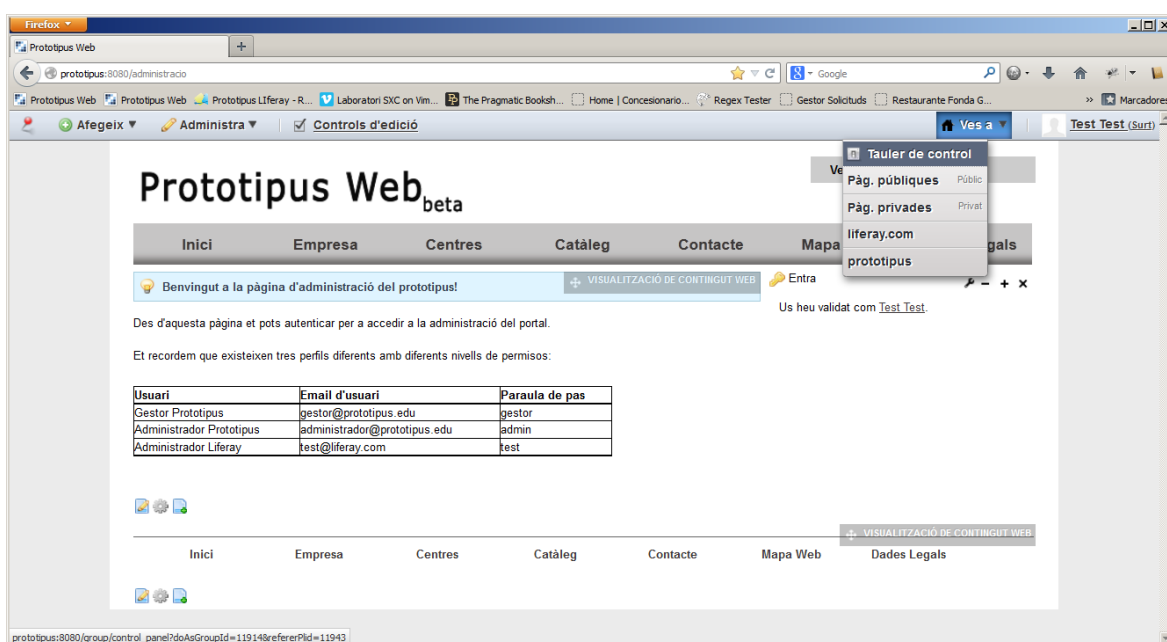
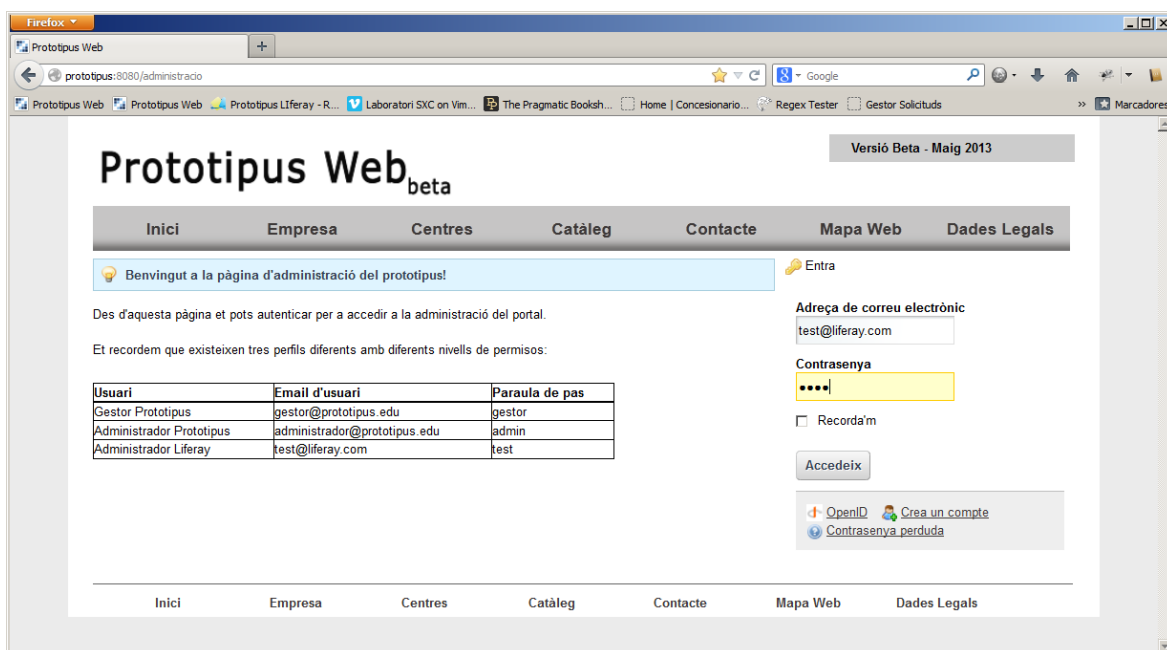
Les imatges del producte associat, en aquesta pàgina, no s'han escalat.



Pàgina Login – Administració

La pàgina d'identificació és pública, però s'ha optat per marcar-la com a privada, de forma que no surt a la botonera d'apartats de la capçalera. Aquesta botonera és dinàmica, de forma que les pàgines que es mostren són les que es troben marcades com a públiques i visibles.

Aquesta pàgina compta amb un portlet anomenat *Entra*, proporcionat d'origen per Liferay, mitjançant el qual es poden identificar els usuaris. A continuació es poden veure dues imatges, una abans i un altre després d'identificar-se al sistema amb un usuari. Es pot apreciar com, quan un usuari es troba identificat, apareix un menú a la part superior del portal.

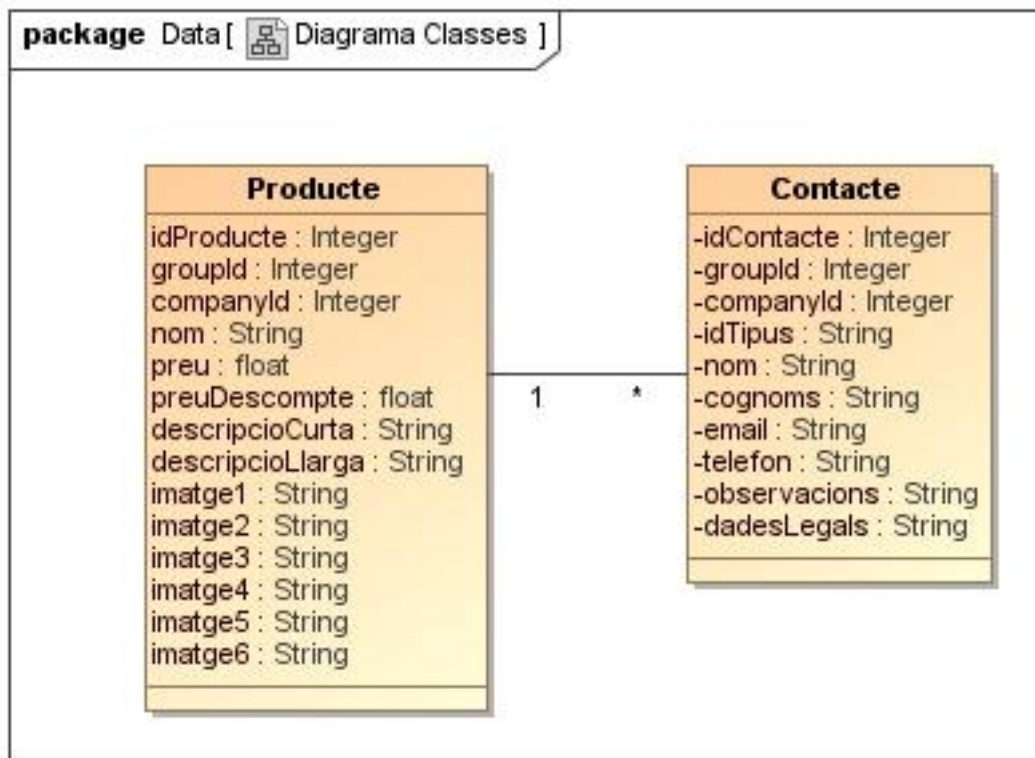


2.5 Diagrama de classes

En aquest apartat es pot veure un diagrama amb la representació de les classes per al model de producte i de contacte del prototipus.

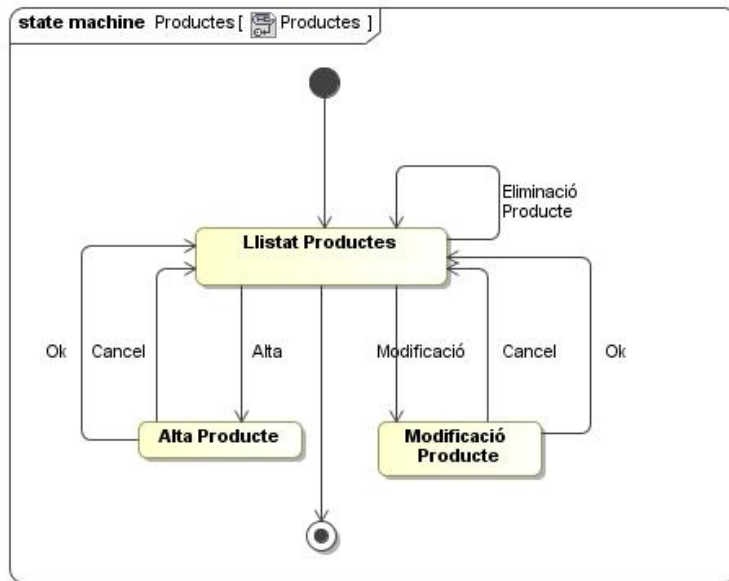
Cada producte es troba associat a un grup. Cada organització web sota Liferay mostra els productes assignats al seu grup en particular, de forma que es poden gestionar productes per a diferents webs de forma senzilla.

Amb els contactes passa el mateix. Gràcies a l'atribut *groupid* és possible seleccionar els contactes només d'una organització (web) en concret.

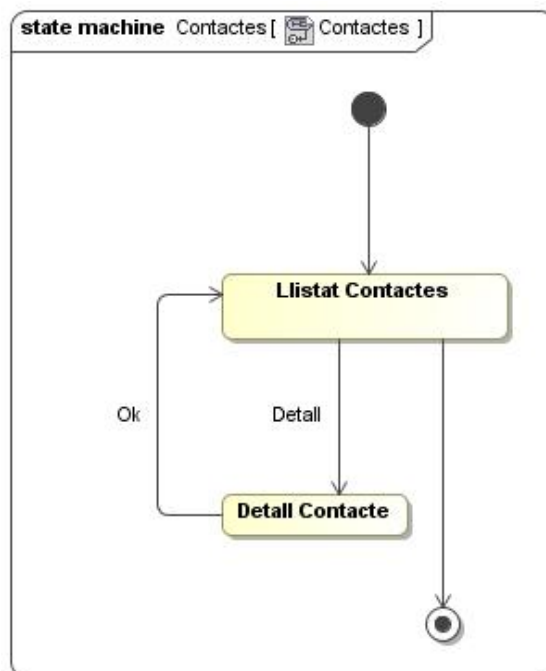


2.6 Diagrames d'estats

El prototip que s'ha desenvolupant no compta amb cap procés molt complex que impliqui la generació de tot un conjunt de diagrames per a poder explicar correctament les seves funcionalitats i accions. Tot i això, és necessari tenir clar com es comporta la administració de gestió de producte. En el següent diagrama d'estats es pot veure amb més claredat aquest punt:



El diagrama d'estats de contacte és més senzill, degut a que no té accions d'alta, modificació ni eliminació de contactes:



2.7 Components utilitzats

Durant el desenvolupament del prototipus s'han hagut d'utilitzar alguns components de Liferay, i també s'han hagut de desenvolupar alguns de nous des de zero. El següent llistat enumera tots els components que s'han utilitzat, de quin tipus són i quina és la seva funció dins del prototipus.

En l'apartat d'arquitectura es parlarà sobre cadascun dels tipus de components, per a poder comprendre la naturalesa de cadascun d'ells.

NOM COMPONENT	TIPUS	FUNCIÓ	NOU?
Tema Web	Tema	Component que genera la presentació visual del prototip per a navegadors web. S'ha creat nou.	Si
Gestió Producte	Portlet	Portlet per a la gestió CRUD de producte. Accessible des de l'administració del portal. S'ha creat nou.	Si
Llistat Producte / Cercador	Portlet	Portlet per al llistat dels diferents productes a la part pública del prototipus. Configurable, pot realitzar cerques. S'ha creat nou.	Si
Detall Producte	Portlet	Detall d'un producte en concret. Conté el seu propi formulari de contacte. S'ha creat nou.	Si
Mini Detall Producte	Portlet	Petit Detall d'un producte en concret. Serveix per a promocionar el producte (en un lateral per exemple) i accedir al seu detall general. S'ha creat nou.	Si
Formulari de Contacte	Portlet	Formulari de contacte genèric. S'ha creat de zero.	Si
Portlet Entra	Portlet	Portlet per a registrar-se a l'aplicació. Ja ve instal·lat amb Liferay 6.1	No
Carousel	Portlet	Portlet per a mostrar un rotador de promocions. Es troba al <i>Market</i> de Liferay.	No
Visor de Contingut Web	Portlet	Sense cap dubte, un dels portlets més utilitzats en la majoria d'entorns Liferay. Mostra un contingut donat d'alta al portal.	No
Llistat Contacte	Portlet	Portlet per al llistat dels contactes en general, permet també accedir al detall ampliat de cada contacte. Accessible des de l'administració del portal. Nou.	Si
Plantilla Home (1-70-30-1)	Plantilla	Plantilla per a la home, amb un apartat per al rotador de promocions i un altre per a productes destacats.	Si
Plantilla Interior (70-30-1)	Plantilla	Plantilla per a les pàgines interiors, amb un lateral per a productes destacats.	Si
Altres Plantilles	Plantilla	Existeixen altres conjunts de plantilles (70-30), (1) pròpies de Liferay. S'utilitzen sense modificacions.	No
Estructura Web Centre	Estructura Contingut Web	Estructura per als continguts webs de centre. S'ha de crear a partir de l'HTML de la pàgina de centre.	Si
Plantilla Web Centre	Plantilla Contingut Web	Plantilla web per als continguts webs de centre. S'ha de crear a partir de l'HTML de la pàgina de centre.	Si

3. Arquitectura

En aquest apartat es parlarà, primer de tot, de Liferay a nivell global. Es donarà una visió general del portal, i dels seus components.

En segon lloc, es mostrarà com s'ha configurat l'entorn de desenvolupament, per a que es tingui una visió clara d'aquest entorn.

En tercer lloc, i entrant una mica més a fons, es detallaran els diferents components sobre els que s'ha treballat en el prototipus.

Finalment, al subapartat *Tecnologies utilitzades*, es llisten de les diverses tecnologies que s'utilitzen al projecte.

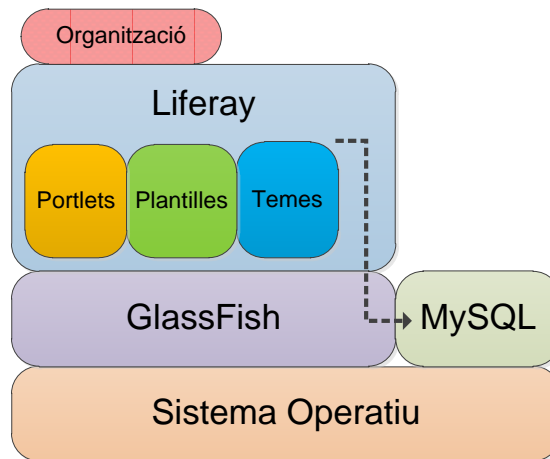
3.1 Liferay, visió global.

Per a qualsevol neòfit que vulgui entendre el funcionament d'un entorn Java existeixen tot un conjunt de conceptes i tecnologies que poden arribar a ser difícils d'entendre en un primer moment. *Servidor d'aplicacions, màquina virtual, datasources, instàncies, servidors virtuals, portal, organitzacions, Velocity*, etcètera.

A més a més, Liferay es pot desplegar (o instal·lar, com es prefereixi) sobre múltiples combinacions possibles de programari: sobre una base de dades Oracle mes un servidor d'aplicacions JBoss, sobre una base de dades MySQL mes un servidor Tomcat, etcètera.

Per si no fos suficient, es poden realitzar instal·lacions de Liferay relativament senzilles, com la que s'ha realitzat per aquest prototipus, però també molt complexes: es poden arribar a muntar entorns de portal en clúster amb múltiples nodes, amb integració de sistemes de tercers com Alfresco, amb SSO o inici únic de sessió (en anglès *Single Sign-On*), etcètera.

Liferay és un programari que necessita executar-se sobre un servidor d'aplicacions. En realitat, amb un contenidor de Servlets i JSP's en té prou, per la qual cosa programaris com Apache Tomcat també serveixen. En el cas del prototipus s'ha utilitzat GlassFish, que és un servidor d'aplicacions de codi lliure desenvolupat per Sun Microsystems (ara Oracle).



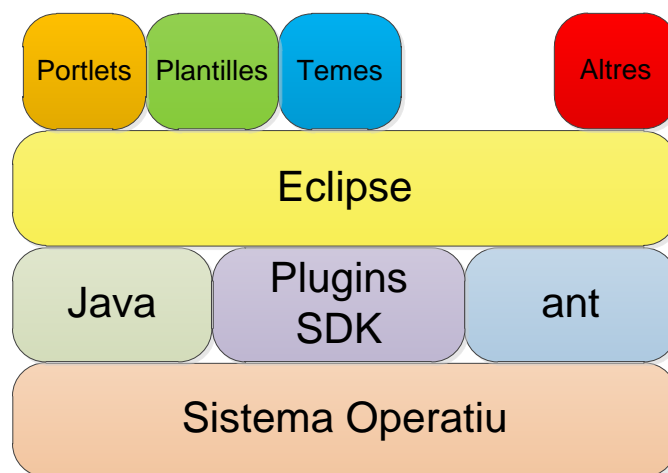
A nivell de desenvolupament no és indispensable, tot i que és molt aconsellable utilitzar també una base de dades. En el cas del prototipus s'ha optat per MySQL. Mitjançant la definició d'una font de dades (*datasource* en anglès) al GlassFish, Liferay l'utilitzarà com a repositori de les seves dades.

Els diferents components que desenvolupem i/o despleguem a Liferay s'instal·laran sobre GlassFish, i passaran a trobar-se disponibles per a l'eina de portal.

Finalment, els usuaris accediran a diferents aplicacions web, definides mitjançant la configuració d'unes unitats lògiques anomenades Organitzacions (veure el subapartat 4.3 Membres: Usuaris, Rols i Organitzacions).

3.2 Entorn de desenvolupament

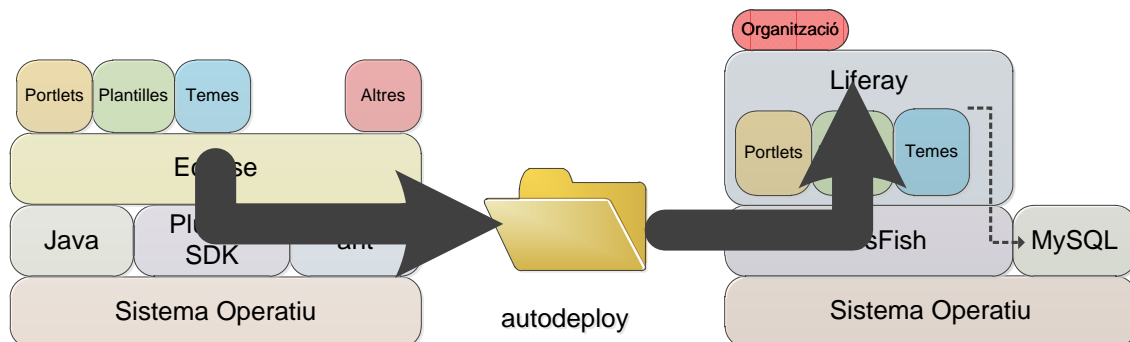
L'entorn de desenvolupament utilitza el muntatge vist al punt anterior per a desplegar els diferents components que es realitzin:



Sobre el sistema operatiu, i gràcies a tecnologies com ant, Plugins SDK (de la que es parlarà a l'annex A, punt A.7 *Mètodes de desenvolupament: Plugins*

SDK, Liferay IDE i Entorns Ext) i Java, es poden desenvolupar components de diversos tipus.

Per a instal·lar un component a Liferay s'utilitza *ant*, que s'encarrega de realitzar les tasques que tingui assignades (compilació, acoblament, etcètera) i de copiar el resultat en una carpeta *autodeploy* on el servidor escolta per a trobar nous components a instal·lar.



Per al desenvolupament del prototipus s'ha utilitzat Eclipse. Igualment, és possible desenvolupar components per a Liferay sense Eclipse.

3.3 Arquitectura dels components

Tal i com s'ha vist en els dos punts anteriors, existeixen diferents tipus de components que formaran el prototipus. Anem a veure'ls una mica més en detall:

Portlets

Cada portlet és una petita aplicació web que s'executa sobre el portal. Es poden tenir varis a la vegada sobre una mateixa plana.

Als annexos E i F es veuran molt més en detall aquests components.

Temes

Els temes doten a cada pàgina d'un aspecte visual, mitjançant bàsicament HTML i CSS.

El desenvolupament de temes es basa en una sèrie de regles jeràrquiques a partir d'un tema en blanc que proporciona Liferay.

El seu desenvolupament pot semblar menys complicat que el d'un portlet, però és indispensable comptar amb un bon tema per a obtenir un producte de bona qualitat.

A l'annex C, *Desenvolupament d'un tema a mida*, s'aprofundeix en el desenvolupament i la configuració d'un tema base.

Plantilles

Les plantilles són uns components que donen estructura a una part dels temes on s'afegeixen els portlets. Es veurà com construir-les a l'annex D. Dels tres components que s'han treballat en aquest TFG, són els més senzills.

Hooks

Els components hooks serveixen per a modificar el comportament de Liferay. Així doncs, si es vol sobreescrivir una configuració o un comportament determinat de Liferay, és possible fer-ho de forma senzilla mitjançant aquest tipus de components.

No s'ha utilitzat cap component de tipus *hook* per a aquest projecte.

Estructures i plantilles web

Alhora de mostrar un contingut dins d'un apartat/pàgina pot ser útil generar una estructura i una plantilla associada a aquesta estructura, de forma que futurs continguts del mateix tipus puguin reutilitzar el format que s'ha generat de forma fàcil.

Aquest és el cas per a, per exemple, els continguts de centre. S'ha creat doncs una estructura i una plantilla per al cas del centres.

Al subapartat *A.5 Continguts Web, Estructures web i Plantilles web de l'annex A* es pot veure que la utilització de plantilles web i estructures és molt senzilla i que dona moltes possibilitats de presentació.

Es important NO confondre plantilla web amb les plantilles (templates), un tipus de component del que s'ha parlat abans.

3.4 Tecnologies i programari utilitzat

Per a finalitzar aquest apartat, es dona a continuació un llistat amb els diversos programaris, tecnologies i altres que s'han utilitzat durant el TFG.

NOM	DESCRIPCIÓ
VMware Player / Oracle VM VirtualBox	Programaris de virtualització sobre els que s'han muntat les màquines virtuals amb el prototip i les eines de desenvolupament. S'han utilitzat els dos, degut als problemes que s'han donat en aquest punt.
Ubuntu 12.04.2	Sistema Operatiu per a la màquina virtual.
Java SE 6	Versió 6 de la plataforma Java.

Glassfish v3.1.2	Servidor d'aplicacions sobre el que s'ha muntat el programari de portal.
MySQL 5.5	Base de dades. També s'han fet proves amb Hypersonic Database, una base de dades per a desenvolupament 100% Java.
Liferay 6.1 CE GA2	Versió de lliure utilització del programari de portal. Aquesta és l'última versió estable.
Eclipse Juno	IDE de desenvolupament, s'ha utilitzat juntament amb Plugins SDK. L'altra opció hauria estat utilitzar Liferay IDE, que es basa en Eclipse e incorpora tot un seguit de mòduls per al desenvolupament de temes, portlets, etcètera.
Hibernate / Spring Service Builder	Liferay incorpora una eina anomenada Service Builder que treballa per sota amb Hibernate i Spring.
Velocity	Llenguatge d'script que s'ha utilitzat per a la creació de la estructura web i de la plantilla associada.
Control de Versions (SVN)	S'ha utilitzat un servidor gratuït que oferia un repositori Subversion. També s'han fet còpies manuals del codi font.

4. Muntatge de l'organització des de zero.

Si una empresa, entitat o persona vol desenvolupar una web sobre Liferay, el primer que ha de fer és saber que vol. Així doncs, haurà de seguir una metodologia, més o menys semblant a la que s'ha seguit per al desenvolupament del prototipus.

Això implica recopilar les necessitats del producte/web, realitzar un anàlisi, i posar a treballar a les diferents parts de la organització per a poder fer la web.

Deixant de banda anàlisis i desenvolupaments, una tasca important que haurà de realitzar-se de forma contínua serà la de gestionar el portal i les organitzacions que el formen.

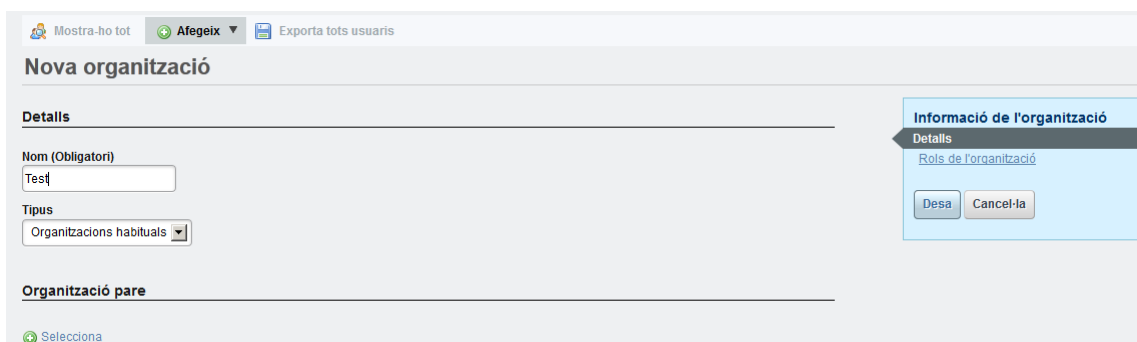
En aquest apartat es detallaran els passos necessaris per a muntar una organització com la del prototipus, sense haver-se de preocupar del tema (que algú muntarà) ni de les funcionalitats (que es tindran instal·lades i disponibles al portal).

4.1 Planejament inicial

El primer que s'ha de fer alhora de muntar una nova web és conèixer bé quins seran els seus apartats, continguts, etcètera. En aquest cas, es té tot l'anàlisi funcional que serveix de guia per al muntatge de l'organització.

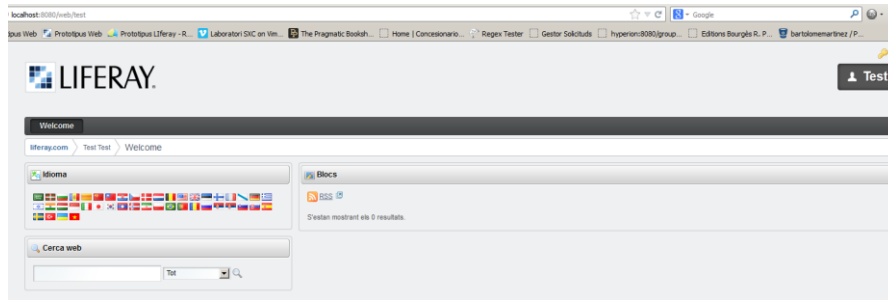
4.2 Creació de la organització i del lloc web

S'haurà de muntar una nova organització per a la nova web. En aquest cas s'anomenarà *Test*, ja que la organització prototipus ja es troba muntada al portal. Això es farà des de *Portal > Users and Organizations* (si es té Liferay configurat en català hi ha alguns menús que no es troben traduïts).



The screenshot shows the 'Nova organització' (New Organization) form in Liferay. At the top, there are navigation links: 'Mostra-ho tot', 'Afegeix', and 'Exporta tots usuaris'. The form is titled 'Nova organització' and has a 'Detalls' section. Under 'Detalls', there is a 'Nom (Obligatori)' field with the value 'Test' and a 'Tipus' dropdown menu set to 'Organitzacions habituals'. On the right side, there is a 'Informació de l'organització' sidebar with 'Detalls' and 'Rols de l'organització' options, and 'Desa' and 'Cancel·la' buttons. At the bottom left, there is a 'Selecciona' link.

Aquesta organització es trobarà visible a <http://localhost:8080/web/test>. No tindrà cap pàgina associada encara.

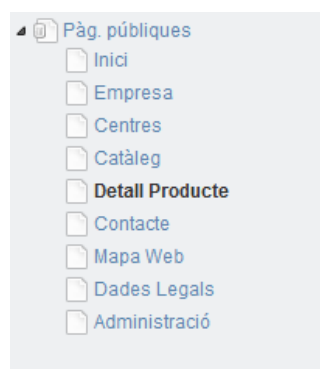


També serà necessari generar un nou lloc web. Una vegada creat el lloc web, serà necessari (en cas que es vulgui associar a un domini) posar el nom del seu domini a la pestanya *URL del lloc*.



4.3 Creació de les pàgines

Des del menú lateral d'administració és possible generar les pàgines necessàries. Així doncs, s'hauran de donar d'alta tot un seguit de pàgines públiques a partir de l'anàlisi, (*Inici, Empresa, Catàleg*, etcètera). És important donar d'alta també una pàgina *Administració*, oculta, que es configurarà per accedir a l'administració a posteriori, amb un portlet per identificar-se.



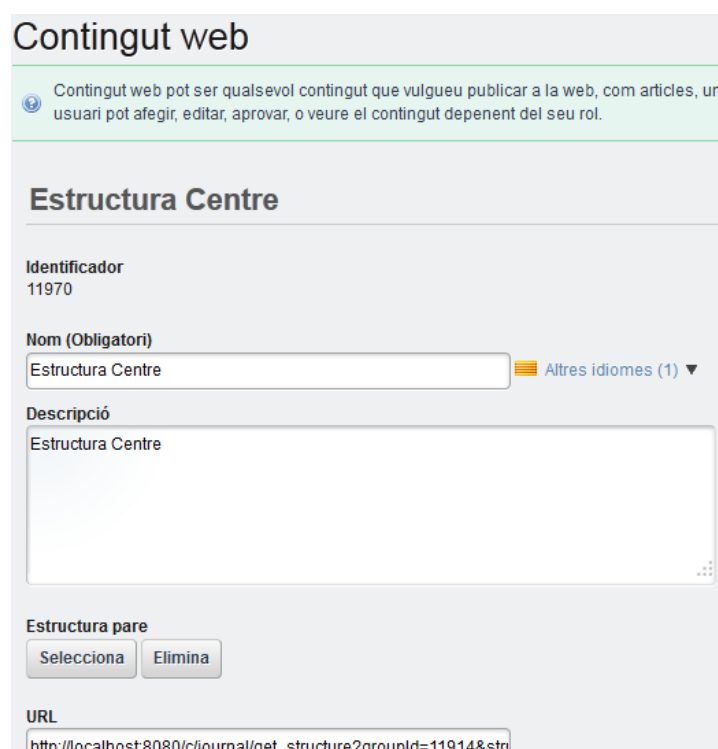
Des del propi apartat de pàgines és possible seleccionar el tema que es vol mostrar. En aquest cas, seleccionarem el Tema *Prototipus* (NOTA: les metadades que indiquen l'autor es generen automàticament amb el *Plugins*

SDK, s'haurien de canviar per a indicar que aquest tema s'ha muntat per aquest TFG!).



4.4 Continguts. Estructures i plantilles web.

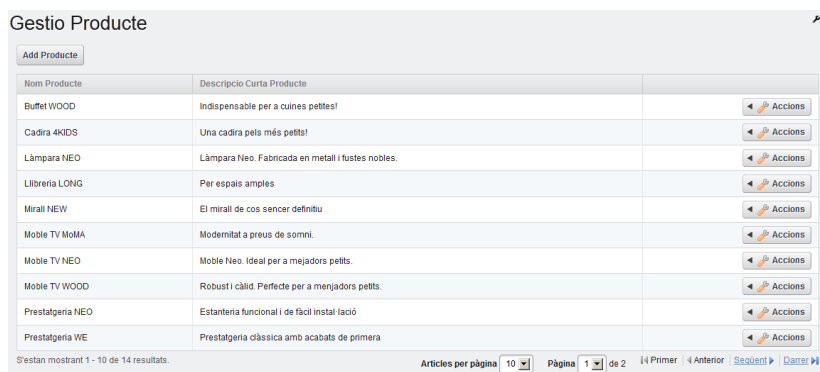
Ara que ja es té muntat l'esquelet bàsic, és el moment de muntar les pàgines. Abans, però, s'haurà de crear la estructura web i la plantilla web per a la pàgina de centres amb *Velocity*, i muntar-les a *Contingut Web > Estructura* i a *Contingut Web > Plantilla* respectivament. A continuació una imatge de la pantalla d'alta d'estructura.



Hi ha més informació sobre Estructures Web i Plantilles Web als annexes.

4.5 Muntatge dels portlets

Ara si, ja es poden muntar les planes. Mitjançant el menú superior de cada plana és possible seleccionar els portlets que es necessitin per a cada plana. Depenent de la funcionalitat de cada portlet, aquests s'hauran de configurar, com és el cas d'alguns dels portlets que s'han generat pel prototipus, que requereixen del domini on es troben disponibles les diferents imatges dels productes. A continuació es pot veure la plana de vista i la de preferències del portlet de *Gestió de Producte*.



Nom Producte	Descripció Curta Producte	
Buffet WOOD	Indispensable per a cuines petites!	Accions
Cadira 4KIDS	Una cadira pels més petits!	Accions
Làmpara NEO	Làmpara Neo. Fabricada en metall i fustes nobles.	Accions
Llibreria LONG	Per espais amples	Accions
Mirall NEW	El mirall de cos sencer definitiu	Accions
Moble TV MoMA	Modernitat a preus de somni.	Accions
Moble TV NEO	Moble Neo. Ideal per a menjadors petits.	Accions
Moble TV WOOD	Robust i càlid. Perfecte per a menjadors petits.	Accions
Prestatgeria NEO	Estanteria funcional i de fàcil instal·lació	Accions
Prestatgeria WE	Prestatgeria clàssica amb acabats de primera	Accions

S'estan mostrant 1 - 10 de 14 resultats. Articles per pàgina: 10 Pàgina: 1 de 2 Primer Anterior Següent Darrer



Gestio Producte

Directorio del servidor per a imatges:

Aquest portlet, tal i com es pot veure a la segona imatge, requereix que se li indiqui on s'han de deixar les imatges, a *Preferències*.

4.6 Usuaris i rols

Depenent de les necessitats de l'organització, serà necessari crear nous usuaris, i associar-los a determinats rols depenent de les funcions que es vol que compleixin. Als annexos es pot ampliar informació sobre aquest punt.

4.7 Altres

La gestió d'una organització pot incloure altres tasques a configurar, potser a Liferay o al servidor on es trobi muntat. En el cas del prototipus, per exemple, ha estat necessari configurar un Servidor Virtual per a servir les imatges que s'han pujat al servidor via un domini apart.

Un altra opció hauria estat estudiar d'integrar les imatges a l'apartat de documents de Liferay.

A continuació es pot veure una captura del virtual server que s'ha configurat.

Configuration Name: server-config

Id: imatges.prototipus

Hosts: *
Comma-separated list of hosts or IP addresses

State:

SSO:

SSO Cookie Http Only: **Enabled**
Support for HttpOnly flag for JSESSIONIDSSO cookie

Network Listeners:
Comma-separated list of network listeners

Default Web Module:

Log File:
Default is the log-root attribute of the domain

Docroot:
Absolute path to root document directory for server

Access Log

Access Logging:

Directory:
Absolute path to server access logs

Ara si, es tindria una organització plenament operativa. No s'ha d'oblidar, però, que serà necessari realitzar un treball de manteniment constant a tota l'organització.

5. Conclusions

Tal i com s'ha pogut veure al llarg de tot el document, les eines de portal confirmen tot un univers que es pot utilitzar per a muntar molts tipus d'aplicacions web diferents.

Aquest TFG intenta ser una eina per a l'aprenentatge de tot aquest univers. No s'ha volgut només que aquest coneixement fos adquirit per l'estudiant, sino que s'ha volgut generar un entorn i una documentació (els Annexos) que pogués ser el punt de partida per a nous reptes personals o col·lectius.

Inicialment el TFG van plantejar una sèrie de reptes a assolir: configurar adequadament un entorn Liferay, generar un tema que s'integrés amb el portal bé i desenvolupar portlets que exploressin diferents àrees i tecnologies.

Molts dels objectius inicials del TFG s'han assolit, però no tots. A la llista d'objectius no assolits també cal afegir nous reptes que es plantegen a partir dels coneixements adquirits.

Així, al primer grup, el de objectius no assolits, quedaria sobretot per tancar la integració de Liferay amb els projectes *Solr* i *Lucene*. El buscador de Producte és a dia d'avui molt senzill, i integrar aquests productes seria molt interessant.

També s'hauria d'estudiar quins problemes han ocasionat que el prototipus sobre Linux no hagi donat un rendiment acceptable, el que ha fet que s'hagi acabat treballant sobre el Windows que feia de host.

Al segon grup s'inclouen tot un seguit de nous objectius, com són:

- Provar a realitzar portlets MVC
- Aprofundir coneixements en Spring
- Aprofundir coneixements en Hibernate
- Aprofundir coneixements sobre l'eina Service Builder de Liferay.
- Desenvolupar portlets que facin ús d'AJAX

Es pot dir que tot i que el resultat és satisfactori s'obren molts nous fronts d'estudi i anàlisi.

Per últim, la metodologia de desenvolupament –clàssica, basada en presa de requeriments, anàlisi, desenvolupament i documentació- s'ha seguit sense complicacions greus, tot i que s'han trobat problemes – de rendiment amb la Màquina Virtual Linux- que han obligat a prendre decisions ràpides per a no desviar-se de la planificació inicial.

6. Glossari

- **AJAX (Asynchronous JavaScript And XML):** tècnica mitjançant la qual es poden llençar crides de forma asíncrona al servidor.
- **Alfresco:** programari de gestió de continguts de codi lliure, molt potent i utilitzat a dia d'avui.
- **Ant:** programari d'execució de tasques en paquets, projecte de la Fundació Apache.
- **CMS (Content Management System):** sistema de gestió de continguts, programari que serveix per a controlar tot el cicle de vida dels continguts d'un sistema.
- **Eclipse:** programari de desenvolupament de software, molt utilitzat avui en dia.
- **Estructura Web:** fitxer XML que conté els diferents atributs d'una entitat determinada. S'utilitza per a definir continguts amb el format d'aquesta entitat a Liferay.
- **GlassFish:** una de les moltes implementacions d'un servidor d'aplicacions que hi ha avui dia a l'univers Java EE.
- **Hibernate:** eina ORM (Object Relational Mapping), és a dir, que serveix per a *mapejar* objectes dins d'una base de dades relacional clàssica.
- **Java:** plataforma de programari molt utilitzada avui en dia.
- **Java EE:** versió Enterprise (orientada a usos empresarials) de la plataforma Java.
- **JCP (Java Community Process):** procés per a certificar/estandarditzar processos tècnics de la plataforma Java.
- **JSR 168:** especificació de portlets 1.0.
- **JSR 286:** especificació de portlets 2.0.
- **Liferay:** programari de portal de codi lliure.
- **Linux:** sistema operatiu molt utilitzat a dia d'avui, sobretot a servidors i sistemes crítics.
- **Lucene:** API de codi obert que permet recuperar informació, implementa indexació i cerca. Pertany a la Fundació Apache.
- **Plantilla:** component Liferay que dona format a l'àrea on es carreguen els portlets dins d'una pàgina.
- **Plantilla Web:** plantilla en Velocity que dona format als atributs definits per una Estructura Web a Liferay.
- **Plugins SDK:** eina de desenvolupament per a Liferay que permet la generació de l'esquelet d'un component determinat (portlet, tema, etcètera) mitjançant ant.
- **Portlet:** miniaplicació web que requereix d'un contenidor de portlets per a executar-se.
- **Service Builder:** eina de Liferay per a implementar la capa de serveis i de persistència definint entitats i altres elements a un fitxer xml.
- **Single Sign-On:** procediment mitjançant el qual un usuari s'autentica una sola vegada per accedir a un conjunt de sistemes.
- **Solr:** programari per a realitzar cerques, utilitza Lucene. Pertany a la Fundació Apache.

- **Spring:** *framework* per a desenvolupar aplicacions.
- **SVN:** acrònim de **SubVersioN**, un programari de control de versions.
- **Tema:** component que serveix per a donar un aspecte determinat a una pàgina u organització Liferay.
- **TFG:** acrònim de *Treball Final de Grau*.
- **Velocity:** llenguatge per a la escriptura de plantilles. Projecte de la fundació Apache.
- **Visor de Contingut Web (Web Content Display):** tipus de portlet que utilitza Liferay que mostra un contingut web determinat a una plana.
- **VM:** acrònim de *Màquina Virtual*.

7. Bibliografia

Bauer, C., & King, G. (2005). Hibernate in action.

Java Community Process. JSR-000168 Portlet Specification, (Final Release)
(Març-Abril 2013)
<http://jcp.org/aboutJava/communityprocess/final/jsr286/index.html>

Java Community Process. JSR-000286 Portlet Specification, (Final Release)
(Març-Abril 2013)
<http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html>

Johnson, R., Hoeller, J., Arendsen, A., & Thomas, R. (2009). Professional Java Development with the Spring Framework. John Wiley & Sons.

Sezov, R. (2012). Liferay in action.

Walls, C., & Breidenbach, R. (2005). Spring in action. Dreamtech Press.

Yuan, J. X. (2009). Liferay Portal 5.2 systems development. Packt Publishing Ltd.

Annex A. Liferay: conceptes bàsics

Liferay és un sistema complex amb molt conceptes, funcionalitats i tecnologies amb els que és necessari estar familiaritzat per a poder assolir el desenvolupament d'un projecte amb un mínim de garanties.

En els següents punts es detallaran els conceptes més importants.

A.1 Portal

Durant tot el document s'ha parlat de portal, donant a entendre que es té clar per part del lector que és un programari de portal.

Per portal s'entén aquell programari web que permet als usuaris i empreses crear i emmagatzemar contingut, a la vegada que es posa a la seva disposició tot un seguit de funcionalitats molt diverses.

Gràcies a la capacitat dels portals d'utilitzar funcionalitats prèviament encapsulades, es permet a usuaris i empreses la reducció de costos i temps, millorant a la vegada la satisfacció dels usuaris finals i dels diferents processos de negoci.

Liferay és un dels programaris de portal líder actualment (*Gartner Magic Quadrant for Horizontal Portals*, Setembre 2012) dins d'aquest tipus d'eines.

A.2 Administració (Tauler de Control)

Per accedir a Liferay amb permisos i no com a *Guest* (convidat o usuari anònim), és necessari identificar-se. En el cas del prototipus, la pantalla d'identificació es diu administració, i es troba oculta tal i com s'ha comentat en algun punt d'aquest document (<http://prototipus:8080/administracio>).

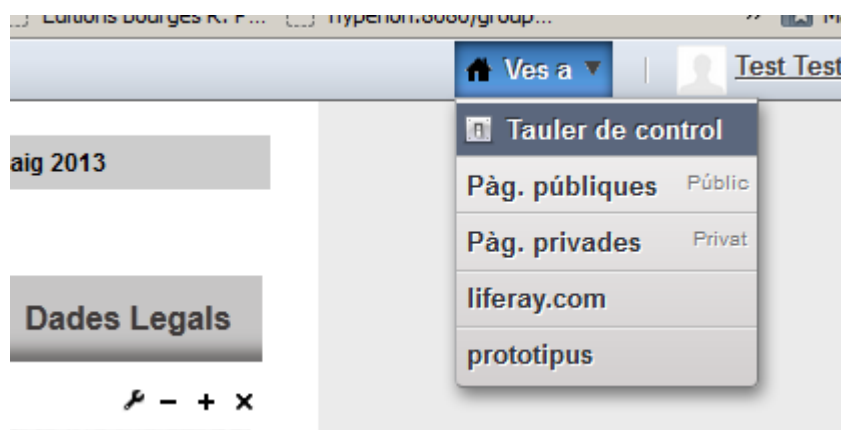
Una vegada identificats al sistema, es veuran dos canvis significatius a les pàgines.

Per una banda, alguns dels portlets tindran diferents botons de configuració, preferències, etcètera, en funció dels permisos que tinguem assignats. A la següent imatge es poden veure marcats diferents grups de botons amb accions per a un lateral una vegada ens hem identificat com administradors.



En segon lloc, a dalt de tot de les pàgines es podrà accedir a un menú d'administració, una de les opcions del qual, Tauler de Control, ens porta directament a la administració de Liferay.

De nou, depenent dels permisos del nostre usuari tindrem en aquest espai més o menys accions visibles.



A.3 Membres: Usuaris, Rols i Organitzacions

Liferay té la capacitat d'organitzar la informació dels seus membres de diverses formes, gràcies a una sèrie de conceptes que implementa com organització, usuari, rols, llocs web, localitzacions etcètera.

Aquesta organització de la informació pot arribar a tenir certa complexitat. En el cas de prototip, s'han estudiat i treballat quatre conceptes bàsics: Usuari, Rol, Organització i Lloc web..

Un usuari és la representació d'un usuari físic al sistema. Així, un usuari té la seva pròpia informació, i es pot assignar a diferents organitzacions, com es veurà a continuació.

A continuació es pot veure una imatge amb la pantalla d'edició d'un dels usuaris del prototipus, *Gestor Prototipus*.

Aquest usuari té una organització associada (la del prototipus), i també té assignats una sèrie de rols que li donen la possibilitat de realitzar certes accions sobre el sistema (modificació de continguts, parametrització de portlets, etcètera).



Un rol és una entitat que s'utilitza per a definir permisos. Existeixen dos àmbits o àrees on un rol pot treballar:

- Portal : Els permisos assignats a aquesta àrea afecten a tot el portal.
- Organització: Els permisos signats a aquesta àrea afecten a una organització en concret.

Fins la versió 5.2.3 de Liferay s'utilitzava també l'àmbit de *Comunitat*, però sembla ser que aquest àmbit ha estat eliminat de la versió 6.

D'aquesta forma, es poden detallar permisos per a cada usuari per a tot el portal i no només per a una organització en concret.

A continuació es pot veure un llistat amb els quatre rols que s'han definit per al portal, per a Gestor(Editor) Prototipus i per a Administrador Prototipus (IMPORTANT: no confondre amb Administrador Portal).

Nom ▲	Tipus	Descripció
 Administrador Prototipus	Organització	Administrador Prototipus
 Administrador Prototipus Role	Habitual	
 Administrador	Habitual	Administrators are super users who can do anything.
 Gestor Prototipus	Organització	Gestor per a donar d'alta contingut i treballar amb els productes i contactes
 Gestor Prototipus Role	Habitual	

I per a veure el detall de permisos d'un d'aquests rols, la següent imatge ens dona una part de la relació de permisos sobre diferents portlets per a el rol Gestor Prototipus, de tipus Organització.

Gestor Prototipus

[Edita](#)
[Definiu permisos](#)

Afegeix permisos

Resum

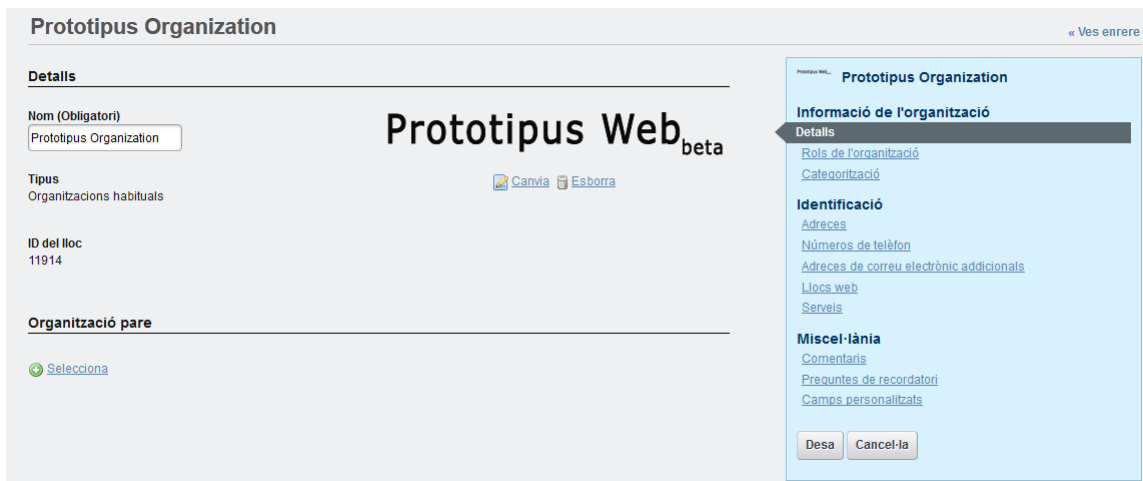
S'estan mostrant els 13 resultats. Articles per pàgina

Resource Set	Recurs	Acció
Contacte		Veure
Contingut web	Contingut web	Actualitza
Detall Producte		Veure
Gestio Contacte		Accedeix des del tauler de control
Gestio Contacte		Veure
Gestio Producte		Accedeix des del tauler de control
Gestio Producte		Veure
Llistat Producte		Veure
Mini Detall Producte		Afegir a una pàgina

Per últim, es troben les organitzacions i els seus llocs web associats. Una organització és un a col·lecció jerarquizada d'usuaris. A més a més, poden tenir pàgines associades.

D'aquesta manera, mitjançant una entitat organització, amb certes pàgines i usuaris relacionats, es pot generar una web.

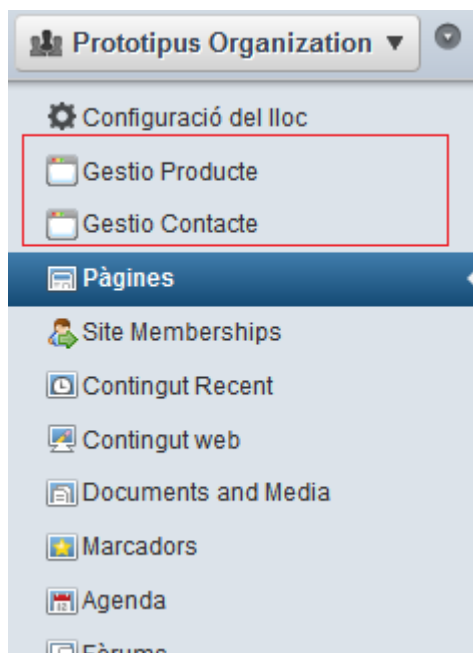
En el cas del prototipus, s'ha generat una organització amb les pàgines enumerades en l'apartat d'arquitectura.



Com es pot veure en la imatge anterior, una organització compta amb un nom, un logo i tot un seguit de meta informació associada.

Per altra banda, al menú lateral es pot seleccionar la organització sobre la que es treballa, el que ens dona accés a les seves pàgines, tema associat, continguts, i en el nostre cas, entre altres opcions, productes i contactes.

A la següent imatge es pot veure un detall del menú lateral, amb les opcions de gestió de producte i de contacte ressaltades, entre altres opcions.



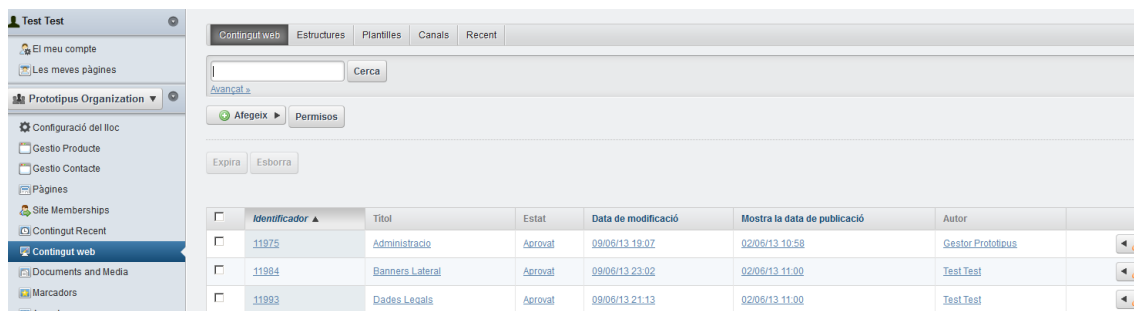
A.4 Gestor de continguts (CMS)

Liferay, com a eina de portal que és, pot gestionar contingut (textos, imatges, documents i altres) i mostrar-los als usuaris en funció de múltiples paràmetres:

- rols dels usuaris
- organització a la que pertanyen els continguts

- permisos dels continguts
- dates de publicació dels continguts
- etcètera

A la imatge del lateral de la pàgina anterior es pot observar com una de les opcions que existeixen és la de *Contingut Web*. Mitjançant aquesta opció es pot accedir a un llistat amb tots els continguts per a una organització. El llistat compta amb un cercador a la part superior, molt útil en cas de comptar amb una organització molt gran amb molts continguts.



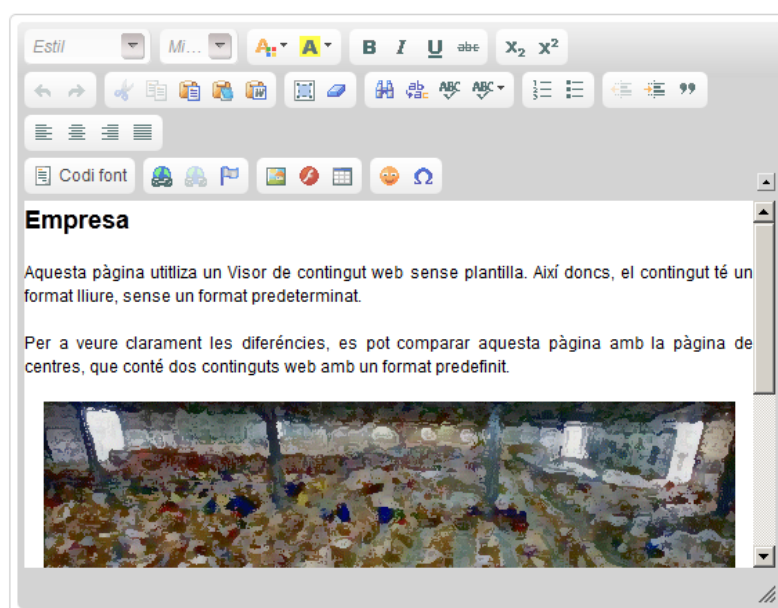
Identificador	Títol	Estat	Data de modificació	Mostra la data de publicació	Autor
11975	Administracio	Aprovat	09/09/13 19:07	02/09/13 10:58	Gestor Prototipus
11984	Banners Lateral	Aprovat	09/09/13 23:02	02/09/13 11:00	Test Test
11993	Dades Legals	Aprovat	09/09/13 21:13	02/09/13 11:00	Test Test

Dins d'aquest apartat és possible donar d'alta nous continguts, i realitzar tot un seguit d'accions sobre els ja existents: *modificar*, *clonar*, *expirar*, *esborrar*, etcètera.

A.5 Continguts Web, Estructures web i Plantilles web.

Els continguts web són, per defecte, una capsa d'HTML sense un format preestablert. Això dona molta llibertat al editor del contingut, que pot fer i desfer sense cap tipus de restricció.

A la següent imatge, es pot veure una part d'un dels continguts web per defecte del prototipus, que no deixa de ser un contingut editat amb un element WYSIWYG (*What You See Is What You Get*).



De vegades, però, pot ser necessari donar cert format a un contingut, de manera que els editor no hagin de donar format a tot un HTML, i es puguin centrar en l'edició pròpiament dita.

Per això existeixen les Estructures Web i les seves Plantilles Web associades. Un exemple que es pot trobar al prototipus (de fet, és l'exemple d'aquesta funcionalitat que s'ha implementat) són els continguts de centre.

Aquests continguts defineixen per una banda una estructura en XML, i per altra una plantilla en Velocity (un llenguatge de plantilles per a Java d'Apache) que li dona format.

Així, a partir d'un XML que fa d'estructura

```
1 <?xml version="1.0"?>
2
3 <root>
4   <dynamic-element name="nombre_centre" type="text_box" index-type="" repeatable="false"/>
5   <dynamic-element name="imatge_centre" type="image" index-type="" repeatable="false"/>
6   <dynamic-element name="mapa" type="image" index-type="" repeatable="false"/>
7   <dynamic-element name="direccio" type="text_box" index-type="" repeatable="false"/>
8   <dynamic-element name="telefon" type="text_box" index-type="" repeatable="true"/>
9   <dynamic-element name="fax" type="text_box" index-type="" repeatable="true"/>
10  <dynamic-element name="cp" type="text_box" index-type="" repeatable="false"/>
11  <dynamic-element name="email" type="text_box" index-type="" repeatable="true"/>
12  <dynamic-element name="descripcio" type="text_area" index-type="" repeatable="false"/>
13 </root>
```

I d'una plantilla Velocity associada

```
3     <div class="estil_029">
4         <h2 class="text_031">${nombre_centre.getData()}</h2>
5         <div></div>
7     <div></div>
8     <div class="salt_linia_001"><!-- --></div>
9     </div>
10    <div class="estil_012">
11        <p>${direccio.getData()}</p>
12        #foreach( $telefonaux in $telefon.getSiblings() )
```

Es pot donar d'alta contingut amb un format determinat. A continuació es pot veure com la pantalla d'edició d'un contingut té, ara, un format completament diferent.

Títol (Obligatori)

nombre_centre

Traduïble

imatge_centre

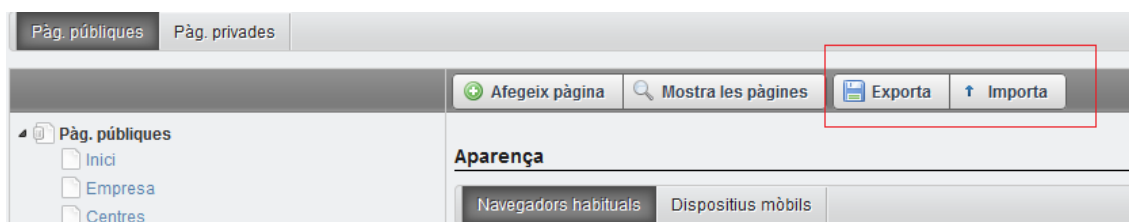
[[Mostra](#)]

Traduïble

A.6 Importació i exportació de continguts

Muntar una organització sobre Liferay de forma adequada és un treball més fàcil amb totes les opcions que dona el portal, però tot i això s'haurà de dedicar molt de temps per a que el resultat sigui òptim.

Així doncs, és una bona idea realitzar, de forma periòdica o com a base per a crear noves organitzacions exportacions de tot el contingut duna organització. Aquesta opció es troba disponible sota l'opció de Pàgines del menú lateral d'administració.



Des d'aquest apartat també es poden crear noves pàgines, modificar i eliminar les ja existents i realitzar tot un seguit d'accions, com per exemple afegir un codi JavaScript per a una pàgina en concret.

A.7 Mètodes de desenvolupament: Plugins SDK, Liferay IDE i Ext

Una vegada vistos diversos punts de l'administració del portal (hi ha molts més, però amb els punt vistos és possible realitzar una bona gestió de Liferay) és el moment de fer-se una pregunta important: com es desenvolupen noves funcionalitats per a Liferay?

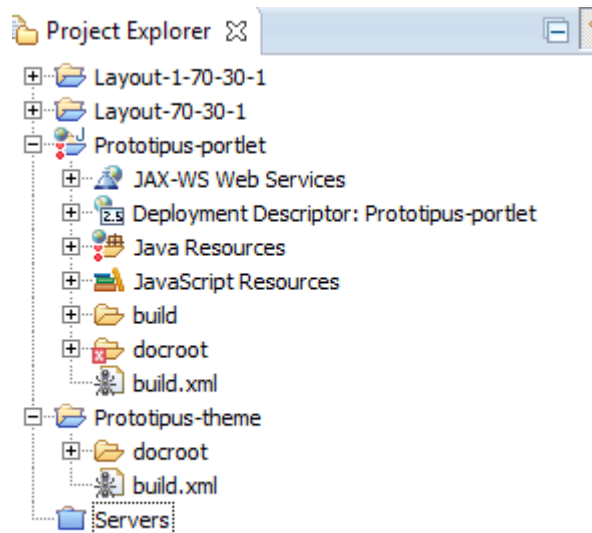
A partir de l'experiència acumulada en el desenvolupament sobre aquesta eina, es pot dir que existeixen dues formes d'afrontar un desenvolupament sobre Liferay: mitjançant un entorn Ext i mitjançant Plugins SDK.

Plugins SDK és un mòdul que es pot descarregar des de la web de Liferay que permet la generació de projectes base per a Eclipse. Així doncs, mitjançant

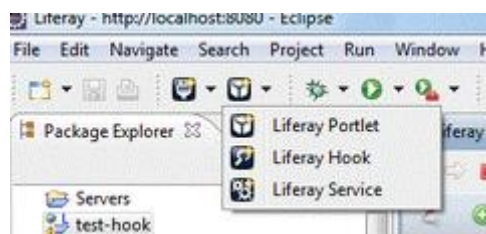
aquest entorn, que utilitza ant per a generar projectes es poden generar temes, portlets, plantilles i tota la resta de mòduls que Liferay suporta.

Aquests projectes es podran a posteriori importar en Eclipse, des d'on es podrà començar a implementar.

A la següent imatge es poden veure diversos projectes al project explorer d'Eclipse.



Aquest mètode de desenvolupament s'està substituint des de la darrera versió de Liferay per un entorn similar però integrat totalment a Eclipse. És el conegut com a Liferay IDE.



A dia d'avui, sembla ser que ja comença a ser un entorn prou estable com per a desenvolupar sobre sense problemes.

L'últim mètode de desenvolupament que es comentarà és el desenvolupament mitjançant l'entorn d'extensió. Aquest mètode de desenvolupament es troba en desús, i s'està substituint a les últimes versions del producte per un tipus de mòduls anomenats Ext que es poden trobar al Plugins SDK.

Bàsicament, el que es fa amb l'entorn d'extensió és treballar sobre una versió local del codi font del producte, de forma que per una banda es té accés a moltes més funcionalitats, mètodes i altres, i per altra es genera una versió pròpia de Liferay, el que pot suposar no poder aplicar noves actualitzacions, problemes de migracions, i tot una llista de problemes molt llarga.

Així doncs, es recomana utilitzar per a desenvolupar qualsevol de les dues primeres opcions, i utilitzar la última només en casos extrems on no hi hagi cap altre alternativa.

Com a últim punt d'aquest apartat, indicar que és molt possible que mitjançant Hooks (un altre tipus de component del paquet Plugins SDK) es puguin realitzar modificacions al codi font sense haver de casar-se amb els inconvenients de l'entorn EXT. Els hooks (Ganxo en anglès) es poden instal·lar i desinstal·lar fàcilment, al contrari que els mòduls Ext.

A.8 Service Builder

Service Builder és una eina generada per Liferay mitjançant la qual es poden generar objectes Hibernate, serveis web, classes Spring i altres.

A partir d'un fitxer XML anomenat service.xml, es poden redefinir diferents accions. En el nostre cas, s'ha utilitzat per a generat tota la capa de persistència per a les dues funcionalitats de producte i contacte.

Les diferents opcions que dona l'XML en qüestió es troben documentades al fitxer DTD.

A l'apartat de portlets es parlarà més en detall d'aquesta eina. Per ara, només dir que especificant correctament les entitats al fitxer service.xml i mitjançant la definició dels mètodes que es volen implementar, Liferay s'ocupa de generar tota la capa de serveis.

Aquest eina és molt potent com es podrà veure al prototipus.

Annex B. Instal·lació i configuració d'un entorn de desenvolupament

Per a comptar amb un entorn de desenvolupament 100% operatiu és necessari realitzar tot un seguit de passos, els quals es detallen en aquest annex. Liferay suporta múltiples entorns. Aquest annex és doncs una guia, però es poden seguir altres vies per a tenir un entorn operatiu.

Per a realitzar aquest annex B s'ha utilitzat una màquina virtual Ubuntu 12.04, descarregada del repositori VirtualBoxes de sourceforge.net (<http://sourceforge.net/projects/virtualboximage/>).

Aquesta màquina virtual utilitza l'usuari i contrasenya *ubuntu / reverse*.

B.1 Repositoris

Per a instal·lar els diversos programaris d'aquesta MV s'ha donat accés als següents repositoris a Ubuntu:

```
#sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu hardy main multiverse"  
#sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu hardy-updates main multiverse"  
#sudo add-apt-repository "deb http://archive.canonical.com/ lucid partner"
```

A continuació, s'han actualitzat les fonts.

```
#sudo apt-get update
```

B.2 Instal·lació JDK

En cas de que el sistema no compti amb un JDK instal·lat, serà necessari instal·lar-lo:

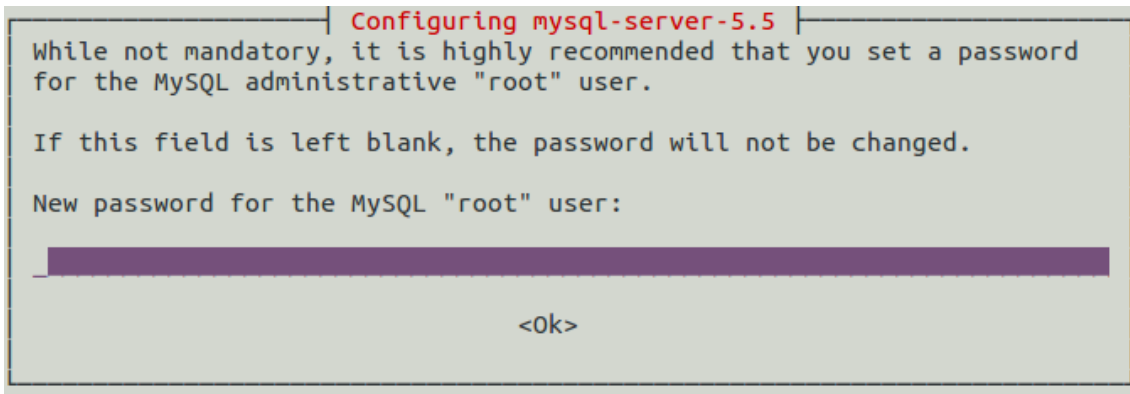
```
#sudo apt-get install sun-java6-jdk
```

B.3 MySQL Server

També s'ha d'instal·lar un servidor MySQL.

```
# sudo apt-get install mysql-server
```

(Deixem el password en blanc)



Una vegada instal·lada, s'haurà de crear un nou usuari.

```
#mysql --user=root
mysql> CREATE USER lportal;
mysql> SET PASSWORD FOR 'lportal' = PASSWORD('lportal');
mysql> GRANT ALL PRIVILEGES ON localhost.* TO lportal @'% ' IDENTIFIED
BY 'lportal';
mysql> commit;
```

Un altra opció en aquest punt és utilitzar el client de MySQL per a crear l'usuari.

B.4 Driver MySQL

Per a configurar el DataSource del GlassFish (o Tomcat) cap a la base de dades serà necessari utilitzar el driver de MySQL, disponible a la URL <http://dev.mysql.com/downloads/connector/j/>

Així doncs, el descarreguem i el guardem per a pròxims passos.

B.5 Client MySQL

Per a poder treballar amb MySQL, anirà molt bé tenir instal·lat *MySQL Workbench*. Aquest programari dona accés a la base de dades mitjançant una interfície visual molt més agradable que la consola.

```
#sudo apt-get install mysql-workbench
```

B.6 Liferay Portal

Per a realitzar aquest TFG, s'han instal·lat dues versions de Liferay, una amb un servidor d'aplicacions GlassFish i un altre amb servidor Tomcat.

La versió que s'ha configurat i que es troba plenament operativa a la màquina virtual és la versió amb Tomcat.

Recordar que s'han produït problemes de rendiment amb la màquina GlassFish durant el desenvolupament, el que finalment va fer que s'optés per desenvolupar sobre Windows 7 en GlassFish. Aquests problemes no són

provocats per Linux, sino per utilitzar una màquina virtual amb un servidor d'aplicacions que consumeix molts recursos sobre un equip de recursos limitats.

Així doncs, per la màquina virtual, s'ha instal·lat Tomcat. Tomcat és molt més lleuger que GlassFish, el que repercuteix directament sobre el rendiment de la màquina (positivament).

Les versions que s'han baixat e instal·lat des de la pàgina de Liferay són:

- Liferay Portal 6.1 Copmmunity Edition 6.1 CE GA2 Bundles with Tomcat
- Liferay Portal 6.1 Copmmunity Edition 6.1 CE GA2 Bundles with Glassfish

Per a instal·lar-les, només cal descomprimir-les, i renombrar els directoris si es vol. En aquest cas s'han copiat a `/opt/liferay/bundles` amb els següents noms

```
ubuntu@ubuntu-VirtualBox:/opt/liferay/bundles$ ls -la
total 16
drwxrwxr-x 4 ubuntu ubuntu 4096 Jun 15 20:15 .
drwxr-xr-x 3 ubuntu ubuntu 4096 Jun 15 20:13 ..
drwxr-xr-x 5 ubuntu ubuntu 4096 Jul 31 2012 liferay-glassfish
drwxr-xr-x 5 ubuntu ubuntu 4096 Jul 31 2012 liferay-tomcat
ubuntu@ubuntu-VirtualBox:/opt/liferay/bundles$
```

És necessari copiar el driver de MySQL que abans s'ha descarregat a `/opt/liferay/bundles/liferay-tomcat/tomcat-7.0.27/lib`

B.7 Plugins SDK

El Plugins SDK també s'ha de baixar i descomprimir des de la pàgina de Liferay, a Downloads > Additional Files

La versió ha estat Plugins SDK 6.1 CE GA2

S'ha descomprimit a `/opt/liferay/`

```
ubuntu@ubuntu-VirtualBox:~$ cd /opt/
liferay/
VBoxGuestAdditions-4.2.12/
ubuntu@ubuntu-VirtualBox:~$ cd /opt/liferay/
bundles/
liferay-plugins-sdk-6.1.1/
ubuntu@ubuntu-VirtualBox:~$ cd /opt/liferay/
ubuntu@ubuntu-VirtualBox:/opt/liferay$ ls -la
total 16
drwxr-xr-x 4 ubuntu ubuntu 4096 Jun 15 20:22 .
drwxr-xr-x 4 root root 4096 Jun 11 19:13 ..
drwxrwxr-x 4 ubuntu ubuntu 4096 Jun 15 20:15 bundles
drwxr-xr-x 12 ubuntu ubuntu 4096 Jul 31 2012 liferay-plugins-sdk-6.1.1
ubuntu@ubuntu-VirtualBox:/opt/liferay$
```


B.8 Eclipse

Per a instal·lar Eclipse s'ha utilitzat la comanda

```
#sudo apt-get install eclipse-platform
```

i també

```
#ln -s /usr/lib/jni/libswt-* ~/.swt/lib/linux/x86/
```

per a les llibreries SWT

B.9 Configuració Tomcat-Liferay

S'ha de generar un fitxer a `/opt/liferay/bundles/liferay-tomcat/tomcat-7.0.27/webapps/ROOT/WEB-INF/classes` anomenat `portal-ext.properties` amb la informació següent per a connectar-nos a la base de dades

```
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://localhost/liferay_portal?useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
jdbc.default.username=lportal
jdbc.default.password=lportal
```

B.10 Configuració Plugins SDK

Plugins SDK necessita saber on es troba instal·lat el servidor i quin tipus de servidor és.

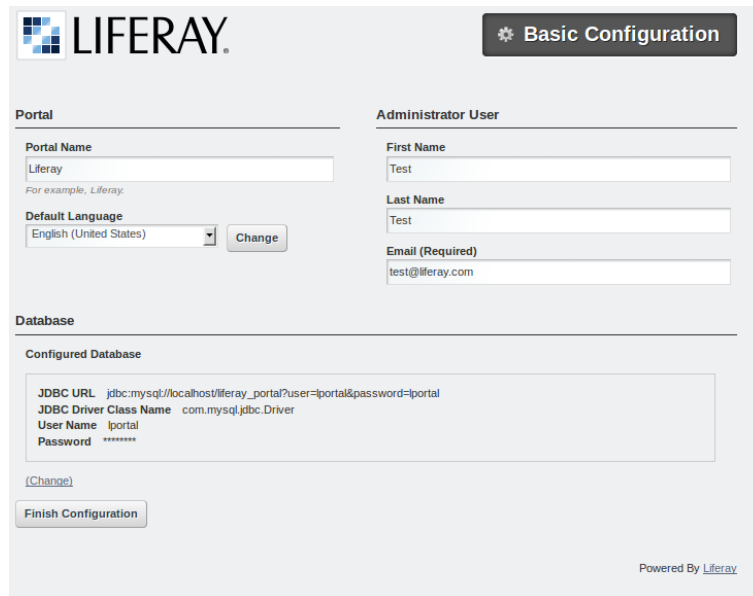
Per això és necessari generar un fitxer `build.{nom.usuari}.properties` on `{nom.usuari}` és el nom de l'usuari de la nostra màquina que s'utilitza.

Així, en aquest cas s'ha generat el fitxer `build.ubuntu.properties` (utilitzant de model l'existent `portal.properties`) a `/opt/liferay/liferay-plugins-sdk-6.1.1/`

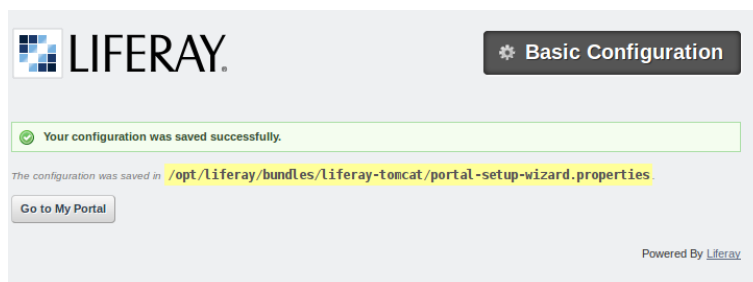
Dins del fitxer s'ha reconfigurat la línia que apunta al appserver tomcat

```
ubuntu@ubuntu-VirtualBox: /opt/liferay/liferay-plugins-sdk-6.1.1$ more build.ubuntu.properties
#
# Specify the paths to an unzipped Tomcat bundle.
#
app.server.type=tomcat
app.server.dir=${project.dir}/../bundles/liferay-tomcat/tomcat-7.0.27
app.server.deploy.dir=${app.server.dir}/webapps
app.server.lib.global.dir=${app.server.dir}/lib/ext
app.server.portal.dir=${app.server.dir}/webapps/ROOT
```

Si en algun moment es volgués utilitzar GlassFish, s'hauria de canviar els valors d'aquests camps per apuntar al mateix. AL fitxer pare *build.properties* hi ha un exemple.



The screenshot shows the Liferay Basic Configuration wizard. It is divided into three main sections: Portal, Administrator User, and Database. The Portal section includes fields for Portal Name (Liferay), Default Language (English (United States)), and a Change button. The Administrator User section includes fields for First Name (Test), Last Name (Test), and Email (Required) (test@liferay.com). The Database section shows the Configured Database with fields for JDBC URL, JDBC Driver Class Name, User Name, and Password. A Finish Configuration button is at the bottom.



The screenshot shows the Liferay Basic Configuration wizard after successful completion. A green message box states "Your configuration was saved successfully." Below this, it indicates the configuration was saved in the file `/opt/liferay/bundles/liferay-tomcat/portal-setup-wizard.properties`. A "Go to My Portal" button is visible at the bottom.

També s'ha d'instal·lar unes llibreries (EJC) que utilitza Plugins SDK.

```
#apt-get install ejc
```

B.11 Provar l'entorn

L'usuari administrador per defecte del portal és *test@liferay.com / test*.

Una forma molt fàcil de testejar que tot l'entorn està funcionant correctament és generar un portlet buit a Eclipse i desplegar-lo

```
ubuntu@ubuntu-VirtualBox:/opt/liferay/liferay-plugins-sdk-6.1.1/portlets$ ls -la
total 32
drwxr-xr-x  2 ubuntu ubuntu 4096 Jul 31  2012 .
drwxr-xr-x 12 ubuntu ubuntu 4096 Jun 15  20:43 ..
-rw-r--r--  1 ubuntu ubuntu  657 Apr  2  2012 build-common-portlet.xml
-rw-r--r--  1 ubuntu ubuntu 8268 Jul 18  2012 build.xml
-rw-r--r--  1 ubuntu ubuntu  854 Apr  2  2012 create.bat
-rwxr-xr-x  1 ubuntu ubuntu  694 Apr  2  2012 create.sh
ubuntu@ubuntu-VirtualBox:/opt/liferay/liferay-plugins-sdk-6.1.1/portlets$ ./create.sh Test "Test"
Buildfile: /opt/liferay/liferay-plugins-sdk-6.1.1/portlets/build.xml

create:
[copy] Copying 9 files to /opt/liferay/liferay-plugins-sdk-6.1.1/portlets/Test-portlet
[mkdir] Created dir: /opt/liferay/liferay-plugins-sdk-6.1.1/portlets/Test-portlet/docroot/WEB-INF/tld
[copy] Copying 7 files to /opt/liferay/liferay-plugins-sdk-6.1.1/portlets/Test-portlet/docroot/WEB-INF/tld

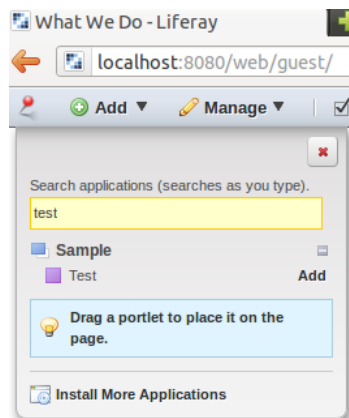
BUILD SUCCESSFUL
Total time: 0 seconds
ubuntu@ubuntu-VirtualBox:/opt/liferay/liferay-plugins-sdk-6.1.1/portlets$
```

En cas de que create.sh no tingui permisos d'execució serà necessari assignar-li permisos:

```
#chmod 755 create.sh
```

```
ubuntu@ubuntu-VirtualBox:/opt/liferay/liferay-plugins-sdk-6.1.1/portlets$ ls
build-common-portlet.xml build.xml create.bat create.sh Test-portlet
ubuntu@ubuntu-VirtualBox:/opt/liferay/liferay-plugins-sdk-6.1.1/portlets$ cd Test-portlet/
ubuntu@ubuntu-VirtualBox:/opt/liferay/liferay-plugins-sdk-6.1.1/portlets/Test-portlet$ ls
build.xml docroot
ubuntu@ubuntu-VirtualBox:/opt/liferay/liferay-plugins-sdk-6.1.1/portlets/Test-portlet$ ant
Buildfile: /opt/liferay/liferay-plugins-sdk-6.1.1/portlets/Test-portlet/build.xml
```

Ara, a Liferay, si s'accedeix com administrador, es podrà veure el portlet desplegat.



Amb la qual cosa ja es pot afirmar que l'entorn és plenament operatiu.

Annex C. Desenvolupament d'un tema a mida

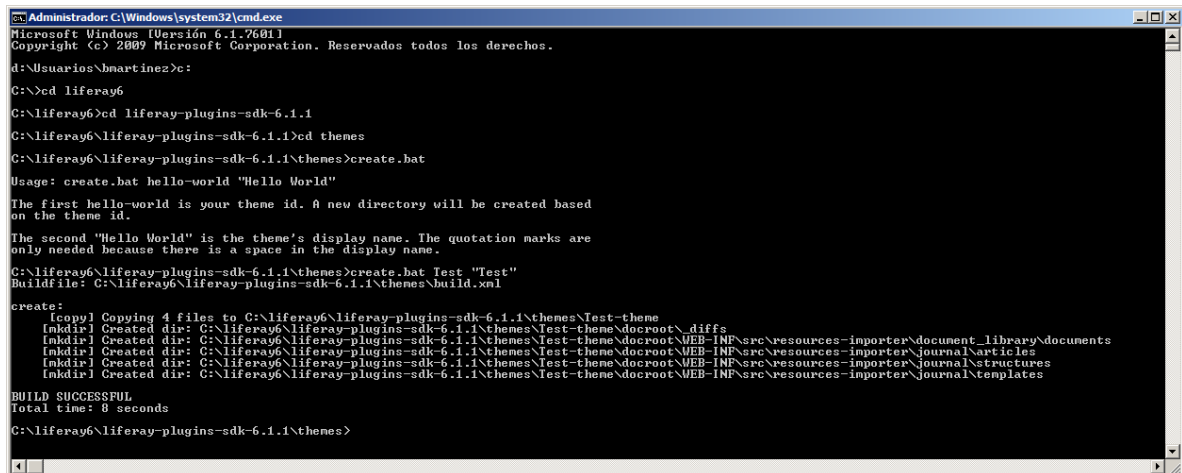
Una vegada s'ha instal·lat correctament tot l'entorn de desenvolupament és el moment de desenvolupar el primer dels nostres mòduls, el tema.

Per a desenvolupar el tema s'hauran de seguir els següents passos:

- Creació del tema base
- Generació d'un projecte Eclipse associat
- Directori `_diffs`
- Estudi de la maqueta
- Creació de les estructures base. Carpeta templates
- Altres : Codi estàtic
- Desplegament del tema

C.1 Creació del tema base

Per a generar el tema base ens hem de situar amb una consola a la carpeta *themes* dins de l'arrel de Plugins SDK. Amb la comanda *create* es llença una acció d'ant que crea un projecte base amb els identificador de tema i el nom que li indiquem.



```
Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versi3n 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

d:\Usuarios\hmartinez>c:
C:\>cd liferay6
C:\liferay6>cd liferay-plugins-sdk-6.1.1
C:\liferay6\liferay-plugins-sdk-6.1.1>cd themes
C:\liferay6\liferay-plugins-sdk-6.1.1\themes>create.bat
Usage: create.bat hello-world "Hello World"
The first hello-world is your theme id. A new directory will be created based
on the theme id.
The second "Hello World" is the theme's display name. The quotation marks are
only needed because there is a space in the display name.
C:\liferay6\liferay-plugins-sdk-6.1.1\themes>create.bat Test "Test"
Buildfile: C:\liferay6\liferay-plugins-sdk-6.1.1\themes\build.xml

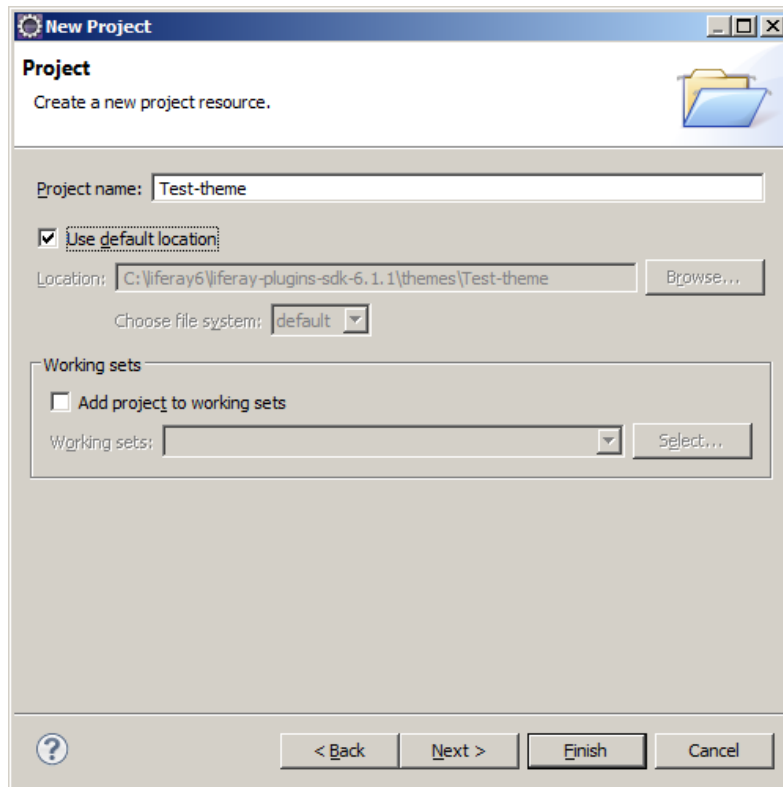
create:
[copy] Copying 4 files to C:\liferay6\liferay-plugins-sdk-6.1.1\themes\Test-theme
[mkdir] Created dir: C:\liferay6\liferay-plugins-sdk-6.1.1\themes\Test-theme\docroot\diffs
[mkdir] Created dir: C:\liferay6\liferay-plugins-sdk-6.1.1\themes\Test-theme\docroot\WEB-INF\src\resources-importer\document_library\documents
[mkdir] Created dir: C:\liferay6\liferay-plugins-sdk-6.1.1\themes\Test-theme\docroot\WEB-INF\src\resources-importer\journal\articles
[mkdir] Created dir: C:\liferay6\liferay-plugins-sdk-6.1.1\themes\Test-theme\docroot\WEB-INF\src\resources-importer\journal\structures
[mkdir] Created dir: C:\liferay6\liferay-plugins-sdk-6.1.1\themes\Test-theme\docroot\WEB-INF\src\resources-importer\journal\templates

BUILD SUCCESSFUL
Total time: 8 seconds
C:\liferay6\liferay-plugins-sdk-6.1.1\themes>
```

C.2 Generació d'un projecte Eclipse associat

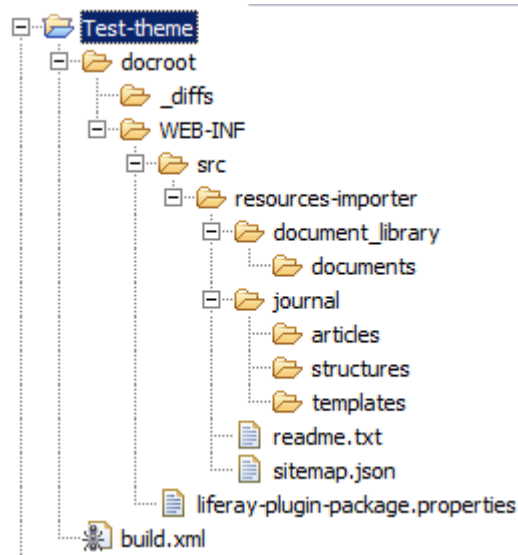
Ara que ja s'ha creat el tema base, s'ha de crear un nou projecte a Eclipse, en el directori que s'ha creat, de forma que podrem treballar amb les fitxers generats via ant.

A Eclipse s'ha de fer *File > New > General > Project* i s'ha de generar el projecte a sobre del directori abans creat.



El tema no té cap tipus de fitxer Java, així que no és necessari un Java Project. Amb un projecte genèric hi ha prou.

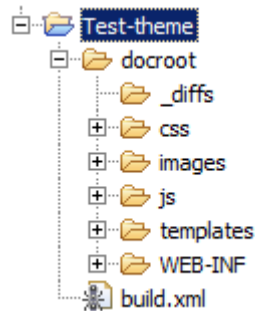
Una vegada creat el projecte, és pot veure el següent arbre al Project Explorer:



Tal i com es pot apreciar, el projecte es troba gairebé buit. Això canviarà de seguida.

S'ha d'executar la tasca per defecte del fitxer build.xml que hi ha a l'arrel del projecte. La tasca és *deploy*.

Una vegada s'ha executat la tasca i es refresca el project explorer es pot veure com s'han generat tot un seguit de fitxers a diversos subdirectoris:



A tasca deploy ha generat tots els fitxers base sobre els que hem de treballar.

La generació d'aquests fitxers és, en part, configurable. Així, si s'edita el fitxer *build.xml* es pot veure com hi ha una propietat *theme.parent* que genera els fitxers base a partir d'un valor *_styled*. Es pot generar un tema molt més senzill amb l'opció *_unstyled*.

```
<?xml version="1.0"?>
<!DOCTYPE project>

<project name="Test-theme" basedir="." default="deploy">
  <import file="../build-common-theme.xml" />

  <property name="theme.parent" value="_styled" />
</project>
```

C.3 Directori *_diffs*

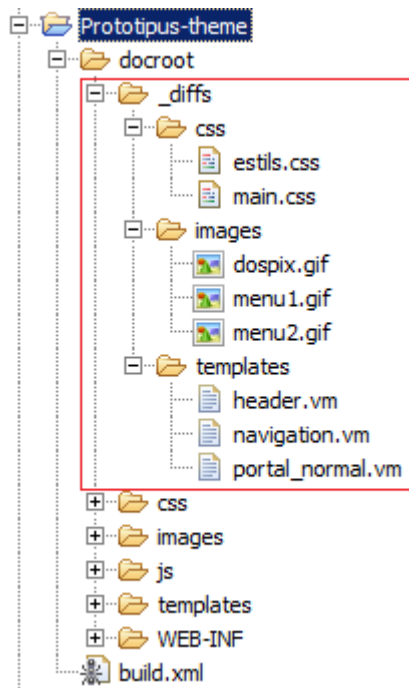
A la última imatge del *project explorer* es pot apreciar que la tasca d'ant ha generat un directori *_diffs* que es troba buit.

Liferay treballa muntant el tema a partir d'aquest directori, sobreescrivint tots els fitxers que hi hagi en aquest directori sobre el seu directori pare *docroot*.

D'aquesta forma, al eliminar un fitxer del directori *_diffs* es tornarà a incloure l'original dins del tema.

Aquesta forma de treballar, tot i que en un primer moment pot semblar estranya és molt útil, ja que davant de qualsevol problema al tema degut a un mal disseny en un fitxer ens permet recuperar fàcilment la versió original.

Si observem el codi font de *Prototipus-theme* veurem com a *_diffs* hi ha molt pocs fitxers. De fet, només es trobaran els fitxers estàtics que s'han afegit (css, imatges i altres) i els fitxers *.vm*, que com es veurà són plantilles de Velocity.



C.4 Estudi de la maqueta HTML

Una vegada s'ha generat la base del tema és important tenir clar el codi font de la maqueta HTML.

Només destacar en aquest punt que sempre és preferible la utilització d'una maqueta que utilitzi objectes de tipus div per a posicionar els seus elements.

La utilització de taules, tot i que pot arribar a ser imprescindible en alguns casos concrets, es desaconsella per norma general.

C.5 Creació de les estructures base. Carpeta *templates*

Ara que ja es coneix a fons la maqueta que es vol muntar és important saber per on hem de començar. El fitxer *docroot/templates/portal_normal.vm* és un bon punt.

Liferay utilitza, com ja s'ha comentat alguna vegada, Velocity per al muntatge de les seves plantilles. Mitjançant aquest llenguatge es pot accedir a diferents variables Java des dels fitxers *.vm*

El fitxer *portal_normal.vm* és el fitxer base (l'esquelet) sobre el que es munta qualsevol pàgina del tema. Així, al codi font del projecte es pot observar com en aquest fitxer importa altres fitxers *.vm*, com per exemple la botonera.

Una part important d'aquest fitxer és la comprovació de l'atribut *selectable*, que permet incloure tot l'apartat *portlet.vm* que és on es mostren els diferents portlets que hi hagi a cada pàgina.

És en aquest portlet.vm on Liferay muntarà el *Layout Template* que nosaltres indiquem, que donarà format a la pàgina.

```
<div id="content">
  #if ($selectable)
    $theme.include($content_include)
  #else
    $portletDisplay.recycle()

    $portletDisplay.setTitle($the_title)

    $theme.wrapPortlet("portlet.vm", $content_include)
  #end
</div>
```

Val la pena estudiar cadascun dels fitxers *.vm* del *docroot*, per a estudiar la millor forma d'aplicar la maqueta al tema base.

IMPORTANT: Recordar que no s'ha de confondre *Layout* amb *Layout Template*. El primer és un component, el segon una plantilla per a una estructura.

C.6 Altres : Codi estàtic

A més dels fitxers de *Velocity* que donen estructura al tema s'hauràn d'incloure els diferents fitxers estàtics que hi hagi a la maqueta al tema.

Per una banda, tenim les imatges. Si el nom d'alguna d'elles és el mateix que el nom d'una imatge del tema base original, aquesta es sobreescrirà. Així doncs, s'ha de vigilar aquest punt.

Per altra banda, és possible que en un tema determinat es vulgui sobreescrir una imatge o un conjunt d'elles. En aquest cas és tan fàcil com incloure les noves imatges al directori *docroot/images*.

També és important incloure correctament els fitxers *css*. En primer lloc serà necessari incloure els fitxers al directori *docroot/css*. En segon lloc s'ha d'indicar l'ordre de càrrega del nou fitxer *css* a *docroot/main.css*, de forma que Liferay sàpiga exactament l'ordre que ha d'aplicar per a carregar els *css* correctament.

En el cas que es vulgui modificar una propietat concreta de les ja existents es pot fer de dues formes diferents: o mitjançant un nou fitxer que sobreescrigui les propietats anteriors o copiant el fitxer *.css* original amb les modificacions que es vulguin fer al directori *docroot/_diffs/images*

C.7 Desplegament del tema

Una vegada fets tots els canvis és el moment de desplegar el tema sobre Liferay.

Tornem a executar la tasca *deploy*, amb el que es genera el tema i es copia al directori *dist* de *Plugins SDK*.

També es copia en el directori *autodeploy* del servidor d'aplicacions, el que fa que es desplegui automàticament sobre *Liferay*.

Altres tasques importants

A més a més de la tasca *deploy*, hi ha algunes tasques del fitxer d'ant que poden ser molt útils alhora de desenvolupar un theme.

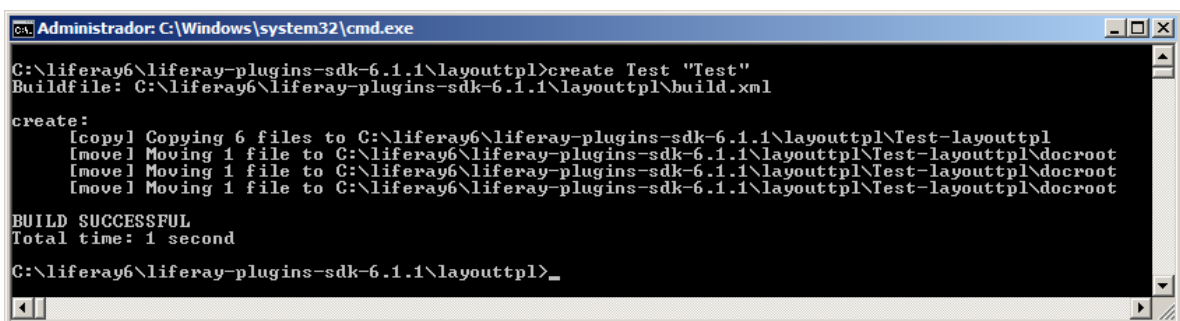
- *clean*: Aquesta tasca permet netejar el *docroot*. Durant el desenvolupament és possible que es copin fitxers al *docroot* que posteriorment no es tornin a utilitzar (al ser esborrats del *_diffs*). Mitjançant aquesta comanda és possible netejar el directori *docroot* per a comptar amb un tema actualitzat.
- *war*: Construeix el fitxer *WAR* (Web ARchive, fitxer que encapsula tot el tema) i el copia a *dist*, sense fer el *deploy*.

Annex D. Desenvolupament d'una plantilla a mida

El desenvolupament d'una plantilla és molt semblant al desenvolupament d'un tema. A continuació es poden veure els passos que s'han de seguir.

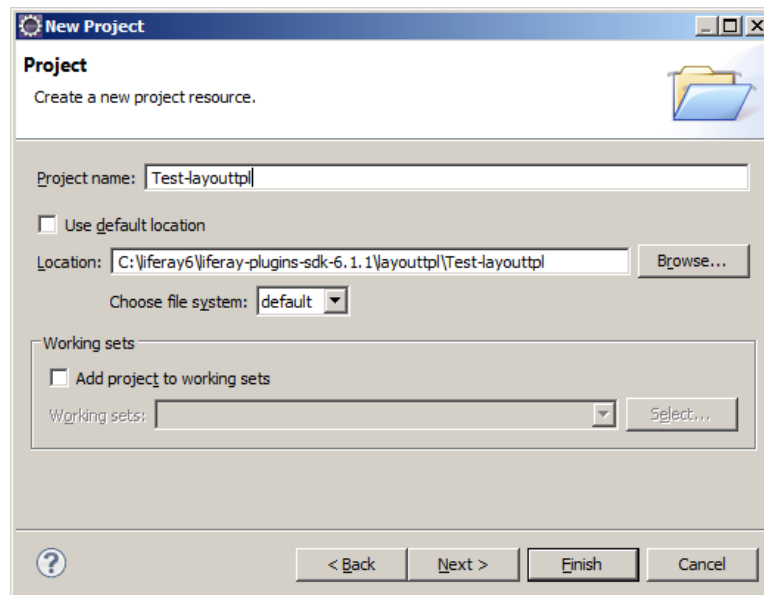
D.1 Passos per a generar una plantilla

El primer pas, al igual que en el punt anterior, és la generació de la base del projecte a partir del fitxer *create* que es troba al directori *layouttpl* del *Plugins SDK*.



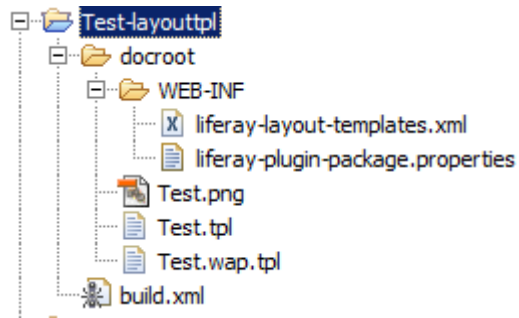
```
Administrator: C:\Windows\system32\cmd.exe
C:\liferay6\liferay-plugins-sdk-6.1.1\layouttpl>create Test "Test"
Buildfile: C:\liferay6\liferay-plugins-sdk-6.1.1\layouttpl\build.xml
create:
[copy] Copying 6 files to C:\liferay6\liferay-plugins-sdk-6.1.1\layouttpl\Test-layouttpl
[move] Moving 1 file to C:\liferay6\liferay-plugins-sdk-6.1.1\layouttpl\Test-layouttpl\docroot
[move] Moving 1 file to C:\liferay6\liferay-plugins-sdk-6.1.1\layouttpl\Test-layouttpl\docroot
[move] Moving 1 file to C:\liferay6\liferay-plugins-sdk-6.1.1\layouttpl\Test-layouttpl\docroot
BUILD SUCCESSFUL
Total time: 1 second
C:\liferay6\liferay-plugins-sdk-6.1.1\layouttpl>_
```

Una vegada creat el layout s'ha de generar un projecte sobre Eclipse, de tipus Project, que apunti directament al directori que s'ha generat amb la comanda *create*.



Al igual que amb els Temes, els layouts no inclouen fitxers font java (*.java*), de forma que amb un rojecte de tipus Project (genéric) és suficient.

Una vegada s'ha generat el projecte, es pot veure el següent al *project explorer*.



A continuació es detalla cadascun dels fitxers amb una descripció sobre el que aporten a la plantilla:

- build.xml: fitxer amb les tasques d'ant. Bàsicament s'utilitzarà *deploy*.
- Test.tpl i Test.wap.tpl: Fitxers amb la plantilla HTML que s'utilitzarà. El primer per a dispositius web. El segon fitxer, és per a dispositius wap, però és possible que a dia d'avui no s'utilitzi (no s'ha trobat informació al respecte, i al prototipus s'ha duplicat el primer .tpl).
- Test.png: imatge que es mostrarà de la plantilla, per a poder reconèixer clarament el format.
- Liferay-layout-templates.xml i liferay-plugin-packages.properties: fitxers amb diferents atributs i meta informació: descripcions de la plantilla, autor nom del fitxer icona, etcètera.

D.2 El fitxer .tpl

Tal i com s'ha pogut veure al punt anterior, el fitxer on es concentra tota la lògica de creació d'una plantilla és el fitxer *.tpl*.

Aquest fitxer genera l'estructura que es vol a partir de *divs* i taules, utilitzant identificadors i noms d'atributs css per a les taules i *divs* que utilitza.

Els dos templates que s'han generat s'han fet prenent com a model els diferents .tpl que Liferay porta per defecte.

De fet, Liferay ja porta uns quants templates per defecte, per la qual cosa en un projecte normalment no farà falta desenvolupar moltes plantilles.

A continuació es pot veure un fragment d'un fitxer TPL (1-70-30-1), on es pot apreciar com s'utilitzen els atributs *class* del css de Liferay per a donar diferents amplades a la plantilla (en aquest cas, es generen dues files, una amb un 70% d'amplada i un altra amb un 30%).

```
<table class="portlet-layout">
<tr>
  <td class="aui-w70 portlet-column portlet-column-first" id="column-2">
    $processor.processColumn("column-2", "portlet-column-content portlet-column-content-first")
  </td>
  <td class="aui-w30 portlet-column portlet-column-last" id="column-3">
    $processor.processColumn("column-3", "portlet-column-content portlet-column-content-last")
  </td>
</tr>
</table>
```

Annex E. Especificació de portlets

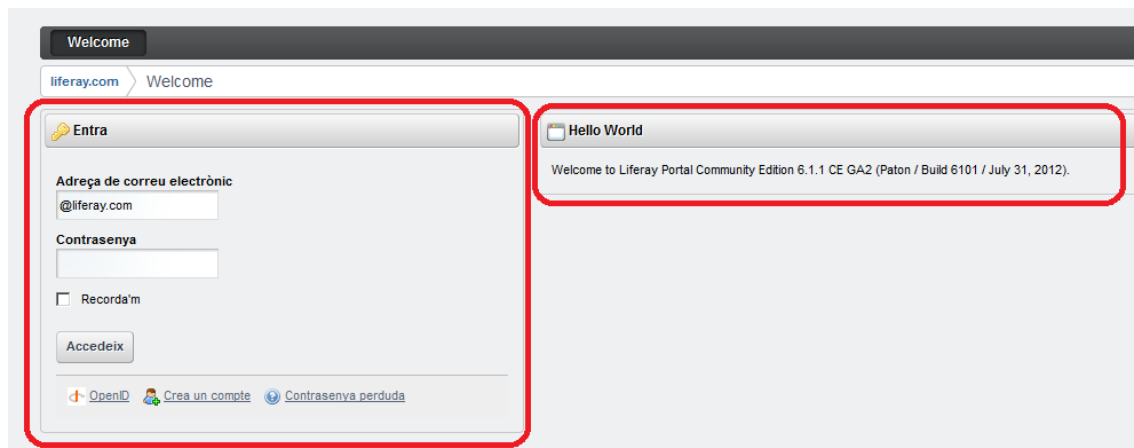
El desenvolupament de portlets és sense cap mena de dubte un dels punts més importants d'aquest prototipus. De fet, una eina com Liferay és, sobretot, un contenidor de portlets,

En aquest apartat es revisaran els conceptes més importants de la JSR 286 i també de la JSR 168, que és la especificació de portlets 1.0. A continuació es detallarà com desenvolupar un portlet a partir de les eines instal·lades al prototipus.

E.1 Que és un portlet?

Tal i com ja s'ha comentat en aquest document, un portlet és un component que sol·licita i genera contingut de forma dinàmica. Aquest contingut s'integra amb altres continguts per a formar un document sencer o pàgina.

Aquesta definició, que és una mica difusa, queda mol més clara si es veu qualsevol pàgina d'un contenidor de portlets com Liferay. Per exemple, en una pàgina per defecte d'una organització ens trobem aquests dos portlets:



A nivell intern, un portlet no és gaire diferent d'una aplicació web estàndard. De fet, els portlets s'empaqueten en un *Web Archive* (WAR). La diferència més significativa és que, a més a més del fitxer descriptor web.xml presenten un altre fitxer anomenat portlet.xml.

En el cas de Liferay, també es presenta un fitxer extra anomenat liferay-portlet.xml. Més endavant es veurà el contingut d'aquests dos fitxers.

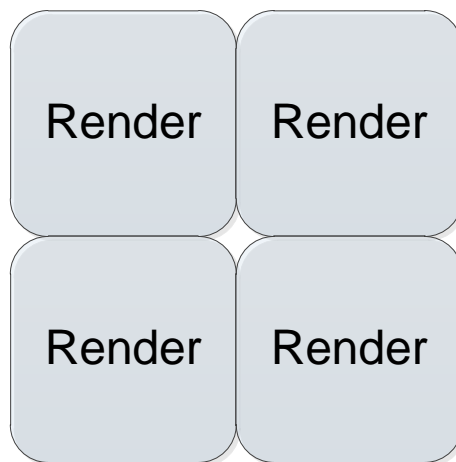
E.2 Fases

Així doncs, es pot veure que treballar amb portlets és, mes o menys, com desenvolupar una aplicació web 'tradicional' sobre Java, només que ara no es treballa amb Servlets. Més o menys.

Mentre que els Servlets tenen un tipus de sol·licitud o *request*, els portlets (segons l'especificació JSR 168) tenen dues:

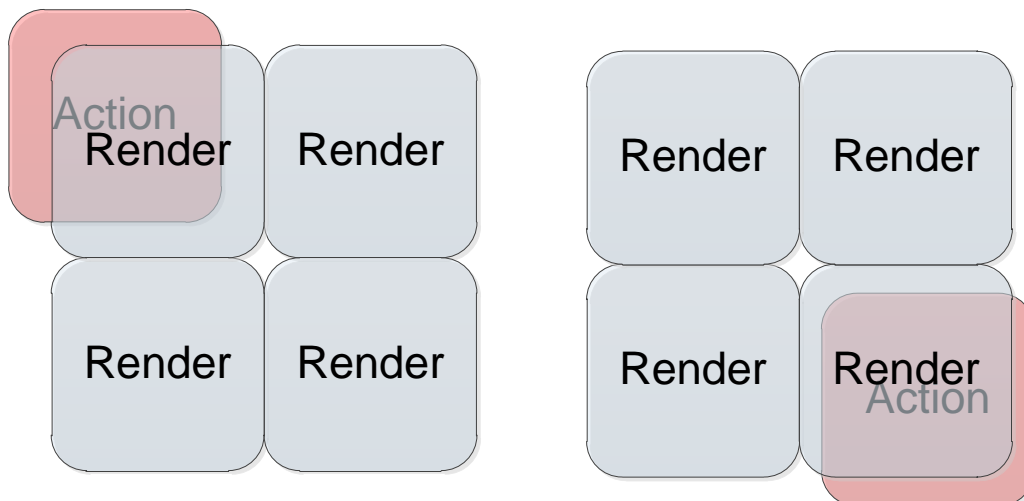
- Fase *Render*: Cridada cada vegada que el portlet necessita repintar-se a la pàgina.
- Fase d'acció (Action Phase): Cridada com a resultat d'una ActionURL. El portlet realitza algun tipus de processament, i a continuació realitza la fase de renderitzat.

Així doncs, imaginem que tenim quatre portlets en una pàgina i carreguem aquest pàgina en un navegador web. Quina fase es llançarà per a cadascun d'ells? Doncs la fase render.



Ara, si un d'ells realitza una ActionURL (és a dir, una específica com per exemple una crida per a donar d'alta unes dades d'un formulari), aquest portlet llançarà una Fase Action... i a continuació TOTS llançaran una fase render.

És important veure que només es llança una Action Fase en tot aquest procés. Així doncs, no és el mateix el primer cas de la següent figura i el segon cas. Tot depèn de qui realitza l'acció.



La especificació JSR 286 afegeix dues fases més, anomenades *Event Phase* (es crida al llençar-se un event) i *Resource Serving Phase*, cridada per el mètode *serveResource*. No entrarem a detallar aquestes fases ja que no s'han treballat.

E.3 Cicle de vida d'un portlet

Una vegada vistes les fases que pot tenir un portlet (si més no, les més importants), és important conèixer també el seu cicle de vida.

El cicle de vida d'un portlet té quatre fases:

- *init()*: mètode per a inicialitzar el portlet. Es crida, com no, una sola vegada amb la seva creació.
- *processAction()*: mètode cridat cada vegada que es realitza una acció (o ActionURL) sobre el portlet.
- *render()*: mètode per a realitzar el renderitzar (o pintat) del portlet.
- *destroy()*: mètode que s'executa una sola vegada, al eliminar el portlet. Bàsicament serveix per a alliberar possibles recursos que utilitzi el portlet.

Com es pot veure, aquest cicle de vida és lògic, si es té en compte el punt anterior sobre fases. Abans de renderitzar, acció.

E.4 Modes d'un portlet

Un altre dels punts importants dins dels portlets són els modes.

Cada portlet té diferents modes, i en cada moment un portlet es pot trobar només en un d'ells. Aquest mode actual indica el que està fent el portlet en aquell moment.

Existeixen tres modes per defecte, que són els que normalment s'utilitzen. Aquests modes son Vista, Edició i Ajuda (en anglès, *View, Edit i Help* respectivament).

El Mode Vista (View Mode) és el mode principal, on es mostra la pantalla que realitza les tasques públiques del portlet.

El Mode Edició (Edit Mode) és el que es veu al accedir a les preferències del portlet. És un mode per a configurar el portlet.

El mode ajuda dona informació sobre el portlet. Per exemple, com configurar-lo i el que fa exactament. En els nostres portlets no s'ha implementat.

Un exemple clar dels dos primers modes es pot trobar al portlet de Minidetall de Producte. Aquest portlet, que serveix per a mostrar un producte en un lateral com si fos una promoció, té un Mode Vista on es veu el Producte seleccionat i un Mode Edició on es pot escollir el producte a mostrar.

Mini Detall Producte

✎ - + ✕



Moble TV NEO

Moble Neo. Ideal per a mejadors petits.

Mini Detall Producte

[Ves a la pàgina principal](#)

URL Imatges:

Pàgina de detall:

Producte a mostrar:

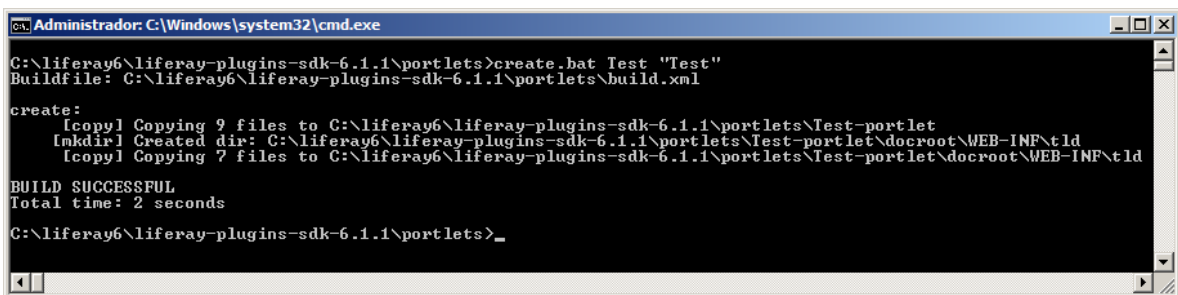
Annex F. Desenvolupament de portlets

Ara que es coneix la especificació de portlets (com a mínim els seus punts més importants), és el moment d'entrar a desenvolupar aquest tipus de components sobre Plugins SDK.

F.1 Creació del projecte e integració a Eclipse

El primer pas a l'hora de desenvolupar un portlet es generar un projecte, tal i com ja s'ha fet amb els temes i els *layouts*.

Per això es necessita una consola situada al subdirectori *portlets* de Plugins SDK i la comanda *create*.

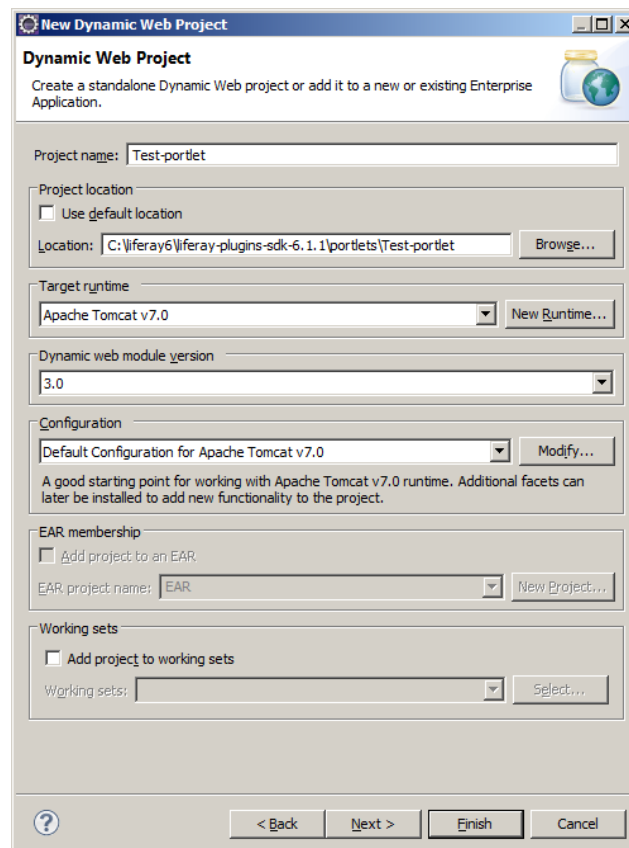


```
Administrator: C:\Windows\system32\cmd.exe
C:\liferay6\liferay-plugins-sdk-6.1.1\portlets>create.bat Test "Test"
Buildfile: C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\build.xml

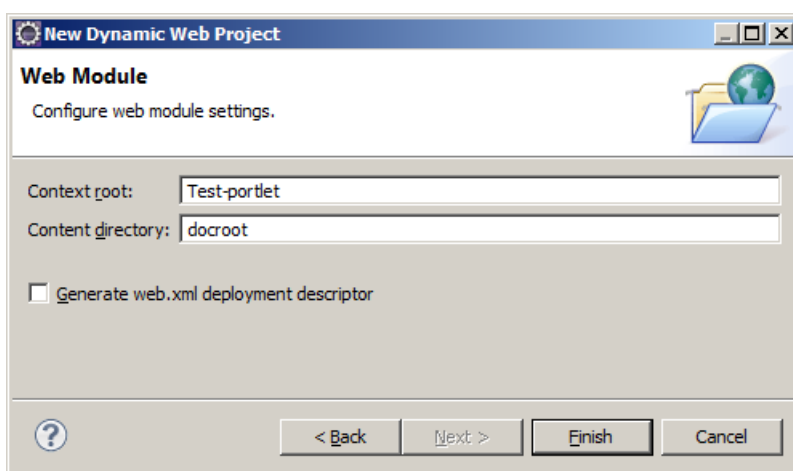
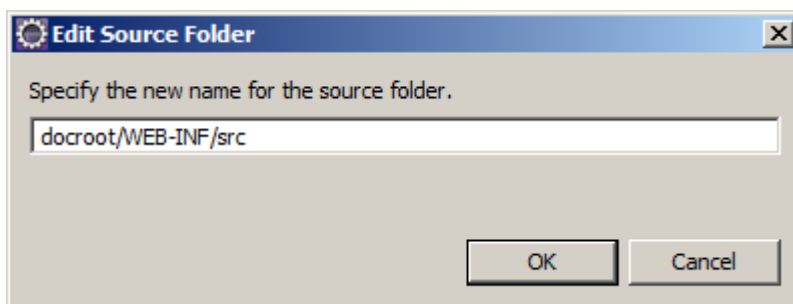
create:
[copy] Copying 9 files to C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\Test-portlet
[mkdir] Created dir: C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\Test-portlet\docroot\WEB-INF\td
[copy] Copying 7 files to C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\Test-portlet\docroot\WEB-INF\td

BUILD SUCCESSFUL
Total time: 2 seconds
C:\liferay6\liferay-plugins-sdk-6.1.1\portlets>
```

Ara que s'ha generat la base del portlet es pot generar el projecte a Eclipse. A diferència que amb els projectes que s'han creat fins ara per a temes i plantilles, és necessari que aquest projecte sigui de tipus Dnamic Web Project.



Com es pot veure a la imatge de la pàgina anterior, el directori amb els fonts (.java) s'haurà de situar a *docroot/WEB-INF/src*, i el directori de contingut a *docroot*.

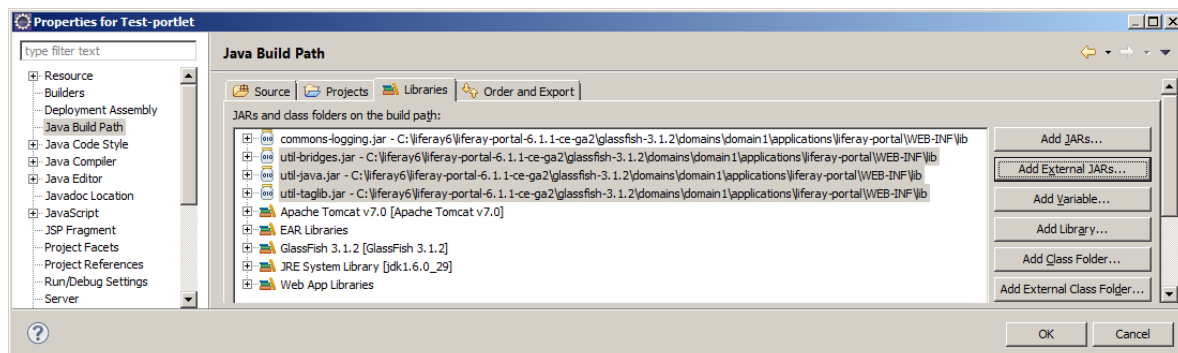


Ara que el projecte ha estat creat, és necessari configurar les seves dependències.

Serà necessari incloure el Server Runtime de Glassfish. També serà necessari incloure una sèrie de llibreries externes que es troben a *%GLASSFISH%\domains\domain1\applications\liferay-portal\WEB-INF\lib*: commons-logging.jar, util-bridges.jar, util.java.jar i util-taglib.jar.

IMPORTANT: *%GLASSFISH%* és la arrel del servidor d'aplicacions.

El projecte hauria de quedar com la imatge següent.



Ara si, el projecte ja es troba ben configurat.

F.2 Els fitxers portlet.xml i liferay-portlet.xml

Dins del directori WEB-INF es troben els fitxers descriptors d'una aplicació. Entre ells, es poden trobar portlet.xml i liferay-portlet.xml. És molt interessant revisar les possibles opcions que es poden utilitzar en aquests dos fitxers. Només es dealaràn aquí els punt més importants:

Fitxer liferay.xml

Fitxer descriptor principal, indica la classe java principal del nostre portlet (tag portlet-class). Es poden també incloure paràmetres inicials per a un portlet (init-param) i guardar preferències (tag preferences).

Fitxer liferay-portlet.xml

Serveix bàsicament per a donar informació extra a Liferay que no es contempla a l'especificació. Per exemple, en el cas del prototipus s'utilitza per a indicar si un portlet es vol o no mostrar al Tauler de Control.

Liferay integra de forma molt elegant aquest problema, fent possible que un portlet es carregui al tauler de control en un punt determinat gràcies a un parell de propietats al fitxer liferay-portlet.xml

```
<?xml version="1.0"?>
<!DOCTYPE liferay-portlet-app PUBLIC "-//Liferay//DTD Portlet Application 6.1.0//EN" "
<liferay-portlet-app>
  <portlet>
    <portlet-name>Gestio Producte</portlet-name>
    <control-panel-entry-category>content</control-panel-entry-category>
    <control-panel-entry-weight>1.5</control-panel-entry-weight>
  </portlet>
  <portlet>
    <portlet-name>Llistat Producte</portlet-name>
    <instanceable>true</instanceable>
  </portlet>
```

F.3 Implementació. Mètodes a implementar. javax.portlet.GenericPortlet.

Per a desenvolupar els portlets del prototipus s'ha extés la classe javax.portlet.GenericPortlet

Els portlets hauràn d'implementar diferents mètodes, que s'encarreguen de les diferents accions que hem vist a l'apartat 9.

Així per exemple, s'han d'implementar els mètodes doView(RenderRequest, RenderResponse), doEdit(RenderRequest, RenderResponse),

doHelp(RenderRequest, RenderResponse) per a cadascun dels modes que es vagina utilitzar.

També s'haurà d'implementar el mètode processAction(ActionRequest, ActionResponse) per a del diferents ActionURL del portlet. De fet, enlloc d'implementar directament el mètode, el que es farà és utilitzar anotacions per a generar un conjunt de mètodes que representaran les diferents ActionURL que es poden donar al portlet.

El que s'està fent mitjançant anotacions és fer de *controlador* de les diferents accions que es poden donar sobre el portlet.

Així, per exemple, al portlet de Gestió de Productes, existeixen diferents mètodes que s'encarreguen de diferents accions, com *newProducte* i *addProducte*

```
@ProcessAction(name = "newProducte")
public void newProducte(ActionRequest request, ActionResponse response) {
}

@ProcessAction(name = "addProducte")
public void addProducte(ActionRequest request, ActionResponse response)
throws Exception {
}
```

F.4 Portlets MVC

Per a la creació de portlets senzills com els que s'han realitzat no és necessària la utilització de Bastiments Model-Vista-Controlador.

Tot i això, seria molt interessant ampliar el prototipus amb nous portlets que fessin ús de Struts, Spring o JSF per exemple.

F.5 Capa de Servei

Els portlets necessiten realitzar diferents accions sobre el sistema. S'han de guardar i consultar dades, realitzar transaccions (no en aquest prototipus), eliminar entrades de la base de dades, etcètera.

Amb l'ajuda del fitxer service.xml del que ja es va parlar al punt 4.8 s'utilitzarà la eina Service Builder de Liferay per a generar tota la capa de serveis de l'aplicació.

En aquest fitxer es poden definir les diferents entitats que es volen gestionar, indicant quin tipus de serveis es volen generar (locals o remots), per quins camp es vosl consultar (finders), etcètera.

Així per exemple, imaginem que es vol generar una entitat Test amb un identificador i un nom. Si es vol poder cercar per nom també, s'haurà de generar un tag finder. El codi final del fitxer xml seria el següent:

```

<?xml version="1.0"?>
<!DOCTYPE service-builder PUBLIC "-//Liferay//DTD Service Builder 6.2.0//EN"
"http://www.liferay.com/dtd/liferay-service-builder_6_2_0.dtd">

<service-builder package-path="edu.uoc.test">
  <namespace>PROTOTIPUS</namespace>

  <!-- Entitat Test -->
  <entity name="Test" uuid="true" local-service="true" remote-
service="false">

    <!-- Clau primària, IDTest -->
    <column name="idTest" type="Long" primary="true" />

    <column name="nom" type="String" />

    <!-- Orde -->
    <order by="asc">
      <order-column name="nom" />
    </order>

    <!-- Mètodes Finder (per a la cerca) -->
    <finder name="Nom" return-type="Collection">
      <finder-column name="nom" />
    </finder>
  </entity>

</service-builder>

```

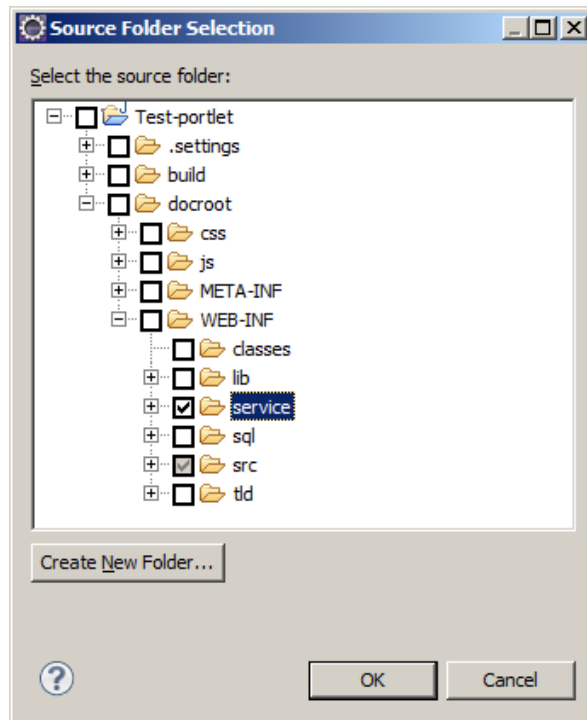
Amb la tasca d'ant *build-service* es generaran tot un seguit de classes i mètodes amb les funcionalitats que s'han definit. A la següent imatge es pot veure un fragment de l'execució de la tasca, on es generen diverses classes.

```

[echo] Building Test
[echo] Writing C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\Test-portlet\docroot\WEB-INF\src\edu\uoc\test\service\persistence\TestPersistenceImpl.java
[echo] Writing C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\Test-portlet\docroot\WEB-INF\service\edu\uoc\test\service\persistence\TestPersistence.java
[echo] Writing C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\Test-portlet\docroot\WEB-INF\service\edu\uoc\test\service\persistence\TestUtil.java
[echo] Writing C:\liferay6\liferay-plugins-sdk-6.1.1\portlets\Test-portlet\docroot\WEB-INF\src\edu\uoc\test\model\impl\TestModelImpl.java

```

Aquestes noves classes font serà necessari incloure-les al projecte. Així, s'hauràn de tornar a editar les propietats del projecte i afegir el directori *docroot\WEB-INF\service* com a directori amb codi font.



En cas que sigui necessari generar nous mètodes a la capa de servei, , es podran definir els mètodes que es necessitin, i amb la mateixa tasca es generaran aquests mètodes a les classes interfacie pertinents.

Així, al prototipus s'han generat alguns mètodes de cerca, a la classe `edu.uoc.prototipus.service.impl.ProducteLocalServiceImpl`.

F.6 Portlets de tercers

La versió 6.1 de Liferay incorpora una nova eina, anomenada Liferay Marketplace que permet descarregar e instal·lar fàcilment portlets de tercers.

És necessari tenir un usuari registrat a la web de Liferay.com per a utilitzar Marketplace. De fet, el compte d'usuari de Liferay.com mantindrà un registre de tots els portlets que s'hagin descarregat i utilitzat.

Mitjançant aquest mètode es va trobar un portlet forà interessant, anomenat Carousel Portlet. Aquest portlet mostra un rotador d'imatges (promocions) marcades amb un *tag* determinat al sistema.

De fet, la configuració és una mica més complicada, i obliga a la creació d'un nou tipus de document a l'apartat *Documents and Media*.

El realment interessant d'aquest punt és només deixar constància de l'existència del Marketplace, que pot ajudar no només a una empresa a estalviar-se moltes hores de desenvolupament, sino també a obtenir ingressos a partir d'algun portlet que tingui èxit al Marketplace (al Marketplace hi ha portlets gratuïts, i de pagament).