

# **Desenvolupament d'una aplicació online amb J2EE**

Tienda Ebooks

**Carlos Alonso Alvarez**  
ETIS

**Vicenç Font Sagrista**

14 de Juny de 2011

A Jose,  
gracies per la teva filosofia mostrada al llarg de totes les hores dedicades.  
A María,  
gracies per fer-ho possible.

## Resum del treball

Aquest treball final de carrera està integrat a l'àrea de J2EE. Aquest treball consisteix en l'anàlisi, disseny i implementació d'una aplicació sota el paradigma de Java 2 Enterprise Edition. L'aplicació que es dissenya es una botiga de llibres electrònics pensada per a impulsar aquest sector. És una aplicació oberta a tothom, en el sentit que serà una aplicació pública d'Internet.

En l'anàlisi i el disseny de l'aplicació es veurà el procés que requereix qualsevol projecte informàtic amb els diagrames necessaris resultants d'una fase de recollida de requisits, anàlisi d'aquests i disseny, que guiarà més tard el procés d'implementació posterior.

Tot el projecte està basat en el llenguatge de programació Java amb l'arquitectura J2EE. Java aporta que l'aplicació sigui portable a qualsevol màquina i l'arquitectura J2EE en proporciona les eines necessàries per a construir l'aplicació distribuïda.

En el disseny d'aquest projecte també s'han considerat diversos frameworks i patrons de disseny per a facilitar la tasca d'implementació de l'aplicació. Entre la gran varietat que hi ha al mercat, s'ha decidit fer ús dels patrons DAO i MVC, i com a frameworks s'ha optat per fer ús d'Hibernate i Struts 2.

Per a executar l'aplicació s'ha optat per fer servir programari lliure, ja que és pot trobar fàcilment a Internet i és molt fiable.

# Índex

<b>1. Introducció</b>	6
<b>1.1. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC</b>	6
<b>1.2. Objectius del TFC</b>	7
<b>1.3. Enfocament i mètode seguit</b>	7
<b>1.4. Planificació del projecte</b>	8
<b>1.5. Productes obtinguts</b>	8
<b>1.6. Descripció dels altres capítols de la memòria</b>	8
<b>2. Recollida i documentació de requisits</b>	9
<b>2.1. El context de programari</b>	9
2.1.1. Subsistema d'usuaris	9
2.2.2. Subsistema administrador	10
2.2.3. Subsistema venda	11
2.2.4. Subsistema social	11
2.2.5. Subsistema de presentació	11
<b>2.2. Casos d'ús</b>	12
2.2.1. Guions	12
2.2.2. Actors	12
2.2.3. Casos d'ús	12
2.2.4. Documentació dels casos d'ús per subsistema	14
<b>3. Anàlisi</b>	22
<b>3.1. Paquets d'anàlisi</b>	22
<b>3.2. Especificació de les classes d'anàlisi</b>	22
3.2.1. Identificació classes d'entitat	22
3.2.2. Diagrama de classes	23
<b>4. Disseny</b>	24
<b>4.1. Decisions de disseny. Patrons i frameworks</b>	24
4.1.1. J2EE	24
4.1.2. Struts2	26
4.1.3. Hibernate	27
4.1.4. MVC	27
4.1.5. DAO	28
<b>4.2. Arquitectura de l'aplicació</b>	29
<b>4.3. Disseny dels casos d'ús</b>	31
4.3.1. Diagrames de seqüència	31
4.3.2. Diagrames d'activitat	39
<b>4.4. Disseny de la persistència</b>	40
<b>4.5. Disseny de la interfície d'usuari</b>	40
4.5.1. Elements comuns a totes les pàgines	40
4.5.2. Pàgines	41
<b>5. Implementació</b>	47
<b>5.1. Requeriments de programari</b>	47
<b>5.2. Eines de desenvolupament</b>	47
<b>5.3. Script de creació de la base de dades</b>	47
<b>6. Conclusions</b>	51
<b>7. Glossari</b>	52
<b>8. Bibliografia</b>	54
<b>Annex 1</b>	56
<b>Manual d'instal·lació</b>	56

## Índex de figures

Il·lustració 1. Planificació	8
Il·lustració 2. Diagrama de casos d'ús	13
Il·lustració 3. Subsistema d'usuaris	14
Il·lustració 4. Subsistema de venda	16
Il·lustració 5. Subsistema social	18
Il·lustració 6. Subsistema administració	19
Il·lustració 7. Diagrama de paquets d'anàlisi	22
Il·lustració 8. Diagrama de classes	23
Il·lustració 9. Exemple aplicació J2EE	24
Il·lustració 10. Capes aplicació J2EE	25
Il·lustració 11. Contenedors J2EE	26
Il·lustració 12. Funcionament Struts	27
Il·lustració 13. Patró MVC	28
Il·lustració 14. Patró DAO	29
Il·lustració 15. Arquitectura de l'aplicació	30
Il·lustració 16. Diagrama de seqüència de "Registrarse"	31
Il·lustració 17. Diagrama de seqüència de "Identificarse"	32
Il·lustració 18. Diagrama de seqüència de "Ver Perfil" i "Mis Libros"	33
Il·lustració 19. Diagrama de seqüència de "Publicar Ebook"	34
Il·lustració 20. Diagrama de seqüència de "Buscar Libro"	34
Il·lustració 21. Diagrama de seqüència de "Ver Catalogo"	35
Il·lustració 22. Diagrama de seqüència de "Ver Libro"	35
Il·lustració 23. Diagrama de seqüència de "Leer Ebook"	36
Il·lustració 24. Diagrama de seqüència de "Descargar Ebook"	36
Il·lustració 25. Diagrama de seqüència de "Comentar/Puntuar"	37
Il·lustració 26. Diagrama de seqüència de "Alta Ebook"	38
Il·lustració 27. Diagrama d'activitat del cas d'ús "Publicar ebook"	39
Il·lustració 28. Diagrama de persistència	40
Il·lustració 29. Barra principal	41
Il·lustració 30. Pantalla d'inici	41
Il·lustració 31. Pantalla d'exploració de catàleg	42
Il·lustració 32. Pantalla registre	43
Il·lustració 33. Pantalla publicar anònim	43
Il·lustració 34. Pantalla descarregues i lectures	44
Il·lustració 35. Pantalla de resultats de cerca	45
Il·lustració 36. Pantalla veure llibre	45

# 1. Introducció

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

L'objectiu d'aquest Treball Final de Carrera és la construcció d'una plataforma de lectura i venda d'ebooks. La idea fonamental darrera aquest projecte es la incentivació de la tecnologia dels llibres electrònics, ja que es una tecnologia que no s'està exprimint totes les seves possibilitats.

En aquesta societat, on cada vegada es disposa de més gadgets electrònics, cada cop amb més funcionalitats, els ebooks son el futur de la literatura. Aquesta tecnologia permet portar els llibres a qualsevol lloc còmodament, sense haver de portar un incòmode llibre de paper a sobre. Juntament, els ebooks i els gadgets, ofereixen la possibilitat de llegir llibres al mòbil, a una tableta, a un lector de llibres electrònics, etc.

Amb aquest projecte es vol apropar mes aquesta tecnologia a les persones i per això es construirà una plataforma on podrem llegir i comprar llibres electrònics.

Aquest projecte parteix de la necessitat de dissenyar una aplicació, en aquest cas, una aplicació de comerç electrònic, que sigui distribuïda, escalable i robusta. Aquesta aplicació estarà construïda sota el paradigma de l'orientació a objectes i amb arquitectura distribuïda.

L'objectiu d'aquest treball es posar en pràctica tots els coneixements adquirits al llarg de la carrera realitzant una solució informàtica al problema plantejat, analitzant els requeriments d'una necessitat concreta i analitzar-los i dissenyar una aplicació que els solucioni, mitjançant un projecte informàtic amb un pla de treball ben estructurat.

El punt de partida d'aquest treball ha sigut els coneixements adquirits al llarg de la carrera, tant de programació com de realització de projectes, i posar-los en practica per una solució real. El projecte que es duu a terme necessita de tecnologies que no hem estudiat a la carrera.

En aquest punt, ha calgut un extens estudi sobre programació distribuïda, i en aquest cas concret, sobre l'arquitectura J2EE, sota la qual està construïda l'aplicació. A més s'han estudiat els patrons de disseny emprats en aquest projecte i dels frameworks que s'han fet servir per a facilitar la tasca de programació.

L'aportació d'aquest treball ha esta la posada en practica dels coneixements teòrics adquirits, com per exemple realitzar tot el procés d'anàlisi i disseny d'una aplicació real. També cal esmentar l'aportació de coneixements sobre programació distribuïda, concretament amb arquitectura J2EE. Els coneixements aportats aquí es podran aplicar després amb altres arquitectures i llenguatges ja que les bases d'aquest tipus de programació s'han assolit amb aquest treball.

Cal esmentar també la cerca i estudi dels patrons de disseny, vistos molt breument a Enginyeria de Programari, així també com els coneixements aportats pels frameworks utilitzats. Per últim, també cal esmentar els coneixements que han aportat la programació de les pagines web, amb HTML i CSS.

## **1.2. Objectius del TFC**

L'objectiu principal d'aquest treball és donar una solució a una botiga de llibres electrònics online. Els objectius marcats eren assolir els coneixements de programació distribuïda per a donar solució a la necessitat anterior. D'aquesta manera, ha calgut estudiar l'arquitectura J2EE.

A més de J2EE i Java per a la programació d'aquesta aplicació, s'han marcat com a objectiu fer servir per a aquest fi els frameworks Struts2 i Hibernate i els patrons de disseny DAO i MVC. Això comporta realitzar un extensa cerca d'informació sobre aquestes eines per a després fer un anàlisi i un estudi en profunditat de cadascuna per a poder-les fer servir en aquesta aplicació.

Assolir els coneixements necessaris per a la capa de presentació de l'aplicació, amb la tecnologia JSP, HTML i CSS per a fer una presentació de l'aplicació atractiva per a l'usuari.

## **1.3. Enfocament i mètode seguit**

Per a la correcta realització d'aquest treball s'ha optat per realitzar l'estudi convenient i assolir els coneixements necessaris sobre totes les noves tecnologies emprades en l'aplicació sobre les quals es tenia desconeixement previ.

Així, l'enfocament ha estat primer adquirir els coneixements necessaris per a construir l'aplicació i després realitzar el projecte com qualsevol projecte informàtic.

El mètode seguit ha sigut estudiar a fons les necessitats i els requeriments de l'aplicació, fent posteriorment l'anàlisi i el disseny adequats a les necessitats de l'aplicació, sota el paradigma de l'orientació a objectes i programació distribuïda.

## 1.4. Planificació del projecte

Al principi d'aquest projecte es va dissenyar un diagrama de temporització de les tasques que calien per a la realització del treball. Aquest diagrama s'ha respectat al llarg del projecte. La planificació proposada es pot veure en el diagrama de Gantt que es troba a continuació:



II-lustració 1. Planificació

## 1.5. Productes obtinguts

El producte principal obtingut en aquest treball és una aplicació distribuïda d'una botiga de llibres electrònics.

Els diferents productes que hem obtingut es detallen a continuació.

- L'arxiu de distribució de l'aplicació amb extensió war que caldrà instal·lar al servidor adequat i que conté tots els arxius class de les classes de l'aplicació juntament amb les llibreries necessàries per a que l'aplicació funcioni correctament.
- El codi font de totes les classes emprades per a l'aplicació.
- La documentació detallada de totes les classes de l'aplicació en format javadoc.
- Els scripts necessaris per a crear la base de dades que fa servir l'aplicació i per a realitzar els inserts d'exemple necessaris per a veure el funcionament de l'aplicació.

## 1.6. Descripció dels altres capítols de la memòria

En la resta de capítols d'aquesta memòria es poden trobar les diferents etapes del projecte. Així, trobarem la recollida de requisits de l'aplicació, l'anàlisi d'aquests i per últim, el disseny de l'aplicació.



## 2. Recollida i documentació de requisits

### 2.1. El context de programari

La botiga tindrà un catàleg de llibres que oferirà als usuaris registrats, encara que qualsevol invitat podrà consultar el catàleg. Quan l'usuari vulgui un llibre, el seleccionarà i un cop registrat i identificat al sistema, podrà veure el llibre en qüestió online. Això vol dir que tots els llibres s'oferiran gratuïtament per llegir online des de la pagina web, però no es podran guardar. Si un usuari vol descarregar el llibre per llegir-lo amb un dispositiu o llegir-lo offline, llavors se li oferirà la possibilitat de descarregar el llibre per un mòdic preu.

Cada usuari pot puntuar i comentar els llibres que vulgui. Quan un usuari seleccioni un llibre, podrà veure les puntuacions i els comentaris que hagin fet altres usuaris.

Els usuaris del servei també podran publicar els seus propis llibres, sempre subjecte a l'aprovació d'un administrador del sistema. Això serà així per a controlar una mica els continguts que es publicaran al servei. L'usuari podrà escollir el preu al que es vol posar el seu llibre en descarrega. Els llibres d'usuaris anònims es publicaran com a tal i es farà constar que es un llibre d'un usuari del nostre servei.

Els administradors del servei seran els encarregats de gestionar el catàleg dels llibres que s'oferen, incloent nous o eliminant els que ja hi son. També gestionaran les ofertes que es facin i acceptaran o no els llibres del usuaris anònims per a la seva publicació.

La interfície d'usuari serà amigable i de fàcil utilització. La navegació entre pàgines serà intuïtiva.

El sistema estarà dividit en cinc subsistemes que seran els següents: sistema d'usuaris, sistema administrador, sistema venta, sistema social i sistema de presentació.

En els apartats següents podem trobar les funcionalitats del sistema segons els subsistemes corresponents.

#### 2.1.1. Subsistema d'usuaris

Aquest subsistema s'encarregarà de totes les tasques relacionades amb els usuaris de l'aplicació. Les seves funcions seran les següents:

- Alta usuaris (registrar-se)

Aquesta opció serveix per a que nous usuaris es registrin al sistema.

- Ver perfil

L'usuari podrà veure les seves dades personals en qualsevol moment amb aquesta funcionalitat.

- Identificar-se

Permetrà als usuaris identificar-se al sistema per a gaudir dels privilegis dels usuaris registrats.

- Publicar llibres

Aquesta funció donarà el suport necessari per a que un usuari registrat pugui publicar a l'aplicació els seus propis llibres.

- Contactar

Amb aquesta funció, un usuari o un invitat pot posar-se en contacte amb l'administrador del sistema per a qualsevol dubte, suggeriment, etc. Només caldrà emplenar un formulari.

- Veure els meus llibres

Aquesta funció estarà disponible quan un usuari hagi fet alguna descarrega o llegit algun llibre online. Llavors, podrà veure totes les descarregues o lectures que hagi realitzat.

### **2.2.2. Subsistema administrador**

El sistema administrador portarà a terme totes les tasques de gestió de l'aplicació per part dels administradors. Les funcions de que consta són les següents:

- Alta llibres

Amb aquesta funció els administradors afegiran nous ebooks al catàleg de l'aplicació.

- Baixa llibres

Aquesta funció serà l'encarregada d'eliminar ebooks del sistema.

- Realitzar ofertes

Els administradors podran publicar eventuais ofertes amb aquesta funció.

- Acceptar ebook anònim

Aquesta funcionalitat permetrà publicar al sistema un nou ebook d'un usuari anònim després d'haver comprovat que el seu contingut respecta les normes.

- Veure contactes

L'administrador podrà veure els contactes que els usuaris hagin realitzat.

- Veure ebooks anònims pendents d'acceptar

Se li presentarà a l'administrador una llista amb tots els ebooks anònims pendents d'acceptar que hi hagi al sistema.

### **2.2.3. Subsistema venta**

El sistema de venta serà l'encarregat de presentar el catàleg de llibres als usuaris i permetrà la visualització online del llibre o la seva descarrega i la gestió de la compra. Les seves funcions seran:

- Cercar llibre

Aquesta funció permetrà fer una cerca a la base de dades del sistema segons algun criteri establert.

- Explorar catàleg

Permetrà la visualització completa del catàleg dels llibres que té el sistema.

- Visualitzar ebook

Un cop seleccionat cert llibre, aquesta funció permetrà llegir-lo online.

- Descarregar i comprar ebook

Aquesta funció s'encarregarà de fer el càrrec de l'import a l'usuari i de la gestió de la seva compra i la descarrega de l' posterior.

- Llegir ebook online

Amb aquesta funció els usuaris poden llegir els ebooks online des de la pagina web de forma totalment gratuïta.

### **2.2.4. Subsistema social**

Aquest sistema contindrà la part més social de l'aplicació que seran la gestió dels comentaris i puntuacions que els usuaris facin sobre els ebooks. Les seves tasques seran:

- Puntuar ebook

Aquesta funció permetrà a l'usuari registrat fer puntuacions sobre els llibres del sistema.

- Comentar ebook

Aquesta funció permetrà a l'usuari registrat fer comentaris sobre els llibres del sistema.

### **2.2.5. Subsistema de presentació**

Aquest sistema serà l'encarregat de gestionar tota la capa de presentació del sistema als usuaris.

## 2.2. Casos d'ús

### 2.2.1. Guions

En principi ens trobem davant de tres guions al sistema que són els següents:

- Guió d'invitat. Un invitat pot fer ús del sistema. Aquest pot entrar al sistema i pot registrar-se com un usuari nou per gaudir de l'aplicació completa. També pot fer una cerca sobre el catàleg del sistema sobre un determinat llibre, autor, etc, o bé pot explorar el catàleg complet.
- Guió d'usuari registrat. Quan un usuari registrat accedeix al sistema, el primer que hauria de fer es identificar-se. Un cop realitzada la identificació, com en el cas dels usuaris no registrats, pot fer cerques o explorar el catàleg complet del sistema. Quan l'usuari troba un llibre que vol llegir, pot visualitzar aquest llibre online a través del sistema. Amb un llibre seleccionat, també pot comprar-ho i descarregar-ho a l'ordinador o a un dispositiu mòbil. També tindrà l'opció de puntuar i comentar el llibre. Un usuari registrat també podrà publicar el seu propi ebook al sistema, supeditat a la revisió per part d'un administrador.
- Guió d'administrador. Els administradors accediran al sistema per tal de gestionar el catàleg d'ebooks (donar d'alta i baixa). També seran els encarregats de gestionar les eventuais ofertes que es publiquin. Part de la seva funció també serà supervisar els ebooks que vulguin publicar els usuaris registrats i decidir en cada cas la seva publicació o no.

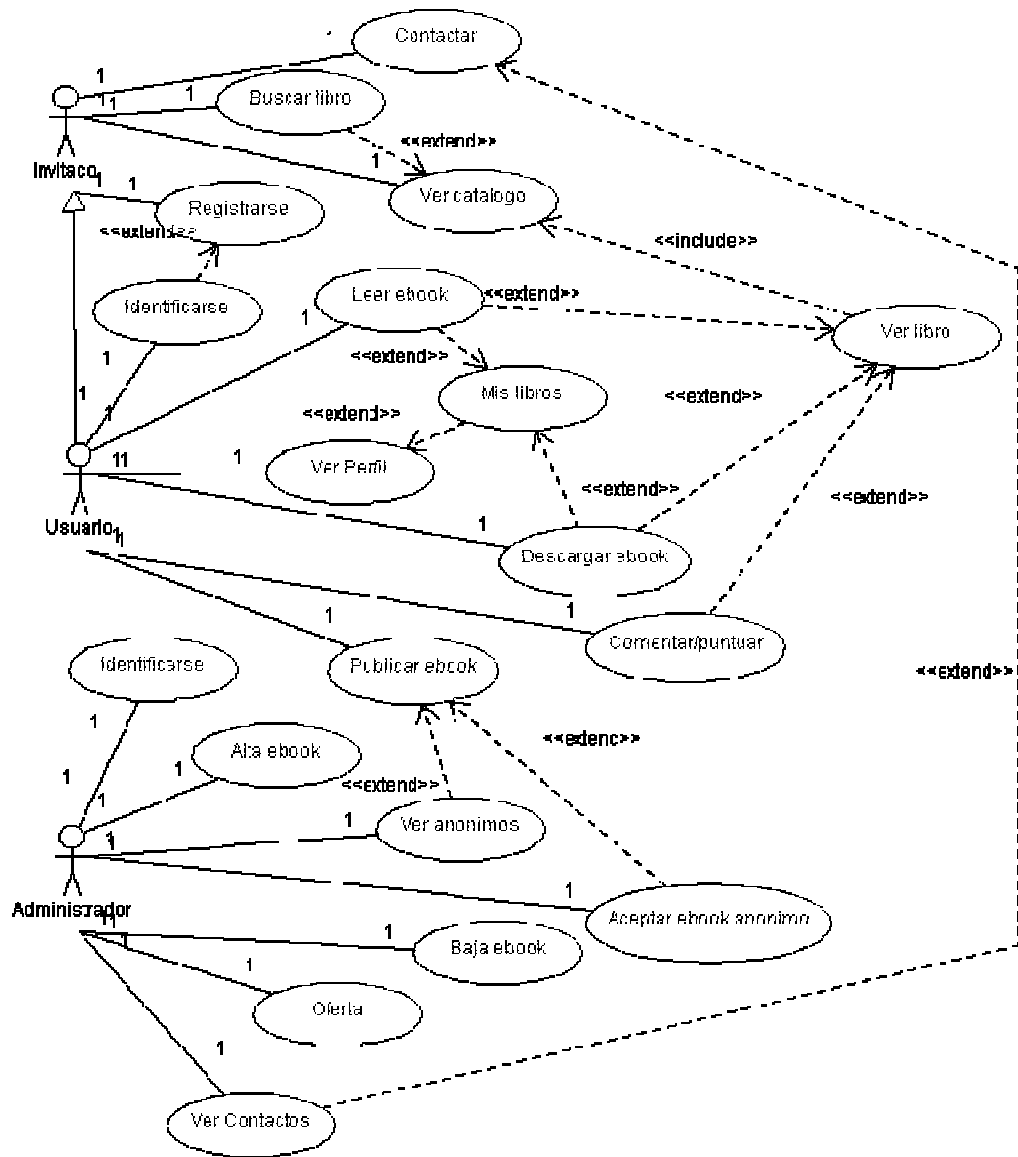
### 2.2.2. Actors

El nostre sistema constarà de tres actors que seran els següents:

- Administrador. Serà la persona encarregada de gestionar tot el catàleg d'ebooks de l'aplicació, així com de realitzar tasques de manteniment puntual.
- Usuari. Els usuaris seran els clients de l'aplicació que estiguin registrats al sistema. Gaudiran de totes les funcionalitats que oferim com visualitzar el catàleg, veure les característiques d'un cert ebook, descarregar-lo o llegir-lo online, publicar els seus propis ebooks i comentar i valorar els ebooks.
- Invitat. Els invitats serà qualsevol client que no estigui registrat al sistema. Òbviament, no podran gaudir de totes les funcionalitats del sistema, però podran visualitzar el catàleg d'ebooks i veure les característiques d'un llibre.

### 2.2.3. Casos d'ús

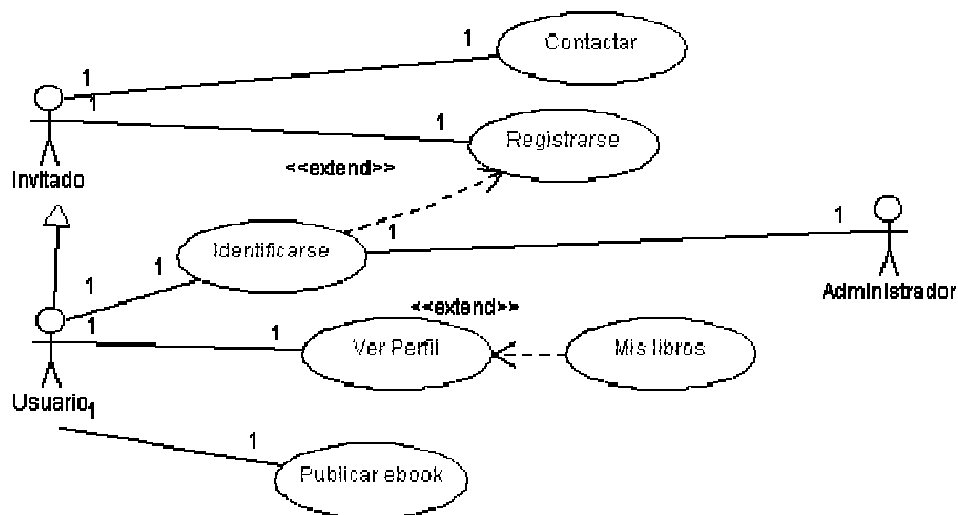
Un cop tenim identificats els actors que hi haurà al sistema, passem a analitzar els casos d'ús.



Il·lustració 2. Diagrama de casos d'ús

## 2.2.4. Documentació dels casos d'ús per subsistema

### Subsistema d'usuaris



II-lustració 3. Subsistema d'usuaris

#### Cas d'ús 1: "Registrarse"

Registrar-se al sistema.

Casos d'ús relacionats: Identificarse.

Actors: invitat.

Precondició: l'invitat no està registrat ja al sistema.

Postcondició: l'invitat passa a estar registrat al sistema.

Un invitat que visiti la nostra aplicació pot registrar-se al sistema per a gaudir de totes les funcionalitats. L'invitat haurà d'accedir a la pagina de registre i emplenar un formulari amb les seves dades i un nom d'usuari que l'identificarà al sistema.

Alternatives i excepcions: l'usuari ja existeix en el sistema, per tant, s'informa mitjançant un missatge d'error i s'avorta el nou registre.

#### Cas d'ús 2: "Identificarse"

Identificar-se al sistema.

Casos d'ús relacionats: Registrarse.

Actors: usuari, administrador.

Precondició: l'usuari ha d'estar registrat al sistema.

Postcondició: l'usuari s'ha identificat al sistema.

Quan un usuari esta registrat al sistema, accedeix al formulari de login i l'emplena amb el seu nom d'usuari i contrasenya. Un cop identificat, ja pot fer servir totes les funcionalitats.

Alternatives i excepcions: l'usuari no existeix al sistema. Hi ha un error en el login.

### **Cas d'ús 3: "Ver Perfil"**

Veure el perfil, és a dir, les dades personals que l'usuari va introduir en fer el registre.

Casos d'ús relacionats: [Mis Libros](#).

Actors: usuari.

Precondició: l'usuari està identificat al sistema.

Postcondició: l'usuari pot veure les seves dades personals.

Un usuari pot veure les seves dades personals sempre que vulgui amb aquesta funcionalitat. Quan l'usuari faci clic a "Perfil", se li presentaran un llistat amb les seves dades personals.

Alternatives i excepcions: l'usuari no està registrat al sistema, per tant no es mostra res.

### **Cas d'ús 4: "Mis libros"**

Veure les descarregues o les lectures d'un usuari.

Casos d'ús relacionats: [Leer ebook](#), [Descargar ebook](#), [Ver Perfil](#).

Actors: usuari.

Precondició: l'usuari ha d'estar identificat al sistema i haver realitzat alguna descarrega o lectura.

Un usuari pot consultar les seves descarregues o lectures que ha realitzat al sistema des de el seu perfil. El sistema li presentarà llavors una llista amb el seu historial dins el sistema amb totes les descarregues o lectures que hagi realitzat.

Alternatives i excepcions: l'usuari no ha realitzat cap descarrega o lectura, llavors no es mostrarà res.

### **Cas d'ús 5: "Contactar"**

Contactar per qualsevol motiu amb un administrador.

Casos d'ús relacionats: Ver Contactos.

Actors: invitado, usuari.

Postcondició: es crea un contacte al sistema amb les dades de l'invitat o l'usuari.

Un invitado o bé un usuari, disposa d'un formulari de contacte per posar-se en contacte amb l'administrador del sistema per qualsevol motiu. No cal estar identificat al sistema.

### Cas d'ús 6: "Publicar ebook"

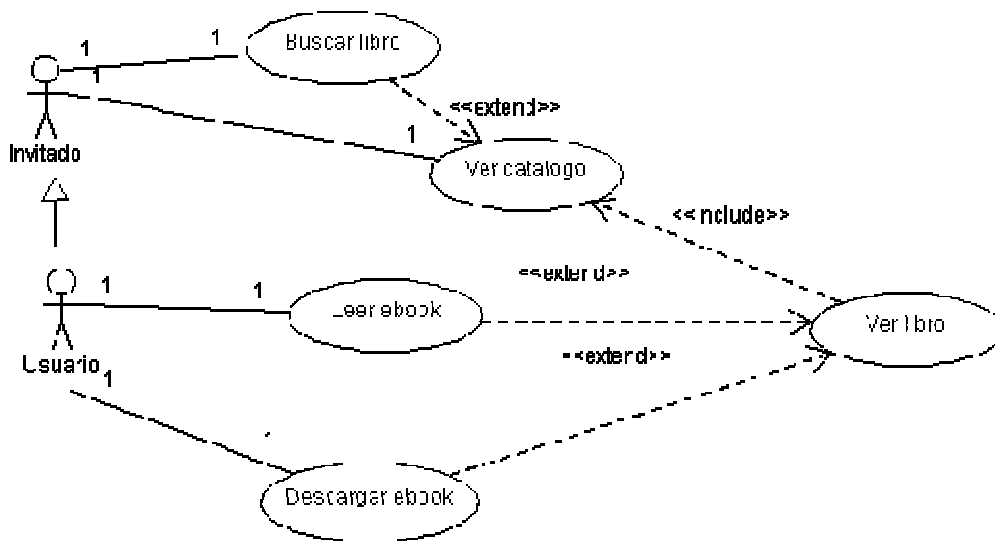
Publicar un ebook al sistema.

Casos d'ús relacionats: Aceptar ebook anónimo, Ver anónimos.

Actors: usuari.

El sistema ofereix a l'usuari identificat al sistema la possibilitat d'afegir ebooks propis al catàleg del sistema.

### Subsistema de venta



II-lustració 4. Subsistema de venta

### Cas d'ús 7: "Buscar libro"

Realitza una cerca al catàleg.

Casos d'ús relacionats: Ver catalogo.



Actors: invitat, usuari.

El client pot realitzar una cerca al catàleg del sistema mitjançant un quadre de cerca. Pot escollir l'opció per la qual buscar al catàleg que seran: "autor", "categoria", "títol" i "usuari".

#### **Cas d'ús 8: "Ver catalogo"**

Explorar el catàleg del sistema.

Casos d'ús relacionats: [Buscar libro](#), [Ver libro](#).

Actors: invitat, usuari.

Un client pot veure la totalitat del catàleg que ofereix l'aplicació.

#### **Cas d'ús 9: "Ver libro"**

Veure les característiques d'un ebook.

Casos d'ús relacionats: [Ver catalogo](#), [Descargar ebook](#), [Leer ebook](#), [Comentar/puntuar](#)

Actors: invitat, usuari.

Un client pot seleccionar veure les característiques d'un cert ebook des del catàleg. Se li presentarà en una nova pagina les característiques de l' ebook com son una imatge de la portada, el títol, una descripció i el preu. També s'ofereix la possibilitat de descarregar o llegir l' ebook i de comentar-lo i puntuar-lo.

#### **Cas d'ús 10: "Leer ebook"**

Llegeix un ebook online.

Casos d'ús relacionats: [Ver libro](#), [Mis libros](#).

Actors: usuari.

Precondició: l'usuari ha d'estar identificat al sistema.

Postcondició: si l'usuari ja havia començat a llegir l' ebook, es carrega des de la base de dades, si es una nova lectura, es crea un nou objecte Lectura.

Un usuari que estigui identificat al sistema podrà llegir qualsevol ebook online.

#### **Cas d'ús 11: "Descargar ebook"**

Descarregar un ebook.

Casos d'ús relacionats: [Ver libro](#), [Mis libros](#).

Actors: usuari.

Precondició: l'usuari no s'ha descarregat l' ebook abans.

Postcondició: l'usuari descarrega l' ebook.

Un usuari identificat al sistema pot descarregar-se un ebook qualsevol. Per a això, abans de procedir a la descarrega, ha de proporcionar una targeta per a fer el càrrec corresponent.

Alternatives i excepcions: l'usuari ja s'ha descarregat l' ebook abans, amb el qual s'avisava l'usuari de la situació.

## Subsistema social



Il·lustració 5. Subsistema social

### Cas d'ús 12: “Comentar/puntuar”

Comentar i puntuar un ebook.

Casos d'ús relacionats: [Ver libro](#).

Actors: usuari.

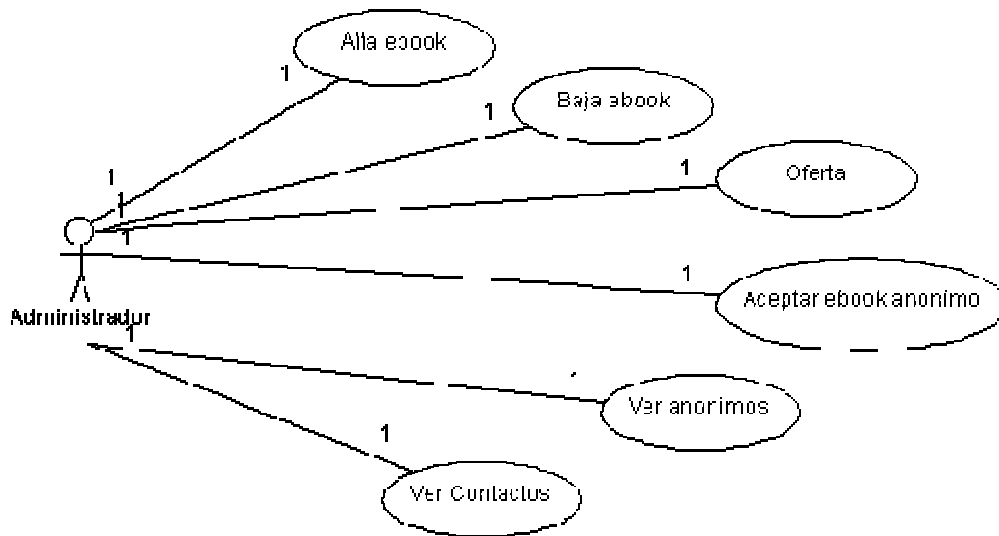
Precondició: l'usuari ha d'estar identificat al sistema, l'usuari no ha comentat i puntuat l' ebook abans.

Postcondició: l'usuari ha comentat i puntuat l' ebook.

L'usuari identificat pot fer un comentari sobre un ebook i donar-li una puntuació.

Alternatives i excepcions: l'usuari ja ha comentat i puntuat abans l', per tant s'avisava de la situació.

## Subsistema administració



II-lustració 6. Subsistema administració

### Cas d'ús 13: "Alta ebook"

Afegir un nou ebook al sistema.

Actors: administrador.

Precondició: el nou ebook no existeix al sistema.

Postcondició: l' passa a formar part del catàleg del sistema.

L'administrador del sistema serà l'encarregat de donar d'alta nous ebooks al sistema completant un formulari amb les característiques del nou ebook.

Alternatives i excepcions: l' ja existeix al sistema.

### Cas d'ús 14: "Baja ebook"

Eliminar un ebook del catàleg del sistema.

Actors: administrador.

Precondició: l' existeix al sistema.

Postcondició: l' ja no existeix al sistema.

L'administrador dona de baixa un ebook del sistema proporcionant l' identificació de l'.

### **Cas d'ús 15: “Oferta”**

Posar un ebook en oferta.

Actors: administrador.

Precondició: l' existeix al sistema.

Postcondició: l' s'ofereix amb una oferta.

L'administrador pot fer ofertes puntuals als ebooks.

### **Cas d'ús 16: “Acceptar ebook anónimo”**

Acceptar un ebook anònim pendent.

Casos d'ús relacionats: Publicar ebook.

Actors: administrador.

Precondició: hi ha un ebook anònim en estat pendent.

Postcondició: l' anònim passa a l'estat acceptat i al catàleg.

Després de que un usuari publiqui un ebook anònim, l'administrador ha de revisar el document per a comprovar que no hi hagi plagis o incompleixi alguna llei o alguna normativa de l'aplicació. Llavors l'administrador accepta l' que passa a estar en estat acceptat i passa a formar part del catàleg de l'aplicació.

Alternatives i excepcions: l'administrador determina que el llibre no pot formar part del catàleg i el rebutja.

### **Cas d'ús 17: “Ver Contactos”**

Veure contactes d'invitats i usuaris.

Casos d'ús relacionats: Contactar.

Actors: administrador.

Precondició: l'administrador està identificat al sistema i existeixen contactes al sistema.

Postcondició: l'administrador pot veure un llistat del contactes.

Quan un invitat o un usuari contacta amb l'administrador, en la pagina d'administració, l'administrador podrà veure un llistat amb tots els contactes que s'hagin produït al sistema.

Alternatives i excepcions: no hi ha contactes, per tant no es mostra res.

**Cas d'ús 18: “Ver Anónimos”**

Veure els ebooks anònims pendents d'acceptar.

Casos d'ús relacionats: Publicar ebook.

Actors: administrador.

Precondició: l'administrador està identificat al sistema i existeixen ebooks anònims pendents.

Postcondició: l'administrador pot veure una llista amb els ebooks anònims pendents.

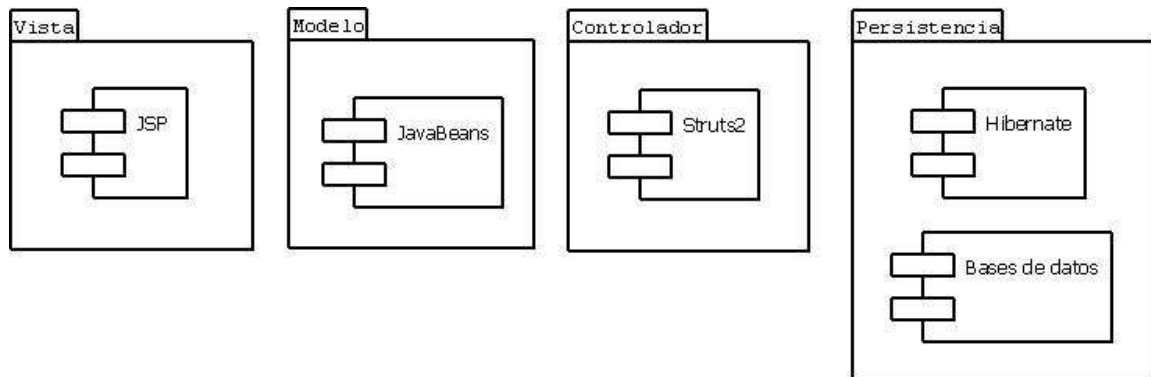
Quan l'administrador vol veure els ebooks anònims que hi ha al sistema pendents d'acceptar, clicarà a “veure anònims” i podrà observar un llistat amb tots els ebooks anònims pendents d'acceptar al sistema.

Alternatives i excepcions: no existeixen ebooks anònims pendents, per tant no es mostra res.

## 3. Anàlisi

### 3.1. Paquets d'anàlisi

Com que l'aplicació que es duu a terme es bastant gran i complexa, s'ha decidit dividir l'aplicació en paquets d'anàlisi separats els uns dels altres per a facilitar la tasca de programació i per a realitzar-la també d'una manera més ordenada i coherent. Els paquets en els quals s'ha dividit l'aplicació són quasi idèntics a l'arquitectura de l'aplicació. Els paquets resultants són els següents:



Il·lustració 7. Diagrama de paquets d'anàlisi

Com es pot veure, es té per un costat el paquet que correspon a la vista. Aquí s'inclourà tota la capa de presentació de l'aplicació, com les pàgines JSP, beans auxiliars, etc.

Al paquet del model s'inclou la programació de tota la lògica de negoci.

En el paquet de controlador tindrem tots els servlets que farà servir l'aplicació. En aquest cas concret, vers que es farà ús d'.struts2 per a construir l'aplicació, aquí hi hauran el actions.

Per últim, al paquet de persistència tindrem totes les funcionalitats que requereix la nostra aplicació en quant a la persistència de les dades. En el nostre cas concret tindrem la construcció de la base de dades amb MySQL i Hibernate.

### 3.2. Especificació de les classes d'anàlisi

#### 3.2.1. Identificació classes d'entitat

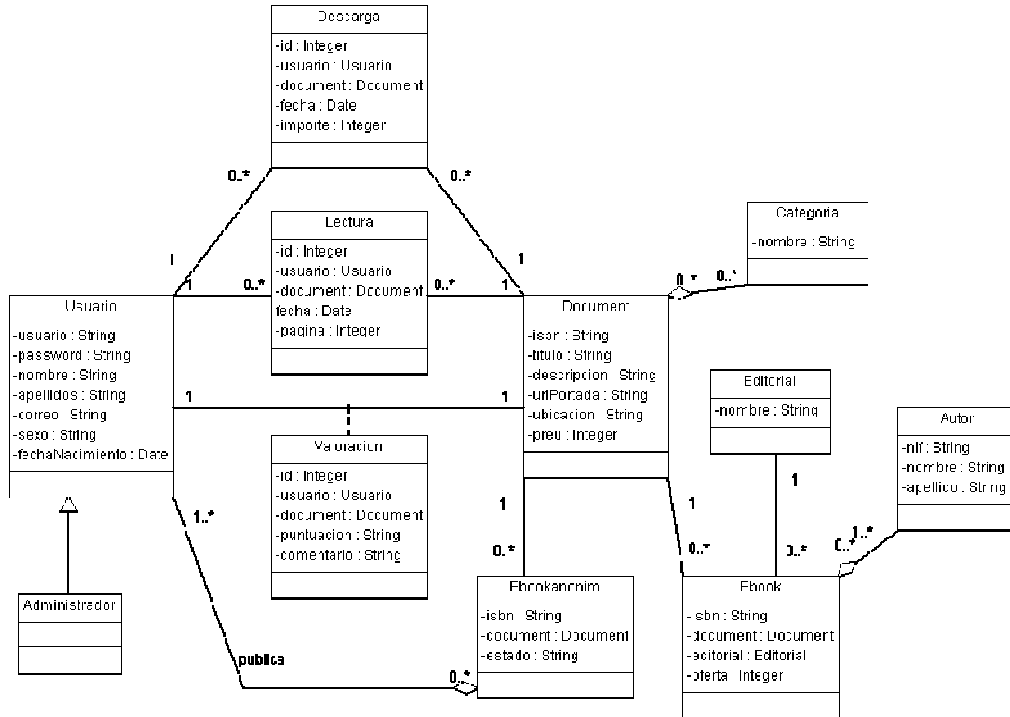
Després de la recollida de requisits i l'anàlisi dels casos d'ús, es detecten les següents classes d'entitat:

Usuari, Administrador, Ebook, Ebook Anònim, Descarrega, Lectura, Valoració.

Amb aquestes classes i els casos d'ús, es passa ara a construir el diagrama estàtic de disseny.

### 3.2.2. Diagrama de classes

A continuació es pot veure el diagrama de classes resultant de l'anàlisi.



Il·lustració 8. Diagrama de classes

## 4. Disseny

### 4.1. Decisions de disseny. Patrons i frameworks

Per a la nostra aplicació, s'ha decidit fer servir com a plataforma de programació J2EE. A més, per facilitar el desenvolupament, s'ha decidit fer servir els frameworks Hibernate i Struts2. Com a patrons de disseny es fa servir el Data Acces Object (DAO) i el Model-Vista-Controlador (MVC).

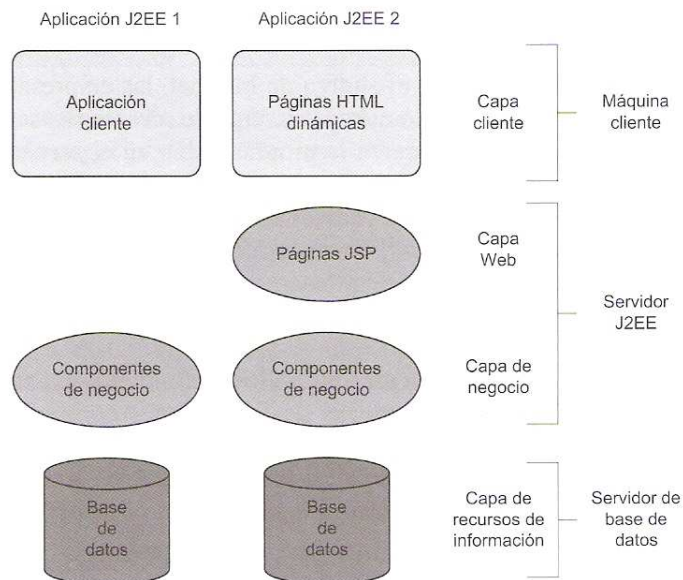
En els apartats següents es realitza una breu explicació de cada framework i patró usat.

#### 4.1.1. J2EE

J2EE es una solució que va oferir Sun Microsystems per l'alta demanda d'una API per programar aplicacions servidores econòmiques i eficients per oferir el seus serveis.

J2EE serveix per a desenvolupar aplicacions distribuïdes basades en components. Així es simple programar, perquè la funcionalitat que deuen proporcionar s'encapsula en els components J2EE. Això permet als programadors dividir l'aplicació segons les seves funcions.

J2EE utilitza un model d'aplicació distribuïda multicapa per a les aplicacions empresarials. La lògica de negoci esta dividida en components segons la seva funció i els diversos components de l'aplicació que constitueixen una aplicació J2EE son instal·lats en diferents maquines segons la funció de la capa a la que pertany el component. A continuació es pot observar dues aplicacions J2EE multicapa dividides per capes.

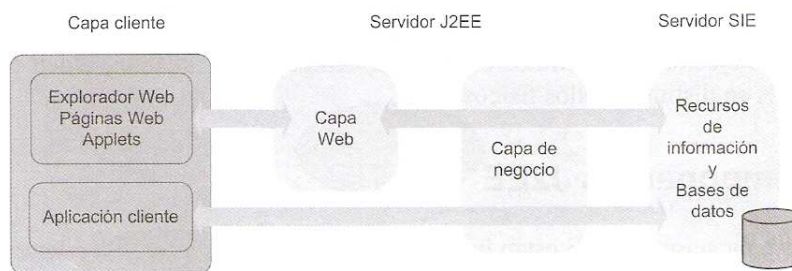


II-Iustració 9. Exemple aplicació J2EE



J2EE es basa en quatre nivells que detallem a continuació:

- Capa client. Els components d'aquesta capa s'executen en la màquina client. També es la capa de presentació. Esta composta dels programes que interactuen amb l'usuari. Aquests programes demanen les dades a l'usuari relacionades amb la petició que es realitzi, envien la petició al component que la processarà. Un cop hagi rebut la resposta, la presentarà a l'usuari.
- Capa web. Els components d'aquesta capa s'executen al servidor J2EE i fan servir el protocol HTTP per a enviar i rebre peticions. Aquesta capa proporciona a l'aplicació funcionalitat de Internet ja que permet proporcionar serveis a la capa client.
- Capa de negoci. Els components d'aquesta capa s'executen en el servidor. Contenen la lògica de negoci, tasques específiques basades en regles que resolen les necessitats d'un negoci en particular, que pot ser accedida concurrentment per múltiples clients. Un component d'aquesta capa sol interactuar amb la capa de recursos d'informació.
- Capa de recursos d'informació. Aquests components s'executen al servidor que administra aquests recursos. Connecta l'aplicació J2EE amb els sistemes i recursos que es troben a la xarxa, com bases de dades, sistemes de tercers, etc., d'aquesta manera els programadors poden aprofitar els recursos dels que ja disposa l'empresa.



II-lustració 10. Capes aplicació J2EE

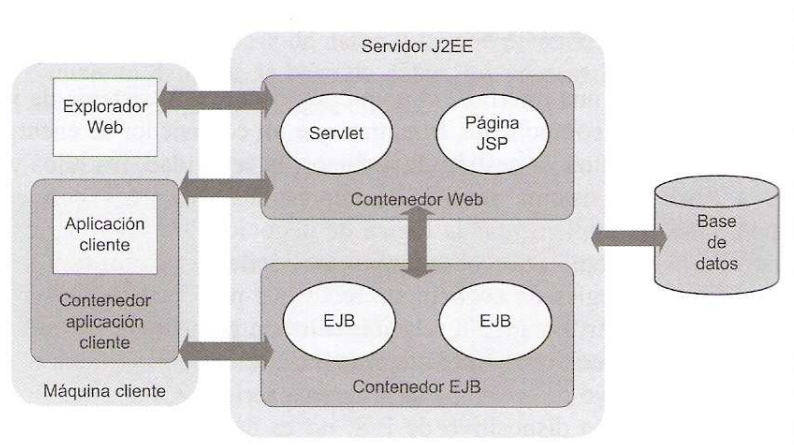
Les aplicacions J2EE estan fetes a base de components, que es una unitat de software independent que s'acobla a l'aplicació amb els seus fitxers i classes i que es comunica amb altres components. J2EE defineix els següents tipus de components:

- Aplicacions client i applets
- Servlets i JavaServer Pages (JSP)
- Enterprise JavaBeans (EJB)

J2EE també disposa de contenidors, que es una interfície entre un component i el sistema de nivell inferior que da suport al component. Un contenidor s'encarrega de la persistència de les dades, la gestió de recursos, la seguretat, els fils i altres serveis a nivell de sistema per als components associats a ell, oferint la possibilitat al programador de concentrar-se totalment en codificar les regles de negoci i no preocupar-se dels serveis del sistema. Els contenidors que J2EE disposa són els següents:

- Contenedor web. És el responsable de l'execució de les pàgines JSP i els servlets.

- Contenedor EJB. És el responsable de l'execució dels enterprise beans.
- Contenedor aplicació client. És el responsable de l'execució d'una aplicació client.
- Contenedor d'applets. És el responsable de l'execució d'applets.



**Il·lustració 11. Contenedors J2EE**

#### 4.1.2. Struts2

Struts és un framework per a aplicacions web sota el patró MVC per a la plataforma J2EE.

Struts simplifica el desenvolupament d'aplicacions distribuïdes implementant el patró MVC. Struts separa la gestió del flux de treball de l'aplicació, del model de negoci i de la interfície.

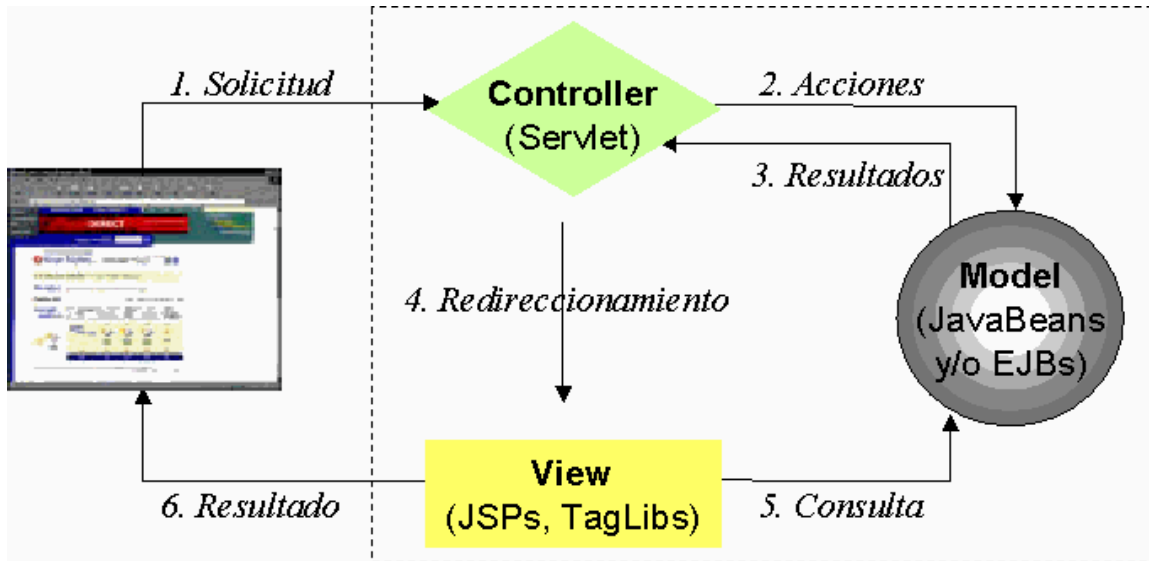
El controlador ja es troba implementat per struts, encara que es pot heretar i modificar, i el flux de treball es pot programar des d'un arxiu de configuració XML. Les accions que s'executaran sobre el model de negoci s'implementen basant-se en classes predefinides pel framework i seguint el patró Facade.

La interfície es desenvolupa mitjançant un conjunt de tags predefinits per Struts que evita l'ús d'scriptlets, petits fragments de codi Java que es poden introduir a la pagina web, el que genera avantatges de manteniment i rendiment.

Struts funciona de la següent manera:

- El navegador client genera una sol·licitud que s'envia i es atesa per Controller.
- El servlet Controller analitza la petició i crida a l'Action corresponent, passant-li els paràmetres de la petició, segons l'especificat al fitxer de configuració XML.
- L'Action farà servir els objectes de negoci necessaris per a atendre la petició.
- Segons el resultat produït per l'Action, Controller derivarà la generació de la interfície resultant en una o més pàgines JSP, que podran consultar els objectes del model.

A continuació podem observar esquemàticament el funcionament d'Struts:



Il·lustració 12. Funcionament Struts

### 4.1.3. Hibernate

Hibernate és un framework que facilita el mapeig objecte-relacional (ORM) per a la plataforma Java que facilita el mapeig dels atributs de les classes del model de negoci i els atributs d'una base de dades relacional. Per a fer aquesta tasca Hibernate proporciona arxius declaratius XML on es declaren les relacions entre els dos tipus de domini.

Hibernate busca solucionar el problema de la diferència entre els objectes de l'aplicació (orientats a objectes) i els objectes d'una base de dades relacional. Això ho aconsegueix permetent al desenvolupador declarar com es el seu model de dades, les relacions que existeixen, etc.

Hibernate convertirà les dades entre els tipus usats per Java i els usats per SQL. Hibernate genera les sentències SQL i evita la programació de SQL hostatjat a l'aplicació, fent el programa més aclaridor, portable i eficient.

Hibernate proporciona un llenguatge de consulta anomenat HQL (Hibernate Query Language) i una API per construir les consultes.

### 4.1.4. MVC

El patró Model-Vista-Controlador separa les dades d'una aplicació en la interfície d'usuari i la lògica de control en tres components. El patró MVC es sol fer servir en aplicacions web on la vista es la pàgina web i el codi que proveeix de dades dinàmiques a la pàgina, el model es el sistema de gestió de bases de dades i la lògica de negoci i el controlador es el responsable de rebre i tractar els esdeveniments que es produeixen a la vista.

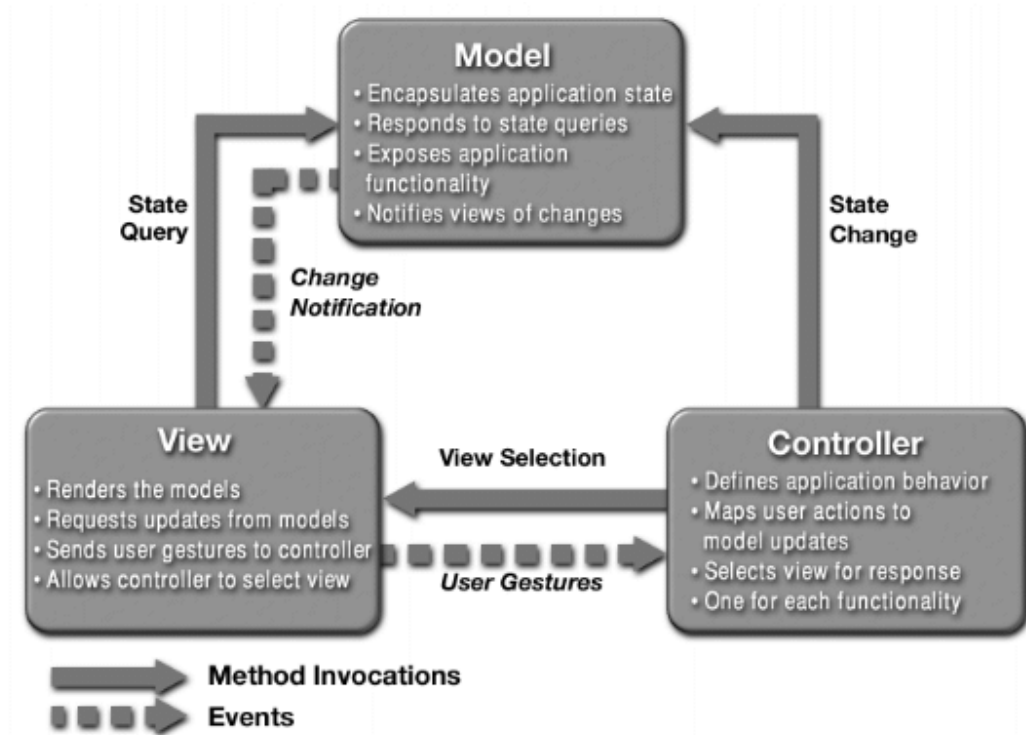
Els tres components del patró són:

- Model. Es la representació de la informació específica amb la qual l'aplicació opera.

- Vista. Presenta el model en un format adequat per a interactuar, normalment la interfície d'usuari.
- Controlador. Es el responsable de respondre als esdeveniments causats per accions de l'usuari i passa peticions al model. Les respostes les passa a la vista.

El flux d'operació que es produeix en un entorn que fa servir el patró MVC és el següent: l'usuari provoca un esdeveniment en la interfície d'usuari, per exemple, prem un botó; el controlador rep la petició sol·licitada i gestiona l'esdeveniment; el controlador accedeix al model i realitza les accions necessàries; el controlador notifica a la vista i delega en ella realitzar els canvis necessaris en la interfície d'usuari, obtenint les dades la vista directament des del model; la interfície d'usuari espera noves interaccions.

A continuació podem veure un esquema del patró MVC:



II-Il·lustració 13. Patró MVC

#### 4.1.5. DAO

El patró Data Access Object pretén fer independent una aplicació d'una base de dades concreta, de com s'accedeix a les dades, etc. Facilita que l'aplicació aconsegueixi les dades o pugui guardar-les sense saber d'on esta aconseguint o guardant aquestes dades.

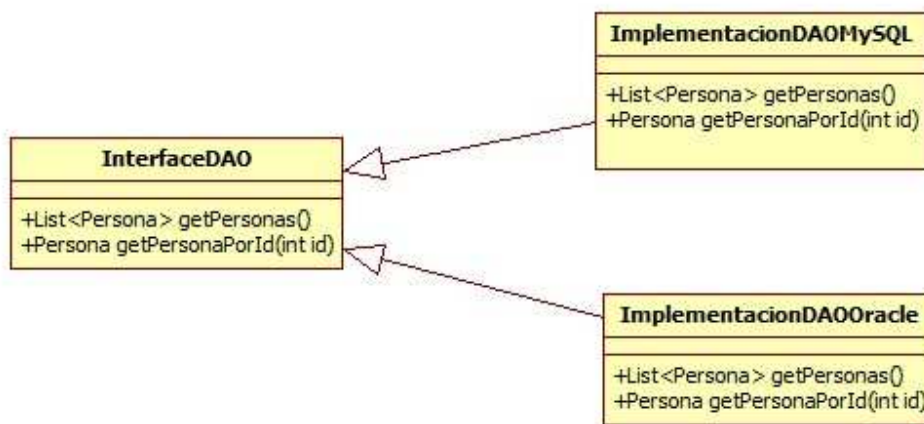
El patró funciona de manera senzilla. En primer lloc, hem de fer les classes que representen les nostres dades, es a dir, el model del negoci de l'aplicació. Després, s'ha de fer una interface. Aquesta interface ha de tenir els mètodes necessaris per a obtenir i guardar objectes del negoci. La

interface no tindrà res que la relacioni amb una base de dades específica ni qualsevol altra cosa específica del mode d'emmagatzemament que es vagi a fer servir.

Amb aquests requisits es deuria construir l'aplicació, fent servir les classes de la lògica de negoci i usant les interfaces corresponents per obtenir i modificar els objectes.

Després d'això, s'implementa la interface DAO ja amb una base de dades concreta o amb alguna eina determinada com Hibernate, iBatis, etc. En fer servir la interface DAO, es podrà passar qualsevol implementació a l'aplicació.

Normalment el patró DAO es completa amb algun tipus de Factoria, que es una classe que al demanar-li la interface, decideix quina implementació s'instancia.



Il·lustració 14. Patró DAO

## 4.2. Arquitectura de l'aplicació

Aquesta aplicació integra varies tecnologies que es detallen a continuació.

Com a llenguatge de programació s'ha triat Java sota la plataforma de J2EE. Aquesta plataforma proveeix de les eines necessàries per a desenvolupar aplicacions distribuïdes. Java com a llenguatge de programació facilita la tasca de l' implementació, ja que es un llenguatge obert, a més de la facilitat de portabilitat que proporciona gràcies a la compatibilitat en múltiples entorns i, a més, afegeix robustesa a l'aplicació.

Per a fer servir l'aplicació sota la plataforma J2EE es necessita un servidor d'aplicacions sobre el qual executar l'aplicació. En aquest cas, s'ha triat com a servidor l'Apache Tomcat ja que es de lliure distribució i s'integra molt bé amb Java i amb l'IDE triat.

Com a IDE per a treballar i desenvolupar l'aplicació s'ha triat NetBeans, que es de lliure distribució i ofereix múltiples funcionalitats.

La interfície d'usuari està desenvolupada sota les tecnologies de J2EE per a pàgines web que son JSP. Per a facilitar el treball, juntament amb les pàgines JSP es fa servir dues biblioteques de tags que son la biblioteca de tags d'struts2 i la biblioteca estàndard de tags de JSP, la JavaServer Page Standard Tag Library (JSTL), concretament la biblioteca bàsica (core).

Per a gestionar la persistència de l'aplicació, es fa servir una base de dades relacional amb el gestor de base de dades MySQL, que com els altres productes, es de lliure distribució.

A part de les tecnologies mencionades anteriorment, es construeix aquesta aplicació amb dos frameworks que son Struts2 i Hibernate.

Struts 2 es una eina de suport a la construcció d'aplicacions distribuïdes que fa servir el patró de disseny MVC. Struts 2 ens dona les eines necessàries per a gestionar la capa de presentació de l'aplicació gracies a la llibreria de tags, i també ens proporciona les eines adequades per a construir els servlets que controlaran l'aplicació, que struts els anomena "actions".

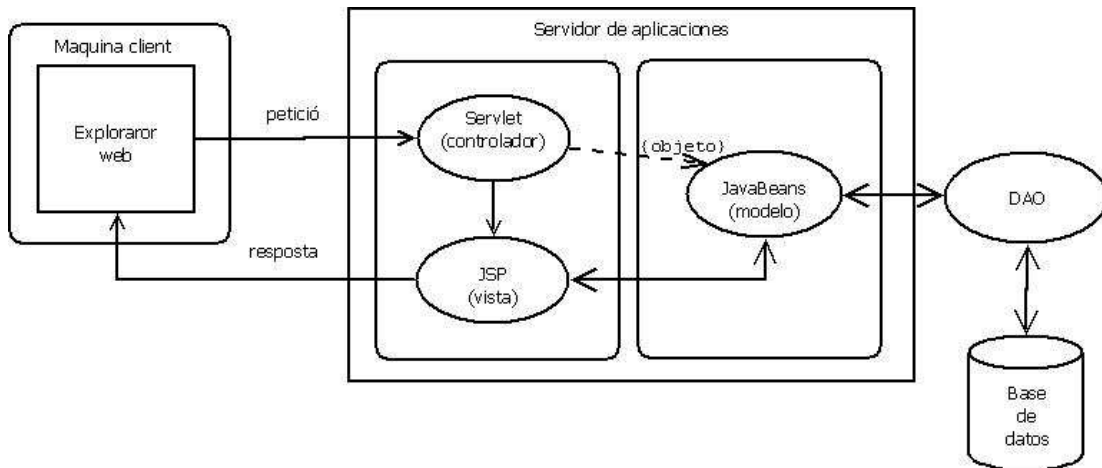
Hibernate es una eina de mapeig objecte-relacional (ORM) per a la plataforma Java que facilita el mapeig dels atributs entre una base de dades relacional i un model orientat a objectes d'una aplicació, mitjançant arxius declaratius. Facilita així la interacció entre els objectes de l'aplicació i la base de dades.

L'aplicació segueix dos patrons de disseny clarament identificats que son el patró MVC (Model – Vista – Controlador) i el patró DAO (Data Access Object).

El patró MVC separa les dades, la interfície d'usuari i la lògica de negoci d'una aplicació en tres components diferents. Aquests components son el model, que integra les dades i la lògica del negoci; la vista, que presenta la interfície d'usuari amb la qual es pot interactuar; i el controlador, que respon a les accions de l'usuari dins l'aplicació i interactua amb el model i la vista.

El patró DAO subministra una interfície comú entre l'aplicació i un o més dispositius d'emmagatzemament o tecnologies d'emmagatzemament. Això vol dir que tenim un objecte DAO que ofereix totes les operacions comunes de persistència d'un objecte de la lògica de negoci. Després especialitzem aquest DAO per cada base de dades que tinguem o per els frameworks que fem servir.

A continuació es pot observar un esquema de l'arquitectura de l'aplicació.



Il·lustració 15. Arquitectura de l'aplicació

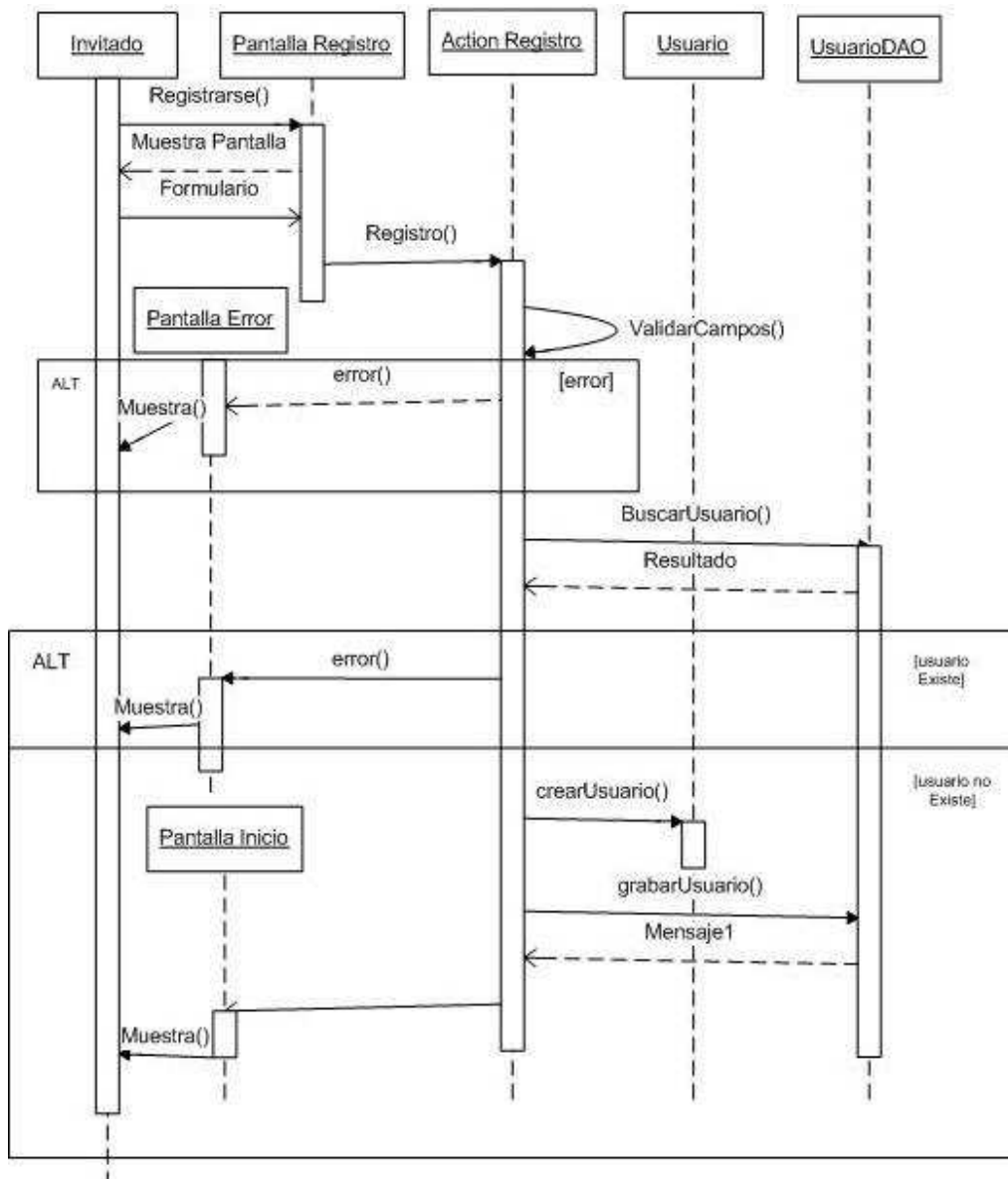
### 4.3. Disseny dels casos d'ús

A continuació es passa a presentar el disseny dels casos d'ús mitjançant diagrames UML dinàmics i d'implementació. Només es presentaran els diagrames dels principals casos d'ús.

#### 4.3.1. Diagrames de seqüència

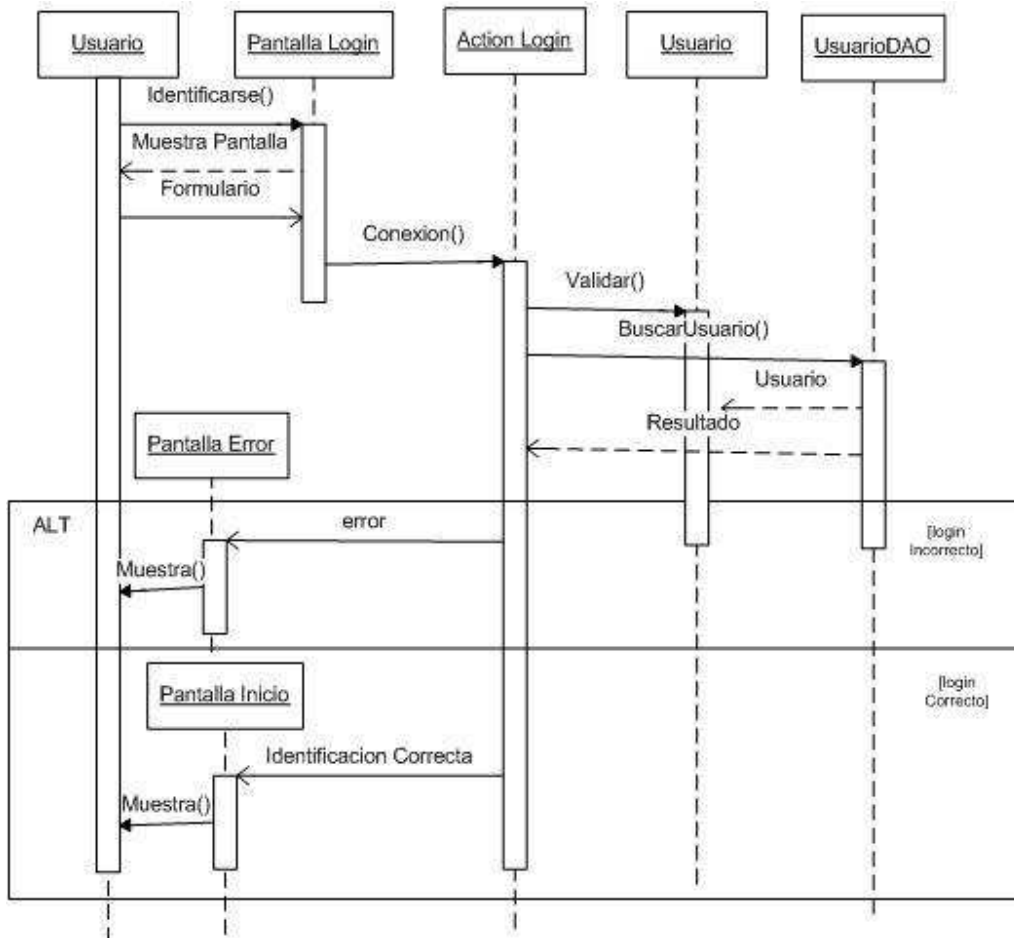
##### Subsistema d'usuaris

Cas d'ús "Registrarse"



II-Iustració 16. Diagrama de seqüència de "Registrarse"

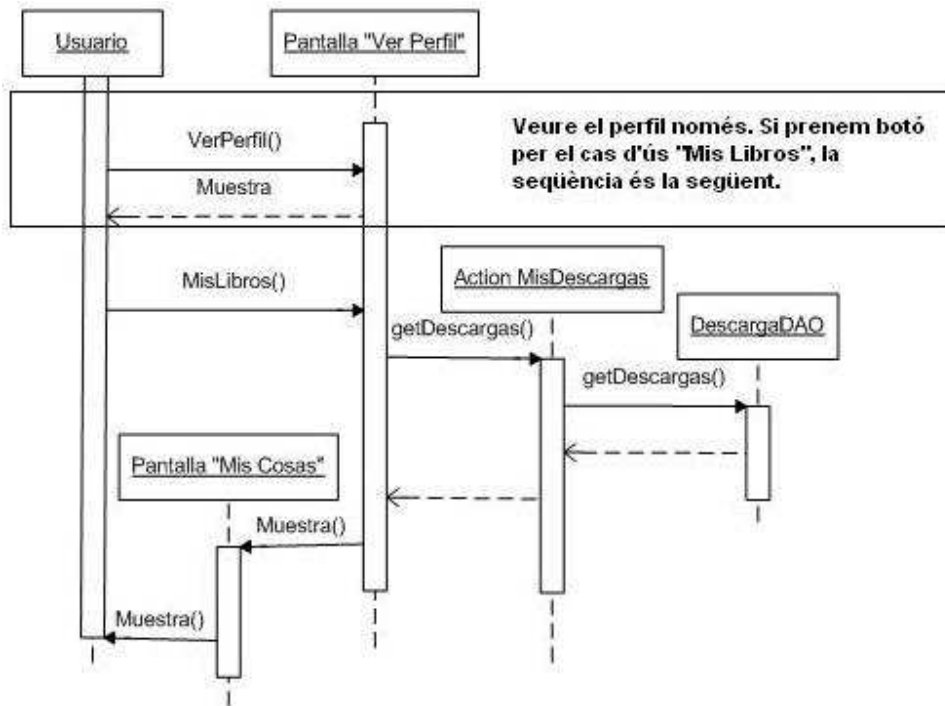
Cas d'ús "Identificarse"



II-lustració 17. Diagrama de seqüència de "Identificarse"



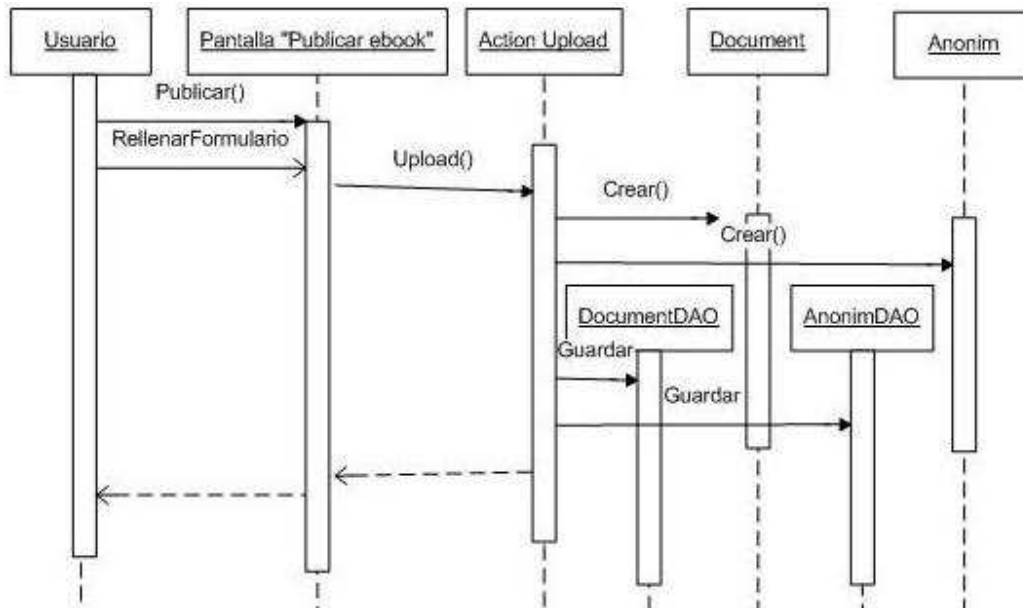
Cas d'ús "Ver Perfil" i "Mis Libros"



Il·lustració 18. Diagrama de seqüència de "Ver Perfil" i "Mis Libros"

La seqüència dins dels requadre pertany al cas d'ús "Ver Perfil". Com que el cas d'ús "Mis Libros" és una extensió de "Ver Perfil", per a veure "Mis Libros" cal executar també "Ver Perfil", ja que es des del perfil on es podrà veure les descarregues i les lectures. A la imatge es mostra la seqüència que mostra les descarregues, però la que mostra les lectures és exactament igual excepte que es crida a l'acció "MisLecturas" i aquest contacta amb "LecturaDAO".

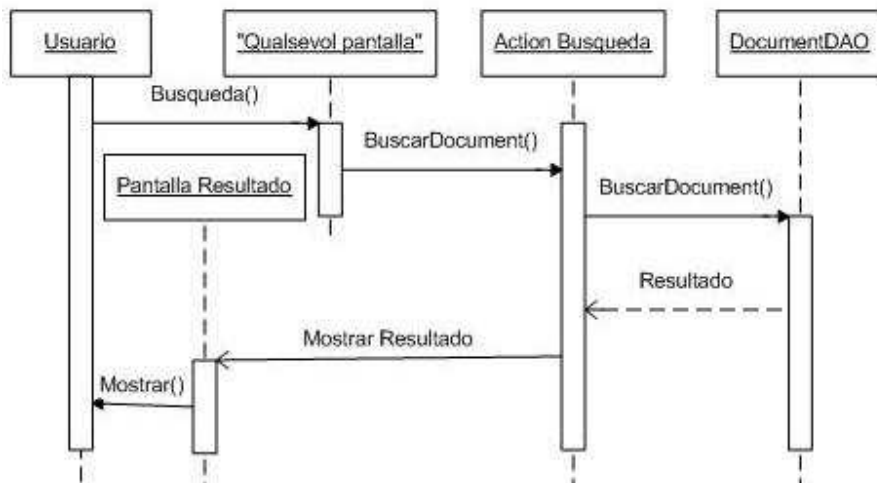
Cas d'ús "Publicar Ebook"



Il·lustració 19. Diagrama de seqüència de "Publicar Ebook"

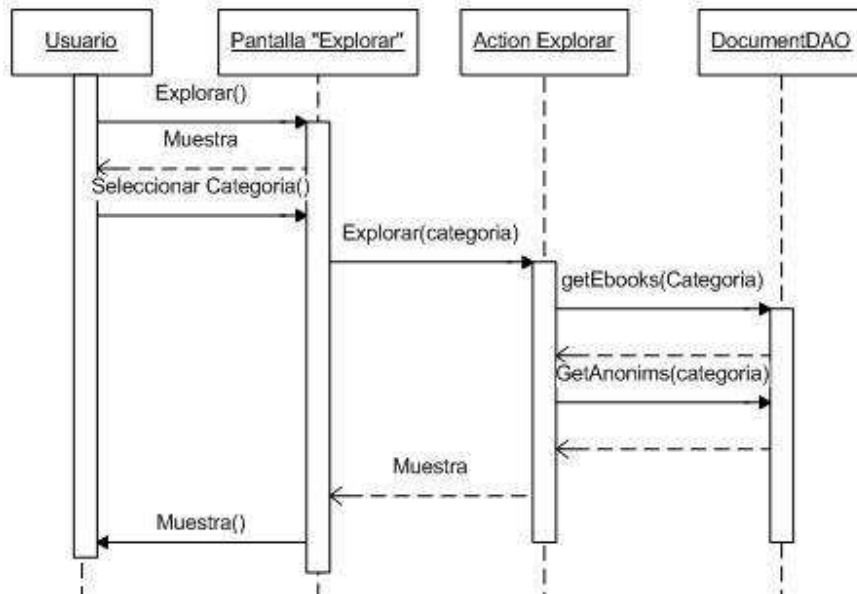
Subsistema de venta

Cas d'ús "Buscar Libro"



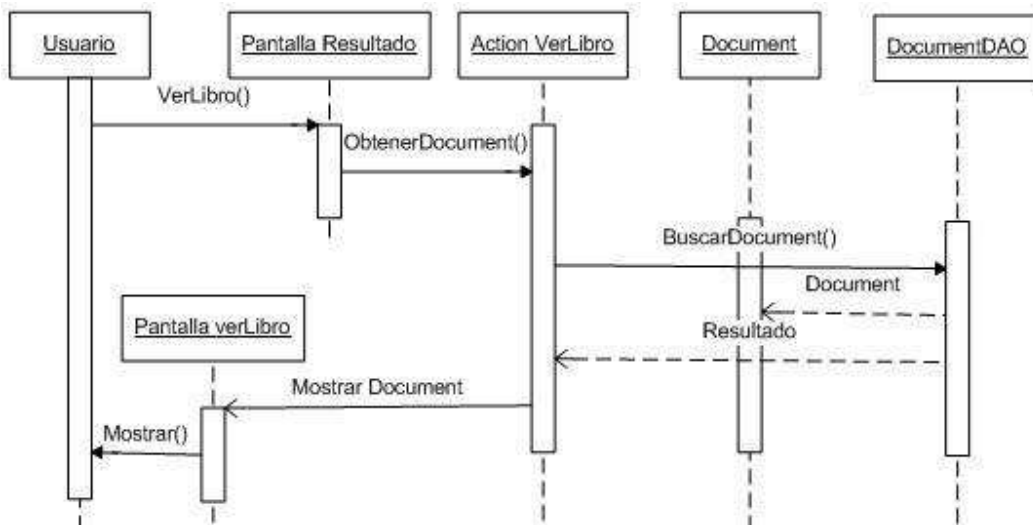
Il·lustració 20. Diagrama de seqüència de "Buscar Libro"

Cas d'ús "Ver Catalogo"



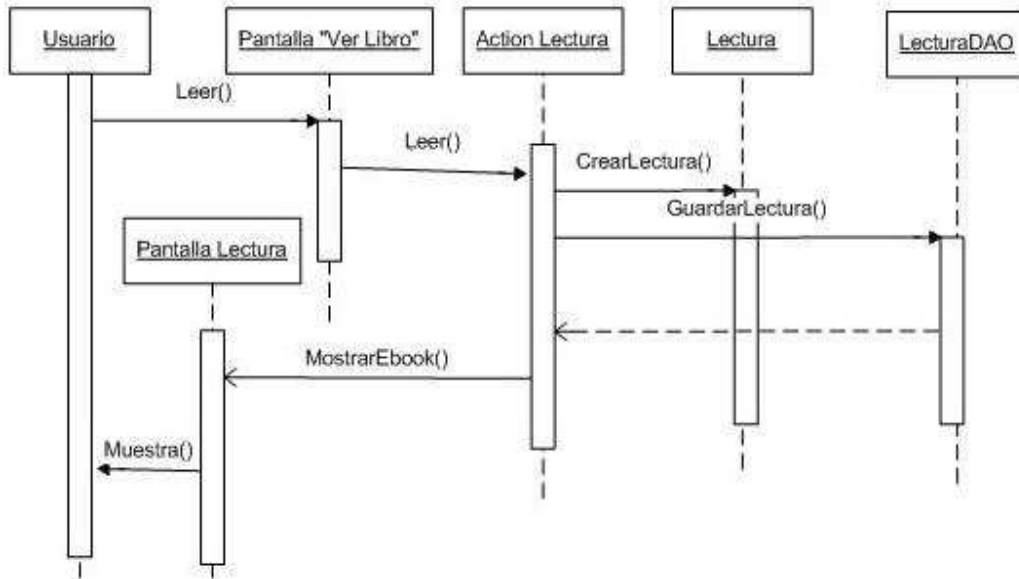
II-lustració 21. Diagrama de seqüència de "Ver Catalogo"

Cas d'ús "Ver Libro"



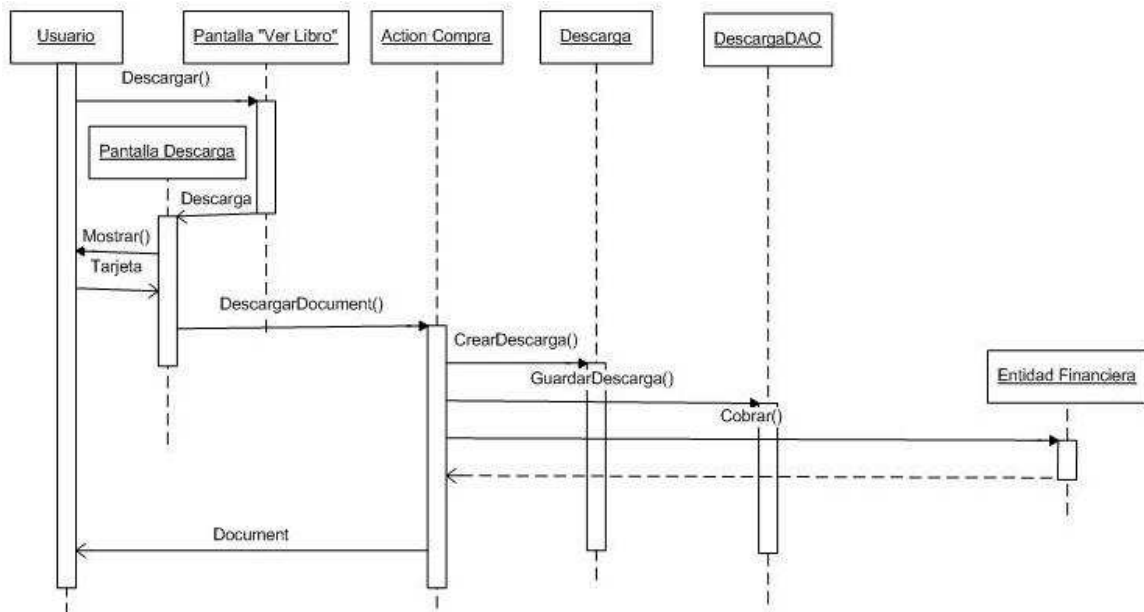
II-lustració 22. Diagrama de seqüència de "Ver Libro"

Cas d'ús "Leer ebook"



II-Ilustració 23. Diagrama de seqüència de "Leer Ebook"

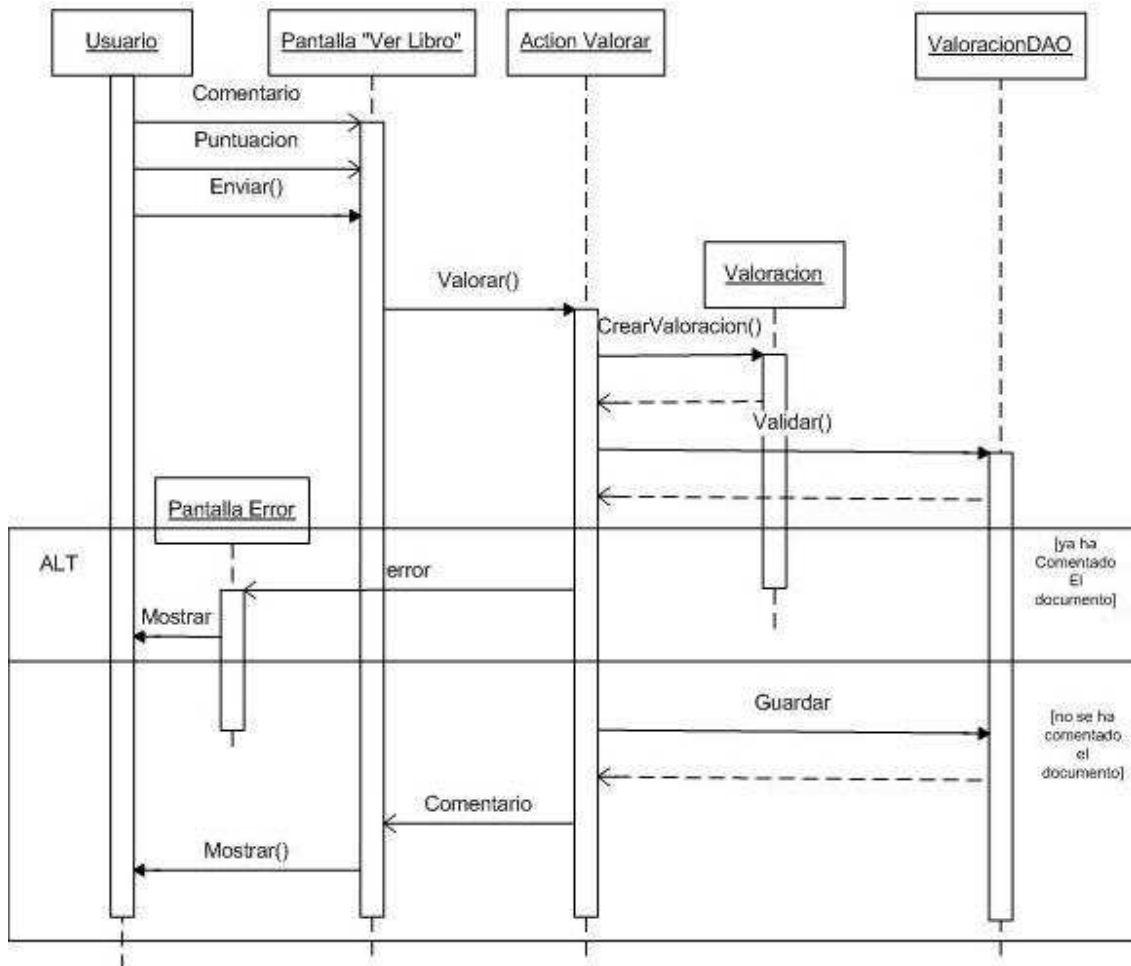
Cas d'ús "Descargar ebook"



II-Ilustració 24. Diagrama de seqüència de "Descargar Ebook"

## Subsistema social

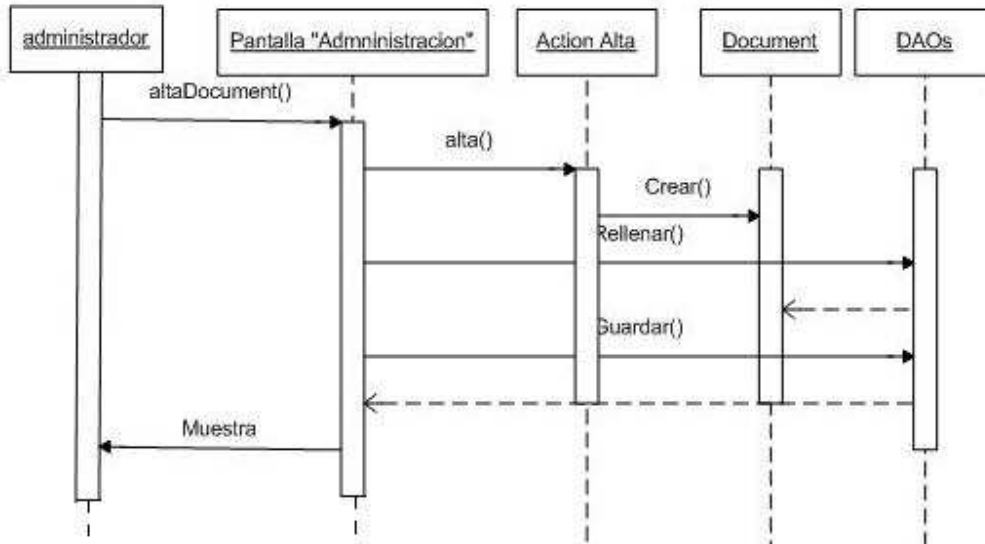
Cas d'ús "Comentar/Puntuar"



II-lustració 25. Diagrama de seqüència de "Comentar/Puntuar"

## Subsistema d'administració

Cas d'ús "Alta Ebook"

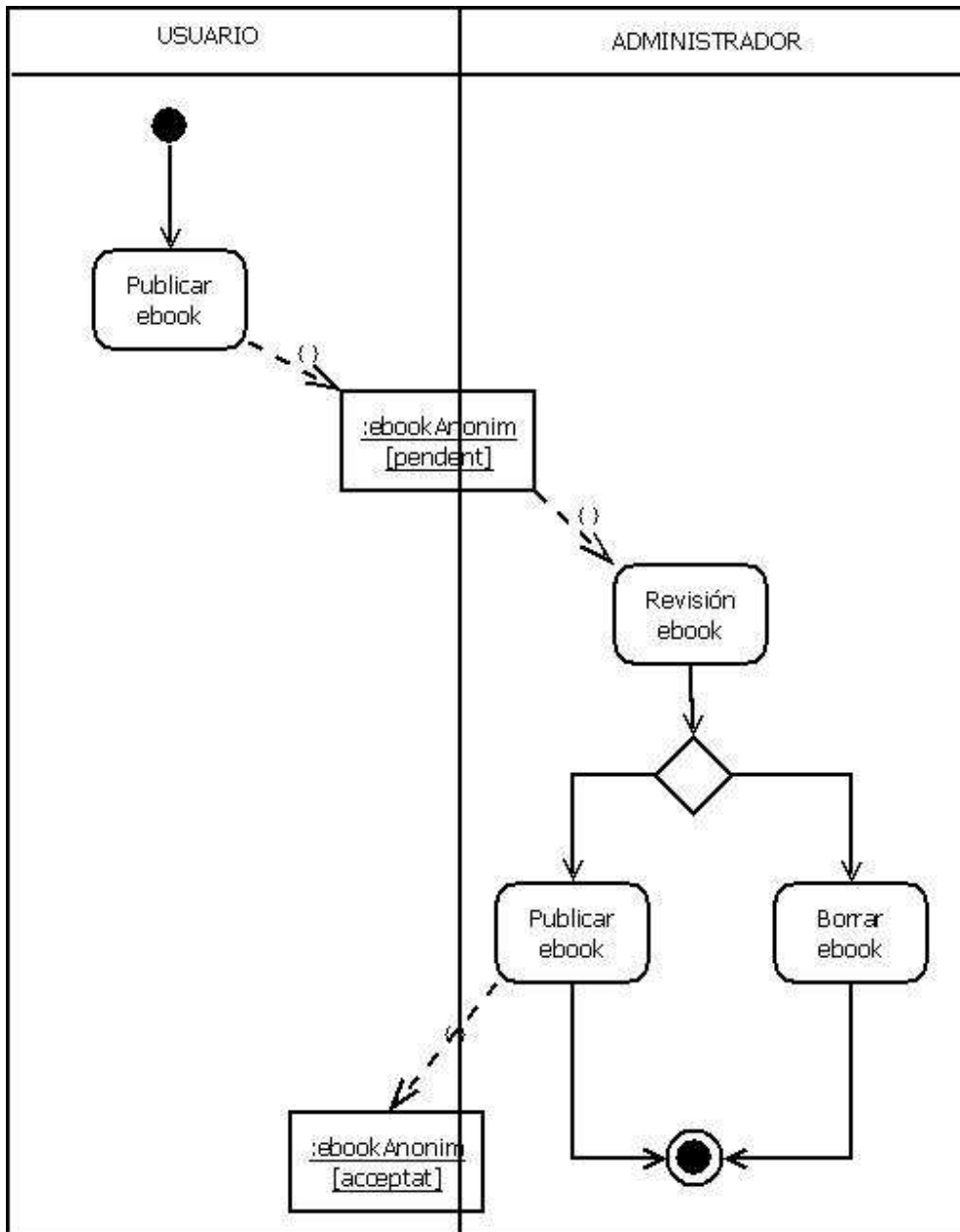


Il·lustració 26. Diagrama de seqüència de "Alta Ebook"

En la il·lustració es pot observar que després de crear el Document, tenim l'acció "Rellenar" fent referència a DAOs. Això es així perquè en donar d'alta un nou ebook, cal o bé actualitzar la base de dades amb la categoria de l'ebook, l'autor, instanciar un nou ebook, etc. o bé crear noves entrades a la base de dades si no existeixen.

### 4.3.2. Diagrames d'activitat

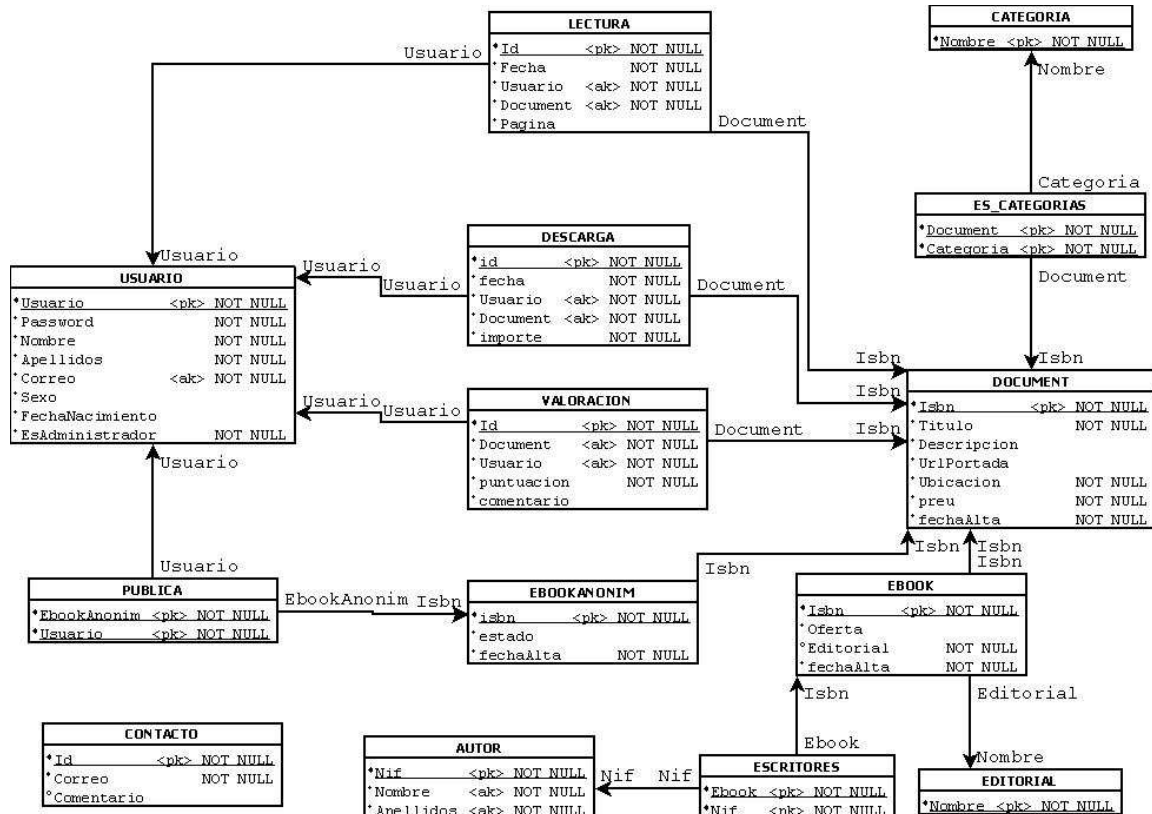
Es realitza un diagrama d'activitat per als casos d'ús "publicar ebook" i "acceptar ebook anònim" que impliquen dos actors diferents i per a plasmar clarament els estats pel qual passa un ebook anònim.



Il·lustració 27. Diagrama d'activitat del cas d'ús "Publicar ebook"

#### 4.4. Disseny de la persistència

A continuació es pot observar el diagrama en UML de la base de dades relacional de l'aplicació.



II-lustració 28. Diagrama de persistència

#### 4.5. Disseny de la interfície d'usuari

Es vol una interfície d'usuari facil d'utilitzar i molt intuïtiva. Es vol que la navegació entre pàgines sigui còmoda i intuïtiva.

##### 4.5.1. Elements comuns a totes les pàgines

Es té una sèrie d'elements comuns a totes les pàgines de l'aplicació. Com que es vol que la navegació pel lloc sigui còmoda, es decideix fer un menú de navegació comú a totes les pàgines. Aquest menú es trobarà al principi de totes les pàgines. A mode de presentació atractiva per l'usuari, es decideix incloure aquests menús sobre una franja blava de fons i la ubicació serà al principi de la pàgina i al centre de la franja, així s'aconsegueix que la navegació estigui sempre ubicada al mateix lloc i es facilita la tasca a l'usuari de navegar entre pàgines.

En aquesta franja comuna també s'inclourà un logotip de l'aplicació que farà de vincle amb la pàgina inicial de l'aplicació.



Per facilitar l'exploració del catàleg de l'aplicació, també es decideix incloure en aquesta franja una petita capsa de cerca de llibres amb diferents opcions de cerca.

A continuació es pot veure una captura de la franja principal:



II-lustració 29. Barra principal

## 4.5.2. Pàgines

La resta de pàgines s'han dissenyat el més còmodes i atractives possibles per a l'usuari. A continuació es presenten una sèrie de captures de les pàgines de l'aplicació.

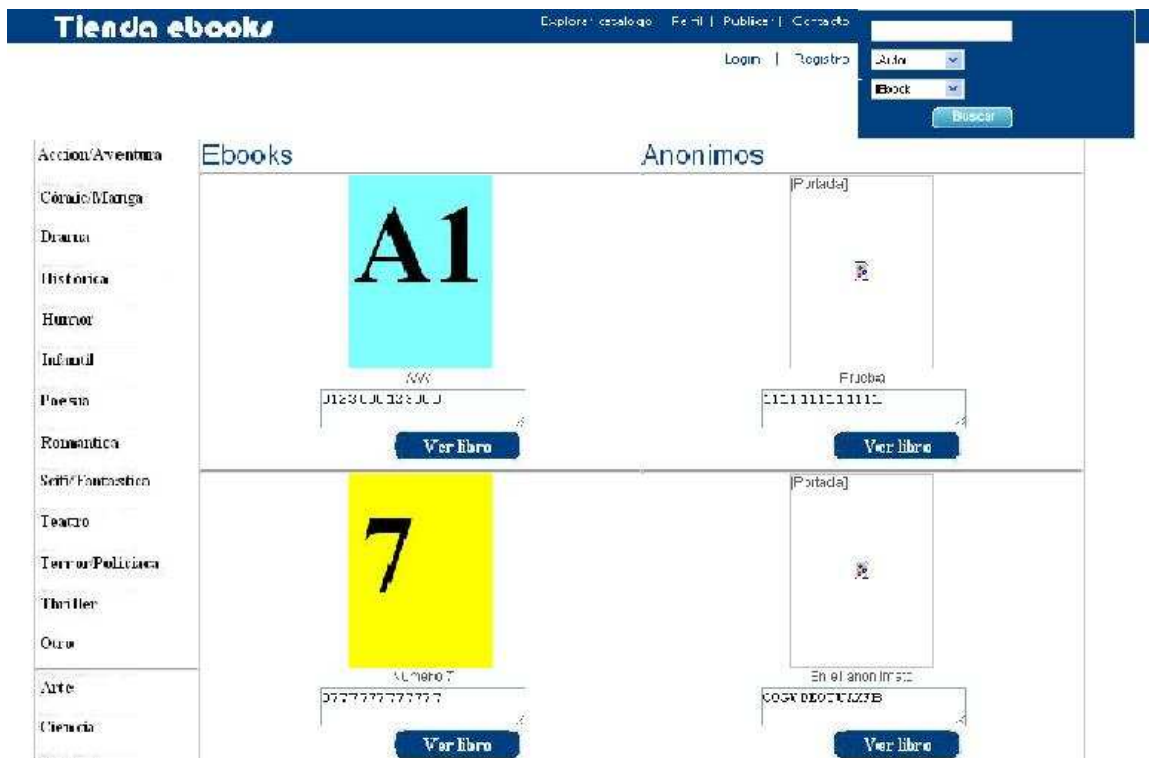


II-lustració 30. Pantalla d'inici

Com podem veure en la pantalla d'inici es té al centre de la pàgina una petita selecció dels últims llibres electrònics incorporats al catàleg de l'aplicació. Aquesta llista consta d'una petita imatge de

la portada del llibre i una mostra d'informació on podem veure l'identificador, el títol, una descripció i el preu.

A la part dreta de la pantalla es té un petit quadre on s'explica breument les característiques de l'aplicació, dissenyat a captar nous usuaris i a sota un formulari de login.



II-lustració 31. Pantalla d'exploració de catàleg

En aquesta pantalla es pot explorar tot el catàleg de l'aplicació. Es disposa a la part esquerra d'una llista amb totes les categories de llibres on podem seleccionar la que volem explorar. A la part central de la pagina es veurà els resultats classificats per ebooks i anònims.

II-lustració 32. Pantalla registre

En aquesta pantalla es on els invitats que vulguin registrar-se hauran d'emplenar un formulari amb les seves dades personals.

II-lustració 33. Pantalla publicar anònim

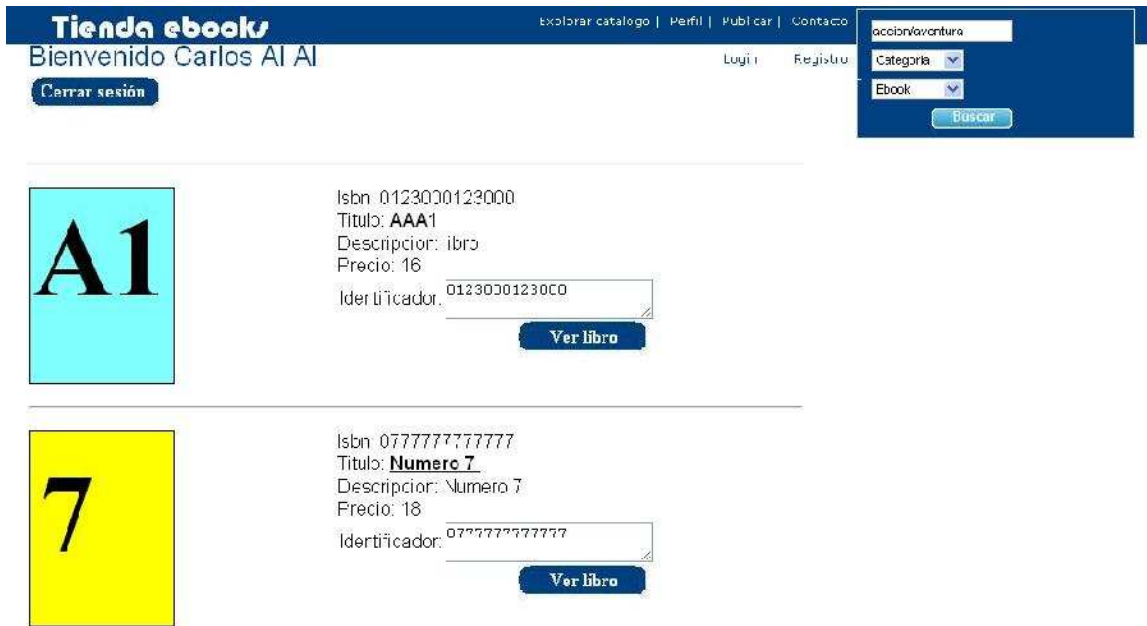
En aquesta pantalla els usuaris que estiguin registrats i identificats al sistema podran pujar els seus propis llibres. El formulari consta d'un títol pel llibre i una breu descripció. Caldrà seleccionar també la categoria del llibre en qüestió, posar un preu i seleccionar l'arxiu amb l'ebook i una portada.

The screenshot shows the 'Tienda ebooks' interface. At the top, there is a navigation bar with links for 'Explorar catalogo', 'Perfil', 'Publicar', and 'Contacto'. The user is logged in as 'Carlos Al Al', with options for 'Cerrar sesión', 'Login', and 'Registro'. A search bar is present with dropdown menus for 'Autor' and 'Ebook', and a 'Buscar' button. Below the navigation, a list of books is displayed with the following details:

Id: 24
Usuario: caros
Documento: 0123000123000
Fecha: 2011-05-27
Importe: 16
Id: 23
Usuario: caros
Documento: 0777777777777
Fecha: 2011-05-24
Importe: 10
Id: 2
Usuario: caros
Documento: 1234567890123
Fecha: 2011-04-06
Importe: 14
Id: 3
Usuario: caros
Documento: 2222222222222
Fecha: 2011-04-06
Importe: 11

### II-Il·lustració 34. Pantalla descarregues i lectures

Aquesta pantalla presenta les descarregues o lectures que hagi fet l'usuari, que pot seleccionar des de la pagina del seu perfil.



II-il·lustració 35. Pantalla de resultats de cerca

Aquesta pantalla mostra els resultats d'una cerca introduïda per un usuari a la capsa de cerca de la barra principal.



II-il·lustració 36. Pantalla veure llibre

Aquesta pantalla mostra les característiques completes d'un llibre. Aquesta pagina es mostra quan l'usuari prem el botó "ver libro". A l'esquerra es presenten les característiques completes del llibre en qüestió i hi ha dos botons per llegir el llibre online o descarregar-lo, en tal cas es presenta una pagina on l'usuari ha d'introduir el numero de targeta. A la dreta hi ha un formulari per puntuar i comentar el llibre i a sota hi ha una llista amb els comentaris i puntuacions que han realitzat els diferents usuaris.

Cal estar registrat i identificat al sistema tan per descarregar, llegir o puntuar i comentar un llibre.

Les funcionalitats de descarrega i presentació online del llibre no es troben implementades en aquest projecte ja que són implementacions més complexes que queden fora de l'abast i els objectius d'aquest Treball Final de Carrera.

## 5. Implementació

### 5.1. Requeriments de programari

Per a la correcta implementació d'aquest programari es necessita d'una sèrie de programari per a desenvolupar l'aplicació que s'ha dissenyat i que serà el següent:

- Per la capa client i fer funcionar l'aplicació, es necessita un navegador web qualsevol que suporti l'especificació HTML 4.01.
- Per al desenvolupament de l'aplicació es necessita el kit de desenvolupament Java, la JDK de J2SE.
- Per a la implementació de la base de dades es fa servir MySQL com a gestor de bases de dades.
- L'aplicació s'executarà sobre el servidor d'aplicacions Apache Tomcat.
- Sistema operatiu Windows XP, que es on s'ha desenvolupat l'aplicació. En principi el software emprat es pot trobar per altres sistemes operatius i l'aplicació, com que està escrita amb Java, es fàcilment portable.
- MySQL Connector Java 5.1.12 (<http://www.mysql.com>) per connectar amb la base de dades.

Adicionalment es necessiten les següents biblioteques que es proporcionen junt amb el projecte.

- Biblioteques de tag de JSTL (JSP Standard Tag Library). En concret la biblioteca core (<http://java.sun.com/jsp/jstl/core>).
- Biblioteques de Hibernate i Struts.

### 5.2. Eines de desenvolupament

Per a facilitar la tasca de programar l'aplicació, es faran servir les següents eines:

- IDE Netbeans.
- Plugin Struts2 per Netbeans 6.9.

### 5.3. Script de creació de la base de dades

```
DROP DATABASE IF EXISTS tienda_ebooks;
```

```
CREATE DATABASE tienda_ebooks;
```

```
USE tienda_ebooks;
```

```
CREATE TABLE Editorial(  
nombre VARCHAR(30) NOT NULL,  
PRIMARY KEY (nombre)
```

) ENGINE = InnoDB;

```
CREATE TABLE Document (
isbn VARCHAR(13) NOT NULL,
titulo VARCHAR(50) NOT NULL,
descripcion VARCHAR(250),
urlPortada VARCHAR(100),
ubicacion VARCHAR(100) NOT NULL,
preu INTEGER NOT NULL,
fechaAlta    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (isbn)
) ENGINE = InnoDB;
```

```
CREATE TABLE Ebook (
isbn VARCHAR(13) NOT NULL,
oferta INTEGER,
editorial VARCHAR(30) NOT NULL,
fechaAlta    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (isbn),
FOREIGN KEY (isbn) REFERENCES Document(isbn)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (editorial) REFERENCES Editorial(nombre)
ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE = InnoDB;
```

```
CREATE TABLE EbookAnonim (
isbn VARCHAR(13) NOT NULL,
estado ENUM('Pendent','Acceptat'),
fechaAlta    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (isbn),
FOREIGN KEY (isbn) REFERENCES Document(isbn)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = InnoDB;
```

```
CREATE TABLE Autor (
nif VARCHAR(9) NOT NULL,
nombre VARCHAR(15) NOT NULL,
apellidos VARCHAR(30) NOT NULL,
PRIMARY KEY (nif),
CONSTRAINT UNIQUE (nombre, apellidos)
) ENGINE = InnoDB;
```

```
CREATE TABLE Escritores (
ebook VARCHAR(13) NOT NULL,
nif VARCHAR(9) NOT NULL,
PRIMARY KEY (ebook, nif),
FOREIGN KEY (ebook) REFERENCES Ebook(isbn)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (nif) REFERENCES Autor(nif)
ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE = InnoDB;
```



```
CREATE TABLE Categoria (  
  nombre VARCHAR(25) NOT NULL,  
  PRIMARY KEY (nombre)  
) ENGINE = InnoDB;
```

```
CREATE TABLE Es_categorias (  
  document VARCHAR(13) NOT NULL,  
  categoria VARCHAR(25) NOT NULL,  
  PRIMARY KEY (document, categoria),  
  FOREIGN KEY (document) REFERENCES Document(isbn)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (categoria) REFERENCES Categoria(nombre)  
  ON DELETE NO ACTION ON UPDATE CASCADE  
) ENGINE = InnoDB;
```

```
CREATE TABLE Usuario (  
  usuario VARCHAR(15) NOT NULL,  
  password VARCHAR(15) NOT NULL,  
  nombre VARCHAR(15) NOT NULL,  
  apellidos VARCHAR(30) NOT NULL,  
  correo VARCHAR(50) NOT NULL,  
  sexo ENUM('Hombre', 'Mujer'),  
  fechaNacimiento DATE,  
  esAdministrador BOOLEAN NOT NULL,  
  PRIMARY KEY (usuario),  
  CONSTRAINT UNIQUE (correo)  
) ENGINE = InnoDB;
```

```
CREATE TABLE Valoracion (  
  id INTEGER NOT NULL AUTO_INCREMENT,  
  document VARCHAR(13) NOT NULL,  
  usuario VARCHAR(15) NOT NULL,  
  puntuacion ENUM('Muy_malo', 'Malo', 'Regular', 'Bueno', 'Muy_bueno') NOT NULL,  
  comentario VARCHAR(250),  
  PRIMARY KEY (id),  
  CONSTRAINT UNIQUE(document, usuario),  
  FOREIGN KEY (document) REFERENCES Document(isbn)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (usuario) REFERENCES Usuario(usuario)  
  ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE = InnoDB;
```

```
CREATE TABLE Descarga(  
  id INTEGER NOT NULL AUTO_INCREMENT,  
  fecha DATE NOT NULL,  
  usuario VARCHAR(15) NOT NULL,  
  document VARCHAR(13) NOT NULL,  
  importe INTEGER NOT NULL,  
  PRIMARY KEY (id),  
  CONSTRAINT UNIQUE(usuario, document),  
  FOREIGN KEY (usuario) REFERENCES Usuario(usuario)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (document) REFERENCES Document(isbn)  
  ON DELETE NO ACTION ON UPDATE CASCADE
```

```
)ENGINE=InnoDB;
```

```
CREATE TABLE Lectura(  
id INTEGER NOT NULL AUTO_INCREMENT,  
fecha DATE NOT NULL,  
usuario VARCHAR(15) NOT NULL,  
document VARCHAR(13) NOT NULL,  
pagina INTEGER NOT NULL,  
PRIMARY KEY (id),  
CONSTRAINT UNIQUE(usuario, document),  
FOREIGN KEY (usuario) REFERENCES Usuario(usuario)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (document) REFERENCES Document(isbn)  
ON DELETE NO ACTION ON UPDATE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE Publica(  
ebookAnonim VARCHAR(13) NOT NULL,  
usuario VARCHAR(15) NOT NULL,  
PRIMARY KEY (ebookAnonim, usuario),  
FOREIGN KEY (ebookAnonim) REFERENCES EbookAnonim(isbn)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (usuario) REFERENCES Usuario(usuario)  
ON DELETE CASCADE ON UPDATE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE Contacto(  
id INTEGER NOT NULL AUTO_INCREMENT,  
correo VARCHAR(50) NOT NULL,  
comentario VARCHAR(500),  
PRIMARY KEY (id)  
)ENGINE=InnoDB;
```

## 6. Conclusions

Aquest Treball Final de Carrera tenia l'objectiu d'ensenyar i donar una base de coneixement suficient per a desenvolupar aplicacions distribuïdes a nivell empresarial. Per a realitzar aquesta tasca, s'ha hagut d'estudiar totes les eines que s'han fet servir per desenvolupar l'aplicació de manera totalment autodidàctica. Gràcies als coneixements adquirits al llarg de tota la carrera d'informàtica, ha sigut una tasca relativament senzilla.

Aquest treball ha posat de manifest els coneixements adquirits al llarg de tota la carrera en la implementació, no només d'una aplicació, sinó de tot un projecte de magnitud mitja-gran, des de la planificació d'aquest fins l'obtenció del producte final.

En aquest cas el TFC és un treball molt beneficiós per a l'estudiant d'informàtica ja que posa a prova tots els coneixements però, el més important de tot és que serveix com a primera toma de contacte amb el que és un projecte informàtic real.

Vers els objectius que s'havien marcat al principi d'aquest projecte, s'han assolit tots. Tant a nivell d'assignatura, els objectius marcats al pla docent del Treball Final de Carrera, com els objectius concrets d'aquest treball, s'han assolit i es poden comprovar en aquesta memòria i en l'aplicació desenvolupada.

La tendència que s'està donant al món professional de la informàtica és migrar les aplicacions empresarials cap a la computació al núvol, és a dir, s'estan adoptant noves aplicacions distribuïdes gràcies al desenvolupament de servidors d'aplicacions. Per tant, aquest treball ha suposat una porta de coneixement cap a aquesta tendència i també una experiència molt valorable en aquest camp que obre una nova opció professional de futur.

## 7. Glossari

### **Administrador**

Persona que s'encarrega de la gestió del sistema

### **API**

Conjunt de funcions i procediment que ofereix una determinada biblioteca per ser emprat per altre software com una capa d'abstracció.

### **Bean**

Component de software reutilitzable. Disposen de una interfície pública per tal de instanciar-los i uns mètodes per accedir als atributs i insertar valors als atributs.

### **Comentar ebook**

Comentaris a mode d'opinió personal que fan els usuaris registrats dels ebooks.

### **Descarregar**

Obtenir un arxiu dels servidors del sistema.

### **ebook**

Libre electrònic

### **ebook anònim**

Libre electrònic en que l'autor es una persona anònima i no un autor conegut.

### **Framework**

Un marc de treball que disposa d'un conjunt de tècniques i eines estàndards per a resoldre un problema particular, orientat a agilitzar i facilitar el desenvolupament i reutilitzar el codi.

### **HQL (Hibernate Query Language)**

Llenguatge de consulta de dades que fa servir Hibernate.

### **IDE**

Entorn integrat de desenvolupament.

### **Identificar-se**

Donar les credencials davant el sistema.

### **Invitat**

Usuari que pot fer servir el sistema però no pot visualitzar llibres online ni descarregar-los.

### **J2EE (Java 2 Enterprise Edition)**

Plataforma per a la programació d'aplicacions distribuïdes sota Java.

### **JSP (JavaServer Page)**

Una tecnologia Java que permet generar contingut dinàmic per una web, usant documents HTML, XML o altres.

### **JSTL (JavaServer Page Standard Tag Library)**

Component de J2EE que estén les pàgines JSP proporcionant quatre biblioteques de tags amb utilitats àmpliament usades en el desenvolupament de pàgines web dinàmiques.

### **Oferta**

Rebaixa sobre el preu original d'un producte o promoció especial.

### **Online**

Sistema que funciona fent ús d'Internet.

### **Patró de disseny**

Plantilla de disseny que es fa servir com a solució a un problema conegut.

### **Publicar ebook**

Mitjà pel qual un usuari registrat pot publicar llibres escrits per ell.

### **Puntuar ebook**

Valoració que els usuaris donen a un llibre segons una escala de valors.

### **Registrar-se**

Complimentar un formulari amb dades personals per formar part del sistema.

### **Tag**

Etiqueta o marca d'un llenguatge basat en XML que defineix algun element.

### **Usuari registrat**

Usuari que pot fer servir el sistema en la seva totalitat.

### **Visualitza ebook**

Lectura d'un llibre electrònic online.

## 8. Bibliografía

Gavin King, Christian Bauer, Max Rydahl Andersen, Emmanuel Bernard, Steve Ebersole, y HardyFerentschik. *Documentación de referencia de Hibernate* [en línea]. Disponible en Web: <<http://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html/>>.

Gavin King, Christian Bauer, Max Rydahl Andersen, Emmanuel Bernard, Steve Ebersole, y HardyFerentschik. *Documentación de referencia de Hibernate (Capítulo 11.5. Modificación de objetos persistentes)* [en línea]. Disponible en Web: <<http://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html/objectstate.html#objectstate-modifying>>.

Gavin King, Christian Bauer, Max Rydahl Andersen, Emmanuel Bernard, Steve Ebersole, y HardyFerentschik. *Documentación de referencia de Hibernate (Capítulo 11.3. Cargando un objeto)* [en línea]. Disponible en Web: <<http://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html/objectstate.html#objectstate-loading>>.

ROSÉS ALBIOL, Francesc. *Introducción a Hibernate* [en línea]. Disponible en Web: <[http://www.javahispano.org/contenidos/es/introduccion\\_a\\_hibernate/](http://www.javahispano.org/contenidos/es/introduccion_a_hibernate/)>.

GONZÁLEZ DUQUE, Raúl. *Hibernate* [en línea]. Disponible en Web: <<http://mundogeek.net/archivos/2007/01/27/hibernate/>>.

FERNÁNDEZ, Oscar. *Ejemplo de aplicación web con Hibernate que funciona usando Netbeans IDE y JSTL (JSP(X))* [en línea]. Disponible en Web: <<http://www.ooscarr.com/nerd/elblog/2009/12/ejemplo-de-aplicacion-web-con-hibernate.php>>.

*Understanding Hibernate element* [en línea]. Disponible en Web: <<http://www.roseindia.net/hibernate/hibernateidgeneratorelement.shtml>>.

*Ejemplo sencillo con Hibernate* [en línea]. Disponible en Web: <[http://chuwiki.chuidiang.org/index.php?title=Ejemplo\\_sencillo\\_con\\_Hibernate](http://chuwiki.chuidiang.org/index.php?title=Ejemplo_sencillo_con_Hibernate)>.

PRIETO, Gustavo. *Manual básico de Struts* [en línea]. Disponible en Web: <[http://www.programacion.com/articulo/manual\\_basico\\_de\\_struts\\_156](http://www.programacion.com/articulo/manual_basico_de_struts_156)>.

GONZÁLEZ DUQUE, Raúl. *Etiquetas Struts 2* [en línea]. Disponible en Web:<<http://mundogeek.net/archivos/2009/02/13/etiquetas-struts-2/>>.

*Primeros pasos con Struts 2* [en línea]. Disponible en Web:<<http://es.debugmodeon.com/articulo/primeros-pasos-con-struts2>>.

SOLIS ZENTENO, Aldo A. *Struts – Hola Mundo* [en línea]. Disponible en Web:<<http://www.gandallasoft.com/2009/12/struts-hola-mundo.html>>.

*Struts File Upload and Save* [en línea]. Disponible en Web:<<http://www.roseindia.net/struts/strutsfileuploadandsave.shtml>>.

*Patrón DAO* [en línea]. Disponible en Web:<[http://chuwiki.chuidiang.org/index.php?title=Patr%C3%B3n\\_DAO](http://chuwiki.chuidiang.org/index.php?title=Patr%C3%B3n_DAO)>.

*web.xml Reference Guide for Tomcat* [en línea]. Disponible en Web:<<http://wiki.metawerx.net/wiki/Web.xml>>.

Javier Ceballos, Fco.. *Java 2: Interfaces gráficas y aplicaciones para Internet*. 3ª edición. RA-MA editorial, 2008. 694 p. ISBN:978-84-7897-859-5

Campderrich Falgueras, Benet. *Enginyeria del programari*. UOC. ISBN: 84-9788-065-X.

Sistac Planas, Jaume. *Bases de dades 1*. UOC. ISBN: 84-9788-299-7.

# Annex 1

## Manual d'instal·lació

Aquest manual d'instal·lació i l'aplicació desenvolupada s'han desenvolupat sobre el sistema operatiu Windows XP.

1. Instal·lar la JDK de Java que podem trobar a <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Un cop instal·lat, cal configurar les variables d'entorn de la següent manera:

- a. Polsa amb el botó dret del ratolí sobre "El Meu PC" i selecciona "Propietats".
  - b. Selecciona la pestanya "Opcions avançades" i polsa en el botó inferior "Variables d'entorn".
  - c. Ves a les variables del Sistema i crea la variable "JAVA\_HOME". Com a valor posa el directori a on es troba JAVA, per exemple: *C:\Archivos de programa\Java\jdk1.6.0\_24*
  - d. Busca la variable anomenada "PATH", fes clic sobre ella i polsa el botó "Modificar". En el quadre de valor de la variable, afegeix al final: `;%JAVA_HOME%\BIN` , i polsa "Acceptar" (si existeix tant com a variable de sistema i com d'usuari, fes el següent per a cada una d'elles).
  - e. Busca de nou en les Variables d'usuari i/o de Sistema, si existeix la variable "CLASSPATH". Si existeix, polsa el botó "Modificar". En el valor de variable, afegeix al final: `;"` (vigila: amb un PUNT al final) i polsa "Acceptar".
  - f. Tanca el quadre de diàleg de variables de entorn amb "Acceptar" i prem a "Aplica" per a les "Propietats del sistema".
2. Instal·lar el servidor d'aplicacions Apache Tomcat. Podem descarregar-lo des de la pagina web del projecte. (<http://tomcat.apache.org> ). Un cop instal·lat, cal configurar-lo de la manera següent:
    - a. Descomprimir el ZIP *apache-tomcat-6.0.29.zip* en la arrel del disc.
    - b. Crear una variable del sistema amb el nom CATALINA\_HOME i com a valor la carpeta a on esta instal·lat el Tomcat, per exemple: *C:\apache-tomcat-6.0.29*. Pots mirar la apartat 3 de la instal·lació del JDK per veure com es fa.
    - c. Buscar la variable "PATH", i afegir al final: `;%CATALINA_HOME%\BIN` (si existeix tant com a variable de sistema i com d'usuari, fes a les dues la mateixa operació).
  3. Instal·lar el gestor de base de dades, en aquest MySQL. Podem descarregar-lo des de la pagina web oficial. (<http://www.mysql.com/downloads/> )
  4. Un cop instal·lat el servidor i la base de dades, hem de copiar a la carpeta "lib" del directori d'instal·lació de l'Apache Tomcat el connector de la base de dades. Podem trobar la llibreria "mysql-connector-java-xxx-bin.jar" al directori que podem descarregar de la pagina web. (<http://www.mysql.com/downloads/connector/j/> )



5. Hem de copiar també l'arxiu "mysql-connector-java-xxx-bin.jar" al directori d'instal·lació de Java a "\$JAVA\_HOME\jdk\_xxx\jre\lib\ext". A més hem d'incloure l'arxiu a la variable d'entorn CLASSPATH afegint ";\$JAVA\_HOME\jdk1.6.0\_25\jre\lib\ext\mysql-connector-java-xxx-bin.jar".
6. Crear la base de dades amb l'script proporcionat. En l'arxiu podrem trobar dos arxius, l'script en format text per poder examinar-lo i el mateix script amb extensió ".sql" per executar-lo amb MySQL. Per executar l'script, hem d'obrir una finestra de "Simbolo del sistema" i executar la comanda "mysql -e "source *script*"" on script es el nom del fitxer, inclosa la ruta. Es pot trobar més informació a la documentació de MySQL (<http://dev.mysql.com/doc/refman/5.0/es/batch-mode.html>).
7. Hem de repetir el pas anterior, però aquesta vegada amb l'arxiu "inserts.sql". Aquest arxiu conté alguns inserts per la base de dades a mode d'exemple.
8. El següent pas es desplegar l'aplicació al servidor. L'aplicació s'adjunta com un fitxer war. Podem fer-ho des de el manager del tomcat, simplement sel·leccionant el war, o bé copiant el war i pegant-lo a la carpeta "webapps" del directori d'instal·lació de l'Apache, després hem d'iniciar el servidor. Per iniciar el servidor anem a la carpeta "bin" del directori de Apache i executem "startup".
9. Caldrà especificar el password de l'usuari root de la base de dades per a poder accedir-hi. L'arxiu on ho hem d'especificar es "hibernate.cfg.xml" i el podem trobar al directori "\$CATALINA\_HOME\webapps\tienda\_ebooks\WEB-INF\classes".
10. Ara ja podem executar l'aplicació. Reiniciem el servidor perquè els canvis surtin efecte i obrim un navegador i escrivim: "http://localhost:8080/tienda\_ebooks".