

# Anàlisi, disseny i implementació d'una aplicació fent servir l'arquitectura J2EE



**Salvador Lorca Sans**

Consultora: Verònica Peña Pastor

Feb06-Jun06

## **Dedicatòria i agraïments**

Aquest document és la culminació d'una etapa important a la meva vida. Per aquest motiu, vull agrair a la meva consultora Verònica per la seva comprensió, l'ajuda i el suport que m'ha proporcionat, i en extensió a tots els consultors, alumnes i el meu tutor Santi.

Gràcies a la meva família per tots els ànims i per creure en mi.

Als amics, vosaltres sou el motor de la meva vida.

Als companys de feina, per aguantar-me cada dia.

Finalment, dedico aquest treball al Marcos i la Marta, per ser tan bons amb mi i per donar-me forces per a arribar fins a aquest punt.

A tots ells, moltes gràcies.

# Índex de continguts

<b>ÍNDEX D'IL·LUSTRACIONS</b> .....	<b>4</b>
1.1. FINALITAT DEL TREBALL FINAL DE CARRERA .....	5
1.2. OBJECTIUS DEL TREBALL FINAL DE CARRERA .....	5
1.3. METODOLOGIA .....	5
1.4. PRODUCTE FINAL I RESULTATS ESPERATS .....	5
1.5. PLANIFICACIÓ DEL TREBALL FINAL DE CARRERA .....	6
1.5.1. Recursos .....	6
1.5.2. Requeriments de software .....	6
1.5.3. Riscos del projecte .....	6
1.5.4. Calendari .....	7
1.5.5. Fases del projecte .....	7
<b>2. L'ARQUITECTURA J2EE</b> .....	<b>10</b>
2.1. QUÈ ÉS J2EE? .....	10
2.2. COM S'IMPLEMENTA? .....	10
2.3. QUIN SERVIDOR D'APLICACIONS UTILITZAR? .....	11
2.4. APRECIACIÓ GLOBAL DE LA PLATAFORMA .....	11
2.4.1. Especificacions J2EE .....	11
2.4.2. Arquitectura de J2EE .....	11
<b>3. FASE D'ANÀLISI</b> .....	<b>13</b>
3.1. RECOLLIDA DE REQUISITS, ENTREVISTES .....	13
3.1.1. Introducció .....	13
3.1.2. Els requisits .....	13
3.1.3. Fonts d'informació .....	13
3.2. ESPECIFICACIÓ .....	13
3.2.1. Descripció .....	13
3.2.2. Procés de la Guia .....	13
3.2.3. Programa "Admin" .....	13
3.2.4. Programa "Viatger" .....	15
3.2.5. Resum esquemàtic .....	15
3.2.6. Recursos necessaris .....	15
3.2.7. Altres opcions .....	15
3.2.8. El model de negoci .....	15
3.2.9. Especificació de classes .....	23
<b>4. FASE DE DISSENY</b> .....	<b>25</b>
4.1. EL PAPER DEL DISSENY .....	25
4.2. LA REUTILITZACIÓ .....	25
4.2.1. Reutilització de classes .....	25
4.2.2. Reutilització de components .....	25
4.2.3. Els patrons .....	25
4.2.4. Els frameworks .....	26
4.3. EL DISSENY ARQUITECTÒNIC .....	26
4.3.1. Establir la configuració de la xarxa .....	26
4.3.2. Establir els subsistemes .....	26
4.4. DISSENY DELS CASOS D'ÚS .....	26
4.5. DIAGRAMA ESTÀTIC .....	33
4.6. DISSENY DE LA PERSISTÈNCIA .....	34
4.7. BASE DE DADES RELACIONAL .....	35
4.8. DISSENY DE LA INTERFÍCIE D'USUARI .....	36
<b>5. FASE D'IMPLEMENTACIÓ</b> .....	<b>49</b>
5.1. EXEMPLE DE CODIFICACIÓ .....	49
5.2. ESTRUCTURA D'ARXIU .....	53
5.3. SCRIPT DE CREACIÓ DE LA BASE DE DADES PER A MYSQL .....	54
<b>6. CONCLUSIONS</b> .....	<b>58</b>
<b>7. GLOSSARI</b> .....	<b>59</b>
<b>8. BIBLIOGRAFIA</b> .....	<b>60</b>
8.1. FONTS D'INFORMACIÓ .....	60
8.2. AUTORS .....	60

## Índex d'il·lustracions

<i>Il·lustració 1. Diagrama de temporització de Gantt.....</i>	<i>8</i>
<i>Il·lustració 2. Taula de tasques per setmanes.....</i>	<i>9</i>
<i>Il·lustració 3. Arquitectura de 3 capes de J2EE.....</i>	<i>10</i>
<i>Il·lustració 4. Diagrama de casos d'ús. Model de negoci.....</i>	<i>16</i>
<i>Il·lustració 5. Diagrama de casos d'ús (complet).....</i>	<i>18</i>
<i>Il·lustració 6. Diagrama de paquets.....</i>	<i>23</i>
<i>Il·lustració 7. Especificació de classes d'entitat.....</i>	<i>24</i>
<i>Il·lustració 8. Atributs de les classes d'entitat.....</i>	<i>24</i>
<i>Il·lustració 9. Diagrama de col·laboració. Recerca de destinacions.....</i>	<i>27</i>
<i>Il·lustració 10. Diagrama de col·laboració. Alta d'usuaris.....</i>	<i>27</i>
<i>Il·lustració 11. Diagrama de col·laboració. Baixa d'usuaris.....</i>	<i>28</i>
<i>Il·lustració 12. Diagrama de col·laboració. Modificació d'usuaris.....</i>	<i>28</i>
<i>Il·lustració 13. Diagrama de col·laboració. Alta de països.....</i>	<i>29</i>
<i>Il·lustració 14. Diagrama de col·laboració. Baixa de països.....</i>	<i>29</i>
<i>Il·lustració 15. Diagrama de col·laboració. Modificació de països.....</i>	<i>30</i>
<i>Il·lustració 16. Diagrama de col·laboració. Alta de zones.....</i>	<i>30</i>
<i>Il·lustració 17. Diagrama de col·laboració. Baixa de zones.....</i>	<i>31</i>
<i>Il·lustració 18. Diagrama de col·laboració. Modificació de zones.....</i>	<i>31</i>
<i>Il·lustració 19. Diagrama de col·laboració. Alta de destinacions.....</i>	<i>32</i>
<i>Il·lustració 20. Diagrama de col·laboració. Baixa de destinacions.....</i>	<i>32</i>
<i>Il·lustració 21. Diagrama de col·laboració. Modificació de destinacions.....</i>	<i>33</i>
<i>Il·lustració 22. Diagrama de col·laboració. Login.....</i>	<i>33</i>
<i>Il·lustració 23. Diagrama estàtic.....</i>	<i>34</i>
<i>Il·lustració 24. Diagrama Entitat - Relació.....</i>	<i>35</i>
<i>Il·lustració 25. Taules de la base de dades relacional.....</i>	<i>36</i>
<i>Il·lustració 26. Identificació de l'usuari. Pantalla Inicial.....</i>	<i>36</i>
<i>Il·lustració 27. Identificació de l'usuari. Login incorrecte.....</i>	<i>37</i>
<i>Il·lustració 28. Pantalla Principal.....</i>	<i>37</i>
<i>Il·lustració 29. Països. Llistat.....</i>	<i>38</i>
<i>Il·lustració 30. Països. Nou i Edició.....</i>	<i>38</i>
<i>Il·lustració 31. Països. Guardar país ja existent.....</i>	<i>39</i>
<i>Il·lustració 32. Països. Eliminar país.....</i>	<i>39</i>
<i>Il·lustració 33. Països. Eliminar país amb una o més zones.....</i>	<i>40</i>
<i>Il·lustració 34. Zones. Llistat.....</i>	<i>40</i>
<i>Il·lustració 35. Zones. Nova i Edició.....</i>	<i>41</i>
<i>Il·lustració 36. Zones. Guardar zona ja existent.....</i>	<i>41</i>
<i>Il·lustració 37. Zones. Eliminar zona.....</i>	<i>42</i>
<i>Il·lustració 38. Zones. Eliminar zona amb una o més destinacions.....</i>	<i>42</i>
<i>Il·lustració 39. Destinacions. Llistat.....</i>	<i>43</i>
<i>Il·lustració 40. Destinacions. Nova i Edició.....</i>	<i>44</i>
<i>Il·lustració 41. Destinacions. Guardar destinació ja existent.....</i>	<i>44</i>
<i>Il·lustració 42. Destinacions. Eliminar destinació.....</i>	<i>45</i>
<i>Il·lustració 43. Usuaris. Llistat.....</i>	<i>46</i>
<i>Il·lustració 44. Usuaris. Nou i Edició.....</i>	<i>46</i>
<i>Il·lustració 45. Usuaris. Guardar usuari ja existent.....</i>	<i>47</i>
<i>Il·lustració 46. Usuaris. Eliminar usuari.....</i>	<i>47</i>
<i>Il·lustració 47. Viatja. Recerca de destinacions.....</i>	<i>48</i>
<i>Il·lustració 48. Viatja. Visualització de destinació.....</i>	<i>48</i>
<i>Il·lustració 49. Exemple de codi de la implementació.....</i>	<i>52</i>
<i>Il·lustració 50. Estructura d'arxius.....</i>	<i>53</i>
<i>Il·lustració 51. Script de creació de la BBDD.....</i>	<i>57</i>

# 1. Introducció

## 1.1. Finalitat del Treball Final de Carrera

En aquest treball es pretén que l'alumne treballi en un projecte des del seu inici fins al seu final, passant per totes les fases de desenvolupament. En concret, treballaré en l'anàlisi, el disseny i la implementació d'una "Guia del viatger" fent servir l'arquitectura J2EE.

Aquesta "Guia del viatger" permetrà als usuaris consultar els principals llocs d'interès d'un país o **destinacions turístiques**, incloent un cercador filtrat per diverses opcions. De la mateixa forma, un administrador tindrà més privilegis sobre aquesta informació, com per exemple donar d'alta noves destinacions turístiques d'una manera clara i senzilla en un entorn web.

Per a poder realitzar el projecte és necessària l'adquisició de coneixements de Java, d'anàlisi i disseny orientat a objectes. És recomanable haver cursat les assignatures de programació de l'Enginyeria, així com tenir coneixements bàsics de l'arquitectura multicapa J2EE, de Bases de Dades relacionals i de llenguatge HTML.

## 1.2. Objectius del Treball Final de Carrera

L'objectiu general és l'anàlisi, el disseny i la implementació d'una "Guia del viatger". Es pretén separar clarament la presentació, la lògica de negoci i l'accés a les dades. Concretament, la presentació es preveu implementar amb la combinació de Servlets i JSP's, mentre que la lògica de negoci i la capa de dades amb Enterprise Java Beans (EJB's).

L'abast del projecte comprendria una funcionalitat general d'un cercador amb una sèrie de filtres de recerca, tenint en compte la limitació de temps imposada en la planificació. Es considera prioritària la facilitat i claredat del disseny enfront dels detalls d'implementació o l'estètica de la interfície d'usuari.

## 1.3. Metodologia

El desenvolupament del projecte seguirà de manera simplificada el cicle de vida del Rational Unified Process (RUP), amb les seves etapes: inici, elaboració, construcció i transició. A cada etapa hi haurà: recollida de requisits, anàlisi, disseny, implementació i proves.

A la fase d'anàlisi, el procés es guia per la identificació de les classes d'entitat, els seus atributs, la identificació de les relacions entre classes i l'elaboració de casos d'ús.

A la fase de disseny s'eliminen les herències entre classes, es dissenya la persistència i la interfície gràfica de l'usuari.

A la fase d'implementació es posen en pràctica la tecnologia J2EE i les pàgines JSP.

## 1.4. Producte final i resultats esperats

Al final del projecte s'obté una aplicació basada en l'arquitectura J2EE, que ens permet consultar dades sobre les diferents destinacions de cada país, així com administrar i observar fotografies de les platges, edificis públics, etc.

Amb l'ús del llenguatge Java, la UOC dona la possibilitat a l'estudiant de l'Enginyeria Tècnica de consultar i informar-se de les característiques d'un determinat país alhora de preparar el seu viatge. Utilitzant la tecnologia J2EE es cerca una solució integral que respongui a les necessitats actuals i futures dels investigadors, dissenyant-la de tal forma que interaccionin de forma transparent, o bé, que es deixin bases sòlides per a la implementació futura de solucions complementàries.

En aquest marc de referència, la plataforma d'aquest projecte informàtic es dissenya per a un funcionament multicapa, on a més del manejador de bases de dades i tenint com a capes intermèdies un servidor http i de Web Caché, es defineixen múltiples capes d'aplicació desenvolupades amb l'estàndard J2EE.

Altres avantatges són l'escalabilitat, la possibilitat de tenir la lògica de negoci independent dels clients, l'alta disponibilitat de les aplicacions i l'accés a les bases de dades, així com l'ús de petits clients i del factor de seguretat entre d'altres.

Entre els principals beneficis que es poden desencadenar amb l'aplicació de J2EE, trobam els següents:

- Enriquiment i robustesa de solucions.
- Generació de components reutilitzables dins l'àmbit dels distints sistemes del projecte, per tant es pot reduir fins a un 50% el temps de desenvolupament projectat.
- Detecció de mecanismes de seguiment integrals al projecte.
- Disminució de temps d'investigació i d'elaboració de prototips.
- Reducció de temps en la replicació del coneixement.

El projecte es realitza seguint la tecnologia J2EE, Java 2 Platform Enterprise Edition, que defineix l'estàndard per al desenvolupament d'aplicacions de múltiples capes.

J2EE simplifica les aplicacions empresarials basant-nos sobre components estandarditzats (modulars), proporcionant un joc complet de serveis, i maneja molts detalls de comportament d'aplicació automàticament, sense programació complexa.

## **1.5. Planificació del Treball Final de Carrera**

### **1.5.1. Recursos**

Els recursos humans es limiten a una persona, amb una dedicació d'unes 100 hores aproximadament (sense comptar la formació al client destinatari de l'aplicació). Aquest temps es pot allargar amb els caps de setmana si fos necessari.

### **1.5.2. Requeriments de software**

- JDK 5.0
- JBoss 4.0.4 GA, incloent el seu servidor web Tomcat Apache
- MySQL Server 5.0, com a sistema gestor de la base de dades
- IntelliJ IDEA 5.0.1, com a eina de desenvolupament en Java

### **1.5.3. Riscos del projecte**

Es consideren els riscos derivats de la inexperiència en implementació d'aplicacions J2EE. Concretament, aquesta falta d'experiència pot conduir a dissenys incorrectes, ineficaços o difícils d'implementar, així com ralentir la implementació. Per a mitigar aquests riscos serà necessari des del principi del projecte dedicar un temps adicional a la formació, mitjançant tutorials i llibres on-line. També s'espera compensar la inexperiència amb el seguiment de solucions proposades per altres, en forma de patrons.

#### 1.5.4. Calendari

El calendari anirà marcat per les dates dels lliuraments:

- 13 de març: planificació del projecte
- 3 d'abril: document de l'anàlisi
- 21 d'abril: document del disseny
- 29 de maig: document de la implementació
- 16 de juny: lliurament final, consistent en una memòria, una presentació en diapositives i el producte final (l'aplicació web)

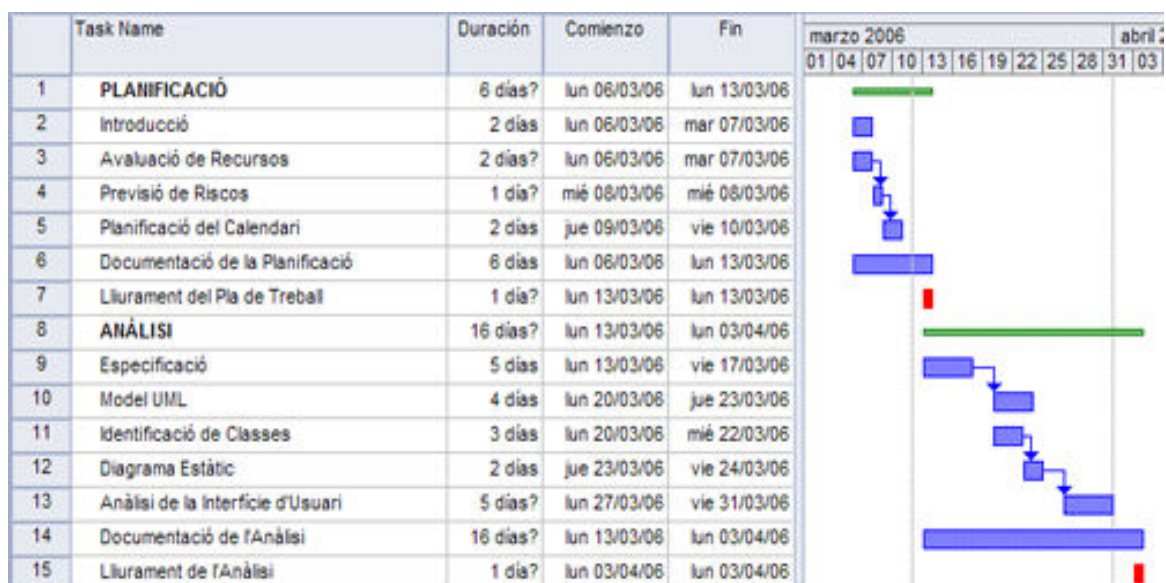
El calendari de les fases i tasques es detalla seguidament en un diagrama de Gantt.

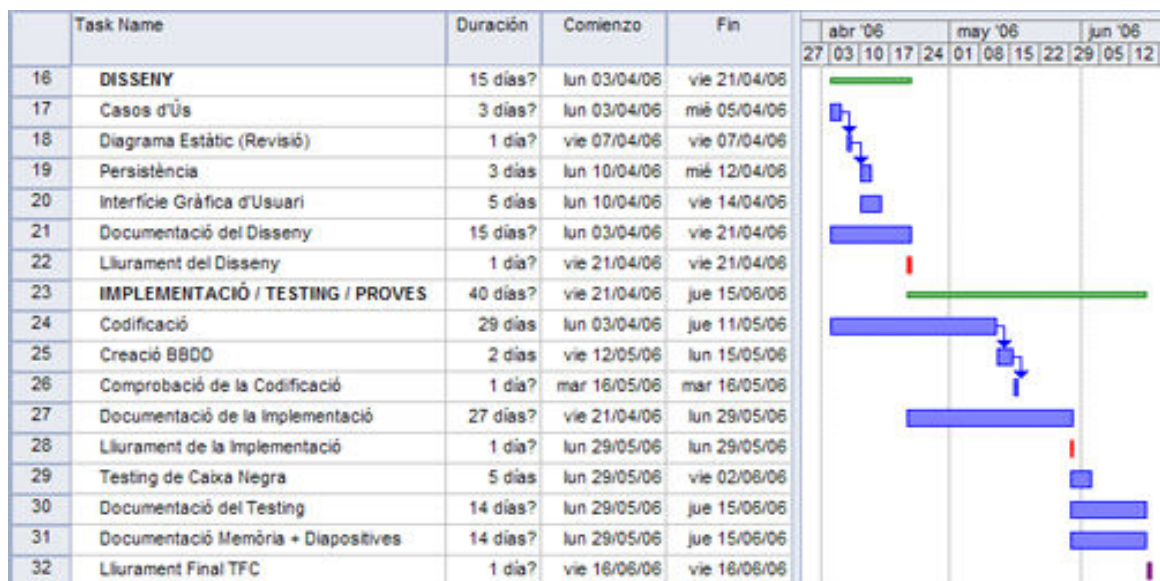
#### 1.5.5. Fases del projecte

El desenvolupament del projecte seguirà bàsicament el mètode RUP, degudament simplificat i considerant el context d'aquest projecte com un TFC. Les fases del projecte seran:

1. Planificació
2. Anàlisi
3. Disseny
4. Implementació
5. Proves

Els temps previstos per a cada tasca i la seva temporització és la que es veu representada en el diagrama de Gantt:





Il·lustració 1. Diagrama de temporització de Gantt

Setmana	Dates	Tasques
1	06/03/06 – 12/03/06	Introducció Avaluació de recursos Previsió de riscos Planificació del calendari Documentació de la planificació
2	13/03/06 – 19/03/06	Documentació de la planificació <b>Lliurament PAC 1 – Pla de Treball (13/03/06)</b> Especificació Documentació de l'anàlisi
3	20/03/06 – 26/03/06	Model UML Identificació de classes Diagrama estàtic Documentació de l'anàlisi
4	27/03/06 – 02/04/06	Anàlisi de la interfície d'usuari Documentació de l'anàlisi
5	03/04/06 – 09/04/06	Documentació de l'anàlisi <b>Lliurament Pràctica – Anàlisi (03/04/06)</b> Disseny de casos d'ús Diagrama estàtic (revisió) Documentació del disseny Codificació de components
6	10/04/06 – 16/04/06	Disseny de la persistència Disseny de la interfície gràfica d'usuari Documentació del disseny Codificació de components



7	17/04/06 – 23/04/06	Documentació del disseny Lliurament PAC 2 – Disseny (21/04/06) Codificació de components Documentació de la implementació
8	24/04/06 – 30/04/06	Codificació de components Documentació de la implementació
9	01/05/06 – 07/05/06	Codificació de components Documentació de la implementació
10	08/05/06 – 14/05/06	Codificació de components Creació de la BBDD Documentació de la implementació
11	15/05/06 – 21/05/06	Creació de la BBDD Comprobació de la codificació Documentació de la implementació
12	22/05/06 – 28/05/06	Documentació de la implementació
13	29/05/06 – 04/06/06	Documentació de la implementació Lliurament PAC 3 – Implementació (29/05/06) Testing de caixa negra Documentació del testing Documentació de la memòria + diapositives
14	05/06/06 – 11/06/06	Documentació del testing Documentació de la memòria + diapositives
15	12/06/06 – 18/06/06	Documentació del testing Documentació de la memòria + diapositives Lliurament final: TFC (16/06/06)

Il·lustració 2. Taula de tasques per setmanes

## 2. L'arquitectura J2EE

### 2.1. Què és J2EE?

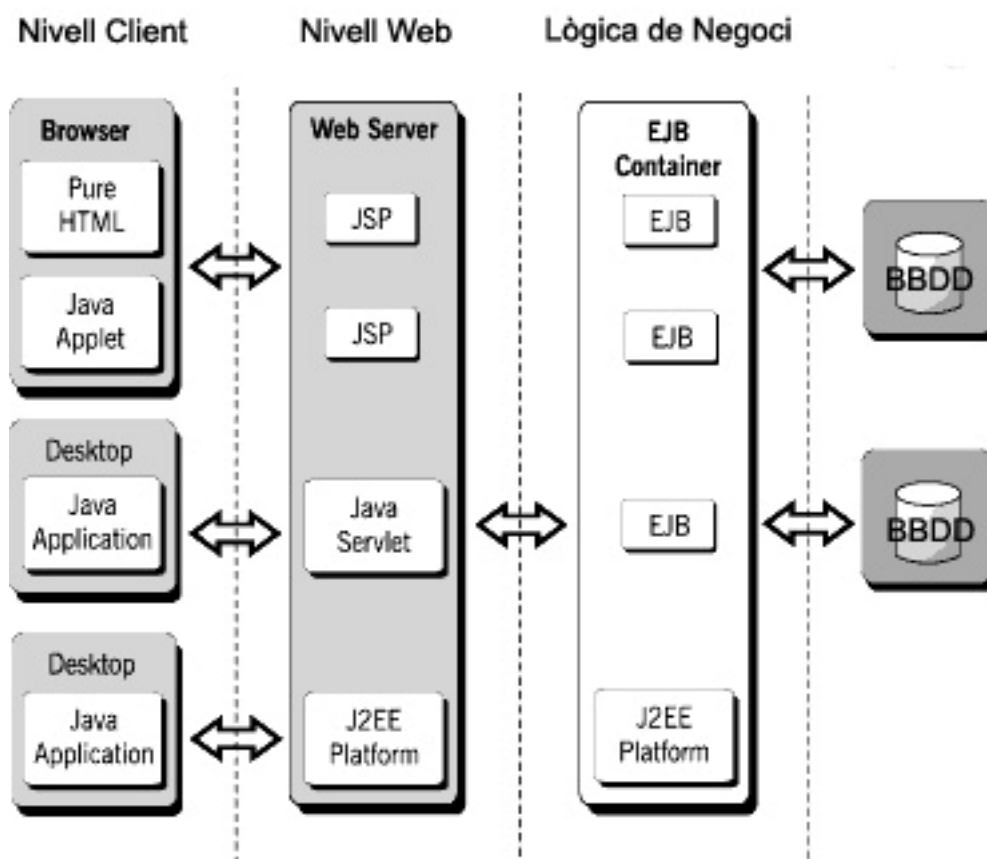
És una plataforma que habilita solucions per al desenvolupament, ús efectiu i maneig de multicapes en aplicacions centralitzades en el servidor.

J2EE utilitza la plataforma Java 2 SE, per a tendir una completa, estable, segura, i ràpida plataforma Java en l'àmbit de l'empresa. Permet estalviar a la companyia, perquè habilita una plataforma que redueix de manera significativa els costos i la complexitat de desenvolupament de solucions multicapa, resultant en serveis que poden ser desenvolupats ràpidament i ampliat fàcilment.

### 2.2. Com s'implementa?

Per a la implementació del codi s'ha utilitzat el llenguatge de programació Java, que permet la seva inclusió en el navegador mitjançant la tecnologia JSP. Així es garanteix la portabilitat entre diferents plataformes, sistemes operatius i navegadors.

L'arquitectura seguida és de 3 capes, basada en J2EE. Es preten separar clarament el Nivell Web i la Lògica de Negoci. Concretament, el Nivell Web es preten implementar amb la combinació de servlets de Java i JSP's (Java Server Pages). La Lògica de Negoci accedirà a la capa de dades amb els EJB's (Enterprise JavaBeans), tal i com es mostra en el següent esquema:



Il·lustració 3. Arquitectura de 3 capes de J2EE

Un **servlet** és un objecte que s'executa en un servidor o contenidor J2EE i està especialment dissenyat per a oferir contingut dinàmic JSP des d'un servidor web.

**JSP** és la tecnologia per a generar pàgines web de forma dinàmica en el servidor, desenvolupat per Sun Microsystems, basat en scripts que utilitzen una variant del llenguatge java.

La tecnologia **Enterprise JavaBeans (EJB)** és una arquitectura de components que permet el desenvolupament ràpid i simplificat d'aplicacions distribuïdes, transaccionals, segures i portables. Els EJB s'executen en el servidor de la plataforma Java 2 Platform, Enterprise Edition (J2EE).

La plataforma J2EE proveeix un nombre important de serveis més enllà de l'hostatjament de components, que inclou també suport per a XML, transaccions i seguretat.

### 2.3. Quin servidor d'aplicacions utilitzar?

**JBoss** és una implementació open source d'un "EJB Container". Mitjançant aquest tipus de producte és possible dur a terme un desenvolupament amb EJB's. Generalment no és distribuït com un producte individual i per aquesta raó se li pot considerar a "JBoss" com un producte diferent, quasi únic.

Com està basat en Java, JBoss pot ser utilitzat en qualsevol sistema operatiu que el suporti. El més important per al projecte és que JBoss implementa tot el paquet de serveis de J2EE.

### 2.4. Apreciació global de la plataforma

#### 2.4.1. Especificacions J2EE

Java 2, Enterprise Edition, aprofita moltes de les característiques de la plataforma Java, com la portabilitat "Write Once, Run Anywhere", l'*Application Program Interface (API)* JDBC per a l'accés a bases de dades, la tecnologia CORBA per a la interacció amb els recursos existents de l'empresa i un model de seguretat que protegeix les dades. Sobre aquesta base, Java 2 Enterprise Edition afageix el suport complet per als components Enterprise Java Beans, els API Java Servlets i la tecnologia JavaServer Pages. L'estàndard J2EE inclou totes les especificacions i proves de conformitat que permeten la portabilitat de les aplicacions a través de l'àmplia gamma de sistemes empresarials compatibles amb J2EE.

#### 2.4.2. Arquitectura de J2EE

J2EE està basat en l'arquitectura de la part del servidor (Served-based). Aquest tipus d'arquitectura concentra la majoria dels processos de l'aplicació en el servidor o en una part d'aquest. L'arquitectura té dues avantatges crítiques en comparació amb altres tipus, aquestes són:

- **Múltiples Clients:** Una arquitectura basada en el servidor requereix una clara separació entre la capa client (interfície) i la capa servidor, en la qual es realitzen els processos de l'aplicació. Això permet que una simple aplicació suporti simultàniament clients amb distints tipus de interfícies.

- Operacions robustes: Les aplicacions basades en el servidor poden ser dividides i distribuïdes en múltiples processadors. Components de l'aplicació poden ser replicats per a donar suport a caigudes instantàniament.

La plataforma de J2EE proveïx un conjunt d'APIs de java i serveis necessaris per al suport d'aplicacions empresarials. La plataforma completa pot ser implementada en un sol sistema, o la plataforma de serveis pot ser distribuïda mitjançant varis sistemes, però totes les APIs especificades han de ser incloses en alguna part del sistema complet.

## **3. Fase d'anàlisi**

### **3.1. Recollida de requisits, entrevistes**

#### **3.1.1. Introducció**

La recollida de requisits per part de l'alumne és fonamental per obtenir informació sobre dos aspectes:

1. Els processos que es tenen que realitzar sobre les dades
2. La forma com es tenen que demanar als usuaris les dades d'entrada i quina funció de l'aplicació desitgen utilitzar a cada moment.

#### **3.1.2. Els requisits**

Els requisits són l'especificació del que té que fer l'aplicació. Són descripcions del comportament, propietats i restriccions de l'aplicació que són necessàries desenvolupar.

Tenen un doble paper: serveixen de base per a un acord entre els usuaris (clients) i els desenvolupadors del software, i també són la informació de sortida per a poder desenvolupar correctament el programa.

#### **3.1.3. Fonts d'informació**

La principal font d'informació per a poder desenvolupar una aplicació són les entrevistes i qüestionaris als futurs usuaris. Una altra font d'informació podria ser una revisió de sistemes pareguts que hagi en el mercat (d'això se li diu *benchmarking*).

En aquest projecte, la principal font d'on trobaré la informació serà mitjançant pàgines d'Internet i un recull de preguntes a la meva consultora Verónica mitjançant el correu electrònic.

### **3.2. Especificació**

#### **3.2.1. Descripció**

El sistema d'informació "Guia del Viatger" pretén ser un sistema informàtic mitjançant el qual es puguin dur a terme consultes de destinacions turístiques en l'àmbit de la comunitat d'estudiants de la UOC. L'aplicació constarà d'un programa basat en l'entorn web i no farà falta que s'instal·li a cap ordinador (només al servidor).

#### **3.2.2. Procés de la Guia**

Per començar, l'administrador haurà d'introduir les dades dels països, zones i destinacions turístiques, així com els usuaris, i fer-ne un manteniment. Per això disposarà de la part del programa "Admin".

#### **3.2.3. Programa "Admin"**

El programa tindrà un seguit d'opcions per a fer el manteniment de les dades. Només els administradors podran accedir-hi:

- Administradors: Tenen un grau màxim de privilegi que els dóna accés a la informació, els permet d'entrar dades al sistema i poden modificar i/o borrar registres.

### Login de l'administrador

Aquesta opció servirà per a identificar-se al sistema com a administrador (nom del campus i contrasenya). Si no reconeguéu l'usuari o la contrasenya, no se li permetrà d'accedir a la pantalla de gestió de dades i se li demanarà que ho torni a intentar.

### Usuaris

Un llistat dels usuaris registrats al sistema és l'inici del subsistema d'usuaris. Des d'aquest llistat es podrà modificar i borrar un usuari. Per donar d'alta un usuari l'administrador ho farà des del menú lateral.

- Altes d'usuaris: Aquesta opció permetrà la introducció del nom d'usuari del campus. Si ja és al sistema, s'indicarà aquesta circumstància i es permetrà un nou intent. En cas que el nom d'usuari del campus no consti encara al sistema, es permetrà la introducció de les altres dades: password, nom, cognoms i perfil. Les dades es poden agafar del campus de la UOC.
- Baixes d'usuaris: Aquesta opció permetrà d'eliminar un usuari de la base de dades. Al llistat d'usuaris apareixerà un botó per eliminar un registre, i es sol·licitarà la confirmació de l'administrador per a esborrar-lo.
- Modificacions d'usuaris: Aquesta opció permetrà de modificar dades dels usuaris bé perquè s'hagi comés una errada en la introducció de dades, bé perquè han canviat les dades des del moment que es van introduir. Al llistat d'usuaris apareixerà un botó per editar un registre i es mostrarà la resta de dades de l'usuari que es poden modificar. Un cop fetes les modificacions caldrà prémer un botó per a fer efectius els canvis.

### Destinacions turístiques

Aquesta opció permetrà de fer les altes, baixes i modificacions de les destinacions turístiques.

- Altes de destinacions: Per a donar d'alta una destinació, cal que hi hagi com a mínim un continent, un país i una zona. Es sol·licitarà el nom de la destinació, una breu descripció, la zona i el fitxer de la imatge (en format *jpeg* o *gif*).
- Baixes de destinacions: Aquesta opció permetrà d'eliminar una destinació de la base de dades. Al llistat de destinacions apareixerà un botó per eliminar un registre, i es sol·licitarà la confirmació de l'administrador per a esborrar-lo.
- Modificacions de destinacions: Aquesta opció permetrà de modificar les destinacions que es desitgin. Al llistat de destinacions apareixerà un botó per editar un registre i es mostrarà la resta de dades de la destinació que es poden modificar. Un cop fetes les modificacions caldrà prémer un botó per a fer efectius els canvis.

## Països i Zones

El manteniment d'aquestes *entitats* es fa similarment. Serviran per a la catalogació de destinacions.

### **3.2.4. Programa "Viatger"**

El programa servirà per a identificar l'usuari a la *guia del viatger*. Consistirà en un login i una cerca de destinacions turístiques.

#### Login de l'usuari

Aquesta opció servirà per a identificar-se al sistema (nom del campus i contrasenya). Si no reconegué l'usuari o la contrasenya, no se li permetrà d'accedir al sistema i se li demanarà que ho torni a intentar.

#### Recerca de destinacions turístiques

Un cop acceptada la petició de *login*, al menú lateral del sistema es presentarà una opció per a accedir a la recerca de destinacions turístiques (*Viatja*) amb unes breus instruccions d'ús. L'usuari haurà de seleccionar un continent, un país i una zona disponibles en tres llistes. Al triar una combinació, es mostrarà una llista de destinacions que s'adeqüen a la recerca. Un cop premem sobre una destinació es visualitzarà la resta de dades informatives.

També es podrà cercar per paraula clau, com per exemple "catedral", juntament amb una selecció de continent, país o zona.

### **3.2.5. Resum esquemàtic**

1. Introducció dels usuaris
2. Introducció de les localitzacions\*
  - a. Països
  - b. Zones
3. Introducció de les destinacions turístiques
4. Recerca de destinacions turístiques

\* No cal introduir els **continents** perquè sempre seran Europa, Amèrica, Àsia, Àfrica i Oceania.

### **3.2.6. Recursos necessaris**

- Un servidor de bases de dades
- Un ordinador connectat a la xarxa internet

### **3.2.7. Altres opcions**

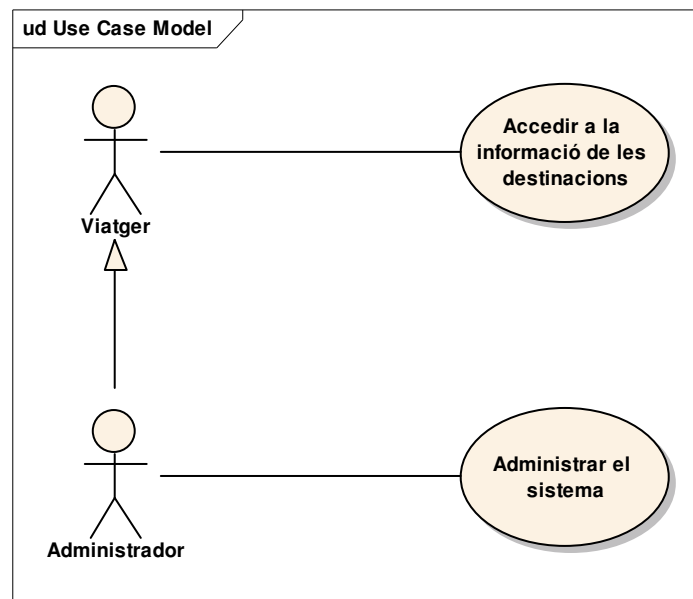
En aquest apartat es presenten les opcions que es creuen interessants per a afegir a l'aplicació, tot i que queden fora de l'abast del projecte a causa de les limitacions de temps per la falta d'experiència en aquest tipus d'aplicacions.

- Visualització ampliada de les imatges de les destinacions

### **3.2.8. El model de negoci**

El model del negoci descriu a grans trets els processos i entitats principals en l'entorn de l'aplicació, intervenint els casos d'ús i unitats de treball.

El diagrama de casos d'ús del model del negoci:



Il·lustració 4. Diagrama de casos d'ús. Model de negoci

### Els guions

Són les operacions més freqüents que els usuaris realitzen en el seu treball i resultaran útils per a identificar els casos d'ús.

- **Administrador:** Identificar-se al sistema, alta d'usuaris, baixa d'usuaris, modificació d'usuaris, alta de països, baixa de països, modificació de països, alta de zones, baixa de zones, modificació de zones, alta de destinacions, baixa de destinacions, modificació de destinacions i recerca de destinacions.
- **Viatger:** Identificar-se al sistema, consulta de destinacions.

### Identificació dels actors

Un actor és un paper d'alguna entitat externa que es preveu que interactuarà amb l'aplicació (obtenint o aportant informació). Es suposa que els actors no duen a terme cap seqüència de casos d'ús determinada. Cada actor correspon almenys a un usuari concret. No es necessari que es detallin els actors, basta amb identificar-los. Així mateix, no té sentit que dos actors intervinguin exactament en els mateixos casos d'ús.

Tipus d'actors:

1. Usuari final (viatger)
2. Usuari gestor del sistema (administrador)

En el nostre projecte tenim dos actors: administrador i viatger, podent l'administrador realitzar les mateixes funcions que el viatger, però no al contrari. Els dos actors són usuaris finals.

### Identificació dels casos d'ús

Els casos d'ús:



- Són processos autònoms iniciats per un actor o per un altre cas d'ús.
- Representen funcions ofertes per l'aplicació e identifiquen les entrades i les sortides.
- Descriuen el *qué* d'aquestes funcions, i no el *com*.
- Poden servir de base per a proves de caixa negra.
- Els casos que es descriuen per primer cop a una iteració determinada tenen que encaixar amb els de les iteracions anteriors.

Ara que ja entenem què signifiquen els casos d'ús, podem identificar els següents:

1. Login
2. Cercar destinacions turístiques
3. Alta d'usuaris
4. Baixa d'usuaris
5. Modificació d'usuaris
6. Alta de països
7. Baixa de països
8. Modificació de països
9. Alta de zones
10. Baixa de zones
11. Modificació de zones
12. Alta de destinacions
13. Baixa de destinacions
14. Modificació de destinacions

#### Identificació de les relacions entre casos d'ús

Tenim tres tipus de relacions entre casos d'ús:

- **Relacions d'extensió** (extend): Estan lligades a una condició quan hi ha diversos fluxos de processos possibles, o bé casos especials, o errors que han de ser tractats de manera diferent.

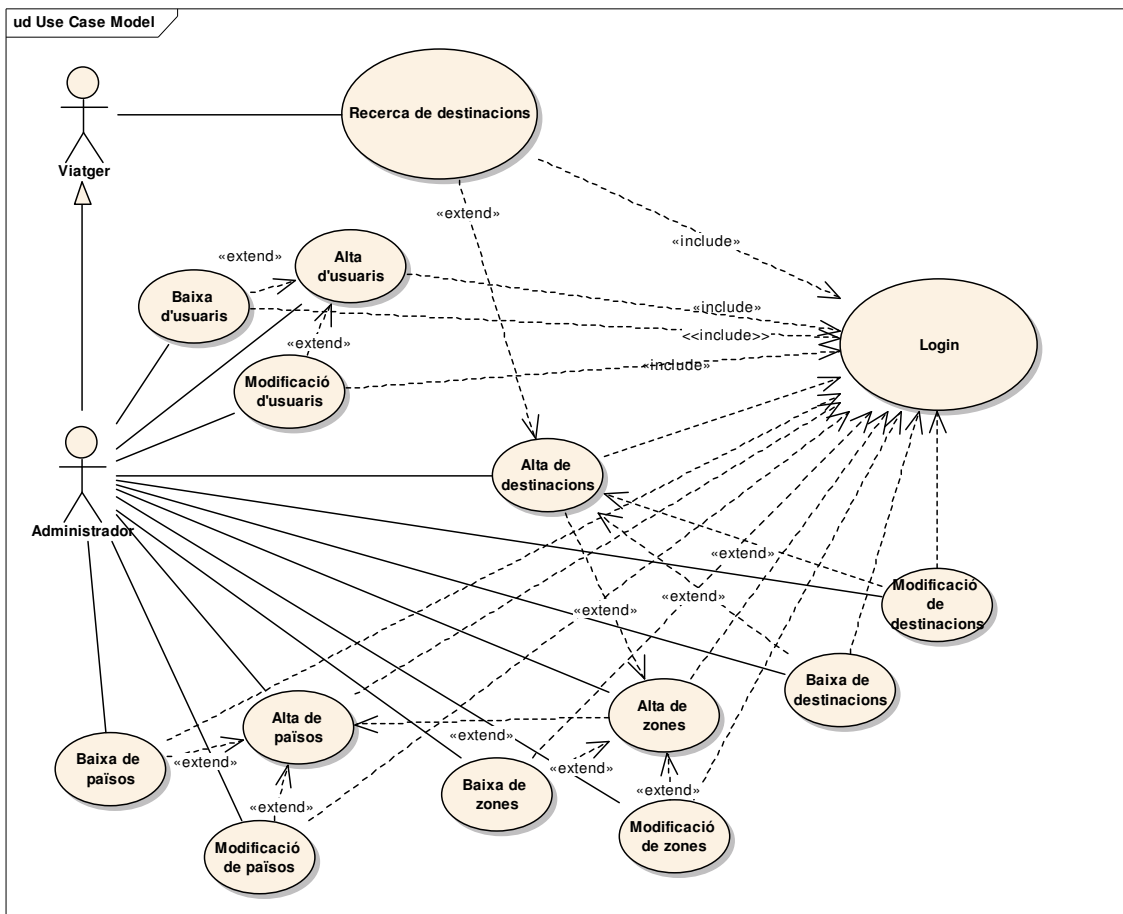
En el nostre cas tenim:

- Dos relacions d'extensió entre "Alta d'usuaris" i els casos d'ús "Baixa d'usuaris" i "Modificació d'usuaris", ja que si no hem donat d'alta un usuari no podem eliminar ni modificar cap usuari.
- Dos relacions d'extensió entre "Alta de països" i els casos d'ús "Baixa de països" i "Modificació de països", ja que si no hem donat d'alta un país no podem eliminar ni modificar cap país.
- Dos relacions d'extensió entre "Alta de zones" i els casos d'ús "Baixa de zones" i "Modificació de zones", ja que si no hem donat d'alta una zona no podem eliminar ni modificar cap zona.
- Dos relacions d'extensió entre "Alta de destinacions" i els casos d'ús "Baixa de destinacions" i "Modificació de destinacions", ja que si no hem donat d'alta una destinació turística no podem eliminar ni modificar cap destinació.
- Una relació d'extensió entre "Alta de zones" i "Alta de països". Abans d'afegir una zona ha d'existir un o més països.

- Una relació d'extensió entre "Alta de destinacions" i "Alta de zones". Abans d'afegir una destinació turística ha d'existir una o més zones.
  - Una relació d'extensió entre "Recerca de destinacions" i "Alta de destinacions". Abans de cercar una destinació turística ha d'existir una o més destinacions.
- **Relacions d'inclusió** (include): És un recurs per a evitar tenir que escriure una mateixa part del procés dins de diversos casos d'ús.

En el nostre cas tenim:

- Tantes relacions d'inclusió com casos d'ús derivats del procés de login, per a que els administradors i els viatgers puguin efectuar les seves operacions un cop identificats al sistema.
- **Relacions d'especialització:** Dels dos casos d'ús relacionats, un n'és una versió especialitzada de l'altre.



Il·lustració 5. Diagrama de casos d'ús (complet)

## Documentació textual dels casos d'ús

### Cas d'ús "Recerca de destinacions"

Resum funcionalitat general: Cerca una destinació turística a la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals del *Viatger* i ocasional de l'*Administrador*.

Actors: *Viatger* i *Administrador*.

Casos d'ús relacionats: Login, Alta de destinacions.

Precondició: Han d'existir destinacions i l'usuari ha d'estar registrat i identificat.

Postcondició: Cap, és una consulta.

Descripció: L'usuari utilitza el filtre de cerca i obté un llistat de les destinacions que concordin amb els paràmetres introduïts.

### Cas d'ús "Alta d'usuaris"

Resum funcionalitat general: L'*Administrador* podrà donar d'alta els usuaris a la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: *Administrador*.

Casos d'ús relacionats: Login.

Precondició: L'*administrador* té que estar registrat i identificat.

Postcondició: L'usuari queda donat d'alta al sistema.

Descripció: L'*administrador* entra a l'opció "Usuaris" i dins d'aquesta opció entra a "Crear Nou". Emplenarà les dades i les enregistrarà al sistema.

### Cas d'ús "Baixa d'usuaris"

Resum funcionalitat general: L'*Administrador* podrà donar de baixa usuaris de la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: *Administrador*.

Casos d'ús relacionats: Login, Alta d'usuaris.

Precondició: L'usuari a donar de baixa ha d'existir i l'*administrador* ha d'estar registrat i identificat al sistema.

Postcondició: L'usuari queda donat de baixa del sistema.

Descripció: L'*administrador* entra a l'opció "Usuaris" i dins d'aquesta opció pitjarà el botó "Eliminar" de l'usuari que vulgui esborrar.

### Cas d'ús "Modificació d'usuaris"

Resum funcionalitat general: L'*Administrador* podrà modificar les dades d'un usuari.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: *Administrador*.

Casos d'ús relacionats: Login, Alta d'usuaris.

Precondició: L'usuari ha d'existir i l'*administrador* ha d'estar registrat i identificat al sistema.

Postcondició: L'usuari queda modificat amb les seves noves dades.

Descripció: L'*administrador* entra a l'opció "Usuaris" i dins d'aquesta opció pitjarà el botó "Editar" de l'usuari que vulgui modificar. Emplenarà les noves dades i guardarà els canvis.

### Cas d'ús "Alta de països"

Resum funcionalitat general: L'Administrador podrà donar d'alta els països a la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'Administrador.

Actors: Administrador.

Casos d'ús relacionats: Login.

Precondició: El país no ha d'existir i l'administrador ha d'estar registrat i identificat al sistema.

Postcondició: El país queda donat d'alta al sistema.

Descripció: L'administrador entra a l'opció "Països" i dins d'aquesta opció entra a "Crear Nou". Emplenarà les dades i les enregistrarà al sistema.

### Cas d'ús "Baixa de països"

Resum funcionalitat general: L'Administrador podrà donar de baixa països de la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'Administrador.

Actors: Administrador.

Casos d'ús relacionats: Login, Alta de països.

Precondició: El país ha d'existir i l'administrador ha d'estar registrat i identificat al sistema.

Postcondició: El país queda donat de baixa del sistema.

Descripció: L'administrador entra a l'opció "Països" i dins d'aquesta opció pitjarà el botó "Eliminar" del país que vulgui esborrar.

### Cas d'ús "Modificació de països"

Resum funcionalitat general: L'Administrador podrà modificar les dades del país.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'Administrador.

Actors: Administrador.

Casos d'ús relacionats: Login, Alta de països.

Precondició: El país ha d'existir i l'administrador ha d'estar registrat i identificat al sistema.

Postcondició: El país queda modificat amb les seves noves dades.

Descripció: L'administrador entra a l'opció "Països" i dins d'aquesta opció pitjarà el botó "Editar" del país que vulgui modificar. Emplenarà les noves dades i guardarà els canvis.

### Cas d'ús "Alta de zones"

Resum funcionalitat general: L'Administrador podrà donar d'alta les zones a la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'Administrador.

Actors: Administrador.

Casos d'ús relacionats: Login, Alta de països.

Precondició: La zona no ha d'existir i l'administrador ha d'estar registrat i identificat al sistema.

Postcondició: La zona queda donada d'alta al sistema.

Descripció: L'administrador entra a l'opció "Zones" i dins d'aquesta opció entra a "Crear Nova". Emplenarà les dades i les enregistrarà al sistema.

### Cas d'ús "Baixa de zones"

Resum funcionalitat general: L'*Administrador* podrà donar de baixa zones de la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: *Administrador*.

Casos d'ús relacionats: Login, Alta de zones.

Precondició: La zona ha d'existir i l'*administrador* ha d'estar registrat i identificat al sistema.

Postcondició: La zona queda eliminada del sistema.

Descripció: L'*administrador* entra a l'opció "Zones" i dins d'aquesta opció pitjarà el botó "Eliminar" de la zona que vulgui esborrar.

### Cas d'ús "Modificació de zones"

Resum funcionalitat general: L'*Administrador* podrà modificar les dades de la zona.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: *Administrador*.

Casos d'ús relacionats: Login, Alta de zones.

Precondició: La zona ha d'existir i l'*administrador* ha d'estar registrat i identificat al sistema.

Postcondició: La zona queda modificada amb les seves noves dades.

Descripció: L'*administrador* entra a l'opció "Zones" i dins d'aquesta opció pitjarà el botó "Editar" de la zona que vulgui modificar. Emplenarà les noves dades i guardarà els canvis.

### Cas d'ús "Alta de destinacions"

Resum funcionalitat general: L'*Administrador* podrà donar d'alta les destinacions a la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: *Administrador*.

Casos d'ús relacionats: Login, Alta de zones.

Precondició: La destinació no ha d'existir i l'*administrador* ha d'estar registrat i identificat al sistema.

Postcondició: La destinació queda donada d'alta al sistema.

Descripció: L'*administrador* entra a l'opció "Destinacions" i dins d'aquesta opció entra a "Crear Nova". Emplenarà les dades i les enregistrarà al sistema.

### Cas d'ús "Baixa de destinacions"

Resum funcionalitat general: L'*Administrador* podrà donar de baixa destinacions de la base de dades.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: *Administrador*.

Casos d'ús relacionats: Login, Alta de destinacions.

Precondició: La destinació ha d'existir i l'*administrador* ha d'estar registrat i identificat al sistema.

Postcondició: La destinació queda eliminada del sistema.

Descripció: L'*administrador* entra a l'opció "Destinacions" i dins d'aquesta opció pitjarà el botó "Eliminar" de la destinació turística que vulgui esborrar.

#### Cas d'ús "Modificació de destinacions"

Resum funcionalitat general: L'Administrador podrà modificar les dades de la destinació.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'*Administrador*.

Actors: Administrador.

Casos d'ús relacionats: Login, Alta de destinacions.

Precondició: La destinació ha d'existir i l'*administrador* ha d'estar registrat i identificat al sistema.

Postcondició: La destinació queda modificada amb les seves noves dades.

Descripció: L'*administrador* entra a l'opció "Destinacions" i dins d'aquesta opció pitjarà el botó "Editar" de la destinació turística que vulgui modificar.

Emplenarà les noves dades i guardarà els canvis.

#### Cas d'ús "Login"

Resum funcionalitat general: Identificació dels usuaris registrats per a accedir al sistema.

Paper dins el treball de l'usuari: És un cas d'ús principal del *Viatger* i de l'*Administrador*.

Actors: *Viatger* i Administrador.

Casos d'ús relacionats: Tots els relacionats amb els actors *Viatger* i *Administrador*.

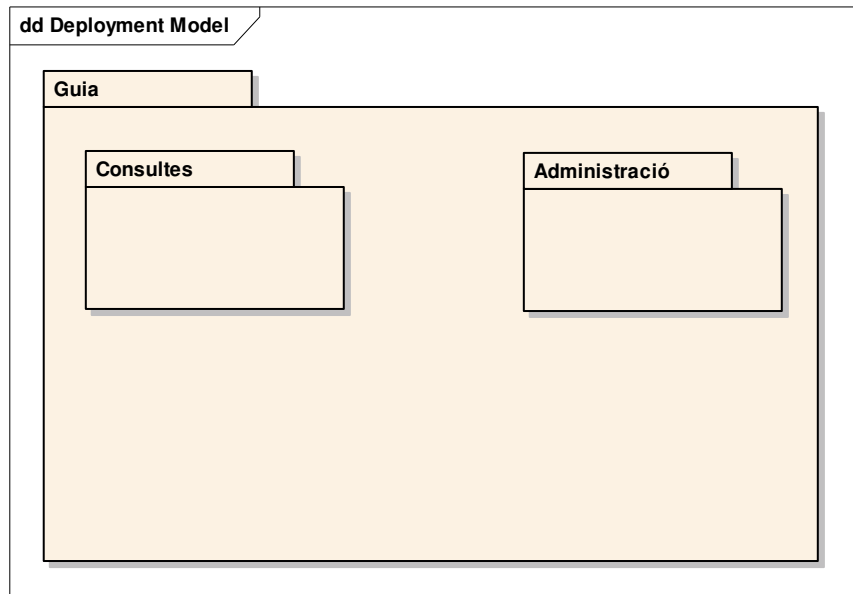
Precondició: Els usuaris han d'estar donats d'alta.

Postcondició: Els usuaris queden identificats en el sistema, podent accedir a les funcionalitats específiques de cadascun. Descripció: Un usuari introdueix el seu nom d'usuari del campus de la UOC i el password, i a continuació pitja el botó "Acceptar". El procés del sistema comprova si l'usuari ha introduït les dades correctament i en cas afirmatiu li permet l'entrada a la "Guia del *Viatger*" amb els privilegis que li corresponen.

#### Diagrama de paquets

Els paquets d'anàlisi corresponen a un o diversos subsistemes complets, mentre que els paquets de serveis són subdivisions dels paquets d'anàlisi des d'un punt de vista de client.

En el nostre cas tan sols es veu un únic paquet d'anàlisi "GV", que comprendrà tots els casos d'ús analitzats anteriorment. L'esquema és el següent:



Il·lustració 6. Diagrama de paquets

### 3.2.9. Especificació de classes

En aquest apartat es durà a terme l'especificació de les classes de l'anàlisi.

Hem de distingir tres tipus de classes d'anàlisi:

Classes de frontera: Els objectes de les classes frontera defineixen les interfícies amb els actors: finestres, diàlegs per pantalla i menús. Generalment seran pantalles de computador. S'utilitzen també per a cridar altres subsistemes. Un cas d'ús pot tenir diversos objectes frontera, i de vegades dos o més casos d'ús poden compartir un mateix objecte frontera. N'hi ha d'haver almenys un per a cada actor.

Classes d'entitat: Les classes d'entitat contenen els objectes bàsics que corresponen al món real. Normalment s'emmagatzemen a una base de dades (MySQL en el nostre projecte). Sovint són independents de l'aplicació. Generalment els objectes d'una d'aquestes classes no necessiten conèixer res uns dels altres. Es comuniquen a través dels objectes de les classes de control. Els objectes de les classes d'entitat només tenen coneixement de les coses que poden fer per a ells mateixos.

Classes de control: Cada cas d'ús ha de tenir almenys un objecte d'una classe de control que s'ocupi de la comunicació entre objectes. L'objecte de control serà el que s'ocuparà de dirigir els diferents camins del cas d'ús.

El modelat de les classes de control i de frontera s'acostuma a fer a la fase de disseny. Per tant, realitzarem la identificació de les classes d'entitat.

Classes d'entitat: Consisteix en identificar unes primeres classes mitjançant les quals es poden especificar els casos d'ús en forma d'interaccions. Moltes d'aquestes classes passaran al disseny i a la implementació. Cercam classes de conjunts d'objectes homogenis, que tenen uns mateixos atributs i operacions. Primer de tot, s'identificaran les classes d'entitat a partir dels casos d'ús. Per a cada cas s'indiquen les classes que es localitzen. Quan una classe ja s'havia identificat en un cas d'ús anterior, el seu nom anirà seguit

d'un asterisc, mentre que les classes que són dubtoses aniran seguides d'un interrogant.

Cas d'ús nº	Nom	Classes d'entitat
1	Recerca de destinacions	<i>Usuari, Administrador?, Destinacio, Viatger?</i>
2	Alta d'usuaris	<i>Usuari*</i>
3	Baixa d'usuaris	<i>Usuari*</i>
4	Modificació d'usuaris	<i>Usuari*</i>
5	Alta de països	<i>Pais, Continent</i>
6	Baixa de països	<i>Pais*, Continent*</i>
7	Modificació de països	<i>Pais*, Continent*</i>
8	Alta de zones	<i>Zona, Pais*</i>
9	Baixa de zones	<i>Zona*, Pais*</i>
10	Modificació de zones	<i>Zona*, Pais*</i>
11	Alta de destinacions	<i>Destinacio*, Zona*</i>
12	Baixa de destinacions	<i>Destinacio*, Zona*</i>
13	Modificació de destinacions	<i>Destinacio*, Zona*</i>
14	Login	<i>Usuari*, Administrador?*, Viatger?*</i>

Il·lustració 7. Especificació de classes d'entitat

Així doncs, la primera llista de classes d'entitat és: *Usuari, Destinacio, Continent, Pais, Zona*.

Hem descartat *Administrador* i *Viatger*, ja que són actors i no són rellevants. Tan sols la identificació de l'usuari serà significativa per a l'aplicació, per això basta amb una classe *Usuari* i ens evitarà errors futurs alhora d'escollir la categoria dels usuaris.

#### Atributs de les classes d'entitat

Els atributs de les classes d'entitat són generalment dades anomenades explícitament en els casos d'ús. Els usuaris els usen directament.

Classes d'entitat	Atributs
<i>Usuari</i>	nom_campus (string), password (string), nom (string), cognoms (string), email (string) i perfil (integer)
<i>Destinació</i>	id (integer), nom (string), descripcio (string), zona (integer), imatge (string)
<i>Continent</i>	id (integer), nom (string)
<i>Pais</i>	id (integer), nom (string), bandera (string), continent (integer)
<i>Zona</i>	id (integer), nom (string), pais (integer)

Il·lustració 8. Atributs de les classes d'entitat



## 4. Fase de disseny

### 4.1. El paper del disseny

La fase de disseny fa de pont entre l'anàlisi i la realització (implementació). Parteix del document d'especificació, considerant que aquest document satisfà els requeriments del programari que es desenvoluparà.

Definit al RUP (Rational Unified Process), el disseny es realitza al final de la fase d'elaboració i al principi d'una construcció. Per tant, és una etapa molt important i obligatòria.

### 4.2. La reutilització

La reutilització té quatre modalitats:

1. De classes
2. De components
3. Els patrons
4. Els *frameworks*

Òbviament, la reutilització comporta unes avantatges:

- Es redueix el desenvolupament de l'aplicació
- Disminueix el treball de manteniment
- Millora la fiabilitat
- El codi reutilitzat és més eficient
- Es guanya coherència
- Es preserva la inversió

#### 4.2.1. Reutilització de classes

Bàsicament es reutilitzaran les classes que tenen funcions independents del domini. La reutilització de classes del mateix domini tan sols és possible en sistemes de programari de la mateixa organització. Les classes que es reutilitzen es solen agrupar per funcions en llibreries, com els paquets de Java.

#### 4.2.2. Reutilització de components

Un component és un conjunt de classes, els objectes de les quals col·laboren en temps d'execució amb vistes a dur a terme una funció concreta.

Un component es comportarà com una classe, i per reutilitzar-lo correctament haurem de conèixer el contracte (que descriu els efectes de les operacions compreses a la interfície i les invariants que pot tenir).

#### 4.2.3. Els patrons

Són una manera organitzada de recollir l'experiència dels dissenyadors de programari per a tornar-la a utilitzar en casos semblants.

És una idea de disseny e implementació detallada i pràctica, com una recepta, que constitueix un esbós de solució d'un problema que es presenta amb una certa freqüència. Els detalls canvien per a cada cas. Els patrons tenen les següents característiques:

- Recullen experiència

- Són bastant amplis
- Creen vocabulari
- Són un instrument de documentació
- Si s'utilitza el mateix patró, facilitarà la coherència dels dissenys per a tot on es presenti la mateixa problemàtica.
- No donen la solució completa a un problema, sinó genèrica que és necessari completar.
- Ajuden a afrontar la complexitat del disseny.

#### **4.2.4. Els frameworks**

El *framework* és un conjunt de classes que constitueix una aplicació incompleta i genèrica. Si es materialitza de forma adequada, s'obtenen aplicacions especialitzades d'un cert tipus.

Si comparem els patrons amb els *frameworks*:

- Els patrons són més petits. Un *framework* pot contenir diversos patrons, mai al contrari.
- Els patrons són més abstractes. Cada vegada que s'aplica un mateix patró obtenim un programa diferent.
- Els patrons són menys especialitzats. Es poden utilitzar en qualsevol tipus d'aplicació.

### **4.3. El disseny arquitectònic**

Té com a objectiu definir les grans línies del model de disseny.

Com a activitats a seguir, podem distingir:

- Establir la configuració de la xarxa
- Decidir la utilització d'un framework ja disponible
- Establir els subsistemes

#### **4.3.1. Establir la configuració de la xarxa**

Consisteix en determinar els nodes que hi haurà, les característiques que tindran, les connexions entre aquests, els protocols de comunicació en quant a ample de banda, la fiabilitat i la qualitat ...

En el cas del projecte, es desenvoluparà com a mòdul del campus de la UOC.

#### **4.3.2. Establir els subsistemes**

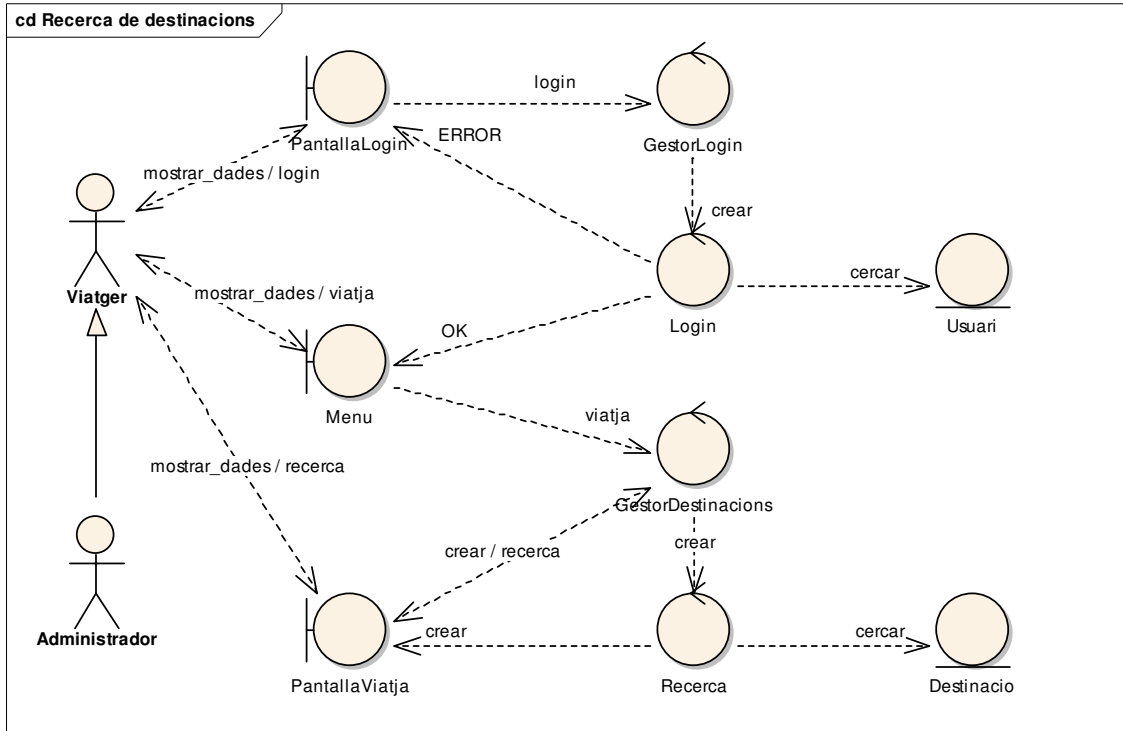
Els subsistemes poden ser propis del projecte, reutilitzats d'altres projectes, o bé, programes del mercat. Segons l'etapa prèvia d'anàlisi, tenim els subsistemes següents:

- Consultes
- Administració

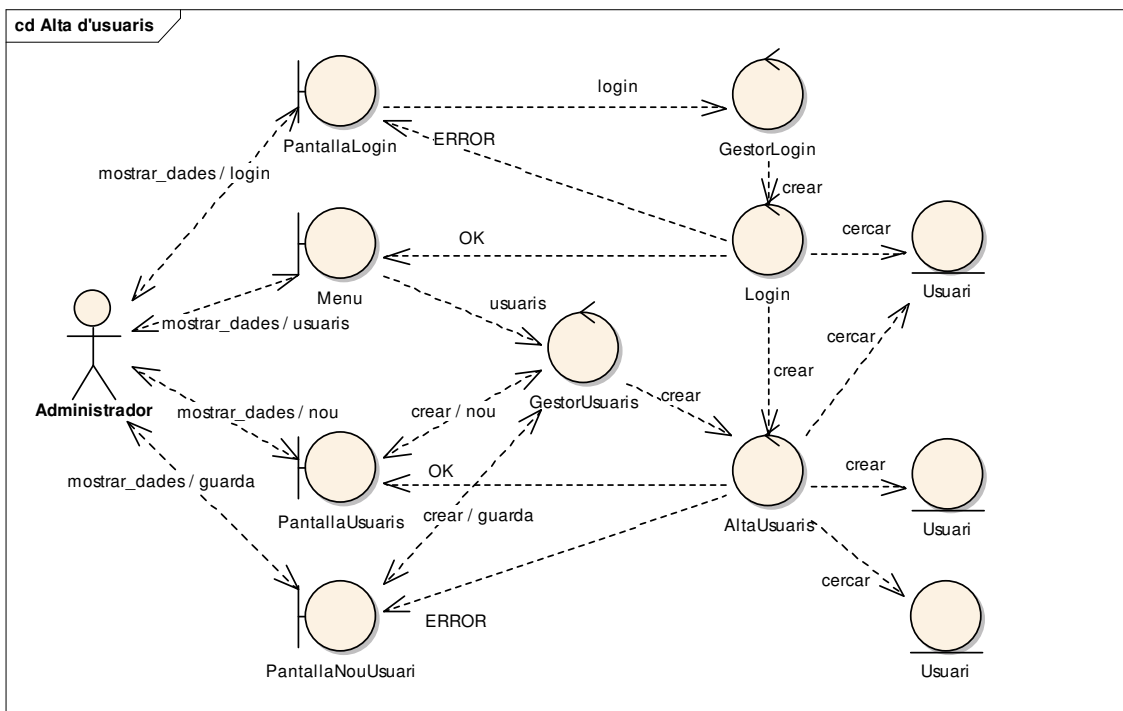
### **4.4. Disseny dels casos d'ús**

Parteix del diagrama de col·laboració, i es consideren per separat les classes de frontera, d'entitat i de control.

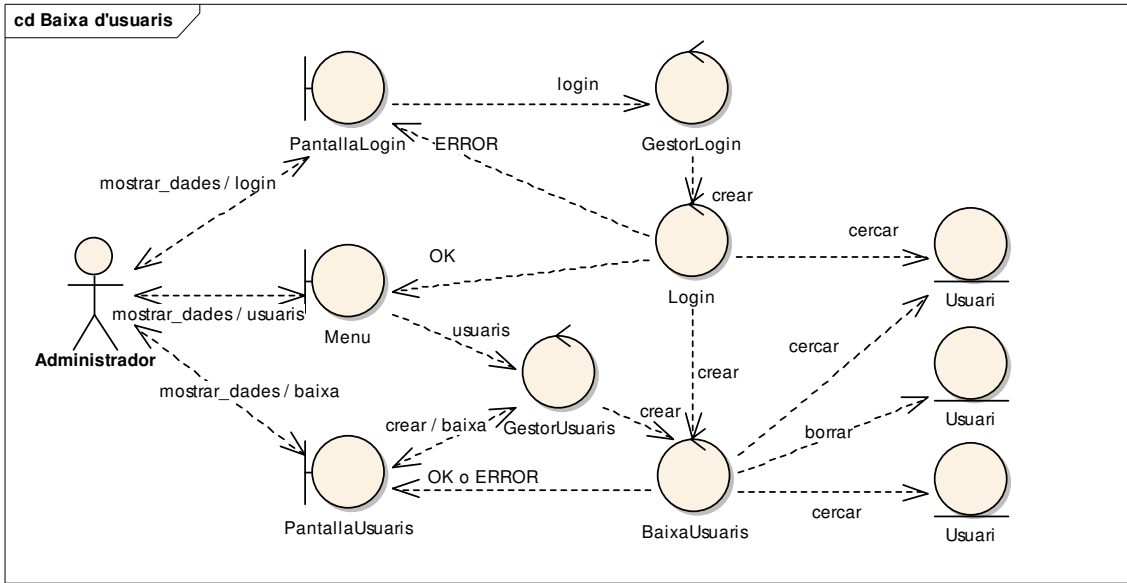
Una manera de cercar i documentar les classes frontera, control i les operacions dels tres tipus de classes és elaborar diagrames de col·laboració simplificats dels casos d'ús. Més endavant convé realitzar un diagrama estàtic, en el qual figurin totes les classes amb els atributs i operacions identificades, juntament amb les relacions.



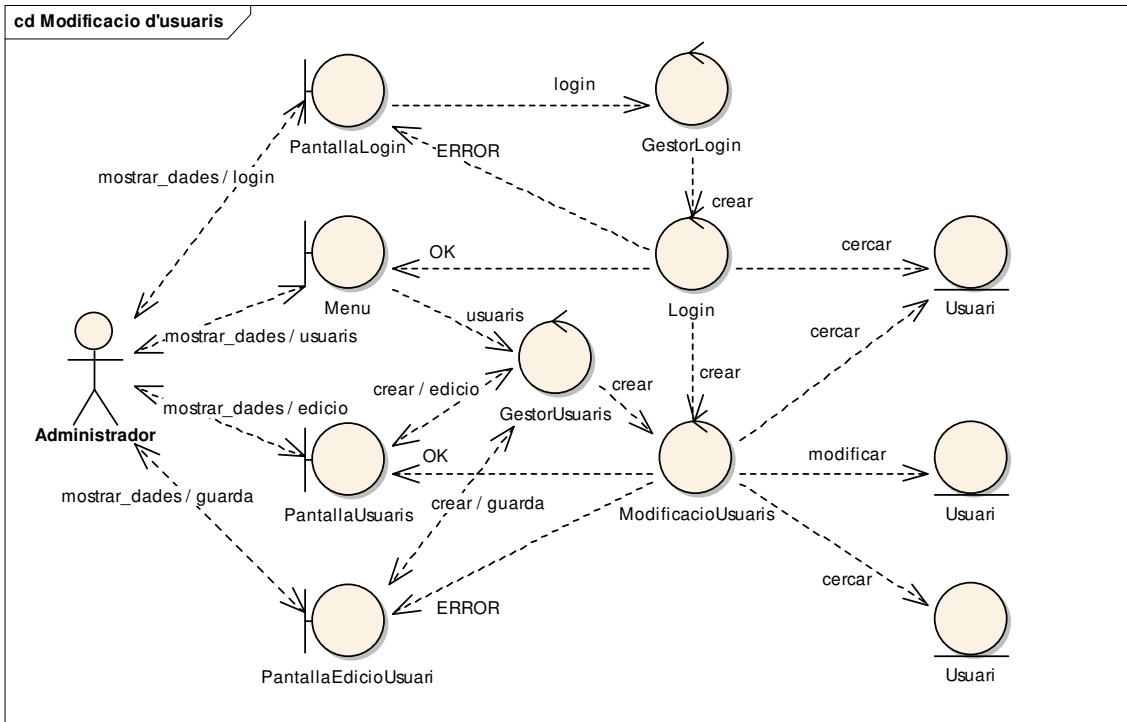
Il·lustració 9. Diagrama de col·laboració. Recerca de destinacions



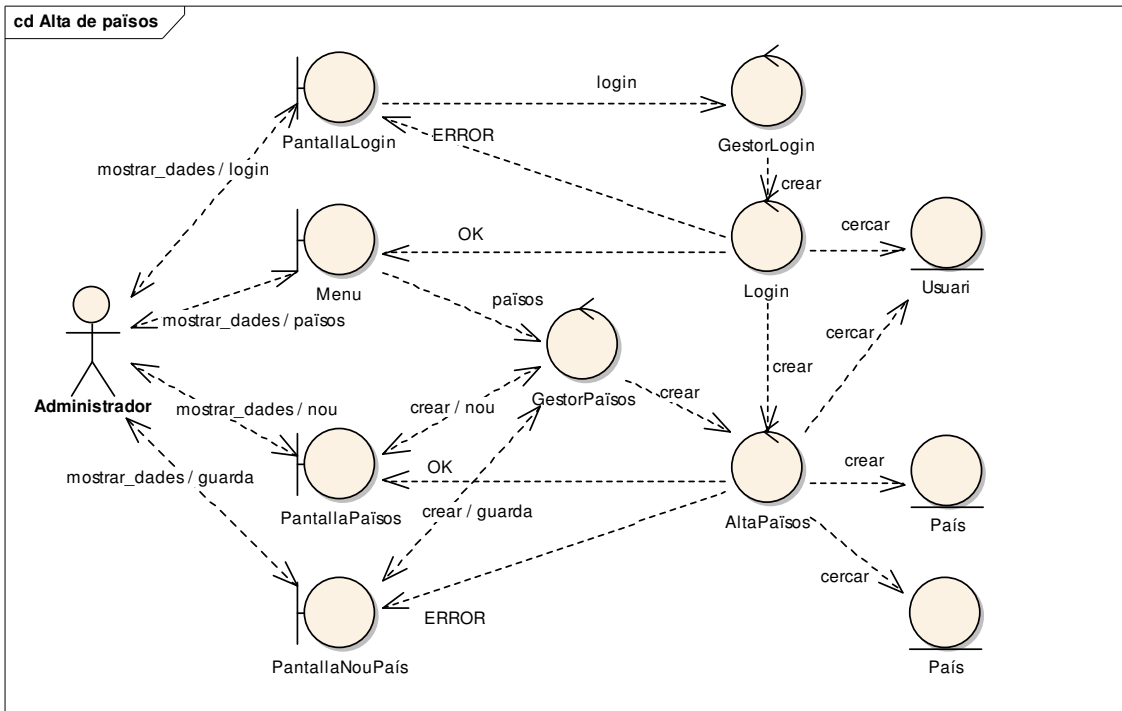
Il·lustració 10. Diagrama de col·laboració. Alta d'usuaris



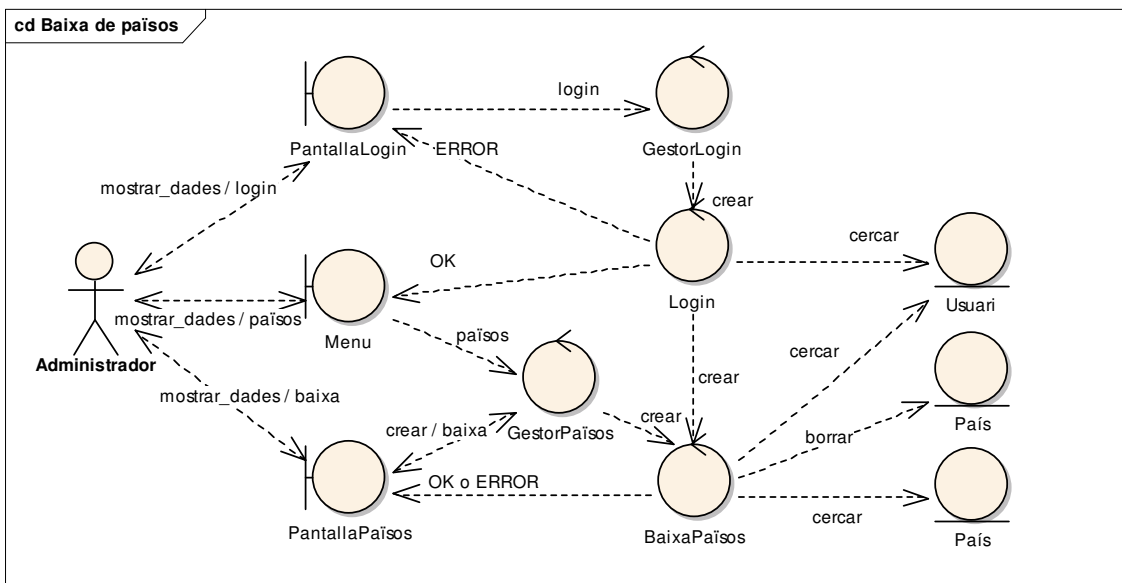
II·lustració 11. Diagrama de col·laboració. Baixa d'usuaris



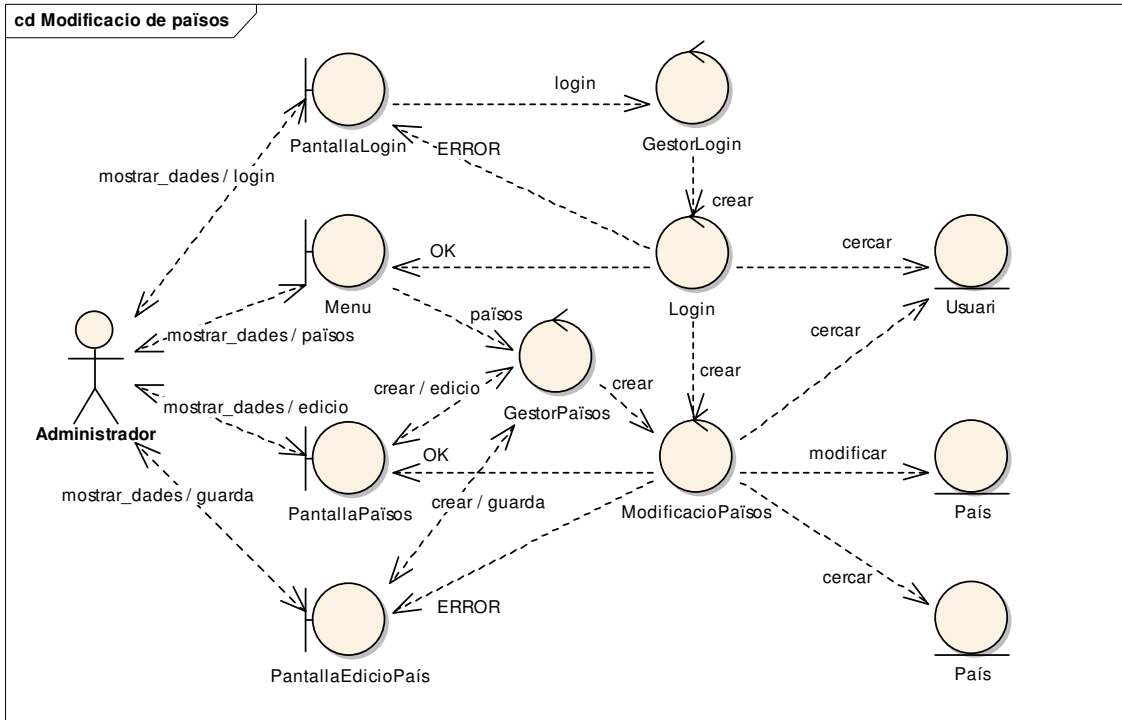
II·lustració 12. Diagrama de col·laboració. Modificació d'usuaris



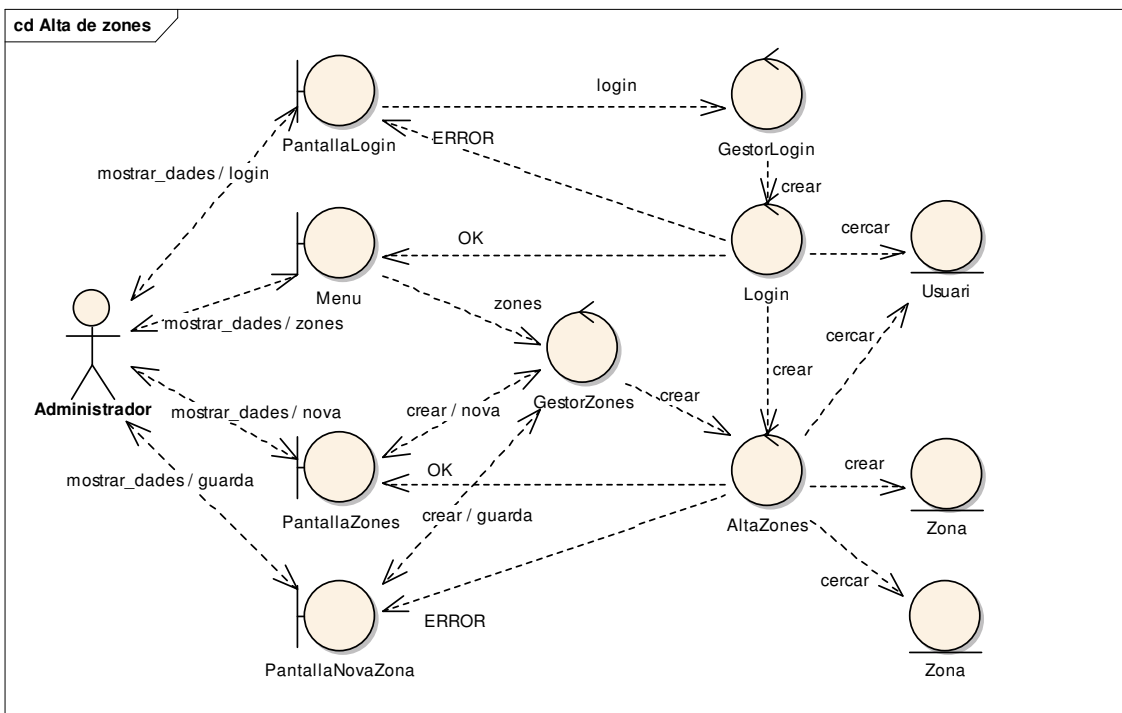
Il·lustració 13. Diagrama de col·laboració. Alta de països



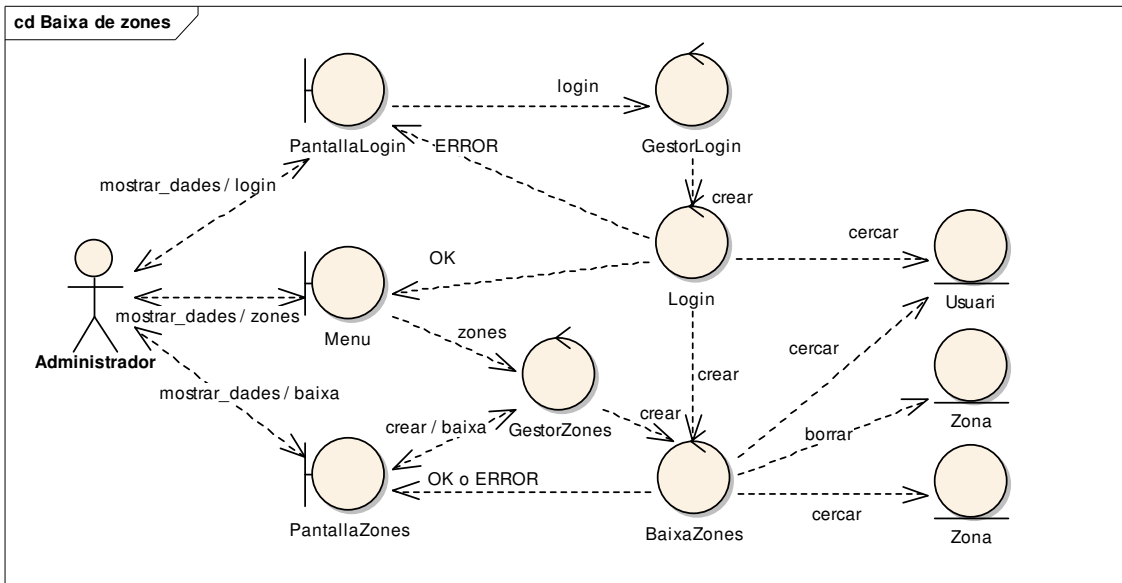
Il·lustració 14. Diagrama de col·laboració. Baixa de països



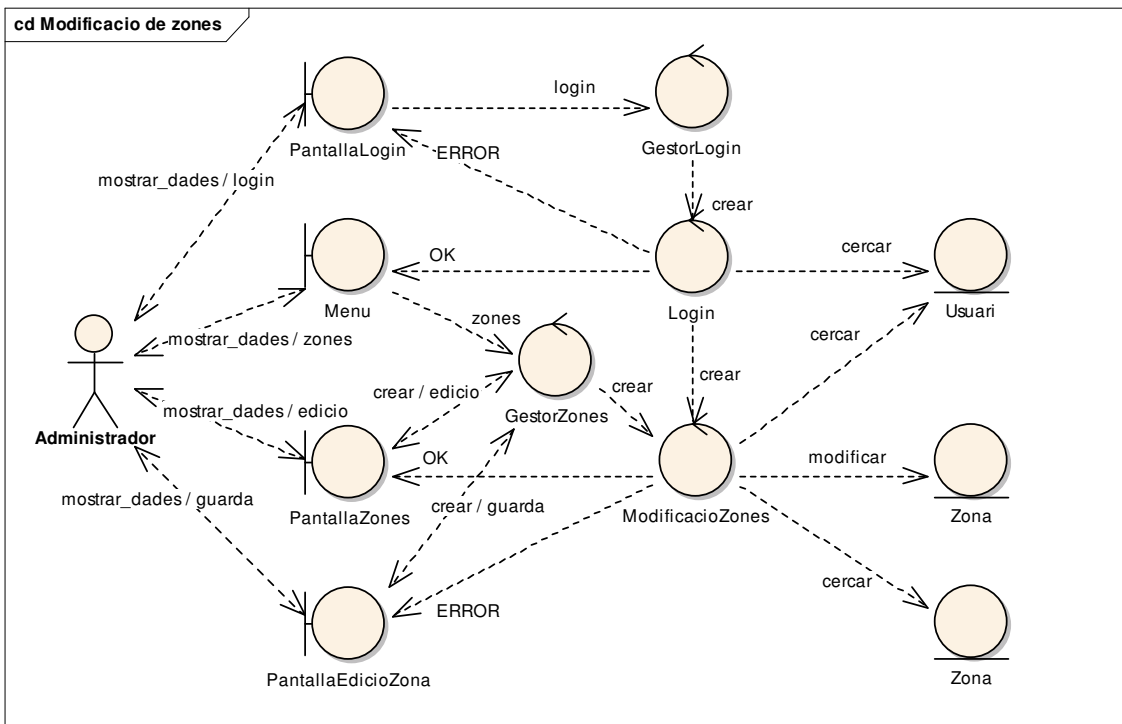
II·lustració 15. Diagrama de col·laboració. Modificació de països



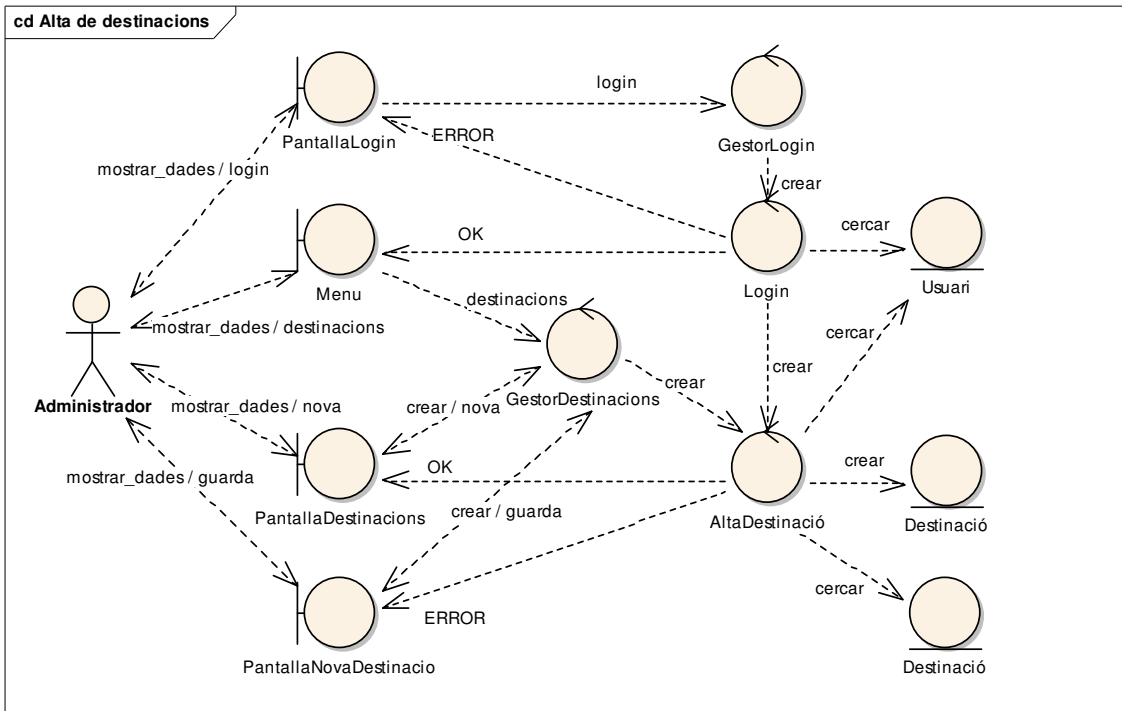
II·lustració 16. Diagrama de col·laboració. Alta de zones



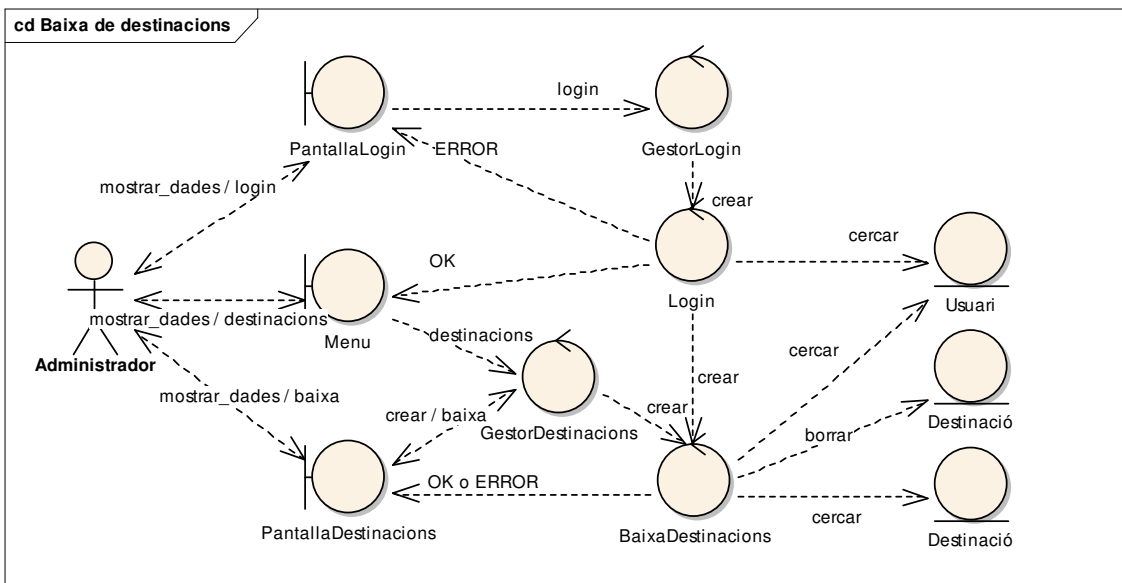
Il·lustració 17. Diagrama de col·laboració. Baixa de zones



Il·lustració 18. Diagrama de col·laboració. Modificació de zones

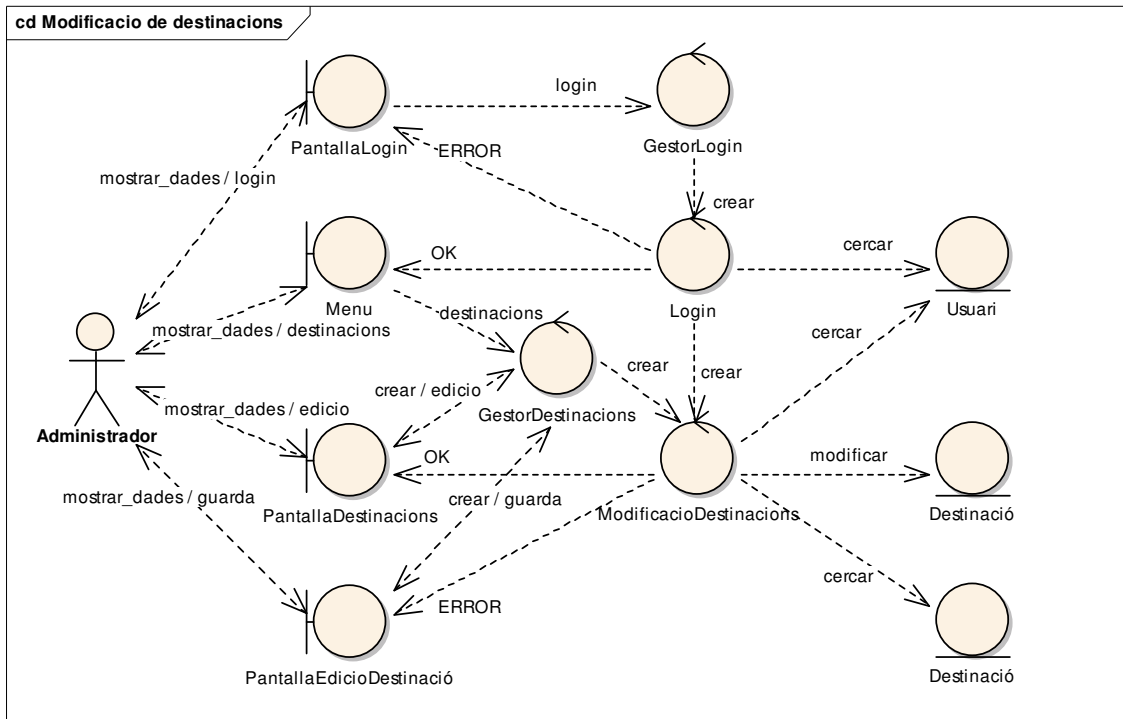


Il·lustració 19. Diagrama de col·laboració. Alta de destinacions

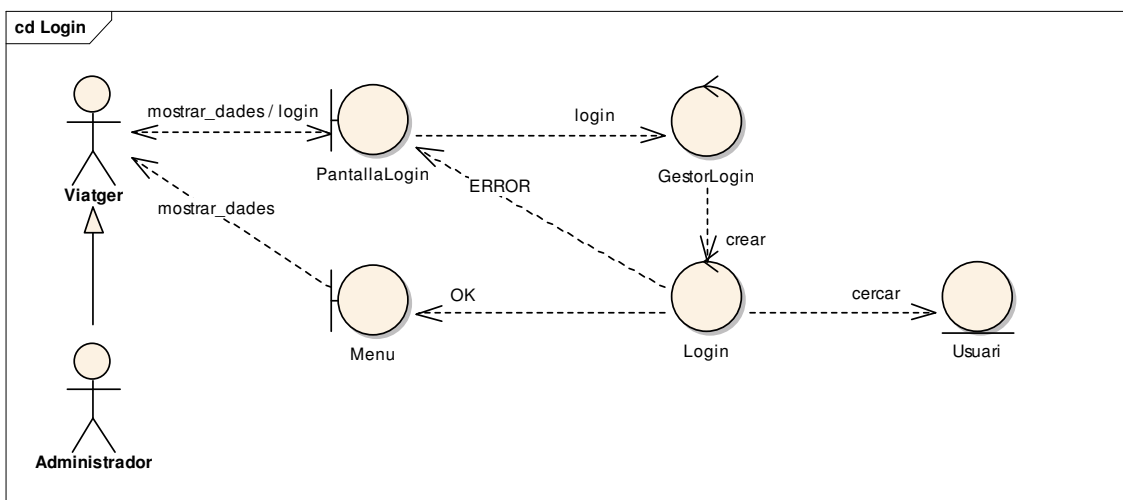


Il·lustració 20. Diagrama de col·laboració. Baixa de destinacions





II·lustració 21. Diagrama de col·laboració. Modificació de destinacions



II·lustració 22. Diagrama de col·laboració. Login

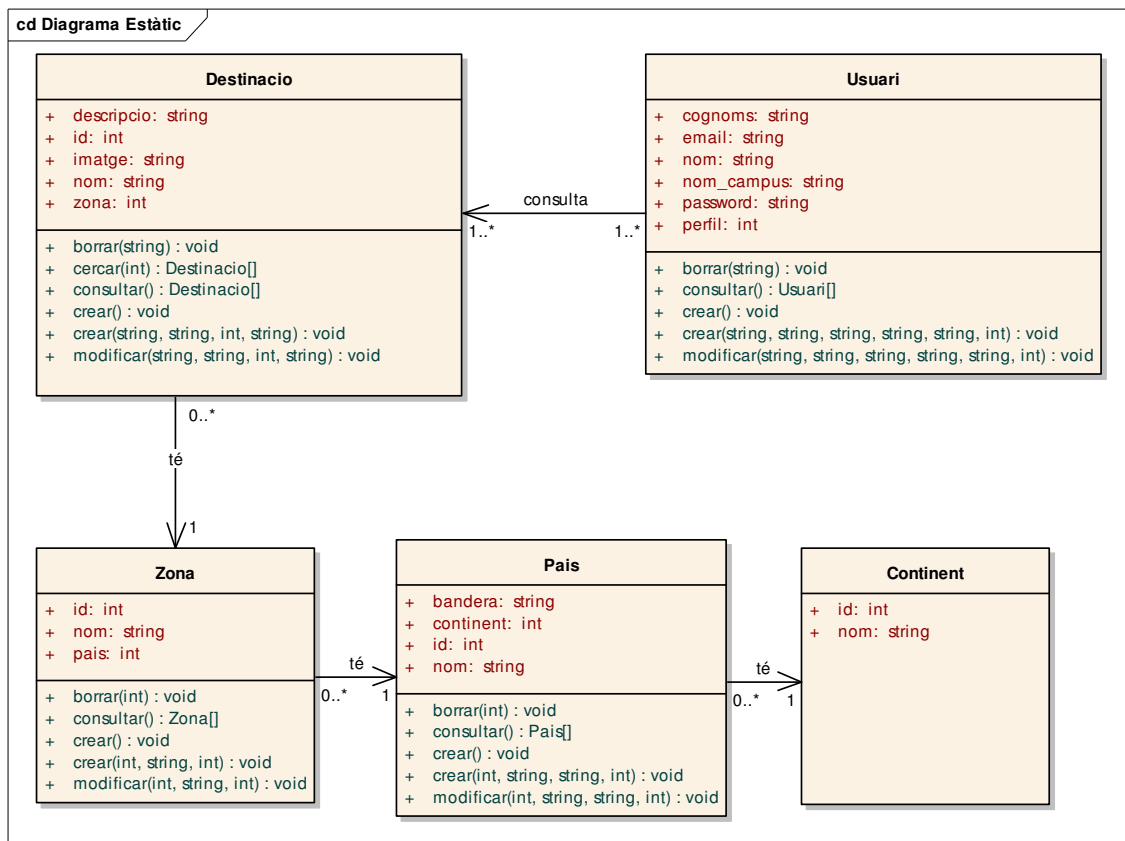
#### 4.5. Diagrama estàtic

El diagrama estàtic del disseny té en compte els següents aspectes:

- La normalització dels noms
- Reutilització de les classes
- Adaptació de l'herència
- La millora del rendiment
- L'increment de la velocitat
- Reducció del trànsit de missatges
- Classes temporals

A les classes del diagrama estàtic s'indiquen només aquells atributs que s'han afegit a la versió d'anàlisi, i les operacions s'indiquen totes.

Com que no hem detectat cap herència, no es pot eliminar.



Il·lustració 23. Diagrama estàtic

#### 4.6. Disseny de la persistència

Per a la persistència del projecte utilitzarem una base de dades relacional, concretament MySQL.

Per a l'elaboració del disseny de la persistència, primer de tot hem d'eliminar l'herència que hi pugui haver (en aquest cas no és necessari ja que és inexistent).

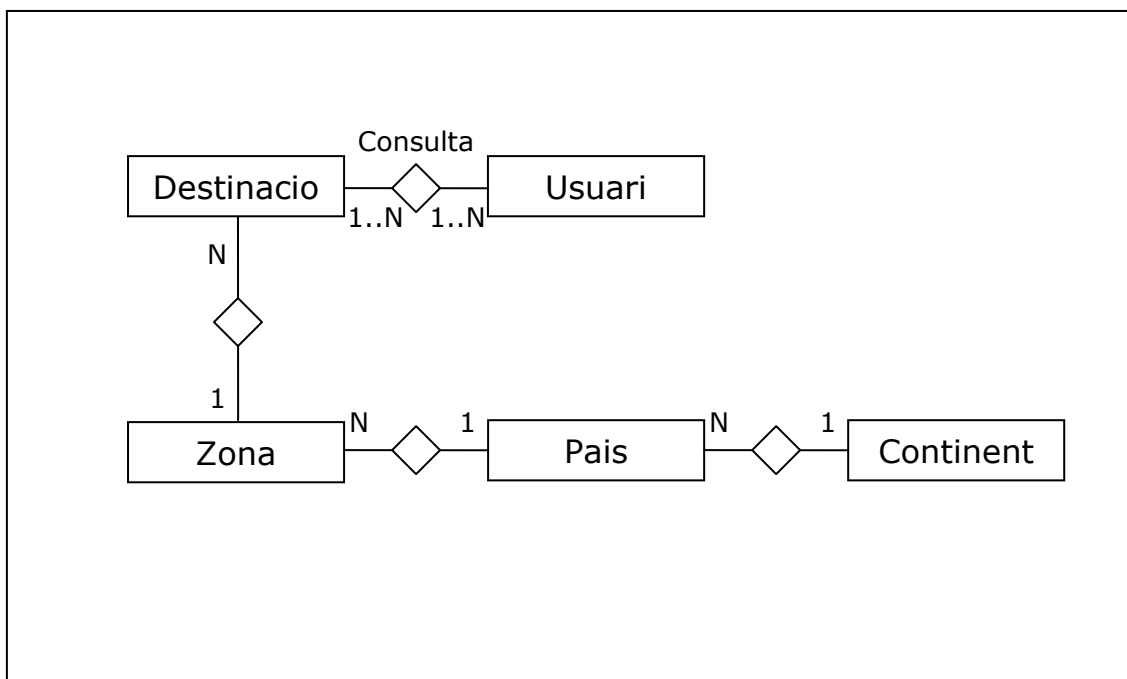
Tot seguit, hem d'elaborar el diagrama Entitat – Relació equivalent al diagrama estàtic de les classes persistents, i finalment obtindrem l'especificació de la base de dades relacional.

Per a efectuar aquest procés, determinam les classes que han de ser persistents:

Classes persistents: *Usuari, Destinacio, Zona, Pais, Continent.*

Classes temporals: *Perfil.*

El diagrama Entitat – Relació que correspon a aquest projecte és el següent:



Il·lustració 24. Diagrama Entitat - Relació

#### 4.7. Base de dades relacional

A continuació es descriuen totes les taules que intervindran en la "Guia del Viatger", amb els seus atributs, tipus i si són clau primària o forana.

Taula Usuari	Tipus	Propietat
nom_campus	string	Clau primària
password	string	
nom	string	
cognoms	string	
e-mail	string	
perfil	int	

Taula Destinacio	Tipus	Propietat
id	int	Clau primària
nom	string	
descripcio	string	
zona	int	Clau forana
imatge	string	

Taula Zona	Tipus	Propietat
id	int	Clau primària
nom	string	
pais	int	Clau forana

Taula Pais	Tipus	Propietat
id	int	Clau primària
nom	string	
bandera	string	
continent	int	Clau forana

Taula Continent	Tipus	Propietat
id	int	Clau primària
nom	string	

Il·lustració 25. Taules de la base de dades relacional

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

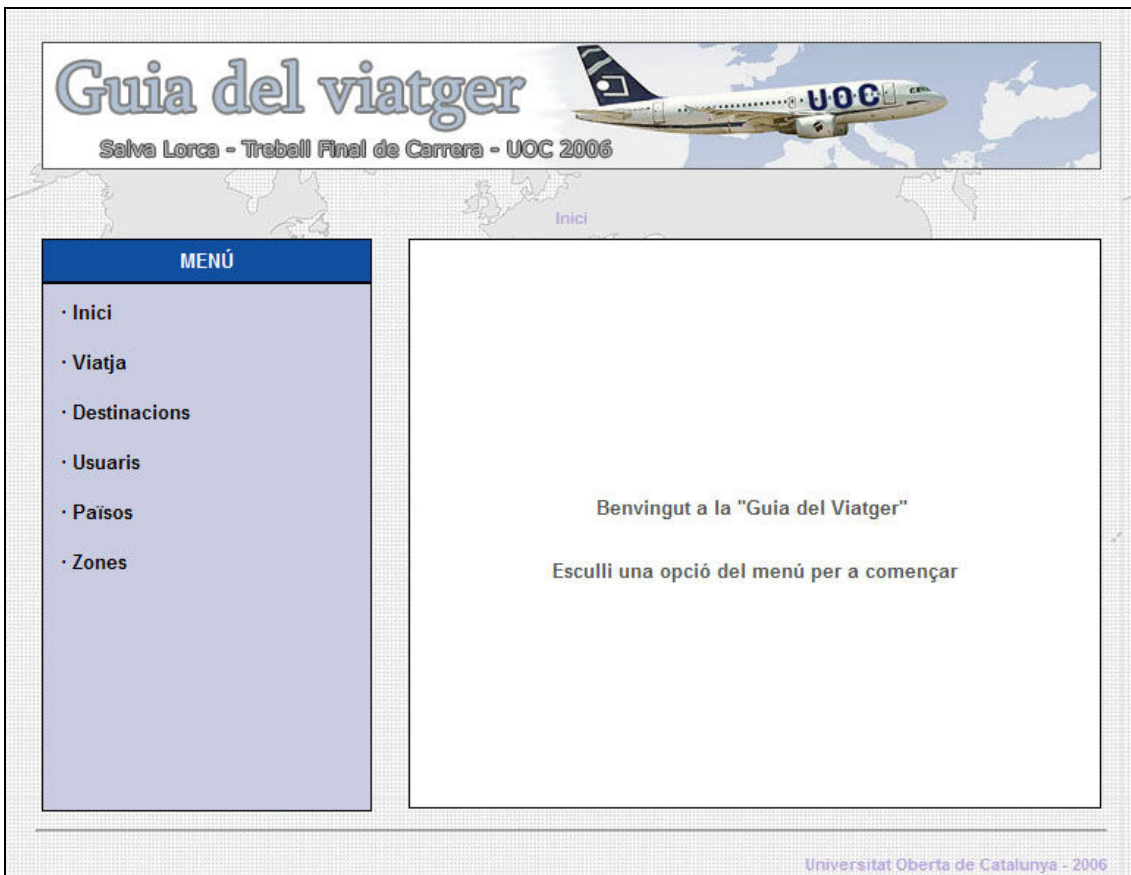
El disseny de la interfície d'usuari ens permet observar com es representen les diferents dades, com s'implementen els diàlegs i el disseny d'algunes finestres.



Il·lustració 26. Identificació de l'usuari. Pantalla Inicial



Il·lustració 27. Identificació de l'usuari. Login incorrecte



Il·lustració 28. Pantalla Principal

**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006



[Inici](#) > Administració de països

**Administració de països**

Llistat

Crear Nou

**Llistat de països:**

Nom	Bandera	Continent	Editar	Borrar
Espanya		EUROPA		
Finlàndia		EUROPA		
Itàlia		EUROPA		
Xina		ASIA		

Universitat Oberta de Catalunya - 2006

Il·lustració 29. Països. Llistat

**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006

[Inici](#) > Administració de països

**Administració de països**

Llistat

Crear Nou

**Dades generals:**

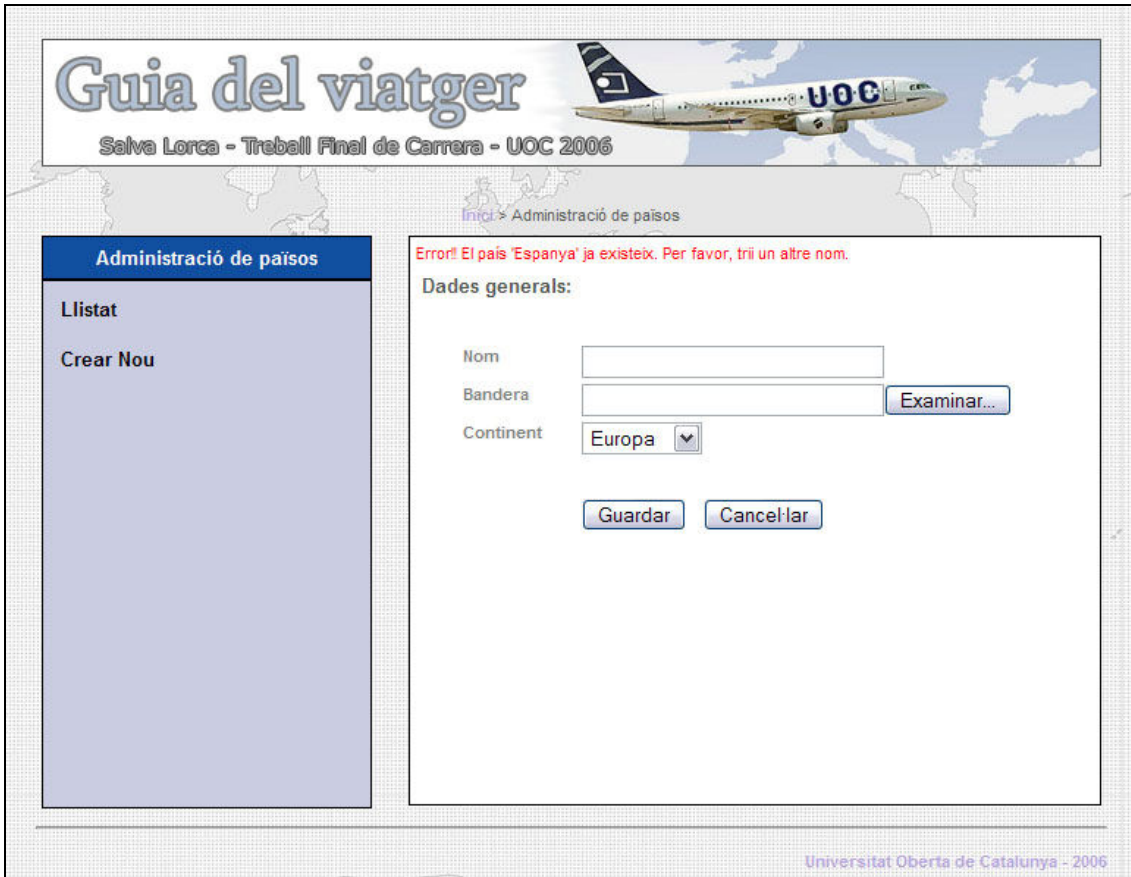
Nom:

Bandera:

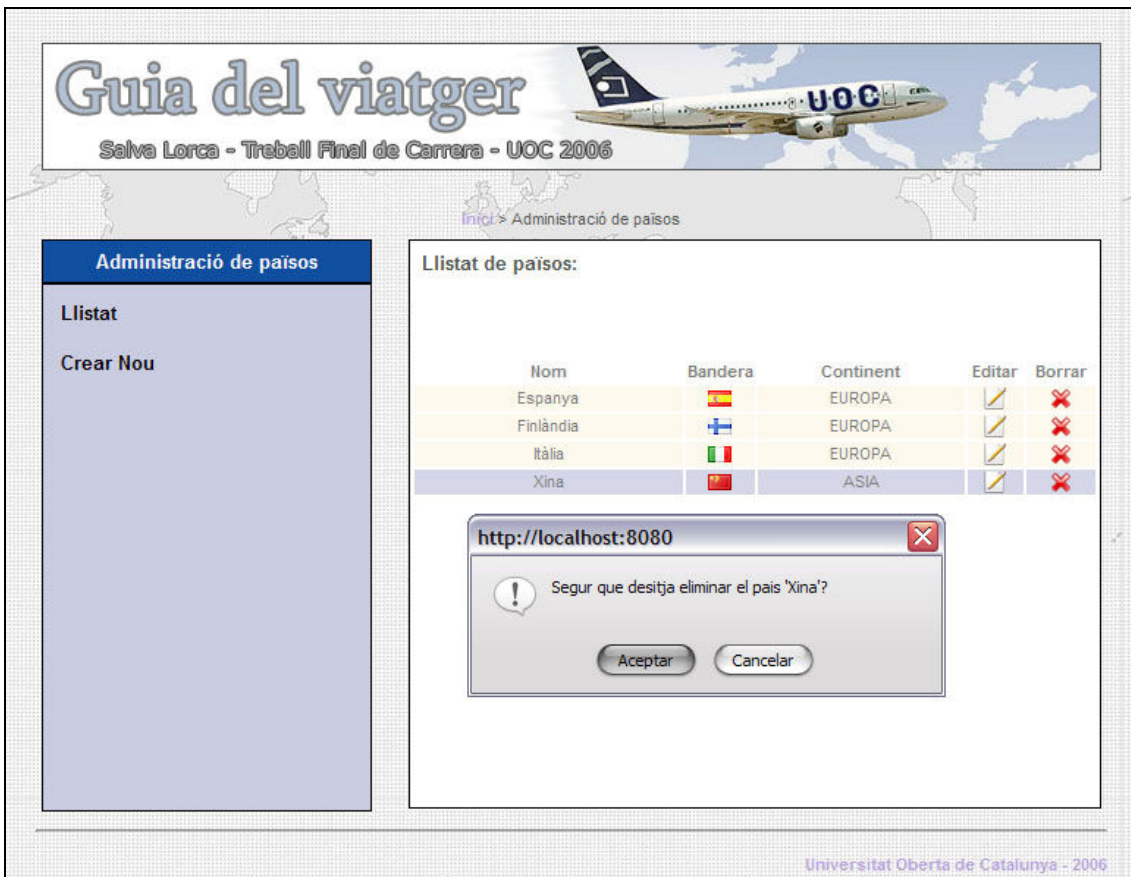
Continent:  ▼

Universitat Oberta de Catalunya - 2006

Il·lustració 30. Països. Nou i Edició



Il·lustració 31. Països. Guardar país ja existent



Il·lustració 32. Països. Eliminar país

**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006

[Inici](#) > Administració de països













**Administració de països**

Llistat

Crear Nou

**Llistat de països:**

El país Xina conté una o més zones. No es pot eliminar

Nom	Bandera	Continent	Editar	Borrar
Espanya		EUROPA		
Finlàndia		EUROPA		
Itàlia		EUROPA		
Xina		ASIA		

Universitat Oberta de Catalunya - 2006

II·lustració 33. Països. Eliminar país amb una o més zones

**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006













[Inici](#) > Administració de zones

**Administració de zones**

Llistat

Crear Nova

**Llistat de zones:**


Nom	País	Editar	Borrar
Helsinki	 Finlàndia		
Mallorca	 Espanya		
Pekí	 Xina		
Roma	 Itàlia		

Universitat Oberta de Catalunya - 2006

II·lustració 34. Zones. Llistat



**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006



[Inici](#) > Administració de zones

**Administració de zones**

Llistat

Crear Nova

**Dades generals:**


Nom

Pais

Universitat Oberta de Catalunya - 2006

Il·lustració 35. Zones. Nova i Edició

**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006



[Inici](#) > Administració de zones

**Administració de zones**

Llistat

Crear Nova

Error! La zona 'Mallorca' ja existeix. Per favor, triu un altre nom.

**Dades generals:**

Nom

Pais

Universitat Oberta de Catalunya - 2006

Il·lustració 36. Zones. Guardar zona ja existent

**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006

[Inici](#) > Administració de zones

**Administració de zones**

Llistat

Crear Nova

Llistat de zones:

Nom	País	Editar	Borrar
Helsinki	Finlàndia		
Mallorca	Espanya		
Peki	Xina		
Roma	Itàlia		

http://localhost:8080

! Segur que desitja eliminar la zona 'Mallorca'?

Universitat Oberta de Catalunya - 2006

II·lustració 37. Zones. Eliminar zona

**Guia del viatger**  
Salva Lorca - Treball Final de Carrera - UOC 2006

[Inici](#) > Administració de zones

**Administració de zones**

Llistat

Crear Nova

Llistat de zones:

La zona Mallorca conté una o més destinacions. No es pot eliminar

Nom	País	Editar	Borrar
Helsinki	Finlàndia		
Mallorca	Espanya		
Peki	Xina		
Roma	Itàlia		

Universitat Oberta de Catalunya - 2006

II·lustració 38. Zones. Eliminar zona amb una o més destinacions

# Guia del viatger

Salva Lorca - Treball Final de Carrera - UOC 2006



Inici > Administració de destinacions

## Administració de destinacions

Llistat

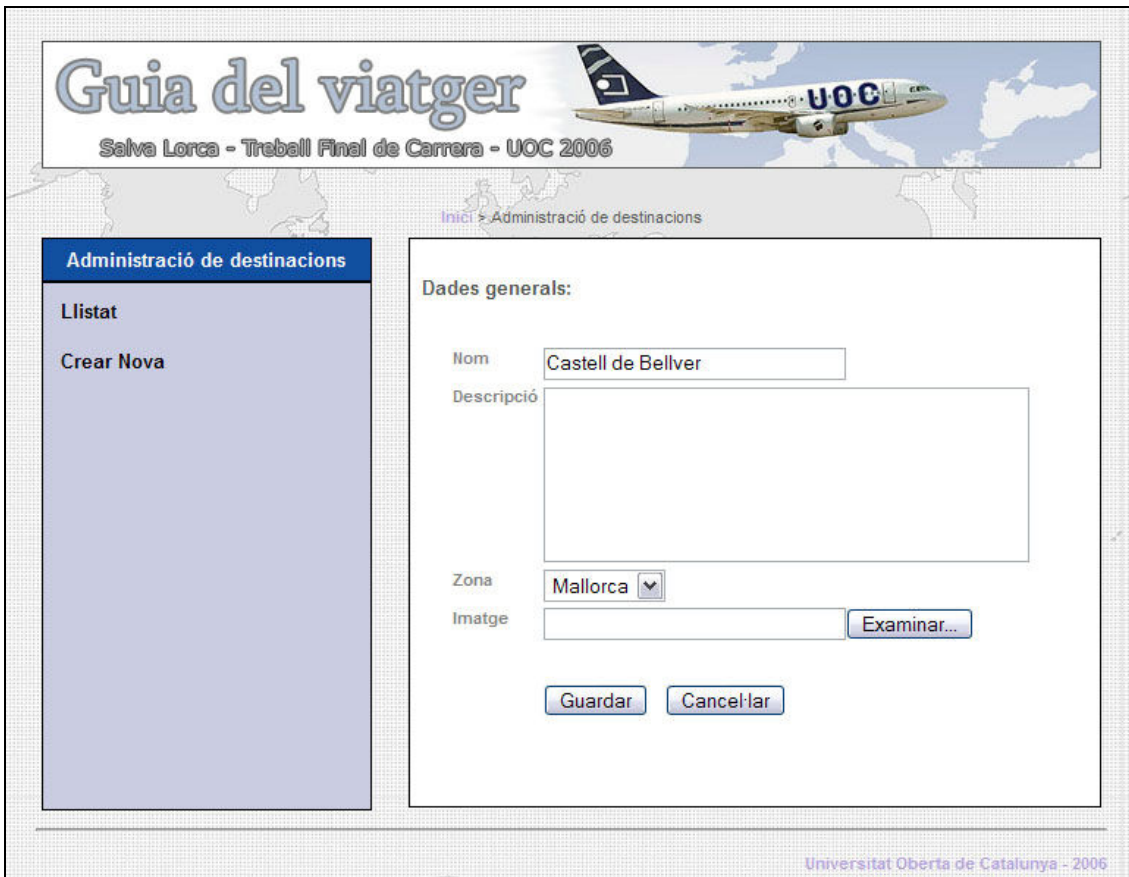
Crear Nova

## Llistat de destinacions:

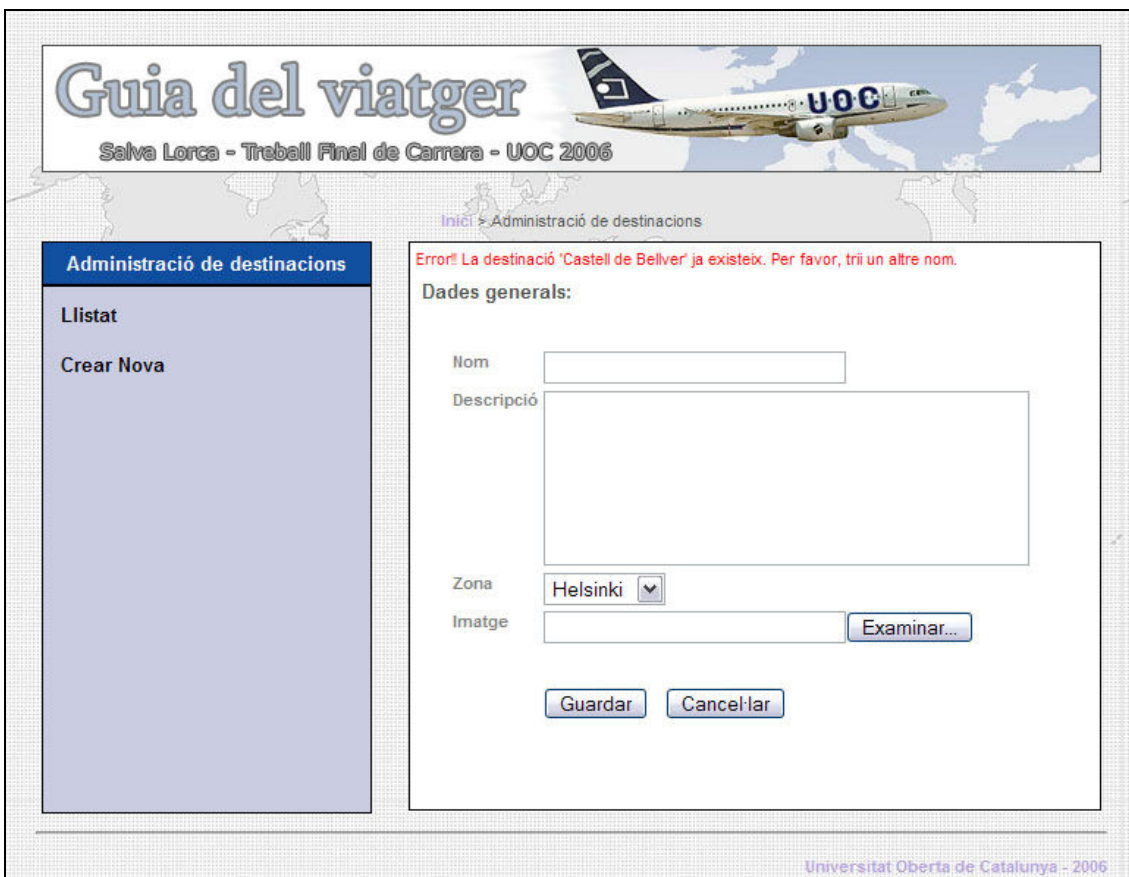
Nom	Zona	Imatge	Editar	Borrar
Auditori Rengo Piano	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Basilica de Sant Pere	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cala Varques	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cartoixa de Valldemossa	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Castell de Belver	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Castell de Capdepera	Mallorca		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Castell de Sant Angelo	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Colisèum	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
El Calvari (Pollença)	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Estació de Sóller	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fonts Ufanés de Campanet	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fòrum romà	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
La ciutat romana de Pollentia	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
La fortaleza de Suomenlinna	Helsinki	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
La Seu	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Muralla Xinesa	Pekí	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Museu Borghese	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Panteó	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Plaça de la República	Roma	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Puig de Galatzó	Mallorca	<a href="#">Veure</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Universitat Oberta de Catalunya - 2006

Il·lustració 39. Destinacions. Llistat




Il·lustració 40. Destinacions. Nova i Edició



Il·lustració 41. Destinacions. Guardar destinació ja existent

# Guia del viatger

Salva Lorca - Treball Final de Carrera - UOC 2006



Inici > Administració de destinacions

**Administració de destinacions**

Llistat

Crear Nova

Llistat de destinacions:

Nom	Zona	Imatge	Editar	Borrar
Auditori Rm Renzo Piano	Roma	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Basilica de Sant Pere	Roma	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Estació de Sóller	Mallorca	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fons Ufanés de Campanet	Mallorca	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fòrum romà	Roma	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
La ciutat romana de Pollentia	Mallorca	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
La fortalesa de Suomenlinna	Helsinki	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
La Seu	Mallorca	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Muralla Xinesa	Pekí	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Museu Borghese	Roma	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Panteó	Roma	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Plaça de la República	Roma	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Puig de Galatzó	Mallorca	Veure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

http://localhost:8080

! Segur que desitja eliminar la destinació 'La fortalesa de Suomenlinna'?


Aceptar Cancelar

Universitat Oberta de Catalunya - 2006

Il·lustració 42. Destinacions. Eliminar destinació

# Guia del viatger

Salva Lorca - Treball Final de Carrera - UOC 2006



[Inici](#) > Administració d'usuaris

**Administració d'usuaris**

Llistat

Crear Nou

**Llistat d'usuaris:**


Nom	Cognoms	E-mail	Editar	Borrar
Jordi	Aguiló Fiol	jaguilof@uoc.edu		
Fernando	Alonso	falonso@uoc.edu		
Carme	Eugenio Maneiro	ceugenio@uoc.edu		
Xisca	Figuerola Rovira	ffiguerola@uoc.edu		
Diego	Lorca Crespí	dlorca@uoc.edu		
Salvador	Lorca Sans	salvinha@uoc.edu		
Gabriel Alfonso	Martínez Cortés	gmartinezco@uoc.edu		
Sandra	Moratilla Mora	smoratilla@uoc.edu		
Marcos	Patricio Amate	mpatricio@uoc.edu		
Juan	Patricio Torrens	jpatricio@uoc.edu		
Verónica	Peña Pastor	vpena0@uoc.edu		
Juana	Perea Pérez	jpereap@uoc.edu		
Miquel	Piulats Mateu	mpiulats@uoc.edu		
Juanmi	Sastre Cuesta	jsastrec@uoc.edu		

Universitat Oberta de Catalunya - 2006

II·lustració 43. Usuaris. Llistat

# Guia del viatger

Salva Lorca - Treball Final de Carrera - UOC 2006



[Inici](#) > Administració d'usuaris

**Administració d'usuaris**

Llistat

Crear Nou

**Dades identificació:**

Nom del campus:

Clau:

**Dades generals:**

Nom:

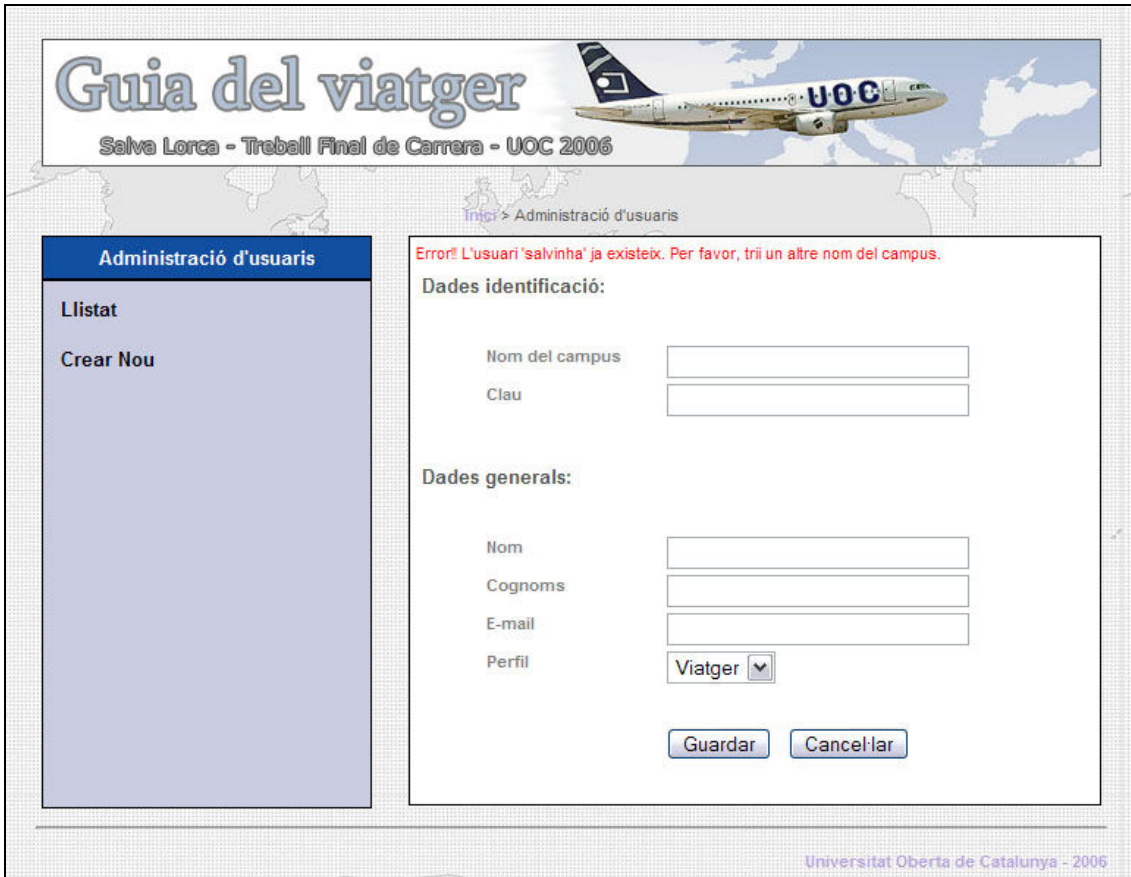
Cognoms:

E-mail:

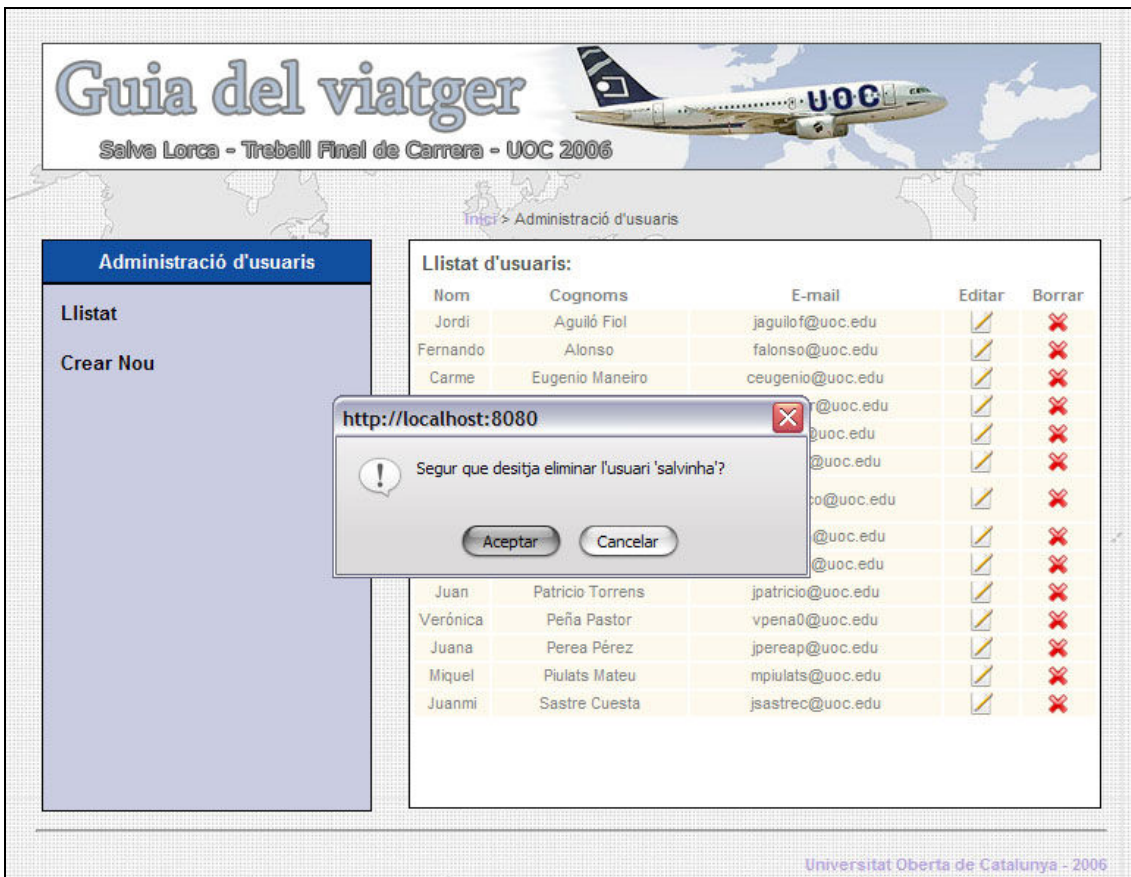
Perfil:  ▼

Universitat Oberta de Catalunya - 2006

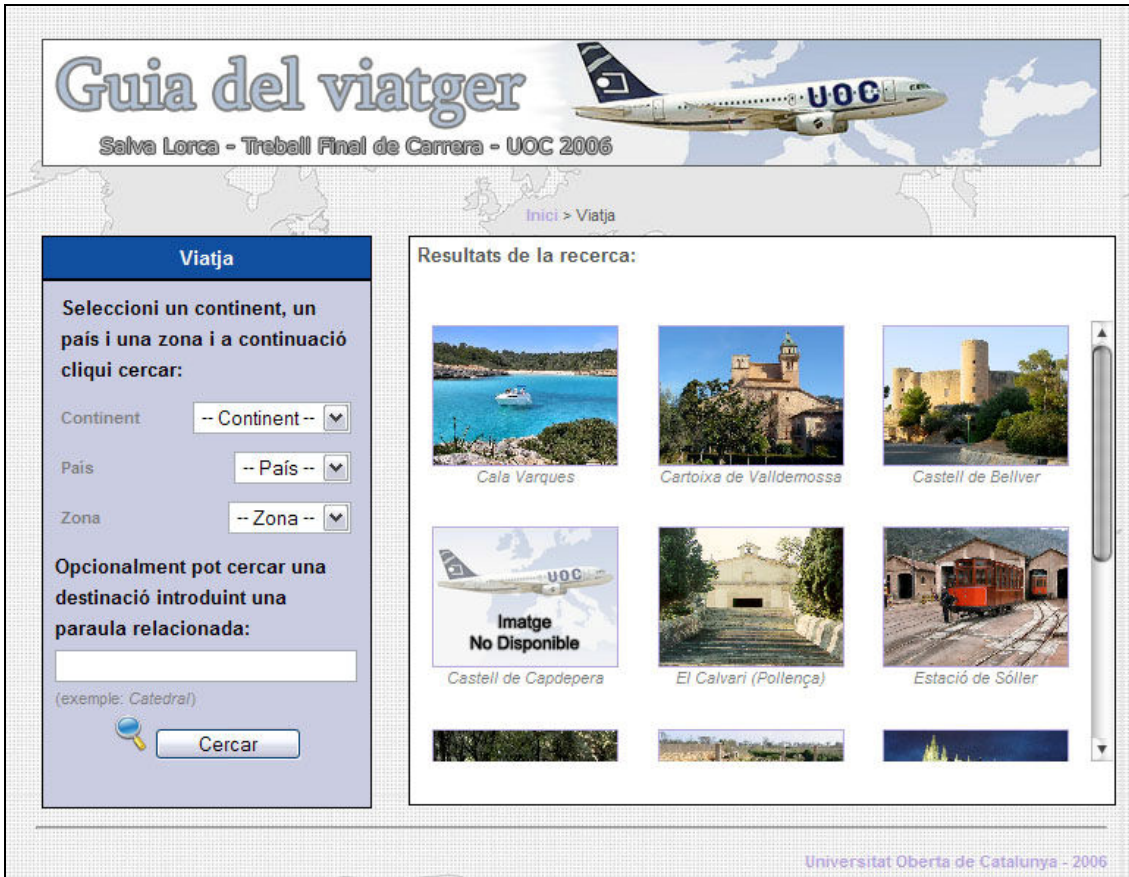
II·lustració 44. Usuaris. Nou i Edició



Il·lustració 45. Usuaris. Guardar usuari ja existent



Il·lustració 46. Usuaris. Eliminar usuari



II·lustració 47. Viatja. Recerca de destinacions



II·lustració 48. Viatja. Visualització de destinació



## 5. Fase d'implementació

### 5.1. Exemple de codificació

Com ja hem anomenat al principi de la memòria, el TFC està implementat en el llenguatge Java, un llenguatge orientat a objectes (OO) pur que facilita l'obtenció gairebé directa de la implementació a partir del disseny.

A continuació donem la codificació en Java del *servlet* ControladorPaisos, com a exemple complet de la implementació de la Guia del Viatger.

```
package gv.servlet;

import gv.ejb.*;
import gv.model.PaisModel;
import gv.model.ZonaModel;

import java.io.IOException;
import java.rmi.RemoteException;
import java.util.*;
import java.sql.SQLException;

import javax.ejb.CreateException;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.*;
import java.lang.reflect.UndeclaredThrowableException;

/**
 * Servlet que fa de controlador per al perfil PAIS.
 * Reb totes les peticions de pais, tria la lògica de negoci,
 * invoca el mètode de negoci i redirecciona a la vista adequada en cada cas.
 */
public class ControladorPaisos extends HttpServlet {

    private PaisHome paihome=null;
    private ZonaHome zonhome=null;
    // constants per a les operacions
    static final String LLISTAT_PAISOS = "llistatPaisos";
    static final String BORRAR_PAIS = "borrarPais";
    static final String FORM_NOU_PAIS = "formNouPais";
    static final String NOU_PAIS = "nouPais";
    static final String CARREGAR_PAIS = "carregarPais";
    static final String EDITAR_PAIS = "editarPais";

    // constants per a les pàgines de sortida
    static final String PG_ERROR = "/index.jsp";
    static final String PG_MOSTRAR_PAISOS = "/paisos.jsp";
    static final String PG_FORMULARI_PAISOS = "/nou_pais.jsp";
    static final String PG_FORMULARI_PAISOS_EDICIO = "/editar_pais.jsp";

    /**
     * mètode d'inicialització que recupera un home object per a l'EJB PaisEJB
     */
    public void init(ServletConfig sc) throws ServletException {
        super.init(sc);
        Context ctx;
        try {
            ctx = new InitialContext(System.getProperties());
            Object obj = ctx.lookup("gv.PaisEJB");
            Object objz = ctx.lookup("gv.ZonaEJB");
            paihome = (PaisHome) javax.rmi.PortableRemoteObject.narrow(obj,
                PaisHome.class);
            zonhome = (ZonaHome) javax.rmi.PortableRemoteObject.narrow(objz,
                ZonaHome.class);
        }
        catch (NamingException e) {
```

```

        e.printStackTrace();
    }
}

/**
 * Mètode que sobrecarrega el service de HttpServlet
 * L'utilitzem per a tractar les requests, generar les response i
 * redirreccionar a les pàgines adients en cada cas
 */
public void service(HttpServletRequest request, HttpServletResponse response)
throws IOException {
    String operacio = request.getParameter("operacio");
    String pk = request.getParameter("pk");
    String seguentPag = PG_ERROR;
    String status;
    try {
        if (operacio.equals(LLISTAT_PAISOS))
        {
            Collection paisos = llistarPaisos();
            request.setAttribute(LLISTAT_PAISOS, paisos);
            seguentPag=PG_MOSTRAR_PAISOS;
        }
        if (operacio.equals(BORRAR_PAIS))
        {
            try
            {
                Zona zo;
                zo =zonhome.create();
                String existeix;
                existeix = request.getParameter("pk");
                //System.out.println("CERCAM SI EXISTEIX ALGUNA ZONA AMB AQUEST
PAIS: "+existeix);
                ZonaModel z = zo.loadZonaPaisos(existeix);
                if(z == null) //Usuari inexistent
                {
                    Pais pai;
                    pai = paihome.create();
                    pai.delete(pk);
                    Collection paisos = llistarPaisos();
                    request.setAttribute(LLISTAT_PAISOS, paisos);
                    seguentPag=PG_MOSTRAR_PAISOS;
                }
                else
                {
                    Pais pa;
                    pa = paihome.create();
                    PaisModel p = pa.loadById(z.getPais());
                    status = "El país "+p.getNom()+" conté una o més zones. No es
pot eliminar";
                    request.setAttribute("status", status);
                    Collection paisos = llistarPaisos();
                    request.setAttribute(LLISTAT_PAISOS, paisos);
                    seguentPag=PG_MOSTRAR_PAISOS;
                }
            }catch (Exception e)
            {
                e.printStackTrace();
                status = "Error en temps d'execució. Consulti amb
l'administrador!!";
                request.setAttribute("status", status);
                seguentPag=PG_MOSTRAR_PAISOS;
            }
        }
        if (operacio.equals(FORM_NOU_PAIS))
        {
            seguentPag=PG_FORMULARI_PAISOS;
        }
        if (operacio.equals(NOU_PAIS))
        {
            try
            {
                Pais pai_nou;
                pai_nou = paihome.create();
                String existeix;
                existeix = request.getParameter("nompai");
                System.out.println("CERCAM SI EXISTEIX: "+existeix);
            }
        }
    }
}

```

```

PaisModel p = pai_nou.loadByNom(existeix);
if (p == null)//si no existeix l'usuari el cream nou
{
    p = new PaisModel();
    p.setId(request.getParameter("id"));
    p.setNom(request.getParameter("nompai"));
    p.setBandera(request.getParameter("banpai"));
    p.setContinent(request.getParameter("conpai"));
    pai_nou.save(p);
    request.setAttribute("id",p);
    Collection paisos = llistarPaisos();
    request.setAttribute(LLISTAT_PAISOS, paisos);
    seguentPag=PG_MOSTRAR_PAISOS;
}
else
{
    status = "Error!! El país '"+existeix+"' ja existeix. Per favor,
trii un altre nom.";
    request.setAttribute("status", status);
    seguentPag=PG_FORMULARI_PAISOS;
}
}catch (Exception e)
{
    e.printStackTrace();
    status = "Error en temps d'execució. Consulti amb
l'administrador!!";
    request.setAttribute("status", status);
    seguentPag=PG_FORMULARI_PAISOS;
}
}
if (operacio.equals(CARREGAR_PAIS))
{
    try
    {
        Pais pai_nou;
        pai_nou = paihome.create();
        System.out.println("CERCAM SI EXISTEIX: "+pk);
        PaisModel p = pai_nou.loadById(pk);
        if (p != null)//si existeix el país carregam les dades al formulari
        {
            request.setAttribute("id",p);
            request.setAttribute("nompai", p.getNom());
            request.setAttribute("banpai", p.getBandera());
            request.setAttribute("conpai", p.getContinent());
            seguentPag=PG_FORMULARI_PAISOS_EDICIO;
        }
        else
        {
            status = "Error!! El país no existeix. Per favor, trii un altre
país";
            request.setAttribute("status", status);
            seguentPag=PG_FORMULARI_PAISOS_EDICIO;
        }
    }catch (Exception e)
    {
        e.printStackTrace();
        status = "Error en temps d'execució. Consulti amb
l'administrador!!";
        request.setAttribute("status", status);
        seguentPag=PG_FORMULARI_PAISOS_EDICIO;
    }
}
if (operacio.equals(EDITAR_PAIS))
{
    try
    {
        Pais pai_nou;
        pai_nou = paihome.create();
        String existeix;
        existeix = request.getParameter("nompai");
        pk = request.getParameter("pk");
        //System.out.println("CERCAM SI EXISTEIX: "+existeix);
        PaisModel p = pai_nou.loadByNomExistent(existeix,pk);
        if (p == null)//si no existeix el pais el cream nou
        {
            p = new PaisModel();
            p.setNom(request.getParameter("nompai"));

```

```

        if (request.getParameter("banpai").equals(""))
        {
            p.setBandera(request.getParameter("bandera"));
        }
        else
        {
            p.setBandera(request.getParameter("banpai"));
        }
        p.setContinent(request.getParameter("conpai"));
        p.setId(request.getParameter("pk"));
        pai_nou.update(p);
        request.setAttribute("pk",p);
        Collection paisos = llistarPaisos();
        request.setAttribute(LLISTAT_PAISOS, paisos);
        seguentPag=PG_MOSTRAR_PAISOS;
    }
    else
    {
        status = "Error!! El país '"+existeix+"' ja existeix. Per favor,
triï un altre nom.";
        request.setAttribute("status", status);
        seguentPag=PG_FORMULARI_PAISOS;
    }
} catch (Exception e)
{
    e.printStackTrace();
    status = "Error en temps d'execució. Consulti amb
l'administrador!!";
    request.setAttribute("status", status);
    seguentPag=PG_FORMULARI_PAISOS;
}
}
String perusu;
perusu = request.getParameter("perusu");
request.setAttribute("perusu", perusu);
//redireccionem a la JSP adient
RequestDispatcher rd =
getServletContext().getRequestDispatcher(seguentPag);
rd.forward(request, response);
} catch (Exception e) {
    e.printStackTrace();
}
}

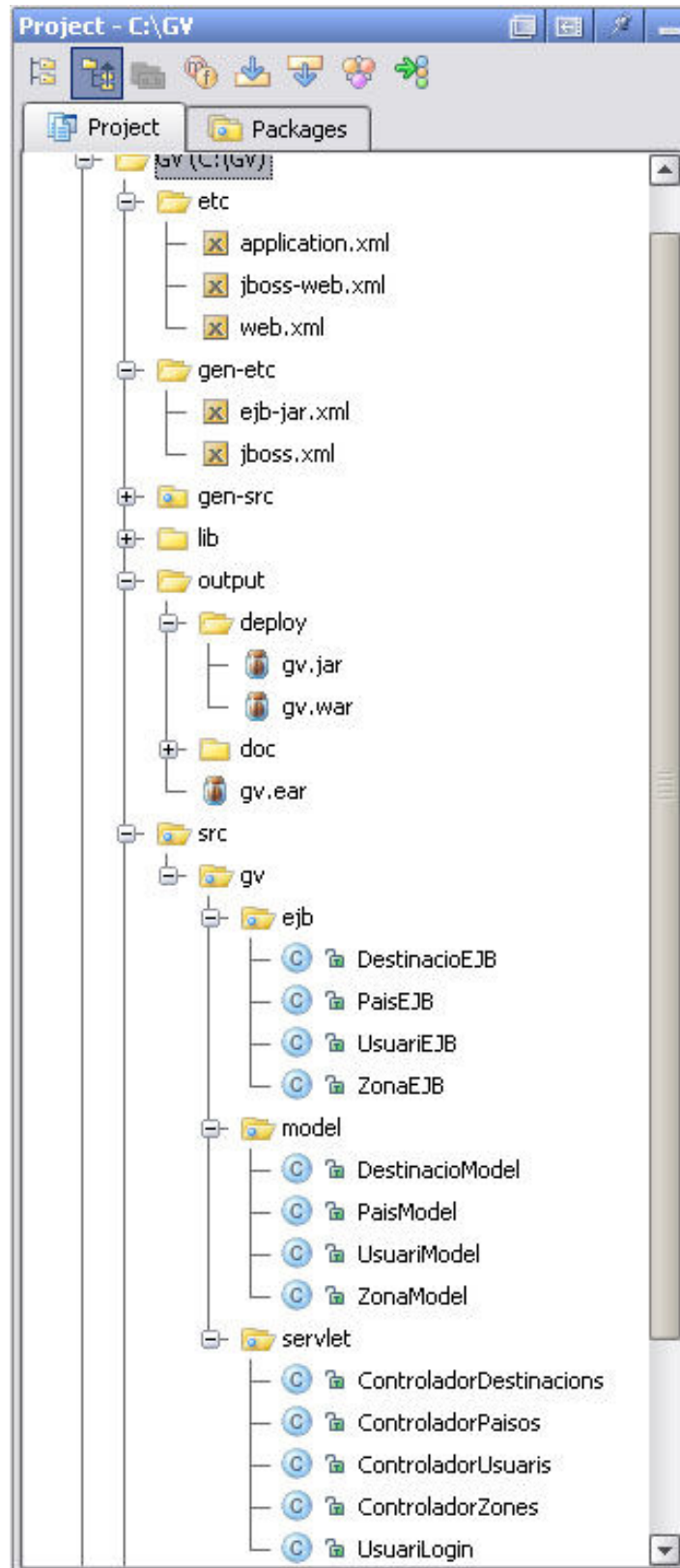
/**
 * Mètode que llista tots els paisos
 * @return llista de paisos
 * @throws java.rmi.RemoteException
 * @throws javax.ejb.CreateException
 */
private Collection llistarPaisos() throws RemoteException, CreateException,
SQLException, UndeclaredThrowableException {
    Pais pais = paihome.create();
    return pais.loadAll(1);
}
}
}

```

Il·lustració 49. Exemple de codi de la implementació

## 5.2. Estructura d'arxius

La separació de capes ve reflexada en la següent il·lustració:



Il·lustració 50. Estructura d'arxius

### 5.3. Script de creació de la base de dades per a MySql

```
-- MySQL dump 10.10
--
-- Host: localhost   Database: GV
-----
-- Server version 5.0.20a-nt

CREATE DATABASE `GV`;
USE `GV`;

SET FOREIGN_KEY_CHECKS=0;
--
-- Table structure for table `continent`
--

DROP TABLE IF EXISTS `continent`;
CREATE TABLE `continent` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `nom` varchar(45) NOT NULL default "",
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `continent`
--

/*!40000 ALTER TABLE `continent` DISABLE KEYS */;
LOCK TABLES `continent` WRITE;
INSERT INTO `continent` VALUES (1,'Europa'),(2,'Àfrica'),(3,'Amèrica'),(4,'Àsia'),(5,'Oceania');
UNLOCK TABLES;
/*!40000 ALTER TABLE `continent` ENABLE KEYS */;

--
-- Table structure for table `destinacio`
--

DROP TABLE IF EXISTS `destinacio`;
CREATE TABLE `destinacio` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `nom` varchar(45) NOT NULL default "",
  `descripcio` varchar(1024) default NULL,
  `zona` int(10) unsigned NOT NULL default '0',
  `imatge` varchar(200) default NULL,
  PRIMARY KEY (`id`),
  KEY `FK_destinacio_1` (`zona`),
  CONSTRAINT `FK_destinacio_1` FOREIGN KEY (`zona`) REFERENCES `zona` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 4096 kB; (`id`) REFER
`gv/zona` (`id`)';

--
-- Dumping data for table `destinacio`
--

/*!40000 ALTER TABLE `destinacio` DISABLE KEYS */;
LOCK TABLES `destinacio` WRITE;
INSERT INTO `destinacio` VALUES (1,'Castell de Sant Angelo','Monument romà',4,'Castell Sant
Angelo.jpg'),(2,'Panteó','Panteó romà',4,'Panteo.jpg'),(5,'Muralla Xinesa','És una antiga fortificació
construïda a partir del segle XIV fins al començament del segle XVI, durant la dinastia Ming, per tal de
protegir la Xina dels atacs de les tribus mongoles i turques. Des del segle III aC es van anar construint
diferents fortificacions contra els atacs de les poblacions nòmades del nord, que habitaven les planures
avui ocupades per Mongòlia i Manxúria.',11,'muralla_xinesa.jpeg'),(7,'Auditoriù Renzo
Piano','Auditoriù romà',4,'Auditorium_Renzo_Piano.jpg'),(8,'Basílica de Sant Pere','Basílica situada a la
Ciutat del Vaticà',4,'Basilica_San_Pere.JPG'),(9,'Fòrum romà','El fòrum era el punt de trobada dels
ciutadans i el lloc on se celebraven les eleccions dels magistrats i on s'aixecaven alguns dels principals
edificis de la ciutat. Hi havia el temple més important, la cúria o senat local -centre administratiu de la
ciutat- i la basílica. A tot el voltant s'alçaven estàtues dedicades als emperadors o a personatges
notables de la ciutat.',4,'Forum_roma.jpg'),(10,'Colisèum','Colisèum
romà',4,'Lightmatter_colosseum.jpg'),(11,'Museu Borghese','Antiga residència del Cardenal Scipione
```

```

Borghese.',4,'Museu_Borghese.jpg'),(12,'Plaça de la República','La Plaça de la República forma part de la
gran reurbanització portada a terme a Roma quan aquesta va passar a formar part de la Itàlia
unificada.',4,'Plaza_de_la_Republica.jpg'),(13,'Castell de Bellver','El Castell de Bellver, situat a uns tres
kilòmetres del centre de Palma, s'aixeca a 112 metres d'alçada sobre el nivell del
mar.',13,'castell_de_Bellver.jpg');
UNLOCK TABLES;
/*!40000 ALTER TABLE `destinacio` ENABLE KEYS */;

--
-- Table structure for table `jms_rols`
--

DROP TABLE IF EXISTS `jms_rols`;
CREATE TABLE `jms_rols` (
  `ROLEID` varchar(32) NOT NULL,
  `USERID` varchar(32) NOT NULL,
  PRIMARY KEY (`USERID`,`ROLEID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `jms_rols`
--

/*!40000 ALTER TABLE `jms_rols` DISABLE KEYS */;
LOCK TABLES `jms_rols` WRITE;
INSERT INTO `jms_rols` VALUES
('durpublisher','dynsub'),('publisher','dynsub'),('guest','guest'),('j2ee','guest'),('john','guest'),('durpublish
er','john'),('publisher','john'),('subscriber','john'),('noacc','nobody');
UNLOCK TABLES;
/*!40000 ALTER TABLE `jms_rols` ENABLE KEYS */;

--
-- Table structure for table `jms_subscriptions`
--

DROP TABLE IF EXISTS `jms_subscriptions`;
CREATE TABLE `jms_subscriptions` (
  `CLIENTID` varchar(128) NOT NULL,
  `SUBNAME` varchar(128) NOT NULL,
  `TOPIC` varchar(255) NOT NULL,
  `SELECTOR` varchar(255) default NULL,
  PRIMARY KEY (`CLIENTID`,`SUBNAME`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `jms_subscriptions`
--

/*!40000 ALTER TABLE `jms_subscriptions` DISABLE KEYS */;
LOCK TABLES `jms_subscriptions` WRITE;
UNLOCK TABLES;
/*!40000 ALTER TABLE `jms_subscriptions` ENABLE KEYS */;

--
-- Table structure for table `jms_users`
--

DROP TABLE IF EXISTS `jms_users`;
CREATE TABLE `jms_users` (
  `USERID` varchar(32) NOT NULL,
  `PASSWD` varchar(32) NOT NULL,
  `CLIENTID` varchar(128) default NULL,
  PRIMARY KEY (`USERID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `jms_users`
--

```

```

/*!40000 ALTER TABLE `jms_users` DISABLE KEYS */;
LOCK TABLES `jms_users` WRITE;
INSERT INTO `jms_users` VALUES
('dynsub','dynsub',NULL),('guest','guest',NULL),('j2ee','j2ee',NULL),('john','needle','DurableSubscriberExample'),('nobody','nobody',NULL);
UNLOCK TABLES;
/*!40000 ALTER TABLE `jms_users` ENABLE KEYS */;

--
-- Table structure for table `pais`
--

DROP TABLE IF EXISTS `pais`;
CREATE TABLE `pais` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `NOMPAI` varchar(45) NOT NULL default "",
  `BANPAI` varchar(200) default NULL,
  `CONPAI` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `FK_pais_1` (`CONPAI`),
  CONSTRAINT `FK_pais_1` FOREIGN KEY (`CONPAI`) REFERENCES `continent` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 4096 kB; (`id`) REFER
`gv/continent` (`id`);

--
-- Dumping data for table `pais`
--

/*!40000 ALTER TABLE `pais` DISABLE KEYS */;
LOCK TABLES `pais` WRITE;
INSERT INTO `pais` VALUES (6,'Itàlia','it.gif',1),(9,'Xina','cn.gif',4),(14,'Espanya','es.gif',1);
UNLOCK TABLES;
/*!40000 ALTER TABLE `pais` ENABLE KEYS */;

--
-- Table structure for table `usuari`
--

DROP TABLE IF EXISTS `usuari`;
CREATE TABLE `usuari` (
  `NOMCAMUSU` varchar(15) NOT NULL default "",
  `PWDUSU` varchar(10) NOT NULL default "",
  `NOMUSU` varchar(45) NOT NULL default "",
  `COGUSU` varchar(45) NOT NULL default "",
  `EMAUSU` varchar(45) NOT NULL default "",
  `PERUSU` int(10) unsigned NOT NULL default '1',
  PRIMARY KEY (`NOMCAMUSU`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `usuari`
--

/*!40000 ALTER TABLE `usuari` DISABLE KEYS */;
LOCK TABLES `usuari` WRITE;
INSERT INTO `usuari` VALUES ('ceugenio','ceu','Carme','Eugenio
Maneiro','ceugenio@uoc.edu',2),('dllorca','dll','Diego','Llorca
Crespí','dllorca@uoc.edu',2),('falonso','fal','Fernando','Alonso','falonso@uoc.edu',2),('ffiguerolar','ffi','Xisc
a','Figueroa Rovira','ffiguerolar@uoc.edu',2),('gmartinezco','gma','Gabriel Alfonso','Martínez
Cortés','gmartinezco@uoc.edu',2),('jaguilof','jag','Jordi','Aguiló
Fiol','jaguilof@uoc.edu',2),('jpatricio','jpa','Juan','Patricio
Torrens','jpatricio@uoc.edu',2),('jpereap','jpe','Juana','Perea
Pérez','jpereap@uoc.edu',2),('jsastrec','jsa','Juanmi','Sastre
Cuesta','jsastrec@uoc.edu',2),('mpatricio','mpa','Marcos','Patricio
Amate','mpatricio@uoc.edu',2),('mpiulats','mpi','Miquel','Piulats
Mateu','mpiulats@uoc.edu',2),('salvinha','sal','Salvador','Lorca
Sans','salvinha@uoc.edu',1),('smoratilla','smo','Sandra','Moratilla
Mora','smoratilla@uoc.edu',2),('vpena0','vpe','Verónica','Peña Pastor','vpena0@uoc.edu',1);
UNLOCK TABLES;
/*!40000 ALTER TABLE `usuari` ENABLE KEYS */;

```



```

--
-- Table structure for table `zona`
--

DROP TABLE IF EXISTS `zona`;
CREATE TABLE `zona` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `NOMZON` varchar(45) NOT NULL default '',
  `PAIZON` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `FK_zona_1` (`PAIZON`),
  CONSTRAINT `FK_zona_1` FOREIGN KEY (`PAIZON`) REFERENCES `pais` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 4096 kB; (`id`) REFER
`gv/pais` (`id`); InnoDB f';

--
-- Dumping data for table `zona`
--

/*!40000 ALTER TABLE `zona` DISABLE KEYS */;
LOCK TABLES `zona` WRITE;
INSERT INTO `zona` VALUES (4,'Roma',6),(11,'Pekí',9),(13,'Mallorca',14);
UNLOCK TABLES;
SET FOREIGN_KEY_CHECKS=1

```

Il·lustració 51. Script de creació de la BBDD

## 6. Conclusions

La conclusió general que s'ha obtingut és la plena realització dels objectius marcats a la planificació del treball.

Sobre l'arquitectura Java 2 Platform, Enterprise Edition (J2EE) podem dir que:

- J2EE comporta un desenvolupament complex.
- Existeixen múltiples eines i tecnologies, i hem de saber elegir aquelles que necessitem.
- Necessitat de proves de les diferents parts de l'aplicació, per a poder assegurar una certa qualitat del producte.
- Potenciació del *software lliure*.
- Qualitats de J2EE: escalabilitat, flexibilitat, seguretat, robustesa i qualitat del producte final.

El principal inconvenient ha estat la escassa experiència desenvolupant aplicacions amb l'arquitectura J2EE. Això, amb la gran càrrega de treball ja prevista des del començament ha fet que a vegades tingués la sensació de no avançar realment, sobretot a la fase d'implementació.

Durant el TFC he après a realitzar un treball de síntesi dels coneixements adquirits en altres assignatures de l'Enginyeria. Per tant, he pogut comprovar que he assimilat bé tots els conceptes i amb una de les millors tecnologies que existeixen en l'actualitat.

## 7. Glossari

**Administrador:** Usuari que gestiona els usuaris, els països, les zones i les destinacions. També pot cercar destinacions com el viatger.

**Continent:** Part de la ubicació de les destinacions turístiques. Els 5 continents configurats són: Àfrica, Amèrica, Àsia, Europa i Oceania.

**Destinació turística:** Lloc d'interés d'un país que el viatger cerca a la Guia.

**Identificació:** Procés de la Guia del Viatger on l'administrador i el viatger es connecten al programa.

**País:** Part de la ubicació de les destinacions turístiques. Un continent té més d'un país.

**Viatger:** Usuari que realitza recerques de destinacions turístiques.

**Viatja:** Part de la Guia del Viatger on es pot cercar les destinacions turístiques.

**Zona:** Part de la ubicació de les destinacions turístiques. Un país té una o més zones, i una zona engloba totes les destinacions turístiques.

## 8. Bibliografia

### 8.1. Fonts d'informació

[Adictos al trabajo](#)  
[The Server Side](#)  
[JavaWorld](#)  
[Pàgina de Sun sobre J2EE](#)  
[Programación.net - Java](#)  
[JavaHispano](#)  
[MySQL](#)  
[MySQL Hispano](#)  
[Jboss](#)  
[Wikipedia](#)

### 8.2. Autors

**Sun Microsystems, Inc.** (2004). *Introducción a J2EE. E-Book*

**Sun Microsystems, Inc.** (2003). *Core J2EE Patterns Index*

**Roman, Ed.** (2004). *Mastering enterprise JavaBeans(tm)*

**Xhafa, Fatos** (2002 – 2<sup>a</sup> edició). *Tècniques de desenvolupament de programari*.  
Barcelona: UOC

**Sun Microsystems** (2005). *The J2EE™ 1.4 Tutorial for Sun Java System Application Server Platform Edition 8.1 2005Q2 UR2*.  
California: Network circle.