

# **gosSOS – J2EE**

**Nom Estudiant: Jordi Sànchez Cano**  
ETIG

**Nom Consultor: Jose Juan Rodriguez**

Data Lliurament: 18 de juny de 2007

## **Dedicatòria i agraïments**

---

Vull dedicar aquest treball a Leyla, la meva companya de vida que ha estat al meu costat durant tota la carrera, on la nostra relació no ha sigut cap obstacle per a realitzar els meus estudis, si no que tot el contrari.

Encara que no he pogut dedicar el temps necessari com aquest projecte es mereix, vull agrair la inspiració i companyia de la meva gossa Lluna. Aquesta gosseta de 3 anys la varem recollir de la gossera i, encara que porta amb nosaltres un parell d'anys, té pors que mai sabrem què li va passar. Bé, el que és més important, és que ara podrà tenir una vida digna.

També vull agrair l'ajuda i els ànims del meu consultor Josep.

## Resum

---

Aquest projecte té com a finalitat fer accessibles els gossos que hi ha a una gossera per mitjà d'Internet per a qualsevol usuari d'arreu del món. **L'objectiu principal és l'adopció d'animals sense casa.**

Es vol que els usuaris puguin veure els diferents gossos que hi ha entre una o varies gosses i la possibilitat de poder-ne adoptar. Es dona la possibilitat de cercar gossos per característiques com grandària o sexe i es pot veure fotografies. S'intenta que el portal sigui també informatiu en referència a l'abandonament d'animals. D'aquesta manera, l'administrador de cada gossera juga un paper molt important: gestionar la informació com gossos, les seves fotografies, les reserves per a adopció pendents, etc.

També permet als usuaris introduir un gos perdut, amb els detalls d'aquest i la fotografia. D'aquesta manera, cada vegada que l'administrador introdueixi un gos perquè ha arribat a la gossera, pot comprovar si pertany a algun usuari relacionant-lo per característiques com grandària, sexe, etc.

El programari utilitzat i necessari per a posar en marxa l'aplicació és gratuït, on només serà necessari despeses en concepte de connexió a internet, servidor i opcionalment altres dispositius de seguretat com ara un tallafocs.

S'utilitzen varies tecnologies: la tecnologia J2EE per a l'aplicació web en si. Dins de J2EE es fa us del patró MVC Struts, per a la implementació de tipus Model-Controlador-Vista. Per a contenir les dades es fa servir l'SGBD MySQL Server 5.

## Índex de continguts

---

Dedicatòria .....	2
Resum .....	3
Introducció.....	5
Objectius .....	7
Planificació del projecte.....	9
Productes obtinguts.....	11
Descripció dels altres capítols .....	11
Anàlisi.....	12
Actors i guions .....	12
Model del domini.....	14
Casos d'ús .....	15
Descripció textual dels casos d'us .....	17
Diagrames de col·laboració.....	27
Glossari del model de negoci.....	29
Disseny .....	30
Disseny arquitectònic.....	30
Revisió del diagrama estàtic .....	32
Disseny de la persistència .....	34
Diagrama E-R .....	34
Model relacional.....	35
Estructura de la interfície d'usuari.....	37
Instal·lació .....	39
Conclusions .....	40
Bibliografia .....	42
Annex A: script de creació de la BD .....	43

## Índex de figures

---

Figura 1. Temporització de les tasques.....	11
Figura 2. Diagrama de classes.....	14
Figura 3. Diagrama de cassos d'ús d'usuari.....	15
Figura 4. Diagrama de cassos d'ús d'administrador.....	16
Figura 5. Diagrama d'activitat: introduir gos perdut .....	21
Figura 6. Diagrama d'activitat: introduir gos abandonat .....	25
Figura 7. Diagrama de col·laboració: introduir gos abandonat .....	27
Figura 8. Diagrama de col·laboració: llistar gossos .....	28
Figura 9. Diagrama de col·laboració: visualitzar notificacions .....	29
Figura 10. Diagrama de col·laboració: notificar perdua .....	29
Figura 11. Esquema general de l'aplicació.....	30
Figura 12. Revisió del diagrama estàtic .....	33
Figura 13. Diagrama E-R .....	34
Figura 14. Estructura de la interfície d'usuari principal.....	37

## Introducció

---

### Justificació del TFC i context en el qual es desenvolupa

Malauradament el nombre d'animals abandonats, no només de gossos, creix. Encara que les campanyes que porta el govern per a evitar-ho són moltes, es segueix abandonant aquests animals que en un principi van ser part de la família i després una càrrega. Però això no és tot, aquests éssers poden acabar sota les rodes d'un vehicle, llençats dins d'un pou, lligats en un arbre fins que morin i altres situacions que costa d'imaginar.

L'ésser humà pensa que té dret sobre la vida dels altres animals, i que són una possessió. Un cadell pot ser un regal de reis o aniversari per a un nen, però és més que això: un ésser viu amb uns drets i unes necessitats que hem de tenir present abans de tenir-los a casa.

A Espanya, el 52% de les famílies tenen un animal de companyia. Quan arriben les festes de nadal, la xifra puja donat que un animal de companyia es considerat un bé de consum. Moltes d'aquestes compres, es converteixen en animals abandonats al mateix any, per la irresponsabilitat i poca reflexió.

A més, la venda d'animals de companyia s'està convertint en un gran negoci i mercat negre. Es poden veure cadells en petites gàbies de botigues d'animals a preus impensables, quan el veritable valor d'un gos és el que et pot aportar.

Acollir un animal abandonat no és la solució, si no que és ja una necessitat davant d'un consum frenètic i irresponsable. D'aquesta manera, s'evita participar en aquesta economia salvatge, i es deixa espai a la gossera per a altres gossos que vagin entrant.

L'adopció és un mitjà que ens permet identificar, mitjançant el xip, als gossos que surten i associar-los amb els seus amos. D'aquesta manera, els amos saben què estan fent i que són responsables.

Després d'aquesta introducció, podria ser interessant tenir accés mitjançant

internet a una gossera i a una sèrie de serveis que es poden dur a terme amb la col·laboració d'aquesta. Per tant, en aquest treball es pretén fer una aplicació web per a fomentar i facilitar l'adopció de gossos abandonats o perduts.

Amb aquest lloc web, es pretèn que un usuari es pugui acostar per a veure els gossos que hi pertanyen, adoptar-ne, denunciar una desaparició, etc.

Per altre costat, permet a les gosses gestionar via web els gossos que tenen com les altes, baixes, adopcions, etc. tenint així les dades sempre accessibles i actualitzades. D'aquesta manera, l'administrador de la gossera és el punt clau en aquesta aplicació perquè són els que mantenen viva la informació.

## Objectius

---

L'objectiu principal d'aquest treball és la creació d'una aplicació web per a la gestió de gosses, en concret es preten gestionar adopcions i denúncies de pèrdues de gossos. Tot això mitjançant la tecnologia J2EE i seguint un cicle de vida iteratiu i incremental.

En general, s'espera que l'aplicació satisfaci les següents funcionalitats (abast):

- **Afegir, esborrar i modificar:**

- ✓ Denúncies de desaparicions, per part dels usuaris.
- ✓ Gossos perduts, que entren a la gossera per part de l'administrador.
- ✓ Notícies, que publica l'administrador.

- **Notificacions**

Els usuaris amb denúncies vigents de gos perdut rebran notificacions que enllacen amb gossos amb semblances que vagin arribant. Es tracta d'un procés automàtic que es generarà cada vegada que s'introdueixi un gos.

- **Registrar-se**

Necessari per a poder introduir denúncies i poder rebre notificacions personals. Possibilitat de modificar les dades d'usuari (adreça, telèfon, contrasenya, etc)

- **Identificar-se**

- **Visualitzar:**

- ✓ Gossos desapareguts
- ✓ Gossos en adopció. Amb possibilitat de fer reserva per adopció

### **Objectius funcionals**

- Plataforma J2EE (jdk 1.5)
- Utilitzar Netbeans com a IDE per a desenvolupament.
- Apache Tomcat com a servidor web i contenidor de Servlets i JSPs.
- Instal·lar i gestionar un SGBD per a la BD de l'aplicació. Faré servir MySQL.
- Utilitzar l'objecte integrat session per al seguiment de les sessions d'usuari.
- Utilitzar un motor de persistència, potser Hibernate.
- Fer servir un patró de visualització com struts.

### **Enfocament i mètode seguit**

L'enfocament del projecte ha sigut bastant lliure per part meva: he pogut triar el tema del TFC, patrons per a l'emmagatzemament i Model-Controlador-Vista. Pel que fa al mètode seguit, he intentat complir amb les pautes especificades pel consultor i temporització:

- Primer he realitzat un calendari per a programar les tasques cronològicament (temporització), i la formalització del tema del TFC amb els objectius d'aquest. Això correspon a la PAC1
- En segon lloc, complint amb la temporització, he seguit un cicle de vida en cascada. On, després de la introducció i temporització de la PAC1, s'entrega la PAC 2 amb l'anàlisi i disseny del projecte.
- En tercer lloc, s'implementa el codi.

Tot això sempre amb l'ajuda i tutorització del consultor.



## Planificació del projecte

---

### 1 Pla de treball. Del 28/02/07 fins al 12/03/07 (12 dies)

- 1.1 Establir objectius
- 1.2 Definició de tasques i temporització
- 1.3 Redacció del document
- 1.4 **ENTREGA PAC 1 (12/03/07)**

### 2 Anàlisi 13/03/07 - 29/03/07 (17 dies)

- 2.1 Casos d'ús i diagrames dinàmics
- 2.2 Definició dels paquets d'anàlisi i de servei
- 2.3 Classes d'entitat
  - 2.3.1 Identificació de les classes i les seves relacions
  - 2.3.2 Diagrama estàtic
- 2.4 Classes frontera i control
  - 2.4.1 Identificació
  - 2.4.2 Diagrames de col·laboració
- 2.5 Anàlisi de la interfície d'usuari
  - 2.5.1 Esquema global del lloc web
  - 2.5.2 Esboços de formularis i altres

### 3 Disseny. Del 30/03/07 al 16/04/07 (18 dies)

- 3.1 Estudiar i triar patrons necessaris
- 3.2 Instal·lació i configuració del SGBD
- 3.3 Revisió del diagrama estàtic
- 3.4 Disseny de la persistència
  - 3.4.1 Disseny conceptual de la BD - Diagrama ER
  - 3.4.2 Disseny lògic de la BD - Model Relacional
  - 3.4.3 Script de creació de la BD
- 3.5 Documentació
- 3.6 **ENTREGA PAC 2 (16/04/07)**

#### **4 Implementació 17/04/07 - 14/05/07 (28 dies)**

- 4.1 Instal·lació i configuració del programari per al desenvolupament
- 4.2 Implementació de la persistència
- 4.3 Classes de control (beans i altres)
- 4.4 Implementació de subsistemes

#### **5 Proves. Del 15/05/07 al 21/05/07 (7 dies)**

- 5.1 Proves unitàries dels subsistemes
- 5.2 Proves generals
- 5.3 Correcció d'errors i excepcions
- 5.4 **ENTREGA PAC 3 (21/05/07)**

#### **6 Memòria. Del 22/05/07 al 18/06/07 (28 dies)**

- 6.1 **LLIURAMENT FINAL (18/06/07)**

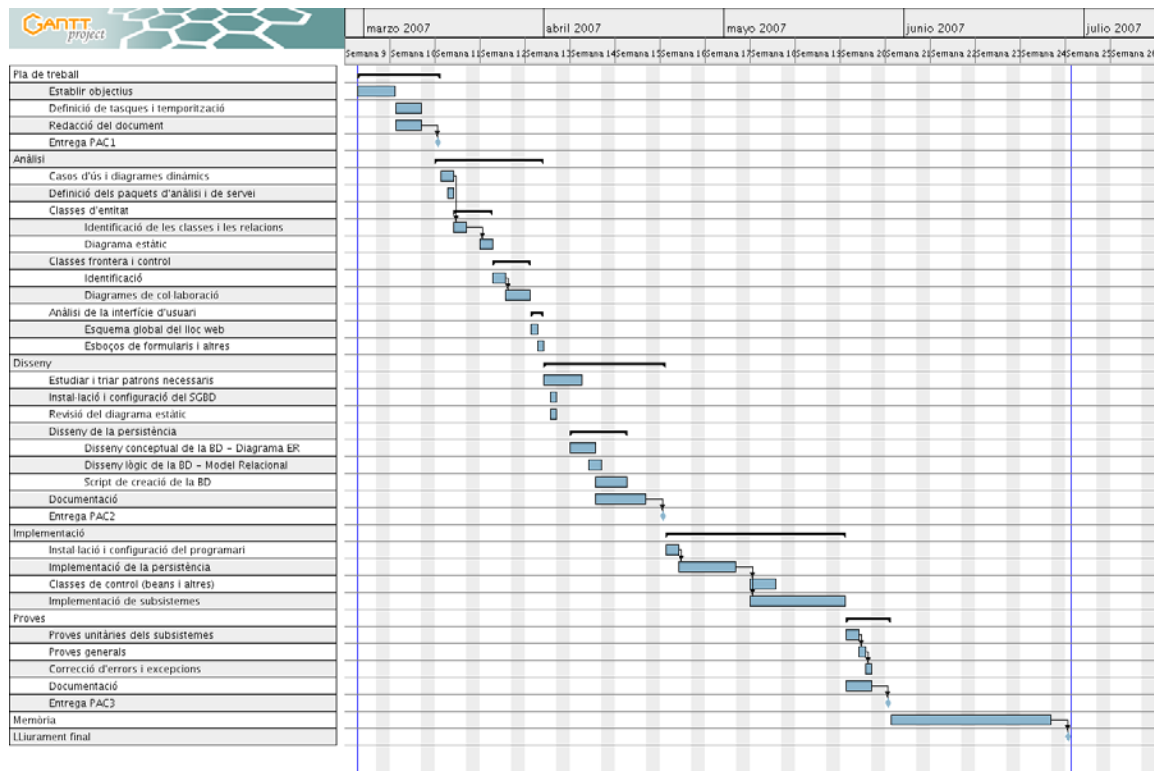


Figura 1. Temporització de les tasques.

## Productes obtinguts

El producte obtingut és una aplicació web per a gestionar gosses: adopcions, denúncies de pèrdues de gossos. Permetrà consultar gossos perduts (no residents a la gossera), com gossos abandonats (residents a la gossera i no reclamats).

A banda del codi obtingut, serà necessari a part el SGBD amb el qual treballarà.

## Descripció dels altres capítols de la memòria.

Als següents capítols es detalla aspectes com l'anàlisi i disseny seguit. S'indica com procedir per a la instal·lació i posada en marxa del projecte. També es comenta l'estructura de la base de dades com el codi necessari per a crear-la.

## Anàlisi

### Actors i guions

---

Intervenien dos tipus d'actors: usuaris i administradors

- **Usuaris:** són les persones que mitjançant el navegador d'internet accediran a la aplicació per a consultar els continguts. N'hi ha de dos tipus, els registrats (i identificats) i els que no identificats. Per a fer tasques on es necessiti saber dades sobre l'usuari que realitza l'acció, es requerirà que s'identifiqui i que es registri si fos necessari.

Es vol que l'aplicació permeti veure els gossos que hi ha a la gossera (abandonats) i els perduts (donats d'alta per usuaris que volen trobar-los). Sobre aquestes cerques, seria interessant filtrar segons característiques importants com ara sexe, tipus de pel, grandària del gos, etc.

Els usuaris amb gos perdut volen rebre notificacions de l'administrador de la gossera si entra un gos que s'assembla al seu.

Els clients volen poder reservar un gos per a adoptar-lo. Cal que un gos estigui un temps prudencial a la gossera abans de donar-lo en adopció, per això és la reserva.

- **Administrador:** persona que normalment és el que porta la protectora d'animals. S'encarrega de fer les tasques de gestió:  
Donar d'alta i de baixa els gossos que entren i surten de la gossera, afegir notícies importants, comprovar si un gos que entra a la gossera es sembla a algun gos desaparegut, etc.

Aquesta persona també és la que rep les notificacions de reserves de gossos per part dels usuaris. S'ha de encarregar de, quan el gos passi el temps prudencial a la gossera, trucar a l'usuari per a que vingui a buscar-lo.

També ha de poder realitzar les mateixes operacions que un usuari normal a dins de l'aplicació

**Model del domini**

A primera vista s'identifiquen les següents classes i les seves relacions:

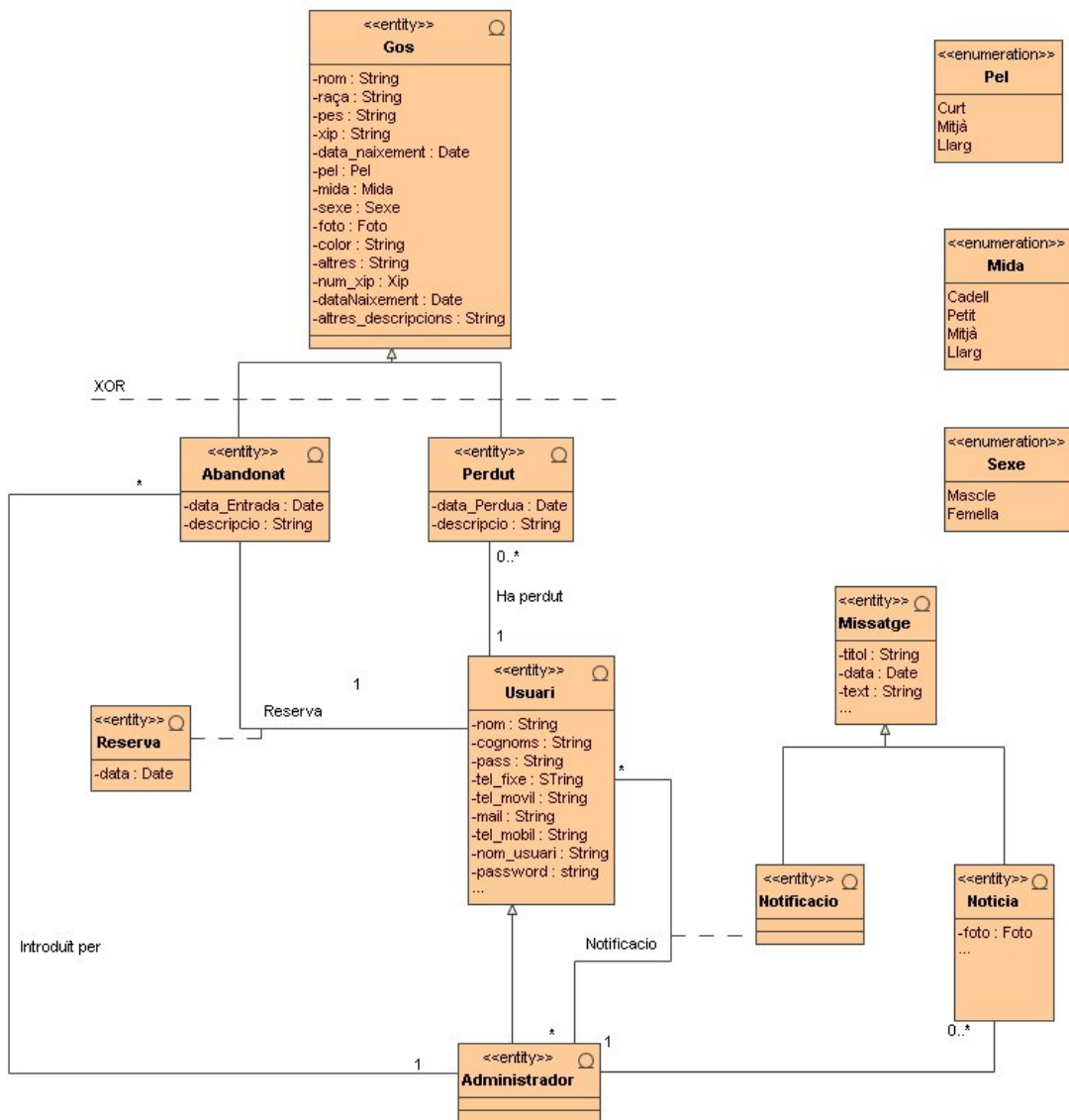


Figura 2. Diagrama de classes



Els casos d'ús relacionats amb l'administrador:

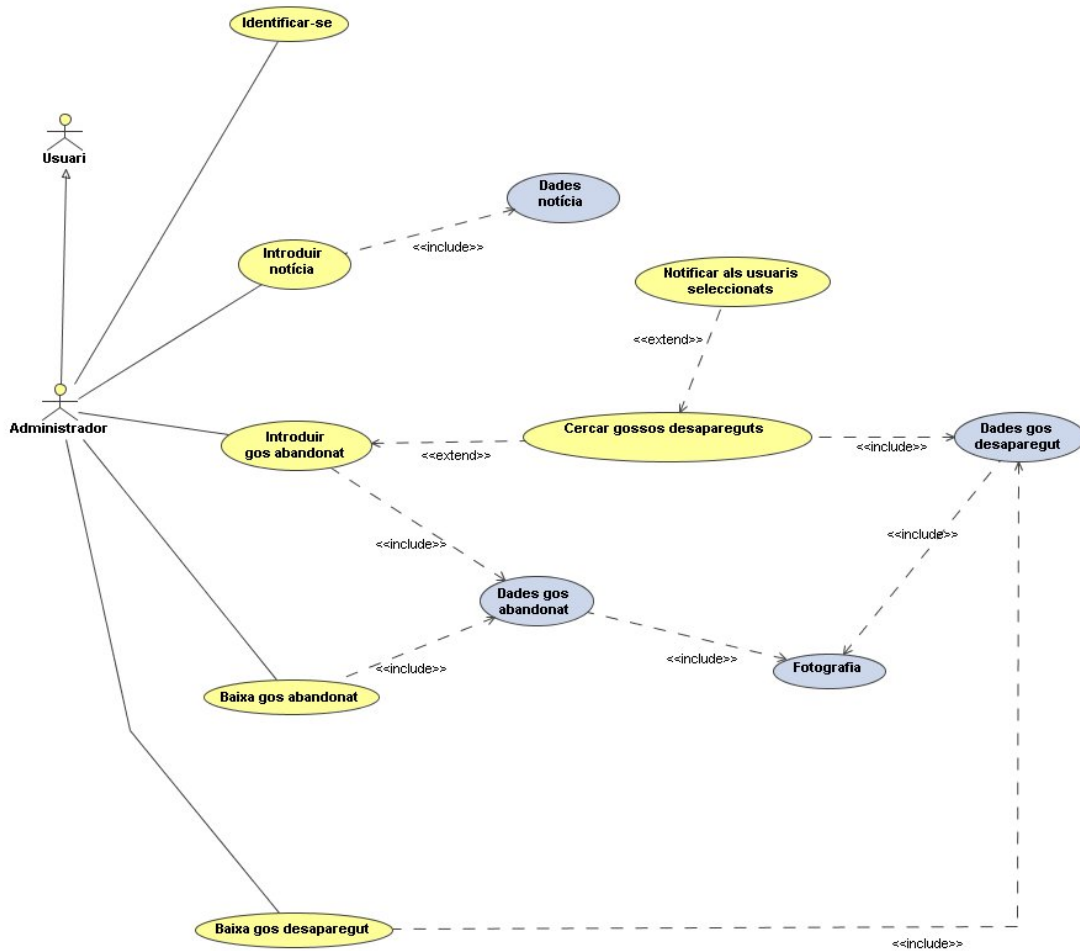


Figura 4. Diagrama de cassos d'us d'administrador



## Descripció textual dels casos d'ús més importants

---

### Identificar-se

**Resum:** permet identificar-se com a usuari al sistema i reconèixer el perfil d'usuari que li permetrà fer unes operacions determinades i visualitzar les seves notificacions.

**Paper dins el treball de l'usuari:** és necessari per a fer operacions com adopció i altres.

**Actors:** Usuari, Administrador

**Casos d'ús relacionats:** Reservar per adopció, Notificar pèrdua, Introduir notícia, Introduir gos abandonat, Baixa gos, Registrar-se

**Precondició:** El sistema està preparat per a validar l'usuari.

**Postcondició:** L'usuari introduït pertany al sistema i en té accés.

**Procés normal principal:**

1. L'usuari introdueix el seu nom d'usuari i contrasenya.
2. El servidor comprova el nom i contrasenya.
3. El servidor inicia la sessió i mostra la pantalla principal, mostrant les notificacions personals.

**Alternatives de procés i excepcions:** si no existeix l'usuari o la contrasenya no és vàlida, es mostra un error.

**Qüestions que cal aclarir:** només el cas d'us Llistar gossos es pot realitzar sense identificació. Tots els altres requereixen d'identificació (i registre si és necessari).

## Llistar gossos

**Resum:** llista els gossos disponibles a la BD.

**Paper dins el treball de l'usuari:** Permet obtenir informació dels gossos per tal de reservar o mirar els desapareguts.

**Actors:** Usuari, Administrador

**Casos d'ús relacionats:** Reservar per adopció

**Procés normal principal:**

1. L'usuari estableix els filtres com el pel, mida, sexe, etc.
2. El servidor retorna una llista dels gossos amb les característiques escollides. Per a cada element, apareixerà una opció d'adoptar o reservar en cas de que el gos porti poc temps a la gossera.

**Alternatives de procés i excepcions:** Si no existeix cap animal amb les característiques desitjades, s'ha de notificar.

## Registrar-se

**Resum:** permet crear un perfil d'usuari per a tenir accés a les activitats del lloc web.

**Paper dins el treball de l'usuari:** és necessari per als casos on es demana identificació

**Actors:** Usuari

**Casos d'ús relacionats:** Identificar-se

**Precondició:** El sistema està preparat per a registrar un usuari.

**Postcondició:** Les dades introduïdes són vàlides i correctes

**Procés normal principal:**

1. L'usuari introdueix les seves dades personals, nick i password
2. El sistema les valida i l'identifica automàticament

**Alternatives de procés i excepcions:** si alguna dada és incorrecta, es notifica a l'usuari per a que introdueixi els camps correctament

**Qüestions que cal aclarir:** hi haurà camps obligatoris, com per exemple e-mail, nick, password, telèfon, etc.

## Reserva per a adopció

**Resum:** permet reservar un gos per a recollir-lo posteriorment

**Actors:** Usuari

**Casos d'ús relacionats:** Registrar-se, Identificar-se, Llistar gossos.

**Precondició:** L'usuari s'ha identificat.

**Postcondició:** L'usuari no té altres reserves pendents

**Procés normal principal:**

1. L'usuari clica sobre la comanda per reservar-lo

2. El sistema envia una notificació a l'administrador amb les dades de l'usuari i el gos

**Alternatives de procés i excepcions:** si l'usuari alguna reserva pendent, no li apareixerà la opció d'adoptar en les llistes.

**Qüestions que cal aclarir:** la reserva per a adopció serveix per potenciar l'acollida. Si el gos porta un temps prudencial a la gossera es considera abandonat i es pot anar a recollir. Si el gos ha entrat fa poc, es considera perdut i cal esperar un temps. D'aquesta manera l'usuari pot disposar del gos passat aquest temps.

### Notificar pèrdua

**Resum:** si un usuari ha perdut el seu gos, pot deixar constància a la web i serà visible en les llistes. L'usuari pot rebre notificacions de l'administrador si en troba un que s'assembli.

**Actors:** Usuari, Administrador

**Casos d'ús relacionats:** Identificar-se, Registrar-se, Introduir gos abandonat

**Precondició:** Cal que l'usuari estigui identificat, i per tant, registrat.

**Postcondició:** Les dades introduïdes són correctes.

### Procés normal principal:

1. L'usuari clica sobre el botó o link corresponent
2. El sistema mostra un formulari per a que l'usuari l'ompli
3. L'usuari introdueix les característiques, nom i foto del gos
4. El sistema comprova que les dades siguin correctes
5. El sistema emmagatzema les dades del gos perdut i les fa accessibles

**Alternatives de procés i excepcions:** si alguna dada és incorrecta, tant els camps requerits com la imatge pujada, es notifica a l'usuari per a que introdueixi els camps correctament o imatge. Amb el següent diagrama d'activitats es mostra la relació que hi ha entre aquest cas i el d'introduir gos abandonat:

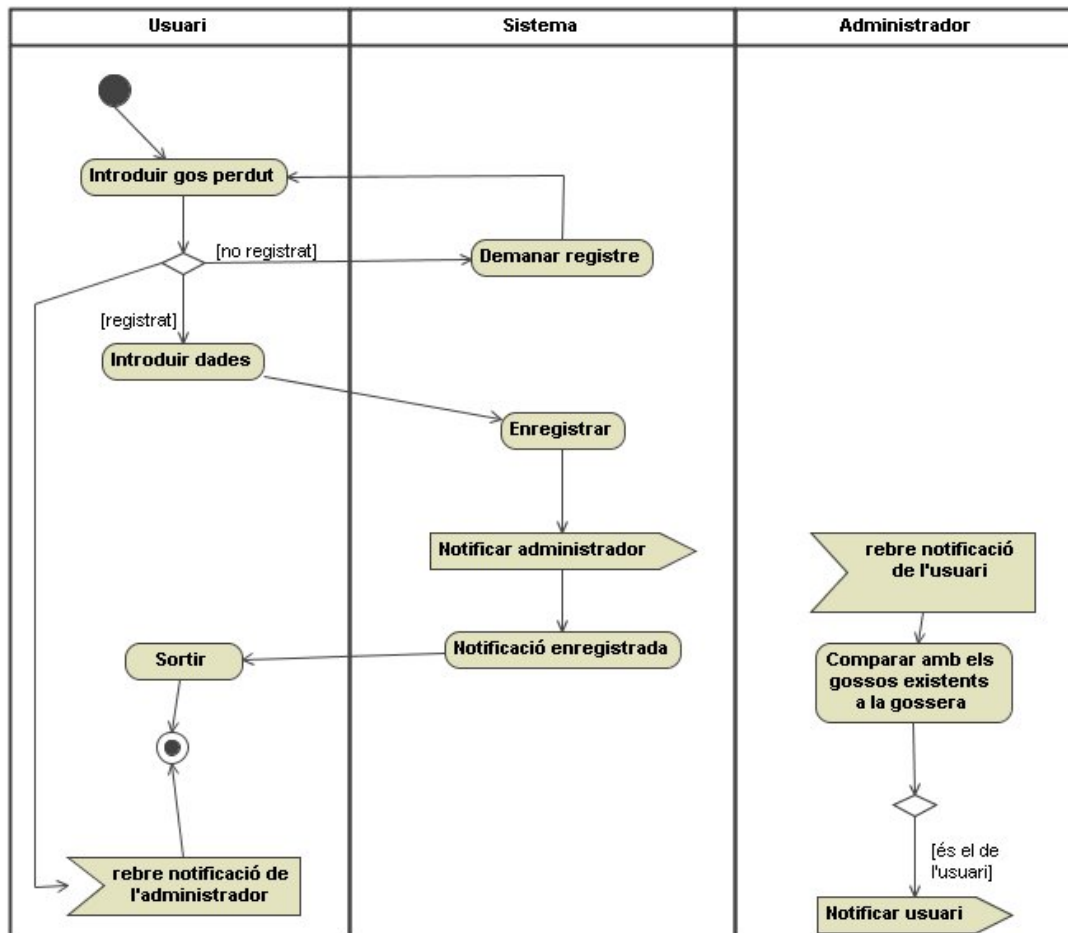


Figura 5. Diagrama d'activitat introduir gos perdut

## Visualitzar notificacions

**Resum:** permet veure les notificacions personals. En el cas dels usuaris, les notificacions provindran de l'administrador si s'ha trobat un gos semblant amb el que s'ha perdut. En el cas de l'administrador, la notificació serà sobre reserves d'adopció dels usuaris.

**Actors:** Usuari, Administrador

**Casos d'ús relacionats:** Identificar-se

**Precondició:** L'usuari s'ha identificat al sistema

**Postcondició:** cap.

**Procés normal principal:**

1. L'usuari accedeix a la seva àrea de notifikacions
2. El sistema mostra la llista de notifikacions

### Visualitzar notícies

**Resum:** permet veure les notícies sobre gossos i altres temes que va penjant l'administrador del sistema.

**Actors:** Usuari, Administrador

**Casos d'ús relacionats:** Introduir notícia,

**Precondició:** no és necessari que l'usuari s'identifiqui al sistema

**Postcondició:** cap.

**Procés normal principal:**

1. L'usuari accedeix a l'àrea de notícies
2. El sistema llista les notícies per ordre cronològic

## Introduir notícia

**Resum:** permet introduir una notícia que es podrà visualitzar a l'àrea de notícies corresponent.

**Actors:** Administrador

**Casos d'ús relacionats:** Visualitzar notícies

**Precondició:** L'usuari és administrador

**Postcondició:** Els camps de la notícia són correctes

### Procés normal principal:

1. L'administrador clica sobre "Introduir notícia"
2. El sistema mostra un formulari per a omplir amb els camps requerits
3. L'administrador pot incloure una imatge fent clic al botó corresponent i indicant el fitxer que correspon
4. L'administrador ha d'especificar un tema i un text
5. L'administrador valida
6. El sistema processa la notícia i la emmagatzema, fent que sigui accessible per a tot tipus d'usuari

**Alternatives de procés i excepcions:** si la imatge introduïda no és correcta, es demanarà a l'usuari que pugi una altra que sigui correcta. Els camps tema i text són obligatori i cal omplir-los, si no són correctes, es notifica a l'usuari.

## Introduir gos abandonat

**Resum:** permet donar d'alta un gos que malauradament ha entrat a la gossera.

**Paper dins el treball de l'usuari:** és el treball principal de l'administrador de la gossera, necessari per tenir l'aplicació viva i actualitzada

**Actors:** Administrador

**Casos d'ús relacionats:** Baixa gos, llistar gossos, Cercar gossos desapareguts, Notificar pèrdua

**Precondició:** L'usuari és administrador.

**Postcondició:** Les dades introduïdes són vàlides i correctes

### Procés normal principal:

1. L'administrador clica sobre "Introduir gos abandonat"
2. El sistema mostra un formulari per a omplir amb els camps requerits relacionats amb les característiques físiques del gos, data de naixement estimada, data d'entrada, etc.
3. L'administrador inclourà la imatge del gos (camp obligatori)
4. L'administrador accepta el formulari i el sistema valida les dades emmagatzemant-les i fent-les accessibles
5. El sistema pregunta a l'administrador si vol cercar gossos desapareguts amb semblances (pel tipus de pel, mida, sexe, etc). Aquesta acció correspon amb el cas d'ús "Cercar gossos desapareguts" que s'explica més endavant.

**Alternatives de procés i excepcions:** es demanarà que es torni a escriure els camps obligatoris i la imatge si la enviada no fos correcte.



**Qüestions que cal aclarir:** aquest apartat és bàsic, donat que les dades dels gossos (que són les principals) entren per aquest cas d'ús. Té una forta relació amb el cas d'ús Notificar pèrdua:

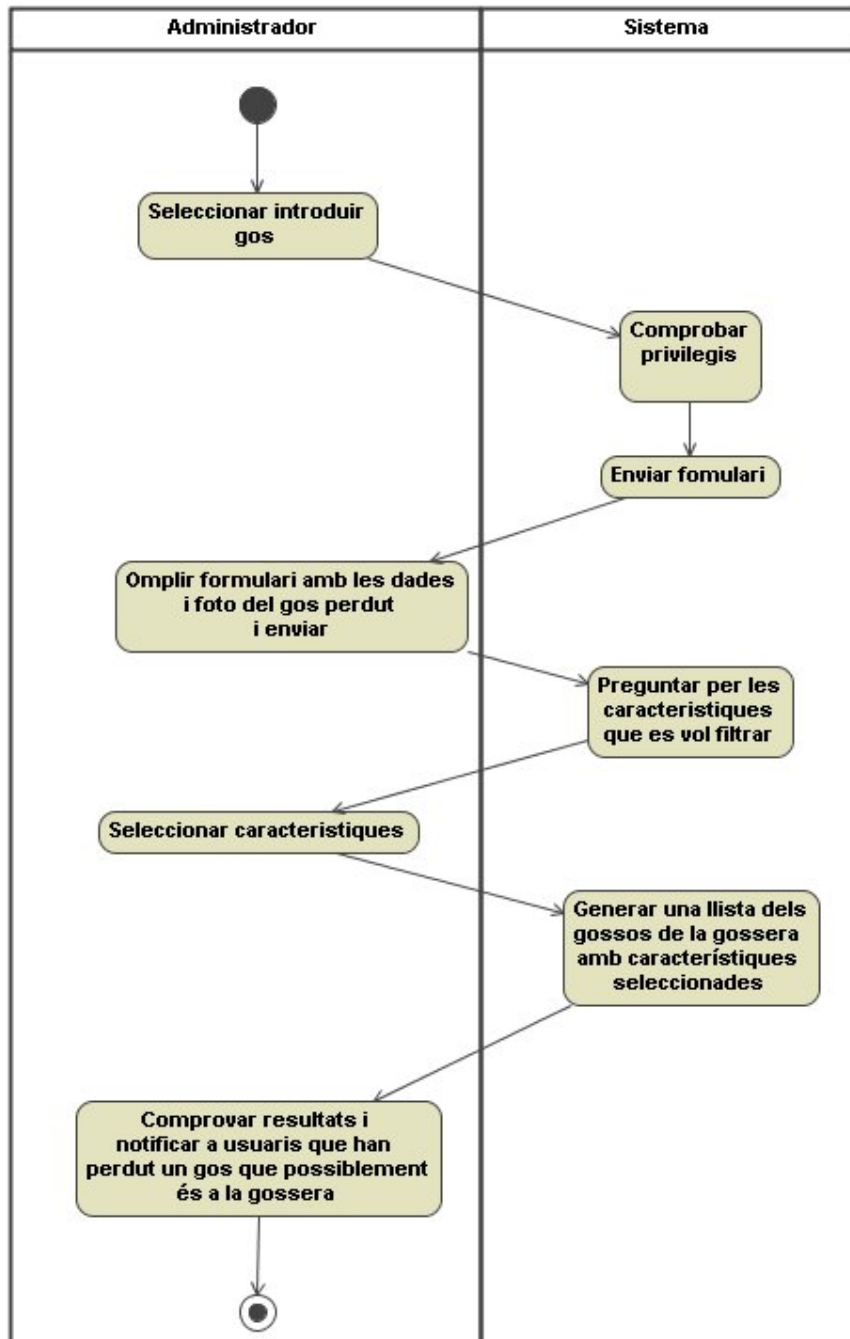


Figura 6. Diagrama d'activitat introduir gos abandonat.

## Cercar gossos desapareguts

**Resum:** serveix per a cercar gossos desapareguts que han introduït els usuaris

**Paper dins el treball de l'usuari:** permet a l'administrador comparar els desapareguts amb els gossos que té a la gossera.

**Actors:** Administrador

**Casos d'ús relacionats:** Notificar pèrdua, Introduir gos abandonat

**Precondició:** L'usuari és l'administrador

**Postcondició:** cap

### Procés normal principal:

Si es prové de "Introduir gos abandonat":

1. Es mostra una llista per seleccionar les semblances del gos introduït que es volen tenir com a filtre
2. Un cop seleccionades i validat el formulari, es mostra la llista amb els gossos.

Si és engegat directament per l'usuari:

1. Es mostra un formulari per a seleccionar diferents característiques
2. Es mostra la llista dels gossos.

**Alternatives de procés i excepcions:** Si no existeix cap animal amb les característiques desitjades, s'ha de notificar.

## Diagrames de col·laboració

Per als casos d'ús més importants, es descriu els diagrames de col·laboració on es distingeixen les classes d'entitat, control i frontera:

### Introduir gos abandonat:

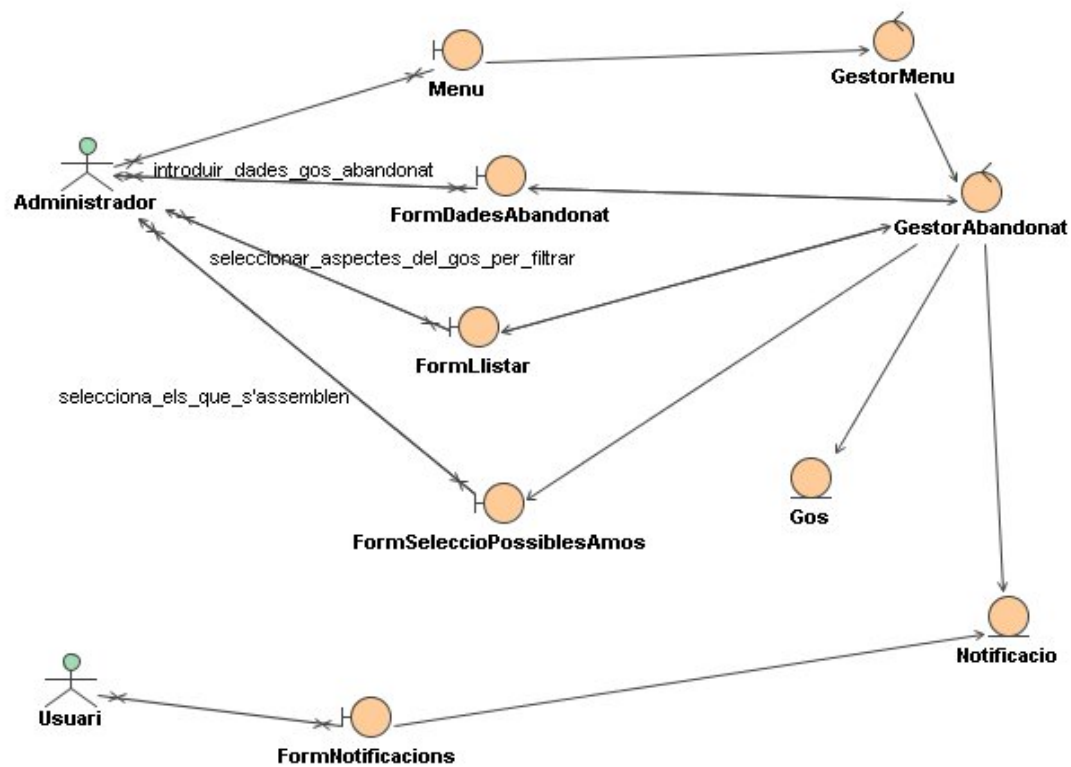


Figura 7. Diagrama de col·laboració: introduir gos abandonat

### Llistar gossos:

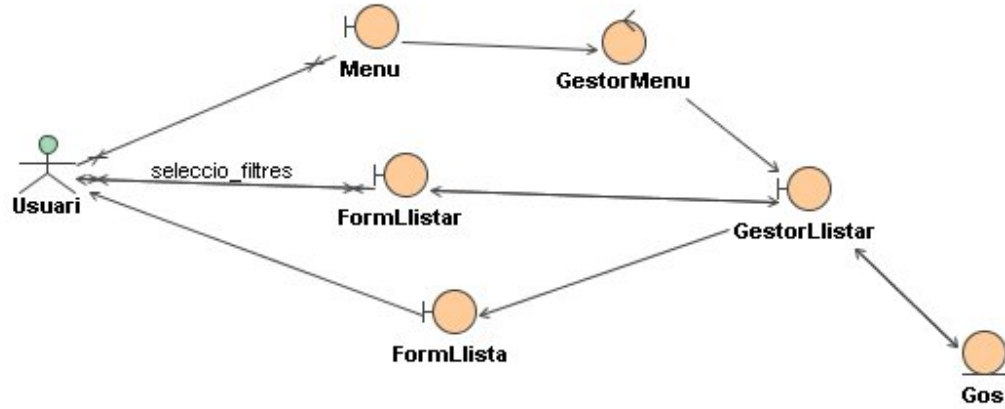


Figura 8. Diagrama de col·laboració: llistar gossos

### Visualitzar notificacions:

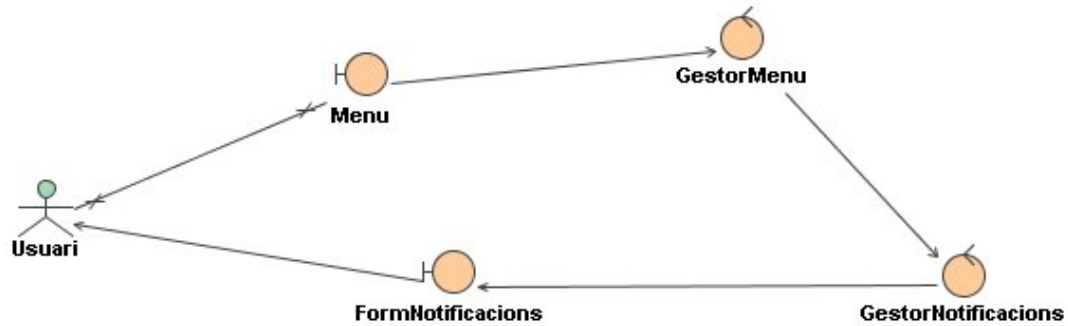


Figura 9. Diagrama de col·laboració: visualitzar notificacions

### Notificar perdua

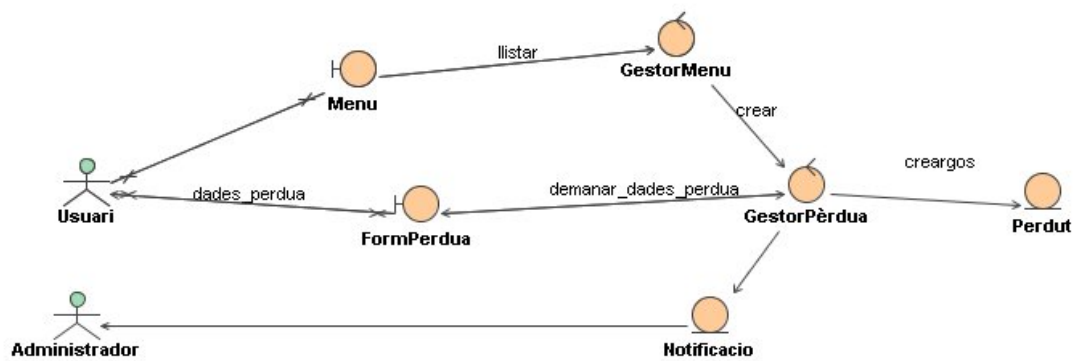


Figura 10. Diagrama de col·laboració: notificar perdua

## Glossari del model de negoci

- Gos perdut: és el gos que tenim al sistema com a perdut, però no està físicament al a gosserra. Els gossos perduts són introduïts pels usuaris que els volen trobar.
- Gos abandonat: són els gossos que es troben físicament a la gosserra a l'espera de que els adoptin.

**DISSENY****Disseny arquitectònic**

L'esquema general seria el següent:

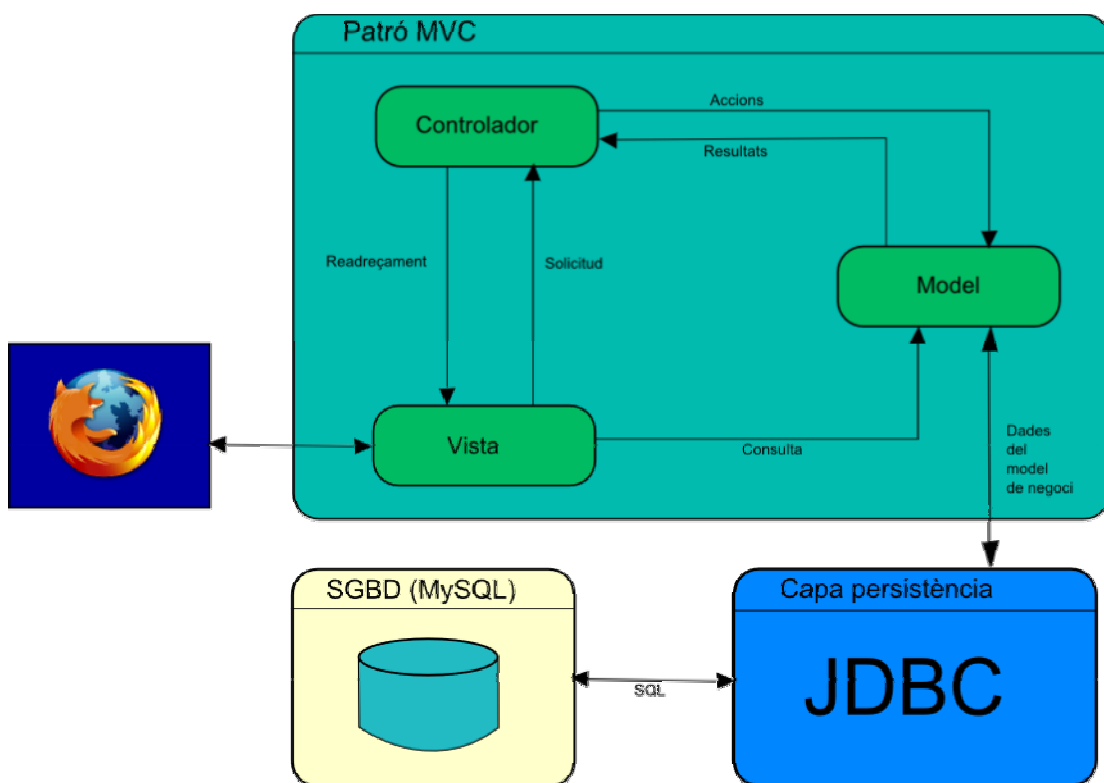


Figura 11. Esquema general de l'aplicació

Justificacions:

**J2EE:** Plataforma de programació que ens permet desenvolupar aplicacions en llenguatge Java. És una plataforma prou sòlida on hi ha solucions per a molts problemes que es presenten al món empresarial. També té un punt molt fort: és gratuït.

**Jacarta Struts** és un framework que implementa el patró MVC2 mitjançant les API's de servlets i JSPs de J2EE. Struts s'ha convertit en un framework molt utilitzat i existeix molta documentació a internet. He optat per aquest patró perquè simplifica la feina separant els tres aspectes bàsics d'una aplicació: Model, Controlador i Vista:

- **Controlador:** es representat per un servlet que li correspon el control i coordinació de les activitats de l'aplicació:
  - Mapeja les accions
  - Selecciona les vistes corresponents a les respostes
  
- **Model:** correspon a la lògica del negoci. Les classes que componen aquest model són les que connecten amb la base de dades mitjançant JDBC. Realitza les següents funcionalitats:
  - Notifica canvis en les vistes
  - Encapsula l'estat de l'aplicació
  - Proporciona dades sobre l'estat del model de negoci.
  
- **Vista:** correspon a les JSP's que conformen l'interfície d'usuari.

**SGBD: MySQL**. És un SGBD relacional molt extès. Ens permetrà emmagatzemar i recuperar les dades amb seguretat. He triat aquest SGBD donat que ha madurat molt a partir de la versió 5, on ja inclou procediments emmagatzemats, triggers, etc. També és programari lliure, on es pot descarregar el SGBD, front-ends, drivers per java, etc. A la web es proporciona una àmplia documentació en castellà.

**Capa d'accés a les dades**, s'utilitza la API JDBC. És una tecnologia que ens aporta totes les classes i interfícies necessàries per a obtenir connectivitat amb la bases de dades, accés a les dades, etc. Aquesta API està inclosa dins del JDK. Cal aportar el driver necessari per a cada SGBD.

## Revisió del diagrama estàtic de disseny

---

- S'elimina l'herència entre Missatge i Notificació i Notícia, eliminant Missatge. No es preveu que sigui necessari el polimorfisme entre aquestes entitats.
- S'han eliminat les classes associatives Reserva i Notificació, fent associacions N-àries
- Encara que es podria eliminar l'herència d'Abandonat i Perdut amb Gos afegint un parell d'atributs a Gos, cal tenir diferenciades aquestes dues classes perquè pot ser necessari afegir atributs i/o mètodes exclusius de cadascun.

El diagrama queda de la següent manera:



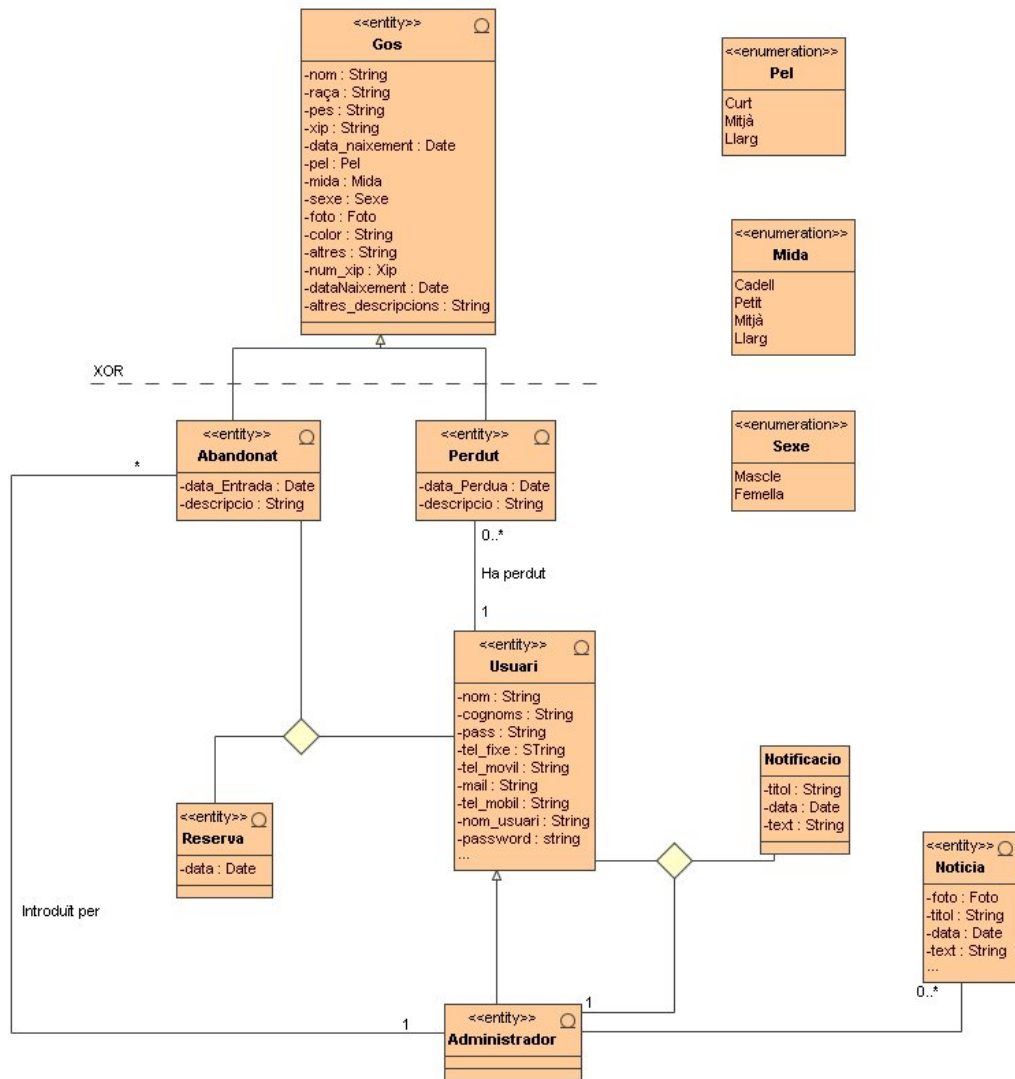


Figura 12. Revisió de diagrama estàtic

**Disseny de la persistència**

El diagrama E-R:

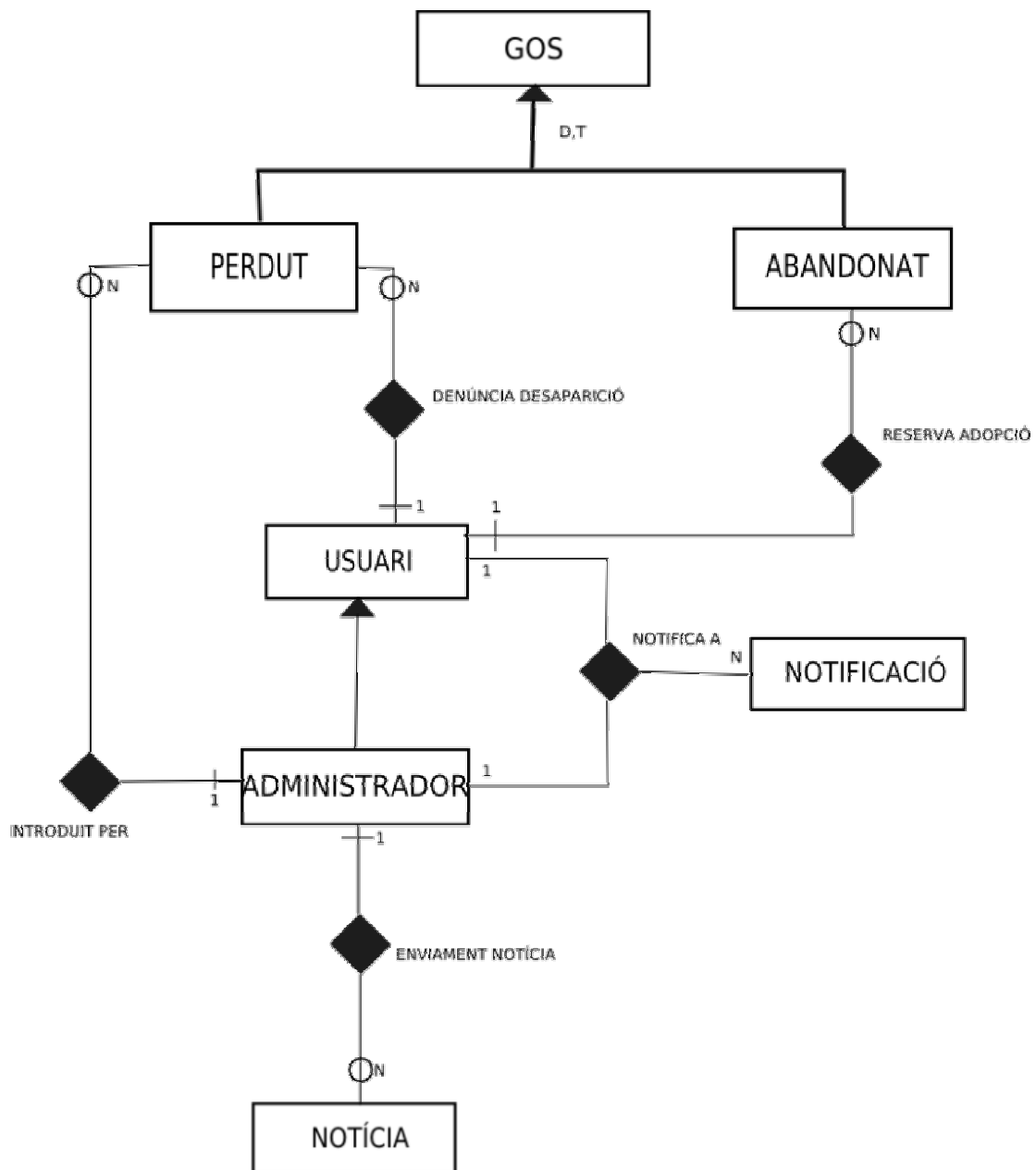


Figura 13. Diagrama ER

Model relacional:

**Gos**(id, nom, raça, pes, xip, data\_naix, pel, mida, sexe, path\_foto, color, comentaris).

- id és clau primària

**Abandonat**(id\_ab, id\_gos, data\_entrada, descripció)

- id\_ab és clau primària
- id\_gos és clau forana de Gos

**Perdut**(id\_per, id\_gos, data\_entrada, id\_usuari, descripció)

- id\_per és clau primària
- id\_gos és clau forana de Gos
- id\_usuari és clau forana d'Usuari

**Reserva**(id\_reserva, id\_ab, id\_usuari, data\_reserva, descripció)

- id\_reserva és clau primària
- id\_ab és clau forana d'Abandonat
- id\_usuari és clau forana d'Usuari

**Notificació**(id\_noti, titol, data, text, id\_rem, id\_des)

- id\_noti és clau primària
- id\_rem és clau forana d'Usuari
- id\_des és clau forana d'Usuari

**Notícia** (id\_notícia, path\_foto, titol, data, text, id\_usr)

- id\_notícia és clau primària
- id\_usr és clau forana d'Usuari

**Usuari** (id\_usuari, nom\_usuari, password, nom, cognoms, tel\_fixe, tel\_mòvil, email)

- id\_usuari és clau forana

**Administrador** (id\_usuari, nom\_gossera, telefon, adreça)

- id\_usuari és clau primària
  - id\_usuari és clau forana d'Usuari
-

## Estructura de la interfície d'usuari

---

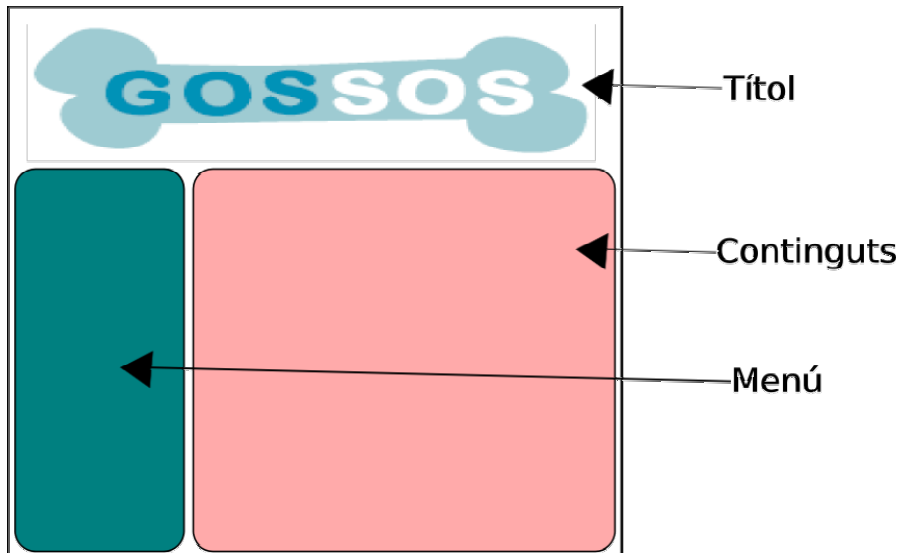


Figura 14. Estructura de la interfície principal d'usuari

### **Menú:**

Contindrà els links següents links dels casos d'ús principals, aquestos poden variar segons si l'usuari s'ha identificat o no, i també si l'usuari identificat és administrador.

- Si l'usuari no està identificat:
  - Llistar
  - Notícies
  - Registrar-se
  - Identificar-se
  
- Si l'usuari està identificat
  - Llistar
  - Notícies
  - Notificar pèrdua
  - Sortir
  - Bústia (veure notificacions)

- Si l'usuari està identificat, i a més, és administrador s'afegiran les següents opcions a banda de les d'un usuari identificat
  - Introduir notícia
  - Alta gos
  - Baixa gos
  - Esborrar perduts
  - Veure adopcions

**Continguts:**

És l'àrea on es visualitzen les interfícies d'usuari.

## Instal·lació

---

Per a posar l'aplicació en marxa, cal tenir instal·lats els següents programes instal·lats:

1. JDK 1.4 o superior.
2. MySQL Server 5
3. Tomcat 5.5

Un cop instal·lat tot això, cal deixar l'arxiu `bd.properties` al mateix directori de treball del Tomcat. Per defecte, aquest directori és el `./bin` del directori d'instal·lació. En aquest fitxer es defineixen els atributs per a connectar l'aplicació amb l'SGBD com el host, port, nom de la base de dades, etc.

Cal crear la base de dades, usuaris i taules a l'SGBD. Per a fer això, només cal executar l'arxiu `gossos.sql` amb **SQL Command Line**:

```
Mysql> source c:/path/gossos.sql
```

Després, cal copiar l'aplicació web al directori de treball del Tomcat. Tota l'aplicació es troba en l'arxiu comprimit `gossos.war`. El copiarem al directori `/webapps` del directori `apache-tomcat`. Cal reiniciar el servei tomcat.

Finalment, s'ha de tenir en compte el límit de les fotos que es vol deixar emmagatzemar a la BD. Aquest límit bé imposta pel MySQL, a la variable `max_packet_size`, que limita l'ample dels paquets per emmagatzemar els BLOB.

L'script que crea la BD `gossos.sql`, defineix un administrador per defecte:

- Nom d'usuari: admin
- Contrasenya: gossos

## Conclusions

---

En primer lloc, puc dir que he après moltes coses en aquest projecte, i d'altres coses que ja sabia les he pogut posar a la pràctica. J2EE és una plataforma molt important i demanada sol·licitada laboralment, i aquest projecte m'ha acostat una mica al món real.

He aconseguit complir amb els requisits que vaig proposar amb el consultor tot i que la realització d'altres assignatures amb el projecte no m'han deixat dedicar-li el temps que es mereix. Amb tot això, per aconseguir tots aquests requisits, he hagut de sacrificar aspectes bàsics com:

- Utilització d'una capa de persistència com ara Hibernate. Això m'hagués ajudat a controlar les sessions dels usuaris sobre la base de dades i també obtenir més aïllament entre les classes d'entitat i la BD.
- Presència. La interfície d'usuari és pobre i s'hauria de millorar. Amb això veig que el HTML i CSS és tot un món i que és molt important tenir una bona base.
- Seguretat. Hi ha lleus aspectes de seguretat que calen tractar.
- Usabilitat. Caldria jugar més amb les interfícies, quadres de diàleg, etc, per a fer-la més fàcil, intuïtiva i agradable.

A banda d'aquests aspectes, m'adono de la importància d'una bona planificació i coneixement de les distintes àries en les que s'ha de treballar.

Tot plegat, he pogut veure les possibilitats d'aquesta plataforma en quant a productivitat, possibilitats i baixos costos en comparació amb altres com ara .NET



### Què s'hauria de millorar

Els punts que podria millorar en un futur són els esmentats anteriorment com afegir capa de persistència amb hibernate, millora de la presentació i usabilitat. I per últim i molt important, vigilar molt la seguretat.

## **Bibliografia**

---

### Libres

- Revista Solo Programadores, especial tutorial d'Struts
- Tecnologías de Servidor con Java: Servlets, JavaBeans, JSP. Grupo Eidos
- J2EE. Apunts de la UOC de Josep Maria Camps i Riba
- HTML, XHTML i CSS. Biblia
- HTML pocket reference

### Internet

- Lloc d'Struts: <http://struts.apache.org/>
- Manual d'Struts: [http://www.programacion.net/java/tutorial/joa\\_struts/](http://www.programacion.net/java/tutorial/joa_struts/)
- L'Api d'Struts: <http://www.programacion.net/java/tutorial/struts/>

## Annex A: Script de creació de taules

---

```
-- Creació i connexió a la BD

drop database gossosDB;
create database gossosDB;

-- connexió a la BD
use gossosDB;

begin work;

CREATE TABLE Usuari
( id_usuari    INTEGER UNSIGNED NOT NULL UNIQUE AUTO_INCREMENT
, nom_usuari  CHARACTER(10) NOT NULL
, password    CHARACTER(10) NOT NULL
, nom         CHARACTER(15) NOT NULL
, cognoms     CHARACTER(40)
, tel_fixe    CHARACTER(10)
, tel_mobil   CHARACTER(10)
, email       CHARACTER(50)
, PRIMARY KEY (id_usuari));

CREATE TABLE Administrador
( id_admin     INTEGER UNSIGNED NOT NULL UNIQUE
, nom_gossera CHARACTER(30) NOT NULL
, telefon     CHARACTER(10) NOT NULL
, adreca      CHARACTER(50)
, PRIMARY KEY (id_admin)
, FOREIGN KEY (id_admin) REFERENCES Usuari(id_usuari));

CREATE TABLE Imatge
(
    id_img      INTEGER UNSIGNED NOT NULL UNIQUE AUTO_INCREMENT
, mime         CHARACTER(20) NOT NULL
, foto         LONGBLOB NOT NULL
, PRIMARY KEY (id_img));

CREATE TABLE Gos
( id_gos       INTEGER UNSIGNED NOT NULL UNIQUE AUTO_INCREMENT
, id_img       INTEGER UNSIGNED NOT NULL UNIQUE
, nom          CHARACTER(15) NOT NULL
, raca         CHARACTER(25) NOT NULL
, pes          DECIMAL(3,1)
```

```

,xip          CHARACTER(25)
,data_naix   DATE NOT NULL
,pel        SMALLINT NOT NULL
,mida       SMALLINT NOT NULL
,sexe       BOOLEAN NOT NULL
,color      CHARACTER(50) NOT NULL
,comentaris CHARACTER(100)
,PRIMARY KEY (id_gos)
,FOREIGN KEY (id_img) REFERENCES Imatge(id_img) ON DELETE CASCADE);

```

```
CREATE TABLE Abandonat
```

```

( id_gos          INTEGER UNSIGNED NOT NULL UNIQUE
,id_usuari       INTEGER UNSIGNED NOT NULL
,data           DATE NOT NULL
,descripcio     CHARACTER(100)
,gabia         INTEGER
,caracter       CHARACTER(50)
,PRIMARY KEY (id_gos)
,FOREIGN KEY (id_usuari) REFERENCES Usuari(id_usuari)
,FOREIGN KEY (id_gos) REFERENCES Gos(id_gos) ON DELETE CASCADE);

```

```
CREATE TABLE Perdut
```

```

( id_gos          INTEGER UNSIGNED NOT NULL UNIQUE
,id_usuari       INTEGER UNSIGNED NOT NULL
,data           DATE NOT NULL
,descripcio     CHARACTER(100)
,PRIMARY KEY (id_gos)
,FOREIGN KEY (id_gos) REFERENCES Gos(id_gos) ON DELETE CASCADE
,FOREIGN KEY (id_usuari) REFERENCES Usuari(id_usuari));

```

```
CREATE TABLE Reserva
```

```

(
 id_reserva     INTEGER UNSIGNED NOT NULL UNIQUE AUTO_INCREMENT
,id_gos         INTEGER UNSIGNED NOT NULL
,id_usuari      INTEGER UNSIGNED NOT NULL
,data          DATE NOT NULL
,PRIMARY KEY (id_reserva)
,FOREIGN KEY (id_gos) REFERENCES Gos(id_gos) ON DELETE CASCADE
,FOREIGN KEY (id_usuari) REFERENCES Usuari(id_usuari));

```

```
CREATE TABLE Notificacio
```

```

(
 id_notifi     INTEGER UNSIGNED NOT NULL UNIQUE AUTO_INCREMENT
,titol        CHARACTER(50)
,data         DATE NOT NULL
,text         CHARACTER(255) NOT NULL
,id_remi      INTEGER UNSIGNED NOT NULL

```

```

, id_dest      INTEGER UNSIGNED NOT NULL
, PRIMARY KEY (id_notifi)
, FOREIGN KEY (id_remi) REFERENCES Usuari(id_usuari)
, FOREIGN KEY (id_dest) REFERENCES Usuari(id_usuari));

CREATE TABLE Noticia
(
  id_noticia  INTEGER UNSIGNED NOT NULL UNIQUE AUTO_INCREMENT
, id_img      INTEGER UNSIGNED NOT NULL UNIQUE
, titol       CHARACTER(50)
, data        DATE NOT NULL
, text        TEXT NOT NULL
, id_usuari   INTEGER UNSIGNED NOT NULL
, PRIMARY KEY (id_noticia)
, FOREIGN KEY (id_usuari) REFERENCES Usuari(id_usuari)
, FOREIGN KEY (id_img) REFERENCES Imatge(id_img) ON DELETE CASCADE);

DELIMITER //
CREATE PROCEDURE esborrarPerduts(IN messos INT)
BEGIN
  DELETE FROM Imatge
  WHERE id_img IN ( SELECT G.id_img
                   FROM Gos AS G, Perdut AS P
                   WHERE G.id_gos = P.id_gos AND
                         P.data < DATE_SUB(now(), INTERVAL messos MONTH));
END//
DELIMITER ;

INSERT INTO Usuari(nom_usuari,password,nom,cognoms,tel_fixe,tel_mobil,email) VALUES
('admin','gossos','Jordi','Sanchez Cano','977233223','667656554','jsanchezcan@uoc.edu');
INSERT INTO Administrador VALUES (1,'Gossera Mollet','935688449','C\Caldes de montbui, 12
Mollet');

CREATE USER 'gossos_user' IDENTIFIED BY 'gossos';
GRANT ALL ON gossosDB.* TO gossos_user;
#GRANT EXECUTE ON esborrarPerduts TO gossos_user;

commit work;

```