



## Cercador de Protectores Animals “Adopta’m”

**Sandra Iglesias Rodríguez**  
Grau Enginyeria Informàtica  
TFG – Java EE

**Carlos Álvarez Matrinez**  
**Albert Grau Perisé**

14/06/2016



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

# Taula de Continguts

1.	Introducció.....	4
2.	Descripció del projecte .....	5
2.1	Descripció general.....	5
2.2	Definició dels objectius.....	6
	Objectiu principal.....	6
	Objectius transversals .....	6
2.3	Etapas del projecte.....	7
3.	Anàlisi de requisits .....	9
3.1	Requisits funcionals .....	10
3.2	Requisits no funcionals .....	10
4.	Especificació .....	12
4.1	Actors del sistema .....	12
4.2	Diagrama de casos d'ús.....	13
4.3	Especificació dels casos d'ús.....	14
4.4	Model conceptual .....	17
5	Arquitectura.....	21
5.1	Patró Model-Vista-Controlador.....	21
	Tecnologies de les capes.....	21
5.2	Disseny tècnic .....	22
	Capa de presentació .....	23
5.3	Tecnologies.....	24
6	Implementació.....	26
6.1	Estructura del projecte .....	26
6.2	Canvis en el disseny .....	27
	Funcionalitat no implementada .....	27
	Canvis en la base de dades .....	28
	Canvis de la capa de presentació .....	28
6.3	Futures millores.....	29
	Estructurals .....	29

Funcionals.....	29
7 Aplicació.....	30
7.1 Funcionalitats de l'aplicació.....	30
Pantalla principal.....	30
Registre d'una associació .....	30
Pantalla d'accés .....	30
Perfil de l'associació.....	31
Dades dels gossos.....	31
Llistat d'usuaris .....	32
7.2 Desplegament.....	33
Preparació de l'entorn .....	33
Desplegament.....	41
8 Manual d'usuari.....	42
8.1 Pantalla principal.....	42
8.2 Pantalla de registre .....	45
8.3 Pantalla d'identificació.....	47
8.4 Pantalla del perfil de l'associació .....	48
8.5 Pantalla de gestió dels gossos.....	49
8.6 Pantalla de llistat d'usuaris.....	51
9 Conclusions.....	52
10 Bibliografia.....	53
Documentació consultada .....	53
"Com fer" i exemples .....	53
Obtenció de dades .....	54

# Índex d'il·lustracions

Il·lustració 1. Diagrama de casos d'us.....	13
Il·lustració 2. Diagrama de classes.....	18
Il·lustració 3. Diagrama de components .....	22
Il·lustració 4. Diagrama Capa de Presentació .....	23
Il·lustració 5. Estructura del projecte .....	26
Il·lustració 6. Ubicació arxiu d'hibernate .....	33
Il·lustració 7. Configuració hibernate usuari .....	33
Il·lustració 8. Hibernate creació taules .....	34
Il·lustració 9. Capçalera usuaris anònims.....	42
Il·lustració 10. Formulari de cerca .....	42
Il·lustració 11. Llistat de gossos .....	43
Il·lustració 12. Detall d'un gos .....	44
Il·lustració 13. Pantalla de registre .....	45
Il·lustració 14. Formulari de registre amb errors.....	46
Il·lustració 15. Pantalla d'identificació.....	47
Il·lustració 16. Capçalera usuaris registrats.....	47
Il·lustració 17. Menú usuaris registrats.....	48
Il·lustració 18. Pantalla perfil associació.....	48
Il·lustració 19. Pantalla gestió dels gossos.....	49
Il·lustració 20. Panell per afegir més informació.....	50
Il·lustració 21. Panell per afegir un gos .....	50
Il·lustració 22. Pantalla llistat usuaris .....	51

# 1. Introducció

Aquest document és la memòria del Treball Final de Grau que ha anomenat Cercador de Protectores Animals "Adopta'm" i que correspon al Grau d'Enginyeria Informàtica de la Universitat Oberta de Catalunya.

A grans trets, el propòsit d'aquest projecte és la creació i posada en funcionament d'un cercador que permeti a les associacions i protectores d'animals donar a conèixer els gossos que tenen en adopció conjuntament en una mateixa pàgina web, amb l'objectiu de facilitar als usuaris interessats en adoptar una via que els permeti trobar el gos que estaven buscant, sense haver de fer el treball xafogós de recórrer les pàgines web de cada associació.

## 2. Descripció del projecte

### 2.1 Descripció general

Aquest Treball de Fi de Grau neix de la necessitat no coberta de facilitar a la societat un cercador d'associacions i protectores animals pel bé de promoure el coneixement d'aquestes entitats i poder incrementar el nombre d'adopcions, que anirà creixent cada cop més, segons l'aplicació web es vagi fent més coneguda.

El propòsit és sense ànims de lucre i només pel bé dels animals i per ajudar a les persones que treballen dia rere dia pel seu benestar a que aconseguixin més adopcions.

El context actual per a les associacions és que arriben més gossos a les protectores dels que es donen en adopció, degut en part pel desconeixement de les persones de l'existència d'aquest tipus d'entitats, per la poca informació sobre l'adopció d'un animal i també en part, per no tenir a l'abast la informació de totes les associacions pròximes a tu.

La situació, per tant, fa que les persones que estan interessades en adoptar busquin a l'associació de la seva ciutat o en les que coneguin, però pel desconeixement potser no visitin la que es troba a la ciutat del costat.

Actualment existeixen cercadors que donen resposta a la cerca d'habitatge o de viatges entre d'altres, però no hi ha cap enfocat a resoldre la problemàtica que tenen les associacions per donar en adopció a tots els seus animals.

## 2.2 Definició dels objectius

### Objectiu principal

L'objectiu principal d'aquest projecte és realitzar una aplicació web que funcioni com a cercador de protectores animals, concretament les associacions i protectores podran registrar-se i incloure els gossos que tinguin en adopció juntament amb una descripció d'aquests, alguna fotografia... i per altra banda, els usuaris podran cercar les protectores tot indicant una població o bé podran cercar un gos en concret indicant-ne la seva raça.

Aquest objectiu té com a finalitat millorar la situació actual dels animals en adopció, fer que les associacions siguin més conegudes i estigui aquesta informació a l'abast de tothom per així aconseguir un increment en el nombre d'adopcions de totes les associacions.

### Objectius transversals

Aquest projecte pretén posar en pràctica molts dels coneixements adquirits durant el grau d'enginyeria informàtica de forma transversal al aplicar-ho en la creació d'una aplicació web i també l'aprenentatge de noves eines que solen utilitzar-se en els projectes reals.

Com a objectius específics es trobarien l'aprenentatge de:

- Portar a terme un projecte de forma individual, de principi a fi.
- Realització d'una aplicació completa i funcional amb la consegüent obtenció d'un producte finalitzat.
- Gestionar un projecte tenint en compte les fases de gestió d'un projecte de programari.



I l'experiència que s'adquireix al llarg del projecte amb la pràctica i l'ús de les diferents eines:

- L'aprenentatge de l'ús de noves eines com ha sigut el cas del framework d'hibernate i el de primefaces.
- L'ampliació del coneixement d'eines conegudes com la tecnologia de Java Server Faces (JSF) o la base de dades de PostgreSQL.

## 2.3 Etapes del projecte

Com en tots els desenvolupaments de programari és necessari fer una planificació i portar a terme una gestió del projecte per poder portar-ne un seguiment i així garantir que es compleixen les fites. També és imprescindible conèixer les fases que comprèn un projecte per seguir un ordre i assegurar-ne la coherència.

En aquest cas, pel que fa a la planificació i degut a la combinació de diferents tecnologies unit amb la poca experiència en aquest àmbit, hi ha hagut certs problemes a l'hora de determinar el temps que em portaria cada tasca.

Les fases que he seguit han sigut les inherents a qualsevol gestió d'un projecte de programari que en ordre serien:

Fase	Descripció
<b>Anàlisi</b>	Investigació i definició de tots els requisits que s'han de complir en l'aplicació final.
<b>Especificació</b>	Anàlisi i especificació de les funcionalitats que garantiran que els requisits es compleixen.
<b>Disseny</b>	Refinament de l'especificació de la fase anterior i disseny de l'estructura de dades.
<b>Implementació i Proves</b>	Implementació del codi del projecte tenint en compte els anàlisis i dissenys de les fases anteriors i la realització de les proves.

### 3. Anàlisi de requisits

Els requisits principals que hauria de complir l'aplicació perquè fos útil tant per les protectores i associacions com pels usuaris han de contemplar un ús i aprenentatge fàcil i ràpid. A més a més, els beneficis (grau de satisfacció, resultat esperat...) de cada usuari ha de compensar al temps d'ús que en faci de l'eina. És molt important fer una estimació clara dels límits i les funcionalitats de l'aplicació per tal d'evitar durant el desenvolupament modificar tasques ja realitzades.

Com es tracta d'un projecte de final de grau, en aquest cas no hi ha hagut *skateholders* pròpiament dits però m'he basat en el coneixement que en tinc de les associacions d'animals i el contacte i conversacions amb els voluntaris i treballadors d'aquestes.

#### Funcionalitats principals

Al tractar-se d'una aplicació web enfocada a les associacions d'animals i a les persones que tenen intenció d'adoptar un gos, les funcionalitats principals serà fer cerques, poder identificar-se i que una associació pugui afegir els gossos que té en adopció. Per aquest motiu s'han agrupat els requisits en dos grans grups: funcionals i no funcionals, on aquest últims recullen els requisits de disseny, seguretat i els que quedarien fóra de rang

## 3.1 Requisits funcionals

En aquest punt es descriuen els requisits que tenen a veure amb les funcionalitats que ha de garantir el sistema i que per tant són imprescindibles per al bon funcionament d'aquesta.

Funcionals	Descripció
<b>FUN001</b>	Les associacions i protectores han de poder registrar-se.
<b>FUN002</b>	Les associacions i protectores registrades han de poder identificar-se i modificar les seves dades.
<b>FUN003</b>	Les associacions i protectores registrades han de poder afegir els gossos que tenen en adopció i modificar les seves dades.
<b>FUN004</b>	Els usuaris podran fer cerques pel nom de la comarca, el de la província, la ciutat, el nom de l'associació o bé pel tipus de raça del gos.
<b>FUN005</b>	Els usuaris podran accedir a la informació d'un gos determinat i veure la seva fotografia.

## 3.2 Requisits no funcionals

Els requisits no funcionals són aquells que el sistema haurà de complir per garantir els aspectes que no tenen a veure directament amb la funcionalitat, però que són necessaris per millorar-ne l'eficiència, la usabilitat, la seguretat,...

Disseny	Descripció
<b>DIS001</b>	L'aplicació ha de ser senzilla, intuïtiva i fàcil d'utilitzar.
<b>DIS002</b>	L'aplicació ha de poder contenir les imatges dels gossos en adopció.

Seguretat	Descripció
<b>SEG001</b>	Una associació o protectora sense identificar no podrà accedir a les funcionalitats de l'aplicació.
<b>SEG002</b>	Una associació o protectora identificades podran sortir de l'aplicació.
<b>SEG003</b>	Les contrasenyes dels usuaris hauran de ser xifrades abans de guardar-les a la base de dades.

Els següents requisits per la limitació del temps estarien fóra de l'abast d'aquest projecte:

Fóra de Rang	Descripció
<b>FR001</b>	Les associacions i protectores registrades han de poder afegir els gats o altres animals que tinguin en adopció i modificar les seves dades.
<b>FR002</b>	La divisió de tasques dels usuaris per rols.
<b>FR003</b>	Canvi d'idioma.

## 4. Especificació

En aquest apartat es defineix el comportament del sistema a implementar. Tenint en compte els requisits recollits s'ha determinat quins actors del sistema i quins casos d'ús es poden realitzar.

### 4.1 Actors del sistema

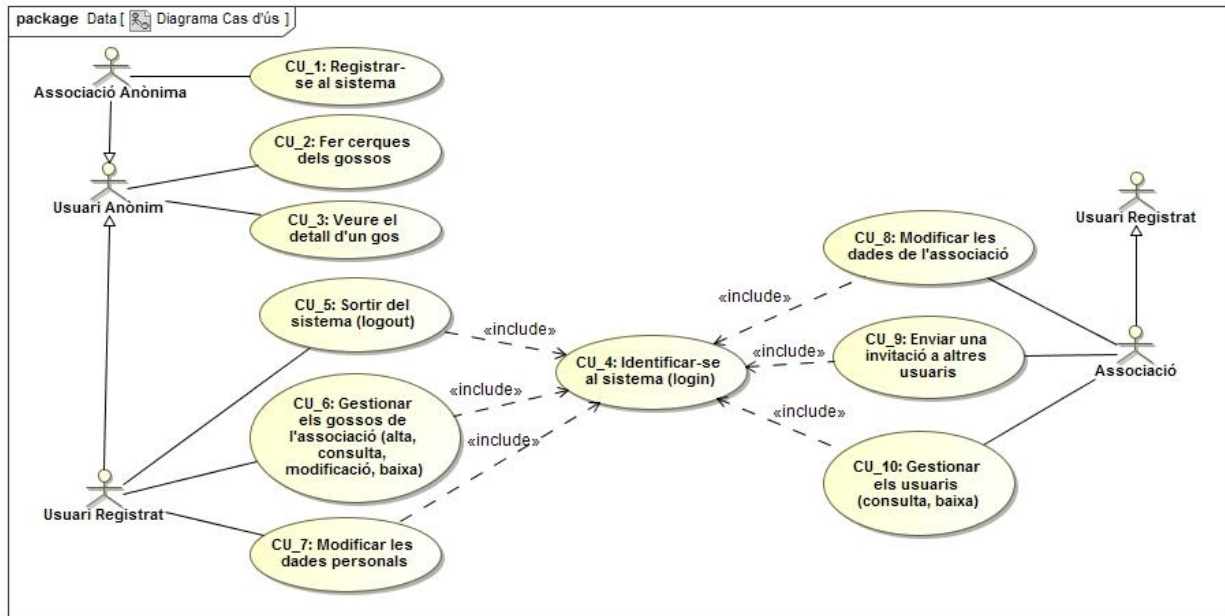
A l'aplicació hi haurà quatre tipus de rols: l'associació anònima, l'usuari anònim, l'usuari registrat i l'associació. S'ha dividit així per diferenciar les associacions dels usuaris registrats encara que en realitat seria una qüestió de rols i les associacions serien els usuaris registrats amb més permisos.

Cada usuari que entri al sistema serà un usuari anònim que tindrà permisos reduïts; els usuaris podran registrar la seva associació que fins que no es registri serà anònima i cada associació només es registrarà un cop. Quan l'associació s'hagi registrat tindrà permisos per poder invitar a altres usuaris; finalment l'usuari registrat és aquell que haurà rebut la invitació de participació de la seva associació i que per tant tindrà permisos per poder gestionar els gossos de l'associació, fer-ne consultes i invitar a altres usuaris a participar entre d'altres.

Actualment, el rols associació i usuari registrat són els que tindran tots els permisos de l'aplicació encara que es preveu fer-ne una separació per tipus de rol.

## 4.2 Diagrama de casos d'ús

Amb aquesta informació i els requisits obtinguts es van representar les funcionalitats que hauria de tenir l'aplicació en el següent diagrama de casos d'ús:



II-lustració 1. Diagrama de casos d'us

## 4.3 Especificació dels casos d'ús

Seguidament es mostra en detall cada cas d'ús que apareix al diagrama, indicant els rols que tenen aquesta funcionalitat, les precondicions que s'han de complir, les postcondicions que es donaran un cop finalitzi el cas d'ús i una breu descripció de cadascun.

Cas d'ús	CU_1: Registrar-se al sistema
<b>Rol</b>	Associació Anònima
<b>Precondició</b>	Cap
<b>Postcondició</b>	Hi haurà una nova associació registrada a la base de dades i aquesta tindrà accés a l'aplicació.
<b>Descripció</b>	Una associació anònima es registra al sistema.

Cas d'ús	CU_2: Fer cerques dels gossos
<b>Rol</b>	Associació Anònima, Usuari Anònim, Usuari Registrat i Associació
<b>Precondició</b>	Seleccionar els paràmetres de la cerca i polsar el botó 'Cercar'
<b>Postcondició</b>	Es mostrarà un llistat amb els gossos que compleixin els paràmetres de la cerca.
<b>Descripció</b>	Es poden fer cerques per diferents paràmetres com àrea geogràfica, raça del gos, mida...

Cas d'ús	CU_3: Veure el detall d'un gos
<b>Rol</b>	Associació Anònima i Usuari Anònim
<b>Precondició</b>	Seleccionar un gos del llistat.
<b>Postcondició</b>	Es mostrarà la informació del gos detalladament.
<b>Descripció</b>	Es pot veure el detall d'un gos.



<b>Cas d'ús</b>	<b>CU_4: Identificar-se al sistema (login)</b>
<b>Rol</b>	Usuari Registrat i Associació
<b>Precondició</b>	Estar registrat al sistema o haver rebut una invitació d'alguna Associació.
<b>Postcondició</b>	S'accedeix al sistema amb els permisos del rol de l'usuari.
<b>Descripció</b>	Autenticació de l'usuari per accedir al sistema.

<b>Cas d'ús</b>	<b>CU_5: Sortir del sistema (logout)</b>
<b>Rol</b>	Usuari Registrat i Associació
<b>Precondició</b>	Estar identificat al sistema
<b>Postcondició</b>	L'usuari o associació deixen de tenir permisos d'usuari registrat a l'aplicació.
<b>Descripció</b>	Es pot deixar d'estar identificat a l'aplicació web.

<b>Cas d'ús</b>	<b>CU_6: Gestionar els gossos de l'associació (alta, consulta, modificació, baixa)</b>
<b>Rol</b>	Usuari Registrat i Associació
<b>Precondició</b>	Estar identificat al sistema
<b>Postcondició</b>	<p>Si és una alta s'afegirà un nou gos a la taula de gossos de la base de dades. Si és una consulta s'accedirà a la base de dades per mostrar els gossos de l'associació.</p> <p>Si és una modificació és modificarà alguna dada d'algun gos de la base de dades.</p> <p>Si és una baixa s'eliminarà un gos de la taula de gossos de la base de dades.</p>
<b>Descripció</b>	Es podran fer consultes, modificacions, altes i baixes de gossos de l'associació.

<b>Cas d'ús</b>	<b>CU_7: Modificar les dades personals</b>
<b>Rol</b>	Usuari Registrat i Associació
<b>Precondició</b>	Estar identificat al sistema
<b>Postcondició</b>	Les dades de l'usuari es modificaran a la base de dades.
<b>Descripció</b>	Cada usuari podrà fer modificacions de les seves dades personals.

<b>Cas d'ús</b>	<b>CU_8: Modificar les dades de l'associació</b>
<b>Rol</b>	Associació
<b>Precondició</b>	Estar identificat al sistema amb rol d'Associació
<b>Postcondició</b>	Les dades de de l'associació es modificaran a la base de dades.
<b>Descripció</b>	Cada Associació podrà fer modificacions de les seves dades.

<b>Cas d'ús</b>	<b>CU_9: Enviar una invitació a altres usuaris</b>
<b>Rol</b>	Associació
<b>Precondició</b>	Estar identificat al sistema amb rol d'Associació
<b>Postcondició</b>	S'enviarà un correu electrònic amb les dades d'accés al e-mail indicat.
<b>Descripció</b>	Cada Associació podrà invitar a usuaris per gestionar els gossos de l'Associació i aquests rebran un nombre d'usuari i una contrasenya per accedir com Usuaris Registrats.

Cas d'ús	CU_10: Gestionar els usuaris (consulta, baixa)
<b>Rol</b>	Associació
<b>Precondició</b>	Estar identificat al sistema amb rol d'Associació
<b>Postcondició</b>	Si és una consulta s'accedirà a la base de dades per llistar els usuaris de l'associació.  Si és una baixa s'eliminarà un usuari de la taula Usuaris de la base de dades.
<b>Descripció</b>	Cada Associació podrà consultar els usuaris que té i podrà donar-los de baixa.

## 4.4 Model conceptual

Un cop analitzats els casos d'ús s'obté la informació necessària per tal de determinar quines classes seran requerides per a la implementació del sistema. Per tant, es podrà començar a dissenyar l'estructura que tindrà la base de dades.

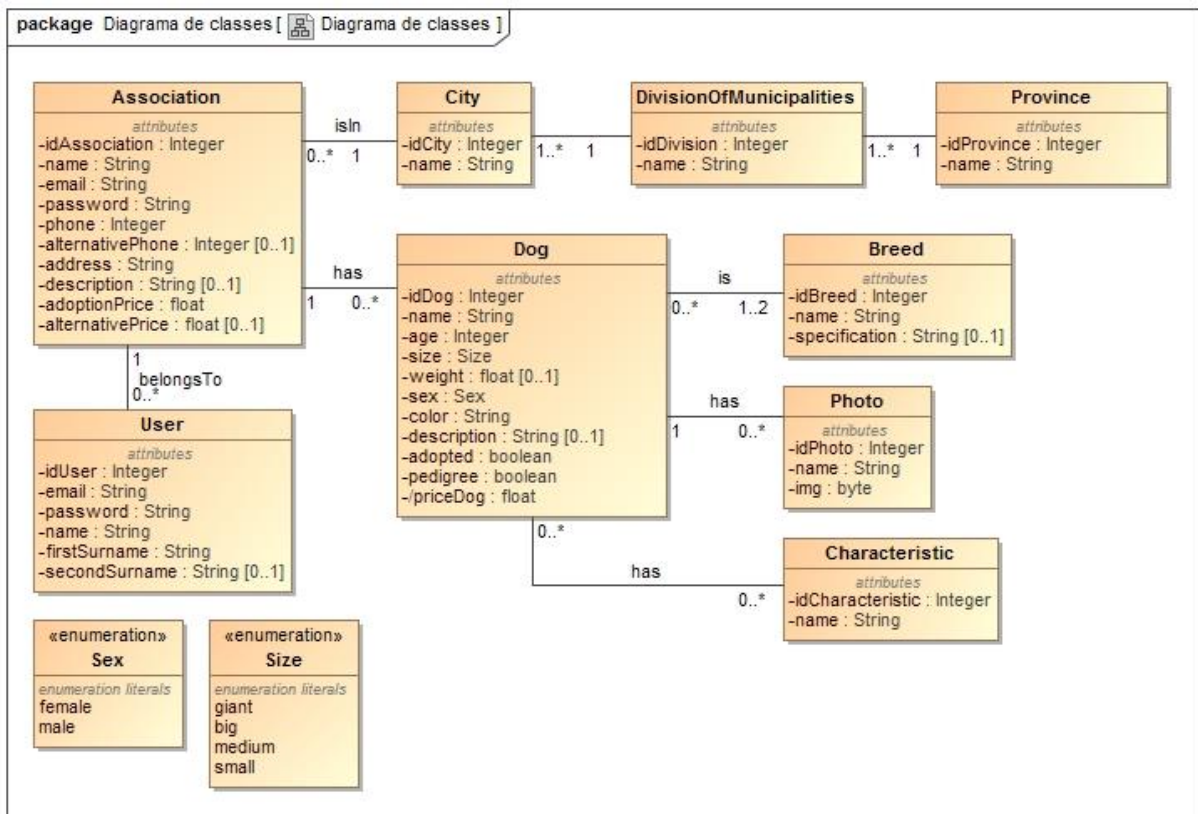
Per una banda és imprescindible que la base de dades guardi la informació dels usuaris registrats i de l'Associació a la qual pertanyen, per tal que només puguin editar les dades relacionades amb la seva Associació. També s'ha de conèixer la informació principal de l'entitat i de l'usuari per distingir-se de la resta.

La taula que emmagatzemarà les dades dels gossos serà una de les que guardarà més quantitat de dades ja que és l'entitat principal, i per tant s'hauran de guardar dades detallades així com el nom, edat, raça, sexe... I per certes dades es necessitaran classes enumeratives que guardin valors constants.

Una propietat important dels gossos és que cadascun pugui contenir com a mínim una imatge i trets característics que l'identifiquin encara que això seria opcional.

Per altra banda s'haurà de guardar en alguna taula la informació sobre on estan ubicades les associacions i per això seria bona idea relacionar-les amb la ciutat o municipi on es trobi.

Llavors el diagrama inicial de classes que es va fer i que respon a aquestes necessitats seria el següent:



II-lustració 2. Diagrama de classes

## Detall de les classes

A continuació es detallaran les classes del diagrama per explicar les relacions i la informació que guardarà cadascuna d'aquestes a la base de dades:

Classe	Descripció
<b>Association</b>	Guardarà dades de l'Associació com el nom, l'adreça, preu de les adopcions... Els únics camps opcionals a l'hora de registrar-se seran el telèfon i la descripció; i com és lògic al principi no tindran gossos ni usuaris associats. Tant el preu d'adopció com la resta de camps són camps obligatoris.
<b>User</b>	Guardarà les dades de l'usuari com el nom i el correu electrònic, encara que no serà necessari que donin moltes dades personals. Els usuaris estaran enllaçats amb l'Associació a la qual pertanyin des del principi.
<b>Dog</b>	Guardarà la informació del gos, el seu nom, pes, mida... i està enllaçat amb l'Associació i les classes que el defineixen com Breed (raça), Photo (imatge) i Characteristic (característiques). Té l'atribut <i>price</i> que és derivat de la classe Association. Tots els camps són obligatoris excepte el pes, la descripció, les imatges i les característiques.
<b>City</b>	En principi només guardarà totes les ciutats de Catalunya perquè els usuaris del sistema en seleccionin una.
<b>DivisionOfMunicipalities</b>	En principi només guardarà totes les comarques de Catalunya perquè els usuaris del sistema en seleccionin una.

<b>Province</b>	En principi només guardarà totes les províncies de Catalunya perquè els usuaris del sistema en seleccionin una.
<b>Breed</b>	Guardarà un registre de les races de gos més comunes.
<b>Photo</b>	Guardarà les fotografies dels gossos i estaran associades a aquests.
<b>Characteristic</b>	Guardarà un llistat de les característiques més comunes que pot tenir un gos com jugarer, tranquil,... perquè els usuaris en puguin seleccionar varies.
<b>Sex</b>	Determinarà si un gos és femella o mascle.
<b>Size</b>	Determinarà la mida del gos, si és petit, mitjà, gran o gegant.

## 5 Arquitectura

### 5.1 Patró Model-Vista-Controlador

Es va decidir que el sistema tingués una estructura en tres capes seguint el patró arquitectònic Model-Vista-Controlador (MVC), per tal de separar la capa que gestionaria la persistència de les dades, la dels processos de negoci i l'encarregada de la presentació de les dades als usuaris.

#### Tecnologies de les capes

La capa de presentació s'ha implementat amb l'ajuda del framework Java Server Faces (JSF) que alhora necessita tres elements: la classe Faces Servlet que ja ve implementada amb la tecnologia i s'encarrega d'examinar les peticions rebudes de la interfície i d'actualitzar-la amb les dades que rep dels Managed Beans; els Managed Bean que és on s'implementa la part lògica de l'aplicació i per tant en són els responsables; i els Facelets que són els fitxers amb extensió xhtml, és a dir, les pàgines de la interfície d'usuari de l'aplicació, i vinculen els components JSF amb els Managed Bean.

La capa de negoci s'ha implementat amb un EJB Session sense estat i amb una interfície remota. I per poder interactuar amb les entitats JPA he configurat i utilitzat el framework d'Hibernate. Per últim, la capa d'integració conté totes les entitats necessàries per al funcionament de l'aplicació i s'han implementat com a classes JPA.

## 5.2 Disseny tècnic

Els dissenys que es van fer inicialment, després de l'obtenció dels requisits i de l'especificació de les funcionalitats, representen aquest patró. A continuació es mostra el diagrama de components que es ceneix a aquesta arquitectura i on es detallen els mètodes que garantiran la funcionalitat per als casos d'ús descrits.

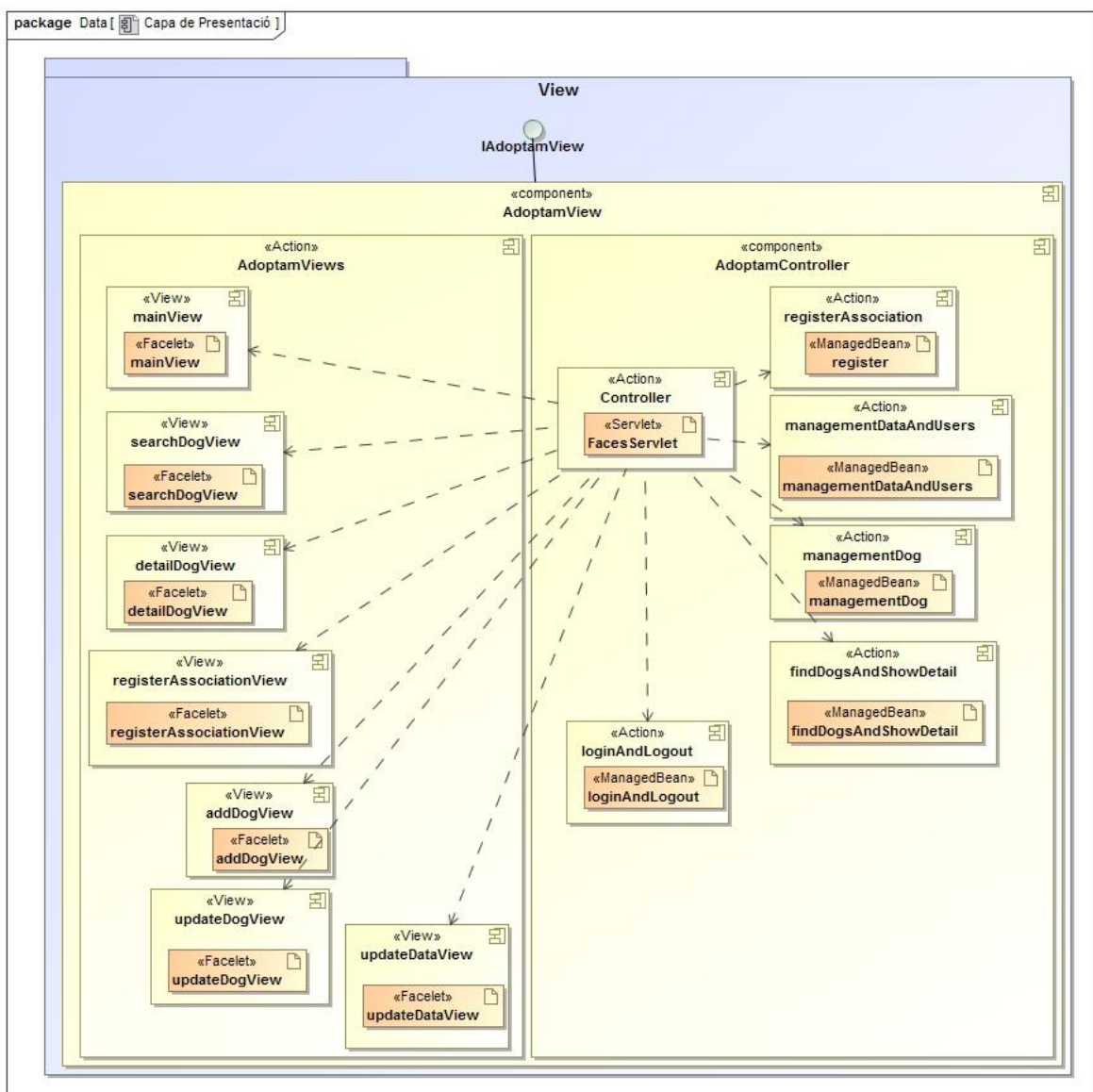


II-lustració 3. Diagrama de components



## Capa de presentació

Fent un refinament del diagrama de components de la capa de presentació (View) aplicant el perfil JavaEE i tenint en compte que s'utilitzarà el *framework* Java Server Faces (JSF), es dedueix que no caldrà implementar el controlador ja que s'utilitzarà el d'aquesta mateixa tecnologia. Per altra banda, les accions estaran implementades com a *Managed Beans*. El diagrama que es va fer a l'inici va ser el següent:



II-Il·lustració 4. Diagrama Capa de Presentació

## 5.3 Tecnologies

Al finalitzar els dissenys es va fer una cerca de les eines més apropiades per realitzar el projecte. A continuació es detallaran les principals eines i tecnologies utilitzades per a la implementació del projecte:

- PostgreSQL 9.4: es tracta d'un sistema de gestió de bases de dades relacional de codi obert que utilitza un model client/servidor i multiprocessos per garantir l'estabilitat del sistema.
- Hibernate 4.2.15: és una eina de mapeig objecte-relacional (Object Relational Mapping) que facilita als desenvolupadors la gestió de la persistència de les dades en una base de dades relacional i el model d'objectes de l'aplicació.
- Hibernate JPA 1.0: es tracta d'una implementació específica d'Hibernate de la Java Persistence API (JPA).
- Tomcat 7.0: és un contenidor d'aplicacions que implementa el Servlet 3.0 i especificacions JavaServer Pages 2.2, i és útil per desplegar les aplicacions que s'estan desenvolupant.
- Primefaces 5.3: és un framework de components per a JavaEE que faciliten la creació de les interfícies d'usuari de les aplicacions web.
- JavaMail 1.4: és la API que proporciona un framework independent de la plataforma per a la construcció i enviaments de correus electrònics.
- Eclipse Kepler: és l'entorn de desenvolupament integrat (IDE) utilitzat per al desenvolupament de l'aplicació.

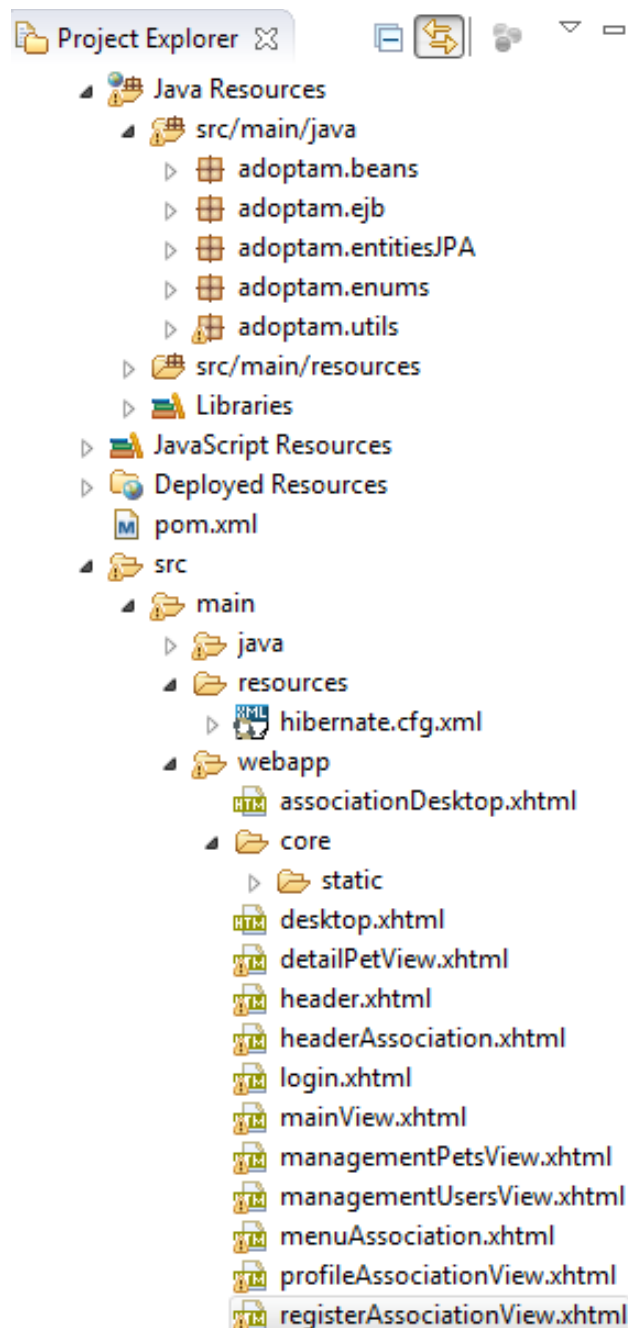
- Apache Maven: és una eina de gestió de projectes de software basada en el concepte d'un model d'objecte de projecte, el pom. Ajuda a generar els arxius necessaris per fer el desplegament.
- MagicDraw: és una eina de modelatge de dissenys de sistemes orientats a objectes i bases de dades.

## 6 Implementació

### 6.1 Estructura del projecte

El projecte Maven agrupa les classes java per una banda i les vistes per l'altra.

L'estructura a l'hora d'entregar el projecte és la següent:



Il·lustració 5. Estructura del projecte

Dins de la carpeta java he dividit per *packages* les diferents parts, en ordre alfabètic anirien primer els Managed Beans, seguit pels components EJB, després les entitats JPA, els enumeratius que siguin necessaris i finalment vaig afegir un altre paquet per a les classes que em serien d'utilitat i es troben fóra de les classes principals del projecte com els converters per a la utilització d'algun component de primefaces o mètodes per a la gestió de les sessions d'Hibernate.

La part de les vistes que també inclou el fitxer css amb els estils, es troba dins la carpeta webapps i aquí es on es configura el fitxer web.xml per al funcionament del servlet Faces Servlet i on vaig haver de configurar un component de primefaces per poder realitzar la pujada d'imatges.

## 6.2 Canvis en el disseny

Degut a la poca experiència que tinc en projectes d'aquest tipus l'estructura final del projecte no ha resultat exactament com havia previst als dissenys inicials, encara que els canvis han sigut per motius de millores o per evitar repeticions de codi.

### Funcionalitat no implementada

Hi ha una funcionalitat que al principi m'havia proposat que era la d'afegir més d'una fotografia per a cada gos, però després em vaig trobar amb la problemàtica de com mostrar més d'una fotografia en la vista utilitzant primefaces.

Encara que existeixen components com el de *galleria*, al treballar amb un altre tipus d'objecte no vaig poder utilitzar-ho i, en comptes d'això, vaig deixar l'entorn preparat perquè es pogués afegir només una fotografia a cada gos.

Però a pesar d'aquest inconvenient, vaig mantenir la relació múltiple entre l'entitat gos i l'entitat foto ja que sí que es poden afegir múltiples fotografies

però pel temps que he tingut per realitzar les implementacions no he pogut preparar les vistes per mostrar múltiples imatges provinents d'una llista. I ho he deixat de cara a futures millores.

## Canvis en la base de dades

Pel que fa a la base de dades els canvis han estat relacionats amb alguns atributs que al final no he afegit com és el cas de que una associació tingui correu electrònic i contrasenya ja que després vaig determinar que els usuaris serien els que s'identifiquessin sense tenir en compte qui havia registrat l'associació.

També fa ser el cas de la relació entre l'entitat gos i l'entitat raça que en un principi vaig pensar que els gossos al no tenir pedigree podria indicar-se més d'una raça però després vaig veure que no tenia massa sentit ja que encara que un gos provingui de més d'una raça de vegades inclús és difícil determinar-ne una d'aquestes i això seria una complicació per als usuaris que ho haguessin d'utilitzar.

## Canvis de la capa de presentació

Pel que fa al diagrama de la capa de presentació que es va fer, l'estructura no es correspon al 100% amb el que s'havia dit ja que en el moment de fer la implementació és quan em vaig adonar de les vistes i managed beans que necessitaria i de quins es podrien agrupar per no repetir codi.

Em va passar similar amb els mockups que encara que tenia uns esbossos de com serien les pantalles, al donar-li unes voltes i pel poc coneixement que tenia de primefaces en un principi els veia d'una manera però després vas adaptant la teva idea al que en realitat es pot fer. I així, realment, el resultat sol ser molt millor.

## 6.3 Futures millores

Com a futures millores tinc moltes idees i crec que l'aplicació és pot millorar en diversos sentits, és a dir, tan estructuralment com funcionalment.

### Estructurals

Estructuralment es podria millorar la distribució de les vistes i afegir-hi el parent-flow. També crec que seria una bona millora si l'aplicació es pogués adaptar al framework de Spring ja que té funcionalitats de seguretat i configuracions dels flows que poden ser d'utilitat.

Per altra banda faltaria afegir els diccionaris i adaptar l'aplicació perquè acceptés un parell d'idiomes com a mínim.

### Funcionals

En temes de funcionalitat crec que es podrien afegir de noves com per exemple:

- Permetre a les associacions que puguin afegir gats o altres animals també i per tant s'haurien de fer les modificacions pertinents a la base de dades i al projecte.
- Preparar l'aplicació perquè els animals poguessin tenir més d'una fotografia, en aquest cas caldria modificar mètodes i preparar les vistes.
- Crear rols d'usuari per als usuaris de l'associació per tal de distribuir-ne les funcionalitats i que no tothom pugui eliminar gossos i usuaris o controlar qui envia les invitacions d'usuari.

## 7 Aplicació

### 7.1 Funcionalitats de l'aplicació

En aquest apartat s'expliquen més detalladament les funcionalitats que s'han implementat durant el projecte. Per a la seva explicació, s'han distribuït les funcionalitats tenint en compte la pantalla a la qual pertanyen.

#### Pantalla principal

- Fer una cerca: qualsevol usuari pot realitzar una cerca indicant-ne algun d'aquests paràmetres: la comarca, la província, la ciutat o el nom de l'associació o bé afitant la cerca seleccionant-ne més d'un. El buscador tindrà en compte els paràmetres que acotin més la cerca i mostrarà el llistat de gossos que compleixin aquests requisits.
- Obtenir més informació d'un gos: es pot veure el detall complet de la informació de cada gos de la cerca.

#### Registre d'una associació

- Donar d'alta una associació: un usuari pot donar d'alta la seva associació indicant-ne el nom, adreça, telèfon, preu de les adopcions,... i també haurà d'indicar informació personal per poder accedir-hi.

#### Pantalla d'accés

- Identificar-se: un usuari que hagi registrat la seva associació o que hagi rebut un missatge d'invitació d'un company ja registrat podrà accedir indicant el seu correu electrònic i una contrasenya.



## Perfil de l'associació

- Modificar les dades de l'associació: un usuari identificat podrà modificar les dades de la seva associació així com la direcció, preu de les adopcions,...
- Modificar les dades de l'usuari: un usuari identificat podrà modificar les seves dades exceptuant el correu electrònic que haurà de mantenir per accedir a l'aplicació.
- Modificar la contrasenya: els usuaris identificats podran modificar la seva contrasenya.

## Dades dels gossos

- Consultar el llistat de gossos: un usuari identificat podrà veure el llistat de gossos afegits de la seva associació.
- Modificació de les dades d'un gos: un usuari identificat podrà modificar les dades d'un gos de la seva associació per millorar-les, actualitzar-les,... així com afegir-li una foto, actualitzar-la o eliminar-la si prèviament ja en tenia una.
- Eliminació d'un gos: un usuari registrat podrà donar de baixa un gos del llistat de la seva associació.
- Afegir un gos: un usuari registrat podrà afegir un gos a la seva associació tot indicant-li un nom, sexe, raça, descripció,...

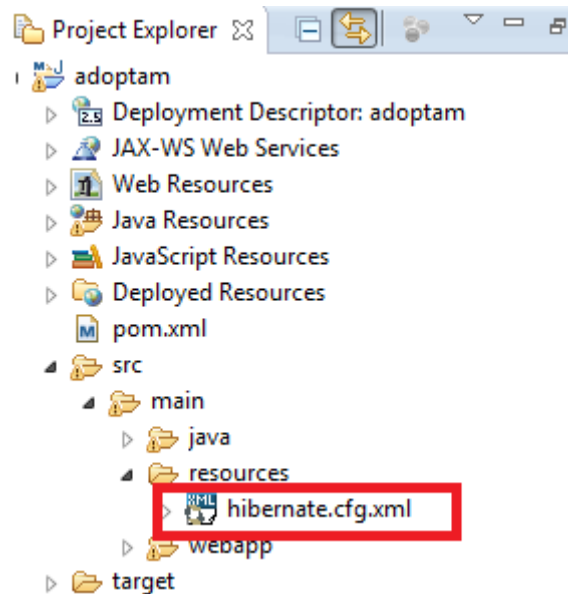
## Llistat d'usuaris

- Consultar el llistat d'usuaris: un usuari registrat podrà veure el llistat d'usuaris que formen part de la seva associació.
- Modificar el rol d'un usuari: un usuari registrat podrà canviar el rol d'altres usuaris de la seva associació perquè siguin administradors.
- Eliminar un usuari: un usuari registrat podrà donar de baixa a un usuari de la seva associació.
- Invitar a un usuari: un usuari identificat podrà enviar invitacions a altres usuaris perquè col·laborin amb la seva associació, tot indicant-ne un nom i cognom i una adreça de correu electrònic. L'usuari invitat rebrà un correu electrònic amb una contrasenya que s'haurà generat de forma automàtica perquè pugui accedir.

## 7.2 Desplegament

### Preparació de l'entorn

Primerament s'haurà de crear un rol anomenat 'adoptam' amb contrasenya 'adoptam' a la base de dades o fer els canvi de nom d'usuari i contrasenya a l'arxiu de configuració d'hibernate ('hibernate.cfg.xml') que es troba dins la carpeta *resources*:



Il·lustració 6. Ubicació arxiu d'hibernate

```
1 <?xml version='1.0' encoding='UTF-8'?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4     "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
5
6 <hibernate-configuration>
7     <session-factory>
8
9         <!-- Connection settings -->
10        <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
11        <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/postgres</property>
12        <property name="hibernate.connection.username">adoptam</property>
13        <property name="hibernate.connection.password">adoptam</property>
14
15        <!-- SQL dialect -->
16        <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
17
```

Il·lustració 7. Configuració hibernate usuari

Després es podrà crear l'estructura de la base de dades i per fer-ho hi ha dues opcions: o bé de forma manual amb l'script que adjunto amb la memòria, o bé es podrà fer de forma automàtica modificant el valor del paràmetre `hibernate.hbm2ddl.auto` de l'arxiu de configuració d'hibernate ('hibernate.cfg.xml') de 'validate' a 'create'.

```
6<hibernate-configuration>
7  <session-factory>
8
9    <!-- Connection settings -->
10   <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
11   <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/postgres</property>
12   <property name="hibernate.connection.username">adoptam</property>
13   <property name="hibernate.connection.password">adoptam</property>
14
15   <!-- SQL dialect -->
16   <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
17
18   <!-- Print executed SQL to stdout -->
19   <property name="show_sql">>true</property>
20
21   <!-- Drop and create database on startup -->
22   <property name="hibernate.hbm2ddl.auto">validate</property>
23
24   <property name="hibernate.jdbc.use_streams_for_binary">>true</property>
25   <property name="current_session_context_class">thread</property>
26
27   <!-- Annotated entity classes -->
28   <mapping class="adoptam.entitiesJPA.AssociationJPA" />
29   <mapping class="adoptam.entitiesJPA.BreedJPA" />
30   <mapping class="adoptam.entitiesJPA.CharacteristicJPA" />
31   <mapping class="adoptam.entitiesJPA.CityJPA" />
32   <mapping class="adoptam.entitiesJPA.DivisionOfMunicipalitiesJPA" />
33   <mapping class="adoptam.entitiesJPA.ProvinceJPA" />
```

## II-Il·lustració 8. Hibernate creació taules

En aquest últim cas s'haurà de crear l'esquema amb nom 'adoptam' primerament i de forma manual. I no s'ha d'oblidar tornar a modificar el valor de 'create' a 'validate' un cop creada.

Script de la creació de les taules:

```
CREATE SCHEMA adoptam;
ALTER SCHEMA adoptam OWNER TO adoptam;

--Table for the provinces
CREATE TABLE adoptam.province (
  id integer NOT NULL,
  name_province character varying(255) NOT NULL,
  CONSTRAINT province_pkey PRIMARY KEY (name_province),
  CONSTRAINT province_unique UNIQUE (id)
);
```

```
ALTER TABLE adoptam.province OWNER TO adoptam;
```

```
CREATE SEQUENCE adoptam.province_seq
```

```
  START WITH 1
```

```
  INCREMENT BY 1
```

```
  NO MINVALUE
```

```
  NO MAXVALUE
```

```
  CACHE 1;
```

```
ALTER TABLE adoptam.province_seq OWNER TO adoptam;
```

```
-- Table for divisions of municipalities ("comarques")
```

```
CREATE TABLE adoptam.division_of_municipalities (
```

```
  id integer NOT NULL,
```

```
  name_division character varying(255) NOT NULL,
```

```
  name_province character varying(255) NOT NULL,
```

```
  CONSTRAINT fkf814f32e25bb3cf5 FOREIGN KEY (name_province)
```

```
    REFERENCES adoptam.province (name_province) MATCH SIMPLE
```

```
    ON UPDATE NO ACTION ON DELETE NO ACTION,
```

```
  CONSTRAINT division_pkey PRIMARY KEY (id),
```

```
  CONSTRAINT division_unique UNIQUE (id)
```

```
);
```

```
ALTER TABLE adoptam.division_of_municipalities OWNER TO adoptam;
```

```
CREATE SEQUENCE adoptam.division_seq
```

```
  START WITH 1
```

```
  INCREMENT BY 1
```

```
  NO MINVALUE
```

```
  NO MAXVALUE
```

```
  CACHE 1;
```

```
ALTER TABLE adoptam.division_seq OWNER TO adoptam;
```

```
-- Table for cities
```

```
CREATE TABLE adoptam.city (
```

```
  id integer NOT NULL,
```

```
  name_city character varying(255) NOT NULL,
```

```
  id_division integer NOT NULL,
```

```
CONSTRAINT city_pkey PRIMARY KEY (name_city),
CONSTRAINT fkf814f32e25bb3cf5 FOREIGN KEY (id_division)
REFERENCES adoptam.division_of_municipalities (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT city_unique UNIQUE (id)
);
```

```
ALTER TABLE adoptam.city OWNER TO adoptam;
```

```
CREATE SEQUENCE adoptam.city_seq
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE adoptam.city_seq OWNER TO adoptam;
```

```
-- Table for associations
```

```
CREATE TABLE adoptam.association (
id integer NOT NULL,
date_join_association timestamp without time zone DEFAULT now(),
name_association character varying(255) NOT NULL,
address character varying(255) NOT NULL,
description character varying(255),
phone integer,
adoption_price character varying(255) NOT NULL,
name_city character varying(255) NOT NULL,
CONSTRAINT association_pkey PRIMARY KEY (id),
CONSTRAINT fkf814f32e25bb3cf5 FOREIGN KEY (name_city)
REFERENCES adoptam.city (name_city) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

```
ALTER TABLE adoptam.association OWNER TO adoptam;
```

```
CREATE SEQUENCE adoptam.association_seq
START WITH 1
INCREMENT BY 1
NO MINVALUE
```

```

NO MAXVALUE
CACHE 1;

ALTER TABLE adoptam.association_seq OWNER TO adoptam;

-- Table for users
CREATE TABLE adoptam.user (
  id integer NOT NULL,
  date_join_user timestamp without time zone DEFAULT now(),
  email character varying(255) NOT NULL,
  password character varying(255) NOT NULL,
  name_user character varying(255) NOT NULL,
  first_surname character varying(255) NOT NULL,
  second_surname character varying(255),
  is_admin boolean NOT NULL,
  id_association integer NOT NULL,
  CONSTRAINT user_pkey PRIMARY KEY (id),
  CONSTRAINT fkf814f32e25bb3cf5 FOREIGN KEY (id_association)
    REFERENCES adoptam.association (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT user_unique UNIQUE (email)
);

ALTER TABLE adoptam.user OWNER TO adoptam;

CREATE SEQUENCE adoptam.user_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE adoptam.user_seq OWNER TO adoptam;

-- Table for breed
CREATE TABLE adoptam.breed (
  id integer NOT NULL,
  name_breed character varying(255) NOT NULL,
  CONSTRAINT breed_pkey PRIMARY KEY (id)
);

```

```
ALTER TABLE adoptam.breed OWNER TO adoptam;
```

```
CREATE SEQUENCE adoptam.breed_seq
```

```
  START WITH 1
```

```
  INCREMENT BY 1
```

```
  NO MINVALUE
```

```
  NO MAXVALUE
```

```
  CACHE 1;
```

```
ALTER TABLE adoptam.breed_seq OWNER TO adoptam;
```

```
-- Table for dogs
```

```
CREATE TABLE adoptam.dog (
```

```
  id integer NOT NULL,
```

```
  date_join_dog timestamp without time zone DEFAULT now(),
```

```
  name_dog character varying(255) NOT NULL,
```

```
  age character varying(255) NOT NULL,
```

```
  color character varying(255) NOT NULL,
```

```
  weight character varying(255),
```

```
  sex character varying(255) NOT NULL,
```

```
  size character varying(255) NOT NULL,
```

```
  description character varying(255),
```

```
  adopted boolean NOT NULL,
```

```
  pedigree boolean NOT NULL,
```

```
  id_association integer NOT NULL,
```

```
  added_by integer NOT NULL,
```

```
  id_breed integer NOT NULL,
```

```
  CONSTRAINT dog_pkey PRIMARY KEY (id),
```

```
  CONSTRAINT fkf814f32e25bb3cf5 FOREIGN KEY (id_association)
```

```
    REFERENCES adoptam.association (id) MATCH SIMPLE
```

```
    ON UPDATE NO ACTION ON DELETE NO ACTION,
```

```
  CONSTRAINT fkf814f32e25bb3cf0 FOREIGN KEY (added_by)
```

```
    REFERENCES adoptam.user (id) MATCH SIMPLE
```

```
    ON UPDATE NO ACTION ON DELETE NO ACTION,
```

```
  CONSTRAINT fkf814f32e25bc3cf0 FOREIGN KEY (id_breed)
```

```
    REFERENCES adoptam.breed (id) MATCH SIMPLE
```

```
    ON UPDATE NO ACTION ON DELETE NO ACTION
```

```
);
```



```
ALTER TABLE adoptam.dog OWNER TO adoptam;

CREATE SEQUENCE adoptam.dog_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE adoptam.dog_seq OWNER TO adoptam;

-- Table for photos
CREATE TABLE adoptam.photo (
  id integer NOT NULL,
  name_photo character varying(255) NOT NULL,
  image bytea NOT NULL,
  dog_id integer NOT NULL,
  CONSTRAINT photo_pkey PRIMARY KEY (id),
  CONSTRAINT fkf814f32e25bb3cf5 FOREIGN KEY (dog_id)
  REFERENCES adoptam.dog(id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
);

ALTER TABLE adoptam.photo OWNER TO adoptam;

CREATE SEQUENCE adoptam.photo_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE adoptam.photo_seq OWNER TO adoptam;

-- Table for characteristics
CREATE TABLE adoptam.characteristic (
  id integer NOT NULL,
  name_characteristic character varying(255) NOT NULL,
  CONSTRAINT characteristic_pkey PRIMARY KEY (id)
);
```

```

ALTER TABLE adoptam.characteristic OWNER TO adoptam;

CREATE SEQUENCE adoptam.characteristic_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE adoptam.characteristic_seq OWNER TO adoptam;

--Table with dog id and characteristic id
CREATE TABLE adoptam.characteristic_dog (
  dog_id integer NOT NULL,
  characteristic_id integer NOT NULL,
  CONSTRAINT characteristic_dog_pkey PRIMARY KEY (dog_id, characteristic_id),
  CONSTRAINT fkf814f32e25bb3cf5 FOREIGN KEY (dog_id)
  REFERENCES adoptam.dog(id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fkf814f32e25bb3ce5 FOREIGN KEY (characteristic_id)
  REFERENCES adoptam.characteristic(id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
);

ALTER TABLE adoptam.characteristic_dog OWNER TO adoptam;

```

Seguidament i abans de la posada en marxa de l'aplicació és necessari afegir l'script que té el nom de 'dades\_inicials.sql' a la base de dades relacional, ja que conté la inserció de les comarques, províncies i ciutats de Catalunya i també conté les dades que formen el llistat de races de gossos i algunes característiques que es podrien ampliar. Aquest per la seva mida no l'adjuntaré al document.

## Desplegament

Després de la preparació de l'entorn ja es podrà desplegar l'aplicació en un servidor, utilitzant l'arxiu amb extensió war obtingut fent un 'maven install' sobre el projecte. I ja es podrà accedir a l'aplicació amb el següent enllaç:

<http://localhost:8080/adoptam/>

## 8 Manual d'usuari

Aquest apartat pretén servir de guia per a l'ús de l'aplicació i per entendre millor les funcionalitats de cada pantalla. Primerament explicaré les funcionalitats de la capçalera que veuran els usuaris anònims.



### II-lustració 9. Capçalera usuaris anònims

Aquesta capçalera té 3 dreceres, una al títol 'Adoptam' que retornarà a l'usuari a la pantalla principal, un altre enllaç el que té per nom 'Registra't' que adreçarà als usuaris al formulari de registre d'una associació i per últim, la paraula 'Identifica't' que dirigeix a l'usuari a la pantalla d'identificació.


A continuació es detallaran les pantalles, començant per la pantalla principals i les pantalles que podran veure els usuaris anònims.

### 8.1 Pantalla principal

Aquesta pantalla conté el formulari des d'on es realitzaran les cerques. Es podrà indicar una província, una comarca, una ciutat, el nom de l'associació si es coneix o una raça o se'n poden indicar més d'un de cop per afitar la cerca.

### II-lustració 10. Formulari de cerca

Quan es comença a escriure per qualsevol dels paràmetres del qual es vulgui fer la cerca, n'apareixerà una llista amb els resultats que coincideixin amb les lletres indicades.



Provincia  
Indiqui una provincia

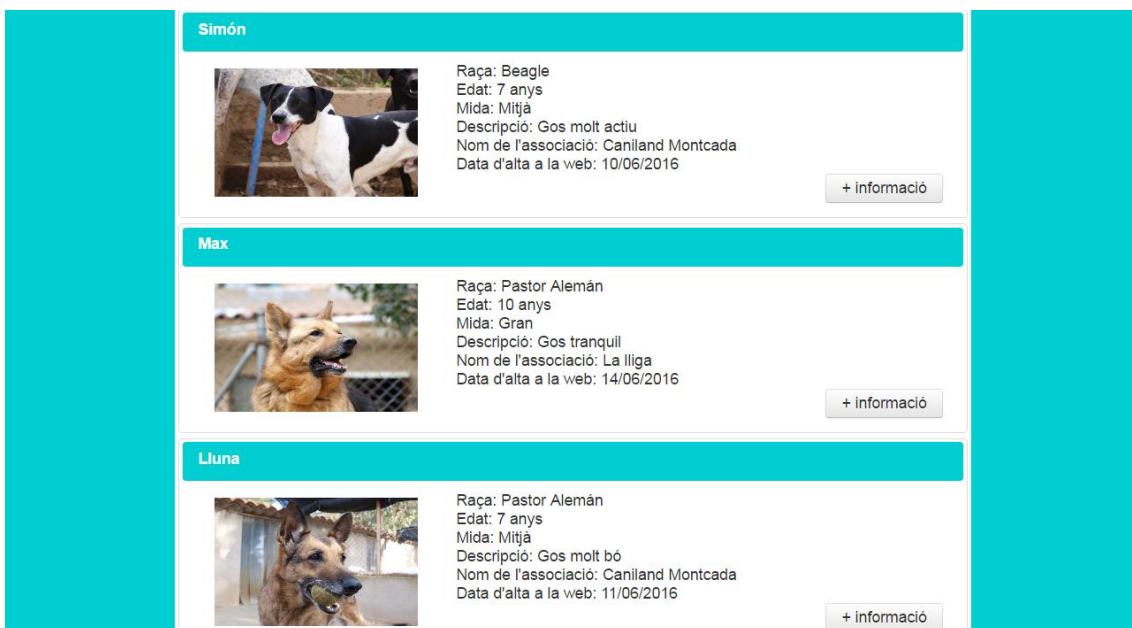
Comarca  
Indiqui una comarca


Ciutat  
Indiqui una ciutat

Associació  
a


- Refugi de Girona
- Mataró Associació
- Associació d'animals de Sant Feliu
- Caniland Montcada**
- La lliga

Al realitzar la cerca, sota el formulari n'apareixerà un llistat amb els gossos que compleixen els requisits, ja sigui els que pertanyin a una associació concreta o si es tracta d'una raça específica per exemple. D'aquests se'n veurà certa informació i una imatge rellevant com el seu nom, la raça i el nom de l'associació a la qual pertànyen.




Simón	
	Raça: Beagle Edat: 7 anys Mida: Mitjà Descripció: Gos molt actiu Nom de l'associació: Caniland Montcada Data d'alta a la web: 10/06/2016
<a href="#">+ informació</a>	

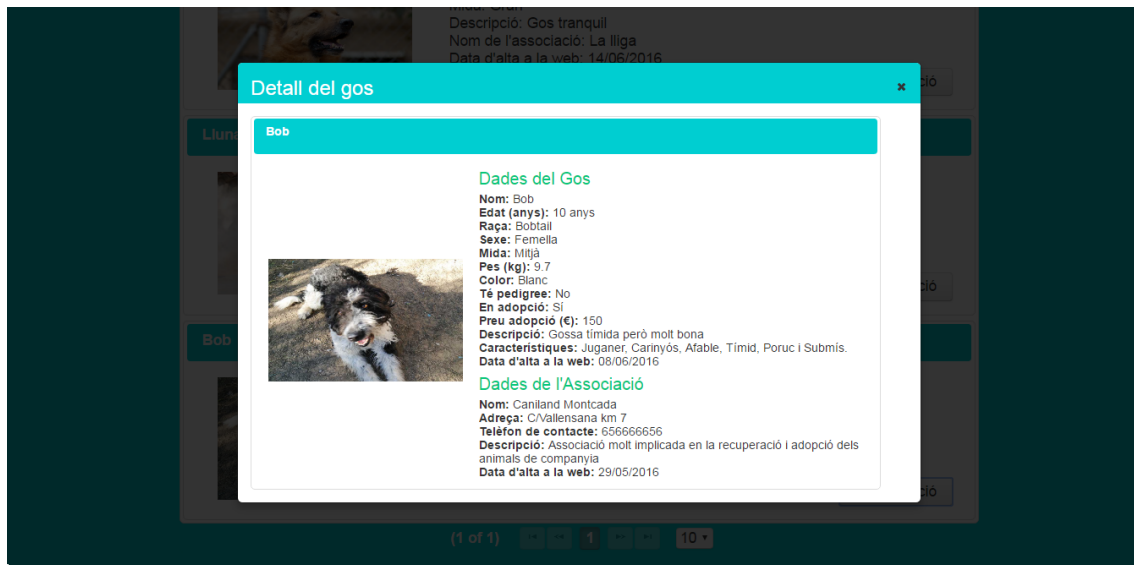
Max	
	Raça: Pastor Alemán Edat: 10 anys Mida: Gran Descripció: Gos tranquil Nom de l'associació: La lliga Data d'alta a la web: 14/06/2016
<a href="#">+ informació</a>	

Lluna	
	Raça: Pastor Alemán Edat: 7 anys Mida: Mitjà Descripció: Gos molt bó Nom de l'associació: Caniland Montcada Data d'alta a la web: 11/06/2016
<a href="#">+ informació</a>	

Il·lustració 11. Llistat de gossos

Per obtenir la informació completa d'un gos, es podrà clicar sobre el botó '+ informació' i s'obrirà una ventana emergent com aquesta:



II-lustració 12. Detall d'un gos

La ventana mostrarà tota la informació relacionada amb el gos així com l'edat, color, una descripció si la té, si encara està en adopció, el preu... I també mostrarà informació sobre l'associació a la qual pertany com la seva ubicació o un telèfon de contacte perquè els usuaris puguin contactar i concertar una cita per veure el gos en persona.

## 8.2 Pantalla de registre

Abans que els usuaris puguin fer cerques, les associacions s'hauran de registrar i aquesta tasca l'haurà de fer un usuari de l'associació ja que cada associació només podrà ser registrada un cop.

El formulari de registre té dues parts diferenciades, per una banda es troba la informació de l'associació on es primordial per al funcionament de les cerques que s'indiqui una ciutat, i per l'altra, la informació de l'usuari que la registra que es necessària per distingir els usuaris i per poder accedir-hi.

**Adoptam** [Registra't](#) [Identifica't](#)

**Registra la teva associació!**

**Dades de l'associació**

Nom de l'Associació:

Telèfon de contacte:

Adreça de l'Associació:

Descripció de l'Associació:

Preu de les adopcions:

Ciutat on es troba:

**Dades de l'usuari per accedir**

Nom de l'usuari:

Primer cognom:

Segon cognom:

Correu electrònic:

Contrasenya:

### Il·lustració 13. Pantalla de registre

Tant aquesta pantalla com la d'identificació tenen un botó al final del formulari per tornar a la pàgina principal.

En el cas que algun dels camps obligatoris no s'haguessin introduït, llavors apareixerien uns missatges d'error com els següents:

The image shows a registration form titled "Registra la teva associació!". The form is divided into two main sections: "Dades de l'associació" and "Dades de l'usuari per accedir".

**Dades de l'associació:**

- Nom de l'Associació: [Redacted]
- Telèfon de contacte: [Empty field]
- Adreça de l'Associació: [Redacted]
- Descripció de l'Associació: [Empty field]
- Preu de les adopcions: [Redacted]
- Ciutat on es troba: [Redacted with error message "Indiqui una ciutat"]

**Dades de l'usuari per accedir:**

- Nom de l'usuari: [Redacted]
- Primer cognom: [Redacted]
- Segon cognom: [Empty field]
- Correu electrònic: [Redacted]
- Contrasenya: [Redacted]

At the bottom of the form is a "Registrar-se" button and a "Tornar Enere" button.

On the right side, there are seven error messages in red boxes:

- El nom de l'associació és necessari
- L'adreça de l'associació és necessària
- El preu de les adopcions és necessari
- Es necessari indicar la ciutat on es troba l'associació
- El nom de l'usuari és necessari
- El primer cognom és necessari
- El correu electrònic ha de tenir un format vàlid
- La contrasenya ha de contenir un mínim de 6 caràcters i un d'ells ha de ser un número

II-lustració 14. Formulari de registre amb errors



## 8.3 Pantalla d'identificació

Els usuaris que hagin registrat la seva associació o que hagin rebut la invitació per part d'un altre usuari podran accedir a l'aplicació indicant la seva direcció de correu electrònic i la seva contrasenya.



Il·lustració 15. Pantalla d'identificació

Abans de descriure les pantalles a les que pot accedir un usuari registrat, indicaré les funcions que són comunes per a aquestes pantalles. Primerament la capçalera com en el cas dels usuaris anònims, el títol “Adoptam” porta a l'usuari a la pantalla principal però en aquest cas també tanca la sessió de l'usuari automàticament. I el mateix passa amb l'enllaç amb nom ‘sortir’ que es troba al costat de la salutació.



Il·lustració 16. Capçalera usuaris registrats

Un altre punt en comú d'aquestes pantalles és el menú que hi ha sota la capçalera:

Perfil Associació

Dades dels Gossos

Llistat d'usuaris

## II-lustració 17. Menú usuaris registrats

En aquest menú hi ha tres opcions:

- Perfil Associació: que dirigeix a l'usuari a la pantalla amb les dades de la seva associació.
- Dades dels Gossos: que dirigeix a l'usuari a la pantalla des d'on es fa la gestió dels gossos.
- Llistat d'usuaris: que dirigeix a l'usuari a la pantalla on es troba el llistat d'usuaris de l'associació.

## 8.4 Pantalla del perfil de l'associació

Des d'aquí l'usuari pot veure les seves dades personals i les dades de l'associació a la qual pertany, també podrà actualitzar-ne els camps i canviar la seva contrasenya.

The screenshot shows a user profile page with three main sections, each with an 'Actualitzar' button at the bottom. The top navigation bar has three tabs: 'Perfil Associació', 'Dades dels Gossos', and 'Llistat d'usuaris'. The 'Perfil Associació' tab is active.

- Dades de l'associació:**
  - Nom de l'Associació: Caniland Montcada
  - Telèfon de contacte: 656666656
  - Adreça de l'Associació: C/Vallensana km 7
  - Descripció de l'Associació: Associació molt implicada e
  - Preu de les adopcions: 150
  - Ciutat on es troba: Montcada i Reixac
- Les teves dades:**
  - Nom: Sandra
  - Primer cognom: Iglesias
  - Segon cognom: Rodriguez
  - Correu electrònic: sandra@gmail.com
- Canvi de Contrasenya:**
  - Contrasenya actual: [input field]
  - Contrasenya nova: [input field]
  - Repetir contrasenya: [input field]

## II-lustració 18. Pantalla perfil associació

## 8.5 Pantalla de gestió dels gossos

Des d'aquesta pantalla es podran realitzar totes les funcionalitats relacionades amb la gestió dels gossos. Al accedir, el primer que l'usuari veurà serà el llistat dels gossos que ha afegir la seva associació:

Gossos de l'Associació												
Nom	Edat	Pes	Sexe	Mida	Raça	Color	Descripció	Característiques	Adoptat?	Té pedigree?	Editar	Esborrar
Bob	10 anys	9.7	Femella	Mitjà	Bobtail	Blanc	Gossa tímida però molt bona	[Juganer, Carinyós, Afable, Tímid, Poruc, Submis]	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Lluna	7 anys	Descone	Femella	Mitjà	Pastor Alemán	Marró i negre	Gos molt bó	[Tranquil, Carinyós, Complexió forta, Complexió atlètica]	<input type="checkbox"/>	<input type="checkbox"/>		
Simón	7 anys	5.6	Mascle	Mitjà	Beagle	Blanc amb taques negres	Gos molt actiu	[Tranquil, Carinyós, Afable, Pèl llarg, Tímid]	<input type="checkbox"/>	<input type="checkbox"/>		

Pàgina: 1 de 1

Afegir més informació

Afegir un Gos

### Il·lustració 19. Pantalla gestió dels gossos

A la taula apareixerà tota la informació sobre els gossos i totes aquestes dades es podran modificar clicant sobre el llapis de la columna 'Editar'. Un cop feta l'edició s'hauran de guardar els canvis polsant la icona del *check* verda, o si pel contrari es vol cancel·lar l'edició també es podrà prémer sobre la creu vermella.

Una altra funcionalitat d'aquesta taula es que es podran eliminar els gossos prement sobre el botó de la columna 'Esborra'. I pels casos en que el gos hagi estat adoptat es podrà canviar l'estat de la columna 'Adoptat?'.

Més avall, sota la taula es troben dos panells, el d'Afegir més informació es per poder associar, actualitzar o eliminar una foto d'un gos:

Gossos de l'Associació												
Nom	Edat	Pes	Sexe	Mida	Raça	Color	Descripció	Característiques	Adoptat?	Té pedigree?	Editar	Esborrar
Bob	10 anys	9.7	Femella	Mitja	Bobtail	Blanc	Gossa tímida però molt bona	[Juganer, Carinyós, Afable, Tímid, Poruc, Submis]	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Lluna	7 anys	Descone	Femella	Mitja	Pastor Alemany	Marró i negre	Gos molt bó	[Tranquil, Carinyós, Complexió forta, Complexió atlètica]	<input type="checkbox"/>	<input type="checkbox"/>		
Simón	7 anys	5.6	Masclle	Mitja	Beagle	Blanc amb taques negres	Gos molt actiu	[Tranquil, Carinyós, Afable, Pèl llarg, Tímid]	<input type="checkbox"/>	<input type="checkbox"/>		

Pàgina: 1 de 1


Afegir més informació

Afegeix una foto

+ Selecciona Afegir

\*Si el gos ja té associada una foto, pots afegir-ne una de nova que sobreescrigui l'actual

Foto actual



Eliminar

## II-lustració 20. Panell per afegir més informació

Però per fer-ho s'haurà de seleccionar prèviament un gos de la llista. Els botons que apareixen són els de 'Selecciona' per escollir una foto, el d'Afegir que es per associar la foto al gos i el d'Eliminar que es per eliminar la foto si ja en té una. També el missatge ens indica que podem sobreescrivre la foto afegint-ne una de nova sense haver d'eliminar-la abans.

El següent panell, que quedaria a sota d'aquest, seria el d'Afegir un gos i que conté el formulari amb les dades que cada gos tindrà associades:

Afegir un Gos

Nom del gos

Edat

Pes (kg)

Mida

Sexe

Color

Descripció

Raça

Característiques

Té pedigree?

Guardar

## II-lustració 21. Panell per afegir un gos

En aquest cas quasi tots camps són obligatoris excepte el pes, la descripció o les característiques però sí que és necessari indicar-ne la resta de camps encara que per exemple de la raça no s'estigui convençut al 100%. En aquests casos, en un altre moment es podrà editar i posar-ne el correcte.

D'aquest formulari molts camps són d'introducció de text, hi ha dos camps que són la mida i el sexe que són de selecció, un camp que és un *checkbox*, la raça és un camp que es completa a mesura que s'escriu i el camp per a les característiques s'ha fet amb selecció múltiple. Per últim per afegir un gos s'haurà de prémer el botó 'Guardar', si tot ha sortit bé sortirà un missatge de confirmació i s'actualitzarà la taula.

## 8.6 Pantalla de llistat d'usuaris

Aquesta pantalla conté el llistat dels usuaris que formen part de l'associació i que per tant se'ls ha hagut d'enviar una invitació. Des d'aquest llistat el que es podrà fer és eliminar usuaris però no es podran modificar les seves dades.

Email	Nom	Primer Cognom	Segon Cognom	Rol Administrador	Guardar	Eliminar
joell@gmail.com	Joel	Iglesias		<input type="checkbox"/>	<input type="button" value="G"/>	<input type="button" value="X"/>
raqui@gmail.com	Raquel	Rodriguez		<input checked="" type="checkbox"/>	<input type="button" value="G"/>	<input type="button" value="X"/>
carlos@gmail.com	Carlos	Amador		<input checked="" type="checkbox"/>	<input type="button" value="G"/>	<input type="button" value="X"/>
sandra@gmail.com	Sandra	Iglesias	Rodriguez	<input checked="" type="checkbox"/>	<input type="button" value="G"/>	<input type="button" value="X"/>

Pàgina: 1 de 1

Invita a altres usuaris perquè puguin col·laborar!

Nom:

Primer cognom:

Correu electrònic:

II-lustració 22. Pantalla llistat usuaris

El formulari que hi ha sota la taula serveix per enviar invitacions a altres usuaris perquè col·laborin en la gestió dels gossos de l'associació i només caldrà indicar-ne un nom, un cognom i la direcció de correu electrònic. Al prémer el botó 'Enviar invitació' es generarà una contrasenya automàticament que serà afegida al missatge que s'enviarà a l'usuari perquè pugui accedir a l'aplicació.

## 9 Conclusions

Trobo que ha sigut una experiència molt enriquidora encara que també ha sigut dura. El desenvolupament del projecte ha tingut la seva complexitat pel que fa a la integració de totes aquestes eines però també és cert que s'aprèn molt dels errors i els entrebancs.

Personalment crec que he après molt de totes les eines que he utilitzat, principalment de les noves com ho ha sigut d'hibernate i de primefaces ja que no havia treballat mai amb elles, però també de les eines amb les que ja havia fet alguna pràctica i de les que he pogut aprendre alguna cosa nova.

Pel que fa a l'aplicació opino que compleix els requisits i funcionalitats que s'esperaven a l'inici del projecte, excepte que un gos pugui tenir múltiples fotografies associades.

Encara que per altra banda vaig incloure alguna funcionalitat nova que vaig veure útil sobre la marxa. Per tant, el resultat del projecte ha estat una aplicació finalitzada i preparada per a la seva posada en marxa.

En conclusió, m'ha resultat molt satisfactòria la creació i desenvolupament de l'aplicació ara que ja es veu finalitzada.

# 10 Bibliografia

## Documentació consultada

Apache Tomcat 7. Documentation Index. Consultada feta a:  
<http://tomcat.apache.org/tomcat-7.0-doc/index.html>

Hibernate ORM. Documentation. Consultada feta a: <http://hibernate.org/orm/>

PrimeFaces Showcase. Consultada feta a:  
<http://www.primefaces.org/showcase/>

JavaMail. Oracle Documentation. Consultada feta a:  
<http://www.oracle.com/technetwork/java/javamail/index.html>

PostgreSQL. PostgreSQL 7.3.21 Documentation. Consultada feta a:  
<https://www.postgresql.org/docs/7.3/static/jdbc-binary-data.html>

Aula Virtual UM. JPA Aplicaciones Distribuidas. Consulta feta a:  
[https://aulavirtual.um.es/access/content/group/3871\\_G\\_2011\\_N\\_N/Teoria/T5B%20-%20JPA.pdf](https://aulavirtual.um.es/access/content/group/3871_G_2011_N_N/Teoria/T5B%20-%20JPA.pdf)

## “Com fer” i exemples

Java Code Geeks. Hibernate Query Language Example. Consultada feta a:  
<https://examples.javacodegeeks.com/enterprise-java/hibernate/hibernate-query-language-example/>

Java Web Development Blog. Hiberante and postgresQL configuration with maven. Consultada feta a: <http://jvmhub.com/2014/10/22/hibernate-and-postgresql-configuration-with-maven/>

JournalDev. Spring Hibernate Integration Example Tutorial. Consulta feta a:  
<http://www.journaldev.com/3524/spring-hibernate-integration-example-tutorial>

JournalDev. Hibernate Tomcat JNDI Datasource Example. Consulta feta a: <http://www.journaldev.com/2905/hibernate-tomcat-jndi-datasource-example-tutorial>

Mkyong. JSF 2.0 Tutorials. Consulta feta a: <http://www.mkyong.com/tutorials/jsf-2-0-tutorials/>

Core Servlets. JSF 2 Page Navigation. Consulta a: <http://courses.coreservlets.com/Course-Materials/pdf/jsf/jsf2/JSF2-Page-Navigation.pdf>

## Obtenció de dades

Viquipèdia. Ciutats de Catalunya per població. Consulta feta a: [https://ca.wikipedia.org/wiki/Ciutats\\_de\\_Catalunya\\_per\\_poblaci%C3%B3](https://ca.wikipedia.org/wiki/Ciutats_de_Catalunya_per_poblaci%C3%B3)

Perros. Razas de perros. Consulta feta a: <http://www.perros.com/razas/>

Gencat. Conèixer. Consulta feta a: <http://web.gencat.cat/es/temes/catalunya/coneixer/>