

PROJECTE FINAL DE CARRERA

Desenvolupament d'una aplicació amb Microsoft .NET

PROTECTORA D'ANIMALS



Memòria del projecte

**Enginyeria tècnica
en informàtica de sistemes (ETIS)**

GENER 2011

Autor: Marc Segura Valls

Consultor: Jordi Sánchez Cano



Índex

Índex figures.....	4
1. Introducció	5
1.1. Justificació del TFC	5
1.2. Objectiu	5
1.3. Enfocament i mètode seguit	6
1.4. Planificació inicial i real	7
1.5. Eines i tecnologies emprades.....	11
1.6. Productes obtinguts.....	13
2. Anàlisi.....	14
2.1. Descripció el projecte	14
2.2. Requeriments funcionals.....	15
2.3. Requeriments no funcionals.....	16
2.4. Diagrames de casos d'ús.....	16
2.4.1 Petició alta nou usuari.....	17
2.4.2 Realitzar cerca animal.....	18
2.4.3 Adopció online.....	19
2.4.4 Publicar cas pèrdua.....	20
2.4.5 Gestió Animals.....	21
2.4.6 Gestió Usuaris.....	22
2.5. Especificació textual dels casos d'ús.....	23
2.5.1 Especificació cas d'ús petició registre.....	23
2.5.2 “ cas d'ús registrar dades del nou usuari.....	23
2.5.3 “ “ “ verificar petició.....	24
2.5.4 “ “ “ notificar incidència petició usuari.....	24
2.5.5 “ “ “ modificar BBDD peticions.....	25
2.5.6 “ “ “ registrar alta usuari.....	25
2.5.7 “ “ “ notificar alta usuari.....	26
2.5.8 “ “ “ autenticar.....	26
2.5.9 “ “ “ mostrar informació.....	27
2.5.10 “ “ “ obtenir fitxa	28
2.5.11 “ “ “ publicar al Facebook.....	28
2.5.12 “ “ “ adoptar online.....	29
2.5.13 “ “ “ notificar animal llest.....	29
2.5.14 “ “ “ registrar adopció.....	30
2.5.15 “ “ “ publicar anunci pèrdua.....	30

2.5.16	Especificació cas d'ús alta animal.....	31
2.5.17	“ “ “ eliminar animal.....	31
2.5.18	“ “ “ modificar animal.....	32
2.5.19	“ “ “ seleccionar i publicar cas urgent.....	33
2.5.20	“ “ “ eliminar usuari.	34
2.5.21	“ “ “ entrar usuari (manera manual).....	34
2.5.22	“ “ “ modificar usuari.....	35
2.6.	Altres diagrames (Diagrames de seqüència).....	36
3.	Disseny.....	37
3.1.	Arquitectura de l'aplicació.....	37
3.2.	Disseny de la interfície de l'usuari.....	40
3.2.1	ASP.NET i AJAX.....	41
3.2.2	Controls i menús d'usuari.....	47
3.2.3	Diagrama de flux de les pantalles.....	49
3.2.4	Captures d'algunes pàgines de mostra del projecte.....	49
3.3.	Diagrama de classes.....	52
3.3.1	Model conceptual.....	52
3.3.2	Diagrama de classes.....	53
3.3.3	Mètodes més rellevants de les classes principals.....	55
3.4.	Disseny de la base de dades.....	57
3.4.1	Taules de la base de dades.....	57
3.4.2	Diagrama de la base de dades.....	58
3.4.3	Declaració de les taules i els seus atributs.....	59
3.4.4	Linq to SQL.....	60
3.5.	Arquitectura física.....	61
4.	Línies de desenvolupament futures i conclusions	62
5.	Bibliografia.....	63

Índex figures

Figura 1. Etapes de desenvolupament i documentació generada.....	6
Figura 2. Dates clau de les tasques del projecte.....	9
Figura 3. Diagrama de Gantt del projecte.....	10
Figura 4. Codi de colors cassos d'ús.....	16
Figura 5. Diagrama cas d'ús petició alta nou usuari.....	17
Figura 6. “ “ “ realitzar cerca.....	18
Figura 7. “ “ “ adopció online.....	19
Figura 8. “ “ “ publicar cas pèrdua.....	20
Figura 9. “ “ “ gestió Animals.....	21
Figura 10. “ “ “ gestió Usuaris.....	22
Figura 11. Diagrama de seqüència Login (Autenticar).....	36
Figura 12. Model tres capes.....	37
Figura 13. Solució tres projectes igual a tres capes.....	38
Figura 14. ScriptManager a la pàgina principal.....	41
Figura 15. ScriptManagerProxy i UpdatePanel a la pàgina entrar animals (exemple.....	42
Figura 16. Resultat ScriptManager i UpdatePanel(AJAX).....	43
Figura 17. Control Timer amb l'event OnTick de AJAX.....	44
Figura 18. Utilització del CalendarExtender de AJAX Control Toolkit.....	45
Figura 19. Utilització de PasswordStrenght de AJAX Control Toolkit.....	45
Figura 20. Menús usuari.....	46
Figura 21. Navegador de registres.....	47
Figura 22. Buscador d'animals.....	47
Figura 23. Mostra navegador GoogleMaps.....	48
Figura 24. Diagrama de flux de les pantalles.....	49
Figura 25. Mostra pàgina principal (Home).....	50
Figura 26. Mostra “ gestió animals.....	50
Figura 27. Mostra “ buscar animals.....	51
Figura 28. Mostra “ mostrar fitxa.....	51
Figura 29. Model conceptual.....	52
Figura 30. Herència usuaris.....	53
Figura 31. Reificació.....	53
Figura 32. Diagrama de classe.....	54
Figura 33 Diagrama base de dades.....	58
Figura 34. Diagrama Linq to SQL en el Visual Basic 2010.....	60
Figura 35. Arquitectura física.....	61

1. Introducció

En aquesta primera part es dona al lector la informació per tal de que tingui una primera presa de contacte amb el projecte

1.1. Justificació del TFC

L'àrea que he escollit per la realització del treball de final de carrera és la plataforma .NET, concretament la versió 4, la més actual a dia d'avui. El que es pretén en aquesta àrea és que ens introduïm de manera sòlida a la ja popular plataforma de Microsoft de desenvolupament de programari.

Per conèixer aquesta plataforma he optat per crear una aplicació web que sigui capaç de gestionar una protectora d'animals que l'únic que buscarà serà fomentar i facilitar l'adopció d'animals abandonats. La missió principal és crear un entorn molt enriquit de diferents tecnologies que ajudin a captivar a un perfil d'usuari incondicional, fent que aquest se senti part del portal. Per això buscarem crear funcions molt interactives, pràctiques i útils que ajudaran a que aquest futur lloc web sigui un referent dins el sector.

Crearem tres perfils d'usuari: el primer és l'administrador que gestiona els animals i també els usuaris i les relacions que existeixen entre ells, el segon és l'usuari registrat que té certs privilegis que no té el tercer perfil que és l'usuari anònim.

1.2. Objectiu

L'Objectiu principal que m'he marcat és el de treure el màxim de rendiment de les tecnologies i llenguatges que giren al voltant del .NET i que aquestes cooperin entre elles al màxim permès.

Dels diversos llenguatges que ofereix .NET m'he proposat aprendre el C# i aquesta decisió ve motiva per la similitud que té amb el Java que tant he tocat durant la carrera i la seva fidelitat a la programació orientada a objectes.

Al tractar-se d'una aplicació web, aquesta s'haurà de fer en el model de desenvolupament Web unificat per Microsoft, es tracta de ASP.NET i per això un clar objectiu és conèixer a fons aquesta tecnologia i a més combinar-la amb AJAX. Per això s'haurà de conèixer el funcionament i les possibilitats que ofereix AJAX.

També vull aprendre a crear una solució amb el Visual Studio que em permeti implementar un disseny d'una arquitectura de tres capes: Presentació, negoci i dades i que estiguin aïllades entre sí.

Conèixer i aplicar el Linq to SQL, el més actual i versàtil component del framework.NET per treballar amb bases de dades.Evidenment a la capa de dades.

I finalment haig d'entrar a fons amb el que es pot arribar a fer amb un servidor de base de dades, en el meu cas el SQL server 2005 i la seva eina d'administració Microsoft SQL Server Managment Studio: Crear scripts amb totes les instruccions i declaracions que ens permetin crear la base de dades amb un sol clic, gestionar usuaris i els seus privilegis, seguretat de la base de dades etc, etc...

1.3. Enfocament i mètode seguit

A l'assignatura de enginyeria del programari vaig veure un model sobre el cicle de vida d'un programa, i que és amb el que em vull guiar per el meu projecte. Aquest model és l'anomenat "cicle de vida clàssic" (també anomenat cicle de vida en cascada). La idea d'aquest model és que cada etapa del projecte es realitza quan acaba l'anterior. Veiem totes les etapes que durem a terme basades amb aquest model en la següent il·lustració:

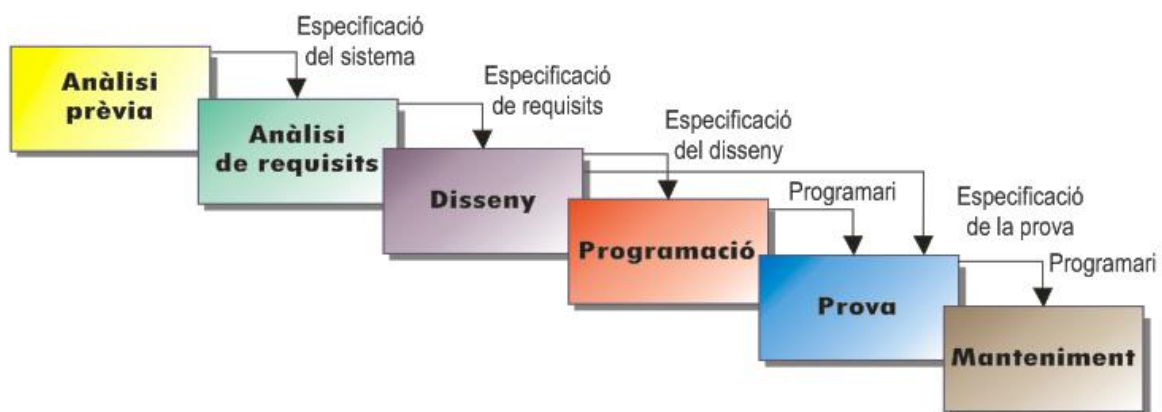


Figura 1. Etapes de desenvolupament i documentació generada

1.4. Planificació inicial i real

El projecte està dividit en quatre etapes que coincideixen amb les quatre Pacs de l'avaluació continuada de l'assignatura: pla de treball més preparació de recursos, anàlisi i disseny, implementació i la fase final que consisteix en la elaboració d'aquesta memòria més una presentació virtual. La temporització està marcada per les dates clau d'entrega de cada fase, a més s'haurà de confeccionar una documentació per cada una .

Al inici es va preveure una planificació de la qual en tot moment s'ha intentat seguir amb la màxima fidelitat possible. Anem a veure-la:

PAC 1 Pla de treball i preparació de recursos: *(del 21/09/2010 al 4/10/2010)*

- ✚ Detallar la descripció del projecte , els objectius, els requeriments, estudi de la idoneïtat del projecte i la planificació.
- ✚ Descarregar de la web que ens va passar el consultor en el taulell de l'aula <http://www.microsoft.com/latam/educacion/programas/aa/elms.aspx>. tot el programari necessari per realitzar el projecte per llavors instal·lar-lo.
- ✚ Adquirir algun llibre recomanat i buscar informació sobre els diferents programes que s'utilitzen. També començar a explorar webs especialitzades en visual basic, ASP i AJAX i Microsoft SQL server, i registrar-nos a fòrums per quan tinguem dubtes poder-los exposar ràpidament.
- ✚ Començar a llegir els materials i conèixer l'entorn .NET i Microsoft SQL.

PAC 2 Anàlisi i disseny: *(del 5/10/2010 al 1/11/2010)*

- ✚ Detallar els requeriments funcionals de l'aplicació i del disseny tècnic seguint les tècniques que vaig aprendre a l'assignatura d'Enginyeria del programari: recollida de documentació de requisits, especificació dels casos d'ús , dels diagrames d'activitat i de seqüència més rellevants, també especificació de les classes d'anàlisi , especificació de la base de dades, etc,etc...

- ✚ Dissenyar el que s'ha especificat anteriorment .També s'haurà de pensar amb la reutilització de codi, i així podem aprofitar classes o controls ja madurs, i que ens siguin útils i ens ajudin a guanyar temps.
- ✚ Disseny de la persistència, que seria tot el disseny relacionat amb la base de dades: persistència amb base de dades orientada a objectes i model de la base de dades relacional.
- ✚ Disseny de la interfície gràfica d'usuari.
- ✚ Paral·lelament anirem estudiant el llenguatge visual c# i les altres tecnologies i eines a utilitzar com ASP.NET , AJAX, HTML, CSS etc,etc..

PAC 3 Implementació: *(del 2/11/2010 al 20/12/2010)*

- ✚ Implementar l'aplicació web a partir de la documentació que s'ha generat a la Pac 2 dins la plataforma .NET amb els components que hem escollit.
- ✚ Penjar l'aplicació en un servidor IIS i fer totes les proves convenients per assegurar el funcionament correcte del lloc web, i resoldre tots els problemes trobats.
- ✚ Crear un manual d'instal·lació i configuració que expliqui pas a pas com posar en marxa l'aplicació.
- ✚ Crear un manual d'usuari que serveixi per donar una introducció ràpida a l'usuari.

PAC 4 Memòria i Presentació virtual: *(del 21/12/2010 al 10/01/2010)*

- ✚ Realitzar una memòria del projecte completa i detallada i que ha d'incloure el treball realitzat durant tot el curs (És aquest document).
- ✚ Crear una presentació virtual sintetitzada que serveixi de suport a la defensa del projecte davant del tribunal.També serà important que en aquesta presentació també es vegi una demostració en viu de l'aplicació.

A la figura 2 podem veure les dates clau i les tasques definides inicialment, i a la figura 3 de l'altre pàgina es pot veure el diagrama de Gantt també definit prèviament, on es pot observar amb detall aquesta previsió inicial.

	Task Name	Duration	Start	Finish	Predecessors
1	<i>Treball Final de Carrera</i>	94 days	Tue 21/09/10	Thu 27/01/11	
2	- Pla de treball	11 days	Tue 21/09/10	Mon 04/10/10	
3	Llegir enunciat elecció proposta	1 day	Tue 21/09/10	Tue 21/09/10	
4	Llegir manual introducció .NET i altres	3 days	Wed 22/09/10	Sat 25/09/10	3
5	Descàrrega i instal·lació programari	7 days	Sun 26/09/10	Mon 04/10/10	4
6	Elaboració document pla de treball	6 days	Mon 27/09/10	Mon 04/10/10	
7	Entrega Pla de treball (PAC1)	0 days	Mon 04/10/10	Mon 04/10/10	6;5
8	-Anàlisi i disseny	20 days	Tue 05/10/10	Mon 01/11/10	7
9	Estudi de Visual c# ASP i AJAX	20 days	Tue 05/10/10	Mon 01/11/10	
10	Recollida documentació dels requisits	4 days	Tue 05/10/10	Sat 09/10/10	
11	Especificació BBDD	1 day	Mon 11/10/10	Mon 11/10/10	10
12	Especificació classes d'anàlisi	3 days	Tue 12/10/10	Thu 14/10/10	11
13	Especificació casos d'ús i altres	2 days	Fri 15/10/10	Mon 18/10/10	12
14	Disseny BBDD	2 days	Tue 19/10/10	Wed 20/10/10	13
15	Disseny estructura estàtica (Diagrama classes)	3 days	Thu 21/10/10	Mon 25/10/10	14
16	Disseny estructura dinàmica (Casos d'ús, etc..)	3 days	Tue 26/10/10	Thu 28/10/10	15
17	Disseny interfície usuari	2 days	Fri 29/10/10	Mon 01/11/10	16
18	Entrega documentació (PAC2)	0 days	Mon 01/11/10	Mon 01/11/10	17;9
19	-Implementació	35 days	Tue 02/11/10	Mon 20/12/10	18
20	Implementació de l'aplicació a la plataforma .NET	21 days	Tue 02/11/10	Tue 30/11/10	
21	Provar i correcció d'errors	11 days	Wed 01/12/10	Wed 15/12/10	20
22	Crear Manual d'instal·lació	3 days	Thu 16/12/10	Mon 20/12/10	21
23	Entrega documentació (PAC3)	0 days	Mon 20/12/10	Mon 20/12/10	22
24	-Memòria i presentació virtual	15 days	Tue 21/12/10	Mon 10/01/11	23
25	Realitzar memòria	10 days	Tue 21/12/10	Mon 03/01/11	
26	Realitzar presentació virtual	11 days	Mon 27/12/10	Mon 10/01/11	
27	Entrega documentació (PAC4 i Final)	0 days	Mon 10/01/11	Mon 10/01/11	26
28	-Debat virtual	4 days	Mon 24/01/11	Thu 27/01/11	

Figura 2. Dates clau de les tasques del projecte

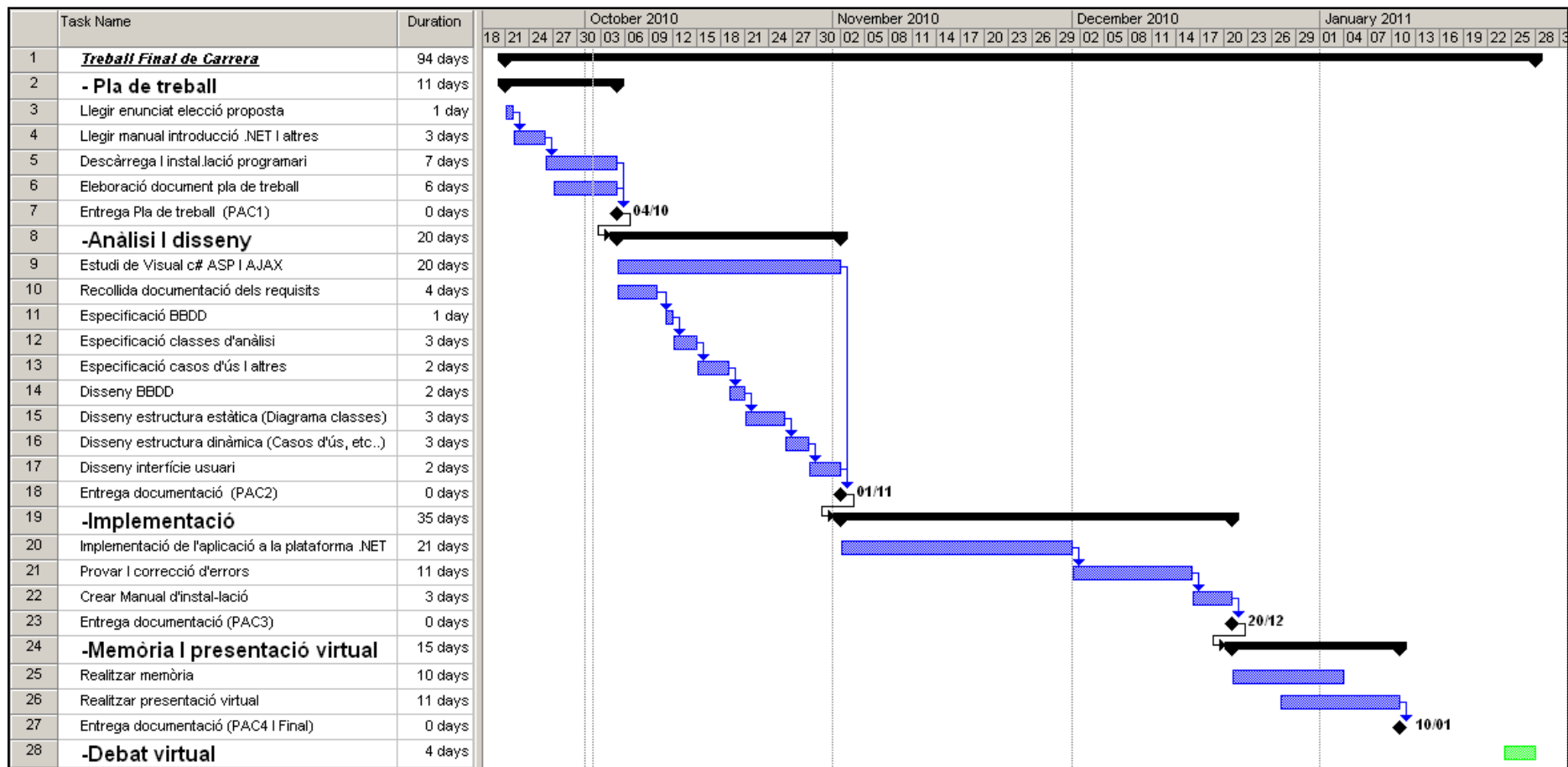


Figura 3. Diagrama de Gantt del projecte

Les diferències entre la planificació inicial i la real han estat poques ja que per obtenir una avaluació del consultor s'han entregat totes les documentacions abans de les dates claus.

La única tasca que no s'ha fet dins la fase que s'havia previst ha estat el disseny de la interfície gràfica que al final s'ha fet a la fase d'implementació, des de zero en Paral·lel amb la programació de l'aplicació, ja que el consultor va dir que no calia que gastéssim moltes hores en el disseny de la interfície gràfica per tal de tenir més temps per l'estudi de la plataforma i les tecnologies que havíem triat.

Les altres diferències respecte a la planificació inicial han estat de temps, en algunes tasques s'ha tardat menys del que s'havia previst i en altres una mica més o fins i tot ha faltat temps per assolir tots els requeriments plantejats a l'inici. Les més destacades han estat a la part d'implementació, on s'han necessitat més de vint-i-un dies per fer la implementació de l'aplicació i ha faltat temps per fer més proves i corregir errors, a més per fer el manual d'usuari i d'instal·lació només he disposat d'un dia.

1.5. Eines i tecnologies emprades

Per dur a terme un projecte d'aquesta magnitud calen un gran ventall d'eines que ofereix el mercat de programari tant de lliure circulació com de pagament.

Jo he escollit les següents:

- ✚ **Visual Studio 2010** com a IDE (Entorn de desenvolupament integrat) i és el que ofereix el creador de la plataforma .NET, per tant el més fiable i del que podem trobar més informació per la xarxa i en els llibres.
- ✚ **Microsoft.NET Framework 4** que és el motor més actual de la plataforma .NET, i aquest ja l'instal·la el visual Studio 2010.
- ✚ **Visual C# 4** , llenguatge de programació basat en la programació orientada a objectes.
- ✚ **Microsoft SQL Server 2005 Developer edition** com a motor de base de dades.
- ✚ **SQL Server Management Studio**, eina essencial per administrar un servidor de base de dades.
- ✚ **Linq to SQL** que és un component del framework que proporciona una infraestructura en temps d'execució per administrar les dades relacionals com objectes dins del Visual Studio.

- ✚ **ASP.NET**, també integrat a la plataforma .NET i és l'artífex de que una aplicació pugui ser Web i per tant utilitzada des de qualsevol explorador
- ✚ **AJAX** que engloba un grup de tecnologies que cooperen entre sí com: XML, HTML, Javascript etc, etc... i serveix per lliurar de càrrega de treball al servidor dient a terme una comunicació asíncrona amb aquest que està en segon pla. També m'he instal·lat la llibreria AJAX control toolkit que ofereix controls d'extensió molt bons per donar més vida als controls convencionals de ASP.NET.
- ✚ **HTML + CSS** per tal de crear i ordenar les pàgines de l'aplicació Web utilitzo les més primàries instruccions de HTML i per donar forma aquestes pàgines he programat una fulla d'estils en cascada CSS.
- ✚ **Microsoft Expression Web**, per fer algun prototip de les pantalles.
- ✚ **IIS (Internet Information Server) 5.1**, com a servidor d'aplicacions Web i que és el que porta el sistema operatiu que he utilitzat, el Windows XP Professional.
- ✚ **Microsoft Word 2007**, utilitzat per la elaboració e tota la documentació.
- ✚ **Corel Draw X3**, programa de dibuix vectorial utilitzat per fer la majoria de il·lustracions de la documentació, inclòs els diagrames de casos d'ús, diagrama de classes, etc...
- ✚ **Microsoft Project 2007**, per la creació del diagrama de Gantt a la part de planificació.
- ✚ **Microsoft PowerPoint**, per la creació d'una part de la presentació virtual del projecte.
- ✚ **Audicity**, és un programa de so que m'ha permès obtenir una bona qualitat de so a la presentació.
- ✚ **Camtasia Studio 7**, per fer les captures en viu de la demostració de l'aplicació i de les diapositives del Powerpoint més l'edició final de so i vídeo per la presentació virtual.

1.6. Productes obtinguts

Durant la realització d'aquest projecte s'han generat els productes que es mostren a la taula següent juntament amb una breu explicació

Producte	Descripció
Pla de treball (PAC 1)	Document inicial que conté informació com: descripció del treball i objectius, estudi de la idoneïtat del projecte, anàlisi de requeriments a grans trets, tecnologies a utilitzar, tasques i recursos necessaris, la planificació i altre informació d'interès.
Anàlisi (PAC 2)	Aquest document presenta una descripció molt més detallada, conté un anàlisi de requeriments més exhaustiu on es detallen els processos més importants a implementar mitjançant diagrames de casos d'ús.
Disseny (PAC 2)	És la documentació que fa referència a l'arquitectura del sistema, el disseny de la base de dades i també el disseny del diagrama de classes.
Implementació (PAC 3)	Codi font de l'aplicació que és una solució de Visual Basic 2010 que conté tres projectes (un per cada capa), i també l'script per la creació de la base de dades.
Manuais (PAC 3)	Manual d'instal·lació i configuració de l'aplicació i un altre manual molt bàsic que té com a missió donar una introducció ràpida a l'usuari.
Memòria (PAC 4)	Aquest document
Presentació virtual (PAC 4)	Vídeo d'uns vint minuts de durada que és un complement de la memòria i que conté els punts més destacats i a més inclou una demostració en viu de l'aplicació i que ajuda a defensar el projecte davant d'un tribunal.

2. Anàlisi

En aquesta part hem de conèixer tot el que ha de fer la nostra aplicació i a partir d'aquí identificar els requeriments funcionals i els no funcionals.

A la Pac 2 es va fer aquesta feina amb la idea de plasmar-ho a la implementació i que al final no s'ha implementat tot exactament igual, ja que sobre la marxa es va creure oportú afegir noves tasques, sobretot per a l'administrador com la possibilitat eliminar i actualitzar usuaris i animals. Al principi només vaig preveure donar d'alta, i això no és suficient ja que un administrador ha de tenir el control absolut de les dades. A més, algunes tasques que en un principi semblaven viables, al final s'ha optat per no implementar-les com per exemple les notificacions per mail automàtiques del sistema.

Per aquesta raó es refà aquesta part deixant només les tasques (representades per els diagrames de casos d'ús) que al final s'han implementat.

Primer donarem una descripció del projecte i d'aquí obtindrem els requeriments del projecte.

2.1. Descripció del projecte

S'ha de desenvolupar una aplicació web capaç de gestionar una protectora d'animals, concretament de gossos i gats. L'ús d'aquest portal està obert tant a usuaris anònims com a usuaris registrats.

Els usuaris anònims igual que els registrats podran veure els animals de que disposa la protectora realitzant cerques filtrades segons el tipus (gat o gos), grandària (cadell, petit, mitjà i gran) i sexe i així obtenir una informació resumida i que només amb un clic podran veure la fitxa de l'animal amb moltes més dades: fotografies, raça, color, pes, comportament, etc... Un cop aquí si l'usuari està registrat i autenticat podrà demanar una adopció on-line i rebrà una notificació feta per un administrador quan l'animal es pugui passar a recollir.

A més tant els anònims com els registrats podran publicar notícies al facebook sobre un animal concret. Els usuaris registrats si en algun moment ells o algun conegut han perdut la seva mascota, podran publicar el cas en un apartat del portal dedicat a aquesta qüestió amb tota la informació oportuna: imatge, ubicació (mitjançant Google Maps) , informació de contacte, etc

La gestió d'aquest portal estarà suportada per els administradors, que s'encarregaran de tota l'entrada i manteniment de les dades dels animals i dels usuaris. Un administrador podrà fer totes les tasques d'un usuari normal i a més tindrà la possibilitat de seleccionar els casos més urgents (3 com a màxim) per a que apareguin en l'apartat corresponent i a la pàgina principal de forma rotativa.

2.2. Requeriments funcionals

Els actors que interactuen amb la nostra aplicació són : usuari anònim, usuari registrat i administrador. L'anònim podrà fer els mínims, l'usuari registrat tindrà més privilegis, i l'administrador tindrà el control total de l'aplicació.

Veiem els requeriments funcionals principals:

- **Cerca animals:** Tots els actors podran veure els animals de que disposa la protectora realitzant cerques filtrades i publicar el cas que es vulgui a la xarxa social facebook.
- **Registrar:** Un usuari anònim pot registrar-se per obtenir més drets en el portal com per exemple fer adopcions des de casa o publicar casos de pèrdues de les seves mascotes o d'algun conegut.
- **Login:** L'usuari registrat i l'administrador per poder gaudir dels seus privilegis té que entrar el seu nom usuari i la seva contrasenya. Tots els usuaris estan registrats a la mateixa taula però gràcies a un atribut anomenat administrador el sistema sap si és normal o administrador, i arrel d'aquesta informació donarà uns privilegis o bé uns altres. Un cop hagi acabat de fer la feina es desconnectaran de la sessió.
- **Gestió Usuaris i Animals:** L'administrador porta la gestió íntegra del portal, pot donar d'alta a nous usuaris, donar-los de baixa o bé actualitzar alguna de les seves dades i també registrar les seves adopcions. Amb els animals practicamente el mateix : donar d'alta, eliminar , actualitzar alguna de les dades i a més publicar casos urgents.
- **Publicar pèrdues:** Tant l'usuari registrat com l'administrador poden publicar un cas de pèrdua d'un animal amb una funció molt útil, que és la possibilitat de marcar la ubicació del lloc on es va veure la mascota per última vegada mitjançant unes llibreries proporcionades per el GoogleMaps.
- **Veure Casos Pèrdues:** Tots els usuaris del portal (inclòs els anònims) podran veure els casos de pèrdua i accedir a GoogleMaps per veure la ubicació on es va veure per última vegada la mascota.
- **Veure Casos Urgents:** Tots els usuaris del portal (inclòs els anònims) podran veure els casos urgents (que són un màxim de tres) i accedir a les dades de l'animal, a més veure el comentari especial per aquest apartat que ha entrat l'administrador.

2.3. Requeriments no funcionals

La interfície de tots els usuaris on realitzaran tots els requeriments funcionals és una interfície web gràcies a ASP.NET.

Les comunicacions entre interfície i lògica de negoci i aquesta amb la base dades ve donada per l'arquitectura de l'aplicació de tres capes que a un apartat posterior s'explicarà. Són tres projectes dins la mateixa solució (un per cada capa) .

2.4. Diagrames de casos d'ús

Per assolir els objectius d'aquest projecte, hem de confeccionar un gran nombre de diagrames de cas d'ús dels quals aquí només en mostrarem els més rellevants.

S'ha creat un codi de colors per associar més ràpid els casos d'ús amb les seves descripcions i així ajudar al lector tenir més agilitat a l'hora de moure's per el document. O sigui que la única raó de la creació d'aquest codi de colors és per el fet de que hi han molts cassos d'ús i qualsevol tècnica que ajudi a diferenciar-los entre ells serà d'agrair.

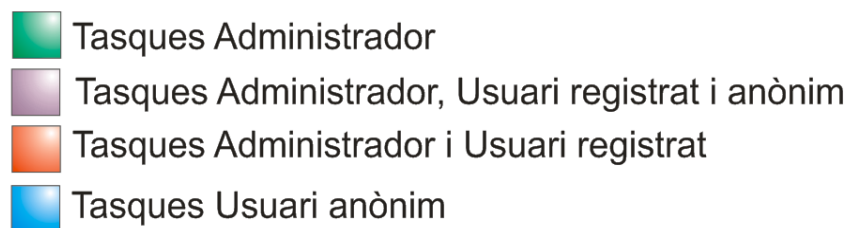


Figura 4. Codi de colors casos d'ús

2.4.1. Petició alta nou usuari

Quan un usuari anònim vol tenir més privilegis haurà de donar-se d'alta, i haurà de ser l'administrador que verifiqui les dades i notifiqui a l'usuari mitjançant un correu si el procés s'ha completat correctament o bé ha sorgit alguna incidència.

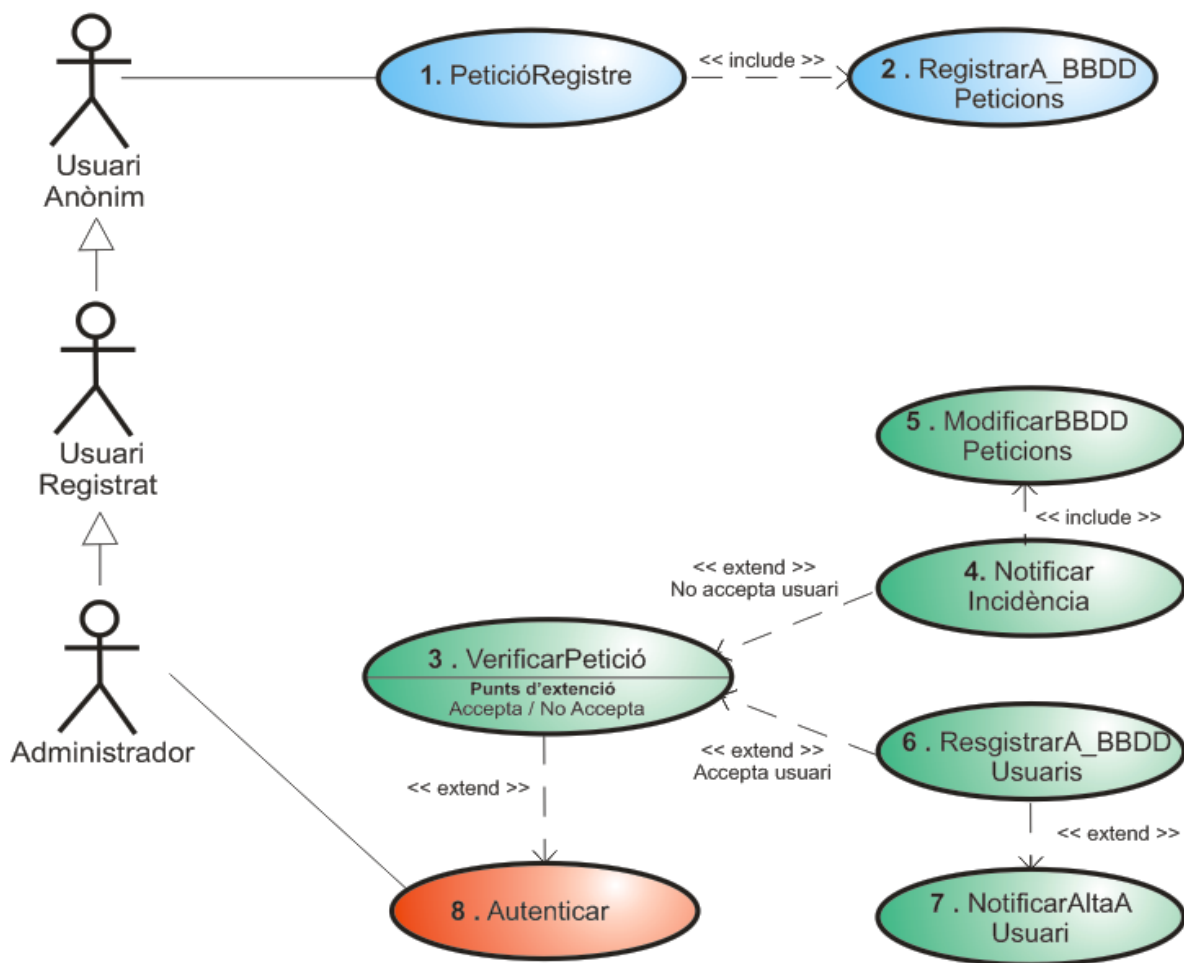


Figura 5. Petició alta nou usuari

2.4.2. Realitzar cerca

Els tres actors podran veure els animals de que disposa la protectora realitzant cerques. Es podran filtrar aquestes cerques segons tipus, grandària i sexe, i això es farà en el cas d'ús "Mostrar Informació". Al fer clic sobre un dels resultats es mostrarà la fitxa amb molts més detalls: fotografies, raça, color, pes, on s'ha trobat, etc... I a més tenen la possibilitat de publicar la notícia al facebook.

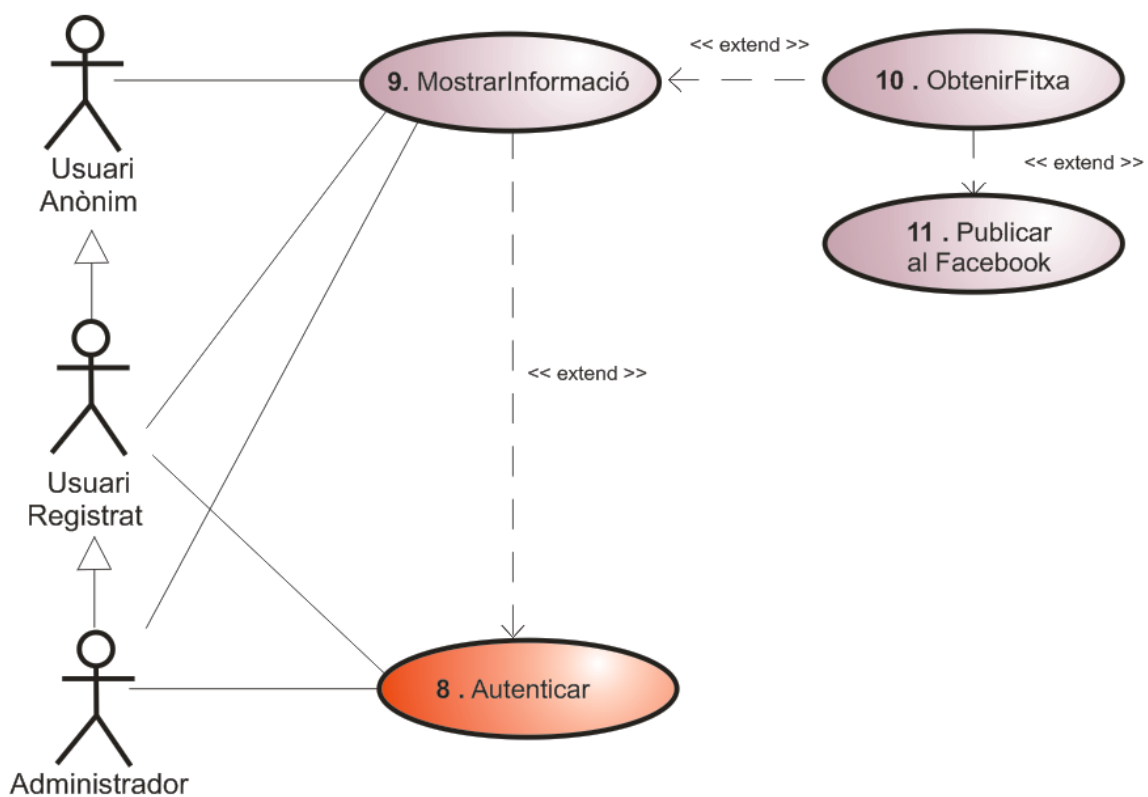


Figura 6. Realitzar cerca

2.4.3. Adopció online

Des d'una fitxa un usuari registrat (també Administrador) tindrà la possibilitat d'adoptar online. Quan un usuari faci una adopció online, sabrà quan ja pot passar a recollir la seva nova mascota mitjançant una notificació. Aquesta notificació la farà l'administrador i a més l'haurà de registrar a la base de dades.

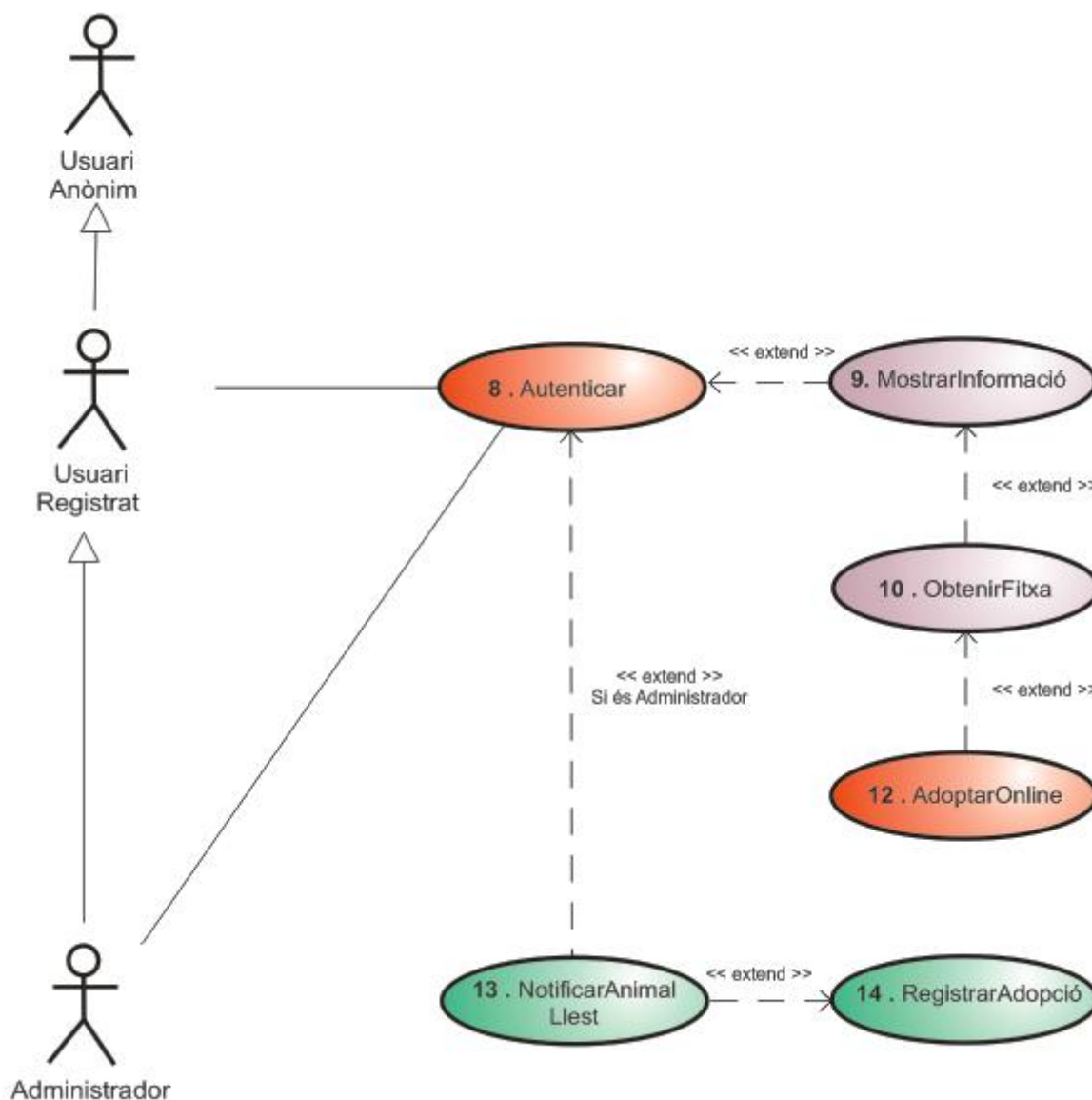


Figura 7. Adopció Online

2.4.4. *Publicar cas de pèrdua*

Si un usuari registrat perd la seva mascota, ho podrà publicar en el portal mostrant la seva fotografia, la ubicació del lloc on es va perdre mitjançant GoogleMaps, informació de contacte, etc.

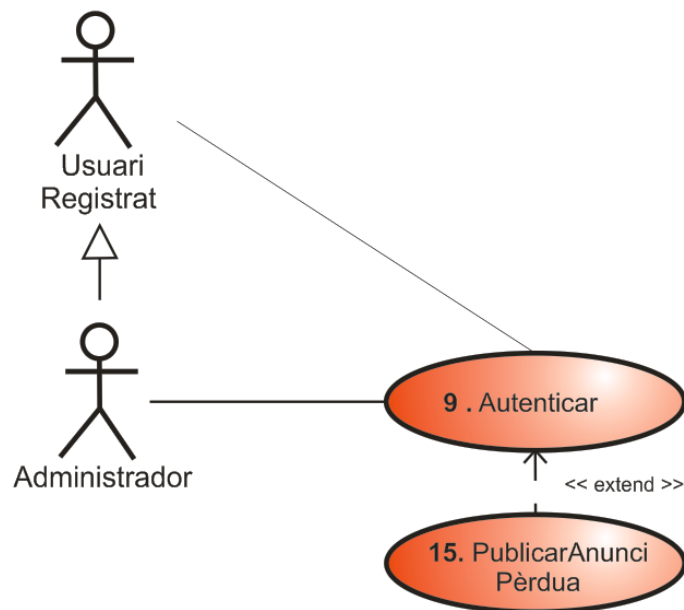


Figura 8. Publicar cas pèrdua

2.4.5. *Gestió Animals*

Els administradors entren les dades dels nous animals que arriben a la protectora i a més en fan el manteniment. No és el mateix eliminar que donar de baixa ja que per donar de baixa el que fa es modificar-se l'atribut de la taula dataBaixa i s'elimina quan s'ha produït algun error o confusió.

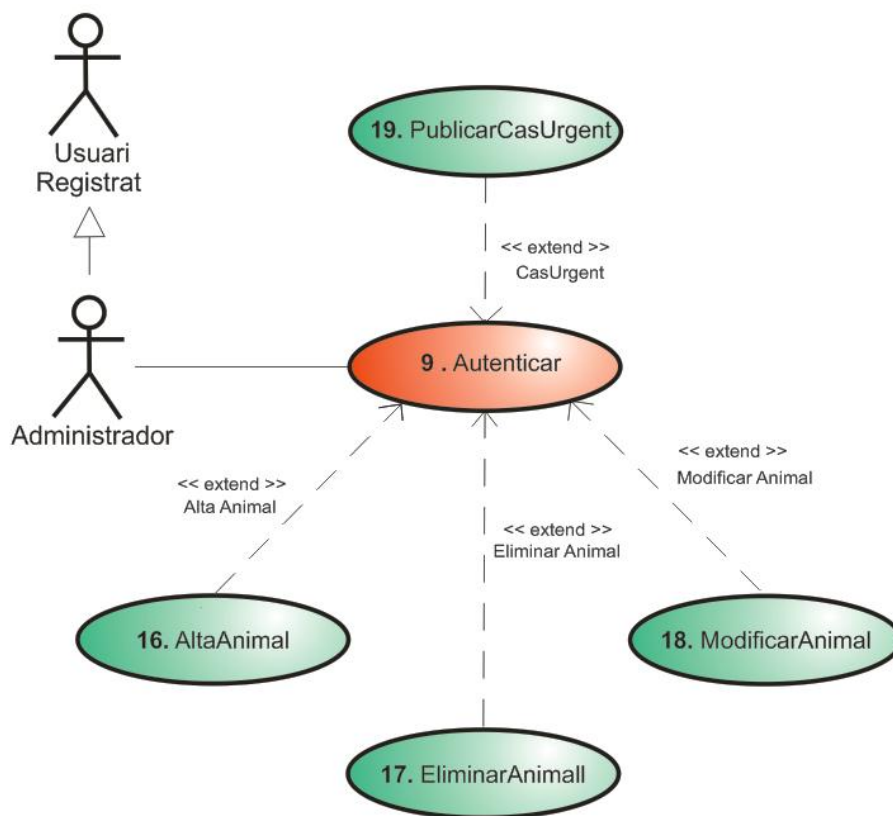


Figura 9. Gestió Animals

2.4.6. *Gestió Usuaris*

Els administradors poden donar d’alta a un nou usuari de manera automàtica a la pantalla de peticions usuaris i a més a la pantalla de gestió usuaris poden actualitzar qualsevol de les seves dades o bé eliminar un registre.No és el mateix eliminar que donar de baixa ja que per donar de baixa el que fa es modificar-se l’atribut de la taula dataBaixa i s’elimina quan s’ha produït algun error o confusió.

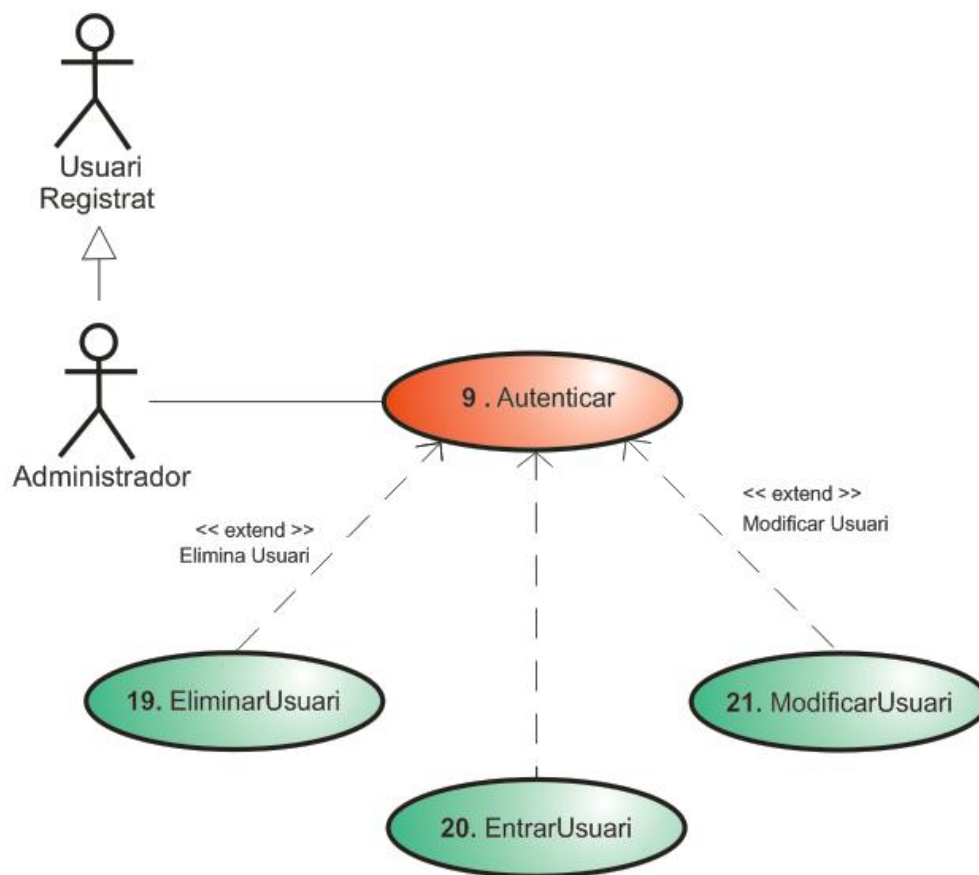


Figura 10. Gestió Usuaris

2.5. Especificació textual dels casos d'ús més rellevants

És important que coneixem els casos d'ús més importants del projecte donant-ne una especificació textual que ens permeti conèixer amb més precisió quines són les tasques d'aquests.

2.5.1. Cas d'ús petició registre

1. PeticióRegistre	
Resum de la funcionalitat	Un usuari anònim vol donar-se d'alta.
Actors	Usuari anònim.
Casos d'ús relacionats	2. RegistrarABBDD_Peticions
Precondició	L'usuari no està registrat i no té privilegis.
Postcondició	L'usuari esta a l'espera per ser donat d'alta
Procés normal principal	<ol style="list-style-type: none"> 1. L'usuari anònim accedeix a l'apartat de "Nou usuari" 2. L'usuari anònim omple el formulari per donar-se d'alta. 3. Un cop entrades les dades l'hi dona en el botó "enviar". 4. S'engega el cas d'ús 2. RegistrarABBDD_Peticions.
Alternatives de procés i excepcions	<ol style="list-style-type: none"> 3a. Algunes de les dades que ha entrat l'usuari són incorrectes(problemes de sintaxis). <ol style="list-style-type: none"> 3a1. El sistema demana a l'usuari que repassi les dades i les rectifiqui i torni enviar-les.

2.5.2. Cas d'ús registrar dades del nou usuari a la base de dades

2. RegistrarABBDD_Peticions	
Resum de la funcionalitat	El sistema envia les dades a la BBDD Peticions altes.
Actors	Usuari anònim.
Casos d'ús relacionats	3.VerificarPetició.
Precondició	L'usuari ha omplert el formulari i la sintaxis és correcta.
Postcondició	L'usuari està registrat a la BBDD peticions altes.
Procés normal principal	<ol style="list-style-type: none"> 1. Es connecta amb la taula peticionsUsuaris. 2. El sistema omple els camps del nou registre a la BBDD.
Alternatives de procés i excepcions	<ol style="list-style-type: none"> 1a. No s'ha pogut connectar amb la taula de laBBDD. <ol style="list-style-type: none"> 1a1. El sistema mostra un missatge d'error dient a l'usuari . que ho intenti més tard i finalitza el cas d'ús. 1b. Si s'ha pogut connectar amb la taula de la BBDD. <ol style="list-style-type: none"> 1b2. Es passa a la tasca següent. 2 a. S'ha produït alguna excepció. <ol style="list-style-type: none"> 2 a1. Es mostra l'excepció per pantalla i es finalitza el cas d'ús.

2.5.3. Cas d'ús verificar petició (Alta nou usuari).

3. VerificarPetició	
Resum de la funcionalitat	Un administrador verifica que les dades del possible nou usuari tinguin sentit i es tracti d'un usuari real.
Actors	Administrador
Casos d'ús relacionats	8.Autenticar, 4.NotificarIncidència, 6.RegistrarABBDDusuaris
Precondició	L'administrador periòdicament comprova les dades de les peticionsUsuaris per si ni ha algun que es vol registrar , i així disposa de les dades d'un usuari que vol ser donat d'alta.
Postcondició	S'ha decidit si es dona d'alta l'usuari o no.
Procés normal principal	<ol style="list-style-type: none"> 1. L'administrador entra a la pàgina NouUsuari. 2. L'administrador escull la petició amb més antiguitat. 3. L'administrador verifica si es tracta d'una petició real.
Alternatives de procés i excepcions	<ol style="list-style-type: none"> 1a. S'ha produït alguna tipus d'excepció . <ol style="list-style-type: none"> 1a1. El sistema mostra un missatge d'error dient a l'usuari . que ho intenti més tard i finalitza el cas d'ús. 3a. Accepta el nou usuari <ol style="list-style-type: none"> 3a1. L'administrador engega el cas d'ús 6.RegistrarABBDDusuaris 3b. No accepta el nou usuari <ol style="list-style-type: none"> 3b1. L'administrador engega el cas d'ús 4.NotificarIncidència.

2.5.4. Cas d'ús notificar incidència petició usuari (Al possible nou usuari).

4. NotificarIncidència	
Resum de la funcionalitat	Es notifica al possible nou usuari mitjançant mail que la seva petició no ha estat acceptada i les raons, donant-li l'opció de replicar.
Actors	Administrador
Casos d'ús relacionats	3. VerificarPetició, 5.ModificarBBDDpeticions.
Precondició	L'administrador ha decidit no donar d'alta l'usuari.
Postcondició	L'usuari anònim està avisat de que no ha estat donat d'alta.
Procés normal principal	<ol style="list-style-type: none"> 1. L'administrador copia el mail del possible usuari. 2. L'administrador envia un correu electrònic a l'usuari informant-lo de que no ha estat donat d'alta i les raons. 3. El sistema engega el cas d'ús 5.ModificarBBDDpeticions.

2.5.5. Cas d'ús modificar BBDD peticions (peticionsUsuaris)

5.ModificarBBDDpeticions	
Resum de la funcionalitat	El sistema elimina el registre de la taula PeticióUsuaris de la petició que no ha estat acceptada.
Actors	Administrador
Casos d'ús relacionats	4. NotificarIncidència
Precondició	Hi ha entrades a la BBDD peticionsUsuaris un registre de petició que no s'ha acceptat
Postcondició	La BBDD peticionsUsuaris esta actualitzada i sense usuaris no acceptats
Procés normal principal	<ol style="list-style-type: none"> 1. L'administrador entra a la pàgina NouUsuari. 2. L'administrador escull la petició no acceptada. 3. L'administrador escull l'opció eliminar registre. 4. El sistema busca el registre i el borra.
Alternatives de procés i excepcions	<p>2a. S'ha produït alguna excepció.</p> <p>2a1. El sistema mostra un missatge d'error a l'administrador i finalitza el cas d'ús.</p>

2.5.6. Cas d'ús registrar alta usuari (UsuarisRegistrats)

6.RegistrarABBDDusuaris	
Resum de la funcionalitat	El sistema entra les dades del nou usuari a la BBDD UsuarisRegistrats.
Actors	Administrador
Casos d'ús relacionats	4. VerificarPetició, 8. NotificarAltaUsuari.
Precondició	L'administrador ha decidit donar d'alta l'usuari i esta a la pàgina nou usuari
Postcondició	L'Usuari ja esta oficialment registrat.
Procés normal principal	<ol style="list-style-type: none"> 1. L'administrador escull la opció alta nou usuari . 2. El sistema busca a la BBDD la taula UsuarisRegistrats i l'actualitza creant una nova entrada amb les dades del nou usuari. 3. El sistema engega el cas d'ús 7. NotilifarAltaUsuari.
Alternatives de procés i excepcions	<p>2a. S'ha produït alguna excepció.</p> <p>2a1. Es mostra l'excepció per pantalla i es finalitza el cas d'ús.</p>

2.5.7. Cas d'ús notificar alta usuari

7. Notificar Alta Usuari	
Resum de la funcionalitat	L'administrador notifica al nou usuari que ja està registrat.
Actors	Administrador
Casos d'ús relacionats	6.RegistrarABBDDusuaris
Precondició	L'Usuari ja esta oficialment registrat.
Postcondició	El nou usuari està avisat que ja ha estat donat d'alta
Procés normal principal	1. L'administrador envia un correu electrònic al nou usuari dient-li que la seva conta està activada i pot utilitzar els tots els serveis de la protectora.

2.5.8. Cas d'ús autenticar (Usuari o Administrador)

8. Autenticar	
Resum de la funcionalitat	Un usuari o administrador accedeix al sistema i aquest comprova que el nom usuari i la contrasenya són correctes.
Actors	Administrador o Usuari registrat
Casos d'ús relacionats	3. VerificarPetició, 9.MostrarInformació, 12.NotificarAnimalLlest, 14.PublicarAnunciPerdua, 15.AltaAnimal, 16.EliminarAnimal, 17.ModificarAnimal 18.PublicarCasUrgent, 19.EliminarUsuari, 20.EntrarUsuari 21.ModificarUsuari.
Precondició	L'administrador o usuari volen entrar al sistema.
Postcondició	L'administrador o usuari ha accedit al sistema.
Procés normal principal	1. Usuari o Administrador escull l'opció "Accedir al sistema" 2. El sistema demana el nom d'usuari i contrasenya. 3. L'usuari o l'administrador introdueixen les dades. 4. El sistema obre la taula "usuarisRegistrats" 5. El sistema comprova les dades introduïdes amb la BBDD. 6. El sistema dona l'OK i Usuari o Administrador reben la benvinguda i accedeixen al sistema disposant dels privilegis que l'hi pertoquen.
Alternatives de procés i excepcions	4a. No s'ha pogut connectar amb la taula de la BBDD. 4a1. El sistema mostra un missatge d'error a l'administrador o a l'usuari i finalitza el cas d'ús. 4b. Si s'ha pogut connectar amb la taula de la BBDD. 4b2. Es passa a la tasca següent. 5a. El nom d'usuari o la contrasenya no són correctes. 5a1. El sistema mostra el corresponent missatge d'error i finalitza el cas d'ús. 5b. S'ha produït alguna excepció. 5b1. Es mostra l'excepció per pantalla i es cancel·la la tasca

2.5.9. Cas d'ús mostrar informació

9. Mostrar Informació	
Resum de la funcionalitat	El sistema mostra la informació resumida de cada animal segons tipus, grandària i sexe.
Actors	Administrador , Usuari registrat o Usuari anònim
Casos d'ús relacionats	8. Autenticar, 10. Obtenir Fitxa.
Precondició	Usuari identificat com a usuari registrat o administrador o bé és un usuari anònim.
Postcondició	La informació resumida dels animals ha estat mostrada a l'usuari registrat o administrador o bé anònim.
Procés normal principal	<ol style="list-style-type: none"> 1. Usuari (normal o anònim) o administrador accedeixen a la pantalla "Animals". 2. Usuari (normal o anònim) o administrador escull opcions filtratge: tipus, grandària i sexe. 3. El sistema busca a la taula Animals i busca coincidències seleccionades amb les opcions del filtre. 4. El sistema mostra a l'usuari o administrador els resultats trobats de manera resumida i amb l'opció d'accedir a les fitxes on hi ha molts més detalls.
Alternatives de procés i excepcions	<ol style="list-style-type: none"> 3a. S'ha produït alguna excepció. <ol style="list-style-type: none"> 3a1. El sistema mostra un missatge d'error a l'administrador o a l'usuari i finalitza el cas d'ús. 4a. Si usuari o administrador va doble clic sobre un animal. <ol style="list-style-type: none"> 4a1. El sistema engega el cas d'ús 10. Obtenir Fitxa.

2.5.10. Cas d'ús obtenir fitxa

10.ObtenirFitxa	
Resum de la funcionalitat	El sistema mostra la fitxa d'un animal que s'ha escollit anteriorment mostrant altres detalls: fotografies, raça, color, pes, comportament, etc.
Actors	Administrador , Usuari registrat o Usuari anònim
Casos d'ús relacionats	9.MostrarInformació, 11.AdoptarOnline, 11.PublicarFacebook, 14.PublicarAnunciPèrdua.
Precondició	L'administrador o usuari han fet clic sobre un animal del gridview.
Postcondició	La informació detallada dels animals ha estat mostrada a l'usuari o administrador.
Procés normal principal	<ol style="list-style-type: none"> 1.El sistema obra la pantalla "Mostrar Animals". 2.El sistema busca a la taula Animals l'animal seleccionat amb la dada indentificadorAnimal 3. El sistema mostra a l'usuari o administrador totes les dades que fan referència a l'animal seleccionat. 4. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	2a. S'ha produït alguna excepció. 2a1. El sistema mostra un missatge d'error a l'administrador o a l'usuari i finalitza el cas d'ús.

2.5.11. Cas d'ús publicar al Facebook

11.PublicarFacebook	
Resum de la funcionalitat	Un usuari registrat o un administrador poden publicar una notícia al facebook (en el seu perfil) sobre un animal concret, mostrant la fotografia principal, enllaç a la fitxa del portal, un comentari del usuari, etc.
Actors	Administrador o Usuari registrat
Casos d'ús relacionats	10.ObtenirFitxa
Precondició	L'administrador o usuari tenen la fitxa d'un animal oberta.
Postcondició	La notícia esta publicada al perfil de l'usuari o l'administrador.
Procés normal principal	<ol style="list-style-type: none"> 1. Usuari (també pot ser anònim) o Administrador escull l'opció "Publicar al facebook" 2. El sistema es connecta amb el portal facebook i espera que l'usuari o l'administrador s'autentiqui. 3. El sistema publica la notícia al portal facebook. 4. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	2a. L'usuari o administrador no s'ha pogut autenticar. 2a1. El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.

2.5.12. Cas d'ús adoptar online

12.AdoptarOnline	
Resum de la funcionalitat	El sistema ofereix la possibilitat d'adoptar online.
Actors	Administrador o Usuari registrat
Casos d'ús relacionats	10.ObtenirFitxa
Precondició	L'administrador o usuari tenen la fitxa d'un animal oberta.
Postcondició	L'administrador o usuari estant a l'espera de la notificació de que l'animal ja està llest per ser recollit.
Procés normal principal	<ol style="list-style-type: none"> 1. Usuari o Administrador escull l'opció "Adoptar online" 2. El sistema modifica el camp adoptarOnline de la taula Animals i omple el registre de la taula Adopcions. 3. El sistema notifica al grup administradors que hi ha una adopció online en procés. (Aquesta part no s'ha fet encara) 4. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	2a. S'ha produït alguna excepció. 2a1. El sistema mostra un missatge d'error a l'administrador o a l'usuari i finalitza el cas d'ús.

2.5.13. Cas d'ús notificar animal Llest

13.NotificarAnimalLlest	
Resum de la funcionalitat	L'usuari o administrador rebran una notificació quan l'animal estigui llest per ser recollit.
Actors	Administrador
Casos d'ús relacionats	8. Autenticar, 13.RegistrarAdopció
Precondició	L'administrador ja ha rebut la notificació de que hi ha una adopció online en procés.(De moment revisa periòdicament la taula adopcions i està previst en un futur automatitzar-ho).
Postcondició	L'usuari o l'administrador esta avisat que pot passar a recollir la seva mascota.
Procés normal principal	<ol style="list-style-type: none"> 1. L'administrador accedeix a l'apartat adopcions online. 2. Valida l'adopció entrant una data en el camp DataAdopció. 3. El sistema engega el cas d'ús 13.RegistrarAdopció. 4. L'administrador envia un mail a l'usuari o administrador que ha fet l'adopció online que ja pot passar a recollir la mascota.
Alternatives de procés i excepcions	1a. S'ha produït alguna excepció. 1a1. El sistema mostra un missatge d'error a l'administrador o a l'usuari i finalitza el cas d'ús.

2.5.14. Cas d'ús registrar adopció

14.RegistrarAdopció	
Resum de la funcionalitat	El sistema actualitza les taula Adopcions de la BBDD entrant les dades de la nova adopció.
Actors	Administrador
Casos d'ús relacionats	13.NotificarAnimalLlest
Precondició	L'administrador ja ha fet la gestió de l'adopció online.
Postcondició	La taula Adopcions està actualitzada
Procés normal principal	<ol style="list-style-type: none"> 1. El sistema obre (connecta) amb la taula Adopcions. 2. El sistema crea una nova entrada amb les dades de la nova adopció. 3. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	1a. S'ha produït alguna excepció. 1a1. El sistema mostra un missatge d'error a l'administrador o a l'usuari i finalitza el cas d'ús.

2.5.15. Cas d'ús publicar anunci pèrdua (mascota)

15.PublicarAnunciPerdua	
Resum de la funcionalitat	L'usuari o administrador que hagi perdut la seva mascota publicar un anunci en el portal incorporant una imatge, ubicació (GoogleMaps),etc.
Actors	Administrador o Usuari registrat
Casos d'ús relacionats	8.Autenticar
Precondició	Usuari identificat com a usuari registrat o administrador.
Postcondició	L'anunci de la pèrdua està publicat en el portal.
Procés normal principal	<ol style="list-style-type: none"> 1. Usuari o administrador obra la pàgina pèrdues. 2. Usuari o administrador escull l'opció entrar pèrdua 3. Usuari o administrador escull l'opció GoogleMaps 4. El sistema obra la pàgina "Escollir Ubicació Pèrdua" 5. El sistema es connecta amb el GoogleMaps. 6. Usuari o administrador selecciona la ubicació de la pèrdua. 7. Usuari o administrador confirma dades. 8. El sistema torna a la pàgina pèrdues i omple el camp ubicació. 9. Usuari o administrador omplen les altres dades i pugen la fotografia i escull l'opció guardar pèrdua. 10. El sistema connecta amb la base de dades (taula pèrdues) i crea una nova entrada amb les dades de la nova pèrdua.
Alternatives de procés i excepcions	1a, 5a, 9a, 10a S'ha produït alguna excepció. 1a1., 5a1,9a1, 10a1, El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.

2.5.16. Cas d'ús alta animal

16.AltaAnimal	
Resum de la funcionalitat	L'Administrador crea un registre d'un nou animal.
Actors	Administrador
Casos d'ús relacionats	8.Autenticar
Precondició	Administrador identificat com a administrador.
Postcondició	Hi ha un nou animal a la base de dades
Procés normal principal	<ol style="list-style-type: none"> 1. L'Administrador escull la opció (a la pantalla principal) gestió animals. 2. El sistema obre la pantalla amb el formulari "Gestió Animals". 3. L'administrador omple els camps i puja les fotografies. 4. L' administrador confirma dades. 5. El sistema envia les dades a la base de dades i mostra el missatge d'èxit. 6. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	<p>3a,5a S'ha produït alguna excepció</p> <p>3a1,5a1 El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.</p>

2.5.17. Cas d'ús eliminar animal

17.EliminarAnimal	
Resum de la funcionalitat	L'Administrador elimina un registre d'un animal.
Actors	Administrador
Casos d'ús relacionats	8.Autenticar
Precondició	Administrador identificat com a administrador.
Postcondició	Hi ha un animal menys a la base de dades
Procés normal principal	<ol style="list-style-type: none"> 1. L'Administrador escull la opció (a la pantalla principal) gestió animals. 2. El sistema obre la pantalla amb el formulari "Gestió Animals". 3. L'administrador tria el mode edició. 4. L' administrador tria el registre que vol eliminar fa clic sobre la icona eliminar 5. El sistema connecta amb la BBDD i la taula animals i borra el registre seleccionat. 6. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	<p>4 a, 5a S'ha produït alguna excepció</p> <p>4a1, 5a1 El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.</p>

2.5.18. Cas d'ús modificar animal

18.ModificarAnimal	
Resum de la funcionalitat	L'Administrador modifica les dades d'un animal.
Actors	Administrador
Casos d'ús relacionats	8.Autenticar
Precondició	Administrador identificat com a administrador.
Postcondició	Hi ha un animal amb alguna dada actualitzada a la base de dades
Procés normal principal	<ol style="list-style-type: none"> 1. L'Administrador escull la opció (a la pantalla principal) gestió animals. 2. El sistema obre la pantalla amb el formulari "Gestió Animals". 3. L'administrador tria el mode edició. 4. L' administrador tria el registre que vol modificar i modificar les dades que vol i fa clic sobre la icona actualitzar dades. 5. El sistema connecta amb la BBDD i la taula animals i actualitza el registre seleccionat. 6. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	<p>4 a, 5a S'ha produït alguna excepció</p> <p>4a1, 5a1 El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.</p>

2.5.19. Cas d'ús seleccionar i publicar cas urgent.

19.PublicarCasUrgent	
Resum de la funcionalitat	Un administrador pot seleccionar un cas urgent i fer que apareguin a la pàgina principal i a l'apartat corresponent (tres com a màxim).
Actors	Administrador
Casos d'ús relacionats	8.Autenticar
Precondició	Administrador identificat com a administrador.
Postcondició	A la pàgina principal i al apartat casos urgents hi ha el nou cas.
Procés normal principal	<ol style="list-style-type: none"> 1. L'Administrador escull la opció (a la pantalla principal) gestió animals. 2. El sistema obre la pantalla amb el formulari "Gestió Animals". 3. L'administrador tria el mode edició. 4. L' administrador tria el registre que vol com a cas urgent i fa clic sobre la icona actualitzar dades. 5. El sistema obra la pantalla entrar cas urgent. 6. L'administrador tria quin es el cas urgent actual que vol sacrificar (en el cas que ja n'hi hagin 3) i sinó tria el primer registre lliure i fa clic sobre la icona registrar. 7. El sistema connecta amb la BBDD i la taula CasosUrgents i actualitza el registre seleccionat. 8. L'administrador torna a la pantalla gestió animals. 9. El sistema torna a la pantalla gestió animals en mode inserció i mostra el primer registre. 10. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	<p>4 a, 5a , 7a , 9a S'ha produït alguna excepció</p> <p>4a1, 5a1, 7a1, 9a1 El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.</p> <p>6a L'administrador vol eliminar un cas urgent.</p> <p>6a1. Tria el cas que vol eliminar i fa clic sobre la icona . eliminar.</p>

2.5.20. Cas d'ús eliminar usuari

20.EliminarUsuari	
Resum de la funcionalitat	L'Administrador elimina un registre d'un usuari.
Actors	Administrador
Casos d'ús relacionats	8.Autenticar
Precondició	Administrador identificat com a administrador.
Postcondició	Hi ha un usuari menys a la base de dades
Procés normal principal	<ol style="list-style-type: none"> 1. L'Administrador escull la opció (a la pantalla principal) gestió usuaris. 2. El sistema obre la pantalla amb el formulari "Gestió Usuaris". 3. L'administrador tria el mode edició. 4. L' administrador tria el registre que vol eliminar fa clic sobre la icona eliminar 5. El sistema connecta amb la BBDD i la taula usuarisRegistrats i borra el registre seleccionat. 6. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	<p>4 a, 5a S'ha produït alguna excepció</p> <p>4a1, 5a1 El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.</p>

2.5.21. Cas d'ús entrar usuari (*És la manera manual de registrar de registrar un nou usuari, la més usual per l'administrador és el cas d'ús 6 registrar alta usuari*),

21.EntrarUsuari	
Resum de la funcionalitat	L'Administrador crea un registre d'un nou usuari.
Actors	Administrador
Casos d'ús relacionats	8.Autenticar
Precondició	Administrador identificat com a administrador.
Postcondició	Hi ha un nou animal a la base de dades
Procés normal principal	<ol style="list-style-type: none"> 1. L'Administrador escull la opció (a la pantalla principal) gestió usuaris. 2. El sistema obre la pantalla amb el formulari "Gestió Usuaris". 3. L'administrador omple els camps amb les dades de l'usuari 4. L' administrador confirma dades. 5. El sistema envia les dades a la base de dades i mostra el missatge d'èxit. 6. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	<p>5a S'ha produït alguna excepció</p> <p>5a1 El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.</p>

2.5.22. Cas d'ús modificar usuari

22.ModificarUsuari	
Resum de la funcionalitat	L'Administrador modifica les dades d'un usuari.
Actors	Administrador
Casos d'ús relacionats	8.Autenticar
Precondició	Administrador identificat com a administrador.
Postcondició	Hi ha un animal amb alguna dada actualitzada a la base de dades
Procés normal principal	<ol style="list-style-type: none"> 1. L'Administrador escull la opció (a la pantalla principal) gestió usuaris. 2. El sistema obre la pantalla amb el formulari "Gestió Usuaris". 3. L'administrador tria el mode edició. 4. L' administrador tria el registre que vol modificar i modificar les dades que vol i fa clic sobre la icona actualitzar dades. 5. El sistema connecta amb la BBDD i la taula usuarisRegistrats i actualitza el registre seleccionat. 6. El sistema finalitza el cas d'ús.
Alternatives de procés i excepcions	<p>4 a, 5a S'ha produït alguna excepció</p> <p>4a1, 5a1 El sistema mostra el missatge d'error corresponent i finalitza el cas d'ús.</p>

2.6. Altres diagrames (Diagrames de seqüència)

Es pot entrar més a fons amb altres tipus de diagrames com els diagrames de seqüència que t'ajuden a obtenir d'una manera prèvia algunes sentències de codi que hauràs d'utilitzar i quines funcions necessitaràs com també els missatges que s'enviaran entre usuari i sistema en cada cas d'ús. Podríem dedicar molt temps a fer tots tipus de diagrames però precisament tenim un problema de temps, i amb els diagrames de cas d'ús ja tenim un molt bon punt de partida per començar a implementar l'aplicació. L'únic cas d'ús que he pensat que necessitava un previ diagrama de seqüència era el de Login(Autenticar) ja que és un dels més complexes.

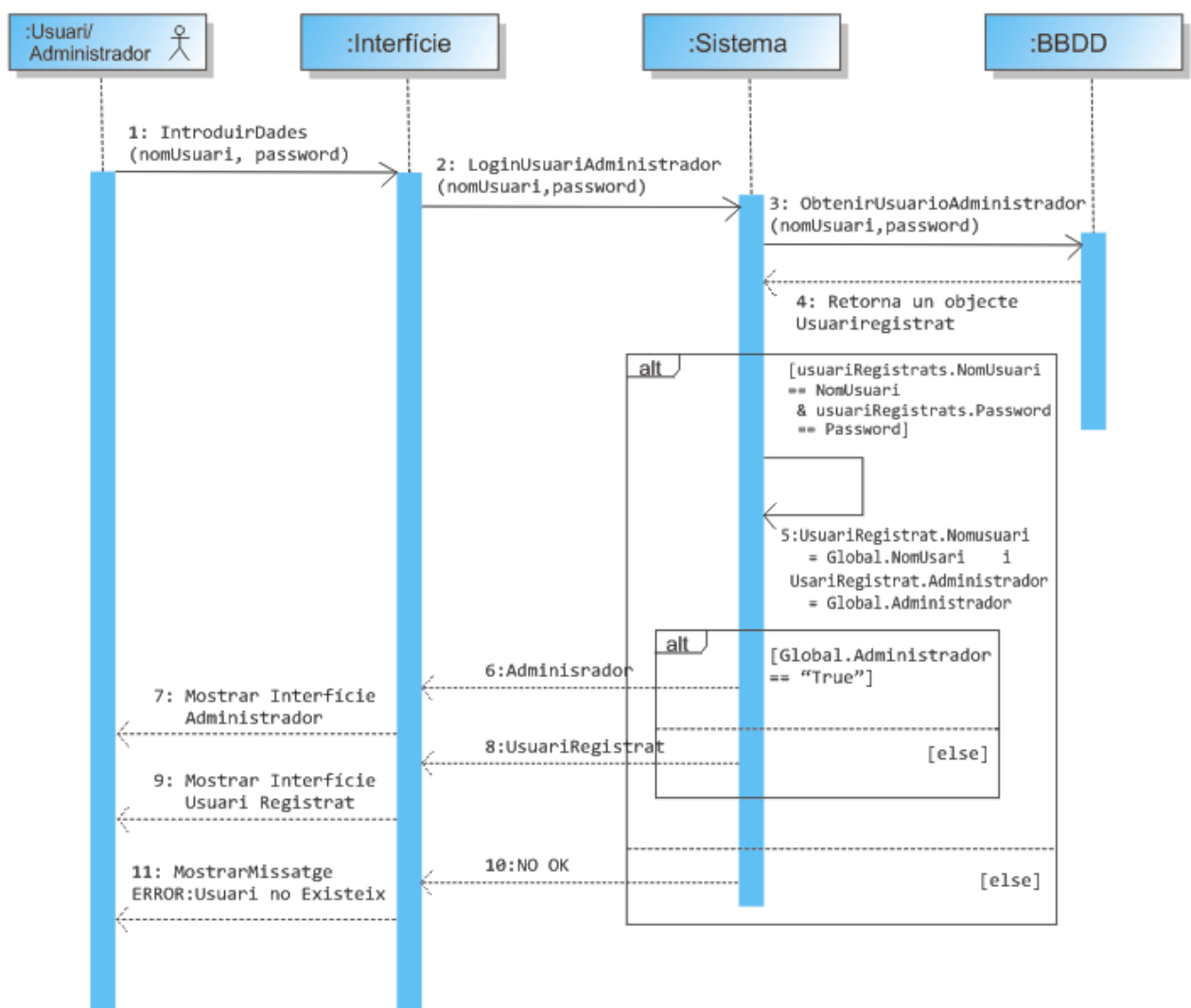


Figura 11. Diagrama de seqüència Login

3. Disseny

Ja arribats aquest punt, tenim la informació suficient per passar a la part de disseny de la nostra aplicació: arquitectura de l'aplicació, model conceptual, diagrama de classes d'entitat, la interfície de l'usuari, la base de dades i alguns altres components més.

3.1. Arquitectura de l'aplicació

L'arquitectura d'aquesta aplicació segueix el model de programació per capes (presentació, negoci i dades) amb l'objectiu de tenir aquetes parts aïllades entre sí per si s'ha de fer algun canvi important que afecti a una part en concret, no haguem de perdre temps revisant entre codi barrejat. A més si en un futur l'aplicació s'ha d'actualitzar amb noves funcionalitats, es podrà repartir la feina en diferents grups de treball permetent-los treballar per separat.

Per exemple, si algun dia es vol canviar el servidor de base de dades per una altre marca comercial o bé per una actualització de versió, ens serà molt més ràpid i còmode dur a terme el canvi tenint la part d'accés a dades aïllada de tota la resta de l'aplicació. Com també si en algun moment hem d'actualitzar certes llibreries de components com Ajax control toolkit, GoogleMaps, facebook, etc,etc.. només ens afectarà a la capa de presentació i les altres dues capes no s'adonaran del canvi.

Veiem més a fons les tres capes:

- ✚ **Capa de presentació:** És la part que veu l'usuari, la interfície gràfica que l'hi transmet la informació i també captura la que aquest en un moment donat pot aportar-li. També controla qüestions com les validacions i filtratges de les dades entrades per l'usuari i captura les diferents excepcions que es poden produir durant l'execució de tota l'aplicació en general (totes les capes) i després les mostra a l'usuari. Aquesta capa només comunica amb la capa de negoci. Està formada per components i tecnologies com, ASP.NET, AJAX, HTML, fulla d'estils CSS i algunes més. A més conté una carpeta on s'hi guarden totes les imatges que surten a la interfície, per els polsadors, el banner etc...
- ✚ **Capa de negoci:** És la capa que rep les peticions del l'usuari i un cop ha realitzat les tasques oportunes envia les respostes en aquest. Aquí és on s'estableixen i es gestionen totes de regles. Aquesta capa comunica amb la capa de presentació, d'aquesta rep les sol·licituds i l'hi presenta els resultats, i també comunica amb la capa de dades per sol·licitar al gestor de base de dades registrar, eliminar, actualitzar o recuperar dades. En aquesta capa també s'hi estableix el disseny del diagrama de classes d'entitat

- Capa de dades:** És on es té contacte directa amb les dades, és la que gestiona el traspàs d'objectes a dades i a l'inversa (Linq to SQL) .En aquesta capa resideixen les dades i aquesta al mateix temps és l'encarregada s'accedir-hi.Pot tenir un o més gestors de bases de dades (en el nostra cas només un).Només comunica amb la capa de negoci que és la que l'hi envia sol·licituds d'emmagatzematge i recuperació de dades.

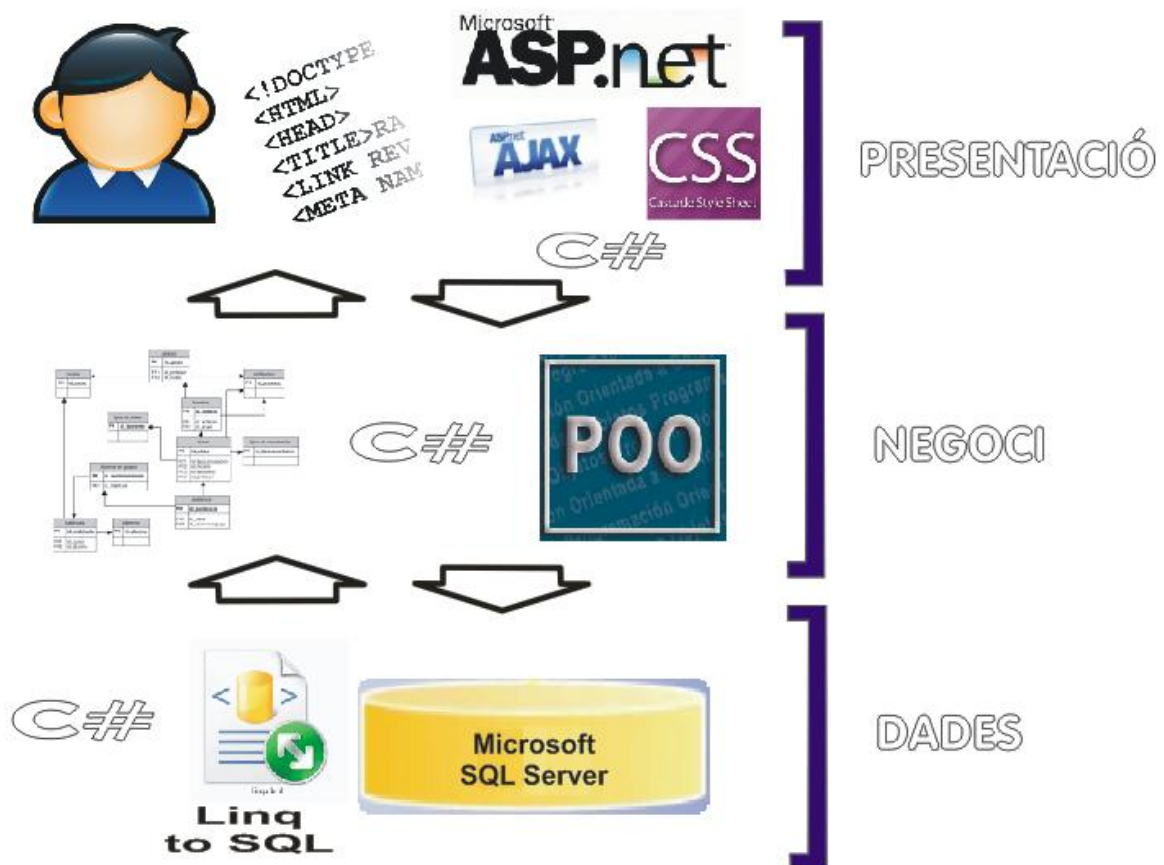


Figura 12. Model tres capes

De la manera que s'han implementat aquestes tres capes en el nostre projecte, ha consistit en que dins d'una solució del Visual Studio 2010 hem creat un projecte per cada capa. Per tots tres projectes s'ha triat el llenguatge Visual C# 4.0.

El projecte de presentació és un projecte Web amb ASP.NET 4, el de lògica és un projecte de llibreria de classes i el projecte de dades és un projecte de llibreria de classes amb un element de classes Linq to SQL afegit posteriorment.

Aquesta projectes estaran lligats entre sí gracies en que a cada projecte s'han declarat namespace "nom del projecte (capa)" a totes les classes i des de la carpeta reference a cada projecte s'agrega la referència del projecte que es vol tenir lligat. A més a principi de cada classe amb la sentència using "namespace" podem cridar totes les classes i mètodes d'aquests espais de noms dels altres projectes.

Això farà que al 'hora de publicar el projecte en un servidor Web el projecte presentació crearà un dll per cadascun dels altres dos projectes (capes) i fer córrer els tres projectes com si es tractés d'un de sol.

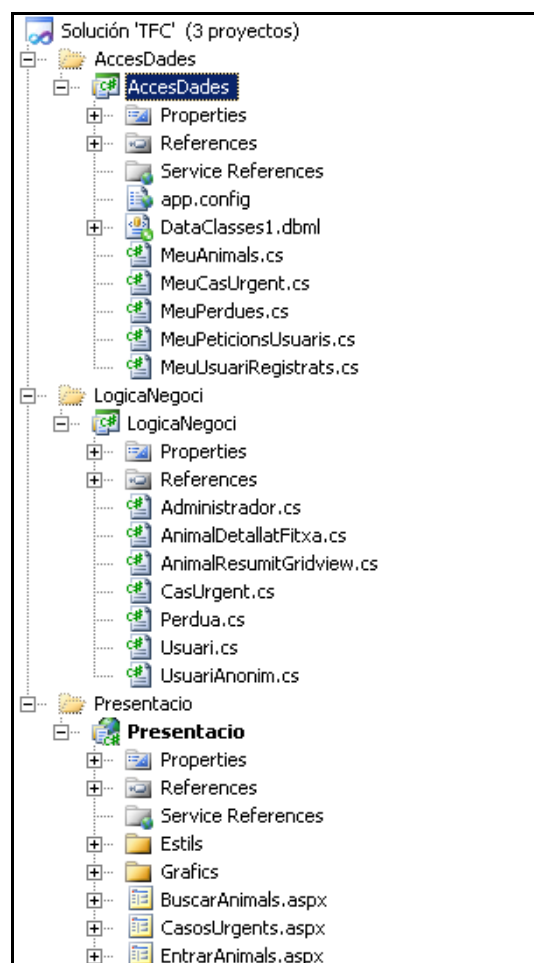


Figura 13. Solució tres projectes = tres capes

3.2. Disseny de la interfície de l'usuari

La interfície d'aquesta aplicació s'ha fet mitjançant pàgines Web i el principal l'objectiu des d'un inici ha estat el d'aconseguir una apariència i un comportament el màxim d'interactiu, àgil i agradable possible, com si d'una interfícies típica de Windows es tractés. Per aquesta raó la interfície s'ha basat en ASP.NET i AJAX.

3.2.1. ASP.NET i AJAX

ASP.NET és l'eina ideal per a crear aplicacions Web, així i tot aquesta tecnologia té el handicap de que cada vegada que es fa un canvi a la pàgina aquesta es càrrega tota de nou, i això precisament no ajuda a la interactivitat i a la velocitat del portal. És per aquesta raó que s'ha afegit en el projecte la tècnica AJAX que és l'acrònim de Asynchronous JavaScript And XML i és una tècnica de desenvolupament que soluciona aquesta limitació fent que parts de l'aplicació s'executin en el navegador de l'usuari mantenint una comunicació asíncrona amb el servidor en segon pla. AJAX esta basat en estàndards coneguts com JavaScript, XML, HTML y CSS (fulla d'estils en cascada).

L'element central de AJAX és l'objecte JavaScript XMLHttpRequest mitjançant de qual es pot interactuar amb el servidor sense que aquest hagi de tornar a carregar tota la pàgina cada vegada. A més tots els navegadors actuals suporten AJAX.

Quan crees un projecte Web ASP.NET amb el Visual Studio 2010 en el quadre d'eines ja hi tens les extensions bàsiques de AJAX : ScriptManager, ScriptManagerProxy, Timer, UpdatePanel i Updateprogress. Que realment es tracta de ASP.NET AJAX que és la llibreria que Microsoft ofereix per utilitzar AJAX dins la plataforma ASP.NET.

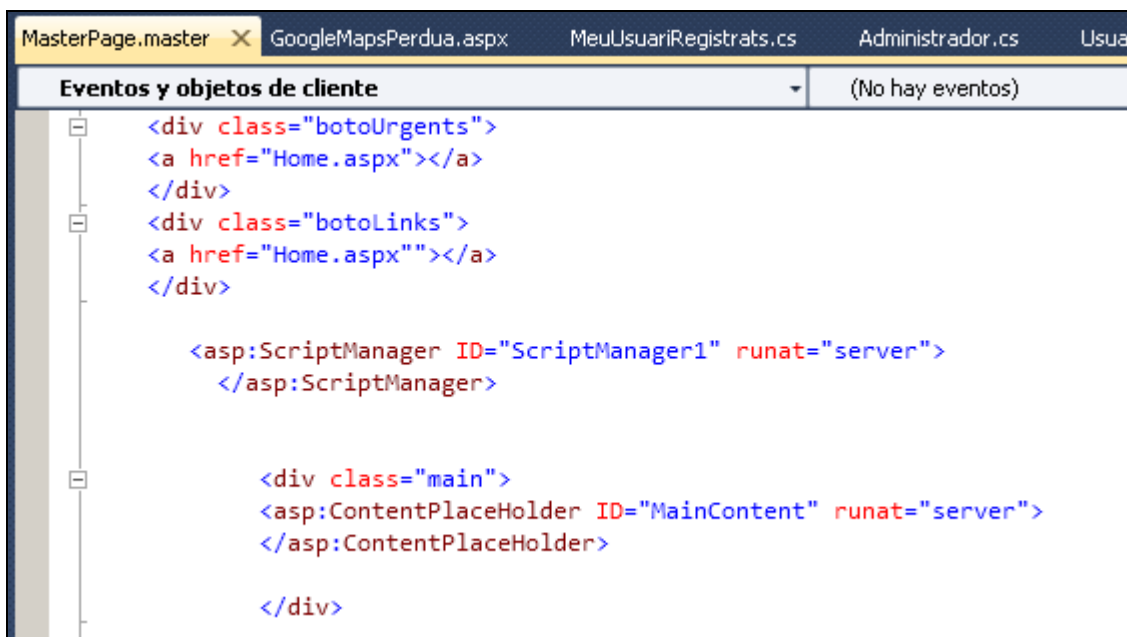
Està clar doncs que utilitzant aquesta tècnica l'usuari gaudirà d'una millor experiència en la visita del nostre portal.

Totes les pàgines .aspx del nostre projecte de presentació utilitzen AJAX, i els controls utilitzats i el perquè es detallen seguidament:

- **ScriptManager:** És un el control no visual perquè una pàgina web pugui utilitzar AJAX. Genera tot el codi JavaScript necessari per la crides asíncrones des del navegador al servidor web. Podríem dir que és el cervell de ASP.NET AJAX. Cada pàgina que utilitza ASP.NET AJAX requereix una instància de la classe ScriptManager. Però si en el nostre projecte hi tenim una MasterPage (pàgina principal) hi ha un altre control que necessitarem per organitzar-ho millor, és l' ScriptManagerProxy.

- **ScriptManagerProxy**: Si en un projecte hi tens una pàgina principal, en aquesta hi col·loques un ScriptManager i dins els controls que vols utilitzar a tot el projecte però si el que vols és utilitzar diferents controls a cada pàgina (el nostre cas), la solució és utilitzar ScriptManager a la pàgina principal i llavors a cada pàgina de contingut agregar-hi un ScriptManagerProxy i dins els controls específics de cada pàgina, i d'aquesta manera aconseguim que només s'executin els controls AJAX de la pàgina que l'usuari estigui utilitzant.
- **UpdatePanel**: Aquest control és el més bàsic i el més utilitzat en qualsevol aplicació web. És un contenidor de controls que intercepta les crides convencionals al servidor dels controls que té i substitueix aquestes crides convencionals per crides de tipus AJAX i retorna les ordres del servidor utilitzant el mateix mecanisme. D'aquesta manera els controls de dins de l'Updatepanel poden cridar al servidor sense que es provoqui un refresc de tota la pantalla. Aquest controls poden ser controls estàndards de ASN.NET

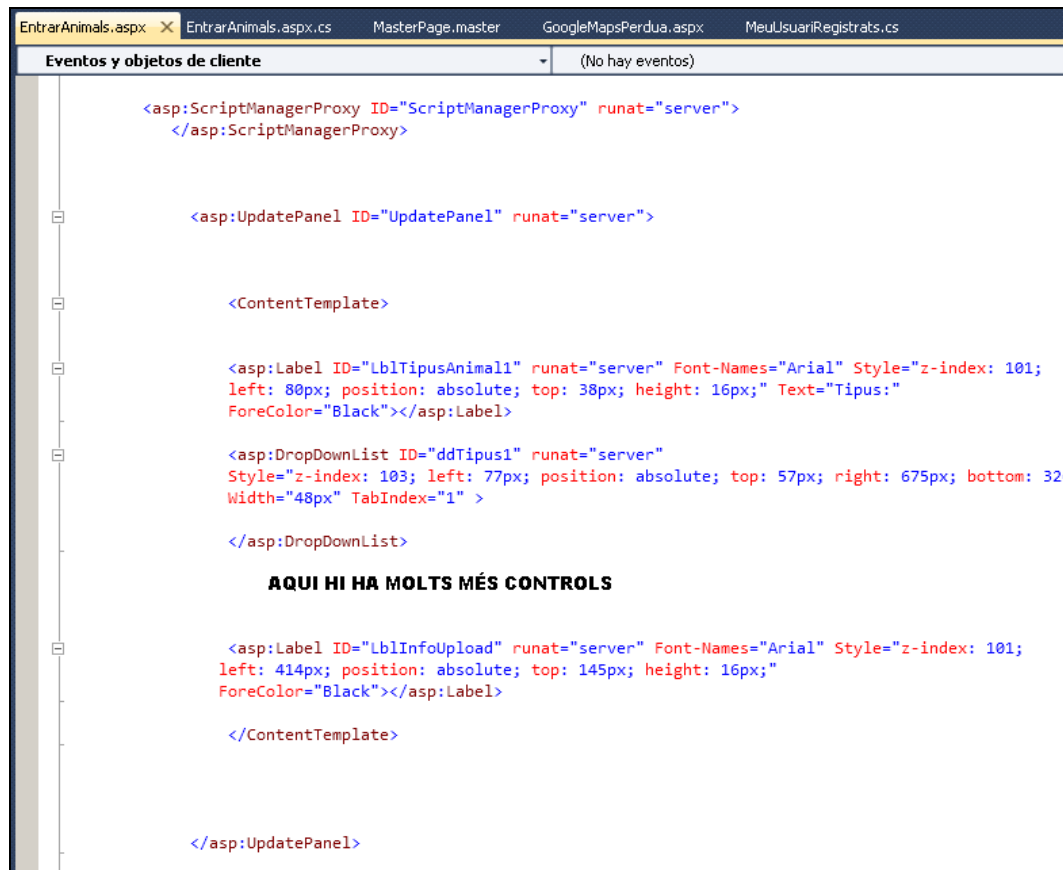
Anem a veure com hem implementat aquests controls dins una pàgina del nostre projecte. Primer a la pàgina principal hi hem agregat un ScriptManager (Figura 14), llavors a una de les pàgines (per exemple la de gestió animals) hi hem agregat ScriptManagerProxy (Figura 15) i a unes línies més avall l'UpdatePanel amb tots els controls com: caixes de text, labels, control image, polsadors, etc... que volem que només es refresquin per exemple quan estem mirant els registres dels animals amb els polsadors d'avançar i retrocedir.



The screenshot shows a Visual Studio IDE window with several tabs: MasterPage.master, GoogleMapsPerdua.aspx, MeuUsuariRegistrats.cs, Administrador.cs, and Usuar. The active window displays the code for a page, showing a tree view on the left and the code on the right. The code includes a ScriptManager control and a ContentPlaceHolder control.

```
Events y objetos de cliente (No hay eventos)
<div class="botoUrgents">
  <a href="Home.aspx"></a>
</div>
<div class="botoLinks">
  <a href="Home.aspx"></a>
</div>
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
<div class="main">
  <asp:ContentPlaceHolder ID="MainContent" runat="server">
</asp:ContentPlaceHolder>
</div>
```

Figura 14. ScriptManager a la pàgina principal

The image shows a screenshot of a Visual Studio IDE window. The title bar at the top lists several files: 'EntrarAnimals.aspx', 'EntrarAnimals.aspx.cs', 'MasterPage.master', 'GoogleMapsPerdua.aspx', and 'MeuUsuariRegistrats.cs'. Below the title bar, there is a tab labeled 'Eventos y objetos de cliente' with a dropdown arrow and the text '(No hay eventos)'. The main area of the window displays ASP.NET code for an UpdatePanel control. The code includes a ScriptManagerProxy, an UpdatePanel, a ContentTemplate, a Label with ID 'LblTipusAnimal1', a DropDownList with ID 'ddTipus1', a bold text string 'AQUI HI HA MOLTS MÉS CONTROLS', and another Label with ID 'LblInfoUpload'. The code is color-coded and includes various attributes like 'runat="server"', 'Font-Names="Arial"', and 'Style="z-index: 101; left: 80px; position: absolute; top: 38px; height: 16px; text="Tipus:"'.

```
<asp:ScriptManagerProxy ID="ScriptManagerProxy" runat="server">
</asp:ScriptManagerProxy>

<asp:UpdatePanel ID="UpdatePanel" runat="server">

    <ContentTemplate>

        <asp:Label ID="LblTipusAnimal1" runat="server" Font-Names="Arial" Style="z-index: 101;
left: 80px; position: absolute; top: 38px; height: 16px; text="Tipus:"
ForeColor="Black"></asp:Label>

        <asp:DropDownList ID="ddTipus1" runat="server"
Style="z-index: 103; left: 77px; position: absolute; top: 57px; right: 675px; bottom: 32
Width="48px" TabIndex="1" >

        </asp:DropDownList>

        AQUI HI HA MOLTS MÉS CONTROLS

        <asp:Label ID="LblInfoUpload" runat="server" Font-Names="Arial" Style="z-index: 101;
left: 414px; position: absolute; top: 145px; height: 16px;"
ForeColor="Black"></asp:Label>

    </ContentTemplate>

</asp:UpdatePanel>
```

Figura 15. ScriptManagerProxy i UpdatePanel a la pàgina entrar animals (exemple)

Finalment a la figura16 de la pàgina següent veiem el resultat, imaginant que només es refresca la sona on hi ha l'update panel.

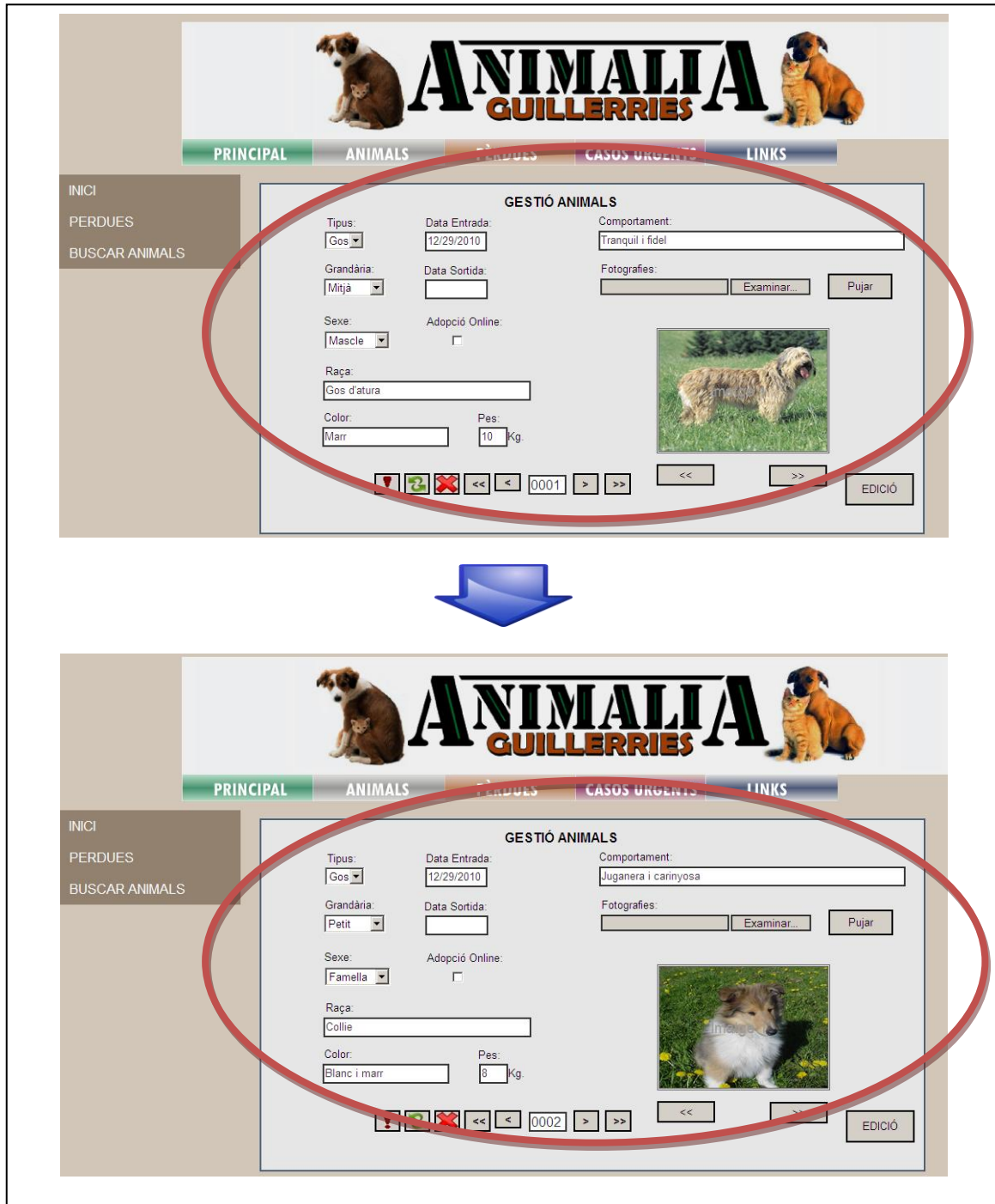
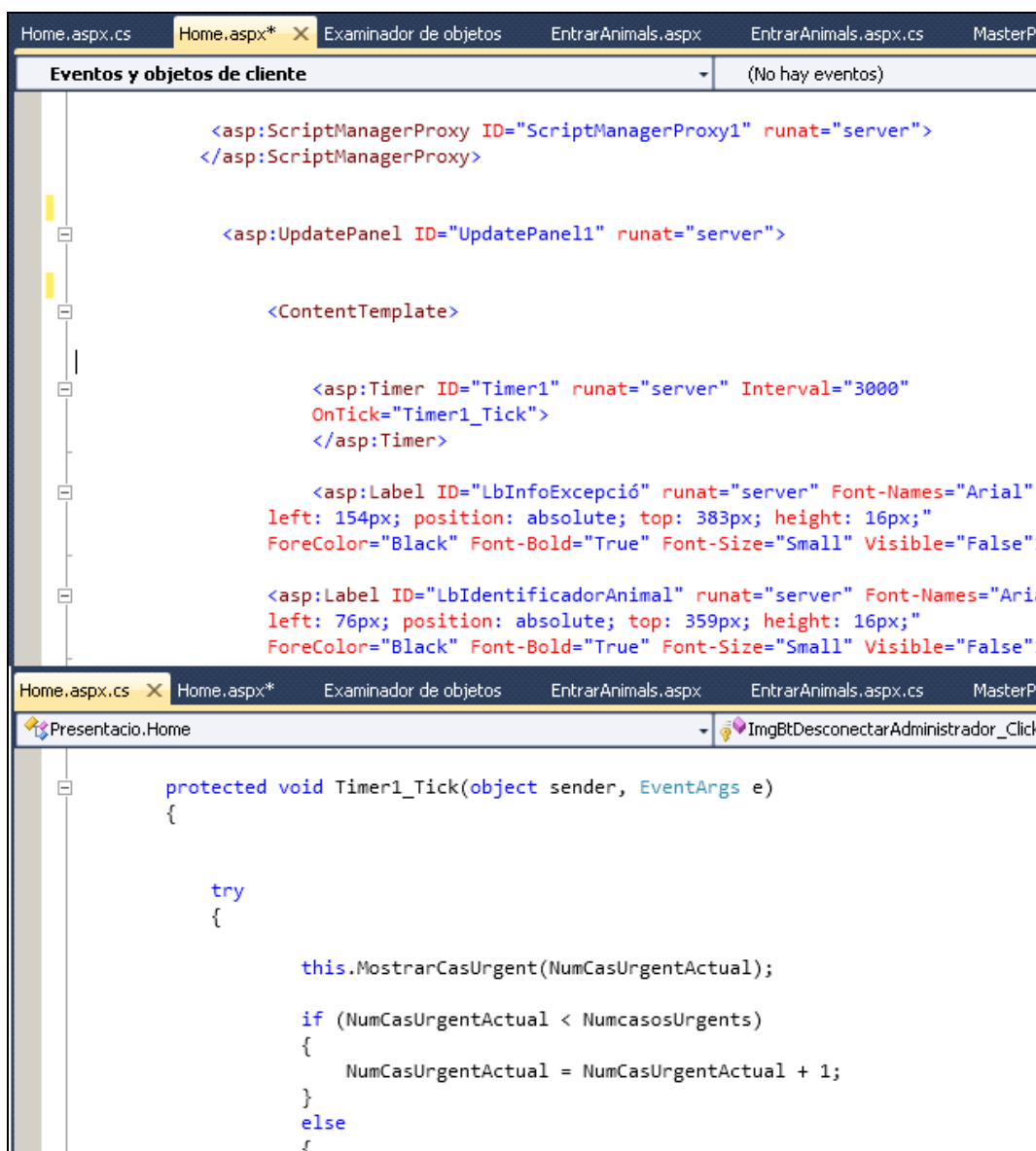


Figura 16. Resultat ScriptManager i UpdatePanel (AJAX)

- **Timer:** Aquest control s'ha utilitzat per cobrir la funcionalitat dels casos urgents que apareixen a la pàgina principal de manera rotativa i refrescant només la regió on hi ha la informació dels animals. Amb aquest control, cada vegada que es supera els temps programat a la propietat "interval" s'executarà el codi que hi ha l'event OnTick del fitxer de codi Visual C# i com que està dins l'UpdatePanel només es refresquen els controls que estan dins aquesta regió.

A la figura 17 de sota podem veure com s'ha programat en el fitxer home.aspx i també l'event OnTick del fitxer [home.aspx.cs](#).



```
Home.aspx.cs Home.aspx* Examinador de objetos EntrarAnimals.aspx EntrarAnimals.aspx.cs MasterP
Eventos y objetos de cliente (No hay eventos)

<asp:ScriptManagerProxy ID="ScriptManagerProxy1" runat="server">
</asp:ScriptManagerProxy>

<asp:UpdatePanel ID="UpdatePanel1" runat="server">

<ContentTemplate>

<asp:Timer ID="Timer1" runat="server" Interval="3000"
OnTick="Timer1_Tick">
</asp:Timer>

<asp:Label ID="LbInfoExcepció" runat="server" Font-Names="Arial"
left: 154px; position: absolute; top: 383px; height: 16px;"
ForeColor="Black" Font-Bold="True" Font-Size="Small" Visible="False">

<asp:Label ID="LbIdentificadorAnimal" runat="server" Font-Names="Ari
left: 76px; position: absolute; top: 359px; height: 16px;"
ForeColor="Black" Font-Bold="True" Font-Size="Small" Visible="False">

Home.aspx.cs Home.aspx* Examinador de objetos EntrarAnimals.aspx EntrarAnimals.aspx.cs MasterP
Presentacio.Home ImgBtDesconectarAdministrador_Click

protected void Timer1_Tick(object sender, EventArgs e)
{

try
{

this.MostrarCasUrgent(NumCasUrgentActual);

if (NumCasUrgentActual < NumcasosUrgents)
{
NumCasUrgentActual = NumCasUrgentActual + 1;
}
else
{
```

Figura 17. Control Timer amb l'event OnTick de AJAX

Per treure més suc de AJAX s'ha agregat en projecte la llibreria AJAX Control Toolkit que ajuda augmentar considerablement les seves possibilitats, ja que és la col·lecció més gran i millor de components de client web disponibles que extenen la funcionalitat dels controls estàndards de ASP.NET. D'aquesta llibreria els controls extensor que hem utilitzat per el nostre projecte han estat:

- **CalendarExtender:** Aquesta extensió mostra a l'usuari un calendari i en pot escollir una data (Figura 18) i en el nostra cas exten una caixa de text normal. En el nostra projecte ho hem programat a les caixes de text de baixa animal i baixa usuari.

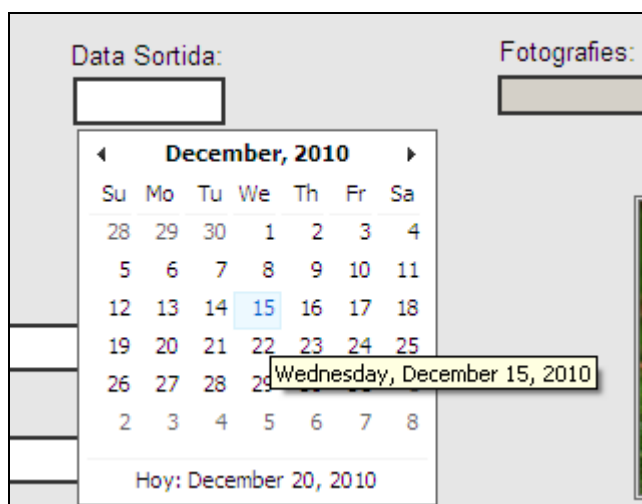


Figura 18. Utilització del CalendarExtender de AJAX Control toolkit

- **PasswordStrenght:** Aquesta altra extensió permet que en una entrada de password informa a l'usuari sobre la robustesa i el límit d'aquest

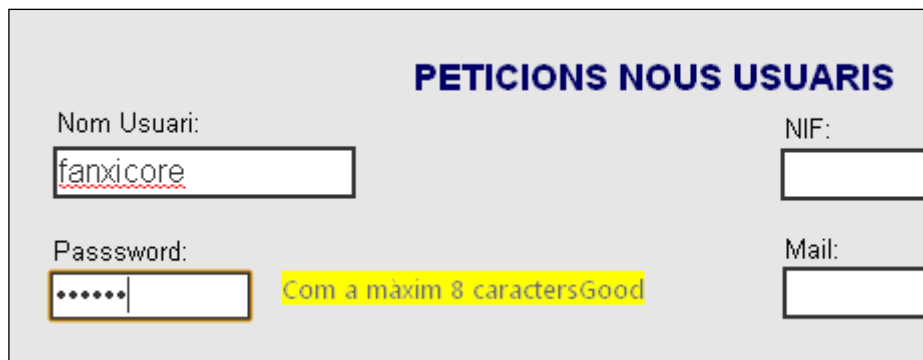
The image shows a web form titled "PETICIONS NOUS USUARIS". It has four input fields: "Nom Usuari:" (containing "fanxicore"), "NIF:", "Password:" (containing "*****"), and "Mail:". A yellow tooltip is visible over the password field, displaying the text "Com a maxím 8 charactersGood".

Figura 19. Utilització del PasswordStrenght de AJAX Control toolkit

- **FilteredTextBox:** També extenent a les caixes de text aquest control m'ha permès controlar des de la mateixa interfície les validacions de caràcters com per exemple que en un camp on només hi vull un nom no s'hi colin caràcters estranys, o bé en un número de telèfon només hi puguin entrar números i moltes situacions més.

3.2.2. Controls i menús d'usuari

Per ajudar a l'usuari a moure's amb agilitat per el portal s'han dissenyat una sèrie de menús i controls molt amigables.

- **Menús:** S'han dissenyat dos menús per moure's per el portal un de lateral i un de superior, i els dos s'han programat a la fulla d'estils CSS que és l'acrònim de CascadingStyle Sheets (fulles d'estil en cascada) i consisteix en la creació d'estils propis dins de les pàgines web o bé en un arxiu separat (el nostra cas) que serà invocat per el sistema al moment de presentar els documents.Gràcies a l'efecte hover de CSS hem fet que quan el punter del ratolí passi per sobre del menú aquest canvi de colors i imatges automàticament.



Figura 20. Menús usuari

- Navegador de registres:** Perquè l'administrador pugui moure per els registres amb control buttons i Imagebuttons s'ha creat uns navegadors de registres que a més permeten fer algunes tasques com : eliminar, actualitzar, seleccionar cas urgent, etc, etc...

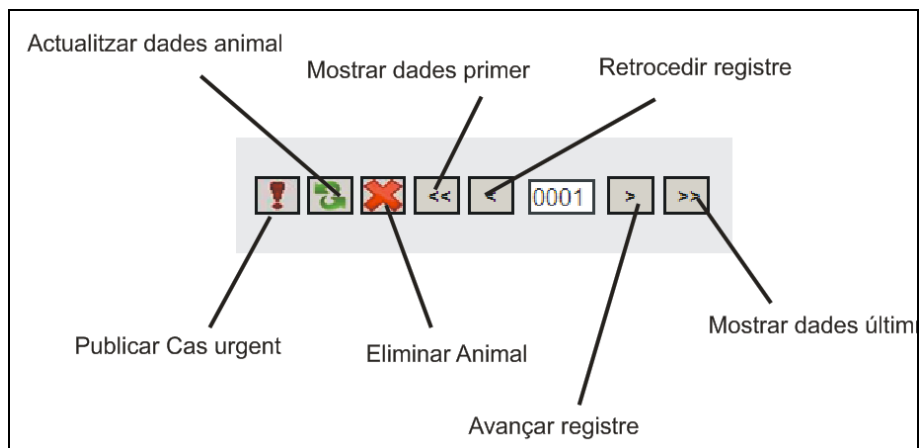


Figura 21. Navegador de registres

- Buscador animals:** Perquè l'usuari pugui buscar la informació (en el nostre cas la dels animals) de manera filtrada i amb agilitat s'han utilitzat uns combobox ja amb les dades , un polsador i un gridview lligat a un origen de dades.

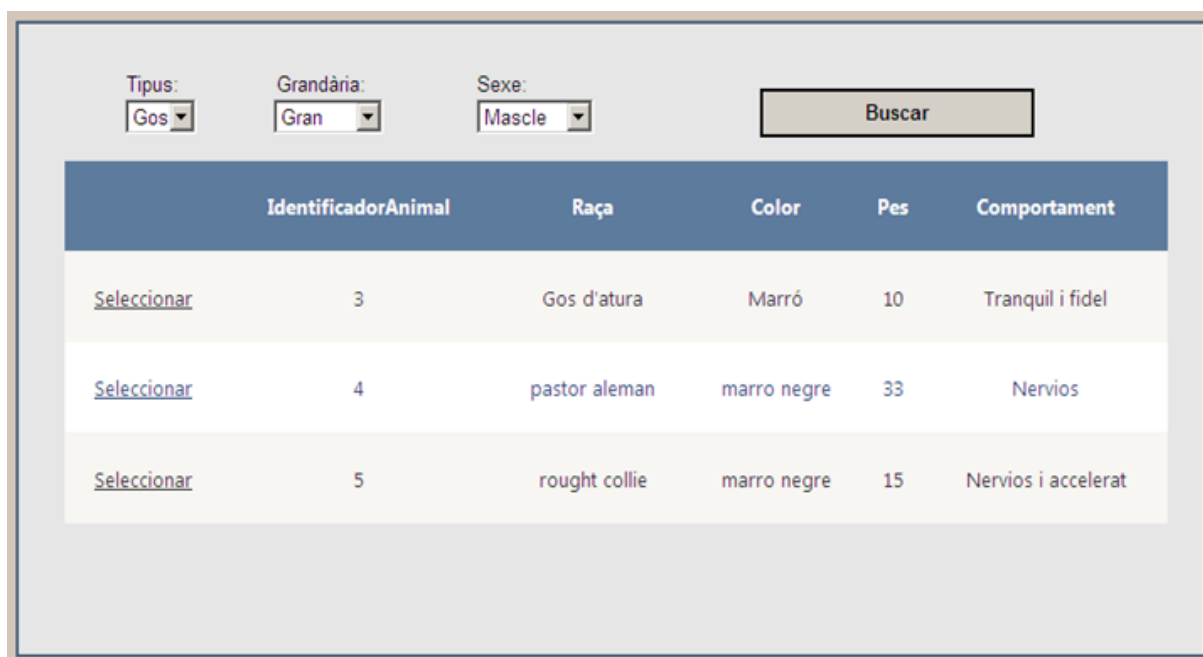


Figura 22. Buscador d'animals

- **Navegador GoogleMaps:** Quan un usuari ha de marcar la ubicació del lloc on va veure la seva mascota per última vegada, les llibreries de GoogleMaps ofereixen una sèrie de control i opcions que faciliten la navegació de l'usuari per el mapa. Pot canviar el mode de visió (satèl·lit o mapa), moure el zoom al l'slider de la pantalla i també amb la rodeta del ratolí.



Figura 23. Mostra navegador GoogleMaps

3.2.3. Diagrama de flux de les pantalles

Els usuaris del portal han de poder moure's amb molta agilitat, sinó podem córrer el risc de que deixin de visitar-lo, per aquesta raó s'ha dissenyat una interfície que permeti anar des de qualsevol pàgina a qualsevol altre. La figura 23 no ho mostra ja que sinó el flux estaria ple de línies, però hem de saber que des de qualsevol pàgina podem anar a principi (Home) i a més a qualsevol pàgina que es tingui permís. El següent diagrama de flux de les pantalles mostra el mapa general de pantalles de la nostra aplicació i quin tipus de login es necessita per accedir-hi.

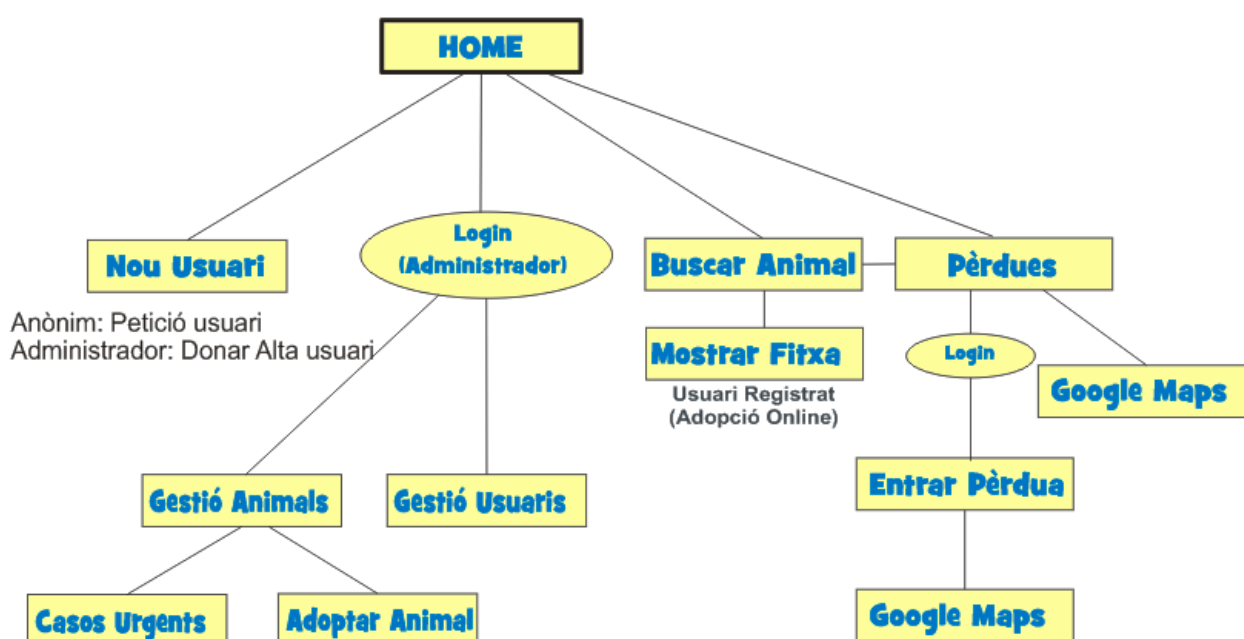


Figura 24. Diagrama de flux de les pantalles

3.2.4. Captures d'algunes pàgines de mostra del projecte

A les següents pàgines es mostren algunes d'aquestes pàgines per tenir una idea de com és realment la nostra interfície gràfica. No hi són totes perquè seria informació molt redundant i la idea és mostrar l'estil de la nostra interfície.



Figura 25. Mostra pàgina principal (Home)

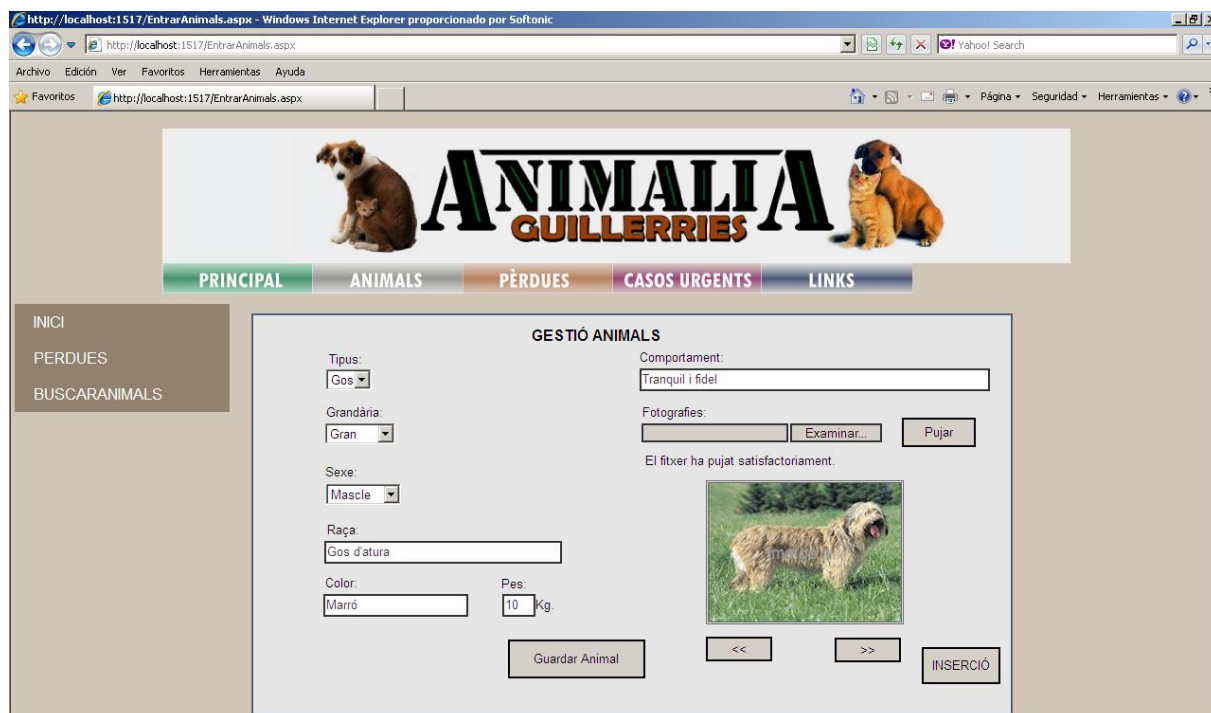


Figura 26. Mostra pàgina gestió animals

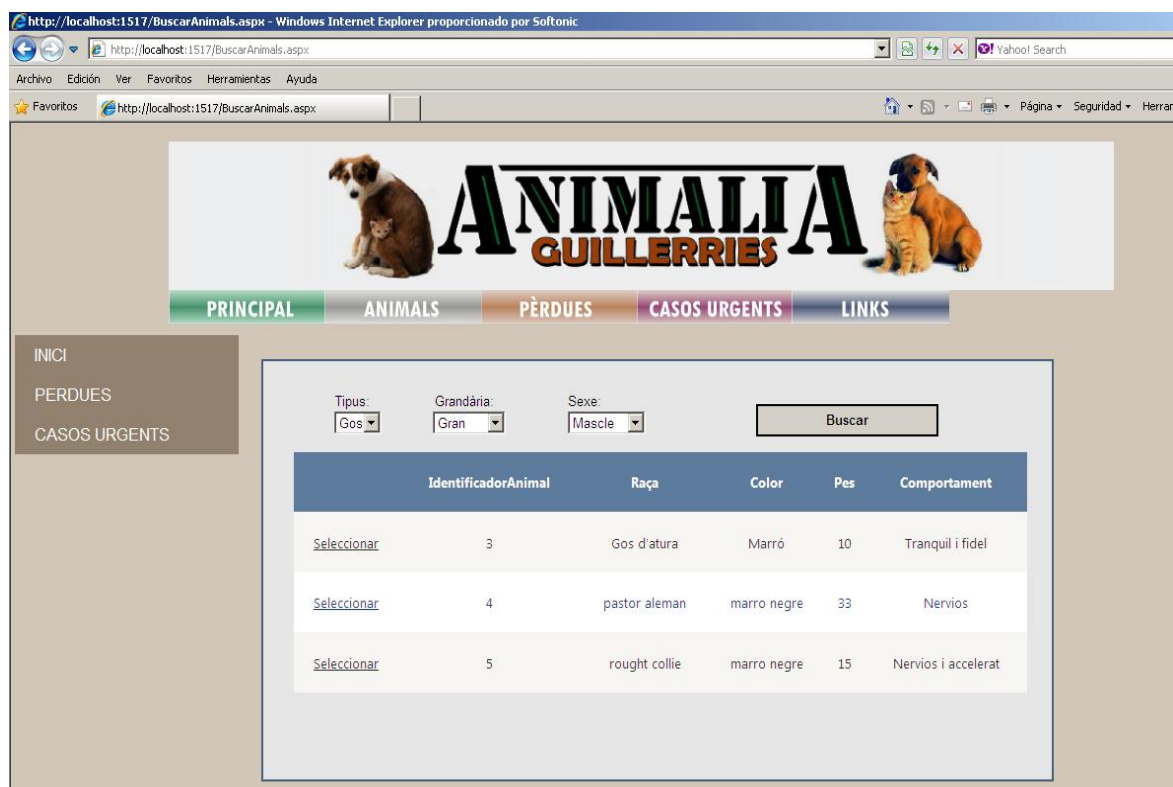


Figura 27. Mostra pàgina buscar animals



Figura 28. Mostra pàgina mostrar fitxa

3.3. Diagrama de classes

Com molt clar a quedat a la capa de lògica negoci és on s'estableixen les regles i per tant també les relacions per això en aquesta implementem el diagrama de classes que s'ha engendrat a partir d'un model conceptual previ.

3.3.1 Model conceptual

Primer s'ha confeccionat les relacions del model conceptual de les classes d'entitat, per després crear el diagrama de les classes d'entitat. A la Pac 2 vaig fer algu semblant però no vaig contemplar la part en que l'administrador publica casos urgents.

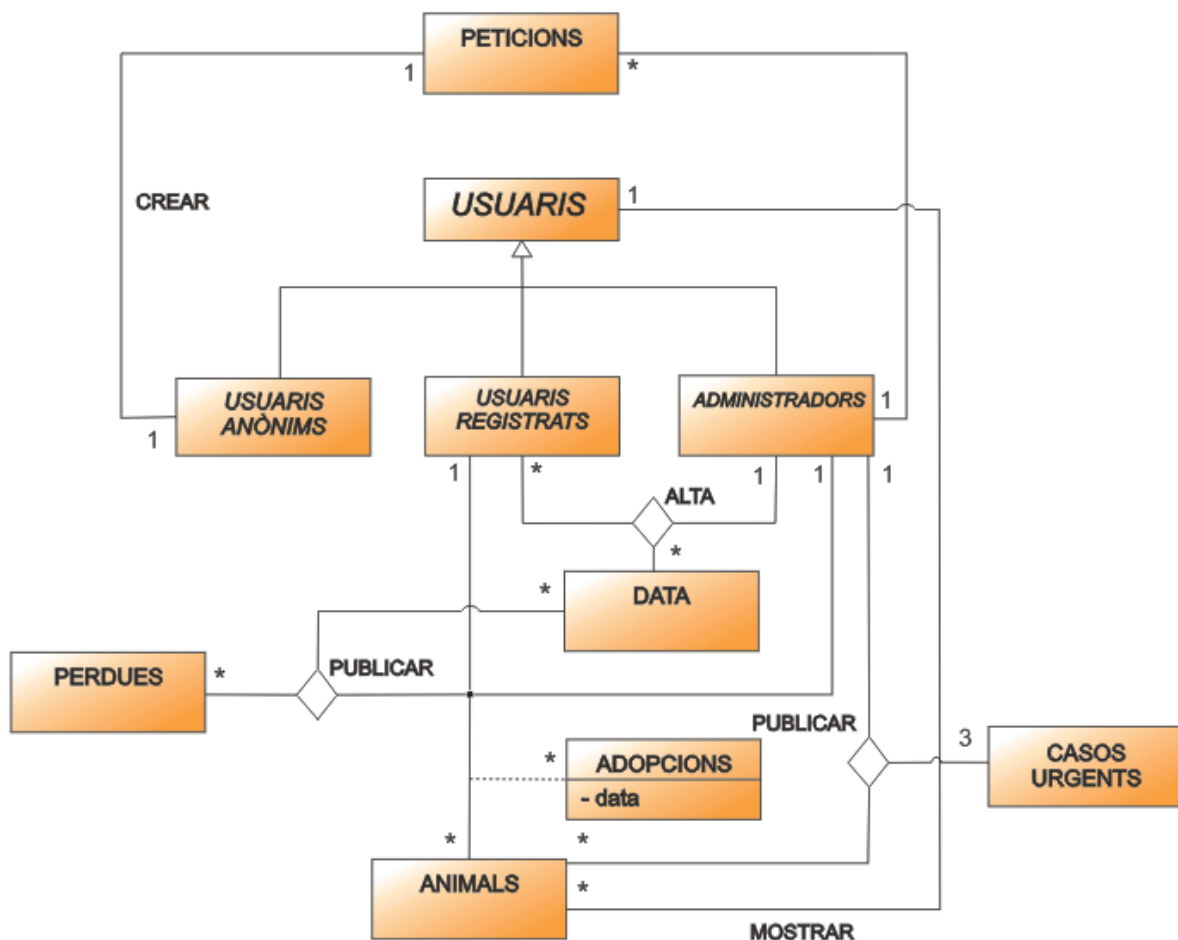


Figura 29. Model conceptual

3.3.2 Diagrama de classes

Ja podem crear el diagrama de les classes d'entitat, amb els atributs i les eventuais relacions d'herència, associacions, agregacions i composicions.

L'entitat Usuaris anònims només utilitza els mètodes de la cerca i els atributs són una mica diferents als dels usuaris registrats, per aquesta raó hem de modificar l'herència de les classes d'usuari i administradors. Administradors hereda de usuaris perquè pot utilitzar tots els mètodes que utilitza un Usuari normal i els atributs són els mateixos.

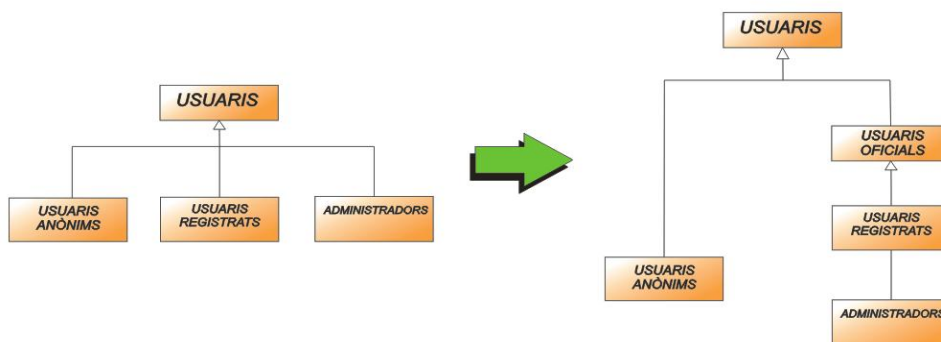


Figura 30. Herència usuaris

I a més, com que la classe UsuariAnònim no té res a veure amb les dels altres dos tipus d'usuari, per a la implementació del diagrama de classes la separem de les classes UsuariRegistrat i Administrador. Per tant desapareix la classe abstracte Usuaris i UsuarisOficials.

En el model conceptual hi tenim una classe associativa que s'haurà de transformar a classe normal perquè el nostre llenguatge de programació no les tracta. Aquesta transformació s'anomena reificació, i el que es fa és relacionar la nova classe amb les dues classes que participen a l'associació. Per tant la classe associativa Adopcions s'haurà de convertir a una classe normal i mantenint els mateixos atributs.

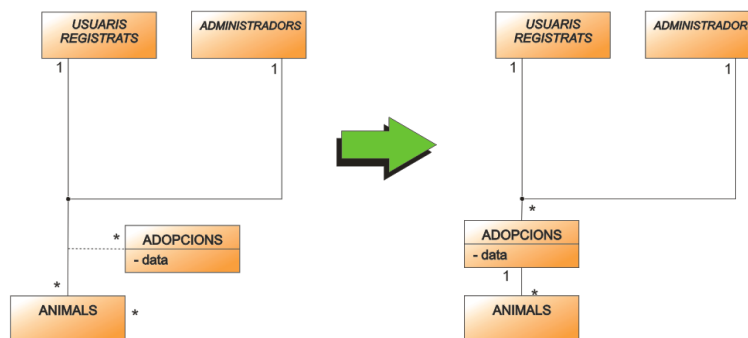


Figura 31. Reificació

El darrer pas consisteix a substituir les associacions per atributs i aquests tindran l'estereotip "ref".

Ara ja podem crear el diagrama de classes, i aquest diagrama és el que s'implementa a la capa de lògica de negoci. La implementació no serà estrictament igual ja que algunes tecnologies que utilitzem per algunes funcions en concret ens obligaran a saltar-nos algunes regles establertes, però com a guia i punt de partida aquest diagrama és ideal. Per exemple hem hagut de crear una classe per els animals resumits del grid view que vol unes declaracions especials per l'origen de dades, i una altre d'animals detallats per la fitxes dels animals

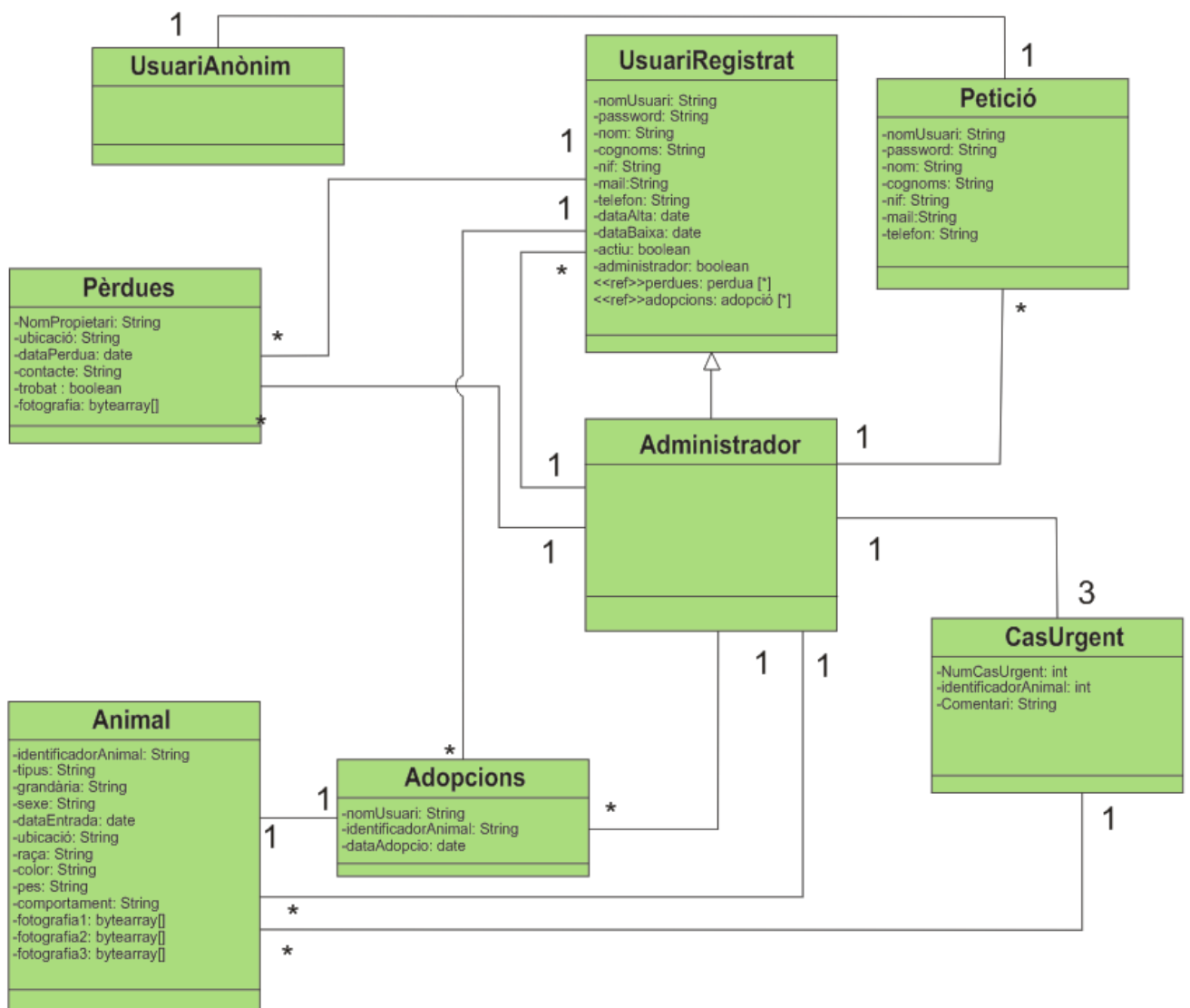


Figura 32. Diagrama de classes

3.3.3 Mètodes més rellevants de les classes principals

En el diagrama de classes no hi ha l'espai suficient per veure els mètodes que implementa cada classe, per aquesta raó en aquest apartat veurem els més rellevants. També donem per entès que tindran els mètodes d'accés getters i setters. Tampoc mostrarem tots els paràmetres dels mètodes per problema d'espai, però que amb la informació que ja tenim són fàcils d'imaginar.

UsuariAnònim

+Constructor

Constructor amb totes les dades de la petició com entrada, i també per sobrecàrrega hi ha un constructor sense paràmetres.

+RegistrarPeticioUsuari

Mètode que registra una petició d'un usuari anònim.

+LoginUsuariAdministrador

Mètode que comprova que l'usuari està registrat i si a més és administrador o usuari normal.

Usuari (Registrat)

+Constructor

Constructor amb totes les dades de l'usuari com entrada, i també per sobrecàrrega hi ha un constructor sense paràmetres.

+List<casUrgent> ObtenirCasosUrgents

Mètode que retorna tots els casos urgents emmagatzemats a la base de dades.

+List<AnimalsResumitsGridView> LlistarAnimalsAmbFiltre

Mètode que demana el llistat dels animals segons tipus grandària i sexe a la cap de dades.

+AnimalDetallatFitxa

Mètode que obté els atributs d'un animal concret segons identificador.

+NomUsuariExisteix

Mètode que verifica si un nom usuari existeix.

+ActualitzaAnimal

Mètode que permet a l'usuari actualitzar l'atribut adopció online.

+RegistrarPèrdua

Mètode que permet a l'usuari o administrador registrar una cas de pèrdua.

Usuari (Registrat)**+ObtenirDadesAnimalPerdut**

Mètode que obté els atributs d'un animal perdut concret segons identificadorAnimal.

+UsuariExist

Mètode que comprova si l'usuari està registrat.

Administrador (Filla de Usuari)**+RegistrarAnimal**

Mètode que permet a l'administrador registrar a un nou animal.

+ActualitzarAnimal

Mètode que permet a l'administrador actualitzar un animal.

+EliminarAnimal

Mètode que permet a l'administrador eliminar un animal.

+RegistrarCasUrgent

Mètode que permet a l'administrador registrar un nou cas urgent.

+ActualitzarCasUrgent

Mètode que permet a l'administrador actualitzar un nou cas urgent.

+ObtenirNumeroCasosUrgents

Mètode que permet a l'administrador saber quants casos urgents hi ha registrats.

+ObtenirCasUrgent

Mètode que permet a l'administrador obtenir un cas urgent concret.

+EliminarCasUrgent

Mètode que permet a l'administrador eliminar un cas urgent concret

+RegistrarNouUsuari

Mètode que permet a l'administrador registrar a un nou usuari.

+ActualitzUsuariAdministrador

Mètode que actualitza a un usuari/administrador registrat.

+List<String> ObtenirLlistatDeTotsNomsUsuaris

Mètode que retorna una llista amb el nomusuari de tots els usuaris.

3.4. Disseny de la base de dades

La base de dades neix a partir de la informació que ens dona el diagrama de classes, i les taules de la base de dades pràcticament seran traduccions de les classes d'entitat.

3.4.1 Taules de la base de dades relacional

-**UsuarisRegistrats** (NomUsuari, Password, Nom, Cognoms, Nif, Mail, Telefon, Actiu, DataAlta, DataBaixa);

-**Animals** (IdentificadorAnimal, Tipus, Grandària, Sexe, Ubicació, Raça, Color, Pes, Comportament, DataEntrada, DataSortida, AdoptarOnline, Fotografia1, Fotografia2, Fotografia3,)

-**Pèrdues** (IdentificadorAnimal, NomPropietari, Ubicació, DataPerdua, Contacte, trobat, Fotografia)

{ IdentificadorAnimal } és clau forana a Animal.

{ NomPropietari } és clau forana a Usuari i Administrador.

-**Adopcion** (NomPropietari, IdentificadorAnimal, dataAdopció)

{ NomPropietari } és clau forana a Usuari i Administrador.

{ IdentificadorAnimal } és clau forana a Animal.

-**Peticions** (NomUsuari, Password, Nom, Cognoms, Nif, Mail, Telefon);

-**CasosUrgents** (NumCasUrgent, IdentificadorAnimal, comentari)

{ IdentificadorAnimal } és clau forana a Animal.

A més a l'script per crear la base de dades l'hi hem afegit algunes restriccions com per exemple en els camps tipus, grandària i sexe de la taula animals que només poden tenir uns valors determinats, i també el numero de casos urgents que no pot ser més gran de 3.

3.4.2 Diagrama de la base de dades

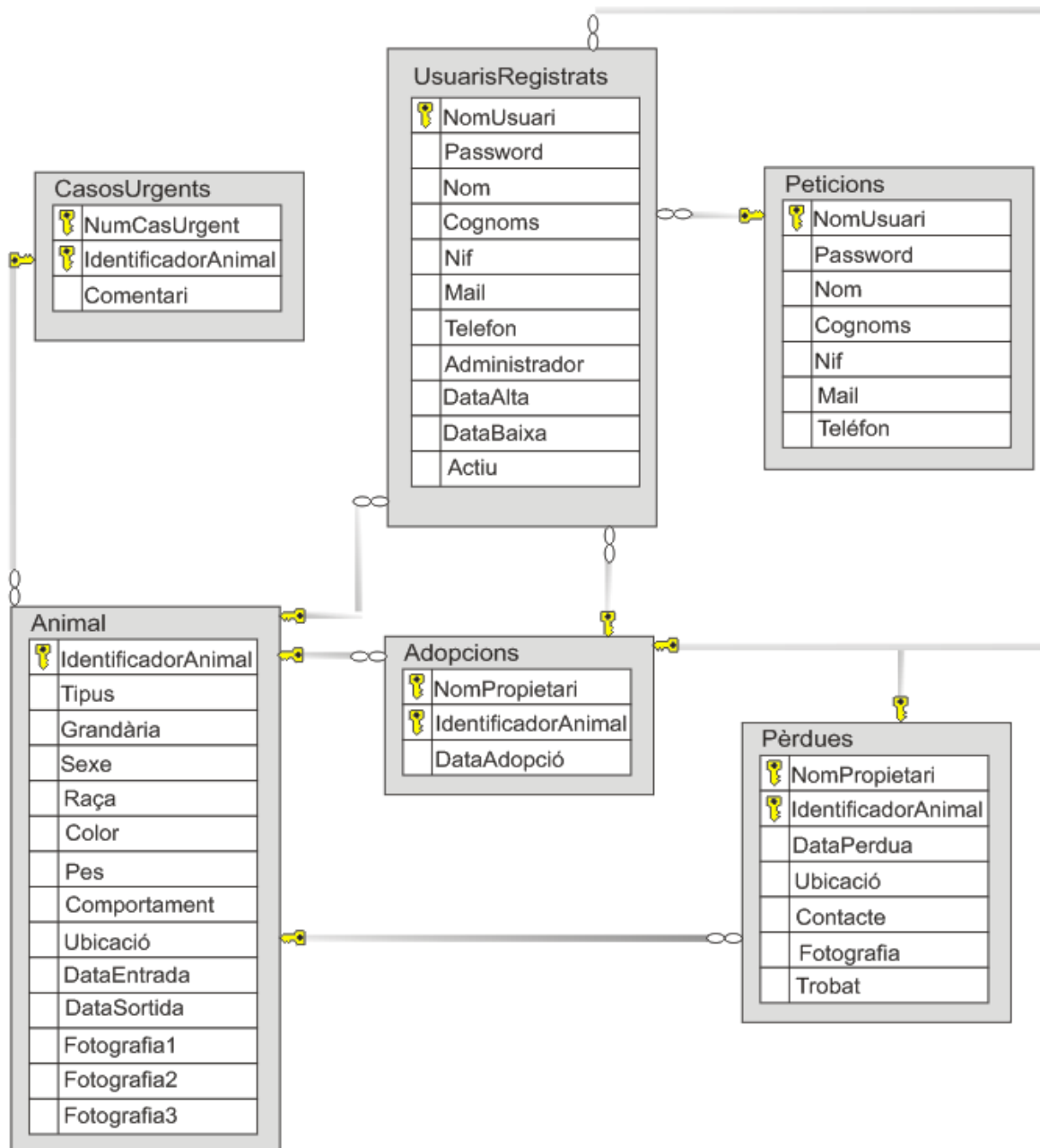


Figura 33. Diagrama base de dades

3.4.3 Declaració de les taules i els seus atributs

Veiem a continuació com hem declarat aquestes taules en un script que executem des del Microsoft SQL Management Studio.

```
CREATE TABLE UsuarisRegistrats
(
    NomUsuari VARCHAR(12) NOT NULL ,
    Password VARCHAR(8) NOT NULL,
    Nom VARCHAR(20) NOT NULL,
    Cognoms VARCHAR(30) NOT NULL,
    Nif VARCHAR(10) NOT NULL,
    Mail VARCHAR(40) NOT NULL,
    Telefon VARCHAR(15) NOT NULL,
    Actiu BIT NOT NULL,
    Administrador Bit NOT NULL,
    DataAlta DATETIME NOT NULL,
    DataBaixa DATETIME NULL,
    CONSTRAINT PK_UsuarisRegistrats PRIMARY KEY CLUSTERED (NomUsuari)
);

CREATE TABLE Animals
(
    IdentificadorAnimal INT NOT NULL,
    Tipus VARCHAR(3) NOT NULL,
    Grandaria VARCHAR(6) NOT NULL,
    Sexe VARCHAR(7) NOT NULL,
    Raça VARCHAR(30) NOT NULL,
    Color VARCHAR(20) NOT NULL,
    Pes INT NOT NULL,
    Comportament VARCHAR(100) NOT NULL,
    DataEntrada DATETIME NOT NULL,
    DataSortida DATETIME NULL,
    Ubicacio VARCHAR(200) NULL,
    AdoptatOnline Bit NOT NULL,
    Fotografial1 IMAGE NULL,
    Fotografia2 IMAGE NULL,
    Fotografia3 IMAGE NULL,
    CONSTRAINT ck_Animals CHECK (Tipus='Gat' OR Tipus='Gos') ,
    CONSTRAINT ck_Animals2 CHECK (Grandaria='Cadell' OR Grandaria='Petit' OR
    Grandaria='Mitjà' OR Grandaria='Gran') ,
    CONSTRAINT ck_Animals3 CHECK (Sexe='Mascle' OR Sexe='Famella'),
    CONSTRAINT pk_Animals PRIMARY KEY CLUSTERED(IdentificadorAnimal)
);

CREATE TABLE Perdues
(
    IdentificadorAnimal INT NOT NULL,
    NomPropietari VARCHAR(12) NOT NULL,
    DataPerdua DATETIME NOT NULL,
    Contacte VARCHAR(50) NOT NULL,
    Ubicacio VARCHAR(200) NULL,
    Fotografial1 IMAGE NOT NULL,
    Trobat BIT NOT NULL,
    CONSTRAINT pk_Perdues PRIMARY KEY
    CLUSTERED(IdentificadorAnimal, NomPropietari),
    CONSTRAINT fk_Perdues1 FOREIGN KEY (NomPropietari) REFERENCES
    UsuarisRegistrats (NomUsuari)
);
```

```

CREATE TABLE CasosUrgents
(
    NumCasUrgent INT NOT NULL,
    IdentificadorAnimal INT NOT NULL,
    Comentari VARCHAR(100) NULL,
    CONSTRAINT pk_CasosUrgents PRIMARY KEY CLUSTERED (NumCasUrgent),
    CONSTRAINT fk_CasosUrgents FOREIGN KEY (IdentificadorAnimal) REFERENCES
Animals (IdentificadorAnimal),
    CONSTRAINT ck_CasosUrgents CHECK (NumCasUrgent < 4)
);

CREATE TABLE Adopcions
(
    IdentificadorAnimal INT NOT NULL,
    NomPropietari VARCHAR(12) NOT NULL,
    DataAdopcio DATETIME NOT NULL,
    CONSTRAINT pk_Adopcions PRIMARY KEY
CLUSTERED (IdentificadorAnimal, NomPropietari),
    CONSTRAINT fk_Adopcions FOREIGN KEY (IdentificadorAnimal) REFERENCES
Animals (IdentificadorAnimal),
    CONSTRAINT fk_Adopcions3 FOREIGN KEY (NomPropietari) REFERENCES
UsuarisRegistrats (NomUsuari)
);

CREATE TABLE PeticionsUsuaris
(
    NomUsuari VARCHAR(12) NOT NULL,
    Password VARCHAR(8) NOT NULL,
    Nom VARCHAR(20) NOT NULL,
    Cognoms VARCHAR(30) NOT NULL,
    Nif VARCHAR(10) NOT NULL,
    Mail VARCHAR(40) NOT NULL,
    Telefon VARCHAR(15) NOT NULL,
    CONSTRAINT PK_PeticionsUsuaris PRIMARY KEY CLUSTERED (NomUsuari)
);

```

3.4.4 Linq to SQL

Ja hem comentat que utilitzem Linq SQL per l'accés a les dades des del .NET . Linq to SQL és un component del framework .Net que proporciona una infraestructura en temps d'execució per administrar les dades relacionals com si fossin objectes, i així d'una manera molt més senzilla podem actualitzar, afegir i eliminar dades de la base de dades.

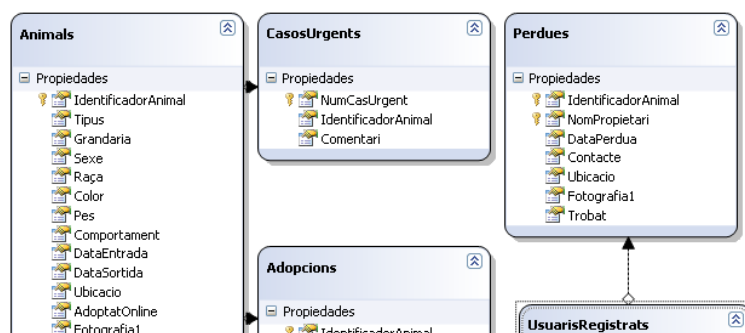


Figura 34. Diagrama Linq to SQL en el Visual Studio 2010

3.4. Arquitectura física

L'arquitectura de la nostra aplicació web amb ASP.NET amb accés a un servidor de base de dades segueix la següent estructura.

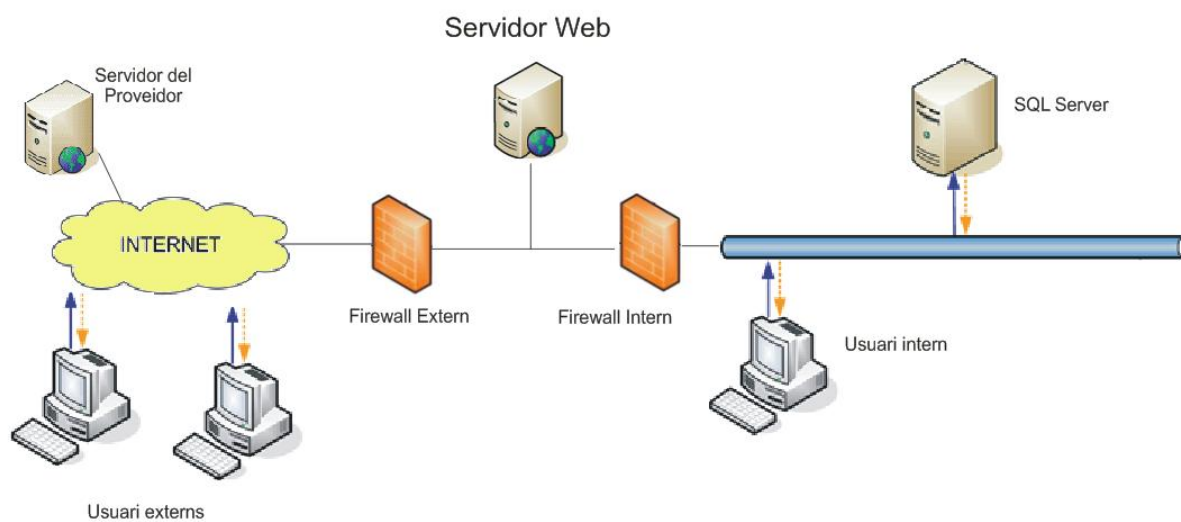


Figura 35. Arquitectura física

4. Línies de desenvolupament futures i conclusions

Aquest projecte encara està obert a créixer amb millores i noves funcionalitats que ajudin a atraure encara més a la gent a visitar el portal web. Per això, si hagués d'estar en la fase de manteniment i seguiment d'aquest projecte hi afegiria un apartat destinat a fòrums i debats, on els usuaris registrats tractessin temes d'actualitat sobre el món animal en general.

A més crearia una botiga virtual on s'hi pogués vendre accessoris per a les mascotes dels usuaris, sense pretendre fer la competència a ningú. D'aquesta forma la protectora tindria una font d'ingressos adicional.

Les conclusions que he extret després de elaboració d'aquest projecte són que la plataforma.NET està en una edat molt madura i ofereix un gran ventall de prestacions, i que a més els seus límits estan molt lluny ja que jo en cap moment n'hi he trobat cap. El límit d'un projecte està en la pròpia creativitat i imaginació d'un mateix.

També vull comentar que al principi tenia molta fe en que els llibres que vaig adquirir serien el meu pilar de suport, i a l'hora de posar-m'hi el meu suport real han estat els fòrum si els "tutorials" que he anat trobant per internet. La informació dels llibres estan molt bé per donar-te una primera presa de contacte però ho toquen tot a grans trets i no són molt especialitzats. En canvi a internet trobes informació molt útil ja que les necessitats que et van sorgint sobre la marxa, coincideixen amb les que algun dia algun altre programador va tenir, i aquest un cop ho ha resolt ho ha volgut compartir amb la resta.

I el més important de tot és que després de fer aquest projecte ja en tinc d'altres en ment i els vull fer en .Net perquè he après molt d'aquesta plataforma i tinc ganes de conèixer-la encara més.

5. Bibliografía

- **Profesor Francisco Javier Ceballos Sierra** Enciclopedia de Microsoft Visual C#.
- **Profesor Francisco Javier Ceballos Sierra** *El lenguaje de programación C#*
- **George Shepherd** ASP.NET 4 Paso a Paso
- **Programacion Web ++** : <http://mycodigosweb.blogspot.com/>
- **Wikipèdia** : <http://es.wikipedia.org/>
- **El Guille**: www.elguille.info/
- **AJAX Control Toolkit**: <http://www.asp.net/ajax/ajaxcontroltoolkit/samples/>
- Pàgines web varies dedicades a la programació.