

Catàleg de gossos – J2EE

Marçal Nebot
ETIG

Verònica Peña Pastor

14 de gener de 2009

Resum

El Treball fi de carrera ha servit per a realitzar un treball de síntesi dels coneixements adquirits en altres assignatures de la carrera i ha requerit posar-los en pràctica conjuntament en un treball concret. També ha sigut necessari realitzar tasques de recerca i investigació. El TFC ha sigut un treball eminentment pràctic i vinculat a l'exercici professional de la informàtica.

En aquest TFC concret s'ha realitzat una aplicació J2EE simple, a mode d'exemple. Concretament s'ha realitzat el típic "Catàleg de gossos" que és l'"Hola món" de les aplicacions J2EE.

El món del J2EE és molt gran, hi ha una infinitat de *marcs de treball*, tècniques i utilitats diferents. Aquest TFC ha utilitzat els Patrons de disseny més habituals, els marcs de treball *Spring* i *Struts2* així com persistència utilitzant *JPA* i ha utilitzat una *llibreria de tags d'AJAX*.

Aquesta memòria pretén reflectir el procés de creació de l'aplicació, explicar les decisions preses i mostra el resultat final, el producte.

Índex

Resum	2
Índex	3
Introducció	5
<i>Justificació del TFC</i>	5
<i>Objectius del TFC</i>	5
<i>Enfocament i mètode seguit</i>	5
<i>Planificació del TFC</i>	6
<i>Producte obtingut</i>	6
<i>Breu descripció dels altres capítols de la memòria</i>	7
Anàlisi de les funcions	8
<i>Anàlisi de requisits funcionals</i>	8
Requeriments aplicació pública	8
Requeriments aplicació privada	8
<i>Casos d'ús</i>	8
Casos d'ús usuari.....	8
Casos d'ús administrador	9
<i>Interfície d'usuari</i>	13
Pantalles usuari	13
Pantalles administrador.....	16
Arquitectura de l'aplicació	21
<i>Introducció</i>	21
<i>Patrons de disseny utilitzats</i>	21
MVC	21
Intercepting Filter	21
Front Controller.....	22
Business delegate	23
Façade.....	24
Application Service.....	24
Data Access Object.....	24
<i>Marc de treball utilitzats</i>	24
Spring framework	24
Struts 2.....	25
<i>Nivell de persistència, JPA</i>	25
<i>Eines</i>	26
Ajaxtags.....	26
Log4j.....	27
Disseny de l'aplicació	28
<i>Model de dades</i>	28
Gos.....	28
Raca.....	28
<i>Entitats</i>	28
<i>DAO's</i>	28
<i>Objectes de negoci i façanes de l'aplicació</i>	30
<i>Accions</i>	30
<i>Filtres</i>	31
<i>JSP's</i>	32
<i>Configuració web</i>	40
<i>Configuració marc de treball Spring</i>	42
<i>Configuració marc de treball Struts</i>	44
<i>Configuració persistència</i>	45
Resum de components	46

<i>La base de dades</i>	46
<i>Marc de treball</i>	46
<i>Component d'Integració</i>	46
<i>Component de la lògica de negoci</i>	46
<i>Component AJAX</i>	46
<i>Aplicació web</i>	46
Conclusions	48
Glossari	49
Bibliografia	50

Introducció

Justificació del TFC

El projecte final de carrera que he escollit havia d'estar relacionat amb J2EE, s'havia de posar en pràctica el màxim de coneixements que hem après durant la carrera i s'havia de realitzar en un període de temps determinat.

Per tant he triat realitzar una aplicació que es pogués dur a terme el temps que teníem, en principi 75 hores, les corresponents a 7,5 crèdits que són els que té l'assignatura.

Degut a aquesta escassetat de recursos he retallat funcionalitats per a concentrar-me en crear una aplicació de qualitat, sobretot en la utilització dels marcs de treball i la utilització de patrons.

Objectius del TFC

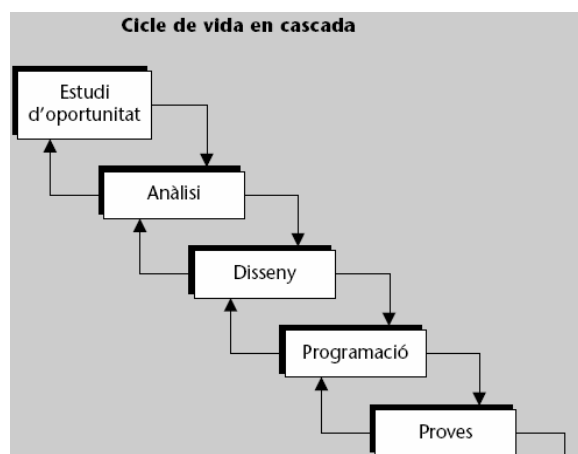
L'objectiu principal d'aquest projecte és crear una aplicació J2EE senzilla, que utilitzi els patrons de disseny més usuals de J2EE, especialment el MVC, els marcs de treball més avançats, segons el meu criteri personal, que utilitzi l'especificació JPA per a la persistència de dades i que utilitzi una eina per a l'ús d'AJAX.

Un altre objectiu important ha sigut crear una aplicació que serveixi com a **punt d'arrencada** per a crear aplicacions que:

- Prescindeixin dels complexos EJB sense perdre les funcionalitats pròpies dels mateixos
- En tinguin prou amb un servidor de pàgines jsp com ara el Tomcat però que es puguin utilitzar en qualsevol servidor d'aplicacions
- Siguin independents de la plataforma i de la base de dades
- Utilitzin només codi obert
- Siguin fàcilment escalables, ampliables i modularidzables
- Estiguin desacoblades i estiguin estructurades per capes: integració, negoci i vista

Enfocament i mètode seguit

L'enfocament i el mètode seguit són els que hem après al llarg de la carrera en diferents assignatures, concretament el cicle de vida en cascada

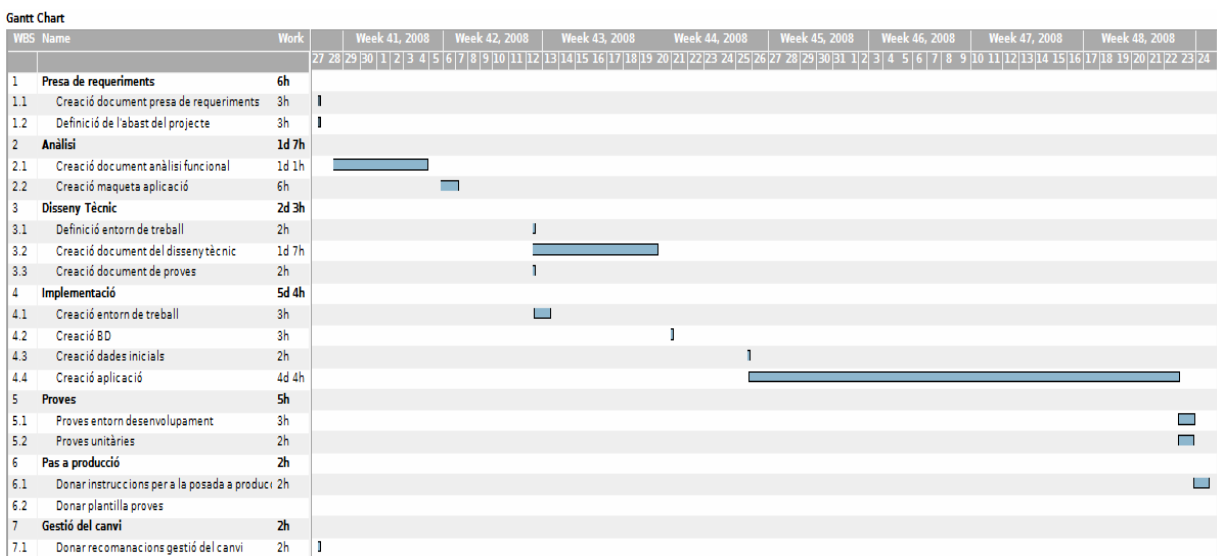


En primer lloc s'ha consensuat amb el consultor l'abast i els requeriments que l'aplicació havia de tenir. Un cop això s'ha aconseguit s'ha procedit a realitzar l'aplicació segons el procediment habitual de creació d'una aplicació informàtica:

- Planificació del projecte
- Realització de l'anàlisi funcional
- Realització del disseny tècnic
- Implementació de l'aplicació
- Realització de proves
- Entrega de l'aplicació

Planificació del TFC

La planificació del projecte s'ha creat mitjançant una eina específica per a tal efecte i ha sigut la següent:



Cal remarcar que s'ha complert escrupolosament.

Producte obtingut

El producte final obtingut és una aplicació web J2EE "Catàleg de gossos" que està composta per una part pública i una altra de privada per a realitzar el manteniment del catàleg i la qual té implementada la seguretat.

L'aplicació està composta per:

- L'aplicació web TFC_MNM.war
- El codi de l'aplicació: classes, fitxers de configuració i scripts de base de dades
- Els documents de l'aplicació: requeriments, planificació, anàlisi i disseny

Breu descripció dels altres capítols de la memòria

Anàlisi de les funcions

Descriu els requeriments i les funcions que realitzarà l'aplicació de catàleg de gossos així com es mostraran els casos d'ús

Arquitectura de l'aplicació

Descriu els marcs de treball, els patrons i les eines que s'utilitzen a l'aplicació

Disseny de l'aplicació

Descriu detalladament la implementació de l'aplicació: les classes que s'han creat, les jsp i els fitxers de configuració

Resum de components

Descriu els diferents components que formen part de l'aplicació

Anàlisi de les funcions

Anàlisi de requisits funcionals

Requeriments aplicació pública

La botiga de gossos vol crear una pàgina web pública on els seus clients i potencials puguin veure tot el seu catàleg de gossos que tenen en venda.

S'haurà de poder accedir des de qualsevol ordinador des de qualsevol lloc connectat a Internet.

Es vol que els gossos estiguin ordenats per races. Al seleccionar una raça haurà d'aparèixer una llista de gossos d'aquella raça. Finalment al triar un gos de la llista es voldrà que apareixi la fitxa del gos.

La fitxa haurà de contenir el nom del gos, una fotografia i una descripció.

Les races hauran d'estar sempre visibles, i quan es visualitzi una fitxa de gos la llista de gossos de la mateixa raça haurà també de ser visible.

Es valorarà molt el fet que la càrrega de la pàgina sigui ràpida.

Requeriments aplicació privada

L'aplicació privada haurà de mantenir les fitxes dels gossos.

S'haurà de poder accedir des de qualsevol ordinador des de qualsevol lloc connectat a Internet.

Hi haurà un sol usuari que hi podrà accedir amb una contrasenya.

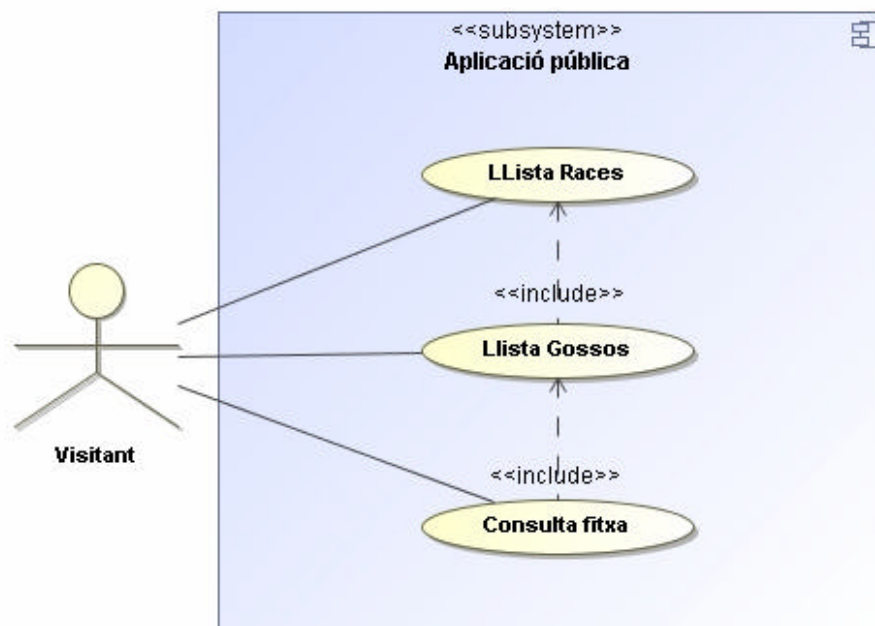
Es valorarà que l'aplicació de manteniment s'assembli al màxim a l'aplicació pública.

Les races dels gossos no es mantindran des d'aquesta aplicació ja que en principi no variaran.

Casos d'ús

Casos d'ús usuari

Seleccionar fitxa gos



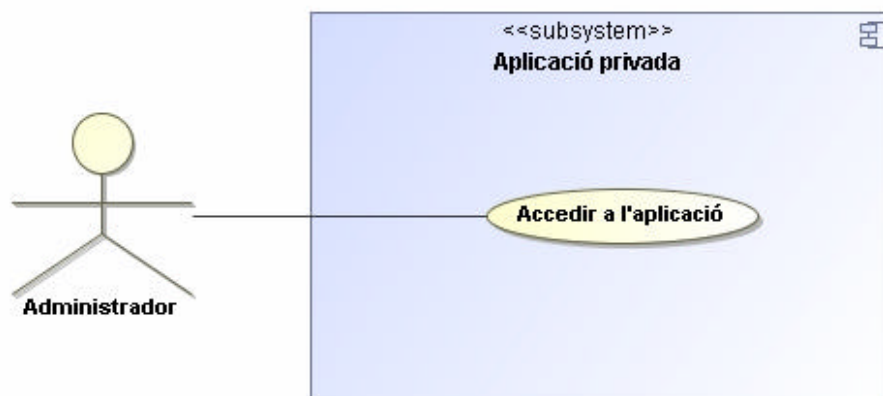
Nom	Llista Races
Descripció	El visitant entra a l'aplicació, aleshores apareix la llista de races del catàleg de gossos. Des de la llista de races el visitant podrà cridar a la llista de gossos d'una determinada raça
Actor	Visitant
Precondició	Haver entrat a l'aplicació pública
Postcondició	Les races surten llistades per pantalla

Nom	Llista Gossos
Descripció	El visitant crida la llista de gossos d'una raça des de la llista de races. La llista de races es manté visible i apareix també la llista de gossos de la raça seleccionada
Actor	Visitant
Precondició	Haver seleccionat una raça des de la llista de races
Postcondició	Les races surten llistades per pantalla i els gossos de la raça seleccionada també

Nom	Consulta fitxa
Descripció	El visitant crida una fitxa de gos d'una llista de gossos. La llista de races i de gossos de la raça seleccionada prèviament es mantenen visibles i apareix també la fitxa del gos seleccionat
Actor	Visitant
Precondició	Haver seleccionat una raça des de la llista de races i haver seleccionat un gos de la llista de gossos de la raça seleccionada
Postcondició	Les races surten llistades, els gossos de la raça seleccionada també i la fitxa del gos seleccionat és visible per pantalla

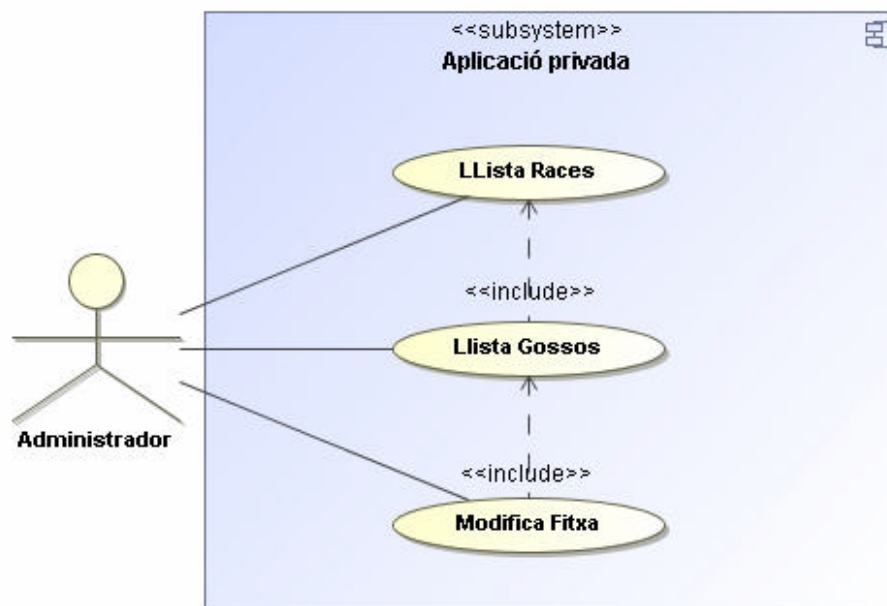
Casos d'ús administrador

Accedir a l'aplicació



Nom	Accedir a l'aplicació
Descripció	L'administrador accedeix a l'aplicació mitjançant una contrasenya Si s'equivoca de contrasenya li apareixerà un missatge d'error Si la contrasenya introduïda és correcta entrarà a l'aplicació
Actor	Administrador
Precondició	Haver entrat a l'aplicació privada
Postcondició	Es mostrarà un missatge d'error si s'ha equivocat a l'introduir la contrasenya S'entrarà a l'aplicació privada si la contrasenya és correcta

Modificar fitxa gos

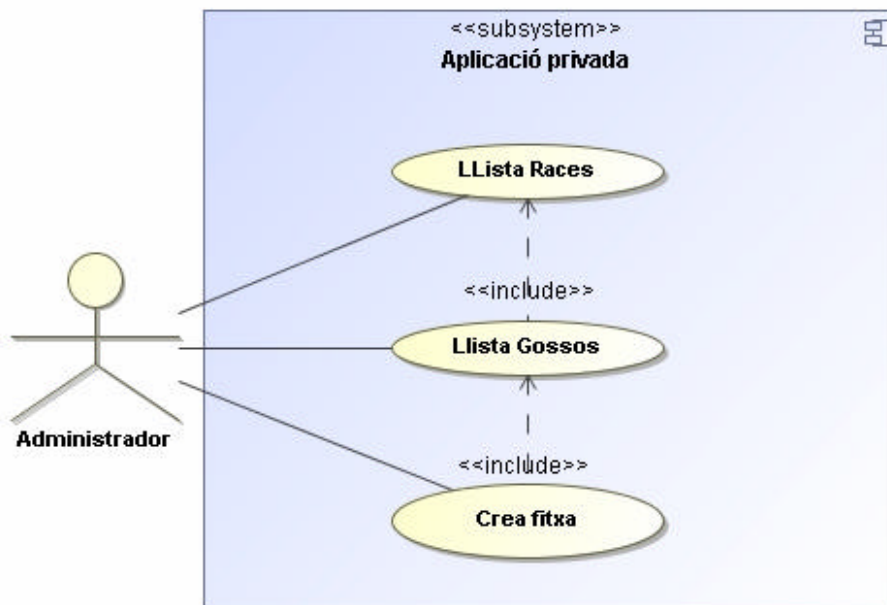


Nom	Llista Races
Descripció	L'administrador entra a l'aplicació, aleshores apareix la llista de races del catàleg de gossos. Des de la llista de races podrà cridar a la llista de gossos d'una determinada raça
Actor	Administrador
Precondició	Haver entrat a l'aplicació privada
Postcondició	Les races surten llistades per pantalla

Nom	Llista Gossos
Descripció	L'Administrador crida la llista de gossos d'una raça des de la llista de races. La llista de races es manté visible i apareix també la llista de gossos de la raça seleccionada Des de la llista de gossos es podrà cridar les opcions de crear, modificar o eliminar un gos
Actor	Administrador
Precondició	Haver seleccionat una raça des de la llista de races
Postcondició	Les races surten llistades per pantalla i els gossos de la raça seleccionada també juntament amb les opcions d'edició dels gossos

Nom	Modifica fitxa
Descripció	L'Administrador crida una fitxa de gos d'una llista de gossos. La llista de races i de gossos de la raça seleccionada prèviament es mantenen visibles i apareix també la fitxa del gos seleccionat podent-se modificar els atributs
Actor	Administrador
Precondició	Haver seleccionat una raça des de la llista de races i haver seleccionat un gos de la llista de gossos de la raça seleccionada per a ser modificat
Postcondició	Les dades introduïdes han hagut de superar la validació Les races surten llistades i els gossos de la raça seleccionada també amb les dades actualitzades (per si ha canviat el nom del gos)

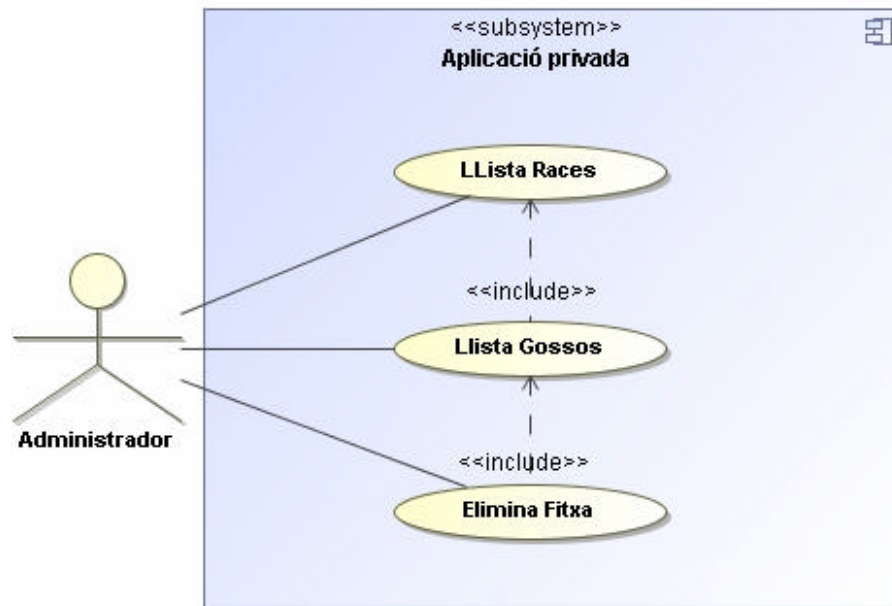
Crear fitxa gos



Nota: Per a consultar els casos d'ús de Llista Races i Llista Gossos mirar el cas d'ús de Modificar fitxa gos.

Nom	Crear fitxa
Descripció	La llista de races i de gossos de la raça seleccionada prèviament es mantenen visibles i apareix també una fitxa de gos buida per a poder introduir els atributs i poder crear una fitxa
Actor	Administrador
Precondició	Haver seleccionat una raça des de la llista de races i haver seleccionat l'opció de creació de fitxa de gos
Postcondició	Les dades introduïdes han hagut de superar la validació Les races surten llistades, els gossos de la raça seleccionada també incloent-hi la nova fitxa

Eliminar fitxa gos



Nota: Per a consultar els casos d'ús de Llista Races i Llista Gossos mirar el cas d'ús de Modificar fitxa gos.

Nom	Elimina fitxa
Descripció	La llista de races i de gossos de la raça seleccionada prèviament es mantenen visibles i s'elimina el gos que es selecciona
Actor	Administrador
Precondició	Haver seleccionat una raça des de la llista de races i haver seleccionat l'opció d'eliminació per a una fitxa de gos
Postcondició	Les races surten llistades, els gossos de la raça seleccionada també però sense la fitxa de gos eliminada

Interfície d'usuari

Pantalles usuari

Pantalla amb categories



Es mostraran les races.

Només es podrà seleccionar una de les races llistades. El resultat serà que es llistaran els gossos de la raça.

Pantalla amb categories i llista de gossos

Races	Gossos	Fitxa
<ul style="list-style-type: none">■ Doberman■ Pastor Alemà■ Golden Retriever■ Cocker■ Galg■ Boxer■ Caniche■ Yorkshire Terrier	<p>Doberman</p> <ul style="list-style-type: none">■ Doberman 1■ Doberman 2■ Doberman 3■ Doberman 4■ Doberman 5■ Doberman 6■ Doberman 7■ Doberman 8	

Es mostraran les races i els gossos de la raça seleccionada.

Es podrà seleccionar una de les races llistades. El resultat serà que es llistaran els gossos de la raça.

També es podrà seleccionar un gos, en aquest cas es mostrarà la fitxa del gos.

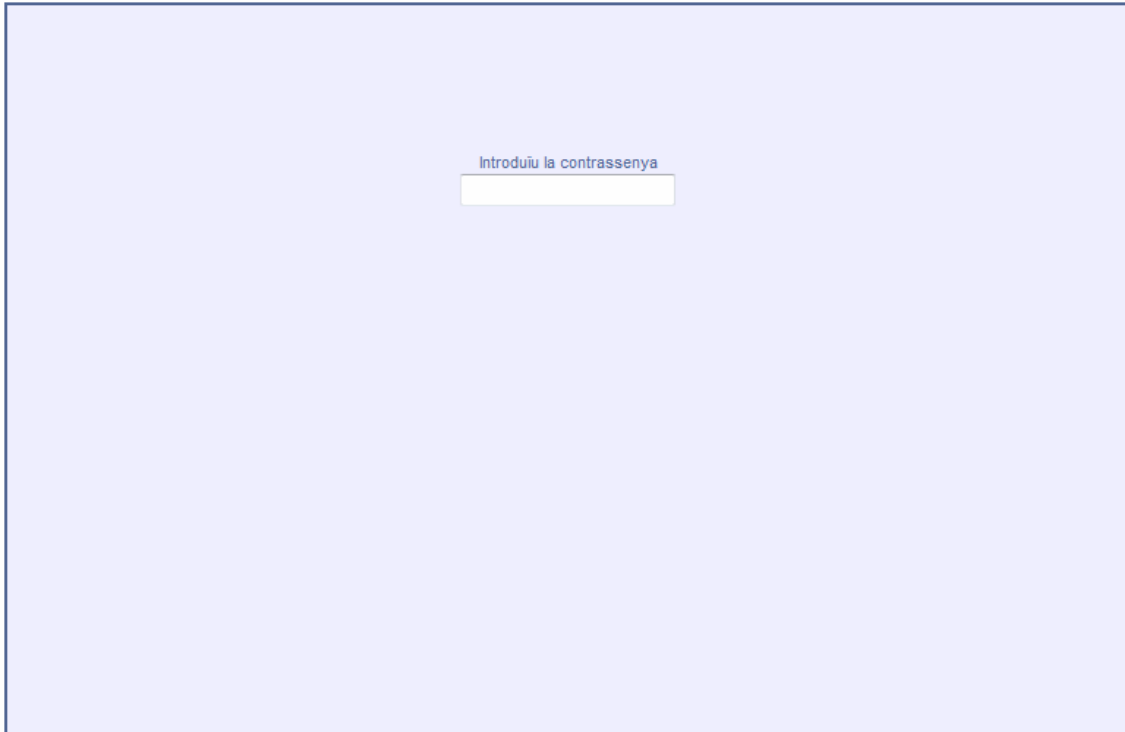
Pantalla amb categories, llista de gossos i fitxa del gos

Races <ul style="list-style-type: none">■ Doberman■ Pastor Alemà■ Golden Retriever■ Cocker■ Galg■ Boxer■ Caniche■ Yorkshire Terrier	Gossos Doberman <ul style="list-style-type: none">■ Doberman 1■ Doberman 2■ Doberman 3■ Doberman 4■ Doberman 5■ Doberman 6■ Doberman 7■ Doberman 8	Fitxa Doberman 1 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent gravida adipiscing libero. Integer viverra augue id lectus egestas laoreet. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris libero enim, mollis vitae, lobortis eu, iaculis et, dui. Sed euismod lorem sit amet libero aliquet pellentesque. Proin vitae augue. Suspendisse condimentum eros</p> 
---	--	---

Es mostraran les races, els gossos de la raça seleccionada i la fitxa del gos seleccionat.
Es podrà seleccionar una de les races llistades. El resultat serà que es llistaran els gossos de la raça i desapareixerà la fitxa del gos seleccionat.
També es podrà seleccionar un gos, en aquest cas es mostrarà la fitxa del gos.

Pantalles administrador

Pantalla introducció contrasenya



The screenshot shows a light blue background with a central white rectangular input field. Above the input field, the text "Introduïu la contrassenya" is displayed in a small, dark blue font.

Si es vol accedir a l'aplicació privada primer s'haurà d'introduir la contrasenya de l'aplicació des d'aquesta pantalla.

En cas de que la contrasenya introduïda sigui incorrecte es mostrarà un missatge d'error.

Quan la contrasenya introduïda sigui correcte es redirigirà a l'aplicació privada.

Pantalla amb categories

Races	Gossos	Fitxa
<ul style="list-style-type: none">■ Doberman■ Pastor Alemà■ Golden Retriever■ Cocker■ Galg■ Boxer■ Caniche■ Yorkshire Terrier		

Es mostraran les races.

Només es podrà seleccionar una de les races llistades. El resultat serà que es llistaran els gossos de la raça.

Pantalla amb categories i llista de gossos



Es mostraran les races i els gossos de la raça seleccionada.

Es podrà seleccionar una de les races llistades. El resultat serà que es llistaran els gossos de la raça.

Es podrà seleccionar un gos, en aquest cas es mostrarà la fitxa del gos per a poder ser modificada.

Es podrà seleccionar l'opció de crear fitxa de gos apretant la fulla blanca al costat del nom de la raça a la llista de gossos.

Es podrà eliminar una fitxa de gos seleccionant l'escombraria al costat de cada fitxa gos.

Pantalla amb categories, llista de gossos i fitxa del gos editable



Es mostraran les races, els gossos de la raça seleccionada i la fitxa amb els atributs modificables. Es podrà seleccionar una de les races llistades. El resultat serà que es llistaran els gossos de la raça.

Es podrà seleccionar un gos, en aquest cas es mostrarà la fitxa del gos per a poder ser modificada.

Es podrà seleccionar l'opció de crear fitxa de gos apretant la fulla blanca al costat del nom de la raça a la llista de gossos.

Es podrà eliminar una fitxa de gos seleccionant l'escombraria al costat de cada fitxa gos.

Es podrà modificar els atributs de la fitxa del gos i editar-los. Un cop editats es mostraran missatges d'error en cas de que les dades no es validin correctament. Si les dades es validen correctament la fitxa desapareixerà i la llista de gossos s'actualitzarà.

Les validacions seran les següents:

- El nom no pot sobrepassar el tamany màxim i com a mínim haurà de tenir 3 caràcters
- La descripció no pot sobrepassar el tamany màxim i és optativa
- La imatge introduïda ha d'existir al servidor i ha de ser del tipus .jpg o .gif

Si no s'introdueix cap imatge la validació es supera i no es modifica la imatge.

Pantalla amb categories, llista de gossos i introducció fitxa de gos

Es mostraran les races, els gossos de la raça seleccionada i la fitxa amb els atributs modificables. Es podrà seleccionar una de les races llistades. El resultat serà que es llistaran els gossos de la raça.

Es podrà seleccionar un gos, en aquest cas es mostrarà la fitxa del gos per a poder ser modificada.

Es podrà seleccionar l'opció de crear fitxa de gos apretant la fulla blanca al costat del nom de la raça a la llista de gossos.

Es podrà eliminar una fitxa de gos seleccionant l'escombraria al costat de cada fitxa gos.

Es podrà modificar els atributs de la fitxa del gos i editar-los. Un cop editats es mostraran missatges d'error en cas de que les dades no es validin correctament. Si les dades es validen correctament la fitxa desapareixerà i la llista de gossos s'actualitzarà.

Les validacions seran les següents:

- El nom no pot sobrepassar el tamany màxim i com a mínim haurà de tenir 3 caràcters
- La descripció no pot sobrepassar el tamany màxim i és optativa
- La imatge introduïda ha d'existir al servidor i ha de ser del tipus .jpg o .gif i és obligatori introduir una imatge

Arquitectura de l'aplicació

Introducció

L'aplicació de Catàleg de gossos és una aplicació J2EE que compleix amb els estàndards d'aquest tipus d'aplicacions utilitzant els patrons propis d'aquesta tecnologia: MVC, filter dispatcher, business delegate i altres. En quant a la persistència de dades utilitza la especificació JPA que forma part de l'última especificació de J2EE.

Per a implementar l'aplicació s'ha utilitzat dos marcs de treball: Spring per a la inversió de control i Struts per a la implementació del model MVC i per a facilitar la construcció de jsp's entre altres ajudes.

També s'ha utilitzat una eina per a l'ús d'AJAX per a fer l'aplicació més dinàmica i ràpida.

Patrons de disseny utilitzats

Els patrons de disseny serveixen per a donar una solució ja provada a problemes comuns. La utilització dels patrons no garanteix la qualitat de l'aplicació que els utilitza però si augmenta la probabilitat de l'èxit.

Uns dels avantatges que s'espera adquirir a l'utilitzar els patrons J2EE i MVC que es mostren a continuació són:

- Ordenació del codi
- Desacoblament de les capes
- Augment de la flexibilitat
- Facilitat de manteniment
- Facilitat de la comprensió

MVC

El patró d'arquitectura i de disseny Model - View - Controller serveix per a isolar la lògica de negoci (Model) de la interfície d'usuari (View) mitjançant un controlador (Controller) que gestiona la comunicació entre un i l'altre.

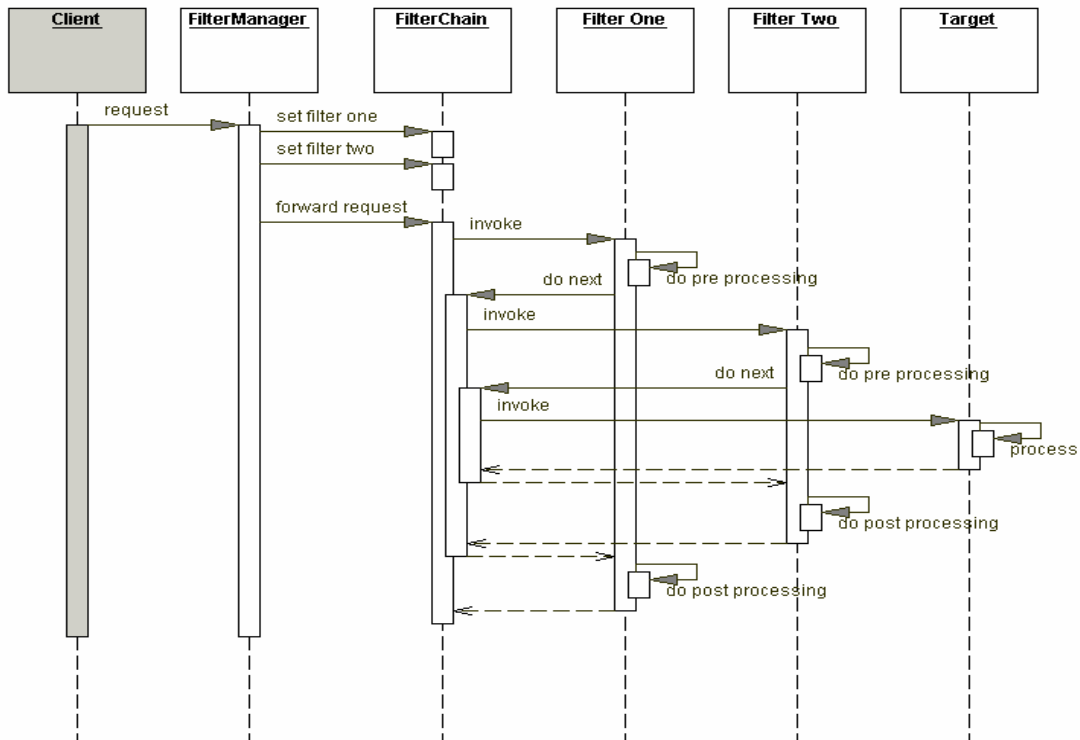
Algunes avantatges d'utilitzar aquest model són:

- Possibilitat de modificar la vista sense afectar el model i viceversa
- Centralització de la navegació de l'aplicació en un sol element, el controlador
- Possibilitat de crear més d'una vista per un sol model
- Ordenació i especialització del codi
- Facilitat de manteniment
- Reutilització de codi

Intercepting Filter

El patró Intercepting Filter consisteix a crear filtres connectables per a processar serveis comuns d'una forma estàndard sense requerir canvis en el codi principal del processament de la petició. Els filtres intercepten les peticions entrants y les respostes sortints, permetent un pre y post-processament. Podem afegir i eliminar aquests filtres a discreció, sense necessitar canvis al codi existent.

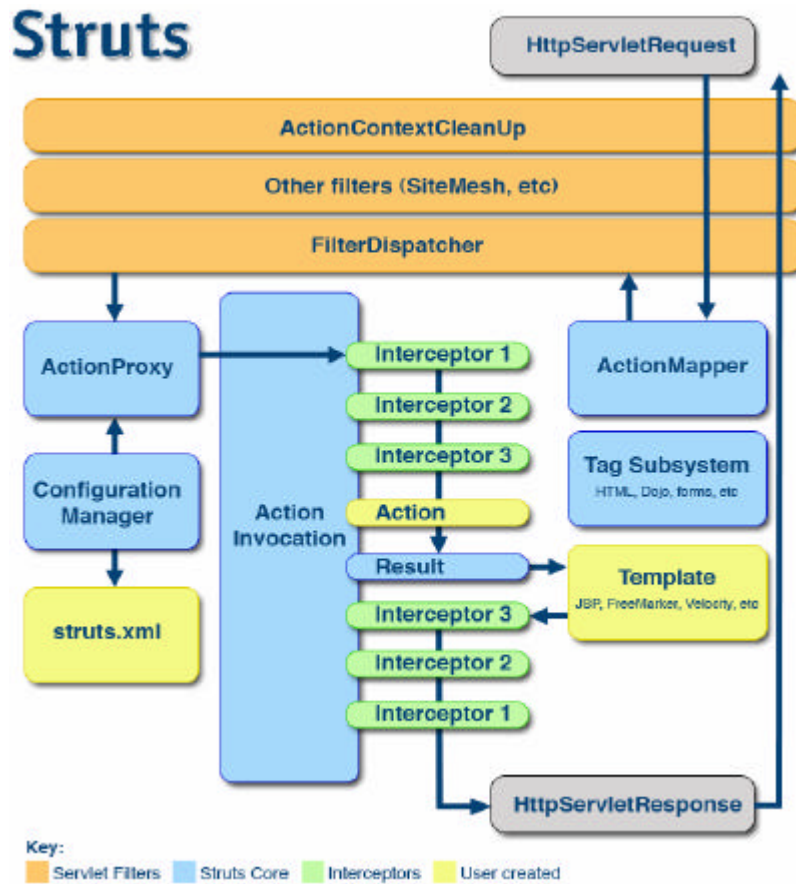
Els filtres que formaran part de l'aplicació del catàleg de gossos es configuraran al fitxer web.xml de la aplicació web i només hi haurà el filtre de seguretat. Si en un futur es vol treure la seguretat només s'haurà de deshabilitar el filtre i si es vol canviar la seguretat de l'aplicació només s'haurà de substituir el filtre per un altre.



Front Controller

El patró Front Controller consisteix a usar un controlador com a punt inicial de contacte per a gestionar les peticions. El controlador gestiona el control de peticions, delega el processament de negoci, controla l'elecció d'una vista apropiada, gestiona errors generals de l'aplicació i el controla la selecció d'estratègies de creació de contingut.

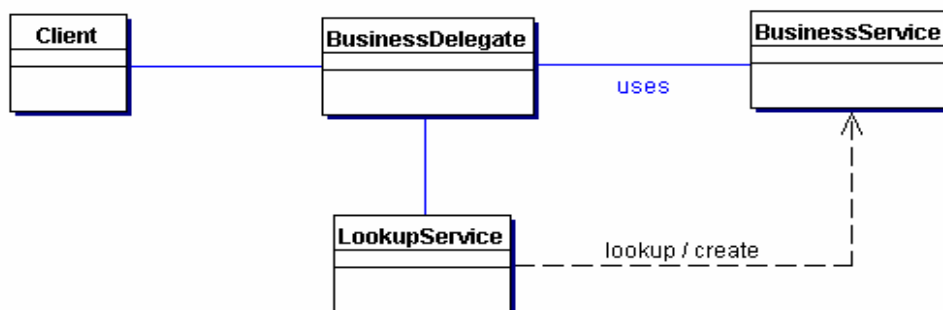
A l'aplicació del catàleg de gossos el controlador serà el que ofereix el marc de treball se Struts 2. El controlador està format per varis elements de Struts Core, pel fitxer de configuració struts.xml i per les accions de l'aplicació.



Business delegate

El patró Business delegate consisteix a reduir l'acoblament entre els clients de la capa de presentació i els serveis de negoci. El Business Delegate oculta els detalls de la implementació del servei de negoci.

A l'aplicació de catàleg de gossos s'implementarà el patró de Business Delegate utilitzant la inversió de control que ofereix el marc de treball de Spring framework, evitant així la creació d'objectes Business Delegate però augmentant el volum de fitxers de configuració.



Façade

El patró façade o facade serveix per a proveir d'una interfície unificada i senzilla que actuï d'intermediària entre un client i un una interfície o grup d'interfície més complex. Les façanes poden instanciar tan els serveis propis de l'aplicació com els serveis externs com per exemple els WebServices.

Application Service

El patró Application Service serveix per a dotar les aplicacions d'una interfície unificada i senzilla que faci d'intermediària entre les façanes i una interfície o grup d'interfícies més complexes com poden ser les lògiques de negoci i d'accés a dades.

Data Access Object

El patró Data Access Object consisteix en abstraure i encapsular tots els accessos a la font de dades. El DAO gestiona la connexió amb la font de dades per a obtenir i emmagatzemar les dades.

Marc de treball utilitzats

Tots els marcs de treball i eines escollides són de **codi obert i gratuïts**. Aquesta és una elecció que té els seus pros i els seus contres i no és l'objectiu d'aquest document argumentar a favor d'una opció o d'una altra.

Cal remarcar la tendència de tots els marcs de treball escollits a la simplicitat, cosa que queda reflectida en l'ús extens de **POJO's** en **Spring framework**, **Struts 2** i **JPA** (per defecte tots els marcs de treball de persistència tenen tendència a utilitzar POJO's) a diferència del **model EJB** (sobretot més en l'especificació 2 que en la 3) i **Struts1**.

Spring framework

Spring framework és un marc de treball **d'inversió de control**. Com a avantatge principal destaca la possibilitat d'utilitzar-lo com una alternativa al **model EJB** possibilitant així la no utilització d'un servidor d'aplicacions a favor d'un servidor menys complex i igualment potent com ara l'**Apache Tomcat**. Tot i així **Spring framework** també es pot utilitzar perfectament amb qualsevol servidor d'aplicacions.

Les funcionalitats clau que m'han fet triar aquest marc de treball són:

- **Inversió de control** mitjançant **injecció de dependència**. Ajuda a **reduir dependències de components** així com força a utilitzar algunes de les millors pràctiques com la **programació orientada a interfícies**
- **Alliberament de complexitat** dels **EJB** i de la obligatorietat d'utilitzar un servidor d'aplicacions
- Permet definir **transaccionalitat declarativa** i transaccionalitat a través de més d'una unitat transaccional (**JTA**)
- Integració fàcil amb **JPA** o qualsevol altre marc de treball de persistència tipus **Hibernate**, **iBatis**, **JDO**, etc ...
- Integració fàcil amb altres marcs de treball per a la implementació del patró de disseny MVC com **Struts 1**, **Struts 2**, **JSF**, **Tapestry** o **WebWork**, així com el propi mòdul de **Spring Web MVC**

S'ha de tenir en compte també desavantatges respecte el **model EJB**:

- Moltes arquitectures ja estan basades en el **model EJB** així que una migració requeriria un gran esforç per a aconseguir el mateix resultat, potser millorat o potser no. A més la corba d'aprenentatge és forta i es desaprofitaria el coneixement que molts arquitectes i programadors ja tenen si es deixa d'utilitzar el **model EJB**
- L'evolució del marc de treball depèn d'una empresa petita, **Spring Source**, en front del **model EJB** que depèn d'una empresa tant sòlida com **Sun Microsystems**
- El **model EJB** garanteix millor la **independència respecte a diferents venedors**, una de les màximes avantatges per les quals les empreses utilitzen el model **J2EE** per a realitzar les seves aplicacions

Struts 2

Struts 2 és un marc de treball que implementa el patró de disseny **MVC2**. A més ofereix les següents avantatges clau:

- Fàcil integració amb **Spring framework**
- **Autocomplimentació** de formularis
- Suport per a la **internacionalització**
- Utilització de **POJO's** per a la implementació de les **accions**
POJO vol dir Plain Old Java Objects, és a dir classes amb atributs, setters, getters i poca cosa més i que no implementen ni extenen de cap altre classe que no sigui Object
- Suport per a la realització de **validacions de formularis**
- **Llibreria de tags** pròpia per a la realització de les jsp's
- **Arquitectura superior** respecte la majoria de MVC comparables

Els desavantatges són:

- No disposa de cap plugin per a cap IDE
- Hi ha pocs desenvolupadors que el coneguin i l'utilitzin, tot i que cada cop n'hi ha més

L'elecció d'un marc de treball per a la implementació MVC d'una aplicació J2EE no és senzilla i és difícil d'argumentar. Jo m'he declinat per **Struts 2** perquè crec que és un bon marc de treball, però aquest motiu no és diferent en **Struts1**, **WebWork**, **AppFuse**, **Tapestry** i molts d'altres. El que en el meu cas personal ha fet declinar la balança és l'aposta de futur, és arquitectònicament més avançat (tots els experts opinen així) i sembla que la majoria d'arquitectes de referència estan apostant per ell i en uns anys és possible que acabi desbancant al líder indiscutible actual que és **Struts1**.

Nivell de persistència, JPA

Java persistence API (JPA) és la part de l'especificació J2EE 5 que fa referència a la persistència. És un marc de treball per a la implementació de la persistència.

Consisteix en:

- Una API definida al paquet **javax.persistence**
- El llenguatge propi de sentències (**JPQL**)
- Metadata de mapeig objecte/relacional

Les grans avantatges són:

- **Independència de l'entorn** del Sistema Operatiu i de la Base de dades o suport de persistència
- **Llenguatge únic** de sentències per a qualsevol entorn, no s'ha de tenir en compte les diferències del llenguatge SQL per a diferents Base de dades
- **Lliure elecció del venedor** podent canviar-lo sense afectar a la implementació, fins i tot en alguns servidors d'aplicacions es pot canviar el venedor sense aturar les aplicacions
- Fàcil **integració amb Spring**
- Eines Case com **Dalí** per a l'Eclipse creades per a facilitar el desenvolupament

Gràcies a les característiques de JPA la tasca de canviar de Base de Dades i de venedor de persistència (Hibernate, GlassFish, TopLink, etc ...) resulta molt menys complex permetent així ser independent i poder variar en funció de les necessitats, ja siguin econòmiques o de rendiment.

Eines

Ajaxtags

L'eina d'**Ajaxtags** és una llibreria de tags que dóna utilitats **AJAX** com ara:

- Autocompletació d'un element de text
- Recarga de contingut de pestanyes
- Recarga de contingut d'elements Html com ara els divs
- Estil Portlet
- Etiquetes dinàmiques
- Actualització de camps

I més utilitats i tot això sense recarregar la pàgina.

Les avantatges d'utilitzar AJAX en una aplicació web són:

- **Actualització de continguts** sense necessitat de recarregar la pàgina
- **Disminució del trànsit** de dades a la xarxa
- **Percepció de més interactivitat** per part de l'usuari

Les desavantatges existeixen i poden arribar a ser greus sobretot en quant a l'accessibilitat, ja que una pàgina amb AJAX no sobrepassaria la prova de la A.

En aquest projecte he considerat que l'accessibilitat no era important i si que ho era la percepció d'interactivitat i la rapidesa en la carrega de les pàgines.

Una opció, degut a la dificultat de crear pàgines accessibles és el fet de crear aplicacions paral·leles per a que puguin ser accessibles, ja que el cost de realitzar aplicacions J2EE accessibles, usables i boniques en molts casos supera el cost de crear dos aplicacions en paral·lel, una d'usable i bonica i l'altre accessible.

Log4j

Log4j és un marc de treball per a realitzar traces d'aplicacions. És un Standard de facto i permet molta flexibilitat.

Disseny de l'aplicació

Model de dades

Gos

Column Name	Datatype	NOT NULL	AUTO INC
ID	INTEGER	✓	✓
racalD	INTEGER	✓	
descripcio	VARCHAR(1000)	✓	
fotografia	VARCHAR(45)	✓	
nom	VARCHAR(45)	✓	

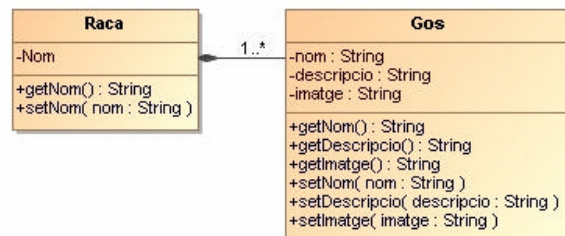
racalD és una clau forana que fa referència al camp ID de la taula Raca

Raca

Column Name	Datatype	NOT NULL	AUTO INC
ID	INTEGER	✓	✓
descripcio	VARCHAR(45)	✓	

Entitats

Paquet org.uoc.tfc_mnm.integracio.entitats



Gos

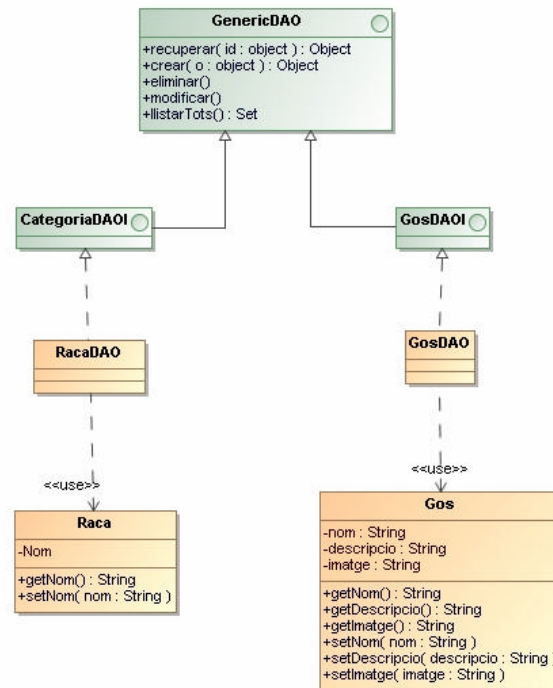
La classe entitat Gos contindrà la informació de la fitxa d'un gos

Raca

La classe entitat Raca contindrà la informació de la fitxa d'una raça

DAO's

Paquet org.uoc.tfc_mnm.integracio.dao



GenericDAO

Interfície que defineix les operacions bàsiques que ha d'implementar un DAO

GosDAOI

Interfície que defineix les operacions que ha d'implementar el DAO que accedeix a la informació de la entitat Gos

RacaDAOI

Interfície que defineix les operacions que ha d'implementar el DAO que accedeix a la informació de la entitat Raca

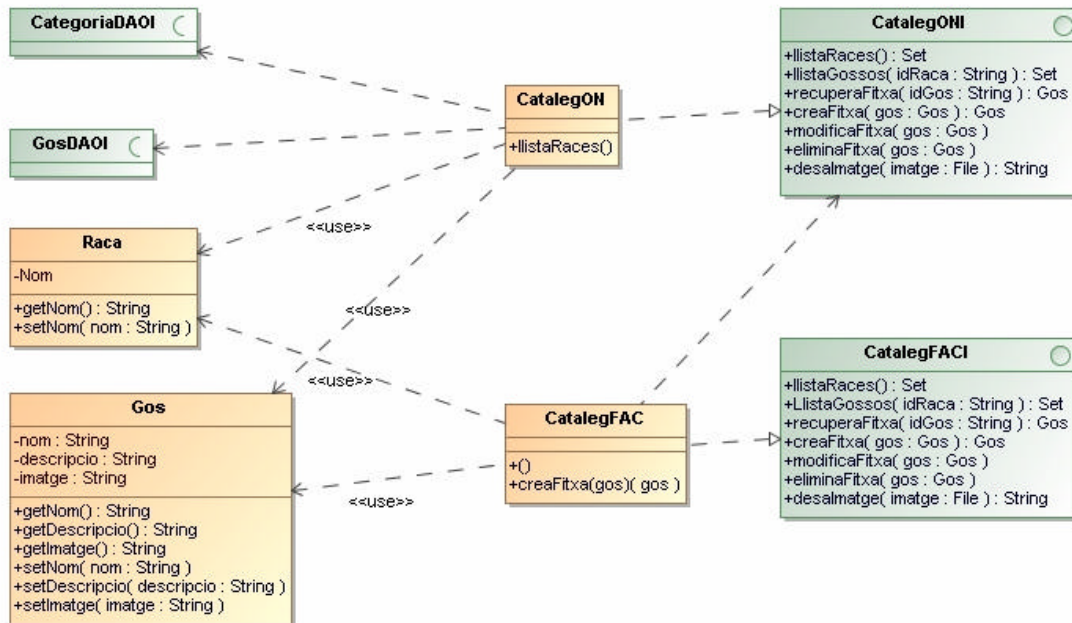
GosDAO

Implementació del DAO de la entitat Gos

RacaDAO

Implementació del DAO de la entitat Raca

Objectes de negoci i façanes de l'aplicació



Paquet org.uoc.tfc_mnm.negoci.on

CatalogONI

Interfície que defineix les operacions que ha d'implementar l'objecte de negoci de l'aplicació

CatalogON

Implementació de l'objecte de negoci de l'aplicació

Paquet org.uoc.tfc_mnm.negoci.facana

CatalogFacI

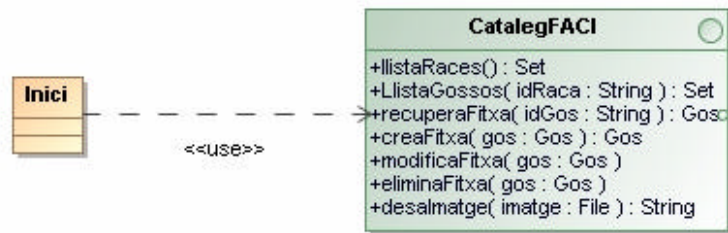
Interfície que defineix les operacions que ha d'implementar la façana de l'aplicació

CatalogFac

Implementació de la façana de l'aplicació

Accions

Paquet org.uoc.tfc_mnm.vista.accions



Inici

Aquesta acció s'encarregarà de recuperar la llista de races

LlistaGossos

Aquesta acció s'encarregarà de recuperar la llista de gossos

MostraFitxa

Aquesta acció s'encarregarà de recuperar la informació de la fitxa d'un gos

EditaFitxaAP

Aquesta acció s'encarregarà de modificar la fitxa d'un gos

CreaFitxaAP

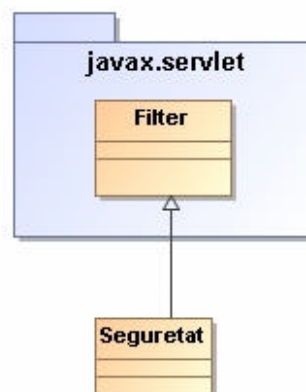
Aquesta acció s'encarregarà d'eliminar la fitxa d'un gos

EliminaFitxaAP

Aquesta acció s'encarregarà d'eliminar la fitxa d'un gos

Filtres

Paquet org.uoc.tfc_mnm.vista.filtres



Seguretat

Filtre que comprovarà que l'usuari hagi introduït la contrasenya correcta quan intenti entrar a alguna pàgina de l'aplicació privada

JSP's

Per a realitzar totes les pantalles s'utilitzaran els tags d'Struts2 i els d'Ajaxtags

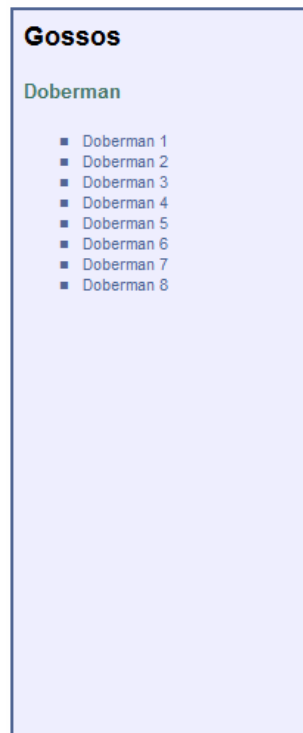
iniciUsuari.jsp

La pàgina estarà formada per 3 elements div.

El primer hi haurà la llistar de les races i els altres només hi apareixerà el nom descriptiu del contingut que hi tindran. Les races hauran d'enllaçar amb l'acció **llistaGossos** que recarregarà el contingut del div de Gossos.

Races	Gossos	Fitxa
<ul style="list-style-type: none">■ Doberman■ Pastor Alemà■ Golden Retriever■ Cocker■ Galg■ Boxer■ Caniche■ Yorkshire Terrier		

gossosUsuari.jsp



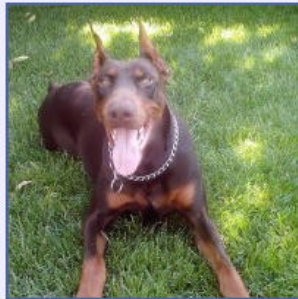
La pàgina contindrà el codi html que hi haurà dins del div de gossos.
Es mostrarà el títol del div, *Gossos*, i el nom de la raça seleccionada i la llista de tots els gossos de la raça seleccionada. Els gossos hauran d'enllaçar amb l'acció **mostraFitxa** que recarregarà el contingut del div de la fitxa.

fitxaUsuari.jsp

Fitxa

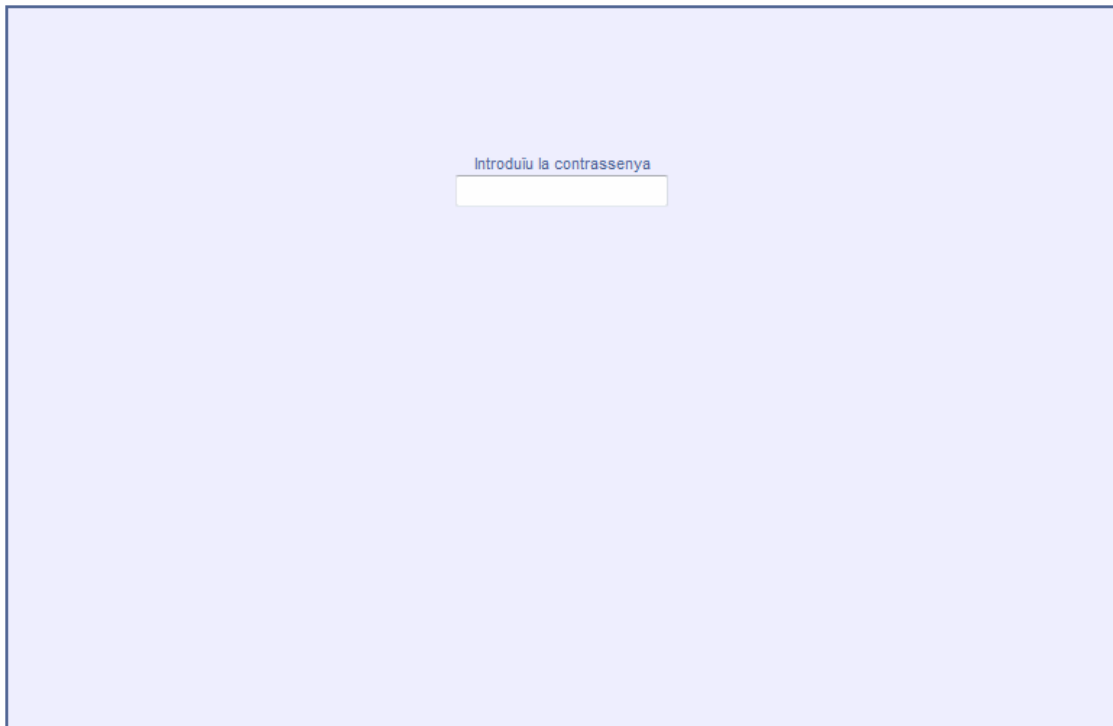
Doberman 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent gravida adipiscing libero. Integer viverra augue id lectus egestas laoreet. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris libero enim, mollis vitae, lobortis eu, iaculis et, dui. Sed euismod lorem sit amet libero aliquet pellentesque. Proin vitae augue. Suspendisse condimentum eros



La pàgina contindrà el codi html que hi haurà dins del div de fitxa.
Es mostrarà el títol del div, *Fitxa*, el nom del gos seleccionat, la descripció i la imatge.

introContrassenya.jsp

A screenshot of a web page with a light blue background. In the center, there is a text label "Introduiu la contrassenya" above a white rectangular input field. The entire content is enclosed in a thin blue border.

La pàgina estarà formada per un div que contindrà un text que digui "introduiu la contrassenya" i al clicar l'"intro" s'executarà el submit del formulari que tindrà associada l'acció **AdminInici**.

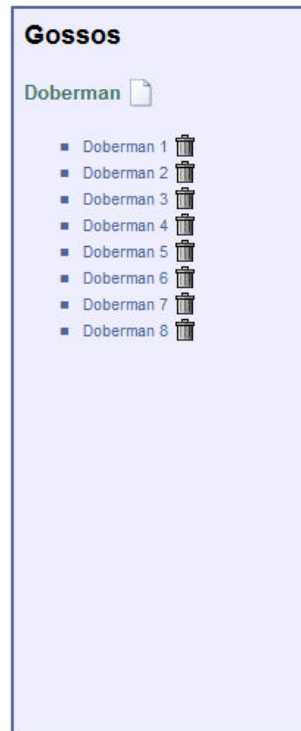
iniciAdmin.jsp

Races	Gossos	Fitxa
<ul style="list-style-type: none">■ Doberman■ Pastor Alemà■ Golden Retriever■ Cocker■ Galg■ Boxer■ Caniche■ Yorkshire Terrier		

La pàgina estarà formada per 3 elements div.

El primer hi haurà la llistar de les races i els altres només hi apareixerà el nom descriptiu del contingut que hi tindran. Les races hauran d'enllaçar amb l'acció **AdminLlistaGossos** que recarregarà el contingut del div de Gossos.

gossosAdmin.jsp



La pàgina contindrà el codi html que hi haurà dins del div de gossos.

Es mostrarà el títol del div, **Gossos**, i el nom de la raça seleccionada i la llista de tots els gossos de la raça seleccionada.

Els gossos hauran d'enllaçar amb l'acció **AdminEditaFitxa** que recarregarà el contingut del div de la fitxa.

Al costat de cada gos també hi haurà una imatge d'una escombraria que enllaçarà amb l'acció d'**AdminEliminaFitxa**.

Al costat del nom de la raça seleccionada hi apareixerà una imatge d'un full en blanc que enllaçarà amb l'acció de **AdminCrearFitxa**.

editaFitxaAdmin.jsp

Fitxa (Edició)

Nom

Descripció:

Imatge



La pàgina contindrà el codi html que hi haurà dins del div de l'edició de la fitxa.
Es mostrarà el títol del div, *Fitxa (Edició)*, el nom del gos seleccionat, la descripció i la imatge.
Permetrà editar tots els atributs.
El formulari de la pàgina tindrà associada l'acció **AdminEditaFitxaFi**.

creaFitxaAdmin.jsp

Fitxa (Creació)

Nom

Descripció:

imatge
 Navega...

crear

La pàgina contindrà el codi html que hi haurà dins del div de l'edició de la fitxa.

Es mostrarà el títol del div, *Fitxa (Creació)*, el nom del gos, la descripció i la imatge. Permetrà editar tots els atributs.

El formulari de la pàgina tindrà associada l'acció **AdminCreaFitxaFi**.

Configuració web

En el fitxer de configuració web **web.xml** es configuren els filtres, concretament el de seguretat, el mapeig de les peticions per a que les tracti totes el controlador de Struts2 i mitjançant dos listeners es configura i s'inicialitza el context de l'aplicació d'Spring i l'eina de log4j.

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5" >

  <display-name>TFC_web</display-name>

  <welcome-file-list>
    <welcome-file>inici.action</welcome-file>
  </welcome-file-list>

  <!-- ### Configuració Seguretat ###
  -->
  <!-- Filtre d'autenticació
  -->
  <filter>
    <filter-name>FiltreAutenticacio</filter-name>
    <filter-class>
      org.uoc.tfc.web.filter.FiltreAutenticacio
    </filter-class>
  </filter>

  <!-- Mapeig per a les peticions d'entrada a l'aplicació d'administració -->
  <filter-mapping>
    <filter-name>FiltreAutenticacio</filter-name>
    <url-pattern>/adminInici.action</url-pattern>
  </filter-mapping>
  <!-- ### FI Configuració per a Struts 2 ### -->

  <!-- ### Configuració Struts 2 ###
  -->
  <!-- Filtre de struts 2
  -->
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.FilterDispatcher
    </filter-class>
  </filter>

  <!-- Mapeig per a que qualsevol petició la tracti el filtre de Struts 2 -->
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <!-- ### FI Configuració per a Struts 2 ### -->

  <!-- Configuració log4j utilitzant Spring -->
  <context-param>
    <param-name>log4jConfigLocation</param-name>
    <param-value>
      /WEB-INF/classes/log4j/log4j.properties
    </param-value>
  </context-param>
  <context-param>
    <param-name>log4jRefreshInterval</param-name>
    <param-value>1000</param-value>
  </context-param>
```



```
<listener>
  <listener-class>
    org.springframework.web.util.Log4jConfigListener
  </listener-class>
</listener>

<!-- FI Configuració log4j -->

<!-- Declaració del listener de Spring per a que iniciï els beans declarats
en el contexte de l'aplicació (applicationContext.xml) -->
<listener>
  <listener-class>
    org.springframework.web.context.ContextLoaderListener
  </listener-class>
</listener>

</web-app>
```

Configuració marc de treball Spring

Al fitxer **applicationContext.xml** és on es defineixen tots els beans que es crearan en el context de l'aplicació.

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    fitxer de configuració d'Spring.
    Aquí es declaren tots el 'Beans' que s'inicialitzaran a l'inici de l'aplicació.
    En aquest fitxer no es declararà explícitament cap 'Bean' sinó que s'inclouran
    fitxers que tindran declarants els 'Beans' agrupats per conceptes:

    - persistència
    - transaccionabilitat
    - DAO's
    - serveis d'aplicació
    - serveis externs
    - façanes d'aplicació
    - accions d'aplicació
    - seguretat
    - etc ...

    Autor: Marçal Nebot
-->
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd" >

    <import resource="classpath:spring/persistencia.xml" />
    <import resource="classpath:spring/daos.xml" />
    <import resource="classpath:spring/serveisAplicacio.xml" />
    <import resource="classpath:spring/facanes.xml" />
    <import resource="classpath:spring/transaccionalitat.xml" />
    <import resource="classpath:spring/accions.xml" />
    <import resource="classpath:spring/seguretat.xml" />
</beans>
```

En el fitxer applicationContext.xml directament no es crea cap bean, sinó que és als fitxers on fa referència. Per exemple en el fitxer persistencia.xml:

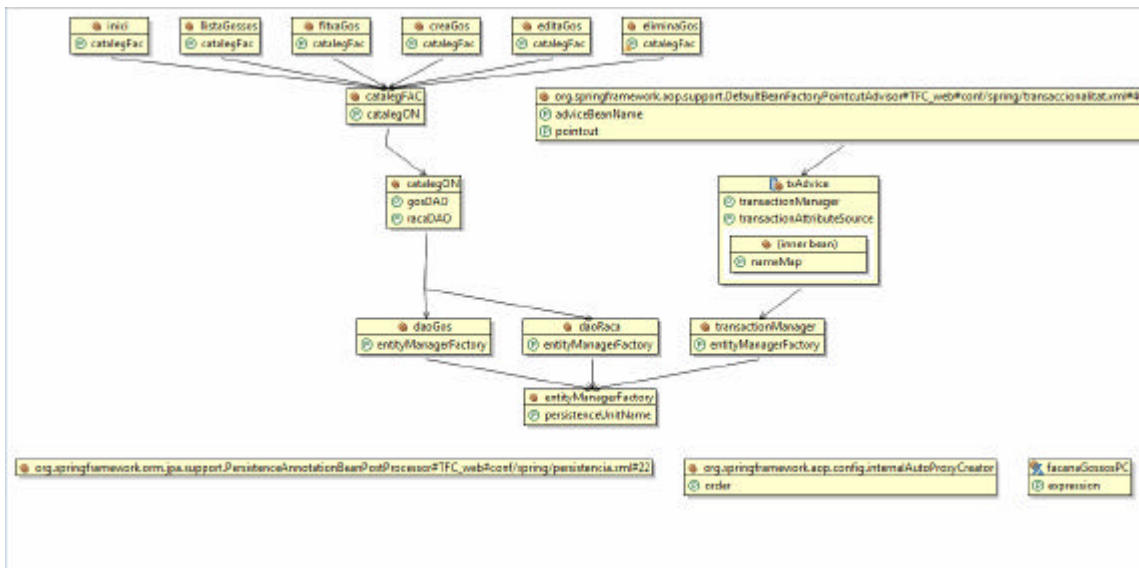
```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    fitxer de configuració d'Spring.
    Aquí es declaren tots els 'Beans' de persistència

    Autor: Marçal Nebot
-->
<beans xmlns=
       xmlns:xsi=
       xsi:schemaLocation=
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd" >

    <!-- Suport per a anotacions de persistència JPA -->
    <bean
class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor" />

    <!-- Gestor d'entitats -->
    <bean id="entityManagerFactory"
class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
        <property name="persistenceUnitName" value="TFC_integracio" />
    </bean>
</beans>
```

Els beans que es creen a l'aplicació són els següents:



Configuració marc de treball Struts

El marc de treball d'Struts es configura des del fitxer **struts.xml**. En aquest fitxer es declaren les accions de l'aplicació i és on es pot modificar alguns paràmetres del marc de treball com ara la factoria d'objectes (en aquest cas Spring) o el nom del recurs d'internacionalització (els fitxers amb els literals de la aplicació amb els diferents idiomes).

struts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

    fitxer de configuració de Struts.
    Aquí es declaran totes les accions de l'aplicació i es configura el framework
    d'struts.

    Autor: Marçal Nebot
-->
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "struts-2.0.dtd">
<struts>
    <!-- Constants de configuració struts -->
    <constant name="struts.objectFactory" value="spring" />
    <constant name="struts.devMode" value="true" />
    <constant name="struts.i18n.reload" value="true" />
    <constant name="struts.custom.i18n.resources" value="messages" />

    <package name="tfc" extends="struts-default">
        <!-- -->
        <!-- Accions de l'aplicació pública -->
        <!-- -->
        <action name="inici" class="inici">
            <result name="success">
                /inici.jsp
            </result>
        </action>
        ...
        ...
        ...

        <!-- -->
        <!-- Accions de l'aplicació privada -->
        <!-- -->
        <action name="adminInici" class="inici">
            <result name="success">
                /adminInici.jsp
            </result>
        </action>
        ...
        ...
        ...

    </package>
</struts>
```

Configuració font de dades

El fitxer **context.xml** és un fitxer que es configura específicament pel servidor Tomcat i ha d'anar a la carpeta META-INF de l'aplicació web. En aquest fitxer es configuren els diferents atributs del context de l'aplicació web, en concret per a la aplicació de catàleg de gossos es defineix el recurs d'accés a la font de dades.

context.xml

```
<Context docBase="/TFC_web" path="/TFC_web" reloadable="true" >
<Resource
  auth="Container"
  driverClassName="com.mysql.jdbc.Driver"
  maxActive="10"
  maxIdle="2"
  maxWait="5000"
  name="jdbc/tfc_mnm"
  type="javax.sql.DataSource"
  url="jdbc:mysql://127.0.0.1:3306/TFC"
  username="root"
  password="root"
/>
</Context>
```

Configuració persistència

El fitxer de configuració de persistència JPA **persistence.xml** configura la font de dades de l'aplicació, el proveïdor JPA i les classes que implementen la persistència.

persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
  <persistence-unit name="TFC_integracio" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>

    <non-jta-data-source>java:comp/env/jdbc/tfc_mnm</non-jta-data-source>

    <class>org.uoc.tfc_mnm.integracio.entitats.Gos</class>
    <class>org.uoc.tfc_mnm.integracio.entitats.Raca</class>

    <properties>
      <property name="dialect" value="org.hibernate.dialect.MySQLDialect"/>
      <property name="hibernate.show_sql" value="true" />
    </properties>

  </persistence-unit>
</persistence>
```

Resum de components

La base de dades

La base de dades és un servidor MySQLSever 5 i conté les dades de les entitats de l'aplicació.

Marc de treball

Els marcs de treball utilitzats, Spring i Struts² estan dins l'aplicació web. Estan compostats per les seves llibreries, les seves dependències i pels seus fitxers de configuració.

Component d'Integració

El component d'integració és una llibreria que conté les entitats i els DAO de l'aplicació i que està dins l'aplicació web. En aquest component es configura l'accés a la font de dades. Aquest component depèn de les llibreries que implementen la especificació JPA, que per a aquesta aplicació s'ha escollit el proveïdor Hibernate.

Component de la lògica de negoci

La lògica de negoci depèn exclusivament del component d'integració i implementa la lògica de negoci. Ofereix una interfície de façana per a que qualsevol aplicació la pugui utilitzar.

Component AJAX

El component d'AJAX és una utilitat de la vista per a fer que la aplicació sigui més dinàmica. És una llibreria de tags per a ser utilitzada a les JSP's.

Aplicació web

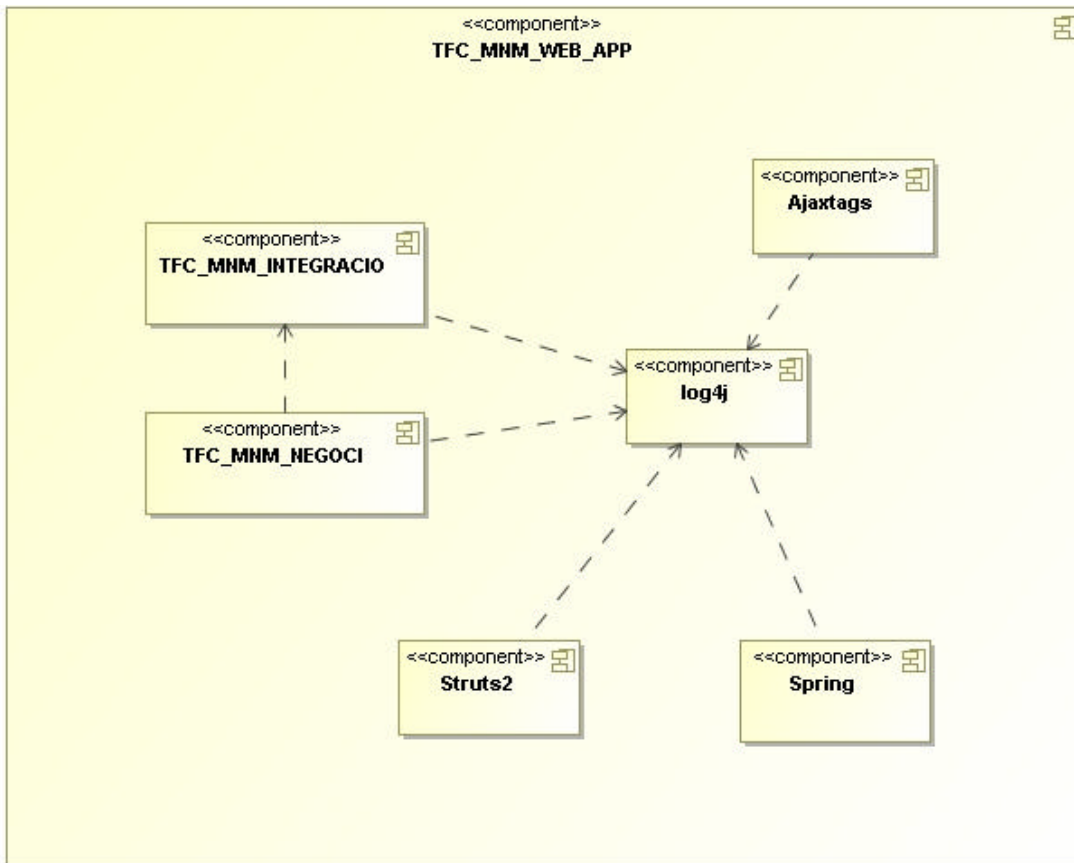
L'aplicació web utilitza directament el component de la lògica de negoci mitjançant la façana i a través d'aquest també utilitza el component d'integració, però no utilitza mai directament aquest component.

S'ajuda dels marcs de treball per la construcció de l'aplicació en tasques com la inicialització de l'aplicació, la configuració del controlador, l'autompliment de formularis, la creació de les JSP's, el tractament d'errors, etc ...

L'aplicació web conté:

- els components dels qual depèn en forma de llibreries
- les accions
- el filtre de seguretat
- les pàgines jsp

- els fitxers de configuració
 - o de l'aplicació web, el web.xml
 - o de Struts, l'.struts.xml
 - o de Spring, l'applicationContext.xml
 - o de logs, el log4j.xml
- els fitxers estàtics com les imatges, els fitxers de JavaScript i els fitxers d'estil.



Conclusions

En la realització d'aquest treball de final de carrera s'ha acomplert l'objectiu principal de crear una aplicació J2EE senzilla. El objectius específics d'aquesta aplicació, utilitzar els patrons de treball més usuals de J2EE, uns marcs de treball avançats, l'especificació JPA per a la persistència de dades i una eina per a l'ús d'AJAX, també s'han aconseguit.

Un altre fita assolida ha estat la creació d'una aplicació base per a crear aplicacions més complexes.

Per a aconseguir-ho he hagut d'utilitzar la metodologia de cicle en cascada, i la planificació que vaig realitzar en un principi he sigut capaç de complir-la. Crec que el factor clau ha sigut no voler fer més del que era capaç de realitzar. Per tant puc dir que no m'he desviat en la planificació, fet que en el món laboral és molt important i sento que en aquest sentit he tingut un gran èxit.

La utilització dels patrons de disseny ha sigut de gran ajuda per a crear una aplicació ben ordenada, fet que crec que he aconseguit i que farà que pugui ser una aplicació que serveixi com a punt d'arrencada per a aplicacions més grans i complexes.

La utilització de marcs de treball fa que l'aplicació sigui més complexa. Per a l'aplicació que he creat, el que he guanyat utilitzant els marcs de treball ho he perdut en quan a temps i complexitat, però potencialment l'aplicació ha crescut en possibilitats. La conclusió és que els marcs de treball no són útils per a aplicacions petites, però surten a compte per a aplicacions mitjanes i són imprescindibles per a aplicacions grans. Com que uns dels meus objectius era crear una aplicació que servís de punt d'arrencada per a aplicacions més grans, la utilització dels marcs de treball cobra sentit.

Glossari

TFC

Treball final de carrera

J2EE

Java 2 Enterprise Edition: especificació java per a la creació d'aplicacions d'empresa creada per Sun Microsystems

Marc de treball

O framework amb anglès, conjunt d'eines que ajuden a desenvolupar una aplicació

MVC

Model-Viewer-Controller, patró de disseny

Spring

Marc de treball d'inversió de control

Struts²

Marc de treball que implementa el model MVC

AJAX

Asynchronous JavaScript and XML, grup de tecnologies per a crear pàgines web interactives

JSP

Java Server Pages, pàgines java que s'executen en un servidor

JPA

Java Persistence API, part de l'especificació J2EE 5 que fa referència a la persistència

POJO

Plain Old Java Object

DAO

Data Acces Object

Bibliografia

Patrons de disseny

[Patrons J2EE](#)

[Wikipedia: Patró MVC](#)

[Wikipedia: POJO](#)

Marc de treball i eines

[Spring Framework](#)

[Spring Source](#)

[Wikipedia: Spring Framework](#)

[Spring vs EJB 3.0 per O'Reilly](#)

[Spring ide](#)

[Struts²](#)

[Wikipedia:MVC java web application frameworks](#)

[Tendència de demanda de feina](#)

[Referència JPA del manual de Sun](#)

[Wikipedia: JPA](#)

[Dali](#)

[Hibernate EntityManager](#)

[AjaxTags](#)

[Wikipedia: Ajax programming](#)

[log4j](#)

[Wikipedia: log4j](#)

Servidors, IDE's i utilitats

[Apache Tomcat](#)

[Configuració font de dades a l'Apache Tomcat amb MySql](#)

[MySql](#)

[Eclipse Europa](#)

[SpringIDE](#)