

UAB

**Universitat Autònoma
de Barcelona**

SIMVAC

Desenvolupament d'un simulador d'una granja
de vaques

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes
realitzat per
Sergi Rodríguez Llavet
i dirigit per
Gonzalo Vera Rodríguez

Escola d'Enginyeria
Sabadell, Setembre de 2010

El sotasignat, *Gonzalo Vera Rodríguez*,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present
memòria
ha estat realitzat sota la seva direcció
per en *Sergi Rodríguez Llavet*

I per a que consti firma la present.
Sabadell, *mes* de *any*

Signat: *Gonzalo Vera Rodríguez*
director/a

RESUM

En aquest projecte s'ha desenvolupat una aplicació que permet simular la producció de llet en una granja de vaques. El projecte està enfocat com a una eina d'ensenyament per a la facultat de veterinària de la UAB, per facilitar als seus alumnes una forma senzilla i eficaç de realitzar les pràctiques sobre la producció de llet d'una granja de vaques.

Per tant l'objectiu principal del projecte serà la implementació d'un programari que permeti aquesta simulació. Aquesta aplicació té com a base un servidor Web format per el servidor Apache HTTP, el servidor Apache Tomcat i el sistema gestor de base de dades MySQL. A partir d'aquests i mitjançant l'entorn de programació Eclipse Galileo s'ha implementat un simulador dins d'un Servlet de Java, que permet als alumnes un accés remot al simulador per tal que puguin realitzar les pràctiques des de qualsevol ordinador amb connexió a Internet.

La descripció general del funcionament del simulador es pot resumir de la següent manera. Cada alumne s'inscriurà a un curs i després de la verificació del professor del curs, li assignarà una granja a l'alumne. L'alumne podrà interactuar amb la granja i amb cadascuna de les vaques que la componen, així com realitzar algunes accions sobre aquestes i veure com afecten a la producció de llet de la vaca. També permet veure l'evolució de la vaca de molts dies amb poc temps, ja que és l'alumne el que decideix quan l'avança un dia en el simulador.

Quan el professor ho desitgi podrà realitzar una avaluació sobre la granja d'un alumne i comprovar quines accions ha realitzat amb èxit sobre les totals i comprovar en tot moment que l'estat de les vaques sigui el correcte.

Per tant en aquesta memòria veurem tots els passos que s'han seguit pel seu desenvolupament fins a la seva finalització.

ÍNDIX DE CONTINGUTS

	Pàg.
RESUM.....	3
ÍNDIX DE CONTINGUTS.....	4
ÍNDIX DE IL·LUSTRACIONS.....	5
Capítol 1 - INTRODUCCIÓ.....	6
1.1 PROBLEMES I ANTECEDENTS.....	6
1.2 CONTEXT.....	7
1.3 MOTIVACIÓ.....	7
1.4 RESUM.....	8
1.5 ESTRUCTURA DE LA MEMÒRIA.....	8
Capítol 2 - ESTUDI DE VIABILITAT.....	10
2.1 INTRODUCCIÓ.....	10
2.2 ESTUDI DE LA SITUACIÓ ACTUAL.....	12
2.3 REQUERIMENTS DEL PROJECTE.....	12
2.4 ALTERNATIVES I SELECCIÓ DE LA SOLUCIÓ.....	14
2.5 AVALUACIÓ TECNOLÒGICA DEL PROJECTE.....	16
2.6 PLANIFICACIÓ DEL PROJECTE.....	17
2.7 AVALUACIÓ DE RISCOS.....	19
2.8 PRESSUPOST.....	22
2.9 CONCLUSIONS.....	25
Capítol 3 - ANÀLISI.....	26
3.1 REQUERIMENTS DEL PROJECTE.....	26
3.2 SIMULACIÓ D'UNA VACA.....	27
3.3 MARC TECNOLÒGIC.....	29
3.4 DIAGRAMES DE CASOS D'ÚS.....	41
Capítol 4 – DISSENY.....	47
4.1 DISSENY DE LA INFRAESTRUCTURA BASE.....	48
4.2 SELECCIÓ DELS LENGUATGES I IDE.....	49
4.3 DISSENY DE LES PÀGINES WEB.....	50
4.4 DIAGRAMA D'ENTITAT – RELACIÓ.....	51
4.5 DIAGRAMA DE CLASSES.....	52
4.6 PLANIFICACIÓ AJUSTADA.....	53
Capítol 5 - IMPLEMENTACIÓ.....	56
5.1 CREACIÓ I CONFIGURACIÓ DE LA INFRAESTRUCTURA.....	56
5.2 CREACIÓ DEL SERVLET BASE.....	59
5.3 CREACIÓ DE LES BASES DE DADES.....	60
5.4 DESCRIPCIÓ DE LES CLASSES DEL SERVLET.....	63
5.5 CREACIÓ DEL SERVLET DE GRÀFIQUES.....	65
5.6 CREACIÓ DE LA INTERFÍCIE WEB.....	66
Capítol 6 - PROVES DEL PROJECTE.....	80
6.1 PROVES EN LA INFRAESTRUCTURA.....	80
6.2 PROVES EN EL SIMULADOR.....	80
6.3 PROVES EN EL FUNCIONAMENT DE LES PÀGINES.....	81
Capítol 7 - CONCLUSIONS.....	83
BIBLIOGRAFIA.....	86

ÍNDIX DE IL·LUSTRACIONS

	Pàg.
Figura 1 - Interacció de l'alumne amb la granja	8
Figura 2 - Esquema del sistema	16
Figura 3 - Diagrama de Gantt del projecte	19
Figura 4 - Cicle d'estats d'una vaca	28
Figura 5 - Cas d'ús accedir.....	42
Figura 6 - Casos d'ús administrador i professor	45
Figura 7 - Casos d'ús alumne.....	46
Figura 8 - Aplicació.....	47
Figura 9 - Disseny general de les pàgines Web.....	50
Figura 10 - Diagrama d'entitat relació	51
Figura 11 - Diagrama de classes.....	52
Figura 12 - Redistribució de tasques	54
Figura 13 - Diagrama de Gantt ajustat.....	55
Figura 14 - Instal·lació de XAMPP.....	57
Figura 15 - Instal·lació de Tomcat	58
Figura 16 - Entorn PhpMyAdmin.....	59
Figura 17 - Diagrama de la base de dades	61
Figura 18 - Pàgina d'inici	67
Figura 19 - Formulari de registre.....	68
Figura 20 - Pàgina de la granja.....	69
Figura 21 - Pàgina de vaca	71
Figura 22 - Pàgina de configuració.....	72
Figura 23 - Formulari de nou curs	72
Figura 24 - Pàgina de gestió de còpies de seguretat	73
Figura 25 - Formulari de modificació dels paràmetres per defecte del simulador.....	73
Figura 26 - Pàgina d'administració d'un curs	74
Figura 27 - Modificació o eliminació de cursos	74
Figura 28 - Formulari d'un nou usuari	75
Figura 29 - Modificació o eliminació d'usuaris	76
Figura 30 - Formulari d'una nova granja	77
Figura 31 - Modificació o eliminació de granges.....	78
Figura 32 - Llistat de granges d'un curs.....	79

Capítol 1 - INTRODUCCIÓ

El projecte que es presenta en el següent document com a treball de fi de carrera consisteix en una aplicació Web, anomenada SimVac, que té com a objectiu principal la simulació de producció de llet d'una granja de vaques.

El client demandant d'aquesta aplicació és la facultat de veterinària de la Universitat Autònoma de Barcelona, i en particular el Catedràtic del Departament de Ciència Animal i dels Aliments, el Dr. i professor Sergio Calsamiglia Blancafort, qui actualment no disposa de cap plataforma informàtica que doni suport a les lliçons impartides a les classes teòriques.

SimVac permet a l'usuari una interacció amb la granja el màxim d'ajustada a la realitat que ha estat possible, i que alhora també dóna uns resultats força reals.

A continuació s'explicarà més detalladament quins han estat els problemes que han decidit a la realització de l'aplicació, el context on s'ha d'encabir i les motivacions que ens han portat a escollir aquest projecte.

1.1 PROBLEMES I ANTECEDENTS

Un dels principals objectius del projecte serà el de solucionar un seguit de problemes i dificultats que es presenten quan els professors de veterinària volen que els seus alumnes posin en pràctica tots els coneixements que han adquirit en les classes teòriques. Per tal de facilitar aquesta tasca el Dr. Calsamiglia va proposar la realització d'una aplicació informàtica que li permetés una interacció alumne/granja que no existia en aquell moment. Per tant, la nostra feina en aquest projecte ha estat la de plasmar les idees del professor en una aplicació que li permeti solucionar aquest problema que té en l'actualitat.

Una altre de les dificultats existents és que l'alumne disposa un temps de pràctica finit, quan finalitza els exercicis proposats pel professor té el problema de no poder seguir practicant si no disposa d'una vaca, i per tant la solució que proposem a aquest problema serà la facilitat d'accés a la aplicació via Web, i alhora, la possibilitat d'experimentar a diari amb la granja.

Un cop hem vist alguns dels problemes i antecedents que ens han conduït a la realització del projecte avaluarem en quin context ens trobem en l'actualitat i les possibles solucions que hem trobat.

1.2 CONTEXT

Si analitzem el context actual, aquests exercicis pràctics que efectuen els alumnes, es realitzen mitjançant suports com el paper o els fulls de càlcul de l'ordinador, i per tant veiem que resulta molt complex que l'alumne tingui una pràctica suficient, ja que no té una interacció amb l'espècie diàriament i tampoc pot observar tots els canvis que provoquen les seves decisions.

Per altre banda, aquesta pràctica també es realitza mitjançant sortides i estades a algunes granges, però això ocasiona altres dificultats com són una despesa en estància i transport, i tampoc soluciona plenament el problema d'interacció amb les vaques, ja que en moltes ocasions l'alumne no pot prendre les seves pròpies decisions ja que això podria suposar un problema per la granja en qüestió.

Podem observar que la situació actual en que es realitzen aquestes pràctiques no és la mateixa que en altres titulacions, com ara la informàtica on gran part dels coneixements teòrics es poden posar en pràctica i d'una manera fàcil i continua.

1.3 MOTIVACIÓ

Després d'analitzar la situació actual i els problemes que existeixen veiem una motivació suficient per tal de millorar-la. En primer lloc la motivació per construir una eina útil per als estudiants de veterinària i alhora estar donant un suport al professor que imparteix la matèria.

A més de la motivació de solucionar els problemes existents, també ha estat important la temàtica del projecte, ja que al ser un tema diferent als que normalment es realitzen en els títols d'informàtica, ha suposat un valor afegit per la seva elecció.

Si analitzem els beneficis i inconvenients de la seva realització creiem que són motius suficients per decantar la nostre decisió.

- ✓ Reducció de despeses en trajectes i estància.
- ✓ Qualitat de la pràctica dels coneixements.
- ✓ Possibilitat de veure l'evolució en pocs dies.
- ✓ Accés a l'aplicació des de qualsevol amb connexió a Internet.
- ✗ Període de formació als usuaris.

1.4 RESUM

Un cop analitzats els problemes, el context i les motivacions, creiem que és necessari i beneficiós la realització d'una aplicació que permeti la millora de la pràctica dels coneixements teòrics impartits a classe i que no suposi una despesa econòmica per l'alumne i que alhora sigui de fàcil accés en tot moment.

Com a una solució inicial per al nostre problema, proposem una interfície Web, que simuli el comportament de la producció de llet de les vaques, i que alhora permeti al professor diferents configuracions per tal de que l'alumne reaccioni als problemes que sorgeixin. En conseqüència, haurem d'utilitzar diferents llenguatges de programació que explicarem més endavant, per tal de realitzar una interfície amigable amb l'usuari, implementar la seguretat pròpia que han de tenir aquest tipus d'aplicacions, i que es pugui instal·lar en la infraestructura actual de la facultat sense que això suposi més despeses.

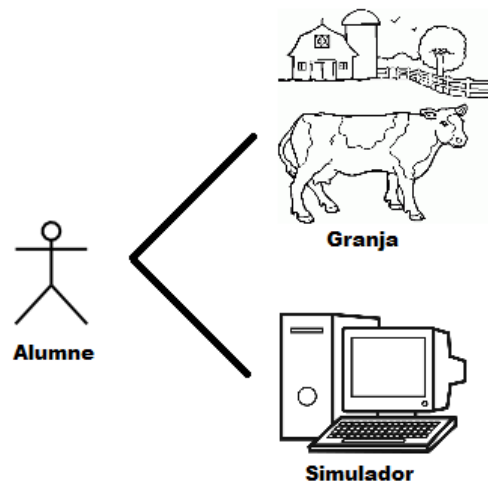


Figura 1 - Interacció de l'alumne amb la granja

1.5 ESTRUCTURA DE LA MEMÒRIA

Per concloure la introducció i després de veure les necessitats existents, farem un breu descripció a l'estructura de la memòria.

En el capítol d'estudi de viabilitat estudiarem la viabilitat del projecte en termes tècnics, econòmics i temporals.

En el següent capítol que és el d'anàlisi, analitzarem la producció de llet d'una vaca, així totes les possibles solucions per la implementació del simulador.

Un cop realitzat l'anàlisi, continuarem amb el capítol de disseny on es seleccionaran la solució per tal de implementar el simulador així com tot el disseny intern d'aquesta.

Després entrarem al capítol d'implementació on explicarem els passos que s'han seguit per tal de realitzar l'aplicació, i aprofundirem en aspectes interns i problemes que han sorgit duran aquesta fase.

Per acabar estudiarem les proves que s'han realitzat sobre l'aplicació i finalitzarem amb el capítol de conclusions on veurem si s'han complert o no els objectius del projecte i una valoració personal sobre ell.

Al final de la memòria hi tenim una bibliografia sobre tota la informació que s'ha utilitzat.

Capítol 2 - ESTUDI DE VIABILITAT

2.1 INTRODUCCIÓ

En aquest capítol estudiarem la viabilitat d'aquest projecte. Es veurà com és la situació actual i els beneficis aportats per l'aplicació. També analitzarem els requeriments necessaris per la seva implementació i altres possibles solucions al problema. I per últim els factors que ens han portat a escollir l'actual.

Un cop haguem escollit la millor de les solucions, coneixerem la seva planificació, avaluarem els riscos que comporti i realitzarem un pressupost estimat del programari.

Finalment hi ha un apartat de conclusions on reunirem els resultats de cada apartat i veurem si és viable o no.

2.1.1 Descripció

L'aplicació a realitzar és una eina de docència que ajuda a la posada en pràctica dels coneixements impartits en classes teòriques de la facultat de veterinària.

El principal problema que existia era poder veure els beneficis o perjudicis de les decisions preses per l'alumne sobre l'espècie de manera virtual, sense que això l'afectés de forma real. Aquesta aplicació ha de permetre la simulació de la producció de llet de diverses granges de tal forma que l'alumne pugui prendre aquestes decisions i veure els efectes que produeixen sobre el rendiment de la granja.

2.1.2 Objectius del projecte

Els objectius del projecte són:

- 1) L'objectiu crític del projecte és la implementació d'un programari que ens permeti simular la granja de vaques.
- 2) Com a objectius prioritaris, la creació de la interfície Web que permeti la visualització, manipulació i configuració del simulador.
- 3) També ha de permetre la interacció amb una base de dades.
- 4) Un altre objectiu prioritari serà la creació de diferents tipus de visualitzacions del programari per cadascun dels tipus d'usuaris.
- 5) Com a objectiu secundari necessitarem implementar les mesures de seguretat per tal de protegir les dades, la integritat dels usuaris i les intrusions via Web.

2.1.3 Parts interessades

A continuació podem veure la taula de les parts interessades en el projecte (Stakeholders).

Descripció	Responsabilitat
Responsable de la facultat	Participa en la seva definició i fa seguiment del projecte.
Director del projecte	Supervisa la feina del alumne.

Taula 1 – Stakeholders

En la següent taula podem observar tots els perfils d'usuaris que s'implementaran en l'aplicació.

Perfil	Responsabilitat
Administrador del sistema o professor	Gestió i control del simulador i d'usuaris.
Alumne	Interactua amb el simulador.
Visitant	Consulta d'informació.

Taula 2 - Perfil d'usuaris

Per últim, podem veure l'equip de projecte (Project Team) i les responsabilitats que te cadascú.

Perfil	Responsabilitat
Cap del projecte (CP)	Defineix, gestiona, planifica i controla el projecte.
Analista (A)	Col·labora amb el cap de projectes en l'estudi de viabilitat i la planificació. Analitza l'aplicació: arquitectura, metodologia, especificacions i estàndards entre altres. Participa en el disseny i validació.
Programador (P)	Disseny i desenvolupa l'aplicació d'acord amb l'anàlisi i planificació prevista. Participa en el procés de validació i implantació.
Tècnic de proves (TP)	Participa en el disseny de les proves internes i externes. Realitza les proves i participa en el procés de control de qualitat.
Director del projecte (DP) i/o tutor	Supervisa la feina de l'alumne, en alguns casos també pot actuar com <i>stakeholder</i> .

Taula 3 - Project Team

2.1.4 Producte i documentació del projecte

- a) Es lliurarà una aplicació informàtica.
- b) Es lliurarà un manual d'usuari i una ajuda en línia.
- c) S'elaborarà una memòria del projecte.

2.2 ESTUDI DE LA SITUACIÓ ACTUAL

En situació actual observem una gran dificultat per poder aplicar els coneixements teòrics que adquireixen els alumnes de forma pràctica. Aquests utilitzen fulls de càlcul i problemes sobre el paper per realitzar-les, i per tant resulta una feina complicada, tant per observar els resultats com per poder veure una evolució al llarg de diversos cicles. Per tal de poder veure com afecten les decisions que prenen els alumnes sobre la producció de llet, haurien de desplaçar-se a una granja físicament i fer un estudi durant un cert temps.

Per tant creiem oportú i necessari la implementació d'aquesta eina que permetrà als alumnes una interacció amb l'espècie i veurà els canvis que es produeixen en la producció de llet segons els criteris de decisió que siguin escollits.

2.3 REQUERIMENTS DEL PROJECTE

En aquest punt analitzarem els requeriments del projecte seguit de les restriccions de sistema que s'han acordat. En primer lloc veurem els requeriments funcionals que podríem definir com a coses que el producte ha de realitzar, i després els no funcionals que són propietats que aquest ha de tenir.

2.3.1 Requeriments funcionals

Llistat de requeriments funcionals són:

1. Simulació de la producció de llet de granges així com les vaques que la configuren.
2. Interfície Web per interactuar amb el simulador.
3. Manteniment (altes, baixes, modificacions) de dades de l'aplicació.
4. Control d'accés dels usuaris de l'aplicació.
5. Còpies de seguretat i recuperació de dades.

2.3.2 Requeriments no funcionals

Llistat de requeriments no funcionals són:

1. Permetre la connexió simultània d'usuaris a l'aplicació.
2. Tolerància a fallides.
3. Control de totes les entrades d'usuaris.
4. El control d'accés es farà d'acord amb una llista d'usuaris i permisos (ACL - Access Control List).
5. La seguretat de les dades: el servidor haurà de tenir restriccions d'accés físic.

2.3.3 Restriccions del sistema

Les restriccions que mostrem a continuació s'han imposat amb la intenció de reduir els costos del projecte i de possibilitar la utilització des de qualsevol ordinador amb accés a la xarxa.

Llistat de restriccions que afecten al programari:

1. L'aplicació s'ha d'implementar en un entorn Linux.
2. Les pàgines Web han de ser accessibles des de qualsevol navegador.
3. L'aplicació ha d'adaptar-se al sistema físic disponible a la facultat.
4. El projecte ha d'estar finalitzat abans del 30 de juny de 2010.
5. L'aplicació s'ha de desenvolupar utilitzant programari lliure.

2.3.4 Catalogar i prioritzar els requeriments

Un cop analitzats tots els requeriments continuarem per dividir-los entre essencial, condicional i opcional, segons la seva prioritat.

Catalogarem un requisit com a essencial quan la seva realització sigui imprescindible per la finalització del projecte. Si el classifiquem com a condicional vol dir que l'aplicació podria arribar a funcionar sense ell, però que creiem que és necessari. I per últim si un requeriment és opcional, vol dir que podem decidir de implementar-lo o no i no afectarà al funcionament final de l'aplicació.

	RF1	RF2	RF3	RF4	RF5
Essencial	X	X	X		
Condicional				X	
Opcional					X

Taula 4 - Requeriments funcionals

A continuació s'analitzaran d'igual forma els requeriments no funcionals :

	RNF1	RNF2	RNF3	RNF4	RNF5
Essencial	X	X	X		
Condicional				X	
Opcional					X

Taula 5 - Requeriments no funcionals

Per últim analitzarem la relació entre objectius i requeriments funcionals i no funcionals. Aquesta és la que ens permetrà analitzar l'ordre amb que posteriorment es realitzaran les funcionalitats i els objectius que complirem quan siguin finalitzats.

	RF1	RF2	RF3	RF4	RF5	RNF1	RNF2	RNF3	RNF4	RNF5
O1	X	X				X	X	X	X	X
O2		X				X	X		X	X
O3			X				X			X
O4				X			X		X	
O5					X		X	X	X	X

Taula 6 - Relació entre objectius i requeriments

2.4 ALTERNATIVES I SELECCIÓ DE LA SOLUCIÓ

Després de saber els requeriments i restriccions del sistema s'han buscat possibles alternatives que permetin el compliment d'aquests.

2.4.1 Alternativa 1

Adquisició i adaptació d'un programari existent, per exemple, Oviprogramari ebs Premium:

http://www.asmexcriadoresdeovinos.org/informa/2005/car_oviprogramariebs.html

En aquest cas seria d'animals ovins i no bovins i s'hauria d'adaptar per obtenir una simulació correcte.

Funcionalitats:

Gestió de dades.

Creació de llistats i filtres.

Creació d'informes.

Simuladors i sistemes experts.

Actualitzacions des d'Internet.

Costos:

9.000€

2.4.2 Alternativa 2

Desenvolupament d'un simulador d'una granja de vaques.

Característiques:

Total independència de l'aplicació.

S'ajusta als requeriments proposats.

Ajustable als recursos disponibles per la facultat.

Costos:

Segons la planificació prevista.

2.4.3 Solució proposada

Després de veure les possibles alternatives, farem una comparació entre les dues possibles solucions al problema plantejat.

	Costos adquisició	Costos adaptació	Nous recursos	Suport	Nivell d'integració	Complexitat	Formació
Alternativa 1	9.000€	Alts	Adaptable	Inclòs en el preu	Baix	Alta	Inclòs en el preu
Alternativa 2	0€	Pressupost projecte	No cal	Inclòs en el projecte	Alt	Mitjana	Inclòs en el projecte

Taula 7 - Comparació de les alternatives.

Els termes de comparació són primerament els costos d'adquisició i d'adaptació i en el cas d'adaptar-la els nous recursos que necessitarem. A continuació analitzem si es disposa de suport als usuaris, així com del nivell d'integració que té sobre els requeriments especificats anteriorment i la complexitat que suposarà la seva realització.

Per últim veurem si el període de formació als usuaris està o no inclòs en el preu total del projecte.

Justificació de la solució.

A la vista de les característiques de les dues alternatives i el quadre comparatiu, veiem que la alternativa 2 seria la més adequada.

Tant pels costos com la feina necessària d'adaptació de la alternativa 1 veiem més factible la segona, que s'ajusta a totes les necessitats que ens demana el responsable de la facultat.

2.5 AVALUACIÓ TECNOLÒGICA DEL PROJECTE

Si analitzem el projecte des del punt de vista tecnològic, veiem que necessitarem una infraestructura de servidor Web que permeti la connexió simultània d'usuaris, i per tant tota les eines necessàries per la realització d'aquest.

L'esquema del sistema és el que es mostra a la següent figura, que consta del servidor WEB que s'encarrega de rebre i contestar peticions, quan els usuaris les sol·licitin des del seu navegador Web. Per sota d'aquest tenim un servidor d'aplicacions que interactua amb la base de dades i genera les funcionalitats del simulador i totes les pàgines Web dinàmiques que es mostren a través del servidor web.

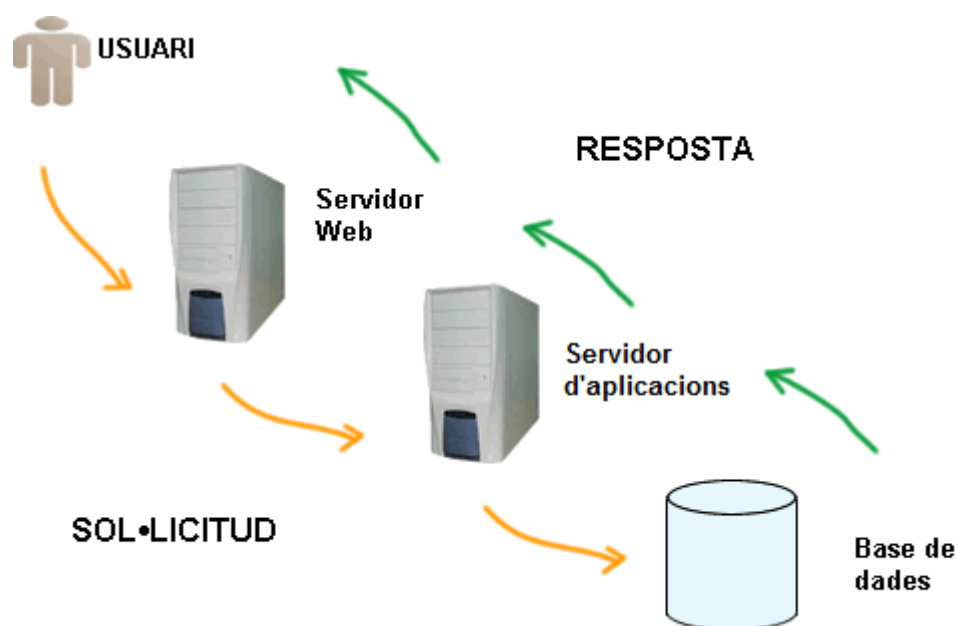


Figura 2 - Esquema del sistema

2.6 PLANIFICACIÓ DEL PROJECTE

Un cop analitzat el projecte tecnològicament continuarem l'estudi avaluant-lo temporalment, i així podem acabar d'analitzar si podrà complir les restriccions temporals que presenta aquest.

Calendari del projecte

El projecte es desenvoluparà d'octubre de 2009 a abril de 2010 amb una dedicació de 9 hores setmanals. El total d'hores dedicades al projecte serà de 253 hores.

Data començament:

1 d'octubre de 2009

Data finalització:

25 d'abril de 2010

Eines de planificació i control:

Microsoft Project (eina de seguiment i control de projectes).

2.5.1 Recursos del projecte

En la següent taula fem l'estimació de les valoracions de cada un dels recursos humans que s'han detallat anteriorment.

Recursos humans	Valoració
Cap de projecte	100 €/h
Analista	50 €/h
Programador	30 €/h
Tècnic proves	20 €/h

2.5.2 Tasques del projecte

Nº	Descripció de l'activitat	Durada	Data d'inici
1	Inici del projecte: assignació i matriculació del projecte	2,5d	02/10/09
2	Planificació	11h	16/10/09
3	Estudi de viabilitat	10h	16/10/09
4	Aprovació estudi viabilitat (Punt de control)	1h	24/10/09

5	Anàlisi de l'aplicació	24h	24/10/09
6	Anàlisi de requeriments (casos d'ús)	10h	24/10/09
7	Anàlisi de dades (base de dades)	5h	31/10/09
8	Anàlisi de la seguretat i legalitat	5h	06/11/09
9	Documentació de l'anàlisi	3h	08/11/09
10	Aprovació de l'anàlisi (Punt de control)	1h	13/11/09
11	Disseny de l'aplicació	19h	13/11/09
12	Disseny de la base de dades	5h	13/11/09
13	Disseny de l'aplicació i interfície Web	5h	15/11/09
14	Disseny de les proves (test)	5h	20/11/09
15	Documentació del disseny	3h	22/11/09
16	Aprovació del disseny (Punt de control)	1h	27/11/09
17	Desenvolupament de l'aplicació	127h	27/11/09
18	Preparació de l'entorn	12h	27/11/09
19	Configuració base de dades	5h	05/12/09
20	Mòdul d'adquisició de dades i funcionalitats de l'aplicació	100h	11/12/09
21	Desenvolupament de la interfície d'usuari	10h	26/02/10
22	Test i proves	16h	06/03/10
23	Proves generals	10h	06/03/10
26	Documentació de desenvolupament i test	5h	13/03/10
27	Aprovació del desenvolupament i proves (Punt de control)	1h	19/03/10
28	Implantació	20h	19/03/10
29	Instal·lació	5h	19/03/10
30	Proves reals sobre la infraestructura final	10h	21/03/10
31	Formació d'usuari	5h	28/03/10
32	Generació de documents (memòria del projecte)	30h	19/03/10
33	Tancament del projecte	1h	23/04/10
34	Defensa del projecte	5h	23/04/10

Taula 8 – Tasques del projecte

2.5.3 Planificació temporal

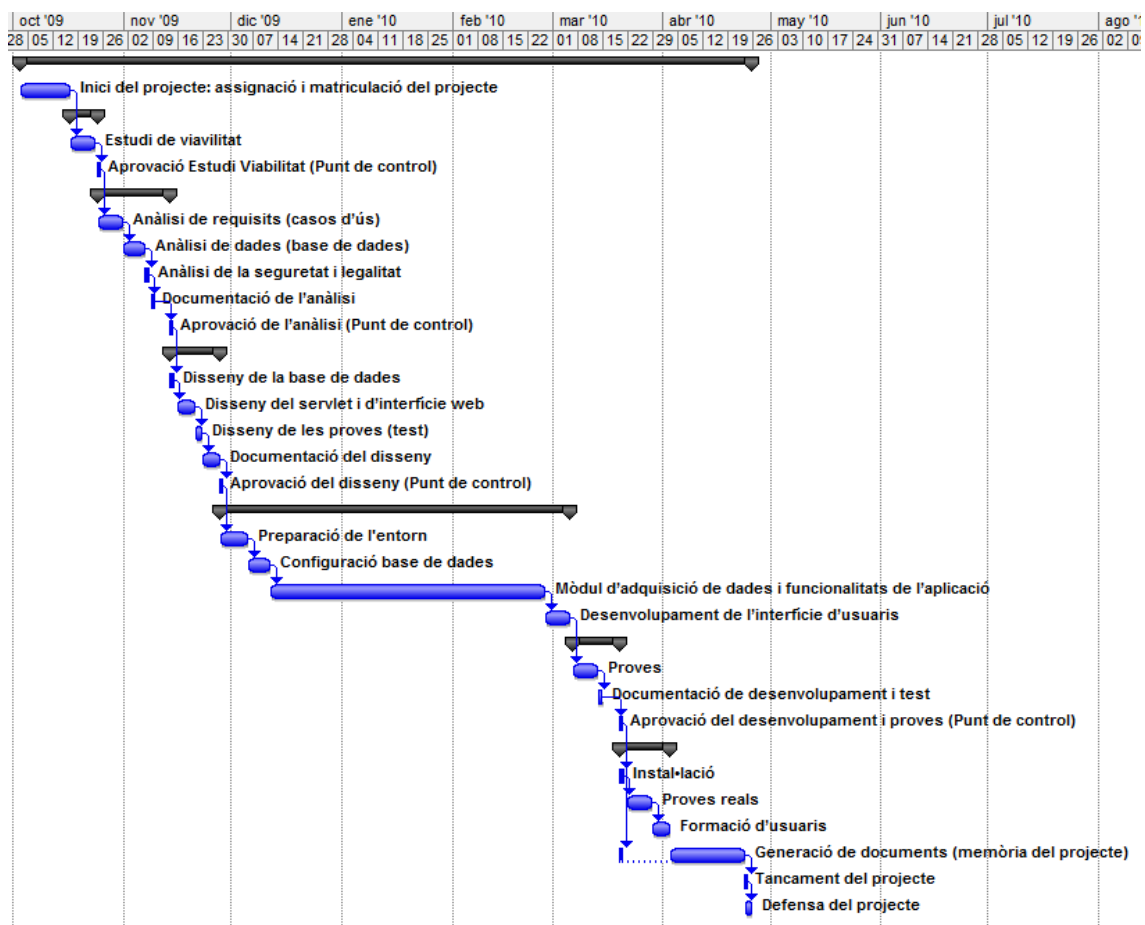


Figura 3 - Diagrama de Gantt del projecte

2.7 AVALUACIÓ DE RISCOS

Continuant amb l'estudi de viabilitat el següent pas és l'avaluació dels riscos que es poden presentar durant la realització del projecte així com una catalogació d'aquests, i el pla de contingència que s'haurà d'utilitzar en cas que apareguin.

2.6.2 Llista de riscos

A continuació s'enumeren els possibles riscos que s'han previst que poden aparèixer durant la realització projecte.

- R1. Planificació temporal optimista: en la fase d'estudi de viabilitat. No s'acaba en la data prevista, augmenten els recursos.
- R2. Manca alguna tasca necessària: en la fase d'estudi de viabilitat. No es compleixen els objectius del projecte.

R3. Pressupost poc ajustat: en la fase d'estudi de viabilitat. Menys qualitat, pèrdues econòmiques.

R4. Canvi de requeriments: en la fase d'estudi de viabilitat, i d'anàlisi. Endarreriment en els desenvolupament i resultat.

R5. Equip del projecte massa reduït: en la fase d'estudi de viabilitat. Endarreriment en la finalització del projecte, no es compleixen els objectius del projecte.

R6. Eines de desenvolupament inadequades: en la fase d'implementació. Endarreriment en la finalització del projecte, menys qualitat,

R7. Dificultat per accedir als *stakeholders*: en la fase d'estudi de viabilitat, anàlisi, proves i formació. Manquen requeriments o són inadequats, endarreriments, insatisfacció usuaris.

R8. No es fa correctament la fase de test: en la fase de desenvolupament, i d'implantació. Manca de qualitat, deficiències en l'operativa, insatisfacció usuaris, pèrdua econòmica.

R9. Incompliment d'alguna norma, reglament o legislació: en qualsevol fase. No es compleixen els objectius, repercussions legals.

R10. Manca d'implantació de mesures de seguretat: en la fase d'estudi de viabilitat, anàlisi i desenvolupament. Pèrdua d'informació, incompliment legal, pèrdues econòmiques.

R11. Abandonament del projecte abans de la finalització: en qualsevol fase. Pèrdues econòmiques, frustració.

2.6.3 Catalogació de riscos

En la següent taula classifiquem els riscos segons la probabilitat de que apareguin així com la gravetat del impacte que ocasionarien sobre el projecte.

	Probabilitat	Impacte
R1	Alta	Crític
R2	Alta	Crític
R3	Alta	Crític
R4	Alta	Marginal
R5	Alta	Crític
R6	Baixa	Crític

R7	Baixa	Crític
R8	Alta	Crític
R9	Mitjana	Crític
R10	Alta	Crític
R11	Mitjana	Catastròfic

Taula 9 – Classificació dels riscos

2.6.4 Pla de contingència

Després de la catalogació de riscos s'ha de pensar en les possibles solucions per tal d'afrontar-les, si és que n'hi ha, i aquestes es recullen a la taula que es mostra a continuació.

	Solució que cal adoptar
R1	Ajornar alguna funcionalitat, fer una assegurança.
R2	Revisar l'estudi de viabilitat, modificar la planificació.
R3	Renegociar amb el client, fer una assegurança.
R4	Renegociar amb el client, ajornar funcionalitat, modificar planificació i pressupost.
R5	Demandar un ajornament, negociar amb el client.
R6	Millorar la formació de l'equip, preveure eines alternatives, millorar la qualitat.
R7	Fixar un calendari de reunions, millorar el contacte amb el client.
R8	Dissenyar els test amb antelació, realitzar tests automàtics, negociar contracte de manteniment, donar garanties.
R9	Revisar les normes i legislació, consultar a un expert.
R10	Revisar la seguretat en cada fase, aplicar polítiques de seguretat actives.
R11	No té solució.

Taula 10 –Pla de contingència

2.8 PRESSUPOST

En aquest punt estudiarem la viabilitat del projecte des del punt de vista econòmic, en primer lloc es faran les estimacions de costos del personal i recursos addicionals, i mitjançant aquestes posteriorment les dels costos de cadascuna de les activitats que el componen.

Per acabar el pressupost farem un resum i l'anàlisi cost benefici per tal d'arribar a alguna conclusió sobre si el projecte és econòmicament viable o no.

2.7.1 Estimació cost de personal

En la següent taula mostrem les hores que realitza cada integrant del projecte així com el cost total d'aquestes.

Cap de projecte	50h	5.250,00 €
Analista	50h	2.675,00 €
Programador	143h	4.227,50 €
Tècnic proves	10h	370,00 €
TOTAL	253h	12.522,50 €

Taula 11 – Estimació cost del personal

2.7.2 Estimació cost dels recursos

Després d'estimar el cost del personal el següent punt que s'ha d'analitzar és el cost dels recursos addicionals necessaris per poder fer el projecte, i aquests es mostren en forma de taula a continuació.

	Cost amortització	Cost unitari	Període amortització	Període utilització
Amortització PC programador	100 €	1200 €	36 mesos	8 mesos
Amortització MS Office	20,8 €	250 €	36 mesos	8 mesos
Amortització MS Project	30 €	360 €	36 mesos	8 mesos

Taula 12 – Estimació costos dels recursos

2.7.3 Estimació cost de les activitats

Tot seguit i a partir de la informació anterior desglossarem la estimació de costos per a cadascuna de les activitats que formen el projecte.

Activitat	Durada	Cost
Inici del projecte: assignació i matriculació del projecte	60h	200,00 €
Planificació	11h	1.100,00 €
Estudi de viabilitat	10h	1.000,00 €
Aprovació estudi viabilitat (Punt de control)	1h	100,00 €
Anàlisi de l'aplicació	24h	1.275,00 €
Anàlisi de requeriments (cassos d'ús)	10h	500,00 €
Anàlisi de dades (base de dades)	5h	250,00 €
Anàlisi de la seguretat i legalitat	5h	250,00 €
Documentació de l'anàlisi	3h	150,00 €
Aprovació de l'anàlisi (Punt de control)	1h	125,00 €
Disseny de l'aplicació	19h	935,00 €
Disseny de la base de dades	5h	230,00 €
Disseny de l'aplicació i l'interfície Web	5h	230,00 €
Disseny de les proves (test)	5h	200,00 €
Documentació del disseny	3h	150,00 €
Aprovació del disseny (Punt de control)	1h	125,00 €
Desenvolupament de l'aplicació	127h	3.810,00 €
Preparació de entorn	12h	360,00 €
Configuració base de dades	5h	150,00 €
Mòdul d'adquisició de dades i funcionalitats de l'aplicació	100h	3.000,00 €
Desenvolupament de l'interfície d'usuaris	10h	300,00 €
Test i proves	43h	1.267,50 €
Proves unitàries	10h	300,00 €
Documentació de desenvolupament i test	5h	150,00 €
Aprovació del desenvolupament i proves (Punt de control)	3h	322,50 €
Implantació	20h	830,00 €
Instal·lació	5h	220,00 €
Proves reals	10h	360,00 €

Formació d'usuaris	5h	250,00 €
Generació de documents (memòria del projecte)	30h	3.000,00 €
Tancament del projecte	1h	100,00 €
Defensa del projecte	5h	500,00 €

Taula 13 – Estimació costos de les activitats

2.7.4 Resum i anàlisi cost benefici

D'acord al pressupost, i considerant que el centre on s'implanta l'aplicació ja té els ordinadors i el programari necessari per que els alumnes puguin interactuar amb ella, els costos pressupostats totals serien els següents:

Cost de desenvolupament del projecte 12.522,50 €
 Cost d'amortització del material 150,80 €

Total: 12.673,30 €

Tenint en compte els beneficis que suposa sobre els alumnes, que són la facilitat de posar en pràctica els coneixements teòrics impartits, l'estalvi en desplaçaments i estància (respecte a fer-ho de forma real) entre altres, considerem que els costos del projecte són assumibles i que el benefici a llarg termini és molt alt.

Cal puntualitzar, que el fet de disposar de l'aplicació de forma fàcil, augmenta i millora la qualitat i quantitat de la pràctica dels coneixements. També es té en compte que la utilització del simulador permet veure informació del cicle sencer de la vida d'una vaca en un temps reduït i que les decisions preses pels alumnes no afecten de forma real a l'espècie, per tant amplia les possibilitats que tindrà el professor respecte a fer-ho sense el simulador.

Per tant l'amortització del projecte seria en pocs mesos després de d'implantació de l'aplicació, ja que aporta uns grans beneficis que justifiquen la inversió inicial, i més si es té en compte que el cost real del projecte al tractar-se d'un projecte de final de carrera és zero.

2.9 CONCLUSIONS

Els resultats de l'estudi de viabilitat mostren que tant des del punt de vista tècnic, econòmic i temporal el projecte és viable.

Tècnicament hem analitzat les eines necessàries per la seva implementació i els resultats obtinguts són positius ja que en el mercat actual podem trobar versions lliures de programari que ens permeten crear-ne.

En termes econòmics veiem que els beneficis que ens aporta l'aplicació són molt superiors als inconvenients que ens pot ocasionar. També podem observar que l'alternativa que hem proposat a l'aplicació implementada, no s'ajusta als requeriments que descrits anteriorment.

En últim lloc, l'estudi des del punt de vista temporal ens demostra que és possible la seva realització dins els límits establerts i que per tant també compleix les restriccions imposades al projecte.

Per tant arribem a la conclusió que **aquest projecte és viable.**

Capítol 3 - ANÀLISI

L'objectiu d'aquest capítol serà l'anàlisi de tot el necessari per la realització del projecte, des de l'estudi de la lactació i els cicles d'una vaca, passant per l'estudi del marc tecnològic i les opcions existents al mercat, fins als possibles llenguatges de programació a utilitzar per la seva realització.

En primer lloc, farem un anàlisi de tots els requeriments que ha de tenir la nostra aplicació.

3.1 REQUERIMENTS DEL PROJECTE

Arribats a aquesta fase s'han analitzat amb més detall, juntament amb el nostre client el professor Calsamiglia, tots els requeriments del projecte, ja que alguns d'ells s'havien de modificar respecte a l'estudi inicial.

3.1.1 Requeriments funcionals

1. Simulació de la producció de llet de granges així com les vaques que la configuren.
2. Interfície Web per interactuar amb el simulador.
3. Manteniment (altes, baixes, modificacions) de dades de l'aplicació.
4. Control d'accés dels usuaris de l'aplicació.
5. Còpies de seguretat i recuperació de dades.
6. Muntar i configurar la infraestructura que permeti suportar el servidor Web i l'accés des d'altres ordinadors.
7. Permetre a l'usuari alumne realitzar les accions d'avançar un dia, de secar, inseminar i preguntar estat per a cada vaca.

3.1.2 Requeriments no funcionals

1. Permetre la connexió simultània d'usuaris a l'aplicació.
2. Tolerància a errades.
3. Control de totes les entrades d'usuaris.
4. El control d'accés es farà d'acord amb una llista d'usuaris i permisos (ACL - Access Control List).
5. La seguretat de les dades: el servidor haurà de tenir restriccions d'accés físic.

6. Generar les bases de dades inicials en cas que no existeixin i afegir un usuari administrador i les dades per defecte a aquestes bases de dades.
7. Generar una aplicació que permeti fer gràfiques via Web.
8. Generar arxius de configuració dels paràmetres del simulador editables pels administradors.
9. Generar diferents vistes per cada tipus d'usuari.

3.1.3 Conclusió dels requeriments

Com es pot observar, s'han afegit alguns requeriments que en un principi no es van tenir en compte, i que el nostre client creu que són imprescindibles per complir les seves expectatives sobre l'aplicació. Alguns d'ells també van ser inclosos quan es van observar algunes mancances alhora de permetre diferents configuracions del servidor ja que en un primer moment s'havia pensat en fer-lo amb una configuració més estàtica.

3.2 SIMULACIÓ D'UNA VACA

Després d'analitzar els requeriments del projecte començarem l'estudi de l'espècie a simular, la vaca. Degut a la complexitat que això suposava, el professor Calsamiglia va simplificar molt aquesta informació, alhora que ens va facilitar les formules per modelar el comportament d'aquestes, que mostrem a continuació.

DESCRIPCION FUNDAMENTAL DE LA VACA (determina la genética potencial de la vaca).

SI NO LO ENTENDEIS BIEN, PODRIAMOS QUEDAR PARA EXPLICARLO: Yo he podido dibujar una curva de lactación (eje x, días en lactación; eje de las Y, producción de leche) con estos datos).

1. Asignar un valor de producción anual potencial (PRODAp) al azar entre 7.000 y 11.000 Litros/año
2. Asignar un valor de proteína potencial (PROTp) al azar entre 2.8 y 3.4
3. Asignar un valor de grasa potencial (GRASp) al azar entre 3.5 y 4.2
4. Asignar un valor potencial de células somáticas (CCSp) entre 80 y 650
5. Asignar un número de parto (PART) entre 1 y 5

A partir de estos valores se calculan las condiciones actuales para la vaca

1. $PICO = f(PARTx) * PRODAp / 200$; donde
 - a. $PART = 1 = 0.75$;
 - b. $PART > 1, PART > 1 = 1.0$
2. Días al pico:
 - a. $PART = 1 = RND$ entre 55-70
 - b. $PART > 1 = RND$ entre 35-55
3. Producción del día 0: $PRODd0 = 0.75 * PRODAp / 200$
4. *Grasa actual*: $GRAS = GRASp * f(x) (x=1)$

5. *Proteína actual*: $PROT = PROTp * f(x) (x=1)$
6. *Células somáticas actuales*: $CCS = CCSp * f(x) (x=1)$

Producción anual potencial	PROPAp	F(RND:7000-11000)
Número de parto	PARTO	F(RND); 1-5
Pico	PICO	Parto 1: $0.75 * PRODAp / 200$ Parto > 1: $PRODAp / 200$
Días a pico	dPICO	F(RND) Parto 1: RND 60-80 Parto > 1: RND 35-55
Tiempo del pico	tPICO	F(RND) Parto 1: RND 25-35 Parto > 1: RND 15-25
Producción Inicial	PRODd0	$0.75 * PICO$
DEL	DEL	$DEL = DEL + 1$
Producción	PROD-PP PROD-PIC PROD-LAC	PROD=
Estado	ESTADO	Postparto (POSTP = 1) Ciclando (CLICLO=2) Inseminada (IA=3) Gestante (GEST=4) Seca (SECA=5)

A partir de aquí, se procede a "pasar días" automáticamente para crear una historia reciente de la vaca (se guardan valores generados

en base de datos; no hay input del usuario, sino que las funciones se determinan al azar)

1. Todos los animales parten del día 0 en un estado postparto (ESTADO=1)
2. Generar un número de días a pasar para esta vaca (entre 1 y 400)
3. Generar un contador de días postparto o días en lactación (DEL)
4. La producción se determina por periodos:
 - a. La producción del día 0 se ha determinado anteriormente (PRODd0)
 - b. Si DEL está entre 0 y 45 → PRODhoy =
 - i. PARTO =1; $1,009 * PROD(ayer)$
 - ii. PARTO >1; $1,016 * PROD(ayer)$
 - c. Si DEL = dPICO; durante 15 días PRODa = PRODa
 - d. Si DEL > (dPIC+15); PRODhoy
 - i. $0,998 * PRODayer$
 - ii. $0,9916 * PRODayer$
5. Pasar días. Los animales pasan de estado PP a estado "Ciclando" (ESTADO 2) de forma al azar entre los días 25 y 40 DEL
6. Cuando están en estado CICLA, se pone en marcha un contador del ciclo que dura entre 18 y 24 días al azar (RND para cada ciclo). Cuando llega el día determinado, se aplica una fertilidad de 35% (probabilidad 0.35 de cambiar de estado a INSEMINADA; ESTADO 3; o se mantiene ESTADO 2)

- a. Si se queda Preñada, inicia un contador de gestación (DIASGEST) que dura 275-285 días (al azar)
7. Cuando días de gestación = 215-225, cambio de estado a SECA. Contador de SECA. El animal deja de dar leche.
8. Cuando se cumplen los DIASGEST, parto, y se inicia el ciclo de nuevo (Estado =1), DEL a cero y vuelve a empezar un ciclo.

Con esto se ha generado el historial reciente de una vaca. Esta operación, en el modelo final, se repetirá n veces para crear la granja.

UNA VEZ GENERADA LA GRANJA, LAS VACAS SEGUIRAN EL CICLO, EXCEPTO QUE LAS ACCIONES DE FERTILIDAD Y SECADO LAS REALIZARA EL USUARIO.

Para ello, cuando el contador del ciclo alcance el valor previsto, el estado de la vaca aparece como CELO. Si ese día el usuario decide inseminar, la fertilidad es una función (valor fijo del 40%). El resto, similar al modelo anterior.

El secado debe ser una acción siempre disponible, pero debe realizarse 60 días previos al parto, o 220 días de gestación. La acción la hará el usuario.

En Inseminación o secado, yo pondría las acciones siempre disponibles en la pantalla, y el estudiante deberá seleccionar la vaca y activar el botón de acción a tomar.

A partir d'aquest document hem extret de tots els paràmetres necessaris per simular la lactació d'una vaca i dels estats que configuren els cicles de lactació. A continuació mostrem una figura explicativa d'aquest cicle.

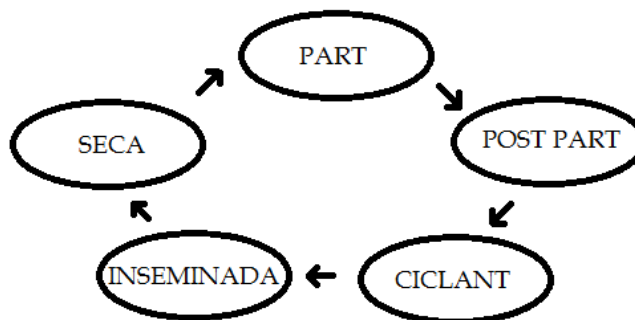


Figura 4 - Cicle d'estats d'una vaca

Per tant, després de l'anàlisi de la vaca comencem a fer un estudi sobre totes les opcions disponibles per tal de proveir la infraestructura necessària per la seva simulació.

3.3 MARC TECNOLÒGIC

El següent punt a analitzar és el marc tecnològic del projecte, que el dividirem en tres parts. En primer lloc estudiarem els servidors Web, que ens crearan la base per implementar l'aplicació i les funcions de servidor. A continuació veurem els sistemes de gestió de bases de dades que serviran per la gestió de la base de dades, i finalment tots els possibles llenguatges de programació per generar Webs dinàmiques i els entorns de programació que els suporten.

3.3.1 Servidors Web

Com que la nostra aplicació ha de ser accessible via Internet, és imprescindible l'ús d'un servidor Web.

La funció d'un servidor Web és la d'acceptar requeriments HTTP des de la xarxa, i proveir respostes HTTP al client que ha fet el requeriment, aquestes respostes normalment són en forma de document o pàgina HTML, que són les que necessitem en aquest projecte.

El seu esquema de funcionament es basa en l'execució del següent bucle infinitament:

1. Espera una petició al port TCP indicat (l'estandard HTTP per defecte és el 80).
2. Rep la petició.
3. Busca el recurs que demana la petició.
4. Envia el recurs reutilitzant la mateixa connexió que s'ha utilitzat en la petició.
5. Torna al punt número 2.

Depenent del tipus de petició que rebí el servidor Web, pot donar diferents tipus de resposta. La més simple possible és una pàgina HTML estàtica que estigui emmagatzemada al servidor local, o bé respostes més complexes, que són generades amb alguns llenguatges de programació que retornen diferents Webs depenent de la petició d'entrada, anomenades Webs dinàmiques.

A continuació veurem algunes possibilitats que trobem al mercat en quan a servidors Web, i les seves característiques més importants.

Apache HTTP Server

És un dels servidors més populars, és flexible, ràpid i eficient, és actualitzat contínuament i està adaptat als protocols HTTP 1.1. La seva última versió estable és la



2.2.15 es va publicar el passat 05/03/2010.

Les seves principals característiques són:

- És una tecnologia gratuïta de codi obert. El fet de ser gratuïta és important però no tant com que sigui de codi font obert. Això li atorga una transparència al programari de manera que podem saber exactament el que estem instal·lant al nostre servidor.
- És un servidor altament configurable de disseny modular. És relativament senzill ampliar les seves capacitats. En l'actualitat existeixen molts mòduls per a aquest servidor i són accessibles a tothom quan els necessiti.
- Disposa d'extensions per treballar amb els llenguatges Perl, PHP i entre altres llenguatges de script. També disposa d'extensió pel llenguatge Java i pàgines JSP, i té tot el suport necessari per generar pàgines Web dinàmiques.
- Permet personalitzar les respostes davant dels diferents errors que es puguin donar durant el seu funcionament. També dóna la possibilitat de que la resposta a aquest error sigui donada a partir d'un script creat per l'administrador.
- És altament configurable en termes de creació i gestió de logs.

Permet la creació d'arxius de log fets a mida per l'administrador, i d'aquesta manera tenir un control total sobre tot el que succeeix dins el servidor.

IIS Server

El servidor Internet Information Server és un servidor de pàgines Web, FTP i Gopher que suporta els protocols HTTP, FTP i GOPHER. La seves principals desavantatges : és un programari privatiu que el seu




codi font no és obert, i que només funciona sota un sistema operatiu Windows. La seva última versió estable és la 7.5 que va sortir al mercat el passat 22/10/2009.

Les característiques més destacades són les següents:

- És molt fàcil d'instal·lar. La instal·lació i posada en funcionament és qüestió de minuts i la publicació de pàgines i arxius és igual de senzilla.
- Treballa com un servei Windows NT, per tant s'integra a la perfecció amb el resta de serveis Windows NT.

- Utilitza una eina de configuració amb vista gràfica i molt senzilla, el control MMC (Microsoft Management Console), que ajuda notablement a les feines d'administració i seguretat del servidor.
- Permet fer l'administració del servidor de manera remota per Web, utilitzant un navegador qualsevol.
- Està completament integrat amb el sistema de seguretat de Windows NT. Els usuaris accedeixen a les pàgines en funció dels permisos que tenen sobre elles
- Permet la publicació de Webs mitjançant eines Web com FrontPage i similars, o mitjançant una còpia simple de l'arxiu sobre la xarxa.
- Permet utilitzar mecanismes de seguretat com l'autenticació xifrada de Windows NT.
- Permet la integració de forma força senzilla del servidor SQL de Microsoft, i diferents sistemes per crear consultes sobre aquestes bases de dades (idc/htx, dbweb i altres).
- Les últimes versions estan adaptades al protocol HTTP 1.1.
- També podem crear pàgines i scripts, suportant Active Server Pages (ASP), protocols CGI i ISAPI, que permeten la generació de webs dinàmiques.

AOLServer

Si comparem aquest servidor amb els analitzats anteriorment, aquest és el que està menys desenvolupat,  la seva última versió estable és la 4.5.0 que data del 27/06/2006, i per tant la majoria de les seves característiques estan desfasades, però de tota manera a dia d'avui encara és utilitzat i dona bons resultats en algunes empreses. Normalment es combina amb el paquet d'eines Web OpenACS ja que està implementat específicament per aquest servidor i es combinen a la perfecció les seves particularitats.

Les seves principals característiques són:

- Com Apache és un servidor relativament fàcil de configurar, té una gran flexibilitat i està basat en un disseny modular, per tant, resulta fàcil augmentar les seves capacitats.
- Utilitza una arquitectura multifil per proporcionar un alt nivell de resposta i eficiència, això fa que sigui molt utilitzat per empreses que necessiten nombroses

connexions a la base de dades i una personalització de les pàgines per a cada usuari.

- Proporciona un accés molt simple a les bases de dades i permet connexions amb altra disponibilitat i alta rapidesa d'accés.
- És un projecte de codi obert i distribuït de forma gratuïta.

Tomcat

Apache Tomcat és un contenidor de servlets que implementa les especificacions de servlet i de Java Server Pages (JSP), proporcionant un entorn per al codi Java a executar en cooperació amb un servidor



Web tot i que també pot ser utilitzat com a servidor Web com els anteriors.

L'última versió estable és la 6.0.29, i el seu llançament va ser el 22/07/2010.

Les seves principals característiques són:

- És un servidor HTTP i un contenidor de servlets alhora, per tant, té un funcionament autònom total.
- És la implementació de referència de les especificacions de servlets 2.4 i de JSP 2.0.
- Pot funcionar com a servidor HTTP autònom o connectat a un altre servidor HTTP com Apache o ISS.
- Dóna la possibilitat d'arrancar varies instàncies Tomcat i permet a entorns de desenvolupament com Eclipse o NetBeans crear-ne de servlets utilitzant la seva pròpia configuració.
- És programari lliure i de codi obert.

Taules comparatives

Per veure d'una forma més intuïtiva totes les característiques dels servidors Web explicats anteriorment, utilitzarem unes taules comparatives de diversos factors que influiran sobre la nostra elecció final.

La primera d'aquestes ens mostra informació general sobre els servidors, entre aquesta informació destaquem si aquest té algun cost o no, o si el codi font és obert i per tant tenim accés a ell per si fos necessària alguna modificació.

Servidor	Creador	Cost	Codi font obert	Llicència del programari	Última versió estable	Data de sortida
AOLserver	NaviSoft	Lliure	Sí	Pública	4.5.0	27/06/2006
Apache HTTP	Fundació del programari de Apache	Lliure	Sí	Pública	2.2.15	05/03/2010
Apache Tomcat	Fundació del programari de Apache	Lliure	Sí	Pública	6.0.29	22/07/2010
ISS Server	Microsoft	Inclòs als productes de la família Windows NT privatiu	No	Propietària	7.5	22/10/2009

Taula 14 – Taula comparativa servidors I

Per poder comparar millor aquests servidors, la taula següent recull un resum de les seves característiques principals en quan a seguretat, contingut dinàmic i algunes característiques addicionals com l'espai de funcionament o si disposen de protocol IPv6 o consola interna d'administració.

	Seguretat			Contingut dinàmic					<i>Funcionament dins l'espai d'usuari o del nucli</i>	<i>Consola d'administració</i>	<i>IPv6</i>
	<i>Autenticació bàsica en l'accés</i>	<i>Autenticació del accés del resum</i>	<i>https</i>	<i>Cgi</i>	<i>FastCGI</i>	<i>Servlet</i>	<i>SSI</i>	<i>ASP.NET</i>			
AOLserver	Sí	No	Sí	Sí	No	No	Sí	Si	usuari	Si	Si
Apache HTTP	Sí	Sí	Sí	Sí	Sí	Si	Sí	Si	usuari	Sí	Sí
Apache Tomcat	Sí	Sí	Sí	Sí	No	Sí	Sí	No	usuari	Sí	Si
ISS Server	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	nucli usuari	Sí	Sí

Taula 15 – Taula comparativa servidors II

Finalment i per acabar la comparació mostrem una taula amb l'anàlisi de quins sistemes operatius suporten aquests servidors, que ens permetrà descartar directament els que no compleixin els requisits del sistema.

Servidor	Windows	Mac	Linux	Solaris	OpenVMS	AIX	z/OS
AOLserver	Sí	Sí	Sí	Sí	No	No	No
Apache HTTP	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Apache Tomcat	Sí	Sí	Sí	Sí	Si	Sí	Sí
ISS Server	Sí	No	No	No	No	No	No

Taula 16 – Taula comparativa servidors III

3.3.2 Sistema de gestió de bases de dades (SGBD)

Com s'ha observat anteriorment en l'estudi de viabilitat, per la realització del projecte és necessari l'ús de bases de dades per tal d'emmagatzemar tota la informació que manipula l'aplicació. Un sistema de gestió de bases de dades és una agrupació de programes que serveixen per definir, construir i manipular una base de dades, per tant, l'haurem d'utilitzar en l'aplicació. Els SGBD han de satisfer les següents propietats:

- Abstracció de la informació. L'informació s'emmagatzema de forma transparent per l'usuari amb diferents nivells d'abstracció.
- Independència. El sistema ha de garantir que qualsevol canvi físic o lògic no afecti a les aplicacions que l'utilitzin.
- Consistència. Totes les dades emmagatzemades han de ser consistents, per tant el sistema ha d'actualitzar les dades cada cop que es vulguin consultar.
- Seguretat. Degut a que molta informació ha d'estar protegida, hem de garantir que la informació és consultada, actualitzada, inserida o eliminada per usuaris autoritzats a fer aquestes tasques. També s'ha de garantir que les transaccions es realitzin de forma segura complint les propietats ACID (Atomicitat, Consistència, Aïllament i Definitivitat).
- Integritat. S'ha de poder recuperar tota la informació que hagi estat perduda en cas que es produeixi una incidència.

- Recuperació. Ha de permetre fer còpies de seguretat de forma eficient per evitar les possibles pèrdues de dades.
- Control de la concurrència. Ha de garantir que l'accés simultani al SGBD, s'efectuï de forma correcta i que no hi hagi inconsistències en la informació.
- Informació sobre les dades o metadades. El sistema ha de tenir informació sobre el mateix sistema, com pot ser els usuaris.

A continuació, veurem alguns els possibles SGBD que podem trobar al mercat i analitzarem les seves característiques principals.

MySQL

MySQL és un dels SGBD més populars, es caracteritza per ser un model relacional, multifil, multiusuari, que utilitza el llenguatge SQL, és de codi obert i de distribució lliure.

La última versió estable de MySQL és la 5.1.39 i està disponible des del 04/09/2009.



Les seves principals característiques són:

- Disseny intern i portabilitat : Està dissenyat i optimitzat per donar una resposta ràpida a les nostres peticions, i en la reserva de memòria. Té una alta portabilitat ja que és compatible amb la majoria de sistemes operatius que hi ha al mercat.
- Diversitat de columnes: Disposa d'una alta varietat de tipus de dades de longitud fixa o variable.
- Sentències i funcions: Compleix els estàndards de sintaxis de sentències i funcions del llenguatge SQL estàndard, a part de tenir funcions pròpies per tal de facilitar el seu ús.
- Seguretat: Posseeix un sistema de privilegis i codis molt flexible i segur que permet la verificació basada en el HOST.
- Escalabilitat: Suporta grans bases de dades amb més de 50 milions de registres.
- Connectivitat: Es fa mitjançant Sockets TCP/IP i per tant es poden fer des de qualsevol plataforma. També dona la possibilitat d'utilitzar "MySQL Connector/ODBC" i "MySQL Connector/J" per clients que utilitzin connexions ODBC i JDBC respectivament.

Oracle

Oracle és considerat un dels SGBD més complets que existeixen, com a trets que el defineixen destaquem



que té suport sobre les transicions, és multifil, té una alta estabilitat, escalabilitat i suport multiplataforma, però és un programari privatiu i per tant es necessita llicència pel seu ús. La última versió estable de Oracle és la 11g R2 i va sortir al mercat el 01/09/2009.

Les seves principals característiques són:

- Està basada completament en un entorn Client – Servidor.
- Té un alt rendiment en les transicions i una alta portabilitat i compatibilitat, és acceptat per la majoria de les plataformes actuals.
- Compleix els estàndards SQL habituals i alguns de l'indústria com SQL-92.
- Alta disponibilitat i control de la concurrència.
- Permet la gestió de grans bases de dades, de fins a mides de Gigabytes.
- Facilitat de gestió de còpies de seguretat les dades amb una autogestió de la seva integritat.
- En quan a connectivitat té unes característiques semblants a MySQL i permet la connexió des de qualsevol plataforma al igual que per clients ODBC i JDBC.

MS SQL Server

Com els dos anteriors, Microsoft SQL Server és un SGBD relacional, també té alta escalabilitat i és multifil, però la principal diferència és que està



implementat per funcionar sobre la plataforma Windows, i com a conseqüència d'això no està tan estès com els dos anteriors.

La última versió estable de MS SQL Server és la 2008 i es va publicar el 02/08/2008.

Les seves principals característiques són:

- Allotjament d'aplicacions: Permet l'execució d'aplicacions que es troben al servidor local o un d'extern.
- Escalabilitat: Té una alta escalabilitat i d'aquesta manera maximitza la utilització del multifil i del hardware del sistema.
- Administració de les bases de dades: Un dels punts que el diferencien respecte als anteriors és que posseeix una gestió de les bases de dades molt simplificada i

intuïtiva, moltes de les seves funcions d'administració i manteniment són automatitzables.

- Alta disponibilitat: Permet configuracions d'alta disponibilitat de fins a vuit nodes en un clúster, i que altres servidors pugin recuperar l'execució d'aplicacions crítiques en cas de parades en el servei previstes o no.
- Seguretat: Té una alta seguretat i tolerància a errors, com els anteriors utilitza mètodes de xifrat de dades quan efectua les transicions.

3.3.3 Llenguatges de programació

Existeixen multitud de llenguatges que ens permeten la programació de webs dinàmiques, per tant, s'han analitzat per constatar quin d'ells és el que s'adapta millor a les caracteritzacions de l'aplicació.

A continuació s'estudiaran alguns d'aquests llenguatges i mostrarem quines són les característiques que els diferencien.

Java

És un llenguatge interpretat, ofereix un índex de reutilització de codi molt elevat, sent possible trobar moltes llibreries lliures de Java. És un llenguatge flexible i potent tot i la facilitat amb la que es programa i dels resultats que ofereix. Un dels trets que el caracteritza i que el fa una eina molt valorada a l'hora de desenvolupar aplicacions distribuïdes és el fet que és un llenguatge multiplataforma.



Per aquesta aplicació en concret la plataforma Java que ens interessa és Java Platform Enterprise Edition o Java EE, que és utilitzada per desenvolupar i executar programari escrit amb el llenguatge Java amb una arquitectura distribuïda amb nivells, basada en components de programari, i que aquest conjunt s'executi dins un servidor d'aplicacions.

Els trets més importants del Java són:

- És simple: Java ofereix gran part les funcionalitats d'un llenguatge complex, però no implementa algunes de les característiques menys usades i més complexes. Esta basat en C++.

- Orientat a objectes : Suporta las tres característiques pròpies del paradigma de la orientació a objectes: encapsulació, herència i polimorfisme.
- Distribuït: proporciona les llibreries i eines per que els programes puguin ser distribuïts, és a dir, que estiguin funcionant a varies màquines interactuant entre elles.
- Robust: realitza multitud de verificacions tant en temps de compilació com d'execució. Aquestes verificacions són sobre límits d'arrays, excepcions i byte-codes.
- D'arquitectura neutral i portable: Qualsevol màquina que implementi el sistema d'execució (run-time) pot executar codi Java, per tant té una alta portabilitat.
- És segur: les aplicacions Java resulten extremadament segures perquè no accedeixen directament a zones delicades de la memòria o del sistema i d'aquesta manera no són susceptibles a intrusions externes.
- Multifil: el seu disseny és multifil, i permet múltiples activitats simultànies en un sol procés.

PHP

El PHP és un llenguatge de programació interpretat que s'utilitza per a generar pàgines Web de forma dinàmica, està extremadament modularitzat i té un tipus de dades, sintaxi i funcions molt semblants al llenguatge de programació C. A diferència de Java és un llenguatge totalment dissenyat per la implementació de webs i per tant les seves característiques hi estan enfocades. Es pot incloure dins de codi HTML i quan aquest és interpretat s'executa el codi de tal forma que genera un codi HTML de sortida.



Les principals característiques del PHP són:

- Dóna suport per a gran quantitat de SGBD com MySQL, Oracle i MS SQL Server entre altres.
- Integració amb diverses biblioteques externes, permet generar documents PDF i analitzar codi XML.
- Ofereix una programació senzilla per la creació de pàgines Web dinàmiques de forma universal i simple.
- Posseeix un codi més fàcil d'administrar i actualitzar que altres tipus de llenguatges.

- És molt fàcil obtenir suport ja que té una gran comunitat de desenvolupadors que el recolzen al ser codi obert.
- Està en continua evolució, freqüentment surten millores i extensions del llenguatge que amplien les seves capacitats.
- Permet el processament d'informació de formularis, la manipulació de cookies i la creació de pàgines Web dinàmiques entre altres.

Active Server Pages (ASP)

L'ASP és un llenguatge semblant als vistos anteriorment, la seva principal diferència és que és un programari privatiu, i el seu codi no és obert. Permet la creació de pàgines Web dinàmiques i es va desenvolupar amb el propòsit de substituir la tecnologia CGI ja obsoleta.



ASP.NET és la seva evolució i millora moltes de les característiques del seu precursor, però continua tenint algunes desavantatges enfront dels vistos anteriorment, com per exemple, si es vol interactuar amb bases de dades només podem fer-ho des de l'ISS Server.

Els factors més importants a tenir en compte són:

- El codi ASP tanmateix com el PHP permeten ser inserits dins el codi HTML de la pàgina en qüestió.
- No és necessari l'ús de cap entorn de programació per tal de programar amb ASP amb un simple editor de text n'hi ha prou.
- Al ser un codi que s'executa al servidor i la seva sortida és en forma de codi HTML, és compatible amb tots els navegadors.
- Permet l'accés a fonts de dades ODBC mitjançant connexions del tipus DSN o sense elles.
- També dona la possibilitat d'utilitzar components que estiguin escrits amb altres llenguatges que són cridats mitjançant els guions ASP, aquests són programats habitualment amb JavaScript o VBScript.

HTML

L'HTML és el llenguatge de marcat que està més estès arreu del món. Està dissenyat per estructurar textos i relacionar-los en forma d'hipertext i també permet la inserció d'objectes tals com imatges per complementar el text. Per tal de realitzar aquesta estructuració utilitza etiquetes o TAG, que permeten definir les parts de la pàgina fins i modificar-ne la font, el text en negreta o cursiva entre altres.

El que diferencia l'HTML dels llenguatges analitzats anteriorment, és que tots ells utilitzen l'HTML per tal de donar les respostes a les peticions Web, però ell per si sol no és capaç de donar respostes dinàmiques a les peticions d'un client, només genera pàgines Web estàtiques. És per això, que quan necessitem que la resposta a les nostres peticions sigui diferent en cada cas s'ha de combinar amb un llenguatge dels anteriors.

3.3.4 Entorns de programació integrats (IDE)

Per poder programar l'aplicació és necessari tenir un entorn integrat de programació que ens ofereixi la possibilitat d'integrar gran part sinó tots els llenguatges que s'emprin finalment.

En aquest cas, també s'ha analitzat que aquests entorns integrin totes les funcions necessàries per implementar webs dinàmiques i suportar servidors com Tomcat per si finalment ens decidim a utilitzar-lo.

De les ofertes que existeixen al mercat s'han analitzat els que figuren a continuació detallant algunes de les seves principals virtuts.

Eclipse

És un entorn integrat de desenvolupament de codi obert programada principalment en Java (per tant, multiplataforma), per a desenvolupar projectes en C, C++, Java, COBOL, Python, Perl, PHP, i molts altres, sempre hi quan instal·lem els connectors corresponents per a cada llenguatge de programació. La versió necessària per aquest projecte és Eclipse Galileo, que integra tots els connectors necessaris per a la implementació de servlets. Permet la edició, compilació, depuració i execució de les aplicacions que es generen en ell.

NetBeans

A l'igual que Eclipse, NetBeans és un entorn de desenvolupament de codi obert implementat en Java, permet escriure, compilar, depurar i ejectar projectes amb qualsevol altre llenguatge de programació. També permet la integració de diferents tipus de mòduls per ampliar les seves funcionalitats.

3.4 DIAGRAMES DE CASOS D'ÚS

Aquests permeten definir el comportament del sistema al complir dels requeriments d'aquest. Aquests diagrames detallen els passos que realitza l'aplicació davant de les possibles accions que realitzi l'usuari.

3.4.1 Accés a l'aplicació

Permet a un usuari accedir al simulador i en cas que no s'hagi registrat també dóna aquesta opció.

Flux bàsic

1. Accedir a la pàgina Web de l'aplicació.
2. L'usuari introdueix el Niu la paraula de pas i el curs.
3. Validació de les dades.
4. El sistema mostra una resposta segons els permisos de cada usuari.

Flux alternatiu

En el pas 2 (usuari no registrat):

1. Accedir a la pàgina de registre.
2. Omplir dades del formulari.
3. Validació de les dades.
4. Guardar informació del usuari.
5. Mostrar missatge de registre correcte.
6. Redirigir a la pàgina d'accés.

En el pas 3 (dades incorrectes):

1. Es mostra un missatge d'error i es demana a l'usuari que torni a escriure correctament les dades.

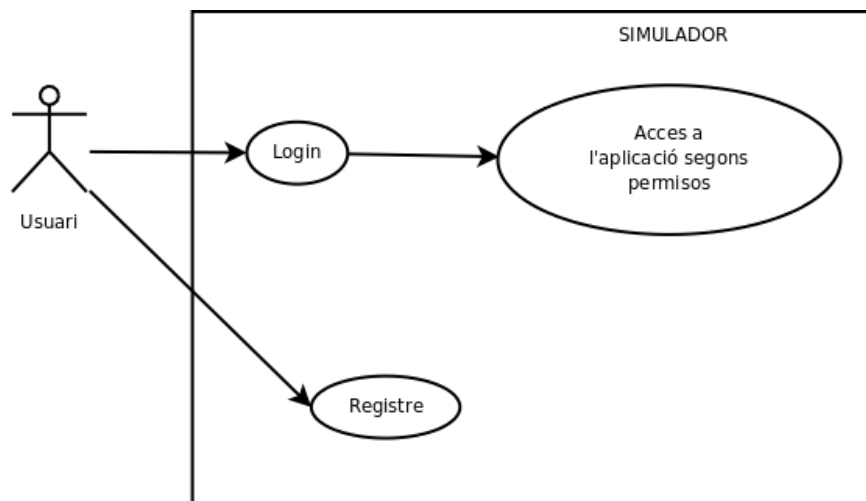


Figura 5 - Cas d'ús accedir

3.4.2 Accions administrador i/o professor

Per poder realitzar aquestes accions és necessari que l'usuari hagi accedit a l'aplicació i disposi de permisos de d'administrador o professor.

- Administrar curs:

Permet als usuaris l'administració del curs. S'hi poden realitzar les accions de crear, modificar i eliminar cursos.

Flux bàsic

1. Accedir al menú de configuració de cursos.
2. Seleccionar el curs que es vol administrar.
2. Escollir l'acció a realitzar sobre ell.
3. Realitzar l'acció o confirmar-la.
4. Actualitzar el nou estat a la base de dades.
5. Mostrar missatge amb la confirmació de l'acció.

Flux alternatiu

En el pas 3 (problemes al realitzar l'acció)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

- Administrar usuari:

Permet als usuaris l'administració dels usuaris del curs. S'hi poden realitzar les accions de crear, modificar i eliminar usuaris.

Flux bàsic

1. Accedir al menú de configuració d'usuaris.
2. Seleccionar l'usuari que es vol administrar.
3. Escollir l'acció a realitzar sobre ell.
4. Realitzar l'acció o confirmar-la.
5. Actualitzar el nou estat a la base de dades.
6. Mostrar missatge amb la confirmació de l'acció.

Flux alternatiu

En el pas 4 (problemes al realitzar l'acció)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

- Administrar granja:

Permet als usuaris l'administració de les granges. S'hi poden realitzar les accions de crear, modificar i eliminar granges.

Flux bàsic

1. Accedir al menú de configuració de granges.
2. Seleccionar la granja que es vol administrar.
3. Escollir l'acció a realitzar sobre ella.
4. Realitzar l'acció o confirmar-la.
5. Actualitzar el nou estat a la base de dades.
6. Mostrar missatge amb la confirmació de l'acció.

Flux alternatiu

En el pas 4 (problemes al realitzar l'acció)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

- Gestionar còpies de seguretat:

Permet als usuaris la creació de noves còpies de seguretat o la càrrega dels que ja existeixen.

Flux bàsic

1. Accedir al menú de gestió de còpies de seguretat.
2. Escollir l'acció de crear o carregar.
3. Realitzar l'acció seleccionada.

4. Visualitzar un missatge confirmant l'acció.

Flux alternatiu

En el pas 3 (problemes en realitzar l'acció)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

- Obtenir vista de granges:

Permet al usuari tenir una visió general de les granges del curs així com visualitzar-les individualment i realitzar accions com assignar alumnes o les que pot realitzar un alumne sobre elles.

Flux bàsic

1. Accedir al menú de visualització de granges.
2. Escollir la granja desitjada.
3. Realitzem les accions pertinents (afegir alumne, passar dia, inseminar,...).
4. Actualitzar el nou estat a la base de dades.
5. Obtenim un missatge de confirmació de l'acció.

Flux alternatiu

En el pas 3 (problemes en realitzar l'acció)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

En el pas 4 (error d'actualització de les bases de dades)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

- Administrar simulador:

Permet a l'usuari administrador el control de les variables per defecte del simulador.

Flux bàsic

1. Accedir al formulari d'edició.
2. Modificació dels paràmetres.
3. Enviar a la base de dades.
4. Obtenim un missatge de confirmació de l'acció.

Flux alternatiu

En el pas 3 (problemes en la base de dades)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

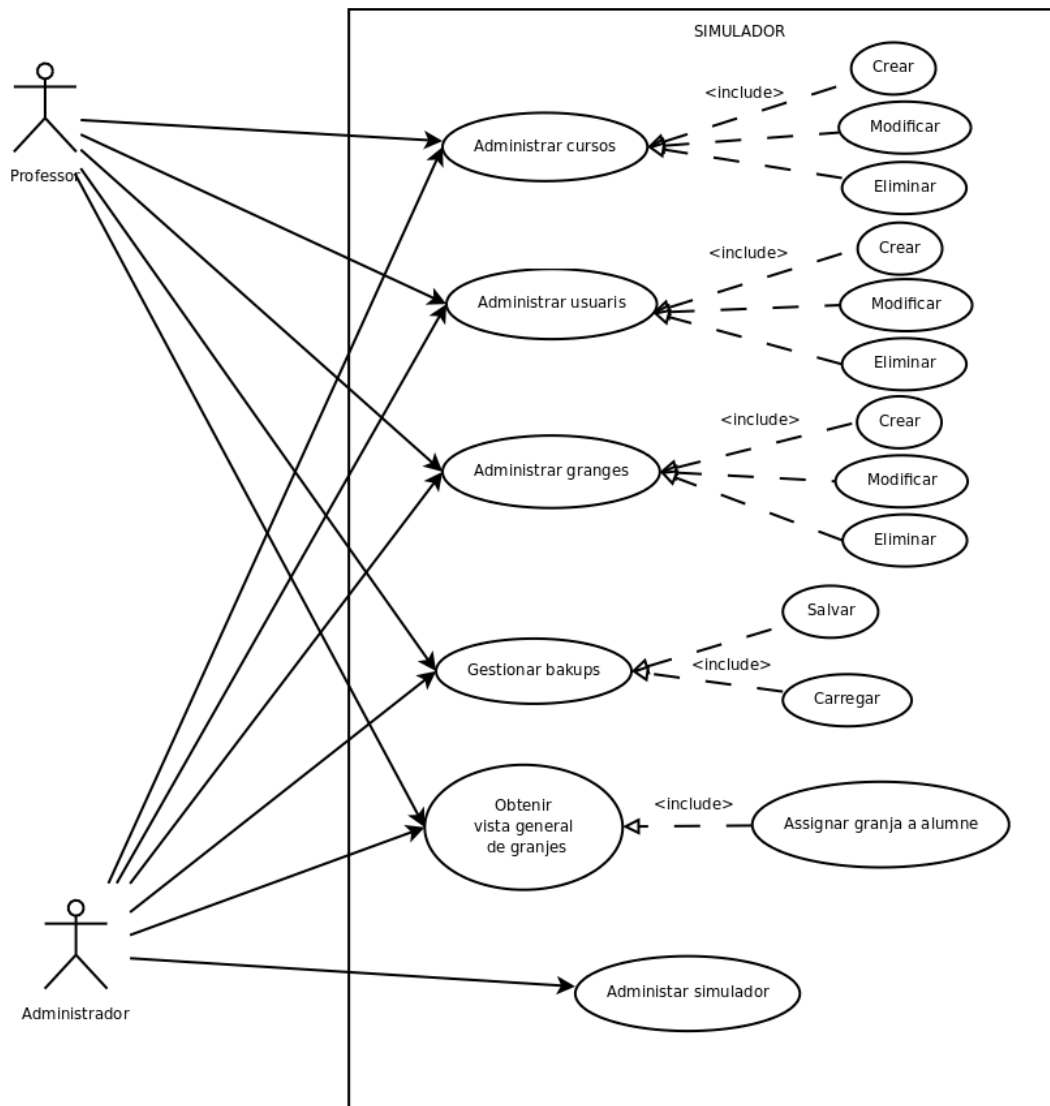


Figura 6 - Casos d'ús administrador i professor

3.4.3 Accions alumne

Aquestes són les accions que realitzen els alumnes sobre la seva granja i les vaques que la componen. Els usuaris administrador i professors, també hi poden accedir mitjançant la vista general de granjes.

- **Obtenir vista de granja:**

Permet a l'usuari tenir una vista general de la granja en qüestió i té la possibilitat de realitzar l'acció de transcórrer un dia o de seleccionar una vaca per visualitzar-la de forma específica. És la vista que veurà l'alumne a l'accedir al simulador.

Flux bàsic

1. Realitzar acció sobre la granja.
2. Actualitzar el nou estat a la base de dades.

3. Obtenim missatge de que s'ha realitzat correctament l'acció

Flux alternatiu

En el pas 1 (error al executar l'acció)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

En el pas 2 (error d'actualització de les bases de dades)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

- Obtenir vista de vaca:

Permet una visió específica de cada vaca que compona una granja, dins d'aquesta s'hi poden realitzar les accions d'inseminar, secar i actualitzar l'estat de la vaca en qüestió.

Flux bàsic

1. Realitzar acció sobre la vaca.
2. Actualitzar el nou estat a la base de dades.
3. Obtenim missatge de que s'ha realitzat correctament l'acció

Flux alternatiu

En el pas 1 (error al executar l'acció)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

En el pas 2 (error d'actualització de les bases de dades)

1. Es mostra un missatge d'error especificant el problema que ha succeït.

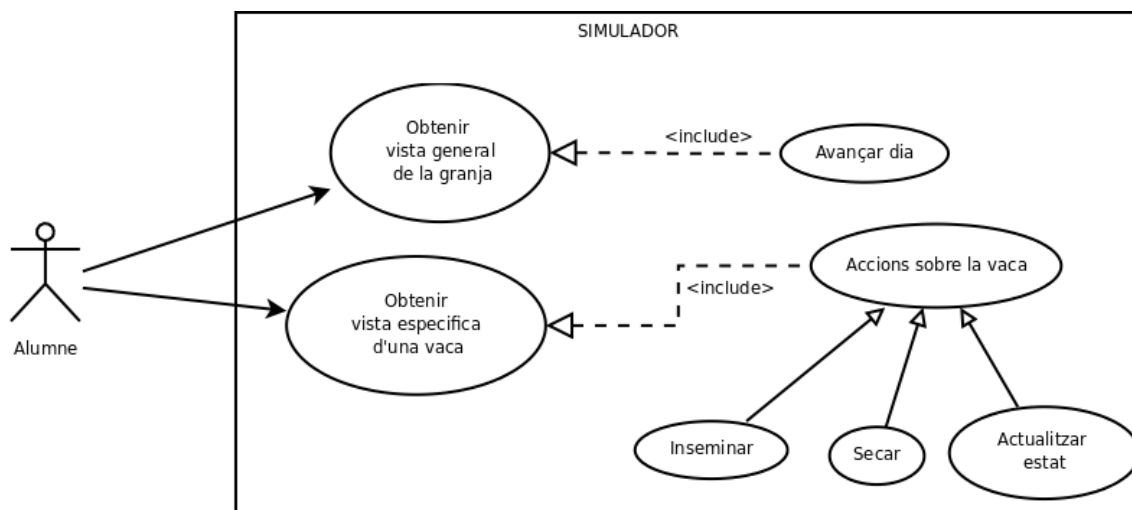


Figura 7 - Casos d'ús alumne

Capítol 4 - DISSENY

Centrant-nos en el disseny del projecte i després de realitzar l'anàlisi de totes les possibles opcions per la seva implementació, creiem que l'esquema final del projecte serà de la forma que es mostra en la figura 8, que consta del servidor d'aplicacions que interactua amb el SGBD i el servidor Web que interactua amb els clients mitjançant el navegador Web.

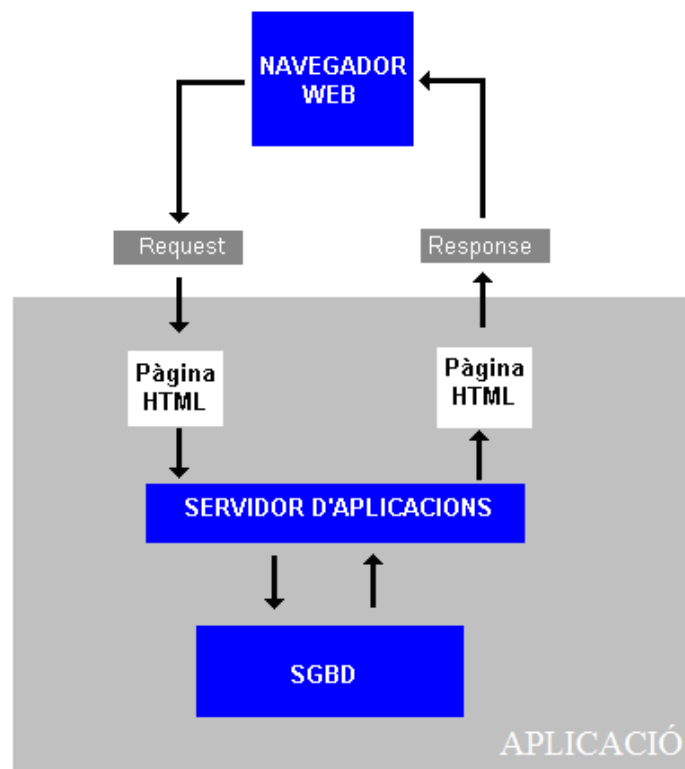


Figura 8 - Aplicació

També explicarem quines de les opcions s'han escollit en quan a eines per la implementació de l'aplicació basant-nos bàsicament en el fet que totes elles són de programari lliure, i en alguns casos pel fet que ja es disposava de certa experiència en el seu ús.

A continuació detallarem les decisions de disseny que hem pres en cadascuna de les parts que configuren l'aplicació, tan en la infraestructura, com en el l'elecció dels llenguatges i entorns de programació i les pàgines Webs resultants.

4.1 DISSENY DE LA INFRAESTRUCTURA BASE

El disseny de la plataforma inicial consta del servidor Web sobre el que muntarem el servidor d'aplicacions, i per tant farem l'elecció basant-nos en un bon rendiment general del servidor però tenint en compte totes les restriccions que tenim en el projecte. S'han descartat els següents servidors Web per les raons que mostrem a continuació.

L'**AOLServer** no s'ha escollit aquesta opció únicament pel desconeixement total de l'entorn i per la dificultat de trobar informació i manuals sobre el seu funcionament i configuració. Així i tot aquesta seria una opció totalment vàlida. En quan a **IIS (Internet Information Server)** es va desestimar perquè el sistema operatiu que utilitzem és Ubuntu, de totes formes si l'aplicació s'hagués implementat en un entorn Windows molt probablement no hagués estat l'elegida per les seves mancances en quan a seguretat. Si analitzem "**Servlet Engines**" i "**Java Application Server**" es poden utilitzar directament sense el suport de cap altre servidor, sempre que tinguem el JDK ("Java Development kit") corresponent, però normalment s'utilitza sobre un servidor com Apache o AOL per tal de millorar l'eficiència alhora de retornar pàgines HTML estàtiques i aquesta ha estat la opció escollida en l'aplicació.

Així doncs, en el problema que ens ocupa, la elecció ha estat muntar el servidor **Tomcat** sobre el servidor **Apache** per tal que millori el seu rendiment. També per la facilitat d'accés als manuals d'Apache i que mitjançant el paquet de programari **XAMPP** és de fàcil instal·lació i configuració. XAMPP és un paquet de programari lliure que conté el servidor HTTP Apache, base de dades de MySQL i eines necessàries per utilitzar el PHP i el llenguatge de programació Perl. El programa es llença sota el GNU de Llicència Pública General i d'un servidor Web, d'ús fàcil, capaç de servir pàgines dinàmiques.

Sobre aquesta plataforma inicial s'ha de muntar el Sistema Gestor de la Base de Dades (SGBD) que ens permetrà la gestió de les dades de l'aplicació. Aquesta elecció s'ha basat sobretot en la compatibilitat amb el sistema operatiu i sobre si el seu programari és lliure o no.

Primerament mostrem els motius pels que s'han descartat els següents SGBD. En primer lloc, **MS SQL server** presenta el principal problema que funciona únicament sobre el sistema operatiu Windows, a part no és programari lliure i per tant, no complia

els requeriments del projecte. En quan a **Oracle** és el més complex de tots, està destinat a grans empreses i volums d'informació més grans, per tant, seria una bona elecció però, té el problema que s'ha de pagar llicència pel seu ús.

Per tant el principal problema que trobem en quan a l'elecció del SGBD és que la majoria d'ells són programari propietaris i no compleixen el requisit que el programari utilitzat en el projecte sigui d'utilització lliure. A conseqüència d'això, la nostra elecció ha estat **MySQL** perquè ens dóna totes les funcionalitats necessàries i és programari lliure.

4.2 SELECCIÓ DELS LENGUATGES I IDE

Un cop haguérem escollit la plataforma de muntatge la primera idea era la utilització de PHP per la realització de l'entorn Web i per la interacció amb les bases de dades, però després del primer anàlisi es va optar per la utilització dels Servlets de Java, que tot i que l'experiència en aquest cap era nul·la, doncs pel tipus d'aplicació que anàvem a realitzar la veiem més idònia, ja que havíem de processar molta informació i resultava més senzill realitzar tota la programació en Java.

A conseqüència d'això alguns dels llenguatges de programació analitzats van quedar descartats. Fer l'aplicació amb **PHP** era factible, ja que és un bon llenguatge per a programació de webs dinàmiques, però com que aquesta aplicació ha de processar grans volums d'informació es va descartar perquè resultava bastant més complicat alhora de programar. Si analitzem l'altre opció que era **ASP** observem que també és un llenguatge de programació de webs dinàmiques però està enfocat a entorns Windows i és programari propietari, així que es va refusar per no complir els requeriments del projecte.

Per tant observant els requeriments que té la nostra aplicació varem optar per l'elecció del Java Servlet ja que ens ofereix les funcionalitat necessàries, és programari lliure i permet la programació en Java que és molt més senzilla de realitzar.

Un cop clara l'elecció dels Servlets de Java teníem dos entorns de programació possibles per la seva implementació, **Eclipse** i **NetBeans** i finalment l'opció escollida ha estat Eclipse. Com s'ha observat en la fase d'anàlisi ambdues aplicacions tenen unes

característiques molt semblants i la nostre decisió s'ha basat principalment amb que ja havíem utilitzat amb anterioritat aquest entorn.

4.3 DISSENY DE LES PÀGINES WEB

El disseny de totes les pàgines Web es basa en una estructura com la que es mostra en la figura següent, que consta d'una àrea de títol centrada a la part superior amb una imatge de fons i el títol imprès sobre ella , una àrea de menú de transició ràpida entre pàgines a la part dreta superior i l'àrea central de la pàgina s'utilitza per la informació pròpia de cadascuna d'aquestes.

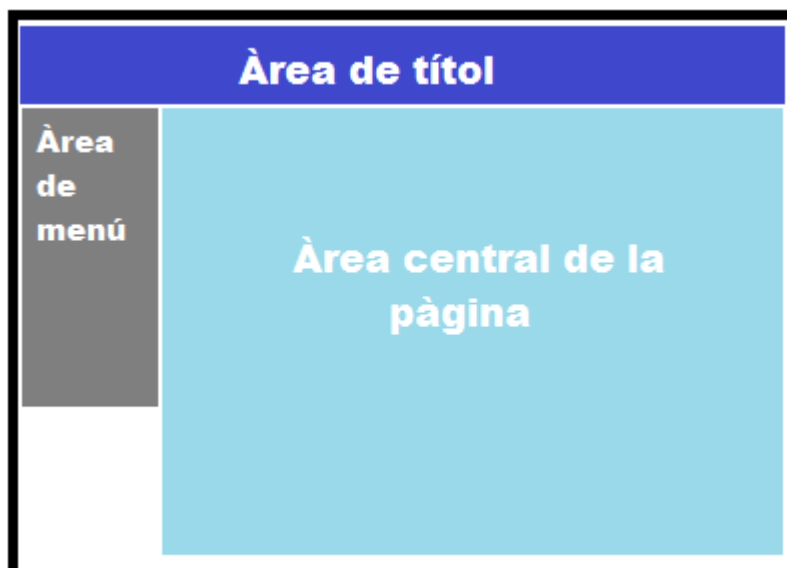


Figura 9 - Disseny general de les pàgines Web

L'única pàgina que té un format diferent al comentat, és la pàgina d'inici de l'aplicació que consta igualment d'una àrea de títol però al no ser necessària una àrea de menú, s'ha utilitzat la pàgina sencera per la distribució dels seus elements.

En aquest punt i per millorar la integritat de l'aplicació s'ha escollit la utilització del llenguatge Javascript, que permeten la validació de tots els formularis Web, i que per tant prevenen a questa de possibles errors derivats d'un ús incorrecte dels formularis de les pàgines Web. JavaScript és un llenguatge script basat en el concepte de prototip, s'utilitza sobretot per la realització de programes que s'encarreguen de realitzar accions dins l'hàbit d'una pàgina Web.

També utilitzarem els CSS i la biblioteca JQuery que milloren la interacció amb l'usuari i permeten millores visuals considerables i fàcils de realitzar. Els fulls d'estil en cascada (CSS) és un llenguatge de fulls d'estil utilitzat per descriure la semàntica de presentació (l'aspecte i format) d'un document escrit en un llenguatge de marques com l'HTML. JQuery és una biblioteca o framework de Javascript, que permet simplificar la manera d'interactuar amb els documents HTML, manipular l'arbre DOM, administrar events, desenvolupar animacions i afegir interacció amb la tecnologia AJAX en pàgines Web.

Per últim i degut al fet que s'han de poder generar gràfiques de la producció de llet de cada vaca utilitzarem les biblioteques JCommon i JFreeChat, que són necessàries per generar gràfiques a partir de valors que es llegeixen des d'una base de dades o s'introdueixen directament des d'una aplicació, també permeten la seva exportació a format d'imatge per ser utilitzats posteriorment des d'una Web o aplicació.

4.4 DIAGRAMA D'ENTITAT – RELACIÓ

En el següent diagrama es dissenya l'estructura bàsica de les entitats i relacions que configuraran la base de dades del simulador.

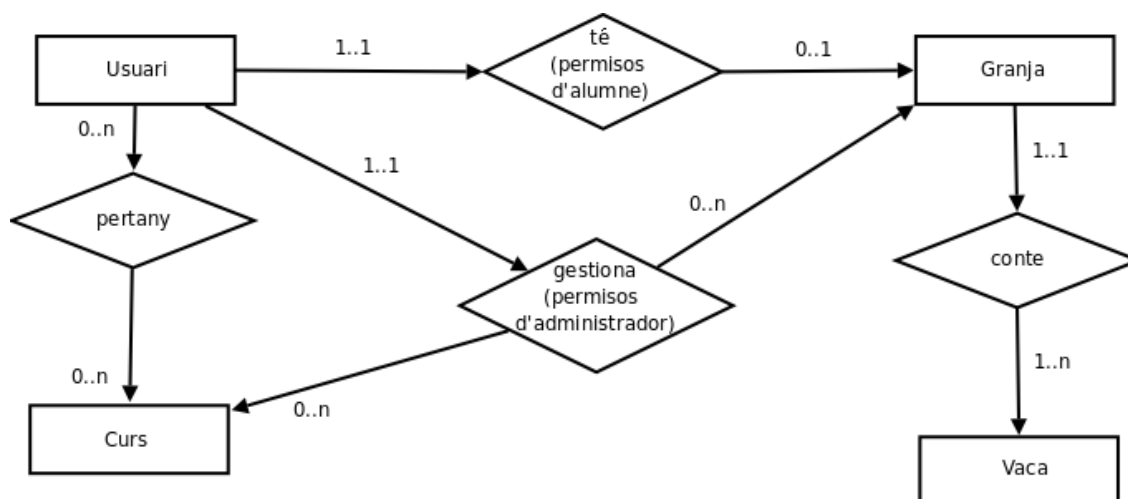


Figura 10 - Diagrama d'entitat relació

Entitat **Usuari**: Conté la informació específica de cada usuari, juntament amb els permisos i curs on pertany. Quan es tracta d'un alumne té assignada una granja i si és administrador o professor té totes les possibilitats de gestió sobre les altres entitats.

Entitat **Curs**: Emmagatzema la informació relativa a un curs.

Entitat **Granja**: Entitat usada per guardar la informació d'una granja així com informació sobre les vaques que s'hi creen.

Entitat **Vaca**: S'hi desa tota la informació de la vaca, entre ells la granja on pertany i tots els paràmetres necessaris per la seva simulació.

4.5 DIAGRAMA DE CLASSES

En el diagrama de classes dissenyarem les classes que s'hauran d'implementar dins l'aplicació així com les variables i mètodes que el configuren.

També es definirà el tipus de relació que hi ha entre aquestes i el factor de numeració que les relaciona per tal de facilitar en la implementació el tipus d'estructura que s'utilitzarà en cada cas.

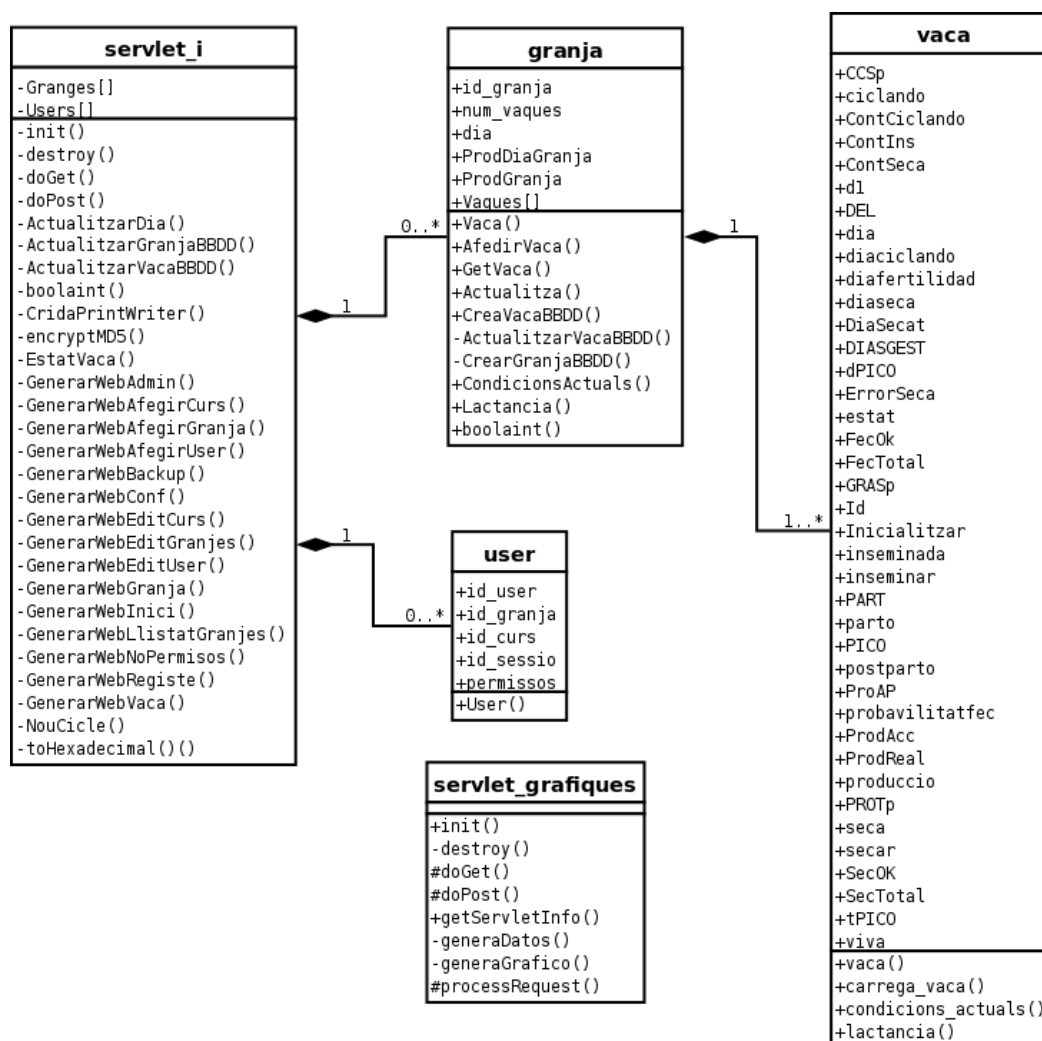


Figura 11 - Diagrama de classes

4.6 PLANIFICACIÓ AJUSTADA

Degut a que les hores dedicades a la realització del projecte no podran ser les que es van estimar durant l'estudi inicial, s'ha refet la planificació modificant el calendari laboral del cap del projecte, analista, programador i tècnic de proves.

També s'ha modificat alguna de les tasques per ajustar-la més al contingut del projecte, i el temps de dedicació a cadascuna d'aquestes. En concret s'ha augmentat el temps de dedicació a l'anàlisi i disseny, i especificat les seves activitats, ja que a causa del desconeixement dels servlets, s'ha dedicat un temps extra en l'aprenentatge d'aquests. També en la fase de desenvolupament s'han detallat més les activitats així com el temps de dedicació a aquestes.

La resta de les activitats no s'han modificat respecte l'estudi inicial.

Calendari del projecte

El projecte es desenvoluparà d'octubre de 2009 a setembre de 2010 amb una dedicació de 6 a 8 hores setmanals. El total d'hores dedicades al projecte serà de 310 hores, això ocasionarà un augment de 57 hores. Data començament continuarà sent el 1 d'octubre de 2009, però la finalització es posposarà fins el dia 5 de setembre de 2010.

Cost del projecte:

Degut a l'augment d'hores dedicades al projecte, el cost total del projecte serà de 13.208,5€ per tant hi haurà un augment de 535,20€ sobre la planificació inicial.

Redistribució de tasques del projecte:

Descripció de l'activitat	Duració	Cost	Data inici
Inici del projecte: assignació i matriculació del projecte	2,5d	200,00 €	24/10/09
Planificació	11 hores	1.100,00 €	07/11/09
Estudi de viabilitat	10 hores	1.000,00 €	01/11/09
Aprovació Estudi Viabilitat (Punt de control)	1 hora	100,00 €	07/11/09
Anàlisi de l'aplicació	43 hores	2.175,00 €	26/12/09
Anàlisi de requeriments	8 hores	400,00 €	14/11/09

Anàlisi de servidors Web	8 hores	400,00 €	22/11/09
Anàlisi de SGBD	8 hores	400,00 €	05/12/09
Anàlisi de llenguatges i entorns de programació	8 hores	400,00 €	12/12/09
Anàlisi d'eines addicionals	6 hores	300,00 €	19/12/09
Documentació de l'anàlisi	3 hores	150,00 €	20/12/09
Aprovació de l'anàlisi (Punt de control)	2 hores	125,00 €	26/12/09
Disseny de l'aplicació	32 hores	1.171,00 €	23/01/10
Disseny de la infraestructura	6 hores	236,00 €	02/01/10
Disseny del servlet i d'interfície Web	5 hores	230,00 €	03/01/10
Disseny de diagrames i informació addicional	8 hores	230,00 €	10/01/10
Disseny de les proves (test)	8 hores	200,00 €	17/01/10
Documentació del disseny	3 hores	150,00 €	23/01/10
Aprovació del disseny (Punt de control)	2 hores	125,00 €	23/01/10
Desenvolupament de l'aplicació	125 hores	3.750,00 €	19/06/10
Creació i configuració de la infraestructura	15 hores	450,00 €	07/02/10
Creació i configuració base de dades	10 hores	300,00 €	20/02/10
Creació i configuració del servlet i del simulador	80 hores	2.400,00 €	23/05/10
Creació i maquetació de la interfície Web	20 hores	600,00 €	19/06/10
Test i proves	21 hores	632,50 €	11/07/10
Proves	15 hores	375,00 €	04/07/10
Documentació de desenvolupament i test	5 hores	150,00 €	10/07/10
Aprovació del desenvolupament i proves (Punt de control)	1 hores	107,50 €	11/07/10
Implantació	19 hores	580,00 €	31/07/10
Instal·lació	5 hores	220,00 €	17/07/10
Proves reals	14 hores	360,00 €	31/07/10
Generació de documents (memòria del projecte)	30 hores	3.000,00 €	29/08/10
Tancament del projecte	1 hora	100,00 €	29/08/10
Defensa del projecte	5 hores	500,00 €	05/09/10

Figura 12 - Redistribució de tasques

Els resultats obtinguts els presentem en el següent diagrama de Gantt.

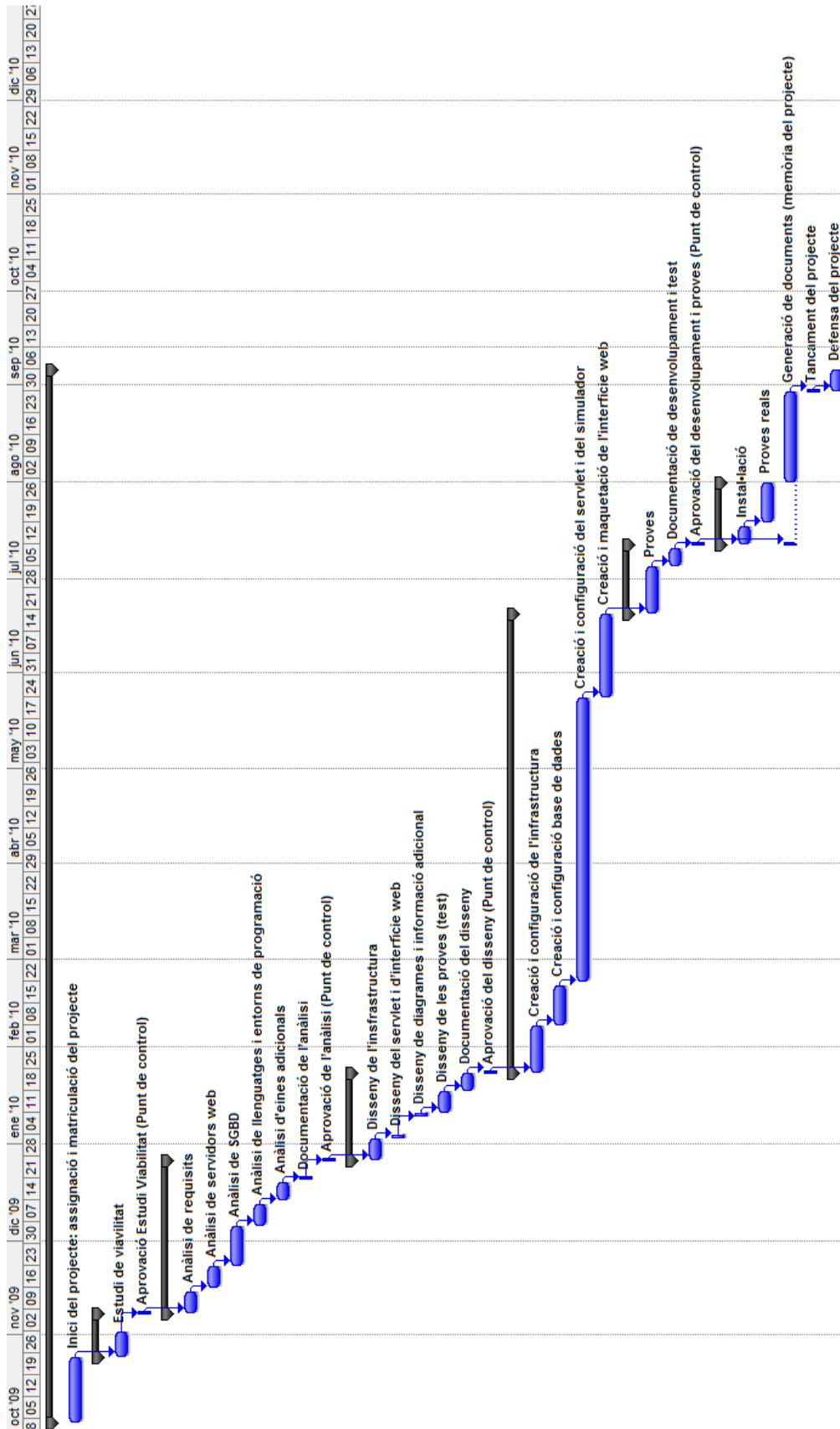


Figura 13 - Diagrama de Gantt ajustat

Capítol 5 - IMPLEMENTACIÓ

En aquest capítol descriurem els passos que s'han seguit per la implementació del projecte.

Començarem per la creació i configuració de la infraestructura base on hi muntarem l'aplicació i després el procés de creació de les bases de dades i l'explicació de cadascuna d'elles.

Un cop haguem vist aquests dos primers punts continuarem amb la implementació del servlet base i sobre ell explicarem també la creació del simulador. Finalment, s'analitzarà també la creació del servlet de gràfiques i de interfície Web que ens permetrà interactuar amb l'aplicació.

5.1 CREACIÓ I CONFIGURACIÓ DE LA INFRAESTRUCTURA

Com s'ha explicat en el capítol anterior l'aplicació servlet que es vol implementar, necessita d'una infraestructura d'Apache, MySQL i Tomcat per tal de funcionar, a més a més, de l'entorn de programació Eclipse per la seva implementació, tot muntat sobre un sistema operatiu base Ubuntu 9.04.

En aquest punt explicarem més detalladament la instal·lació i configuració de tots aquests components. Per tal de simplificar una mica tota la instal·lació i configuració del servidor Apache s'ha optat per instal·lar el paquet de programari Xampp versió 1.7.2, que inclou Apache 2.2.11 i MySQL 5.1.30 dues versions que com s'ha analitzat anteriorment són totalment estables.

Gràcies a la simplicitat de Xampp, s'ha reduït molt tota la instal·lació i posterior configuració, i mitjançant la Web de configuració i seguretat en pocs minuts hem realitzat la configuració. A continuació mostrem la Web de configuració després d'ajustar-la.



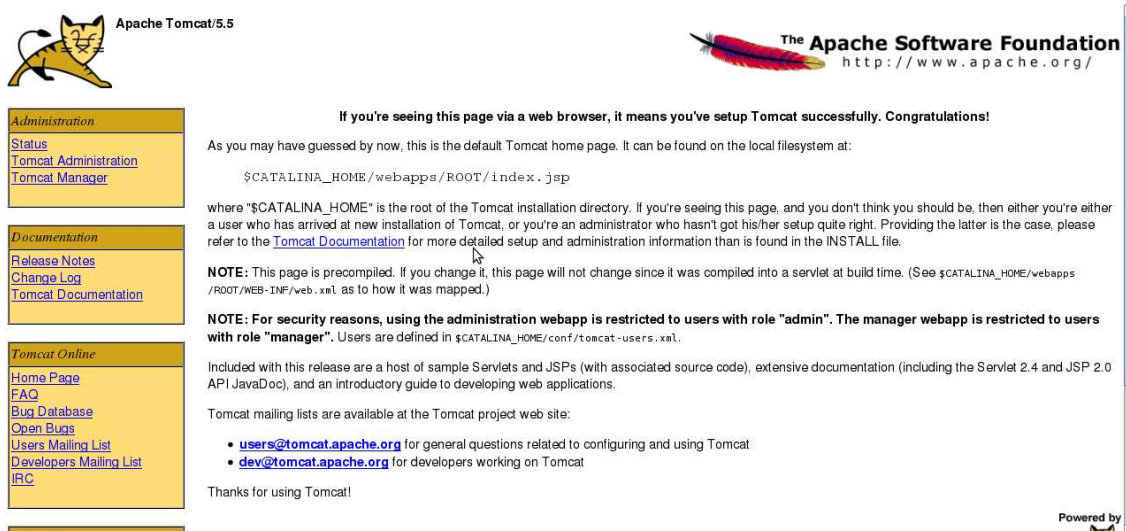
Figura 14 - Instal·lació de XAMPP

El següent pas ha estat la instal·lació del servidor Tomcat amb la versió 5.5 ja que en un principi es va intentar amb la 6 però donava errors en la seva posada en funcionament. Tomcat és el que ens permetrà emmagatzemar els servlets que contindran les pàgines de l'aplicació. No s'ha configurat després de la instal·lació, perquè diversos dels documents que ens van servir de suport per muntar la infraestructura deien que s'havia de fer un cop instal·lat el entorn Eclipse i així es va fer.

Així doncs l'acció posterior a Tomcat ha estat l'entorn de programació Eclipse, que és un dels punts que més problemes ens va ocasionar.

D'Eclipse n'existeixen moltes versions cadascuna enfocada a un tipus d'aplicació diferent, i fins que no es va trobar la versió adequada per la implementació de servlets, versió Galileo, es van anar realitzant proves amb cada versió que anàvem adquirint, ja que la informació sobre l'ús de cada una d'elles va resultar força difícil de trobar.

Un cop es va veure que aquesta era la versió correcta, es va iniciar la configuració del servidor Tomcat, a través de la Web de configuració que mostrem a continuació, que de forma intuïtiva i gràfica va ser configurat ràpidament.



Apache Tomcat/5.5

The Apache Software Foundation
http://www.apache.org/

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
$CATALINA_HOME/webapps/ROOT/index.jsp
```

where "\$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See \$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in \$CATALINA_HOME/conf/tomcat-users.xml.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Tomcat project web site:

- users@tomcat.apache.org for general questions related to configuring and using Tomcat
- dev@tomcat.apache.org for developers working on Tomcat

Thanks for using Tomcat!

Powered by

Figura 15 - Instal·lació de Tomcat

A continuació, es van realitzar les primeres proves de funcionament de tota la infraestructura amb alguns servlets molt simples d'exemple i es van acabar de configurar alguns detalls pel seu correcte funcionament.

Un cop testada la seva funcionalitat es va continuar amb la connectivitat d'Eclipse amb el SGBD MySQL mitjançant el connector mysql-connector-java-5.1.10 i va ser una de les parts on també van sorgir complicacions. El principal problema que vàrem trobar arribats a aquet punt és que només s'inclouïa el connector al Java Build Path, i també era imprescindible la seva inclusió dins les llibreries de la Web, concretament `WebContent/WEB-INF/lib`.

Finalment es van realitzar unes proves simples d'inserció a una base de dades, i un cop comprovat el seu perfecte funcionament mitjançant l'eina PhpMyAdmin, ja teníem tota la infraestructura base muntada per passar a la següent fase, la creació del servlet base.

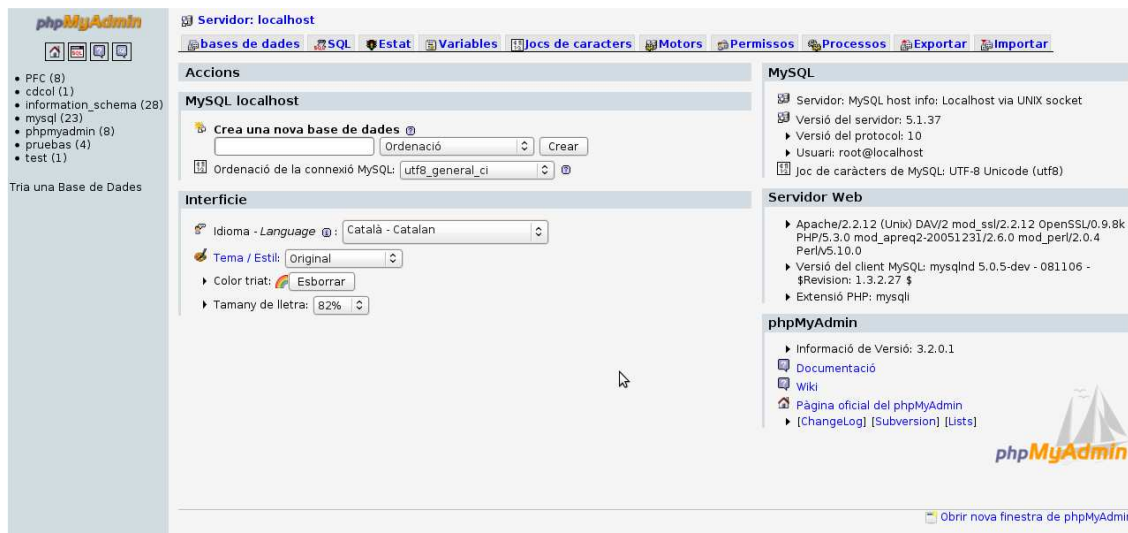


Figura 16 - Entorn PhpMyAdmin

5.2 CREACIÓ DEL SERVLET BASE

Un cop es va tenir tota la infraestructura base per la implementació del servlet vàrem començar a aprofundir en el funcionament dels servlets, ja que mai en anterioritat se n'havia implementat cap. Es van veure quines són les parts principals que el formen, i es van anar afegint al servlet simple que tenien com a base. Finalment la estructura general va ser la següent:

- Variables globals

```
private final String UrlBaseDades = "jdbc:mysql://127.0.0.1:3306/PFC";
private Connection Connexio = null;
private Statement Declaracio = null;
Calendar calendari = Calendar.getInstance();
ArrayList<Granja> Granjes = new ArrayList<Granja>();
ArrayList<User> Users = new ArrayList<User>();
```

Aquestes són les variables globals del servlet i que per tant compartiran tots els usuaris que estiguin connectats al mateix instant. Totes ells són variables que no comporten problemes de concurrència, i que s'utilitzen per la connexió amb la base de dades, per obtenir la data del dia actual i les dues últimes per emmagatzemar els usuaris i granges actius en l'aplicació.

- Constructor del servlet

```
public void init(ServletConfig config) throws ServletException
```

Aquest mètode serveix per la inicialització del servlet cada vegada que es posa en funcionament. És l'encarregat de fer la inicialització per defecte d'un servlet i també

l'utilitzem per a realitzar la connexió amb la base de dades, la creació en cas que no existeixin de les taules necessàries per l'aplicació i les insercions a elles dels paràmetres per defecte i l'usuari administrador en cas que s'hagin creat les taules des de zero.

- Destructor del servlet

```
public void destroy()
```

El mètode `destroy` és el que utilitza el servlet cada cop que es tanca la connexió.

S'encarrega del tancament de totes les instàncies del servlet, de les bases de dades, així com de l'actualització dels índex globals dels usuaris per que siguin els correctes quan es torni a arrancar l'aplicació.

- Mètodes `doGet` i `doPost`

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException
```

Aquests són els mètodes utilitzats pel servlet per donar resposta a les peticions que envii un client. En el cas concret d'aquesta aplicació s'usa el mètode `doPost()` i el mètode "`doGet()`" és redirigit al primer.

És el que conté l'esquelet general de la interfície Web i el que dona una resposta diferent per cada tipus de petició d'entrada que rep el servlet.

També és l'encarregat de reconèixer els usuaris que envien la petició mitjançant variables de sessió, que són guardades dins la base de dades al realitzar el login d'usuari i comparades cada vegada que se n'envia, i d'aquesta manera el tenim identificat mentre no caduqui la sessió. En ells es realitza també la comprovació dels permisos de cada usuari i en cas de no tenir-ne els necessaris no se'l deixa accedir a la Web en qüestió.

5.3 CREACIÓ DE LES BASES DE DADES

Després de tenir clara tota l'estructura dels servlets, es va realitzar un repàs general a tota la sintaxi de MySQL i la específica en Java per començar a crear les bases de dades necessàries des del servlet base que havíem creat.

La base de dades es va crear inicialment de forma senzilla, amb el mínim nombre de registres per taula per tal de facilitar la resolució dels errors que ens anaven sorgint, i quan requeríem l'ús de més camps s'anaven inserint un a un per la mateixa raó esmentada anteriorment, simplificar la feina de localització d'errors.

El diagrama final després de la implementació de l'aplicació és el que es presenta en la figura 16 amb totes les relacions requerides per l'aplicació.

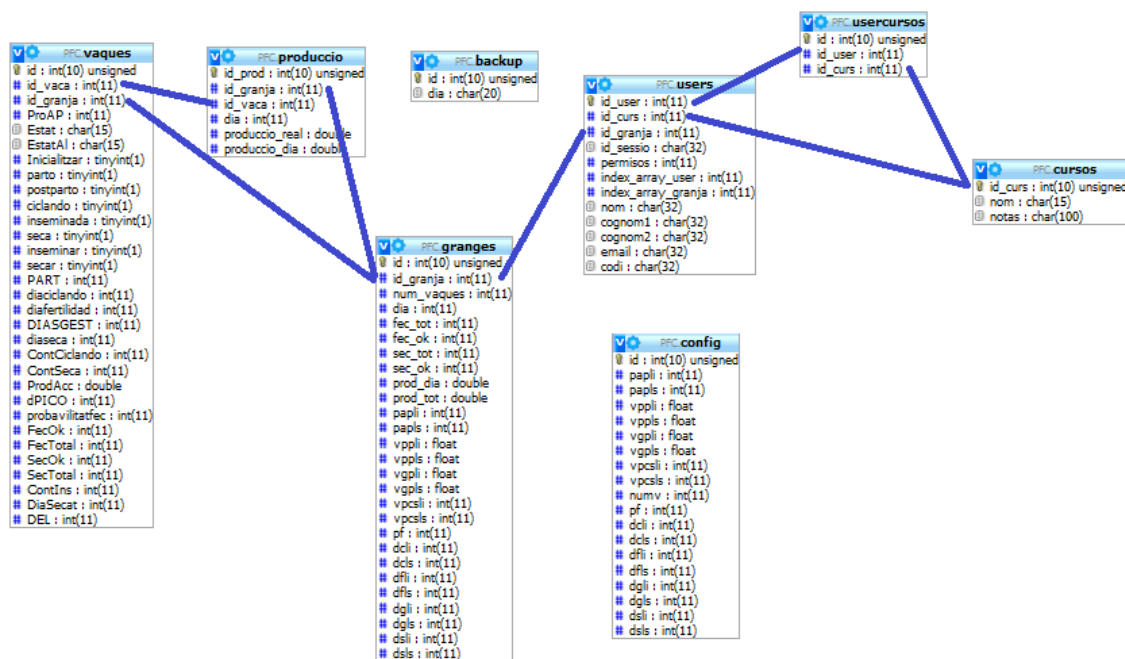


Figura 17 - Diagrama de la base de dades

Taules:

config

Aquesta taula, ens serveix per guardar totes les variables per defecte de l'aplicació, aquestes només són modificables per l'administrador.

La clau primària d'aquesta taula és id, però no es utilitza per l'aplicació i només s'ha creat per l'obligació de existir-ne una.

backup

La taula backup és on desem el nom de les còpies de seguretat generades per l'administrador i/o professors i també ens serveix per saber el nom de l'arxiu en que es troba la còpia sol·licitada quan volem carregar-la.

En aquest cas, la clau primària utilitzada és id i l'utilitzem per referir-nos a ella dins de l'aplicació.

cursos

Els diferents usuaris són creats dins d'un curs existent, i aquesta taula és utilitzada per emmagatzemar la informació de cadascun dels cursos que creen l'administrador i/o professors.

La taula cursos té com a clau primària `id_curs` i la seva utilitat és d'identificar el curs en qüestió a la vegada que també és clau forania d'altres taules.

usercursos

És la taula necessària per relacionar els usuaris amb el curs on pertanyen.

La necessitat d'utilitzar una taula suplementària per implementar aquesta relació a estat que un usuari pot pertànyer a més d'un curs i per tant no era suficient el camp `id_curs` en la taula `users`, i aquest ha estat utilitzat únicament per gravar el curs que s'utilitza en el moment de l'execució.

La taula `usercursos`, com en algunes vistes anteriorment la seva clau primària, `id` no és utilitzada per l'aplicació i només ha estat creada per qüestions de coherència de les bases de dades. Cal remarcar que els dos camps restants, `id_users` i `id_curs` són claus foranies de les taules `users` i `cursos` respectivament.

users

Els usuaris de l'aplicació estan guardats dins la taula `users`, és on s'insereix tota la informació personal d'un usuari i també la relativa al funcionament de l'aplicació, com els índex d'array d'usuari i granja per saber en tot moment on es troba aquest usuari i la seva granja dins l'aplicació, que al ser dinàmica en cada connexió la càrrega en la posició final de les respectives arrays.

En aquest cas la clau primària de la taula, `id_user` és directament el NIU (numero d'identificació universitari) de l'usuari, que al ser un identificador únic és totalment vàlid per aquest camp. Com que cada usuari té assignada una granja (en el cas dels professors o administrador si la volen visualitzar), el camp `id_granja` serà una clau forania que relacioni l'usuari amb la granja assignada o visualitzada.

granges

Taula on es desa la informació relativa a la granja d'un usuari.

Hi ha tota la informació general de la granja, una array amb les vaques que hi pertanyen i tots els paràmetres per la seva evolució. També hi ha camps que són utilitzats per proporcionar la informació necessària per l'avaluació que ha de realitzar el professor a l'alumne.

La seva clau primària és `id`, tot i que en un inici era `id_granja` però per qüestions d'una possible ampliació que implicava la repetició d'entrades amb una mateixa `id_granja` va ser modificada.

vaques

Totes les vaques que pertanyen a una granja són arxivades dins la taula `vaques`. En ella hi figuren totes les dades utilitzades pel simulador per evolucionar els estats i la producció de cada vaca, així com els identificadors de la granja on pertany i les accions que realitzem sobre ella.

En quan a la clau primària de la taula, `id` no és utilitzada com directament per l'aplicació per identificar-la, perquè com que la vaca pertany a una granja la numeració de cada vaca comença des de zero en cada cas i és el camp `id_vaca` el que utilitzem realment. Aquesta taula també té una clau forania, `id_granja`, que és la que associa la vaca amb la granja de la que prové.

produccio

Aquesta taula és utilitzada principalment per generar els gràfics de producció de llet de cada vaca i inclou la producció diària de cada vaca. En un inici no va ser implementada fins que va ser imprescindible la seva utilització, ja que s'havia comés un error a l'augmentar la taula `vaques` horitzontalment afegint la informació diària dels paràmetres de cada vaca.

Consta d'una clau primària `id_produccio` que és usada únicament per raons de no repetició d'aquest camp, i com a claus foranies tenim `id_vaca` i `id_granja` per relacionar amb la vaca i la granja d'on és aquesta producció.

5.4 DESCRIPCIÓ DE LES CLASSES DEL SERVLET

Les diferents classes que presentem a continuació són les que s'han requerit per la implementació del programari. Totes elles formen part del servlet base i són les que

realitzen la funció de simular les granges i vaques, així com els usuaris que tenim actius en ella.

5.4.1 La classe servlet_i

És la classe que implementa el servlet, consta dels mètodes generals que hem vist en el punt anterior, a més de tots els mètodes principals del simulador i és l'encarregat de crear tots els objectes requerits per ell.

També hi podem trobar totes les possibles webs que genera l'aplicació i les funcions necessàries per el xifrat en MD5 de les paraules de pas dels usuaris i la creació d'un `PrintWriter` que és el que permetrà l'escriptura del codi Web des de l'aplicació.

5.4.2 La classe user

És una classe simple en la qual s'hi emmagatzema la informació dels usuaris actius en l'aplicació. El seu constructor és l'encarregat de introduir els valors de les variables en la seva creació, excepte el camp `id_sessio` que es pot modificar externament ja que pot canviar durant l'execució.

5.4.3 La classe granja

Mitjançant aquesta classe simulem tots els paràmetres que utilitzarà l'aplicació sobre la granja. En ella s'inclou la llista de les vaques que pertanyen en aquella granja, les variables pròpies de la granja com el seu identificador o el número de vaques, i tots els mètodes necessaris per interaccionar amb les vaques, com els d'afegir i retornar.

També inclou els mètodes que utilitzarem per la creació de la granja on les diferents vaques que la configuren s'han de crear i avançar en el temps de forma automàtica, de tal forma que l'alumne es pugui trobar vaques en tots els possibles estats del seu cicle i tingui ja una idea de la producció de llet que ha generat fins al moment.

Finalment la classe granja té els mètodes necessaris per la inserció i actualització de camps dins la base de dades i per carregar granges existents quan es realitza una nova connexió.

5.4.4 La classe vaca

Classe que permet a l'aplicació simular la lactància d'una vaca i la seva evolució pels diferents estats que configuren el seu cicle. El seu constructor es limita a generar-ne una i inicialitza els seus valors aleatòriament entre uns límits preestablerts.

Inclou els mateixos mètodes per gestionar la lactància i els estats que la classe anterior, però aquests estan configurats per la interacció amb l'alumne, les accions realitzades no són automàtiques, per tant, la seva correcta evolució dependrà de si aquest ho fa bé o no.

Disposa del mètode `carrega_vaca()` que és l'utilitza't quan es carrega una granja existent per actualitzar els valors que tenen les vaques de la granja en qüestió.

5.5 CREACIÓ DEL SERVLET DE GRÀFIQUES

Per tal de implementar les gràfiques que es mostren dins les webs de l'aplicació, s'ha creat un servlet secundari que s'encarrega de l'exportació de dades de la base de dades, la creació dels objectes necessaris per la seva representació i la impressió de l'objecte i conversió a format imatge per a ser visualitzat en aquestes.

L'estructura general d'aquest servlet és molt similar a l'anterior però més simplificada, ja que les seves funcionalitats són més reduïdes. Els objectes usats per la creació de les gràfiques són els que s'inclouen dins les llibreries JCommon i JFreeChat de Java i després d'importar les dades necessàries de la base de dades s'han introduït dins d'ells i posteriorment s'han convertit a format JPG mitjançant un mètode intern de la llibreria.

Consta únicament de la classe `grafiques` que és l'encarregada de la gestió del servlet i de la generació i conversió de les gràfiques.

També s'hi ha afegit una funcionalitat de zoom per tal de millorar la visualització de les gràfiques, aquesta s'ha realitzat amb el pas dels paràmetres mínim i màxim sobre els que es vol avaluar la gràfica per la URL, i la posterior modificació dels registres importats per ser representats.

La resposta d'aquest servlet a diferència de l'anterior, no és una Web de text normal sinó que és una imatge Web directament, i d'aquesta manera es carrega dinàmicament quan és cridat des del servlet base.

5.6 CREAMIÓ DE LA INTERFÍCIE WEB

Aquesta consta de diverses pàgines HTML que permeten als usuaris la interfície bàsica per interactuar amb l'aplicació, a continuació s'analitzaran cadascuna d'aquestes pàgines i veurem les possibles accions que s'hi poden realitzar. Cal destacar que totes les pàgines excepte la d'autenticació disposen d'un menú d'accés ràpid a la part superior esquerra que facilita la transició entre elles i és la forma de retrocedir sense utilitzar els botons del navegador.

Per a cadascun dels formularis de l'aplicació s'ha realitzat una validació dels camps mitjançant expressions regulars amb Javascript, i algunes de les taules com les del llistat de granges o vaques s'ha afegit la funcionalitat de reordenació dels seus elements amb la llibreria JQuery i en concret les funcions de Tablesorter.

Tota la distribució dels elements de les pàgines així com el seu aspecte s'han dissenyat mitjançant fulls d'estil CCS, també s'han adequat les pàgines per a ser visualitzades correctament amb totes les resolucions de pantalla a partir de 800x600 píxels i s'han centrat horitzontalment a ella.

5.6.1 Pàgina d'inici

Pàgina que permet als usuaris que s'autentifiquin i accedeixin a l'aplicació. Consta dels camps NIU, paraula de pas i curs que permeten a l'aplicació conèixer a l'usuari i a partir d'aquí dona una resposta depenent dels permisos que tingui.

També dona l'opció de registre per a nous usuaris.



Figura 18 - Pàgina d'inici

La seva implementació ha suposat alguns problemes afegits, com per exemple la codificació el codi introduït a MD5 per tal de fer la correcta comprovació amb el que hi ha a la base de dades. És l'encarregada de crear els nous objectes necessaris per identificar l'usuari que ha accedit a l'aplicació i la seva granja associada en cas que sigui un alumne. Per altre banda actualitza a la base de dades tots els índex que ocupen dins les llistes d'usuari i de granja en cas que en tingui juntament amb l'identificador de la sessió actual.

També és la pàgina que realitza la desconnexió d'un usuari de l'aplicació així com la destrucció de tots els objectes que s'hagin creat per a ell.

5.6.2 Pàgina de registre

Des d'aquesta pàgina es permet el registre d'un nou usuari a l'aplicació. Els camps requerits en el registre són el NIU, nom i cognoms del usuari, correu electrònic i curs on pertany.

Registrar-se al simulador

[Endarrera](#)

Introduir les dades d'usuari

NIU
1234567

Paraula de pas
••••

Nom
Joan

Cognom 1
Garcia

Cognom 2
Perez

Email
joan@uab.es

Curs
Curs Primer

Figura 19 - Formulari de registre

Aquesta pàgina realitza la inserció del usuari dins la taula d'usuaris de la base de dades del programari, i mitjançant Javascript es comprova un per un tots els registres per que compleixin el format que ha de tenir cadascun. Si hi ha un usuari registrat amb aquell NIU es dona un missatge d'error dient que no és possible el registre perquè aquell usuari ja existeix i un enllaç amb la pàgina d'accés. En cas contrari es confirma el registre i l'usuari podrà accedir a l'aplicació quan el professor li assigni una granja.

5.6.3 Vista de la granja

Amb aquesta pàgina es genera una vista general de la granja, a la part central hi trobem el llistat de les vaques que li pertanyen i una informació bàsica sobre elles, i a la part superior dreta hi ha la informació referent a la producció diària i total de la granja i el dia actual des de la seva creació.

Si l'usuari que accedeix a la pàgina té permisos d'administrador o professor a part de veure tot el que s'ha esmentat anteriorment, també té un petit menú que li permet assignar alumnes a la granja i la possibilitat de veure el nombre d'inseminacions i secats correctes i totals que han realitzat els alumnes. També té la opció d'eliminar una vaca i crear-ne una de nova al seu lloc, per si modifica algun dels paràmetres que sols afectin a la creació de noves vaques.

En aquest cas, el professor o administrador té accés des de el menú ràpid a la pàgina de configuració i d'administració del curs.

Granja 1

Granges
Configuració
Desconnectar

VACA	Produccio diaria	Produccio acumulada	Estat	Nº part	+
00	81	52922	Ciclant	5	
01	0	21566	Seca	2	
02	30	8326	Inseminada	1	
03	33	10567	Inseminada	1	
04	34	59287	Inseminada	5	
05	47	40196	Inseminada	5	
06	12	35259	Inseminada	4	
07	31	28798	Inseminada	3	
08	56	21109	Inseminada	3	
09	37	1051	Ciclant	1	

Alumnes
1111111
Nou usuari

Estat Granja	
Dia actual	6
Produccio diaria granja	366 litres
Produccio total granja	14145 litres
Fecundacions	0/0
Secats	0/1

Dia Seguent

Figura 20 - Pàgina de la granja

El funcionament intern d'aquesta pàgina és força complex.

Primerament es realitza una cerca a la base de dades sobre la granja en concret per tal de mostrar en forma de llista totes les vaques que hi formen part.

Les funcionalitats de l'alumne consten del botó per avançar el dia, i un enllaç al final de cada vaca amb la vista de vaca d'aquesta. Si avancem un dia, al carregar-se de nou la pàgina crida a un mètode que recorre cada vaca i calcula la lactació i tots els paràmetres que configuren el seu cicle per a un nou dia. També s'encarrega d'actualitzar aquests paràmetres a la base de dades i inserir a la taula de producció les produccions diàries de cada vaca.

Si ho veiem des del punt de vista del professor o administrador, a més de les funcionalitats anteriors tenim dues opcions afegides, la substitució d'una vaca i l'addició d'usuaris a la granja.

La primera inicialitza l'objecte de la classe vaca ja creada, amb tots els paràmetres a zero o els aleatoris corresponents, i esborra tota la informació referent a ella de la base de dades. Un cop esborrada es generen aleatòriament el nombre de dies a transcórrer i finalment cridem als mètodes de la classe vaca que s'encarreguen de la lactació i d'actualitzar la base de dades amb els nous paràmetres.

L'altre funcionalitat és afegir usuaris a la granja, i aquesta realitza una cerca dels alumnes que no tenen granja del curs en qüestió i els mostra en forma de llista desplegable. El professor en selecciona un i posteriorment apareix el botó d'afegir, que

el que fa és modificar el registre d'identificador de granja de l'alumne a la base de dades.

5.6.4 Vista de la vaca

Mitjançant aquesta pàgina podem visualitzar l'estat actual en que es troba cada vaca, així com una gràfica, en la seva part central, amb l'històric de producció des de la seva primer part. Aquesta gràfica té una funció de zoom que permet veure amb més detall la zona que interressi analitzar.

També s'ha incorporat una barra d'accés ràpid a les diferents vaques que configuren la granja amb la possibilitat d'accedir a la següent o a l'anterior sense necessitat de retornar a la pàgina de la granja.

Des d'aquesta també s'hi poden realitzar les accions de "Inseminar", "Seca" i "Actualitzar Estat" que són les que hauran de realitzar correctament els alumnes per demostrar els seus coneixements.

A la dreta de la gràfica tenim una taula que ens resumeix la informació bàsica sobre aquella vaca i sobre les accions que s'hi ha realitzat en aquell moment.

Quan l'usuari que accedeix a la pàgina és un administrador o professor podem observar informació addicional a la taula de la dreta sobre els dies de fertilitat, de cicle, de gestació i secat així com uns contadors dels dies que porten en els dos últims estats, per tal que el professor sàpiga amb exactitud els dies correctes en que l'alumne ha de realitzar les accions, ja que aquests es generen de forma aleatòria i resulta força complicat endevinar-los.

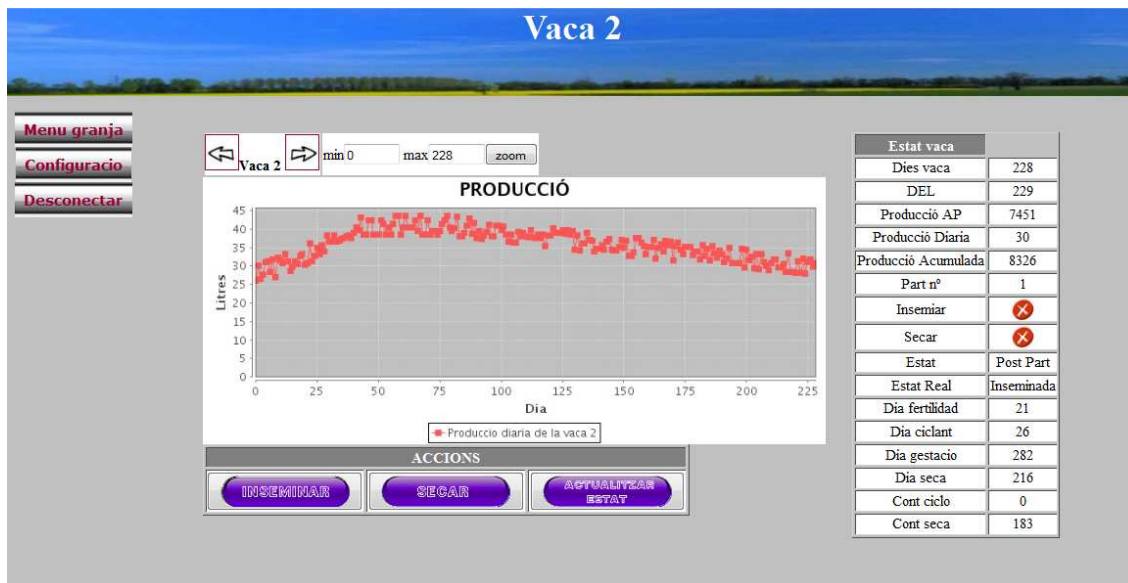


Figura 21 - Pàgina de vaca

Aquesta pàgina té les mateixes funcionalitats independentment dels privilegis de l'usuari.

Primerament permet avançar o retrocedir entre les vaques de la granja mitjançant un enllaç a la mateixa pagina però amb el pas de l'índex d'una vaca anterior o següent. Realitzem una comprovació perquè no excedeixi del rang de vaques.

La segona de les funcionalitats que ofereix és la del rang de zoom de la gràfica, que es realitza mitjançant el pas dels valors màxim i mínim quan realitzem la crida al servlet de gràfiques, i aquest s'encarrega de llegir aquesta informació i processar-la adequadament.

Per últim tenim les accions sobre la vaca, les dues primeres s'implementen per mitjà de variables booleanes que a l'avançar un nou dia són llegides pel simulador i es realitza l'acció pertinent, i la d'actualitzar estat és un simple paràmetre que modifica l'estat analitzant la vaca.

5.6.5 Menús de configuració i vistes professor

Aquest conjunt de pàgines són les que permeten a l'administrador o professor configurar l'aplicació i veure l'estat de totes les granges del curs que administren.

La vista principal del menú de configuració és el llistat dels cursos que pot administrar el professor juntament amb un enllaç que li dona accés a la pàgina de creació de nous cursos. A la part dreta també hi podem trobar un enllaç amb la pàgina per crear i

carregar còpies de seguretat, i si l'usuari que s'hi ha connectat és l'administrador té la possibilitat, mitjançant un altre enllaç d'accedir a la pàgina per modificar tots els paràmetres per defecte del simulador.



Figura 22 - Pàgina de configuració

Aquesta pàgina té un funcionament intern molt simple, ja que es limita a donar accés a altres pàgines per mitjà d'enllaços i només realitza una cerca sobre la taula `usercursos` enllaçant-los amb un JOIN amb la taula `cursos` que pot gestionar el professor o administrador i els mostra en forma de llista.

La pàgina de creació d'un nou curs fa una inserció dins la base de dades amb la informació introduïda al formulari. Aquesta informació és vàlida per Javascript i en cas que sigui errònia es mostra un missatge amb el format correcte.



Figura 23 - Formulari de nou curs

La pàgina d'administració de backup està formada per un llistat format per totes les còpies de seguretat que s'han realitzat al simulador. Aquestes es llegeixen de la base de dades i serveixen com a nom del arxiu on es troba la còpia.

Dóna la possibilitat de crear noves còpies, prement el botó nou, i l'aplicació genera un nom a partir de la data actual i la versió per diferenciar les que s'hagin creat el mateix dia. Un cop generat el nom es crea l'arxiu en qüestió i s'executa la comanda `mysqldump` i si desa el resultat.

Posteriorment si volem carregar una còpia antiga, el procés que realitza és a la inversa, executa la comanda `mysql` i li passem per paràmetre l'arxiu que guarda la còpia.



Figura 24 - Pàgina de gestió de còpies de seguretat

La pàgina d'administració del simulador consta del formulari que es mostra a configuració, i els valors són emmagatzemats a la base de dades. Aquests valors són els que apareixen per defecte en la creació de granges i només són modificables per l'administrador.

The screenshot shows a web interface titled "Configuració per defecte del simulador". On the left, there are two buttons: "Configuracio" and "Desconectar". The main area contains a form titled "Paràmetres per defecte del simulador" with the following fields:

- Numero de vaques: 80
- Produccio anual potencial: entre 7000 i 11000
- Valor de proteina potencial: entre 2.8 i 3.4
- Valor de grasa potencial: entre 3.5 i 4.2
- Valor potencial de celules somàtiques: entre 80 i 650
- Provabilitat de fecundació: 50
- Dies de cicle de la vaca: entre 25 i 40
- Dia de fertilitat: entre 18 i 24
- Dies de gestació: entre 275 i 285
- Dies de secat: entre 215 i 225

At the bottom of the form are "Envia" and "Reset" buttons.

Figura 25 - Formulari de modificació dels paràmetres per defecte del simulador

A continuació s'explicarà amb més detall totes les pàgines de configuració que deriven d'aquesta, i totes les possibles accions que s'hi poden realitzar.

Quan es selecciona un dels cursos del llistat de la pàgina de configuració inicial, s'accedeix a la pàgina que permet administrar aquell curs.

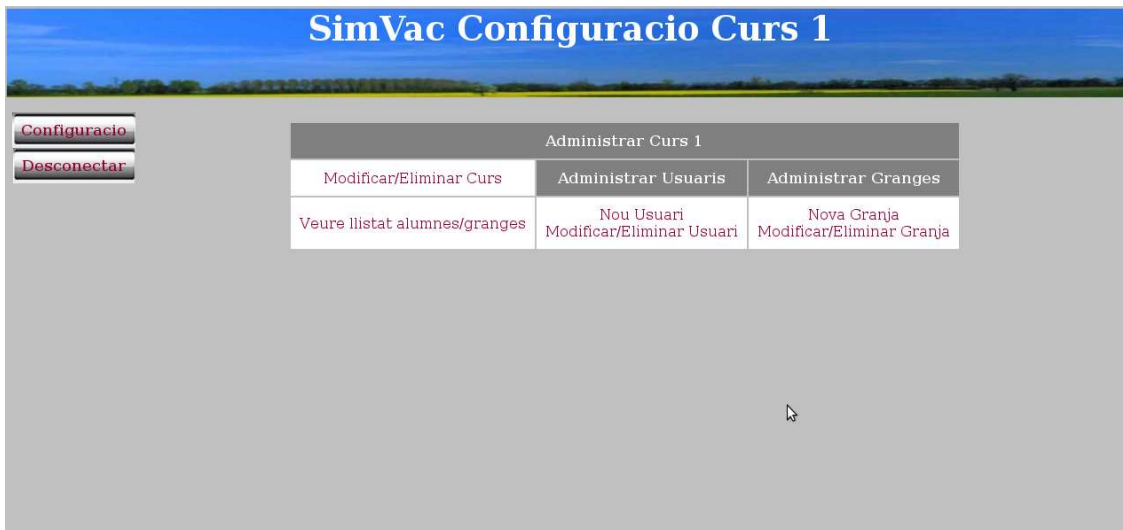


Figura 26 - Pàgina d'administració d'un curs

Com en el cas anterior la pàgina d'administració d'un curs consta de varis enllaços a les funcions que vulguem realitzar.

Aquestes funcions són:

- Modificar o eliminar el curs

Pàgina que permet l'edició o eliminació del curs que s'està administrant, té una pàgina intermèdia que permet escollir l'opció, i després redirigeix al formulari d'edició o a la confirmació d'eliminació.

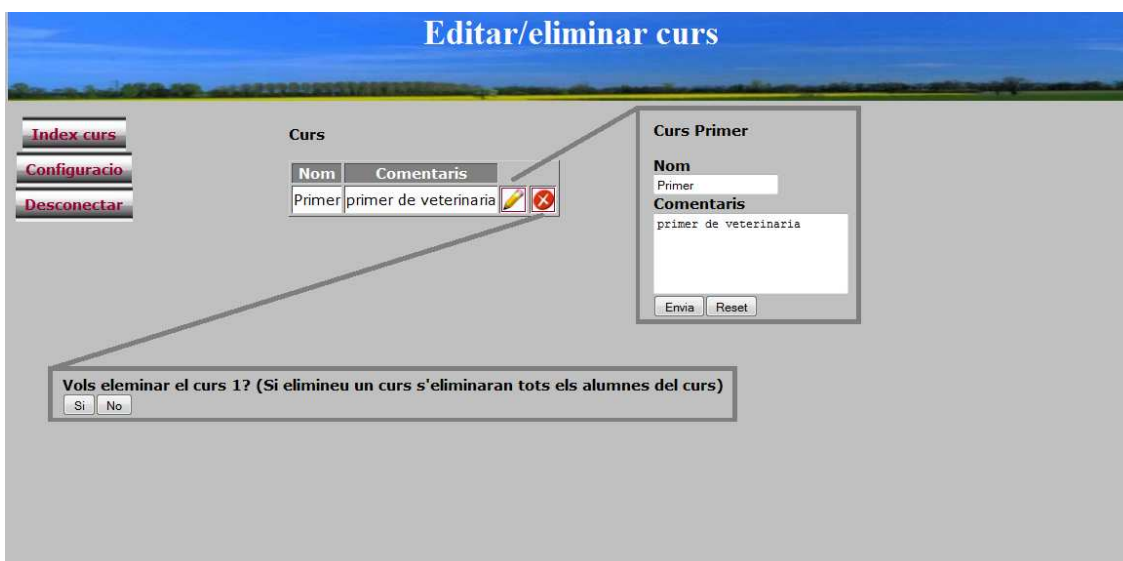


Figura 27 - Modificació o eliminació de cursos

En primer lloc es cerca a la base de dades la informació sobre el curs actual, mostrant-la en forma de llista i permet mitjançant un enllaç la edició o eliminació del curs. En el cas d'editar mostra un formulari on hi carreguem la informació existent a la base de dades perquè es pugui modificar i posteriorment és actualitzada si aquesta s'envia. Els paràmetres es validen abans de ser enviats i en cas que no siguin correctes mostrem el format que han de tenir. En cas que es vulgui eliminar el curs i la confirmació sigui positiva esborrem de la base de dades el curs en qüestió i a conseqüència tots els alumnes que formen part del curs.

- Nous usuaris al curs

Formulari per la inserció de nous usuaris al curs administrat, també se'n poden afegir mitjançant el registre de l'aplicació però des d'aquí es dona l'opció que aquests tinguin els permisos de professor.



Figura 28 - Formulari d'un nou usuari

Aquesta pàgina afegeix a la base de dades l'usuari que es vol registrar. Com la pàgina de registre fa totes les comprovacions necessàries per que les dades introduïdes estiguin en el format correcte i en cas contrari s'informa d'aquest al usuari. El que la diferencia és que permet la creació de nous professors, assignant-los solament un curs.

- Modificar o eliminar usuaris existents del curs

Amb aquesta pàgina visualitzem el llistat d'alumnes que estan registrats al curs en qüestió i les accions d'edició o eliminació que podem realitzar sobre cadascun d'ells.

Un cop seleccionada l'acció s'accedeix al formulari d'edició o a la confirmació d'eliminació respectivament. Si es tenen permisos d'administrador també és possible la gestió de professors així com la seva inscripció en varis cursos.

Figura 29 - Modificació o eliminació d'usuaris

El funcionament intern d'aquesta pàgina és diferent per professors o administrador.

En el cas del professor fa una cerca sobre una taula creada a partir de la unió mitjançant un JOIN de les taules usercursos, cursos i users, i els mostra en forma de llista. Al final de cada fila s'hi ha afegit els enllaços per modificar i eliminar l'usuari en qüestió. En el cas d'edició es carrega sobre un formulari tota la informació sobre l'alumne i es dona la possibilitat de canviar-li els permisos a professor per si el registre s'ha realitzar des de la pàgina de registre. Tots els camps es validen i posteriorment s'actualitzen a la base de dades. En el cas d'eliminació i després de la pertinent confirmació s'elimina l'usuari de la base de dades.

Si l'usuari és administrador pot realitzar les mateixes modificacions sobre els professors que gestionen el curs. Cal puntualitzar que en el cas del professor i degut a que el nombre de cursos és una variable dinàmica, ha ocasionat que els paràmetres d'HTML referents al curs tinguin noms diferents, i per tant quan els llegim s'ha de realitzar un recompte dels existents i crear les variables en funció d'aquesta cerca. Per tal de realitzar la correcta validació d'aquests camps s'han utilitzat variables ocultes que ens

donen informació sobre el número de cursos i els permisos del usuari, per saber els nombre i els noms dels camps a validar.

- Noves granges

Des d'aquesta pàgina el professor pot crear noves granges i decidir els paràmetres que tindrà cadascuna d'elles, es dóna la possibilitat d'escollir el número de granges a crear, i paràmetres com el número de vaques que tindran i els generals que configuren les vaques que s'hi creïn.

Figura 30 - Formulari d'una nova granja

El funcionament intern d'aquesta pàgina és un dels més complexos. En primer lloc inicialitza tots els valors del formulari amb els paràmetres per defecte del simulador que són llegits de la base de dades.

Després de les modificacions que es realitzin fem una comprovació de tots els paràmetres, tan el seu format com els rangs en que són correctes.

Quan es valida aquesta informació l'aplicació crea un nou objecte de la classe vaca, que l'avancem un nombre de dies aleatori mitjançant els mètodes interns d'aquesta classe i es van desant a la base de dades tots els paràmetres tan de producció com l'estat final en que es troba aquesta.

- Modificar o eliminar granges

Pàgina que permet editar o modificar granges existents, aquestes es mostren en forma de llista a la part centrar d'aquesta i s'hi mostren també alguns dels seus paràmetres generals.

Com en els casos anteriors tenim dos enllaços a cada granja que ens redirigeixen al formulari d'edició o a la confirmació d'eliminació d'aquella granja.

Editar/eliminar granges

Granges

Numero	Numero vaques	Dia	PAP	VPP	VGP	VPCS	PF	DC	DF	DG	DS
1	10	6	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
2	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
3	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
4	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
5	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
6	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
7	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
8	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
9	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225
10	10	0	7000-11000	2.8-3.4	3.5-4.2	80-650	50	25-40	18-24	275-285	215-225

PAP - Produccio anual potencial
VPP - Valor de proteina potencial
VGP - Valor de grasa potencial
VPCS - Valor potencial de cel.lules somàtiques
PF - Provabilitat de fecundació
DC - Dies de cicle
DF - Dia de fecundació
DG - Dies de gestació
DS - Dies de secat

Vols eleminar la granja 1?

Granja numero 1

Produccio anual potencial
entre 7000 i 11000

Valor de proteina potencial
entre 2.8 i 3.4

Valor de grasa potencial
entre 3.5 i 4.2

Valor potencial de celules somàtiques
entre 80 i 650

Provabilitat de fecundació
50

Dies de cide de la vaca
entre 25 i 40

Dia de fertilitat
entre 18 i 24

Dies de gestació
entre 275 i 285

Dies de secat
entre 215 i 225

Figura 31 - Modificació o eliminació de granges

Té un funcionament molt semblant a modificar cursos o usuaris, tan en l'edició com en l'eliminació, però a diferència d'aquests no només es mostren les granges que pertanyen als usuaris del curs, sinó que també es llisten les que encara no han estat assignades a cap usuari.

- Veure llistat d'alumnes / granges

Aquesta pàgina és la utilitzada per permetre al professor o administrador tenir una vista general de les granges i els alumnes a qui han estat assignades, i alhora permet tenir un enllaç a la pàgina de vista de granja de cadascuna d'elles.

La informació mostrada també serveix al professor per realitzar l'avaluació sobre les accions que hagi pres l'alumne, ja que resum el total de dies que han transcorregut i les realitzades amb èxit sobre les accions totals.

Granja	Dies	Producció total	Fecundacions	Secats	Alumnes
1	6	14145	0/0	0/1	1111111
2	0	0	0/0	0/0	-
3	0	0	0/0	0/0	-
4	0	0	0/0	0/0	-
5	0	0	0/0	0/0	-
6	0	0	0/0	0/0	-
7	0	0	0/0	0/0	-
8	0	0	0/0	0/0	-
9	0	0	0/0	0/0	-
10	0	0	0/0	0/0	-

Figura 32 - Llistat de granges d'un curs

El funcionament d'aquesta pàgina con en casos anteriors és d'una complexitat baixa. Primerament realitzem una cerca a la base de dades de les granges que pertanyen al curs seleccionat, enllaçant-la amb un JOIN amb la d'usuari. Després mostrem la informació en forma de llista, i tractem de forma diferent si hi ha alguna granja amb més d'un usuari per tal que la visualització d'aquests sigui de forma correcte. Al final de cada fila hi inserim l'enllaç a la granja en qüestió per que el professor hi pugui tenir accés.

Capítol 6 - PROVES DEL PROJECTE

En aquest capítol numerarem totes les proves i els problemes que hem trobat al aplicar-les. Les dividirem entre proves en la infraestructura, on s'ha comprovat el correcte funcionament del servidor, les proves sobre el simulador que verificaren el correcte funcionament d'aquest, i per últim les proves sobre les pàgines Web, que serviran per comprovar que s'hagin previst tots els possibles errors.

6.1 PROVES EN LA INFRAESTRUCTURA

Les proves que s'han realitzat sobre la infraestructura han estat principalment sobre el funcionament d'aquesta, mitjançant en primer lloc una pàgina Web senzilla creada amb PHP que mostrava un formulari per pantalla i inseria les dades a una base de dades de prova.

S'han modificat alguns paràmetres de configuració d'Apache, Mysql, Tomcat i Eclipse tals com el número de port de Tomcat o la ubicació del driver de connexió amb la base de dades pel seu correcte funcionament. També s'han realitzat algunes proves de seguretat com l'intent d'accedir remotament a pàgines de configuració i s'han activat les mesures corresponents per evitar intrusions.

Posteriorment van sorgir problemes en la implementació del servlet base, ja que no es tenia cap coneixement sobre ells i per tant, moltes de les seves parts van ser testades mitjançant el mètode prova i error, tot i que gràcies a uns bons tutorials es va simplificar força la seva correcte implementació.

6.2 PROVES EN EL SIMULADOR

Sobre el simulador s'han anat realitzant proves contínuament, mitjançant la consola de Eclipse s'han comprovat que els valors de les variables siguin correctes en tot moment i que el flux d'execució transcorri correctament en tots els casos. Un cop es tenia una versió estable moltes d'aquestes comprovacions també es realitzaven a partir dels valors emmagatzemats dins la base de dades, ja que resultava més senzill i entenedor, i alhora també es feien comprovacions modificant els paràmetres per defecte del simulador per veure que aquests modifiquin realment el comportament d'aquest.

La segona fase de proves ja va ser la comprovació del correcte funcionament del servlet de gràfiques, primer generant gràfiques externament al simulador i després un cop integrat a ell. Es van testar sobretot el correcte accés a la base de dades ja que en un principi generava errors de concurrència, la funció de zoom perquè sorgien errors en l'escala horitzontal, i per últim de la correcta conversió format imatge ja que en un principi aquests es generaven sobre una aplicació Java.

6.3 PROVES EN EL FUNCIONAMENT DE LES PÀGINES

Sobre el funcionament de les pàgines Web s'ha testat en primer lloc tota la sintaxi HTML, ja que al ser generada mitjançant codi Java resultava més complicat visualitzar els errors. També s'ha efectuat una comprovació mitjançant diferents tipus de pantalla que la resolució d'aquestes no impedeixi la correcta visualització de les Webs

Un altre punt important d'aquestes proves ha estat que tots els formularis de l'aplicació tinguin una correcta validació de camps i no permetin introduir dades incorrectes.

Per concloure les proves s'ha fet una comprovació a totes les funcionalitats que havia de implementar:

- **Login** : Funciona correctament, van sorgir problemes per mantenir una persona identificada i es van arreglar mitjançant les variables de sessió.
- **Desconnexió**: Van sorgir molts problemes amb la desconnexió, ja que no s'eliminaven els objectes creats i en cas d'una reconexió sorgien problemes de funcionament, es va solucionar realitzant una comprovació a la pàgina d'inici que esborrava tota la informació sobrant.
- **Registre**: Funciona correctament.
- **Modificar paràmetres per defecte**: Funciona correctament.
- **Crear curs**: Funciona correctament.
- **Modificar curs**: Funciona correctament.
- **Eliminar curs**: En aquest cas es cometia l'error de no eliminar els usuaris d'aquest curs i per tant els usuaris continuaven tenint accés al simulador, es va corregir i es va realitzar una nova comprovació en el login per tal d'evitar l'error.
- **Crear granja**: Funciona correctament, es van haver de revisar les funcions i generar la carga dels valors per defecte del simulador.
- **Modificar granja**: Funciona correctament, es va afegir un text explicatiu del efecte de les variables que es modifiquen.

- **Eliminar granja:** Es va cometre el mateix error que en el cas dels cursos, ja que no s'eliminaven les vaques pertanyents a la granja i es va solucionar correctament.
- **Crear usuari:** Funciona correctament.
- **Modificar usuari:** Funciona correctament, s'ha ampliat la funcionalitat permetent la modificació dels cursos que gestiona un professor.
- **Eliminar usuari:** Funciona correctament.
- **Crear còpies de seguretat:** Per la creació de còpies de seguretat es van trobar sobretot problemes per la generació de l'arxiu extern, ja que el programa no donava cap error però l'arxiu no era creat correctament, així que es va optar per una creació manual del arxiu i de l' inserció del codi generat per la comanda "mysqldump".
- **Carregar còpies de seguretat:** Per carregar-les no van sorgir els mateixos problemes que alhora de ser creades, primer es va cridar al procés MySQL i posteriorment se li van anar passant els paràmetres un a un des d'un bucle intern que llegia l'arxiu a carregar.
També van sorgir problemes alhora de carregar una còpia de seguretat i continuar amb el funcionament normal de l'aplicació així que es va optar per haver de realitzar una desconexió un cop s'executés la carrega d'aquesta.
- **Visualitzar granges-alumne:** Funciona correctament , es van arreglar alguns problemes alhora de la visualització de varis alumnes en una sola granja..
- **Assignar alumne a granja:** Funciona correctament.
- **Avançar un dia:** Funciona correctament, es va eliminar de la pàgina de la vaca per no confondre al usuari ja que el dia s'avança per a totes les vaques de cop.
- **Inseminar i secar:** Aquestes accions donaven molts problemes de funcionament si es volien processar dins del mateix generador de la Web, així que es va decidir realitzar-les abans de la seva creació per tal de obtenir els valors de forma correcte.
- **Actualitzar estat:** Per l'actualització del estat de la vaca al seu estat real, van sorgir problemes ja que no canviava correctament del estat ciclant al inseminada, i la solució va ser la creació d'un comptador que permetia guardar la validesa de l'acció.
- **Zoom de gràfica:** Funciona correctament, es van corregir alguns problemes sobre l'escala de temps horitzontal.

Capítol 7 - CONCLUSIONS

Per concloure aquesta memòria analitzarem el compliment dels objectius del projecte, les desviacions sobre la planificació ajustada i una opinió personal sobre el projecte en general.

En quan al compliment dels requeriments, podem dir que s'han realitzat satisfactòriament tots els requeriments funcionals sobre l'aplicació.

Permet la simulació de la producció de llet d'una vaca així com una interfície Web que permet la interacció amb el simulador. Això implica el compliment també de la realització de la infraestructura base per suportar aquest simulador i el servidor Web. També implementa un control d'accés sobre els usuaris que s'hi connecten, i permet la realització de totes les accions de manteniment sobre les bases de dades que són requerides. L'últim requisit funcional a complir és la realització de còpies de seguretat així com la seva recuperació.

Per altra banda si s'analitza el compliment dels requeriments no funcionals es pot afirmar que és en gran part com s'explica a continuació. L'aplicació permet la connexió multiusuari, té tolerància a errades i accions incorrectes i mitjançant la consola d'Eclipse es té un control sobre tots els accessos que es realitzen a l'aplicació. Aquest control es realitza d'acord amb l'ACL, utilitzant una llista d'usuaris i assignant els permisos pertinents a cadascun d'ells. L'aplicació genera, en cas que no existeixin, les bases de dades necessàries pel seu funcionament i fa les insercions de l'usuari administrador i de les dades per defecte del simulador. Aquestes últimes són modificables per l'administrador a partir de l'interfície Web un cop ja està en funcionament. També s'ha completat correctament la creació del servlet secundari que genera les gràfiques de la producció de llet així com la seva integració dins la interfície Web final. Per últim l'aplicació genera diferents vistes depenent dels permisos que tingui cada usuari, i permet a l'alumne realitzar les accions que requeria pel seu correcte funcionament.

L'únic requisit que no s'ha complert és la de l'accés físic del servidor, ja que tot el projecte s'ha realitzat en un ordinador propi i no s'ha disposat finalment del servidor necessari a la facultat de veterinària.

Si analitzem les desviacions finals sobre la planificació realitzada en el capítol de disseny, observem que tant sols hi ha hagut petites modificacions en les hores dedicades a la realització de proves i l'elaboració de la memòria. S'ha utilitzat menys hores de les estimades per realitzar les proves pertinents, ja que no s'han efectuat proves sobre la infraestructura de la facultat, i se n'ha utilitzat alguna més en l'elaboració d'aquest document. Per tant l'efecte final d'aquests canvis ha estat equilibrat, i es mantenen iguals tots els punts importants de la planificació tals com la data de final i els costos totals del projecte.

També cal remarcar les possibles ampliacions que es poden fer a l'aplicació, com el mòdul de malalties o d'alimentació, que en un principi van ser desestimats per falta de temps. De totes formes el disseny del simulador s'ha adequat a aquestes possibles ampliacions, per que resulti més senzill en un futur realitzar-les.

Per finalitzar el document faré una valoració a personal sobre el projecte realitzat i els beneficis que m'ha aportat. En primer lloc, vull destacar que un dels motius per la realització del projecte ha estat que durant gran part de la titulació m'he centrat a cursar més assignatures de sistemes, i no tantes de gestió, per tant, tenia una mancança de coneixements en quan a la programació d'aplicacions i en concret en l'entorn Web. D'aquesta manera he aconseguit ampliar el meu ventall de possibilitats laborals, conèixer altres llenguatges de programació que no havia estudiat, i més experiència en els que ja coneixia com alhora d'integrar el sistema en Apache i Tomcat.

Així doncs, m'he pres aquest projecte com una assignatura més i no només m'ha permès saber com hauré d'afrontar un projecte en un futur sinó que també m'ha ensenyat a realitzar autoaprenentatge sobre qüestions desconegudes en el seu inici. Gràcies al fet de tenir límits en les entregues he après a realitzar la feina de forma continuada i gestionar bé el meu temps. Tot i que la seva dedicació ha estat inferior a la desitjada, intentava destinar alguna hora del dia a fer un repàs general del codi i d'aquesta manera tenir fresca la informació i no perdre el fil.

Concloent la meva valoració, només dir que crec que ha estat una feina profitosa que ha aportat moltíssim a la meva formació i que m'ha ensenyat moltes coses útils que utilitzaré en el futur més pròxim.

Finalment donar les gràcies al meu tutor en primer lloc per la paciència que ha tingut amb mi, i per tot el que he après amb ell. En segon lloc a la meva família i amics pel

suport moral que m'han donat en el dia a dia i per últim a la meva parella per tota la ajuda i paciència que ha tingut durant la realització del projecte.

BIBLIOGRAFIA

Definicions i informació general

<http://www.wikipedia.org>

<http://www.alegsa.com.ar/Dic/sghd.php>

<http://www.monografias.com/trabajos4/basesdatos/basesdatos.shtml>

<http://www.webtaller.com/manual-java/caracteristicas-java.php>

<http://www.silicontower.net/alojamiento-web/caracteristicas-detalladas/lenguajes-programacion>

<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>

Instal·lació i configuració de Xampp

<http://www.video-tutorial-joomla.com/servidor/xampp-instalar-servidor-ubuntu.html>

<http://www.forat.info/2008/01/16/como-configurar-phpmyadmin-y-mysql-de-xampp-en-linux-ubuntu/>

<http://www.forat.info/2008/01/13/como-montar-un-servidor-web-con-xampp-en-linux-ubuntu/>

Instal·lació i configuració de Tomcat, Eclipse .

<http://www.gurx.net/index.php?sec=documentacio&element=38>

http://usuarios.multimania.es/neosprogramari/articulos/servidores/tomcat_5.5.php

Descàrrega i informació sobre el connector JDBC

<http://www.ubuntu-es.org/?q=node/99378>

<http://www.ubuntu-es.org/index.php?q=node/42704>

<http://www.kitebird.com/articles/jdbc.html>

<http://dev.mysql.com/downloads/connector/j/>

Manuals sobre Java i Servlets de Java

<http://www.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/javaservlets/servlets.pdf>

<http://sanchez-soft.blogspot.com/2006/07/java-crear-un-servlet-en-eclipse.html>

<http://www.proactiva-calidad.com/java/servlets/introduccion.html>

http://blogs.sun.com/vittal/entry/using_servlets_in_ubuntu

Tutorials HTML, CSS, Javascript, entre altres.

<http://www.w3schools.com>

Tutorial i informació JFreeChart

http://www.programacion.com/articulo/graficas_en_java_con_jfreechart_238

<http://elespaciodejaime.wordpress.com/2009/06/27/jfreechart-facil-creacion-de-graficos-estadisticos-en-java/>

http://www.javamexico.org/blogs/shadonwk/utilizacion_de_graficos_con_jfreechart_al_fin_puede

Tutorial de JQuery + Tablesorter amb descarregues i exemples.

<http://tablesorter.com/docs/>

Manual de MySQL

<http://www.webestilo.com/mysql/>

<http://dev.mysql.com/doc/refman/5.0/es/index.html>

http://www.google.es/url?sa=t&source=web&ct=res&cd=3&url=http%3A%2F%2Fdownloads.mysql.com%2Fdocs%2Frefman-5.0-es.a4.pdf&ei=qqSOSqjCBs_1-Qa42ezyDQ&rct=j&q=manual+mysql&usg=AFQjCNEPfmSx7J64BMUfGnQIUq-0OsdYpw

Informació sobre CSS

<http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>

Informació sobre javascript

<http://www.desarrolloweb.com/articulos/25.php>

Validació de camps en javascript

<http://www.desarrolloweb.com/articulos/1767.php>

http://expo.itch.edu.mx/view.php?f=trad_22

http://www.codehouse.com/webmaster_tools/regex/

Signat: SERGI RODRIGUEZ LLAVET