



**Universitat Autònoma  
de Barcelona**

# **SIMULADOR DE PRODUCTIVITAT EN UNA EXPLOTACIÓ LLETERA PER DISPOSITIUS MÒBILS**

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Sistemes  
realitzat per  
David Ramos Morera  
i dirigit per  
Gonzalo Vera Rodríguez

**Escola d'Enginyeria**  
Sabadell, juny de 2012



## FULL DE RESUM

<b>Títol del projecte:</b> Simulador de productivitat en una explotació lletera per dispositius mòbils	
<b>Autor[a]:</b> David Ramos Morera	<b>Data:</b> Juny 2012
<b>Tutor[a]/s[es]:</b> Gonzalo Vera Rodríguez	
<b>Titulació:</b> Enginyeria Tècnica en Informàtica de Sistemes	
<b>Paraules clau</b> (mínim 3) <ul style="list-style-type: none"><li>• Català: Android, Simulador, Java.</li><li>• Castellà: Android, Simulador, Java.</li><li>• Anglès: Android, Simulator, Java.</li></ul>	
<b>Resum del projecte</b> (extensió màxima 100 paraules) <ul style="list-style-type: none"><li>• Català: En un projecte anterior es va desenvolupar una aplicació web amb Java on s'implementava un simulador de granges de llet. L'objectiu d'aquest projecte és desenvolupar una interfície d'usuari per a dispositius mòbils que permeti interactuar amb les mateixes granges virtuals.</li><li>• Castellà: En un proyecto anterior se desarrolló una aplicación web en Java donde se implementaba un simulador de granjas de leche. El objetivo de este proyecto es desarrollar una interfaz de usuario para dispositivos móviles que permita interactuar con las mismas granjas virtuales.</li><li>• Anglès: In a previous project there was developed a web application in Java where a milk farm simulator was implemented. The target of the actual project is to develop a user interface for mobile devices to allow the users to interact with the same virtual farms.</li></ul>	



# ÍNDEX

---

<b>INTRODUCCIÓ</b> .....	<b>1</b>
1.1 ANTECEDENTS I PROBLEMA ACTUAL.....	1
1.2 OBJECTIUS.....	2
1.3 ESTAT DE L'ART .....	3
1.4 ESTRUCTURA DE LA MEMÒRIA.....	4
<b>2 ESTUDI DE VIABILITAT</b> .....	<b>7</b>
2.1 INTRODUCCIÓ.....	7
2.2 ESTUDI DE LA SITUACIÓ ACTUAL .....	7
2.3 REQUERIMENTS DEL PROJECTE .....	8
2.4 PROPOSTA DE SOL.LUCIÓ I ALTERNATIVES.....	10
2.5 PLANIFICACIÓ DEL PROJECTE.....	11
2.6 AVALUACIÓ DE RISCOS .....	14
2.7 AVALUACIÓ DE COSTOS I BENEFICIS.....	16
2.8 CONCLUSIONS .....	19
<b>3 ANÀLISI</b> .....	<b>21</b>
3.1 INTRODUCCIÓ.....	21
3.2 ANÀLISI DE L'APLICACIÓ ACTUAL.....	21
3.2.1 Anàlisi funcional.....	21
3.2.2 Anàlisi tècnic.....	22
3.3 ESPECIFICACIONS FUNCIONALS.....	25
3.4 ESPECIFICACIONS NO FUNCIONALS.....	30
3.5 MARC TECNOLÒGIC.....	31
3.5.1 Arquitectura de l'aplicació.....	32
3.5.2 Característiques de les aplicacions per a mòbil .....	33
<b>4 DISSENY</b> .....	<b>37</b>
4.1 INTRODUCCIÓ.....	37
4.2 DEFINICIÓ DE L'ARQUITECTURA .....	37
4.3 CAPA DE SERVIDOR .....	41
4.1 CAPA D'ACTIVITATS .....	42

4.2	<b>CAPA DE SERVEIS</b> .....	<b>47</b>
4.3	<b>CAPA DE DADES</b> .....	<b>50</b>
4.4	<b>PLANIFICACIÓ ACTUALITZADA</b> .....	<b>51</b>
<b>5</b>	<b>IMPLEMENTACIÓ</b> .....	<b>55</b>
5.1	<b>INTRODUCCIÓ</b> .....	<b>55</b>
5.2	<b>SERVEIS WEB</b> .....	<b>55</b>
5.3	<b>APLICACIÓ ANDROID</b> .....	<b>58</b>
5.3.1	Control d'accés i nova granja.....	59
5.3.2	Menú Principal.....	61
5.3.3	Indicadors .....	62
5.3.4	Economia.....	64
5.3.5	Llista de vaques .....	65
5.3.6	Detall d'una vaca.....	68
5.3.7	Filtratge de vaques.....	70
5.3.8	Llista de vedells .....	71
5.3.9	Detall d'un vedell.....	72
5.3.10	Filtratge de vedells.....	73
5.3.11	Dieta.....	74
5.3.12	Preferències .....	77
5.3.13	Menú d'opcions.....	79
5.4	<b>DISTRIBUCIÓ DE L'APLICACIÓ</b> .....	<b>81</b>
5.5	<b>CONTROL DE VERSIONS</b> .....	<b>82</b>
<b>6</b>	<b>PROVES</b> .....	<b>83</b>
6.1	<b>PROVES D'INTEGRACIÓ</b> .....	<b>83</b>
6.2	<b>PROVES D'INTERFÍCIE</b> .....	<b>84</b>
6.3	<b>PROVES FUNCIONALS</b> .....	<b>84</b>
6.4	<b>PROVES REALS</b> .....	<b>84</b>
<b>7</b>	<b>CONCLUSIONS</b> .....	<b>87</b>
7.1	<b>SEGUIMENT DEL PROJECTE</b> .....	<b>87</b>
7.2	<b>VALORACIÓ PERSONAL</b> .....	<b>87</b>
7.3	<b>FUTUR DEL PROJECTE I POSSIBLES MILLORES</b> .....	<b>88</b>
<b>8</b>	<b>BIBLIOGRAFIA</b> .....	<b>89</b>
8.1	<b>LLIBRES</b> .....	<b>89</b>
8.2	<b>REFERÈNCIES A INTERNET</b> .....	<b>89</b>

# INTRODUCCIÓ

## 1.1 ANTECEDENTS I PROBLEMA ACTUAL

Actualment els alumnes de primer curs de la Facultat de Veterinària de la UAB disposen d'un simulador d'una granja lletera que s'utilitza com a eina d'avaluació en una de les assignatures. Aquest simulador és una aplicació web desenvolupada en un Projecte de Final de Carrera d'Enginyeria Tècnica en Informàtica de Gestió de fa dos anys, sobre el que es basarà el nostre projecte.

Aquest simulador emula el funcionament d'una granja de vaques en aquells aspectes susceptibles d'ésser mesurats per a l'avaluació dels alumnes. Permet la interacció amb un conjunt de vaques i vedells (la granja) als que els alumnes han d'alimentar, inseminar, i realitzar un conjunt d'accions segons l'estat i situació de cadascun dels animals per tal d'assolir l'objectiu de fer la granja virtual el més rentable possible.

Per tal d'assolir les fites definides en l'assignatura els alumnes han d'accedir periòdicament al simulador durant el transcurs de tot el curs per comprovar l'estat dels animals i realitzar les accions pertinents per millorar la productivitat. Per tant s'ha de tenir en compte que els alumnes fan un ús intensiu de l'aplicació, actualment només accessible des de l'ordinador a través de la pàgina web.

Per millorar l'accés al simulador, millorar l'experiència dels alumnes i fomentar-ne l'ús s'ha plantejat com a solució habilitar una interfície d'accés mitjançant dispositiu mòbil. Això farà l'aplicació més atractiva als alumnes aprofitant l'al·licient que suposa l'ús noves tecnologies a la vegada que permetrà un accés molt més immediat a l'alumne, que no necessitarà estar davant de l'ordinador per tal de fer servir el simulador.

## 1.2 OBJECTIUS

El projecte té com a objectiu crear una aplicació per a mòbil amb les mateixes funcionalitats que l'aplicació web actual, que permeti als usuaris accedir de forma més freqüent i còmoda a realitzar les mateixes accions de seguiment i gestió necessàries per al correctament funcionament de la granja virtual. A més, s'ha de permetre als alumnes utilitzar qualsevol de les dues aplicacions per interactuar sobre la seva granja, que és única per alumne i curs. Això vol dir que l'estat de la granja s'haurà de mantenir consistent i sincronitzat independentment del mètode d'interacció que triï l'alumne cada cop que faci una simulació.

Aquest projecte també permetrà millorar la interacció entre l'alumne i el simulador gràcies a la inherent immediatesa i accessibilitat a l'usuari de les aplicacions mòbils i per tant obrirà un nou ventall de possibilitats per futures versions de l'aplicació.



Figura 1. de la Web al mòbil.



### 1.3 ESTAT DE L'ART

Un cop decidit quin és l'objectiu del projecte el primer pas és determinar si existeix al mercat algun producte que pugui satisfer les necessitats indicades directament o bé algun que es pugui adaptar per tal de satisfer-les. Pel que fa a eines de simulació de granges, hi ha aplicacions amb enfoc molt diversos: des d'enfocs totalment professionals com pot ser **Vaquitec** ([http://www.agritecsoft.com/sp/vaquitec/features\\_vac.php](http://www.agritecsoft.com/sp/vaquitec/features_vac.php)) que permet informes detallats i gestió del bestiar pel preu d'una llicència que va des dels 20€ al mes, fins a jocs de simulació com **Farming Simulator** (<http://www.farming-simulator.com/>) que en la versió de 2011 ha inclòs la possibilitat de compra i gestió de l'alimentació de vaques.

Ara bé, cap de les possibilitats existents compleix amb el requeriment bàsic de poder-se integrar amb l'aplicació web existent per permetre que l'usuari accedeixi des de qualsevol de les dues interfícies i per tant no té sentit plantejar l'aprofitament o adquisició de software ja desenvolupat. Per tant és necessari que implementem una aplicació a mida per satisfer les nostres necessitats. A tal efecte, haurem d'estudiar les tecnologies que existeixen actualment per desenvolupar aplicacions per mòbils, com son **Android** d'Open Handset Alliance o **iOS** d'Apple i determinar quina s'adapta millor a les nostres necessitats.

Tenint en compte l'evolució pel que fa a nombre de vendes de dispositius segons el sistema operatiu (Figura 2) i la situació actual del mercat podem fer-nos una primera idea de quines son les opcions més adients, i que estudiarem amb més detall en els capítols d'anàlisi i disseny d'aquesta memòria.

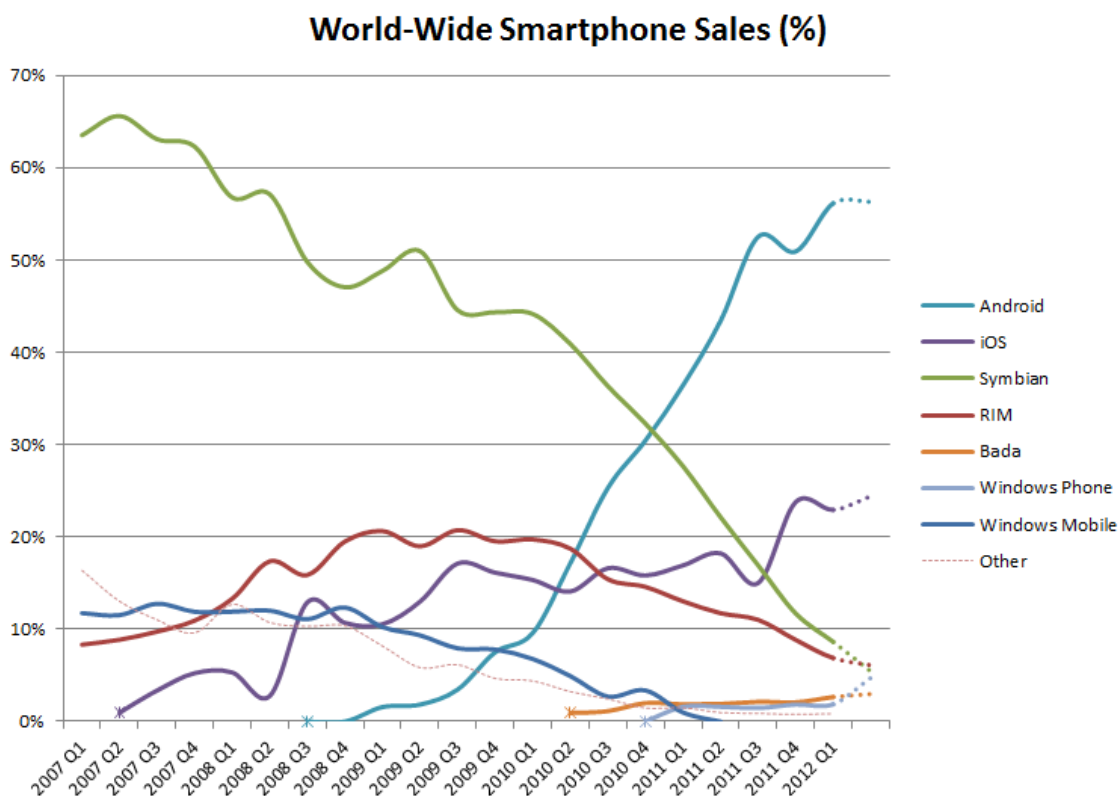


Figura 2. Distribució del mercat de dispositius mòbils segons sistema operatiu.

De la gràfica anterior es pot desprendre que hi han bàsicament dues opcions vàlides quan es planteja escometre un desenvolupament per a mòbil que tingui futur essent aquestes **Android** i **iOS**. Tant **Symbian** com **RIM** de Blackberry tenen clars símptomes de defalliment en contrast amb **Android** que està vivint una clara explosió i **iOS** que disposa del que sembla ser una sòlida base de clients al mercat.

D'aquesta manera concloem amb una proposta que, en principi inclourà una aplicació realitzada en **Android** o **iOS** que s'integrarà amb el programari existent per donar resposta als requeriments establerts.

#### 1.4 ESTRUCTURA DE LA MEMÒRIA

La memòria del projecte continua a partir d'aquest punt amb un estudi de viabilitat en el que s'analitzen les necessitats, es plantegen alternatives i s'identifiquen possibles riscos que puguin aparèixer durant el transcurs del

projecte i s'expliquen les solucions que es prendran donada la situació. Finalment s'estudien els costos, econòmics i temporals, i es calculen els beneficis previstos per tal de determinar la viabilitat del projecte.

A continuació, en el capítol d'anàlisi, s'estudien les funcionalitats que s'hauran d'implementar i es fa un anàlisi de les característiques de l'aplicació web actual amb la que el nostre projecte s'haurà d'integrar, es recullen les possibles especificacions no funcionals que puguin existir i, finalment, s'estudien les opcions tecnològiques disponibles.

Seguidament en el capítol de disseny es prenen les decisions tecnològiques pertinents com son les eines concretes que s'utilitzaran en el desenvolupament. També es defineix l'arquitectura de l'aplicació i es dissenya l'aspecte general que tindran les interfícies d'usuari. Finalment i després d'haver pres totes les decisions pertinents s'elabora la planificació detallada del projecte.

En el capítol d'implementació s'exposa el resultat del desenvolupament dels mòduls definits en el disseny anterior fent especial èmfasi en les parts que han estat més rellevants o han tingut alguna dificultat especial.

A continuació, En el capítol de proves es descriuen les validacions realitzades per tal de certificar el correcte funcionament i la qualitat del producte final tant pel que fa als aspectes tècnics, com és el rendiment, com als aspectes funcionals de l'aplicació.

Per finalitzar, en el darrer capítol, s'exposen les conclusions després de la finalització del projecte i es descriuen els possibles evolutius futurs i següents passos a seguir.



## 2 ESTUDI DE VIABILITAT

### 2.1 INTRODUCCIÓ

En aquest capítol es recull i estudia tota la informació necessària per decidir si el projecte és viable funcional, temporal i econòmicament.

També es proposa la solució a implementar així com una planificació aproximada on s'indiquen les fases principals del projecte i els recursos (de personal i econòmics) que es preveu dedicar a cadascuna d'elles.

### 2.2 ESTUDI DE LA SITUACIÓ ACTUAL

Des del curs passat part de l'assignatura de producció bovina disposa d'unes sessions pràctiques on els alumnes tenen a la seva disposició un simulador en entorn web d'una granja bovina que els permet experimentar amb molts dels factors que en una granja real repercuteixen en els nivells de producció de llet (Figura 3).

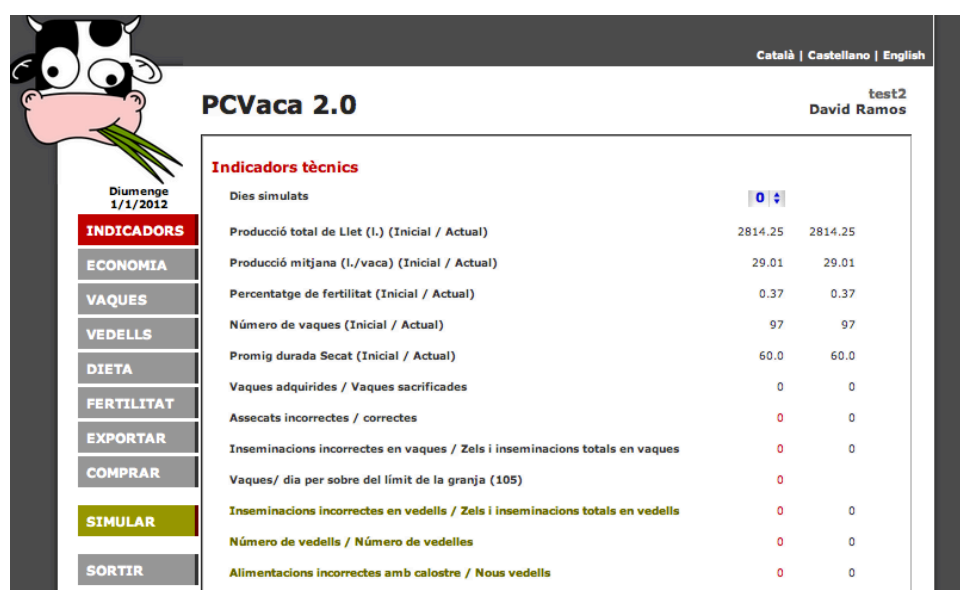


Figura 3. Aplicació web existent.

Aquest simulador ha tingut una bona acollida tant per part del alumnat com del professorat de l'assignatura i, des de la seva primera versió, ha estat sotmès a diferents actualitzacions i millores per aconseguir una simulació més realista.

Per al curs vinent s'ha proposat donar accés a aquest simulador a través d'una aplicació mòbil a més de l'accés actual mitjançant navegador. L'objectiu és donar més accessibilitat a l'aplicació i fomentar-ne així un major ús.

Aquesta nova funcionalitat està orientada en exclusiva als alumnes de l'assignatura ja que es considera que no és necessari traslladar les tasques d'administració que actualment fan els professors a un entorn mòbil.

### 2.3 REQUERIMENTS DEL PROJECTE

En aquest punt s'estableixen els objectius a assolir amb el projecte actual i es determinen els recursos que es consideren necessaris per tal de dur-lo a terme, tant humans com materials.

#### **Els objectius del projecte son:**

- Traslladar a un entorn mòbil l'accés a l'aplicació de simulació d'una granja bovina.
- Oferir una interfície de treball amigable i atractiva per a l'usuari, adaptada a les restriccions inherents dels dispositius mòbils.
- Garantir als alumnes les mateixes funcionalitats independentment del mètode d'accés.
- Estudiar la possibilitat de treballar sense connexió en la mesura del possible, garantint la protecció dels resultats de la simulació per part dels alumnes.

#### **Restriccions del sistema:**

- L'aplicació ha de ser compatible amb el major nombre de dispositius mòbils del mercat.
- A l'hora de tenir en compte la tecnologia per a implementar el servei s'ha de tenir en compte la utilitzada en l'aplicació web existent ja que s'han d'integrar l'una amb l'altra.

El projecte requerirà d'una sèrie de recursos humans en forma d'equip del projecte que hauran d'acomplir les tasques encomanades segons el seu perfil:

### **Equip del projecte**

**Cap de projecte** S'encarrega de la definició i gestió de les diferents fases del projecte garantint que els recursos es distribueixin de manera òptima, que els resultats parcials i finals son satisfactoris i que es compleixen les planificacions.

**Analista** Analitza l'aplicació tant des del punt de vista funcional com tècnic i col·labora amb el desenvolupador en el disseny. Col·labora amb el cap de projecte en la presa de requeriments.

**Desenvolupador** Dissenya la solució a partir de l'anàlisi realitzat per l'analista funcional i implementa l'aplicació.

**Tècnic de qualitat** Defineix el *test plan* a partir dels casos d'ús definits al funcional i s'encarrega de realitzar les proves que garanteixin la qualitat del producte resultant.

Pel que fa a recursos materials caldrà un equip des del qual desenvolupar qualsevol codi amb el programari necessari. També caldrà accés al servidor de la UAB on està ubicada l'aplicació web actual i un dispositiu mòbil on realitzar proves reals del codi desenvolupat.

### **Producte i documentació del projecte**

- S'ha de lliurar la memòria del projecte.
- S'ha de publicar una aplicació que pugui ser descarregada des d'un dispositiu mòbil.

## 2.4 PROPOSTA DE SOL·LUCIÓ I ALTERNATIVES

Proposem desenvolupar una aplicació accessible des del major nombre de dispositius mòbils possible, que s'integri amb l'aplicació actual, permetent als alumnes consultar e interaccionar amb les seves granges (Figura 4).



Figura 4. Esquema lògic: L'alumne accedeix a la mateixa granja des de qualsevol de les dues interfícies.

Cal remarcar que el projecte es centra única i exclusivament en els usuaris que tenen rol d'alumne i les accions i funcionalitats disponibles per a aquest perfil. Per tant no es contempla oferir una solució a les tasques administratives del curs accessibles per als professors i administradors ja que s'entén que aquestes tasques són puntuals i per tant no es beneficien de l'accessibilitat que permeten les aplicacions mòbils, a part de disposar de funcionalitats (com al càrrega d'arxius .csv) difícilment reproduïbles en un dispositiu mòbil.

La primera tasca a realitzar serà analitzar l'aplicació web actual per tal d'extreure'n totes les funcionalitats que estan a disposició de l'alumne. Aquestes són les funcionalitats que haurem de traslladar a l'aplicació que desenvolupem i per tant han d'identificar-se des d'un primer moment.



A continuació és necessari fer un estudi de l'arquitectura de l'aplicació actual i esbrinar quines opcions de comunicació ofereix. Això és així ja que un requisit indispensable de la nostra aplicació és la integració amb l'aplicació actual. Per tant haurem de saber quina informació podem extreure i de quina manera hi podem accedir.

Un cop realitzades les tasques descrites anteriorment dissenyarem i implementarem les solucions a les funcionalitats identificades mitjançant la tecnologia de desenvolupament per a dispositiu mòbil que finalment triem.

El resultat del pas anterior serà una aplicació per a mòbil que publicarem i podrà ser descarregada i instal·lada al dispositiu per tal d'interactuar amb la granja.

## 2.5 PLANIFICACIÓ DEL PROJECTE

En aquest apartat de la memòria es descriuen les tasques que compondran el projecte que volem realitzar així com una estimació preliminar del temps que suposarà cadascuna d'elles.

### Tasques del projecte

- **Inici del projecte:** en aquesta fase s'estableixen els objectius principals del projecte i es realitzen les tasques administratives com la matrícula i l'assignació del projecte. També es realitzarà en aquesta fase l'estudi de viabilitat per assegurar en la mesura del possible que el projecte té uns objectius assequibles.
- **Fase d'anàlisi:** s'establiran amb detall les funcionalitats i casos d'ús que ha de tenir l'aplicació. També es farà un estudi del marc tecnològic i es proposaran les diferents opcions tecnològiques disponibles.

- **Fase de disseny:** amb les funcionalitats a clares es triaran les eines tecnològiques i definirà l'arquitectura de l'aplicació. Es decidiran les capes que tindrà el projecte, els punts de comunicació amb el servidor del projecte web i les línies generals del disseny de les interfícies.
- **Fase d'implementació:** en aquesta fase es desenvoluparà el codi de l'aplicació que oferirà les funcionalitats desitjades.
- **Fase de proves:** en aquesta fase es provarà el codi implementat tant a nivell tècnic com funcional, es faran proves de rendiment i circuits de proves simulant l'ús que els alumnes donaran a l'aplicació un cop aquesta estigui en funcionament.
- **Fase d'implantació:** en aquesta fase el projecte s'instal·larà a producció en el mateix entorn en el que hi ha l'aplicació actual i es realitzaran les proves definitives d'integració sobre l'entorn real. També es definirà el sistema de distribució de l'aplicació.
- **Fase de final projecte:** en aquesta fase es documentarà en la memòria del projecte tot el necessari i es prepararà la defensa del mateix davant del tribunal.

El projecte es desenvoluparà amb una dedicació mitjana de 12 hores setmanals, des del 17 d'octubre fins a la data de finalització. Segons els càlculs realitzats el temps necessari per realitzar totes les tasques del projecte serà d'aproximadament 205 hores. Durant el transcurs del projecte es realitzaran, generalment de forma seqüencial, les tasques que es descriuen en la Figura 5.

ID	Task Name	Duration	Start	Finish
1	<b>Projecte</b>	<b>40,25 days</b>	<b>Mon 17/10/11</b>	<b>Fri 20/04/12</b>
2	Assignació i matriculació del projecte	1 hr	Mon 17/10/11	Mon 17/10/11
3	Reunió de presa de requeriments	12 hrs	Wed 09/11/11	Sat 10/03/12
4	<b>Inici del projecte</b>	<b>1,63 days</b>	<b>Mon 02/01/12</b>	<b>Mon 09/01/12</b>
5	Anàlisi de l'aplicació actual	8 hrs	Mon 02/01/12	Thu 05/01/12
6	Estudi de viabilitat	4 hrs	Fri 06/01/12	Sat 07/01/12
7	Aprovació de l'estudi de viabilitat	1 hr	Mon 09/01/12	Mon 09/01/12
8	<b>Fase d'anàlisi</b>	<b>2,25 days</b>	<b>Mon 09/01/12</b>	<b>Thu 19/01/12</b>
9	Anàlisi de l'aplicació actual	10 hrs	Mon 09/01/12	Sat 14/01/12
10	Definició d'especificacions	4 hrs	Sat 14/01/12	Tue 17/01/12
11	Anàlisi del marc tecnològic	4 hrs	Tue 17/01/12	Thu 19/01/12
12	<b>Fase de disseny</b>	<b>3,25 days</b>	<b>Thu 19/01/12</b>	<b>Fri 03/02/12</b>
13	Definició d'arquitectura	4 hrs	Thu 19/01/12	Sat 21/01/12
14	Definició de la comunicació entre aplicacions	8 hrs	Sat 21/01/12	Thu 26/01/12
15	Disseny de les interfícies	14 hrs	Thu 26/01/12	Fri 03/02/12
16	<b>Fase d'Implementació</b>	<b>9,5 days</b>	<b>Fri 03/02/12</b>	<b>Mon 19/03/12</b>
17	Implementació del mòdul de comunicació	16 hrs	Fri 03/02/12	Mon 13/02/12
18	Implementació de les interfícies	20 hrs	Mon 13/02/12	Fri 24/02/12
19	Implementació de les funcionalitats	40 hrs	Fri 24/02/12	Mon 19/03/12
20	<b>Fase de Proves</b>	<b>1,88 days</b>	<b>Mon 19/03/12</b>	<b>Tue 27/03/12</b>
21	Definició del test plan	1 hr	Mon 19/03/12	Mon 19/03/12
22	Proves de comunicació	4 hrs	Tue 20/03/12	Wed 21/03/12
23	Proves funcionals	8 hrs	Thu 22/03/12	Mon 26/03/12
24	Report de resultats	2 hrs	Tue 27/03/12	Tue 27/03/12
25	<b>Implantació</b>	<b>2 days</b>	<b>Wed 28/03/12</b>	<b>Thu 05/04/12</b>
26	Instal.lació	4 hrs	Wed 28/03/12	Thu 29/03/12
27	Proves de comunicació reals	4 hrs	Fri 30/03/12	Sat 31/03/12
28	Validació del projecte amb el client	8 hrs	Mon 02/04/12	Thu 05/04/12
29	<b>Final del projecte</b>	<b>3,25 days</b>	<b>Fri 06/04/12</b>	<b>Fri 20/04/12</b>
30	Composició de la memòria a partir de la documentació generada	20 hrs	Fri 06/04/12	Tue 17/04/12
31	Preparació de la defensa del projecte	6 hrs	Wed 18/04/12	Fri 20/04/12

Figura 5. Tasques identificades per a l'execució del projecte.

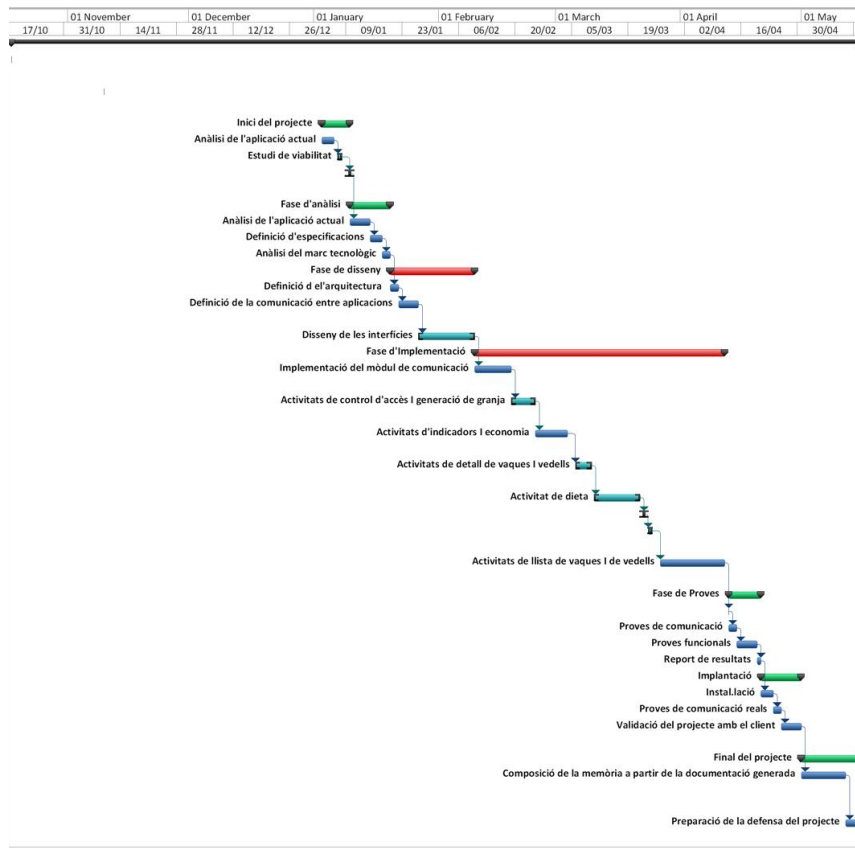


Figura 6. Diagrama de Gantt.

## 2.6 AVALUACIÓ DE RISCOS

L'objectiu d'aquest apartat és identificar possibles riscos que ens podem trobar durant l'execució del projecte i la implantació del mateix. Evidentment no és possible preveure totes les avinenteses possibles però sí com a mínim les més probables per preveure'n la solució.

Per tant a continuació descriurem els diferents riscos que preveiem ens podem trobar del projecte així com els plans de contingència previstos per cadascun d'ells.

### *Planificació equivocada*

**Risc:** no es finalitza el projecte en la data prevista.

**Pla de contingència:** descartar per a la primera entrega aquelles funcionalitats que es considerin secundaries.

#### *No s'identifica alguna tasca necessària*

**Risc:** s'entrega un producte que no resulta útil per a l'usuari

**Pla de contingència:** revisar l'estudi de viabilitat i la planificació en conseqüència.

#### *Modificació de requeriments per part del client*

**Risc:** els nous requeriments no estan contemplats en la planificació i l'entrega es veu afectada.

**Pla de contingència:** negociar amb el client les funcionalitats, deixant per noves versions aquelles que consideri menys rellevants i que suposin un pes similar en recursos als requeriments afegits.

#### *El producte no és usable*

**Risc:** els usuaris troben la interfície incòmode o deficient i no utilitzen l'aplicació

**Pla de contingència:** replantejar la interfície. Estudiar alternatives per dotar d'una millor usabilitat a l'aplicació.

#### *No es pot garantir la seguretat de les dades*

**Risc:** els alumnes poden modificar les dades simulades en el seu benefici fent impossible la avaluació

**Pla de contingència:** revisar de nou els sistemes de seguretat per a garantir la garantia. Si és imprescindible descartar la opció de treballar sense connexió. Revisar l'estudi de viabilitat amb les noves conclusions.

#### *Dificultats per a arribar a l'usuari*

**Risc:** l'usuari no coneix l'aplicació o els seus beneficis i per tant no la utilitza.

**Pla de contingència:** planificar uns millors manuals d'usuari, fer l'aplicació més atractiva i determinar una millor estratègia de venda.

## 2.7 AVALUACIÓ DE COSTOS I BENEFICIS

Actualment es disposa d'una aplicació web que serveix com a suport als alumnes per a la part de pràctiques de la seva assignatura, per tant les funcionalitats bàsiques desitjades estan cobertes. Ara bé, l'adaptació per a dispositius mòbils de l'aplicació comporta nombrosos beneficis.

Per començar, el grau i freqüència d'interacció entre usuari i aplicació que es pot esperar en una aplicació mòbil és molt més elevat que en un aplicació web ja que l'usuari normalment té accés constant al mòbil mentre que connectar-se a una web és una acció que ha de fer de forma puntual i voluntària. Això el que aconseguix és, per una banda, fer que els alumnes accedeixin i interactuïn amb més freqüència i, per altra banda, permet dotar de més complexitat a l'aplicació.

A més, el fet de poder assumir que l'alumne pot interactuar en qualsevol moment obre tot un ventall de possibilitats pel que fa a futures funcionalitats que es poden dissenyar per al simulador que son impensables en un entorn web com poden ser alertes des del servidor o la simulació en temps real.

Aquestes noves funcionalitats aporten beneficis en dos sentits: en primer lloc enriqueixen l'experiència de l'alumne permetent als professors apropar més a la realitat la simulació i, a la mateixa vegada, fan l'eina més interessant per a possibles compradors, fora de la UAB, augmentant-ne el valor intangible. Aquest últim punt és doblement vàlid en el moment actual ja que el fet de tenir l'aplicació adaptada al mòbil és ara per ara una eina de màrqueting molt rellevant ja que les aplicacions per a mòbil es troben actualment en un moment d'expansió tal i com es pot comprovar en la Figura 7.

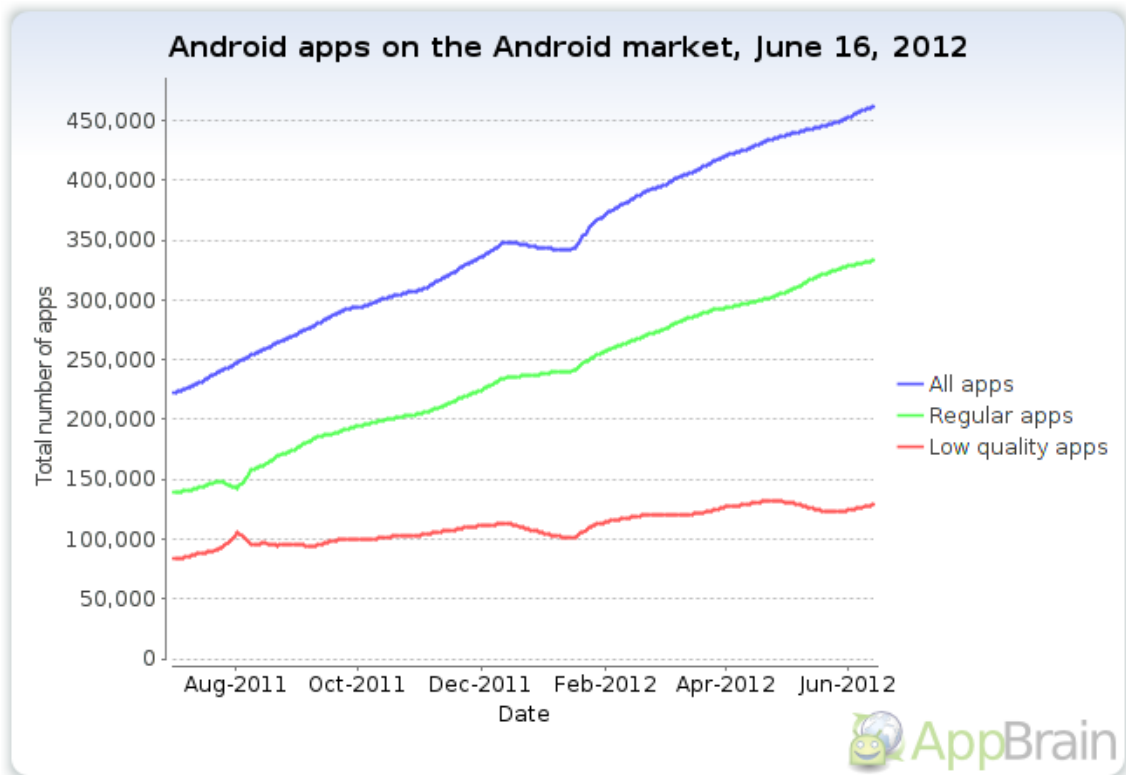


Figura 7. Evolució del nombre d'aplicacions Android publicades.

### Estimació de cost de personal

D'aquesta manera si considerem els següents costos pel que fa a la implementació i implantació de l'aplicació, extrets de la planificació de l'apartat anterior:

Nom del recurs	Tasa estándar	Cost
Cap de Projecte	60,00 €/hora	4.580,00 €
Analista	40,00 €/hora	960,00 €
Desenvolupador	40,00 €/hora	4.440,00 €
Tècnic qualitat	30,00 €/hora	300,00 €

Establím que el cost total del projecte és de 10.280 €.

Si suposem que el fet de disposar de l'aplicació adaptada al mòbil serà un factor decisiu en l'increment de l'expansió de l'aplicació a altres universitats d'un 100% i assumim que durant aquest any l'aplicació s'ha estès a tres noves

universitats podem atribuir els beneficis que s'extreuen de 3 llicències al projecte actual.

Sota aquesta premissa, si assumim que aquestes universitats faran servir l'aplicació per a una de les seves assignatures en classes amb una mitja de 40 alumnes i tenim en compte que aquests alumnes hauran pagat de mitja 160€ de matrícula per a l'assignatura, dels quals una tercera part va destinada a pràctiques, podem establir una llicència a cada universitat de:

$$\frac{160}{3} \times 40 = 2133\text{€/any}$$

Per tant podem assumir que el projecte s'amortitzarà en uns 20 mesos, generant beneficis nets a partir d'aquest moment.

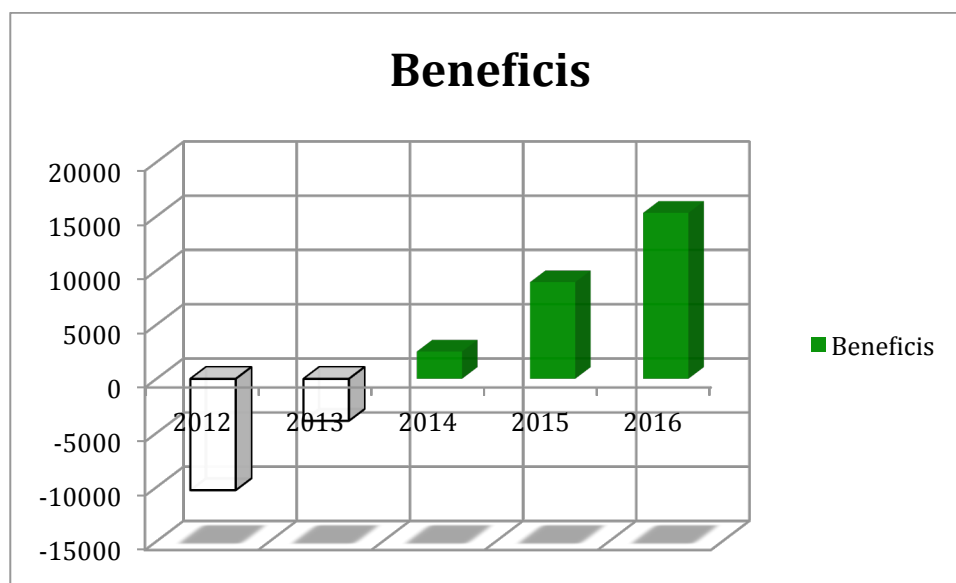


Figura 8. Beneficis generats pel projecte respecte la inversió inicial.

Finalment i com a consideració a tenir en compte, el cost d'adaptar l'aplicació al entorn mòbil és una inversió que es farà inevitable a curt o mig termini donat el fet que tota la tecnologia, segons tots els indicadors mesurables, estan gravitant de forma inexorable cap aquesta direcció. Un exemple representatiu d'aquesta afirmació és Facebook: dels més de 900 milions d'usuaris actius de que disposa, a data de maig de 2012, més de la meitat hi accedeixen des d'aplicacions mòbils (font: <http://en.wikipedia.org/wiki/Facebook>).



Per tant, podem preveure sense por a equivocar-nos massa que tard o d'hora tots els desenvolupaments que actualment existeixen en entorn web hauran d'adaptar-se o perdre competitivitat. Per aquest motiu fer l'adaptació en aquest moment, a més de garantir una imatge de modernitat, suposa un estalvi a futur quan s'hagi de fer de forma obligatòria el canvi de tecnologia.

## 2.8 CONCLUSIONS

En aquest estudi de viabilitat podem concloure que, en primer lloc, existeixen recursos tècnics que estan a l'abast de qui realitza el projecte, per tal de donar resposta a les necessitats que es presenten. En segon lloc, s'ha exposat que es disposa del temps necessari per tal d'assolir els objectius marcats dins del curs actual i, finalment s'ha demostrat que econòmicament el projecte no és només viable si no que a més aportarà rendibilitat a la facultat en un temps més que raonable.



## **3 ANÀLISI**

### **3.1 INTRODUCCIÓ**

En aquest apartat de la memòria del projecte s'exposen les necessitats a nivell funcional del projecte i se n'analitzen tots els aspectes rellevants per tal d'abordar la fase de disseny amb les màximes garanties.

En primer lloc es realitza un estudi de l'aplicació web actual i se n'extreuen les funcionalitats que es volen portar a l'aplicació mòbil. D'aquesta manera s'obté el llistat d'especificacions funcionals del projecte.

També és en aquest apartat s'especifiquen els possibles requeriments no funcionals del projecte com poden ser restriccions d'usabilitat o rendiment i s'estudia el marc tecnològic de l'aplicació.

### **3.2 ANÀLISI DE L'APLICACIÓ ACTUAL**

#### **3.2.1 Anàlisi funcional**

L'aplicació que actualment hi ha desenvolupada té dos objectius clarament diferenciats:

- Com a aplicació per que els professors administrin les granges dels seus alumnes, en realitzin un seguiment detallat i hi interaccionin mitjançant el canvi de variables disponibles a la granja.
- Com a aplicació per la realització de la simulació per part dels alumnes, que els ofereix tant les interfícies on poden interactuar amb les dades de la seva granja i prendre les decisions necessàries.

Aquesta primera diferenciació és rellevant per al projecte en curs ja que s'ha de tenir clar des del primer moment que l'objectiu és donar una alternativa al segon perfil de l'aplicació, és a dir, a la funcionalitat disponible pels alumnes per a la simulació de dades en les seves granges.

Això és així per diversos motius, en primer lloc per què la tasca d'administració és molt més esporàdica que la de simulació i per tant no justifica el cost de desenvolupar una alternativa. A més per la naturalesa de les tasques d'administració aquestes son més còmodament realitzables des d'un PC com pot ser, entre altres, la càrrega de llistats d'alumnes per a un curs determinat a partir d'arxius .csv o la generació de còpies de seguretat de la base de dades.

La part de simulació de granges, per altra banda sí que reuneix les característiques necessàries per tal de ser desenvolupada per un dispositiu mòbil amb millores considerables.

Aquesta part de simulació és la que estudiem detalladament a nivell funcional per tal d'identificar els casos d'ús que traslladarem al nostre programari. Principalment serà tota aquella funcionalitat que permeti a l'alumne realitzar qualsevol acció sobre la granja o qualsevol element de la mateixa i també tota aquella funcionalitat que pugui ser d'ajuda a l'alumne per prendre les decisions necessàries sobre les accions a emprendre. El detall de l'anàlisi realitzat es pot trobar al capítol ESPECIFICACIONS FUNCIONALS.

### **3.2.2 Anàlisi tècnic**

Ja que una de les condicions bàsiques és que l'aplicació actual i la que estem desenvolupant han d'estar integrades i sincronitzades en tot moment, cal veure com està estructurada l'arquitectura de l'aplicació web per decidir quina és la millor manera de comunicar-s'hi.

Aquest anàlisi ens descobreix que l'aplicació web existent s'ubica en un servidor de la UAB i consta d'un model client servidor amb tres capes tal i com es pot veure a la figura 10:

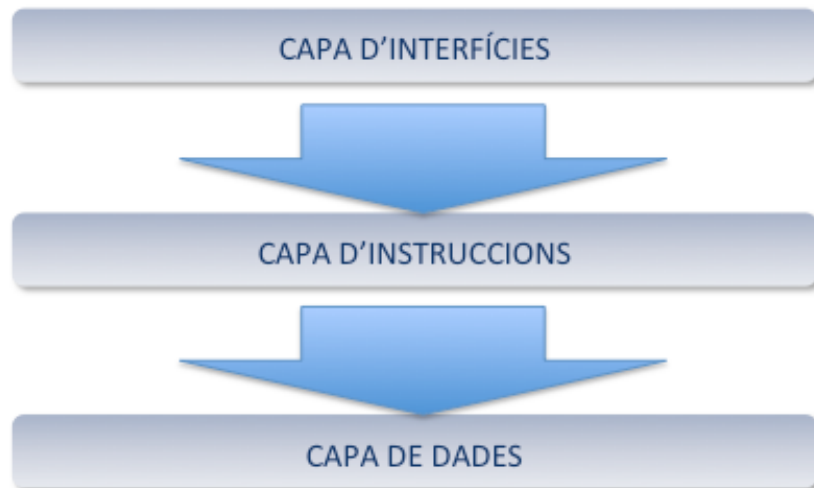


Figura 9. Diagrama de capes de l'aplicació web.

- **La capa d'interfícies:** consta d'una sèrie de petites aplicacions allotjades en la part de servidor que tenen com a objectiu realitzar cadascuna una acció molt concreta anomenades *servlets*. Aquests *servlets* generen el codi HTML de les pàgines (que a la seva vegada invoca a altres *servlets*) de les que consta l'aplicació i fan ús de **la capa d'instruccions** per accedir a consumir i modificar les dades.
- **La capa d'instruccions:** fa de connexió entre la **capa d'interfícies** i la **capa de dades** i a la vegada conté totes les funcions i la lògica de negoci que fan que la simulació modifiqui l'estat de les vaques, vedells i la granja segons les decisions preses per l'usuari.
- **La capa de dades:** dona persistència a la informació de la simulació. És en aquesta capa on s'ubica la base de dades i on es manté la informació de les granges, cursos i alumnes.

Aquesta estructura s'ubica en un servidor Tomcat que permet exposar a Internet els *servlets* abans esmentats els quals s'invoquen a través de peticions GET i POST. Aquests *servlets* son un recurs del llenguatge orientat a objectes Java i retornen respostes en forma de codi HTML al navegador (Figura 10).

```

public final void doGet(final HttpServletRequest req,
final HttpServletResponse response) throws ServletException, IOException {
    HttpSession sesion = req.getSession(true);
    if (isLogueado(req)) {
        if ((Integer) sesion.getAttribute("rol") == Utils.ADMINISTRADOR) {
            ResourceBundle contenidos = cargarIdioma(req);
            PrintWriter out = response.getWriter();
            response.setContentType("text/html");
            printCabecera(out, req);

            out.println("                <div id='Llistat'>");
            out.println("                <div
class='in'>");
            out.println("                <h2>" +
contenidos.getString("admincursoTit") + "</h2>");
            out.println("                </div>");
            out.println("                </div>");
            out.println("                </div>");
            out.println("                <div class='borde'>");

```

Figura 10. Exemple de codi d'un dels servlets de la capa d'interfícies.

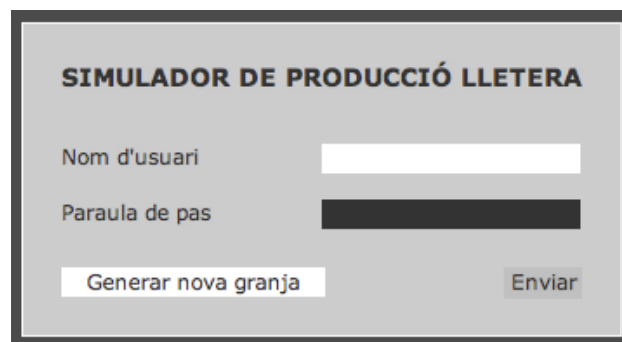
La capa d'instruccions s'implementa a partir d'una sèrie de classes Java utilitzades pels *servlets* de la capa d'interfícies que s'encarreguen de fer les operacions corresponents i d'accedir i actualitzar les dades de la capa de dades.

Les dades de la granja s'emmagatzemen en el servidor gràcies a una Base de Dades MySQL que emmagatzema totes les entitats de l'aplicació en taules relacionals. Aquesta informació va des de les dades dels alumnes fins a les característiques pròpies de cada una de les vaques de cada alumne de cada curs.

### 3.3 ESPECIFICACIONS FUNCIONALS

Analitzada l'aplicació web extraiem una sèrie de funcionalitats que volem implementar en la nostra aplicació i que a son els que es llisten a continuació:

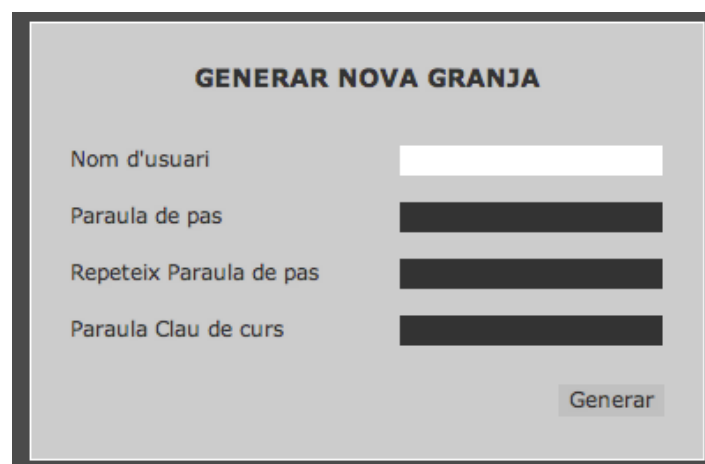
- **Validació d'accés:** aquesta funcionalitat garanteix el control d'accés dels alumnes per tal que cada alumne pugui accedir únicament a la seva granja i també evitar que qualsevol altre usuari no autoritzat pugui influir en l'estat de les simulacions (Figura 11).



The screenshot shows a login interface with the title 'SIMULADOR DE PRODUCCIÓ LLETERA'. It contains two input fields: 'Nom d'usuari' (username) and 'Paraula de pas' (password). Below the fields are two buttons: 'Generar nova granja' (Generate new farm) and 'Enviar' (Send).

Figura 11. Pantalla d'inici.

- **Generació de la granja:** es tracta d'una funcionalitat que l'alumne ha d'executar una única vegada a l'inici del curs i que genera una granja completa amb tot el conjunt d'animals corresponent i creant de forma automàtica tot un històric (Figura 12).



The screenshot shows a form titled 'GENERAR NOVA GRANJA'. It includes four input fields: 'Nom d'usuari' (username), 'Paraula de pas' (password), 'Repeteix Paraula de pas' (Repeat password), and 'Paraula Clau de curs' (Course key word). A 'Generar' (Generate) button is located at the bottom right.

Figura 12. Pantalla d'entrada de dades per la generació d'una granja a un alumne nou.

- **Mòdul d'indicadors:** en aquest mòdul es mostra una visió general de l'històric de la granja, permetent a l'alumne tenir una visió global dels indicadors més rellevants del funcionament de la granja així com una visió de l'evolució de la producció al llarg del temps.

L'objectiu d'aquest mòdul és mostrar la informació a l'alumne per tal que aquest pugui prendre les decisions necessàries pel correcte funcionament de la granja i també disposi d'una visió dels errors comesos (Figura 13).

<b>Indicadors tècnics</b>		
Dies simulats	58 ↓	
Producció total de Llet (l.) (Inicial / Actual)	3182.13	3308.35
Producció mitjana (l./vaca) (Inicial / Actual)	30.59	31.21
Percentatge de fertilitat (Inicial / Actual)	0.38	0.36
Número de vaques (Inicial / Actual)	104	106
Promig durada Secat (Inicial / Actual)	60.0	59.56
Vaques adquirides / Vaques sacrificades	9	10
Assecats incorrectes / correctes	13	2
Inseminacions incorrectes en vaques / Zels i inseminacions totals en vaques	49	70
Vaques/ dia per sobre del límit de la granja (105)	43	

Figura 13. Indicadors genèrics de la granja.

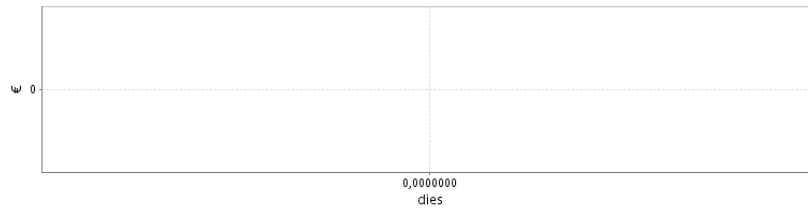
- **Mòdul d'economia:** en aquest mòdul es mostren els indicadors econòmics rellevants de la granja: costos dels recursos, ingressos per venda, etc. de forma que l'alumne pot veure de forma general totes aquelles dades que afecten al balanç econòmic de la granja. Igualment que en el mòdul anterior, aquesta funcionalitat és purament informativa i ha de servir a l'alumne per avaluar el correcte desenvolupament de la granja, per tant des d'aquest mòdul no hi hauran accions específiques a realitzar (Figura 14).



### Indicadors econòmics

Caixa **9700.0€** Ingressos **0.0€** Costos **0.0€**

GRÀFICA D'EVOLUCIÓ DE CAIXA (BLAU), INGRESSOS (VERD) I DESPESES (VERMELL)



Preu del litre de llet	0.32€	Ingres per sacrificar una vaca	800.0€
Preu de venda d'una vedella	120.0€	Preu de venda d'un vedell	60.0€
Cost de compra d'una vaca lactant	1800.0€	Cost de manteniment d'un vedell	2.0€
Cost de manteniment d'una vaca	3.5€	Cost de inseminació	20.0€

Figura 14. Indicadors econòmics de la granja.

- **Mòdul de gestió de vaques:** aquesta part de l'aplicació permet la gestió, control e interacció amb tot el conjunt de vaques que conformen la granja d'un alumne.

Mitjançant una llista amb diverses opcions d'ordenació i filtratge l'alumne interactua sobre cadascuna de les vaques, accedint a la informació en detall de les mateixes, realitzant les accions pertinents per a garantir el correcte funcionament de la granja procurant mantenir al màxim possible el nivell de producció de les vaques (Figura 15).

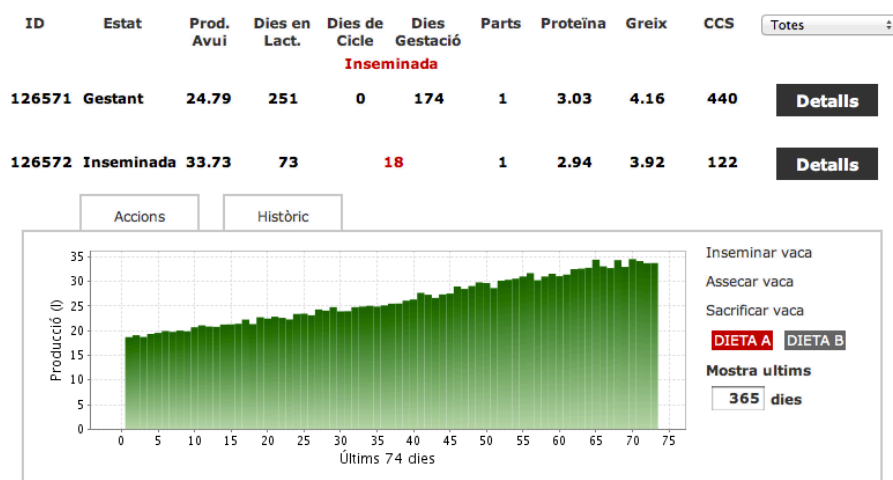


Figura 15. Llistat de vaques amb el detall i gràfica d'una vaca concreta.

- Mòdul de gestió de vedells:** a nivell de funcionalitats aquest mòdul és molt similar al de gestió de vaques ja que consisteix en un llistat ordenable i filtrable en el qual s'actua sobre els seus elements. La diferència amb l'anterior és a nivell de negoci ja que en el cas dels vedells les accions a realitzar per part de l'alumne estan més encarades a garantir la salut dels vedells que a optimitzar la producció (Figura 16).

ID	Estat	Alçada (cm)	Pes (Kg)	Dies de Cicle	Dies Gestació	Edat (dies)	Totes
<b>Inseminada</b>							
126668	(F) Gestant	156.37	562.59	0	147	697	<b>Detalls</b>
126669	(F) Inseminada	134.74	370.88	20		440	<b>Detalls</b>

Accions    Històric

Tria la dieta:

Creixement A   
  Creixement B   
  Creixement C

Ració de Creixement A

Inseminar vedella

Vendre vedell

Figura 16. Llistat de vedells amb el detall d'un d'ells.

- Mòdul de dieta:** aquest mòdul és molt rellevant ja que mitjançant les funcionalitats que ofereix l'alumne compon la dieta de les vaques de la seva granja. Aquesta configuració afecta directament als costos de la granja per una banda, a través del cost dels ingredients, i per l'altra aquesta composició és un dels factors més determinants del nivell de producció de les vaques. Per tant l'alumne ha de trobar l'equilibri entre el preu de la dieta i la qualitat d'aquesta i anar adaptant la composició segons van variant els preus dels ingredients (Figura 17).

**Gestor d'alimentació de les vaques**

**DIETA A**

Nom	PB	EE	Cen	FND	CNF	ENL	MS	€/Kg	Kg/V	V	Kg	€
Ensitjat blat de moro	8.8	3.2	4.3	45.0	38.7	1.45	0.0	0.15	10.0	85	850.0	127.5
Fenc alfals	17.8	1.6	9.2	50.9	20.5	1.13	0.0	0.135	3.0	85	255.0	34.42
Blat de moro	9.4	4.2	1.5	9.5	75.4	2.01	0.0	0.273	4.0	85	340.0	92.82
Ordi	12.4	2.2	2.9	20.8	61.7	1.86	0.0	0.22	3.0	85	255.0	56.1
Soia 44	49.9	1.6	6.6	14.9	27.0	2.13	0.0	0.304	1.0	85	85.0	25.84
Bagaç de cervesa	28.4	5.2	4.9	47.1	14.4	1.71	0.0	0.15	2.0	85	170.0	25.5
<b>TOTALS (%/KG)</b>	<b>14.03</b>	<b>3.13</b>	<b>4.42</b>	<b>35.31</b>	<b>43.08</b>	<b>1.61</b>	<b>0.0</b>					
<b>TOTALS</b>	<b>322.89</b>	<b>72.2</b>	<b>101.69</b>	<b>812.2</b>	<b>991.0</b>	<b>37.06</b>	<b>0.0</b>	<b>4.26</b>	<b>23.0</b>	<b>85</b>	<b>1955.0</b>	<b>362.18</b>

Figura 17. Composició de la dieta de la granja.

- **Mòdul de fertilitat:** es tracta d'una taula que mostra l'històric d'inseminacions de les vaques de la granja i el resultat de les mateixes. L'ús d'aquesta funcionalitat es limita a detectar aquelles vaques que tenen un nivell de fertilitat significativament inferior a la mitja (Figura 18).

Vaques	Intent: 1	Intent: 2	Intent: 3	Intent: 4	1
Vaca: 126571	19/6/2011	10/7/2011			
Vaca: 126572	14/12/2011				
Vaca: 126573	22/8/2011				
Vaca: 126574	24/9/2011				
Vaca: 126575	30/10/2011				
Vaca: 126576	2/11/2011	25/11/2011	17/12/2011		

Figura 18. Taula que mostra l'històric d'inseminacions dels animals.

- **Compra de vaques:** aquesta funcionalitat simplement afegeix un animal generat automàticament a la granja existent al cost indicat en el mòdul d'economia. L'animal apareix en el primer dia de lactància després del seu primer part. L'ús d'aquesta funcionalitat està limitada ja que d'altra manera es corre el risc de fer servir la compra de vaques de forma continuada per tractar de millorar la producció ja que les vaques acabades de comprar sempre produeixen una quantitat de llet sobre la mitja.
- **Simulació:** aquesta funcionalitat avança un dia en el calendari del simulador. Aquesta és l'acció principal del simulador ja que desencadena tots els mecanismes necessaris en tots els apartats, actualitzant tots els valors i estats segons les accions realitzades per l'usuari des de la darrera simulació.

### 3.4 ESPECIFICACIONS NO FUNCIONALS

A més de les especificacions funcionals indicades en l'apartat anterior hi ha una sèrie d'aspectes que s'han de tenir en compte a l'hora d'abordar el disseny del projecte per tal de garantir en la mesura del possible l'èxit del mateix. L'aplicació ha d'esser **compatible** amb el major nombre de dispositius mòbils possibles. Això és així ja que hem d'intentar arribar al major nombre d'usuaris.

La **usabilitat** del programa ha de ser una part molt cuidada del mateix. S'ha de tenir en compte que és una aplicació molt especialitzada en la que l'usuari ha de poder veure i interactuar amb un gran volum de dades de forma simultània. Això generalment és un aspecte problemàtic per a les aplicacions mòbils donada la inherent escassetat de recursos pel que fa a la mida de la pantalla, així que s'haurà de posar especial èmfasi en aquest aspecte.

Donat que l'aplicació ha d'integrar-se amb una aplicació existent haurem de triar solucions tècniques que facilitin aquesta integració. Sabem que l'aplicació web està programada en Java pel que fa als *servlets* i la capa d'instruccions i també sabem que l'aplicació retorna a les peticions realitzades codi HTML. Per tot això a l'hora de tirar com integrem l'aplicació que programarem hem de tenir en compte que, o bé interpretem les respostes HTML que actualment es retornen, reaprofitant així els *servlets* existents o bé creem serveis de comunicació propis amb respostes a mida per a les nostres necessitats. En aquest darrer cas haurem de tenir en compte que haurem d'afegir nou codi a la part de servidor (en llenguatge Java).

Finalment i sigui quina sigui la decisió que prenguem, com que l'aplicació haurà d'intercanviar dades amb el servidor serà important minimitzar el transit d'informació que requereixi l'aplicació. Aquesta restricció és important ja que les connexions per dispositiu mòbil normalment encara tenen límits de tràfic de dades associats a les tarifes i la aplicació que desenvolupem ha de ser el menys costosa possible en aquest aspecte per no des incentivar-ne l'ús.

### 3.5 MARC TECNOLÒGIC

Actualment existeixen nombroses opcions a l'hora de fer un desenvolupament per a dispositius mòbils. S'ha realitzat un anàlisi exhaustiu de les principals opcions que aporta el mercat i s'han avaluat segons les necessitats del projecte. S'ha considerat imprescindible la opció seleccionada estigui suficientment establerta, minimitzant el risc de trobar problemes tècnics no resolts. També s'ha tingut en compte que l'opció de desenvolupament ofereixi un kit de desenvolupament de software (SDK) i un entorn de desenvolupament adequat.



Figura 19. Apple contra Android.

A l'hora de realitzar aquest estudi, les dues principals plataformes de desenvolupament que compleixen amb les condicions descrites són Android i iOS SDK. Per tant hem avaluat les característiques considerades com a més rellevant d'aquestes dues plataformes:

- **Accessibilitat:** a l'hora de desenvolupar per Android es pot fer servir *Eclipse*, que és software lliure i es pot instal·lar a qualsevol sistema operatiu (Windows, Linux i Mac OS) mentre que per a desenvolupar en iOS SDK cal obligatòriament un Mac i, d'un software específic per al desenvolupament anomenat *Xcode* que el qual pot ésser de pagament depenent de la versió.
- **Grau d'implantació:** Android està molt més estès pel que fa a dispositius que no pas iOS SDK que està limitat a dispositius d'Apple.

- **Llenguatge de programació:** Android es programa en Java, que és un llenguatge de programació molt extens mentre que iOS SDK es programa amb Objective-C. Tot i que ambdós llenguatges tenen similituds (una sintaxi heretada de C i mateixos tipus primitius) la corba d'aprenentatge sempre serà menor en la programació en Android.

### 3.5.1 Arquitectura de l'aplicació

El primer que s'ha de decidir quan es planteja la portabilitat d'una aplicació web a Android és si es desenvoluparà una aplicació nativa o bé ens limitarem a adaptar l'aplicació web a les dimensions i característiques dels dispositius mòbils (el que es coneix com a WebApp). Ambdues solucions son perfectament acceptables i cadascuna té els seus punts forts cosa que fa que el triar-ne una o altre depengui dels requeriments concrets que es tinguin per l'aplicació. Això fa que aquesta decisió sigui sempre molt "renyida" (Figura 20).

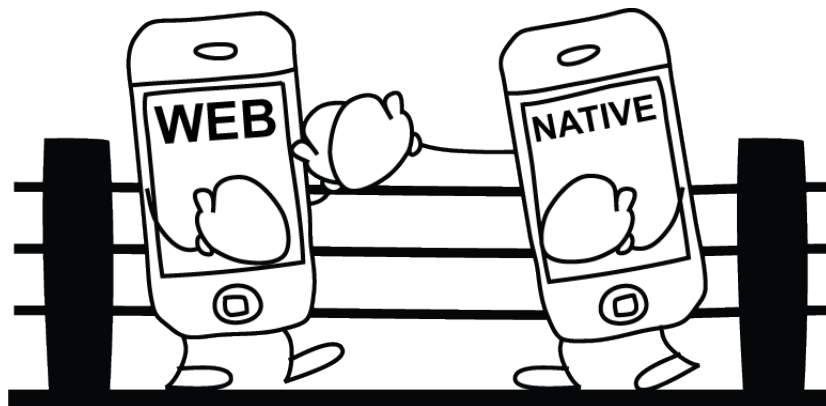


Figura 20. Web adaptada contra aplicació nativa.

Per tal de prendre aquesta decisió és necessari avaluar els pros i contres de cadascuna de les opcions i veure quina s'adapta millor a les necessitats del nostre projecte:

<b>Característica</b>	<b>Aplicació nativa</b>	<b>Web adaptada (webApp)</b>
Accés a Internet	No és necessari	Imprescindible
Instal·lació i actualització	És necessari instal·lar en el dispositiu	Immediata
Interfície d'usuari	Responsiva i funcional. És possible explotar al màxim les eines proporcionades per la plataforma (widgets)	Depenem del navegador, pel que el control és menor. No disposem de les funcionalitats que les aplicacions natives ofereixen a través dels widgets
Compatibilitat entre dispositius	El codi s'ha de programar tenint en compte els diferents dispositius	El CSS permet fàcilment la compatibilitat entre dispositius
animacions i gràfiques	Ràpid i àgil	més costós
Desenvolupament	És necessari utilitzar les eines de desenvolupament determinades	Es poden triar diverses alternatives

Pel que podem comprovar, la web adaptada al mòbil sembla a primera vista la solució més fàcil. A més, alguna de les avantatges de les aplicacions natives com son el funcionament sense connexió a Internet no les podem aprofitar en el nostre cas degut als requeriments particulars del nostre projecte.

Tot i així, la WebApp presenta unes limitacions en el disseny de les interfícies d'usuari que suposen un alt risc de fer que la solució sigui insuficient en algun punt del projecte, fet que posaria en greu perill l'èxit del mateix. Per aquest motiu optarem per l'aplicació nativa ja que, tot i que suposarà més cost de desenvolupament, estem segurs que ens oferirà les eines necessàries per garantir l'èxit del projecte.

### **3.5.2 Característiques de les aplicacions per a mòbil**

Al desenvolupar aplicacions per dispositius mòbils s'ha de tenir en compte que hi ha diferències significatives respecte a les aplicacions web tradicionals. Aquestes diferències s'han de tenir en ment en tot moment i especialment durant la fase de disseny.

La primera diferència i la més evident és que la pantalla d'un mòbil és molt més reduïda que la pantalla d'un ordinador, normalment una pantalla de mòbil mesura de 3 a 3,5 polzades mentre que les pantalles més petites de portàtil

acostumen a ser de 11 a 13 polzades. Això fa que a l'hora de desenvolupar per mòbil s'ha de tenir molt en compte el fet de mostrar a l'usuari només la informació més rellevant i mai mostrar per pantalla informació redundant.

El problema de la pantalla reduïda es compensa en els mòbils gràcies al major nivell d'interactivitat amb l'usuari. Això vol dir que mentre que en una web l'usuari pot només interactuar de dues maneres (tres comptant amb el teclat) amb els elements que es mostren (botó dret i botó esquerre del ratolí), en un mòbil la pantalla tàctil permet capturar molts altres tipus d'accions de l'usuari, per exemple diferenciar un clic curt d'un de llarg, fer zoom separant i ajuntant els dits, detectar canvis d'orientació i moltes altres accions (Figura 21).

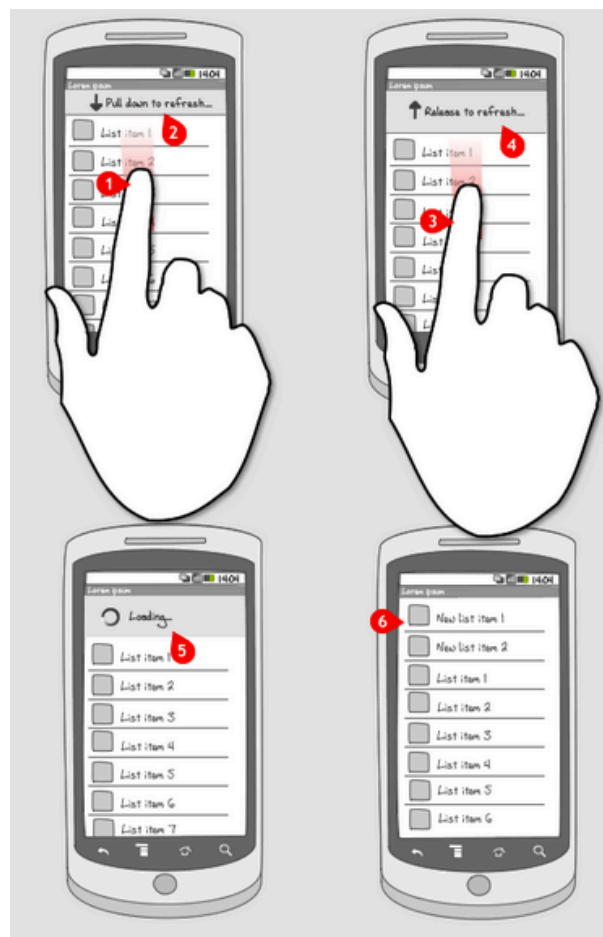


Figura 21. Exemple d'acció per actualitzar el contingut d'una llista en un dispositiu mòbil.



Per finalitzar, tot i que els dispositius mòbils permeten detectar més detalls en les interaccions amb l'usuari també s'ha de tenir en compte que no s'ha d'abusar d'aquestes funcionalitats. El motiu és que si bé podem suposar que tots els usuaris coneixen les accions bàsiques realitzables en una web (barres de desplaçament, botons, *tooltips* o enllaços) no es així pel que fa al funcionament de les aplicacions mòbils les quals son relativament recents. Per aquest motiu dotar una aplicació de masses funcionalitats complexes o desconegudes pot ser contraproduent ja que pot arribar a confondre a l'usuari i acabar per des incentivar-ne l'ús



## 4 DISSENY

### 4.1 INTRODUCCIÓ

En aquest capítol de la memòria recollim totes les decisions tècniques de disseny de l'aplicació. Definirem amb detall l'arquitectura de l'aplicació a partir de l'anàlisi realitzat en l'apartat anterior i també les capes internes que determinem i triarem els recursos tecnològics concrets per a abordar les tasques que ens proposem.

### 4.2 DEFINICIÓ DE L'ARQUITECTURA

L'aplicació que desenvolupem ha d'executar-se en el dispositiu mòbil i ha de ser capaç de connectar amb el servidor de la UAB on està allotjada l'aplicació actual a través d'Internet per tal de consultar i també de modificar informació de la Base de Dades d'aquesta.

Tal i com hem pogut comprovar analitzant l'aplicació existent, existeixen tres capes diferenciades a partir de les quals integrar-se amb el servidor.

#### *Capa d'interfícies*

Aquesta capa està composta per una sèrie de *servlets* que al invocar-los retornen codi HTML que forma les pàgines web que els usuaris veuen.

Integrar-se amb aquesta capa té avantatges considerables, en primer lloc podem suposar que hi han implementats tots els *servlets* necessaris per a interactuar amb les dades que volem simular i per tant suposem que no hauréem de crear codi nou a la banda del servidor, aconseguint així dos objectius: simplificar el desenvolupament i fer que la part del servidor sigui totalment independent de l'aplicació Android, aspecte molt rellevant a l'hora de fer noves versions (Figura 22).

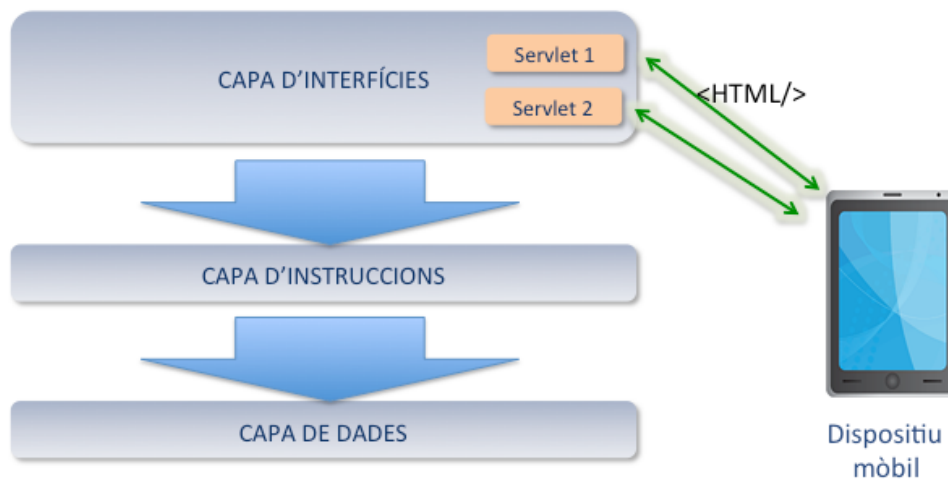


Figura 22. Interacció directa amb la capa d'interfícies de l'aplicació web

Per desgràcia el fet que aquesta capa només retorni respostes en forma de codi HTML fa molt difícil aprofitar-ne els resultats ja que per poder fer ús de la informació hauríem d'extreure-les de tot l'embolcall de codi HTML i qualsevol modificació en aquest, fins i tot per qüestions estètiques, podria repercutir negativament en el funcionament de la nostra aplicació.

### *Capa d'Instruccions*

Aquesta classe està composta per un conjunt de classes Java que s'invoquen des de la capa d'interfícies i entre elles mateixes i que contenen tots els mètodes i funcions per interactuar sobre les dades de la capa de dades.

Integrar-se amb aquesta capa requereix implementar noves interfícies en la capa superior per tal de generar les respostes en un format còmodament llegible per l'aplicació Android com pot ser XML o JSON (Figura 23) però a la vegada fa que la integració sigui immune a la majoria de canvis realitzats en el servidor sempre i quan es mantinguin les classes i mètodes existents entre versions i sempre se n'afegeixin de nous.

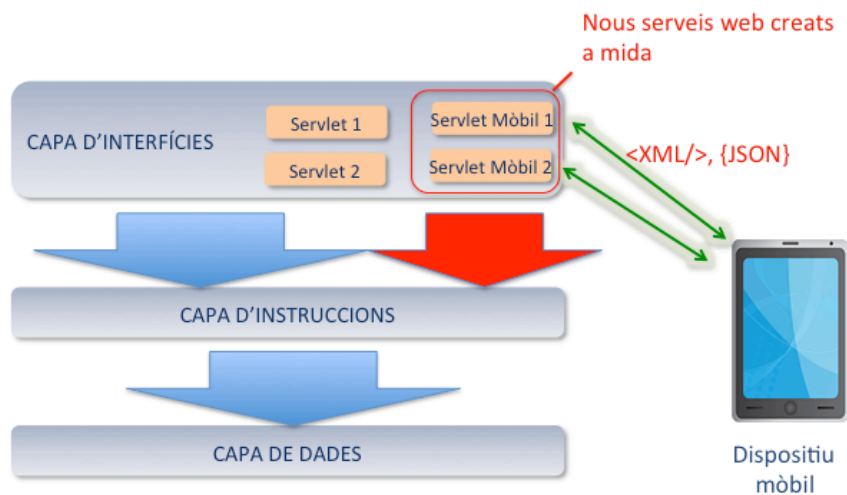


Figura 23. Creació d'interfícies pròpies per interactuar amb el servidor.

### Capa de dades

No ens plantegem ni tan sols integrar-nos directament amb la base de dades ja que per fer això hauríem de replicar el codi de les dues capes superiors, fet que sembla totalment innecessari tenint una capa d'instruccions ben definida.

Pels motius exposats, decidim que crearem un conjunt d'interfícies a mode de *servlets* dedicats exclusivament a l'intercanvi de dades amb l'aplicació Android, permetent-nos així tenir complerta autonomia pel que fa a l'intercanvi de dades. Presa la decisió cal determinar quina tecnologia concreta fem servir per a l'intercanvi de dades.

Un cop decidit com interactuarem amb les dades del servidor cal definir l'estructura de l'aplicació pròpiament dita. Les aplicacions Android es construeixen sobre quatre tipus de components bàsics, cadascun amb les seves pròpies característiques i cicles de vida (<http://developer.android.com/guide/topics/fundamentals.html>):

- **Activitats:** una activitat representa una única pantalla amb Interfície d'usuari. Les activitats en una aplicació Android van executant-se segons les accions realitzades per construir l'experiència d'usuari. Son el punt d'entrada d'informació de l'usuari i també la capa de presentació de la sortida de dades.

- **Serveis:** son components que s'executen en *background* i que no proporcionen interfície d'usuari. S'acostumen a utilitzar per a tasques d'intercanvi de dades per la xarxa i l'objectiu dels mateixos és realitzar tasques pesades de forma asíncrona per tal de no afectar a la interacció de l'usuari amb l'aplicació (activitats).
- **Proveïdors de Contingut:** son els components que permeten interactuar amb l'aplicació amb els conjunts de dades persistents que l'aplicació faci servir. Aquestes dades persistents poden trobar-se en forma d'arxius o bases de dades com SQL i el proveïdor de contingut el que proporciona és una forma d'interactuar amb aquestes.
- **Broadcast receivers:** aquests components s'utilitzen per escoltar esdeveniments procedents de fora de l'aplicació. En el projecte actual no farem servir cap component d'aquest tipus però permeten el tipus de funcionalitat que farà que l'aplicació, en futures versions, rebre avisos i actualitzacions de l'estat de la granja directament del servidor en temps real.

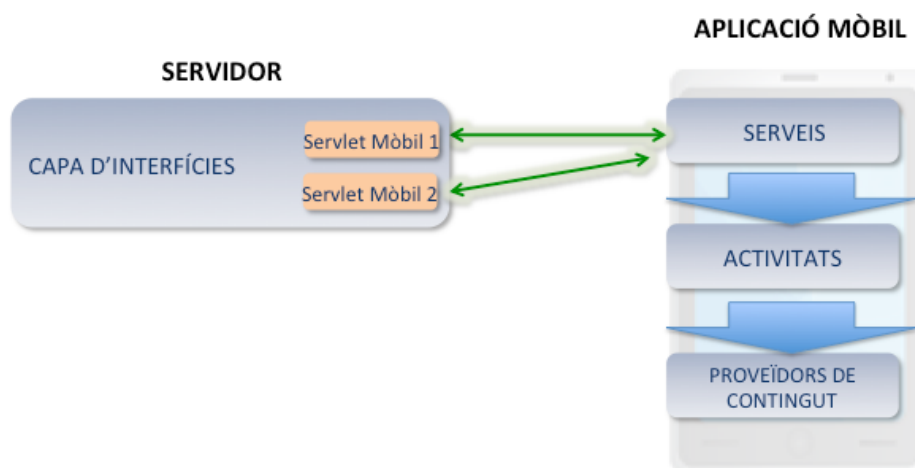


Figura 24. Components de l'aplicació Android.

### 4.3 CAPA DE SERVIDOR

En l'apartat anterior hem decidit que afegirem al servidor una sèrie d'interfícies fetes a mida per tal de gestionar l'intercanvi de dades entre l'aplicació Android i el servidor, per tant haurem d'implementar els serveis web corresponents.

A aquest efecte s'ha realitzat un estudi de les nombroses opcions existents i s'ha arribat a la conclusió que existeixen dos alternatives majoritàries que ens poden servir per als nostres objectius:

- SOAP (Simple Object Access Protocol): és el protocol més estès, defineix com dos sistemes s'han de comunicar mitjançant l'intercanvi de dades XML. Requereix de la creació d'un WSDL on es defineixin totes les accions possibles a realitzar entre els dos entorns.
- REST (Representational State Transfer): actualment és el protocol que està agafant més força, es basa en l'intercanvi de dades a través del protocol HTTP sense necessitat d'afegir una nova capa com en el cas de SOAP i on cada direcció única (URL) equival a una acció.

Per al nostre projecte, REST ofereix un avantatge addicional molt significatiu i és el fet que millora significativament el rendiment i disminueix el volum de transacció de dades respecte SOAP, la qual cosa a l'hora de desenvolupar per a dispositius mòbils on disposem de recursos de xarxa escassos adquireix especial importància.

Finalment i amb l'objectiu també de minimitzar el transit de dades realitzarem l'intercanvi de dades en format JSON (JavaScript Object Notation) en comptes de amb XML. En aquest cas la tria és senzilla ja que és una qüestió simplement de volum de dades.

#### *Jersey*

Jersey és la implementació de referència per construir serveis amb arquitectura REST en Java. A més, ofereix una API que facilita la definició i ús d'aquests

serveis. Per tant aquest és el recurs que utilitzarem per a implementar la part de codi al servidor que hem decidit que necessitem.

#### 4.1 CAPA D'ACTIVITATS

Les activitats són la base sobre la que es construeix tota aplicació en Android. Cada activitat representa una interfície d'usuari amb una sèrie de funcionalitats. Les activitats segueixen un cicle de vida preestablert i per a cadascuna s'ha de definir el comportament que ha de tenir al realitzar qualsevol de les accions possibles, tant si són accions pròpies de l'aplicació (com prémer un botó) com si són accions genèriques del dispositiu mòbil (com deixar l'aparell en repòs). El cicle de vida d'una activitat es descriu en la Figura 25.

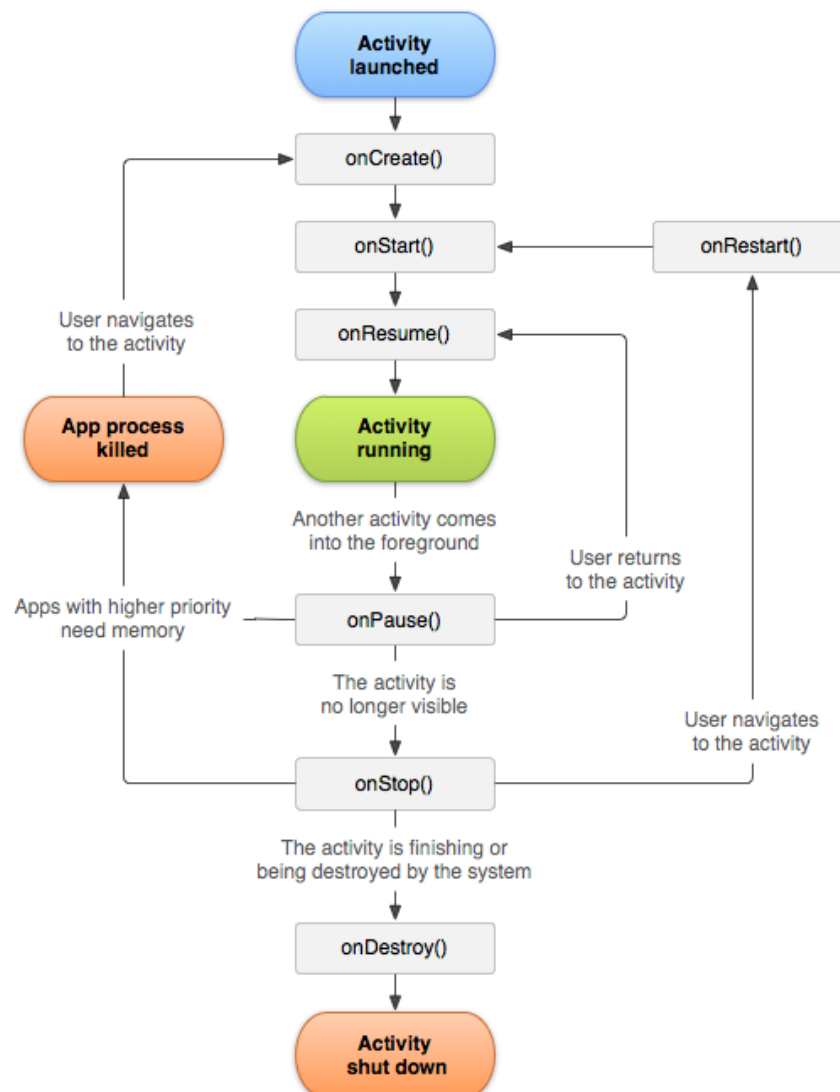


Figura 25. Cicle de vida d'una activitat.



Al crear una activitat nova s'han de definir, com a mínim, alguns dels mètodes descrits en la figura anterior. Aquests mètodes son cridats de forma automàtica quan es realitzen certes accions.

Per exemple *onCreate()* s'invoca quan es crea l'activitat per primer cop, mentre que si fem una acció que fa que un altre activitat es posi al davant de la actual, aquesta invoca el mètode *onPause()* el qual farà el que correspongui, per exemple guardar les dades del context. Si seguidament tanquem l'activitat que acabem d'obrir, la activitat inicial que s'havia pausat no torna a invocar el mètode *onCreate()* si no que invocarà *onResume()*.

Pel nostre projecte identifiquem inicialment les següents activitats que haurem d'implementar:

- **Control d'accés (GranjaSimActivity):** aquesta activitat permetrà a l'alumne identificar-se en l'aplicació per tal d'accedir a la seva granja. Un cop identificat, l'usuari accedirà a un menú principal des del que podrà navegar a les diferents funcionalitats.
- **Generar nova granja (GenerarGranjaActivity):** aquesta activitat serà utilitzada únicament el primer cop que l'usuari accedeix a l'aplicació ja que el que fa és utilitzar el servei de creació de nova granja. Un cop creada la navegació també portarà al menú principal de l'aplicació.
- **Menú Principal (MenuPrincipalActivity):** aquesta activitat serà el centre de l'aplicació i serà des de la que serà possible accedir a la resta de funcionalitats. S'ha pres la decisió d'utilitzar un menú principal en comptes d'una barra lateral permanent amb totes les opcions de menú per qüestions d'estalvi d'espai; en un dispositiu mòbil l'espai de pantalla disponible és generalment molt reduït i s'ha d'administrar correctament.

- **Indicadors (IndicadorActivity):** aquesta activitat contindrà la informació dels indicadors del conjunt de la granja i la gràfica de l'històric de producció.
- **Economia (EconomiaActivity):** aquesta activitat mostrarà els indicadors econòmics de la granja a l'usuari.
- **Llista de vaques (VaquesActivity):** aquesta activitat mostra el conjunt de vaques de la granja. En comptes d'una taula simple, amb les corresponents etiquetes de columna, cada vaca es mostrarà com un objecte de la llista amb tres files de dades (Figura 26) . Això és degut a les restriccions d'espai que ens trobem en el dispositiu mòbil que fa impossible posar una desena de columnes de dades simultàniament si volem mantenir la usabilitat. Des d'aquesta llista es podrà navegar al detall de cada vaca i també a l'activitat que permetrà ordenar i filtrar el llistat.

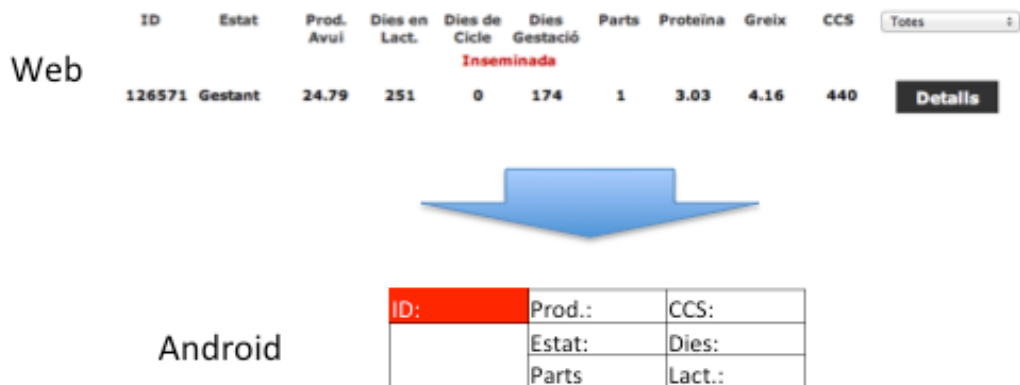


Figura 26. Disseny de l'element vaca a la nova llista pel dispositiu mòbil.

- **Detall d'una vaca (DetallVaquesActivity):** des de la llista de vaques podrem navegar al detall de cadascuna. En l'activitat de detall tindrem disponible la informació històrica de l'animal així com totes les accions disponibles per al mateix.
- **Filtratge de vaques (FiltrarVaquesActivity):** com que la llista de vaques no és una taula a l'ús, no és possible ordenar pels noms de les

columnes (ja que no n'hi ha). És per aquest motiu que habilem una activitat dedicada exclusivament a seleccionar els criteris d'ordenació i filtratge de la llista de vaques.

- **Llista de vedells, detall d'un vedell i filtratge de vedells (VedellsActivity, DetallVedellActivity, FiltrarVedellsActivity):** aquestes tres activitats tenen els mateixos objectius i funcionalitats que les corresponents a les vaques però adaptades a les particularitats dels vedells.
- **Dieta (DietaActivity):** aquesta activitat mostrarà els ingredients que formen la dieta de les vaques de la granja. El disseny d'aquesta activitat i la forma com s'ha resolt ha estat particularment complexa degut a que s'ha d'oferir molta informació de forma simultània a l'alumne, però a la mateixa vegada s'ha de permetre que aquest la ordeni de moltes i diverses maneres per tal de prendre la decisió de modificar el pes dels ingredients en un o altre sentit. El detall de la solució a la que s'ha arribat es descriu en la fase d'implementació però val a dir que aquest ha estat un dels punts més difícils d'aconseguir del projecte, no només a nivell tècnic si no sobretot a nivell funcional (Figura 27).

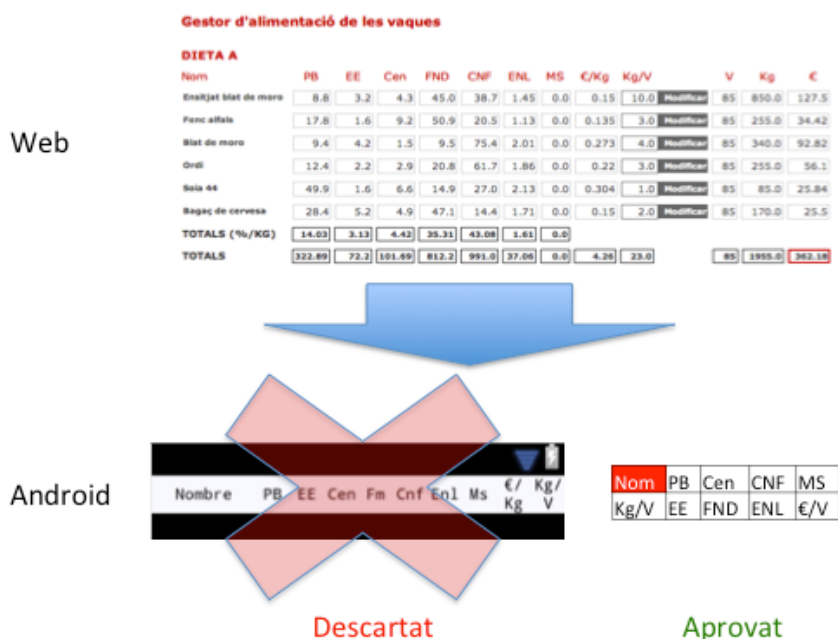


Figura 27. Propostes de disseny dels ingredients de la dieta.

- **Preferències (PreferencesActivity):** aquesta activitat serà la que permetrà veure i modificar les preferències emmagatzemades en l'aplicació. Hi hauran inicialment dues preferències per triar. En primer lloc aquest és l'apartat on es triarà l'idioma de l'aplicació (català, castellà o anglès inicialment). També podrem eliminar l'usuari guardat en memòria ja que, per defecte, quan un usuari s'autentica en l'aplicació es guarden les seves credencials per tal de no sol·licitar-li l'usuari i paraula de pas cada vegada que accedeix a l'aplicació.

Per tant i segons lo exposat en aquest apartat, el disseny de les activitats del projecte queda recollit segons el que es pot veure en el mapa de la Figura 28:

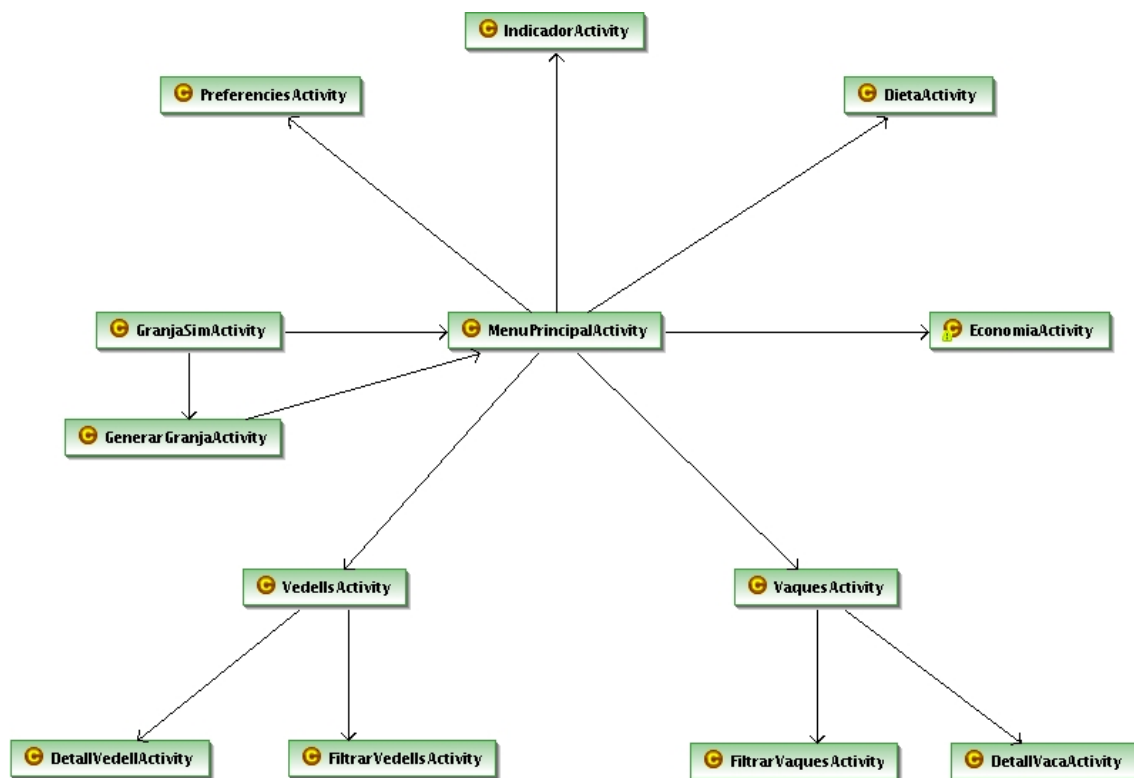


Figura 28. Mapa d'activitats de l'aplicació.

L'aspecte de cada activitat s'ha dissenyat a partir d'arxius XML, anomenats *layouts* en els projectes Android. Aquests *layouts* contenen una sèrie de *widgets* o elements interactius que permeten a l'usuari realitzar les accions desitjades. Aquests *widgets* poden ser des de simples elements com són textos

fixos fins a elements complexes com botons, les llistes desplegable o les vistes Web.

## 4.2 CAPA DE SERVEIS

Els components de la capa de serveis en una aplicació Android permeten l'execució de tasques costoses en background sense afectar a l'usuari. Aquests components s'acostumen a utilitzar per a descarregar informació de la xarxa ja que aquesta és una tasca el temps d'execució de la qual depèn de factors externs al desenvolupament i que per tant ni poden controlar-se ni han d'afectar a l'experiència de l'usuari.

Per al nostre projecte la capa de serveis s'encarregarà de la comunicació i de l'intercanvi de dades entre la nostra aplicació i el servidor mitjançant un conjunt de tasques asíncrones que s'invocaran al realitzar certes accions que requereixin informació del servidor des de la nostra aplicació.

Generalment cada tasca asíncrona invocarà un únic mètode del servidor, esperarà a rebre les dades, tractarà el resultat i el tornarà a l'activitat que l'hagi invocat per tal que aquesta s'actualitzi en conseqüència. En el nostre projecte hem les tasques asíncrones de la capa de serveis s'anomenen amb el sufix *Task* i son cridades per les activitats amb les que interactua l'usuari sempre que s'ha de sincronitzar informació amb el servidor.

Les tasques que crearem per cadascuna de les activitats son les següents:

### *Control d'accés*

**LoginTask:** verifica la identitat de l'usuari mitjançant l'usuari i paraula de pas contra la base de dades del servidor. A més verifica que l'usuari tingui perfil d'alumne per tal d'accedir a la simulació de la granja.

### *Generar nova granja*

**GeneraGranjaTask:** a partir de les dades d'usuari i la paraula de pas del curs sol·licita al servidor que es generi una nova granja.

### *Indicadors*

**CarregaGranjaTask:** carrega l'estat inicial i l'estat actual de la granja, pel que fa als indicadors genèrics de la mateixa com són el nivell de producció, nombre de vaques etc.

**ComprarTask:** crida a l'acció de comprar una nova vaca per a la granja i usuari en qüestió.

**SimularTask:** crida a l'acció de simular un nou dia a la granja en qüestió. Abans de sol·licitar la simulació valida que l'usuari disposi dels permisos corresponents per a realitzar aquesta acció.

**GraficaGranjaTask:** amb els paràmetres de la granja i a data de simulació actual descarrega la imatge de la gràfica de l'evolució de producció de llet de la granja.

### *Economia*

**EconomiaTask:** descarrega l'estat econòmic actual de la granja i els costos actuals de realitzar les accions per a la granja i el curs en qüestió.

**GraficaEconomiaTask:** descarrega la imatge de la gràfica d'evolució econòmica de la granja actual.

### *Llista de vaques*

**CarregaVaquesTask:** descarrega la llista de vaques ordenada i filtrada segons els paràmetres indicats en l'activitat.

### *Detall d'una vaca*

**ModificarVacaTask:** aquest servei modifica l'estat actual de la vaca i posa en cua les accions seleccionades per tal que s'executin amb la següent simulació.

**GraficaVacaTask:** descarrega la imatge de la gràfica corresponent a l'històric de producció de la vaca seleccionada pels darrers dies, que es passen com a paràmetre.

**HistoricVacaTask:** descarrega la taula de esdeveniments que ha patit la vaca des del seu naixement.

**AccionsVacaTask:** aquest servei retorna les accions que hi ha disponibles per a la vaca en qüestió per tal que aquestes opcions es mostrin disponibles en l'activitat.

#### *Llista de vedells*

**CarregaVedellsTask:** descarrega la llista de vedells ordenada i filtrada segons els paràmetres indicats en l'activitat.

#### *Detall d'un vedell*

**ModificarVedellTask:** aquest servei modifica l'estat actual del vedell i posa en cua les accions seleccionades per tal que s'executin amb la següent simulació.

**HistoricVedellTask:** descarrega la taula de successos que ha patit el vedell des del seu naixement.

**AccionsVedellTask:** aquest servei retorna les accions que hi ha disponibles per al vedell en qüestió per tal que aquestes opcions es mostrin en forma de botons d'acció en l'activitat.

#### *Dieta*

**CarregaDietaTask:** sol·licita per a la granja actual el valor dels ingredients disponibles i la distribució dels mateixos.

**ModificarDietaTask:** modifica el valor d'un ingredient de la dieta de la granja seleccionada.

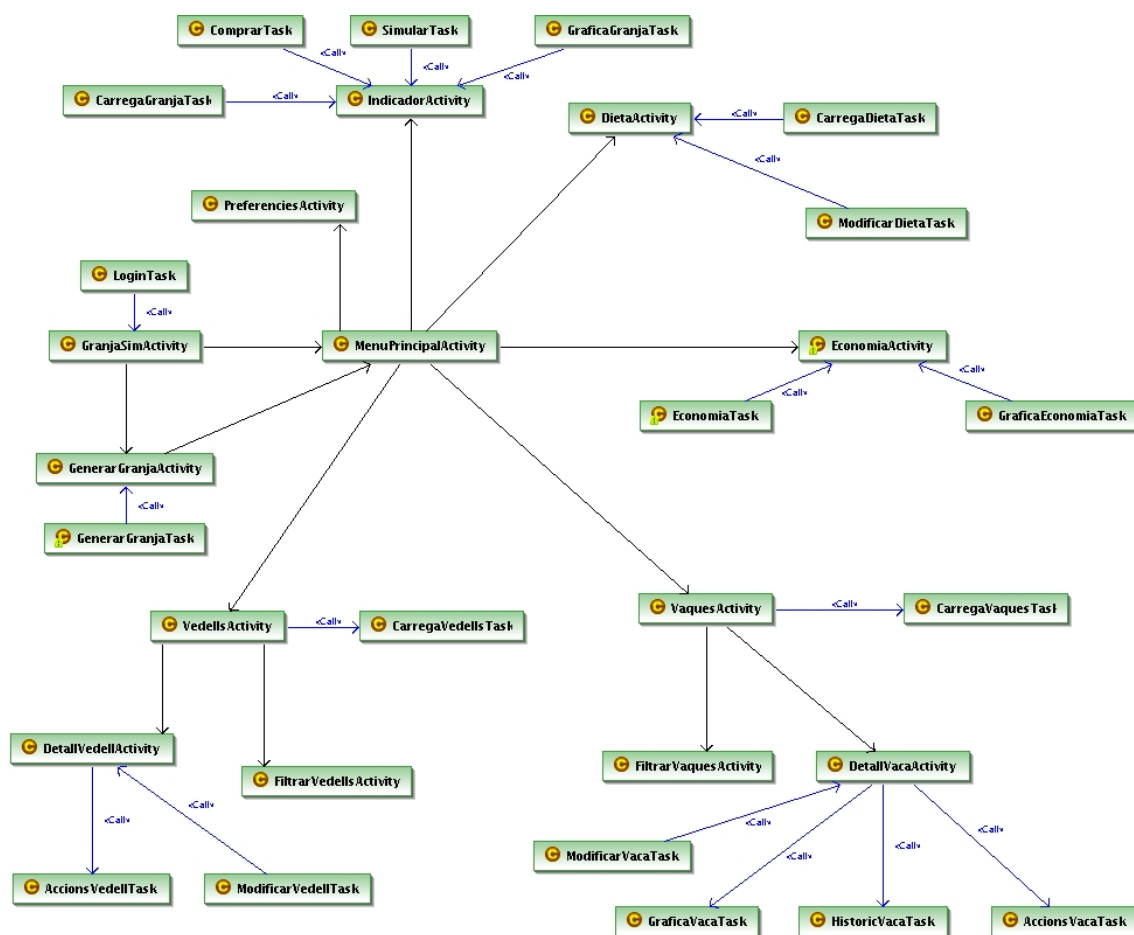


Figura 29. Tasques asíncrones associades a cada activitat.

### 4.3 CAPA DE DADES

Amb l'aproximació realitzada del projecte, només es guardaran com a dades persistents en l'aplicació les preferències personals de l'usuari com son el nom, cognoms i idioma preferit. La informació de la granja sempre es guardarà en servidor.

En un primer moment es va plantejar la possibilitat d'emmagatzemar també simulacions i les dades de la granja en el terminal mòbil però llavors les opcions de falsejar dades i de duplicar simulacions es disparaven. A més com que l'objectiu d'aquest projecte és traslladar l'aplicació al mòbil i no pas refer l'aplicació per tal que funcioni en client aquesta opció es va descartar.



Per emmagatzemar informació en una aplicació Android existeixen diverses possibilitats, des d'arxius a la memòria externa fins a bases de dades en SQL.

Ara bé, ja que l'ús que farem en el nostre projecte de l'emmagatzematge de dades és molt reduït optarem per l'opció més senzilla. Android disposa d'uns arxius predeterminats de preferències (SharedPreferences) que s'utilitzen específicament per guardar aquest tipus d'informació en forma de parells de dades del tipus identificador més valor. A més les activitats tenen una sèrie de mètodes dedicats a accedir, modificar i eliminar els continguts d'aquest arxiu. Les dades d'aquests arxius persisteixen al tancar l'aplicació i per tant són perfectes per els nostres propòsits.

#### 4.4 PLANIFICACIÓ ACTUALITZADA

Acabat el disseny disposem de dades més detallades respecte l'abast de les tasques que hem de realitzar. Per tant arribats a aquest punt és el moment de refer la planificació preliminar que es va fer durant l'estudi de viabilitat per tal de reflectir les desviacions corresponents. Aquestes desviacions afecten son degudes principalment a dos motius:

En primer lloc la fase de disseny ha requerit més temps de l'esperat degut a dificultats inesperades trobades al definir la solució de l'activitat de Dieta.

En segon lloc podem preveure que la fase d'implementació requerirà també més temps del previst inicialment degut sobretot al disseny que hem triat per les activitats de les llistes de vaques i vedells que farà que aquestes activitats siguin molt atractives a nivell funcional i ens permetrà reduir el volum d'intercanvi de dades però a canvi de més complexitat en el desenvolupament.

Pels motius exposats un els paràgrafs anteriors el temps inicial previst per l'execució de les tasques del projecte ha incrementat de en 40 hores fins a les 245 hores tal i com es descriu en la Figura 30.

Id	Nombre de tarea	Duración	Comienzo	Fin
1	<b>PROJECTE</b>	<b>46,75 días</b>	<b>lun 17/10/11</b>	<b>lun 21/05/12</b>
2	Assignació i matriculació del projecte	1 hora	lun 17/10/11	lun 17/10/11
3	Reunió de presa de requeriments	2 horas	mié 09/11/11	mié 09/11/11
4	<b>Inici del projecte</b>	<b>1,88 días</b>	<b>lun 02/01/12</b>	<b>mar 10/01/12</b>
5	Anàlisi de l'aplicació actual	8 horas	lun 02/01/12	jue 05/01/12
6	Estudi de viabilitat	4 horas	sáb 07/01/12	lun 09/01/12
7	Aprovació de l'estudi de viabilitat	1 hora	mar 10/01/12	mar 10/01/12
8	<b>Anàlisi del projecte</b>	<b>2,25 días</b>	<b>mar 10/01/12</b>	<b>vie 20/01/12</b>
9	Recull de requeriments	4 horas	mar 10/01/12	jue 12/01/12
10	Definició de funcionalitats	8 horas	jue 12/01/12	mar 17/01/12
11	Definició d'entitats	6 horas	mar 17/01/12	vie 20/01/12
12	<b>Disseny de la solució</b>	<b>5,75 días</b>	<b>vie 20/01/12</b>	<b>jue 16/02/12</b>
13	Arquitectura de l'aplicació	8 horas	vie 20/01/12	mié 25/01/12
14	Definició de la capa de servidor	10 horas	mié 25/01/12	mar 31/01/12
15	Definició de les activitats de l'aplicació	18 horas	mar 31/01/12	vie 10/02/12
16	Definició dels serveis de comunicació de l'aplicació	10 horas	vie 10/02/12	jue 16/02/12
17	<b>Fase de desenvolupament</b>	<b>13,25 días</b>	<b>jue 16/02/12</b>	<b>mié 18/04/12</b>
18	Implementació de la capa de servidor	20 horas	jue 16/02/12	mar 28/02/12
19	Implementació des activitats de control d'accès i generació de granja	6 horas	mar 28/02/12	vie 02/03/12
20	Implementació de les activitats d'indicadors i economia	12 horas	vie 02/03/12	vie 09/03/12
21	Implementació de les activitats de llistats de vaques i vedells	28 horas	vie 09/03/12	lun 26/03/12
22	Implementació del detall de vaques i vedells	14 horas	lun 26/03/12	mar 03/04/12
23	Implementació de l'activitat de dieta	18 horas	mar 03/04/12	vie 13/04/12
24	Implementació de l'activitat de preferències	4 horas	vie 13/04/12	lun 16/04/12
25	Implementació del Menú principal i del menú d'opcions	4 horas	lun 16/04/12	mié 18/04/12
26	<b>Fase de proves de qualitat</b>	<b>1,88 días</b>	<b>mié 18/04/12</b>	<b>jue 26/04/12</b>
27	Definició del test plan	1 hora	mié 18/04/12	mié 18/04/12
28	Proves de comunicació	4 horas	jue 19/04/12	vie 20/04/12
29	Proves funcionals	8 horas	sáb 21/04/12	mié 25/04/12
30	Report dels resultats	2 horas	jue 26/04/12	jue 26/04/12
31	<b>Implantació</b>	<b>2 días</b>	<b>vie 27/04/12</b>	<b>sáb 05/05/12</b>
32	Instal·lació	4 horas	vie 27/04/12	sáb 28/04/12
33	Proves de comunicació reals	4 horas	lun 30/04/12	mar 01/05/12
34	Validació del projecte amb el client	8 horas	mié 02/05/12	sáb 05/05/12
35	<b>Final del projecte</b>	<b>3,25 días</b>	<b>lun 07/05/12</b>	<b>lun 21/05/12</b>
36	Composició de la memòria a partir de la documentació generada	20 horas	lun 07/05/12	jue 17/05/12
37	Preparació de la defensa del projecte	6 horas	vie 18/05/12	lun 21/05/12

Figura 30. Tasques del projecte.





## 5 IMPLEMENTACIÓ

### 5.1 INTRODUCCIÓ

En aquest apartat de la memòria es descriu quin és el resultat final d'aplicar els dissenys definits en el capítol anterior entrant en detall en aquells aspectes que per la particularitat que sigui ho mereixin. També s'exposarà el resultat del disseny final de les interfícies d'usuari i la forma en la que s'han resolt els requeriments.

En el projecte que ens ocupa haurem de fer desenvolupaments en dos entorns clarament diferenciats: el servidor on crearem els serveis d'intercanvi de dades a mida per la nostra aplicació, i l'aplicació nativa en Android que s'instal·larà al dispositiu mòbil i que es comunicarà amb el servidor i oferirà a l'usuari totes les funcionalitats descrites en la presa de requeriments.

### 5.2 SERVEIS WEB

L'aplicació web actual s'executa en un entorn format per un servidor d'aplicacions Apache Tomcat que publica tots els *servlets* escrits en Java del projecte. Nosaltres el que volem és afegir un conjunt de *servlets* a aquest entorn, però definits de tal manera que consumeixin i retornin informació codificada en JSON en comptes de codi html.

Com que hem finalment hem triat els serveis RESTful per satisfer les nostres necessitats, ens documentem per tal de trobar la forma més correcte de dur a terme la tasca.

#### *Jersey*

Per tal d'utilitzar aquest *framework* que és el que ens hem proposat fer servir el primer que cal fer després d'afegir les llibreries corresponents al projecte és

definir a l'arxiu de configuració de recursos web de l'aplicació (web.xml) la referència al contenidor de serveis que volem implementar:

```
<servlet>
<servlet-name>Jersey REST Service</servlet-name>
<servlet-
class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
<init-param>
<param-name>com.sun.jersey.config.property.packages</param-name>
<param-value>Simulador</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>Jersey REST Service</servlet-name>
<url-pattern>/rest/*</url-pattern>
</servlet-mapping>
```

Figura 32. Configuració del contenidor de servlets.

El codi de la figura anterior inicia el contenidor de *servlets* de Jersey al inicialitzar l'aplicació web. També estem indicant que en el paquet "Simulador" és on trobarà la classe Java que definirà els serveis disponibles. Definim aquest paquet ja que és el mateix que conté la resta de *servlets* de l'aplicació web i és on ubicarem la classe Java que hem d'escriure.

Finalment es defineix que totes les URLs que compleixin el patró "/rest/\*" es redirigiran a aquest contenidor per ser ateses.

A continuació hem creat una classe que conté tots els mètodes necessaris pel nostre projecte (en el paquet indicat anteriorment). Per accedir a la classe i els mètodes en qüestió fem ús de les anotacions de "Path" que serveixen al contenidor de *servlets* per orientar-se i determinar quin mètode ha de respondre la crida. Per il·lustrar aquest fet mostrem la següent part de codi (Figura 33):

```

@Path("/android")
public class AndroidWS {

    @Path("/Login")
    @POST
    @Consumes(MediaType.APPLICATION_JSON)
    public JSONArray Login(JSONObject json) throws JSONException {

        JSONArray ret=new JSONArray();

        String user="";

        String pass="";
        JSONObject result= new JSONObject();

        user = json.get("username").toString();

        pass = json.get("pwd").toString();

        String lang;

        if(json.has("lang")){
            lang=json.getString("lang");
        }else{
            lang="ca";
        }

        ...
    }
    @Path("/GenerarGranja")
    @POST
    @Consumes(MediaType.APPLICATION_JSON)
    public JSONArray GenerarGranja(JSONObject json) throws JSONException
    {

```

Figura 33. Exemple de mètode.

Creant aquesta classe `AndroidWS.class` el que aconseguim és disposar d'un *servlet* per cada mètode que definim a la classe. Per a accedir a cada mètode només caldrà fer una crida a la URL definida en l'anotació `@Path` corresponent i el contenidor de *servlets* s'encarregarà de triar i decidir quina classe i mètode es farà servir.

### *@Path*

Per exemple, fent POST a la URL "http://urlDelProjecte/rest/android/Login" es cridarà al mètode `AdnroidWS.Login` que retornarà una resposta que s'ha

d'interpretar com a JSONArray, mentre que si es fa POST a "http://urlDelProjecte/rest/android/GenerarGranja" el que s'aconseguirà és fer una crida a AndroidWS.GenerarGranja.

*@POST, @GET, @DELETE, @UPDATE*

Cal notar que, donat que els serveis REST aprofiten els mateixos mètodes del protocol HTTP, mitjançant les anotacions corresponents també podem definir quins són els mètodes suportats pels serveis que definim. En el cas de l'exemple exposat només està permès utilitzar els mètodes a partir d'una crida POST. Aquesta característica serà comú per tots els mètodes que hem definit en el nostre projecte ja que restringir les crides possibles a aquest mètode ens garanteixen un major nivell de seguretat.

*@Consumes*

Finalment les anotacions també ens permeten determinar el *MIME type* que espera rebre en la petició realitzada el mètode en qüestió.

### 5.3 APLICACIÓ ANDROID

Havent realitzat totes aquestes accions en el servidor ja disposem d'una forma d'accedir des d'on vulguem a la capa d'instruccions del servidor i tenim la possibilitat d'extreure'n la informació i realitzar les accions que desitgem amb gran comoditat i autonomia.

A més, com que el que hem fet és crear una nova classe Java dedicada exclusivament a les nostres necessitats la nostra aplicació serà molt robusta davant canvis i actualitzacions en la part del servidor tret d'aquelles en la que desaparegui o es modifiqui una funcionalitat molt bàsica de l'aplicació, com podria ser que desaparegui una entitat o un mètode o atribut. Ara bé, com que les actualitzacions normalment són acumulatives, és a dir, es creen funcionalitats i no es destrueixen, creiem que podem tenir la seguretat que aquest cas no es donarà.



A continuació es descriuen els detalls d'implementació de les activitats principals del projecte:

- Control d'accés i nova granja
- Menú principal
- Indicadors
- Economia
- Llista de vaques
- Detall d'una vaca
- Filtratge de vaques
- Llista de vedells
- Detall d'un vedell
- Filtratge de vedells
- Dieta
- Preferències
- Menú d'opcions

### 5.3.1 Control d'accés i nova granja

Les interfícies de control d'accés a l'aplicació fan de punt d'entrada a la resta d'activitats i serveixen per mantenir la seguretat de les dades a través de les credencials dels usuaris. La interfície de la generació de nova granja (Figura 34, dreta) és un cas particular del control d'accés (Figura 34, esquerra) que només s'utilitza una vegada per alumne i curs ja que fa que el servidor creï una nova granja per a l'usuari en qüestió.

Aquesta activitat s'encarrega de validar tant que les dades introduïdes pertanyen a un usuari de l'assignatura com que l'usuari en qüestió té el perfil d'alumne i una granja assignada, característiques necessàries per accedir a les funcionalitats de l'aplicació.

Per comoditat per l'usuari un cop validades les credencials amb els continguts de la base de dades l'aplicació guarda els valors en l'arxiu de preferències i els

utilitza quan cal. Això és cert també després de que l'aplicació es tanqui per qualsevol motiu, per tant l'usuari no haurà d'introduir l'usuari i la paraula de pas cada vegada que accedeixi a l'aplicació. Aquest comportament (el fet de guardar les credencials) que no és aconsellable realitzar lo entorns d'aplicacions web per raons de seguretat és molt freqüent en aplicacions per a mòbil i suposa una millora per a l'usuari.



Figura 34. Generar nova granja i validació d'accés.

### 5.3.2 Menú Principal

L'aplicació web que volem traslladar a Android fa ús d'un menú lateral per navegar entre les funcionalitats més destacades. Aquest disseny és molt útil pel que fa a permetre a l'usuari en tot moment navegar allà on vulgui de forma gairebé immediata, però aquest és un recurs que només es pot utilitzar quan disposem d'espai sobrant en la interfície.

En les aplicacions per a mòbil l'espai disponible a la pantalla és un dels recursos més escassos. Per això no és aconsellable definir en menús fixes que ocupin espai de forma constant tant si s'utilitzen com si no, així que per l'aplicació que hem desenvolupat hem pres la decisió de crear una pantalla específica de menú principal que serà el centre de la aplicació i a partir de la qual navegarem a les demés funcionalitats concretes.

Donat que disposem d'aquesta pàgina principal també hem decidit que, els avisos de fets derivats de la simulació (parts de les vaques o naixements de vedells) també es mostraran en aquesta interfície. Al ser aquest el punt central de l'aplicació també hem tingut en compte que quan es realitza una simulació l'aplicació redirigirà a l'alumne a aquesta activitat estigués on estigués en el moment de simular.

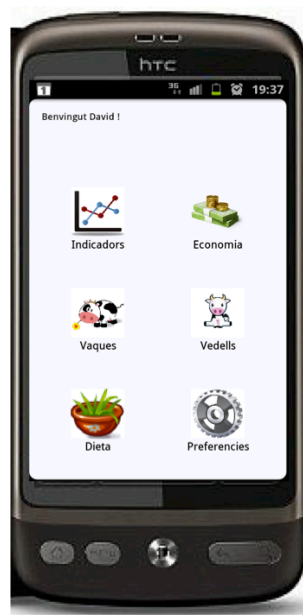


Figura 35. Menú principal de l'aplicació.

### 5.3.3 Indicadors

La activitat implementada pels indicadors de l'estat global de la granja té com a objectiu mostrar a l'alumne la informació de manera que pugui veure les dades de forma còmode per a prendre les decisions pertinents. Ja que la informació a mostrar és considerable s'ha pres la decisió de separar-la en diferents pestanyes mitjançant una funcionalitat nativa d'Android anomenada TabHost. Aquesta classe permet definir diferents vistes d'informació per una mateixa pantalla i portar el control de quin dels conjunts d'informació es mostra a través de widgets en forma de pestanya (Figura 36).

D'aquesta manera pel disseny d'aquesta activitat s'ha dividit la informació en tres pestanyes diferents:

- **Vaques:** en aquest apartat es mostra una llista de valors relacionats amb el conjunt de vaques de la granja en forma de taula amb tres columnes, la primera amb la descripció del valor de la fila. La segona amb el valor en el moment d'iniciar la simulació i la darrera amb el valor actual de l'indicador que es mesura.
- **Vedells:** es mostra la llista de valors relatius a les accions realitzades amb els vedells de la granja en el mateix format que en la pestanya anterior.
- **Gràfica:** es mostra la gràfica amb la evolució temporal dels nivells de producció de la granja. Aquesta gràfica es considerablement gran i per tant havíem de permetre que l'usuari pugues interactuar amb la mida de la mateix. La solució per a aquest problema ha estat mostrar-la a l'alumne dins d'un component anomenat WebView. Aquest component permet mostrar contingut, en el nostre cas una imatge, com si s'estigués mostrant en un navegador de dispositiu mòbil. Això vol dir que el contingut té integrades les funcions que es permeten en un navegador,

com és la opció de fer Zoom amb dos dits, separant-los o ajuntant-los sobre la pantalla.



Figura 36. Pantalla d'indicadors de la granja, pestanyes de vaques i gràfica de producció.

### 5.3.4 Economia

Aquesta activitat té un objectiu molt similar a l'anterior, és a dir mostra de forma general les dades associades a la producció i als costos de realitzar les diferents accions que l'alumne té a la seva disposició. Per aquest motiu i amb l'objectiu també de mantenir la unitat en el disseny de l'aplicació s'ha optat per una solució similar a la de l'apartat d'indicadors, dividint la interfície en dos apartats en aquest cas per tal de mostrar la informació d'una manera més clara i accessible (Figura 37):

- **Economia:** mostrem les dades econòmiques en una taula amb dues columnes.
- **Gràfica:** mostra la imatge de la gràfica de l'evolució de la relació de costos i ingressos de la granja. Igual que en el cas anterior fem ús del tipus de vista WebView per tal de permetre que l'usuari faci Zoom sobre la imatge de forma automàtica i sense necessitat d'escriure codi per tal d'aconseguir aquesta funcionalitat.

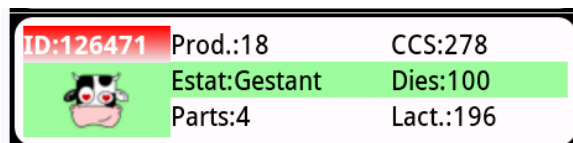


Figura 37. Pantalles de dades econòmiques.

### 5.3.5 Llista de vaques

La implementació d'aquesta funcionalitat ja resultat força complexa degut a diversos factors. En primer lloc, no ha estat possible emular el disseny actual de l'aplicació web al que els alumnes ja estan acostumats ja que fa ús d'una desena de columnes per mostrar la informació de cada vaca.

Tal i com ja hem dit varies vegades molts cops a l'hora de dissenyar una aplicació per a mòbil cal replantejar-se la presentació de les dades ja que moltes vegades les formes típiques (taules, llistes) no valen a causa de l'escàs espai del que es disposa. Aquest és un d'aquests casos (Figura 38):




ID:126471	Prod.:18	CCS:278
	Estat:Gestant	Dies:100
	Parts:4	Lact.:196

Figura 38. Detall d'una vaca en la llista de vaques.

S'ha redistribuït la informació de la vaca en tres files per crear un element que mostra la informació rellevant d'una manera clara i que ara sí pot utilitzar-se en una pantalla reduïda. També s'ha volgut fer més visual el reconeixement de l'estat de la vaca afegint a les diferents imatges segons l'estat, un color de fons que ajudi al reconeixement.

També s'ha utilitzat aquest color per relacionar visualment dues dades, l'estat i els dies. Això últim s'ha fet ja que en l'aplicació web sempre he ha tres columnes de dies (dies de gestació, dies inseminades i dies de cicle) però una vaca en un moment donat només pot tenir un estat i, per tant, només pot tenir un valor diferent de 0 en una d'aquestes tres columnes. Això implica que per cada vaca sempre hi ha dos columnes que no aporten informació fet que si bé a l'aplicació web no dona massa problema no ens podem permetre.

Aquesta decisió presa durant la fase de disseny fa que perdem les capçaleres de les columnes que tindríem si haguéssim pogut optar per un format de taula,

capçaleres que podríem haver utilitzat per a la ordenació dels elements de la llista. De tota manera, com que per filtrar els elements de la llista ja havíem de crear una interfície per al usuari hem decidit crear una nova activitat amb les opcions de filtratge i ordenació dels elements a la que accedirem mitjançant el botó "Filtrar".

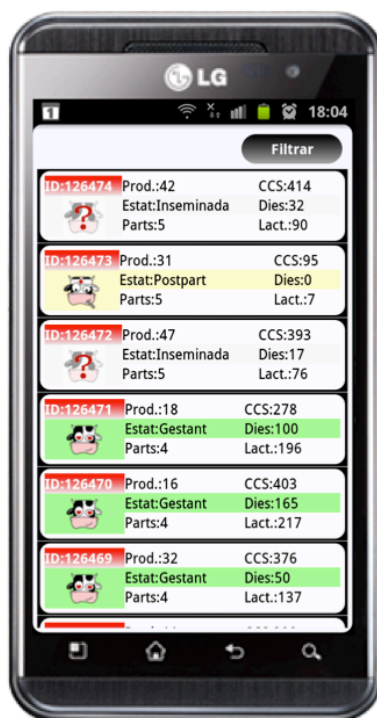


Figura 39. Pantalla completa de la llista de vaques.

Un cop decidit com mostrar les dades ha calgut determinar com fer la gestió de l'intercanvi de dades. La primera proposta i que sembla la més senzilla és descarregar totes les vaques del servidor i muntar amb elles una llista desplaçable per que l'alumne hi navegui.

Aquesta primera aproximació té dos problemes: en primer lloc, descarregar totes les vaques de la granja d'Internet amb una connexió 3G (en els casos en que no hi hagi disponible Wi-Fi) previsiblement trigarà un cert temps. Això fa empitjorar l'experiència d'usuari i dona la sensació de que l'aplicació és lenta. En segon lloc, normalment quan un alumne accedeix a la llista de les vaques no té intenció d'interactuar amb totes elles, si no que el més usual és que només li interessin en aquell moment un grup molt concret de les mateixes. Això fa que si sempre descarreguem totes les dades de la granja estem



transmetent una gran quantitat de dades innecessàries quan el transit de dades ja hem indicat que és un recurs escàs.

Per tots aquests motius hem triat una aproximació molt estesa en les aplicacions per a mòbil que consisteix en anar construint els elements de la llista segons l'usuari s'hi va desplaçant. D'aquesta manera, la primera vegada que l'usuari accedeix a l'activitat, el servei corresponent s'encarrega de demanar els primers elements de la llista (10 en el nostre cas) segons la ordenació que li indiquem per paràmetre i els mostra per pantalla. D'aquesta manera aconseguim una resposta més ràpida, ja que només hem de mostrar 10 elements al carregar la activitat i, per altra banda, el transit de dades és molt inferior.

A continuació, si l'usuari es desplaça fins al final de la llista, es torna a cridar el mateix servei que abans demanant els 10 elements següents. Això es pot aconseguir gràcies a l'ús d'un tipus específic d'activitat anomenat `ListViewActivity`. Aquest tipus específic d'activitat està pensat per aquelles pantalles que contenen llistes de valors que no caben en la mida d'una pantalla i permet coses com detectar quins elements son visibles i capturar fàcilment esdeveniments tals com detectar que l'usuari ha desplaçat la barra de desplaçament fins al final de la pantalla entre altres.

### 5.3.6 Detall d'una vaca

Quan l'alumne selecciona qualsevol dels elements de l'activitat llista de vaques, accedeix a una nova activitat amb el detall de la mateixa. En aquesta activitat de detall hi han tres apartats diferenciats:

- **Dades de la vaca:** per evitar perdre el context de la vaca amb la que s'està tractant, a la part superior de la pantalla es mostra la informació rellevant ampliada de la vaca.
- **Historial:** ocupa la part central de la pantalla i conté la informació passada de la mateixa en dos formats entre els quals l'usuari pot navegar a partir de dues pestanyes:
  - **Gràfica:** imatge de la gràfica de nivells de producció de llet de la vaca fins al dia actual. Per defecte mostra els últims 365 dies però permet modificar aquest valor mitjançant un camp d'entrada de text i el botó "Refrescar". Aquesta gràfica igual que la resta de l'aplicació s'exposa en un WebView i per tant permet a l'usuari fer Zoom.
  - **Històric d'esdeveniments:** es tracta d'una taula en la que es llisten els esdeveniments més importants de la vaca en qüestió i la data en la que van succeir.
- **Accions:** la part inferior de la pantalla està reservada a les diferents accions que es poden realitzar amb la vaca. Aquestes accions depenen de l'estat actual de la vaca i normalment no tenen un efecte immediat, en canvi són accions que queden en cua fins a la següent simulació. Cada vegada que es selecciona una d'aquestes accions s'invoca un servei que modifica l'estat de la vaca al servidor i, seguidament es crida un altre servei que torna a descarregar la llista d'accions disponibles de la vaca.

Així les accions disponibles per a la vaca estan sempre sincronitzades amb el que l'usuari veu per pantalla.



Figura 40. Informació detallada d'una vaca al seleccionar-la a la llista de vaques.

### 5.3.7 Filtratge de vaques

Aquesta interfície que es crida des de la llista de vaques té com a objectiu permetre a l'usuari triar la ordenació i els paràmetres de filtre que fa servir el servei de descàrrega de la llista de vaques per tal de fer més còmode el fet de navegar entre totes les vaques de la granja.

Es permet a l'usuari, mitjançant un desplegable, seleccionar el criteri d'ordenació a partir de qualsevol de les dades de la vaca que es mostren en la llista (identificador, nombre de parts o producció entre altres) i també es permet triar la direcció, ascendent o descendent, dels resultats.

A més també es permet indicar que només es descarreguin a la llista grups concrets de vaques, segons la dieta que tenim assignada a les mateixes o l'estat. Aquesta funcionalitat permet un major control sobre les vaques que apareixen a la llista i per tant fa més fàcil la gestió i la navegació a l'alumne fent que hagi d'interactuar amb menys dades a la vegada.



Figura 41. Opcions de filtratge i ordenació de la llista de vaques.

### 5.3.8 Llista de vedells

La llista de vedells s'ha resolt de la mateixa manera que la llista de vaques ja que les funcionalitats necessàries per ambdues activitats son les mateixes. Això a la seva vegada dota d'homogeneïtat i d'identitat a l'aplicació, fet que sempre suposa una bona decisió de disseny.



Figura 42. Llista de vedells

### 5.3.9 Detall d'un vedell

Al seleccionar qualsevol dels elements de la llista de vedells s'accedeix al detall del mateix. Aquesta activitat està dividida verticalment en tres parts igual que el detall de la vaca. Els apartats superior i inferior segueixen el mateix comportament que en l'activitat per a les vaques.

La part central de l'activitat, per altra banda, és significativament diferent degut que, segons l'estat del vedell en qüestió, la informació que se'n mostra és diferent. Això ha fet que a nivell d'implementació s'hagin dissenyat tres interfícies diferents i, en cada moment, s'utilitza la corresponent.



Figura 43. informació detallada d'un vedell al seleccionar-lo a la llista de vedells.

### 5.3.10 Filtratge de vedells

Aquesta activitat és idèntica pel que fa a funcionalitats a la de filtratge de vaques, i per tant s'han adaptat les opcions als atributs dels vedells mantenint la mateixa solució que en la activitat exposada anteriorment.

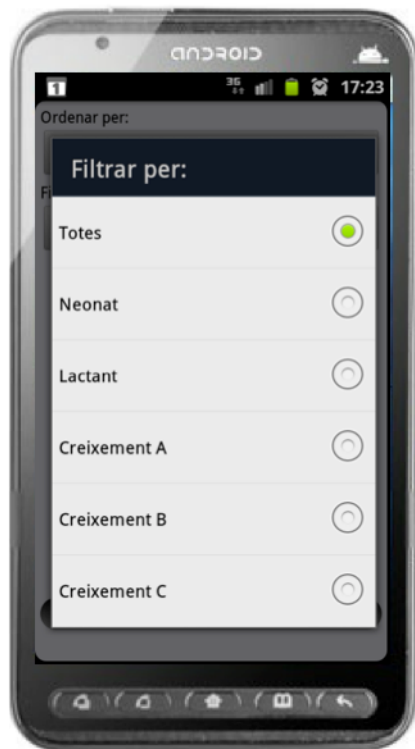


Figura 44. Opcions de filtratge i ordenació de la llista de vedells.

### 5.3.11 Dieta

La implementació d'aquesta activitat ha estat força complicada degut a les necessitats que tenen els alumnes a l'hora de determinar la quantitat de cada ingredient que ha de formar la dieta de les vaques. En primer lloc ens vam trobar amb el mateix problema que amb la llista de vaques i de vedells: no és possible mostrar en una pantalla de mòbil, en vertical, totes les columnes d'informació que hi ha a les dades.

Ara bé, no era suficient la solució adoptada en els casos anteriors ja que segons els requeriments indicats l'alumne havia de poder comparar de forma còmode els valors existents en una columna en concret i entre dues files diferents les quals, al representar ingredients de la dieta, estaven ordenats de forma arbitrària.

La primera part de la solució inevitablement passava per mostrar les dades de cada ingredient en dues files, quedant de la següent manera (Figura 45) després d'haver aplicat les decisions preses en el capítol de disseny:

Ordi	PB	Cen	CNF	MS
	12.4	2.9	61.7	0
<b>Kg/V</b>	EE	FND	ENL	€/Kg
5	2.2	20.8	1.86	0.22

Figura 45. Detall d'un ingredient de la dieta.

El següent pas era resoldre els dos problemes restants: l'alumne ha de poder compara qualsevol dels paràmetres d'un ingredient respecte el mateix paràmetre de la resta d'ingredients (aquests paràmetres corresponen a la composició de l'ingredient, per exemple PB és la quantitat de proteïna) i l'alumne ha de poder comparar dos ingredients a nivell general de forma fàcil i intuïtiva. Es va determinar fer dues coses per assolir aquest objectiu:

En primer lloc es va fer que tots els paràmetres de tots els ingredients es poguessin marcar de tal manera que, si seleccionem, per exemple el paràmetre



PB de l'ingredient Ordi es ressalten tots els paràmetres PB de la resta d'ingredients. D'aquesta manera a l'usuari li és més fàcil comparar els valors del paràmetre entre els diferents ingredients.

En segon lloc i ja que la ordenació que ha de poder fer dels ingredients l'alumne és totalment arbitrària es va implementar una funcionalitat *drag and drop* per tots els ingredients de forma que l'usuari pot arrossegar cada ingredient per la pantalla i ordenar-los manualment sense cap mena de restricció.

Arreglat el punt de la presentació de les dades s'ha creat un menú emergent per cadascun dels ingredients, que apareix quan es manté el dit sobre un ingredient dos segons, per permetre la funció de modificar la quantitat d'aquest ingredient per vaca que forma part de la dieta.



Figura 46. Visió general de la pantalla de dieta i el menú d'accions emergent.

Per acabar amb la implementació d'aquesta activitat s'afegeix a la part inferior de la pantalla l'apartat dels totals de la dieta, amb el mateix disseny que els

ingredients i que, a diferencia d'aquests està ancorat al peu de la pantalla i que permet veure a l'usuari el preu total de la dieta que té seleccionada i també la distribució de cada propietat de la dieta en percentatge per defecte i en valors totals si l'usuari prem durant dos segons aquest apartat de totals.

### 5.3.12 Preferències

Aquesta activitat conté les preferències d'usuari guardades. Inicialment només hi ha dues preferències per seleccionar. En primer lloc les credencials d'usuari que s'emmagatzemen automàticament la primera vegada que l'usuari passa pel control d'accés de forma satisfactòria i que permeten que l'usuari pugui executar l'aplicació sense necessitat d'introduir cada vegada el nom d'usuari i la paraula de pas.

En segon lloc és des d'aquesta activitat des de la que l'usuari selecciona l'idioma de l'aplicació. Inicialment hi ha tres idiomes disponibles tot i que el nombre és ampliable en un futur si es considera necessari. Això és així ja que és molt fàcil canviar els textos de l'aplicació degut a la forma en la que s'estructura una aplicació en Android. La forma de definir les etiquetes de l'aplicació (tots els textos estàtics que apareixen en l'aplicació) és mitjançant la referència a un arxiu que conté totes les cadenes de text associades, cadascuna a un identificador enter.

Aquestes cadenes de text es guarden en un arxiu anomenat *strings.xml* en la carpeta *res/values* del projecte. Per afegir un nou idioma només cal crear una nova carpeta, amb el sufix corresponent a l'idioma desitjat (per exemple per afegir l'idioma alemany a una aplicació cal crear la carpeta anomenada *res/values-de*) i afegir l'arxiu *strings.xml* amb les mateixes cadenes de text traduïdes a l'idioma desitjat.

Finalment aquesta activitat té l'acció de *Netejar* que elimina les preferències guardades (això tanca l'aplicació i fa que la següent vegada s'executa l'aplicació sí que demani usuari i paraula de pas).



Figura 47. Pantalla de preferències guardades.

### 5.3.13 Menú d'opcions

El menú d'opcions és una funcionalitat integrada en totes les aplicacions Android que s'executa quan es prem en el mòbil el botó *Menú* que és un botó comú a tots els dispositius que fan ús del sistema operatiu Android. Per cada aplicació i per cada activitat de l'aplicació es pot definir quines opcions es mostren en aquest menú. En la nostra aplicació hem definit tres opcions que es poden executar des de qualsevol activitat tret de les de filtratge de vaques i de vedells (Figura 48).

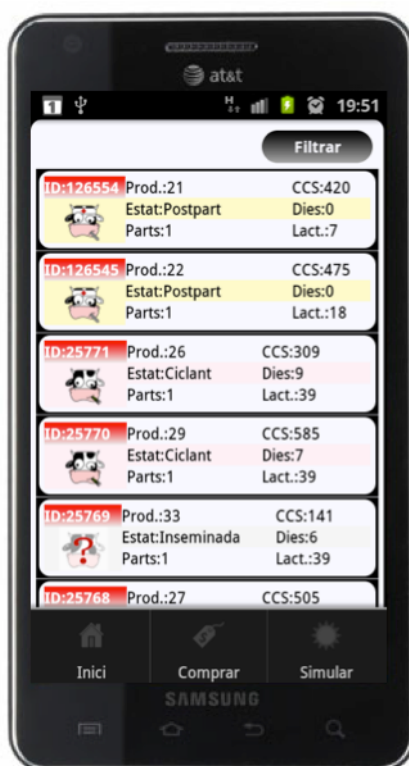


Figura 48. Menú d'opcions accessible des del botó de menú del mòbil.

- **Inici:** aquesta opció tanca la activitat des d'on és cridada i obre el Menú principal. Permet que l'usuari torni al inici de l'aplicació des de qualsevol punt on es trobi.
- **Comprar:** aquesta opció compra una nova vaca per la granja de l'alumne al cost al que estiguin els animals al moment de fer la compra.

Un cop realitzada l'acció, si s'ha pogut comprar correctament, apareix per pantalla un missatge de confirmació amb l'identificador de la vaca comprada.

- **Simular:** aquesta acció avança un dia la simulació de la granja de l'alumne, aplicant totes les accions que l'usuari hagi realitzat sobre vaques, vedells i dieta des de l'última simulació. Un cop feta la simulació porta a l'usuari a l'activitat del Menú principal ja que és allà on apareixen els avisos corresponents si la simulació ha desencadenat esdeveniments rellevants. Internament quan es fa la simulació es crida també l'acció de validar credencials d'usuari. Això es fa d'aquesta manera ja que si no es podria invocar el servei des de qualsevol lloc coneixent l'identificador de la granja. D'aquesta manera, en canvi la simulació només es pot executar si qui fa la crida coneix les credencials de l'usuari. Aquest comportament és l'equivalent a comprovar les variables de sessió en les aplicacions web.

## 5.4 DISTRIBUCIÓ DE L'APLICACIÓ

Un cop desenvolupada és necessari publicar-la per tal que els usuaris la instal·lin al seu dispositiu. El Software Development Kit d'Android ofereix la opció d'exportar l'aplicació a un arxiu amb extensió .apk. Aquest arxiu és un executable de l'aplicació que conté tot el codi i llibreries necessàries i a partir del qual es pot executar l'aplicació des de qualsevol dispositiu amb sistema operatiu Android (Figura 49).

Per publicar una aplicació Android cal, a més del descrit en el paràgraf anterior, que el desenvolupador signi digitalment l'aplicació. Aquesta signatura és utilitzada pel sistema operatiu per validar la identitat de l'autoria de cada aplicació i per establir relacions de confiança entre aplicacions. No és necessari que aquesta signatura es realitzi a través de cap autoritat de certificació si no que pot fer servir certificats auto-signats amb claus generades mitjançant eines estàndard de generació de claus (con *Keytool* o *Jarsigner*).

Un cop generat l'arxiu .apk aquest es pot instal·lar a qualsevol dispositiu Android. El canal comercial més utilitzat per distribuir aplicacions Android és l'aplicació *Google Play* (anteriorment anomenat *Android Market*), ara bé, té una quota de registre de 25€ i pel nostre cas no és necessari ja que simplement ubicant l'arxiu .apk en un recurs de la xarxa i descarregant-la al dispositiu mòbil a través de qualsevol navegador el sistema operatiu detecta el tipus d'arxiu i permet fer la instal·lació.



Figura 49. Procés de generació d'una aplicació Android.

## 5.5 CONTROL DE VERSIONS

Donat que tot i que l'arquitectura d'implementació que s'ha triat permet tenir l'aplicació web i l'aplicació Android bàsicament desacoblades, ha calgut establir un procediment per portar a terme les actualitzacions de codi del servidor de manera que, un cop a producció ambdós serveis, es pugui garantir el correcte funcionament en moments d'actualització de versions.

D'entre les diferents opcions que existeixen al mercat s'ha determinat fer ús del programari GIT (<http://git-scm.com/>), un servei de control de versions que permet la gestió de repositoris de codi. A diferència d'altres sistemes de control de versions, com per exemple *Subversion*, GIT permet a cada desenvolupador disposar de la totalitat del codi en local per treballar. Això dona més independència als desenvolupadors permetent més llibertat a l'hora de fer canvis.



## 6 PROVES

Per tal de validar que el desenvolupament realitzat funciona correctament i dona resposta a tots els requeriments i les funcionalitats plantejades com a objectiu a l'inici del projecte és necessari planificar un conjunt de proves a tots els nivells, des de proves d'integració entre els dos sistemes que s'han de comunicar fins a simulacions del comportament real de l'aplicació realitzant les accions que un alumne faria en el transcurs d'un curs.

També és important fer proves amb diferents dispositius mòbils ja que un dels principals problemes amb que es troben les aplicacions Android es amb incompatibilitats degudes a la gran fragmentació que hi ha en les versions del sistema operatiu i també degudes a les diferències de resolució de pantalla entre els diferents dispositius que poden fer que algunes interfícies no es visualitzin correctament.

### 6.1 PROVES D'INTEGRACIÓ

S'ha validat el correcte funcionament de tots els serveis de l'aplicació que s'integren amb els *servlets* creats a aquest efecte al servidor. En aquest apartat tots els serveis han funcionat correctament, si bé és cert que després de les proves s'han hagut de posar controls d'error a l'aplicació.

La majoria de serveis cridats des de l'aplicació esperen rebre respostes en format JSON però la resposta dels serveis es rep com un *String*. Per aquest motiu el primer que es fa amb la resposta és transformar-la com a JSON. Si per qualsevol motiu (el servidor no respon a temps o es perd la connexió a Internet) no hi ha resposta l'aplicació tractava de convertir a JSON una resposta buida i per tant feia tancar l'aplicació de forma inesperada. El que s'ha fet per arreglar aquest error és posar controls abans de fer la transformació.

## 6.2 PROVES D'INTERFÍCIE

S'han realitzat les proves d'interfícies amb un terminal *Samsung Galaxy S II* i amb un *Samsung Galaxy Note*. També s'ha utilitzat l'emulador que ofereix el kit de desenvolupament de software d'Android per simular les proves amb altres dispositius amb resultats satisfactoris. Tot i les diferències entre les resolucions i mides de pantalla les interfícies escalen correctament per cada dispositiu.

## 6.3 PROVES FUNCIONALS

Per realitzar les proves funcionals de l'aplicació s'ha simulat en la mesura del possible el comportament que tindria un alumne en el transcurs d'un curs de l'assignatura simulant més de trenta dies.

També s'ha fet una demostració funcional a la professora de l'assignatura explicant la forma en la que s'han resolt les necessitats del projecte.

Les proves realitzades i la demostració feta han posat de manifest alguns errors menors com per exemple el nombre de decimals que es mostraven en els totals dels valors dels ingredients de la dieta o bé un funcionament incorrecte de les ajudes visuals en els vedells en zel.

Un cop corregits aquests errors s'ha posat de manifest que l'aplicació respon correctament a totes les situacions provades i no s'han detectat problemes addicionals, per tant considerem que l'aplicació funciona correctament a nivell funcional.

## 6.4 PROVES REALS

Per realitzar les proves reals s'ha duplicat el projecte del simulador en el mateix servidor de producció en el que està funcionant l'aplicació. Seguidament s'ha afegit la classe amb els *servlets* i les llibreries i modificacions en la configuració

necessàries per tal d'habilitar els serveis web afegits. Finalment s'ha exportat l'aplicació Android a un arxiu .apk i s'ha ubicat per la seva descàrrega en el mateix servidor on s'ubica l'aplicació web.

A continuació la professora de l'assignatura ha descarregat l'aplicació i ha realitzat les proves que ha considerat pertinents donant finalment, després d'una ronda de correccions, el seu vist i plau al projecte.

Finalment i amb l'objectiu de mesurar el grau de satisfacció del resultat final de l'aplicació s'ha passat una enquesta preguntant per la opinió sobre diferents aspectes del projecte a la professora de l'assignatura:

## Enquesta Simulador de Productivitat en una Explotació Lletera en Android

Nom:  Lorena Castillejos

Càrrec:  Directora tècnica de l'SNIBA

Puntuï amb un cercle segons la següent escala la resolució i el resultat de cadascun dels aspectes de l'aplicació que es descriuen a continuació:

1	Malament
2	Regular
3	Bé
4	Molt bé
5	Excel·lent

Qüestionari de satisfacció	Puntuació				
1. Facilitat d'ús a nivell general	1	2	3	4	5
2. Compliment de les funcionalitats esperades	1	2	3	4	5
3. Disseny general de les interfícies (aspectes) de l'aplicació	1	2	3	4	5
4. Funcionament i disseny de la pantalla d'Indicadors	1	2	3	4	5
5. Funcionament i disseny de la pantalla d'economia	1	2	3	4	5
6. Funcionament i disseny de la pantalla de llista de vaques / llista de vedells	1	2	3	4	5
7. Funcionament i disseny de la pantalla de detall de vaca / detall de vedells	1	2	3	4	5
8. Funcionament i disseny de la pantalles de filtratge de vaques/ vedells	1	2	3	4	5
9. Funcionament i disseny de la pantalla de dieta	1	2	3	4	5

L'aplicació podria ser utilitzada com a substitut o complement de l'aplicació web actual pels alumnes?

Sí	Només després d'alguns evolutius menors	Només després d'un conjunt elevat d'evolutius	No
----	---	---	----

Comentaris addicionals:

Signatura de l'enquestat



Figura 50. Enquesta de satisfacció emplenada pel client

Els evolutius menors que es consideren imprescindibles per utilitzar l'aplicació resultant del projecte en l'assignatura corresponen a canvis que s'han anat realitzant en l'aplicació web a la vegada que hem estat desenvolupant el nostre projecte i que, per aquest motiu, no s'han pogut incloure en les funcionalitats.

## 7 CONCLUSIONS

### 7.1 SEGUIMENT DEL PROJECTE

Havent finalitzat el desenvolupament del projecte podem confirmar que tant la fase d'anàlisi com la de disseny s'han realitzat de forma correcta pel que fa a la detecció i avaluació de la dificultat i complexitat de les tasques. Tot i així degut sobretot a la necessitat d'adquirir nous coneixements especialitzats per a la resolució d'alguns dels problemes apareguts durant el desenvolupament algunes tasques s'han estès més del previst, fent que la planificació que es va fer abans de començar el projecte s'hagi vist afectada. Sobretot ha augmentat el temps de desenvolupament de l'activitat de la dieta ja que es va realitzar una primera proposta d'interfície que, després de presentar-la, es va descartar ja que no complia amb els objectius desitjats. Això va fer que s'hagués de descartar feina ja realitzada i tornar a plantejar el mòdul.

També ha presentat certs problemes tècnics la implementació de la càrrega progressiva d'elements a les llistes de vaques i vedells. Tot i la interfície d'aquesta funcionalitat és simple la implementació és complexa ja que s'han de capturar i controlar esdeveniments relatius a la posició dels elements de la pantalla (el fet d'arribar al final dels elements visibles de la pantalla) i realitzar crides asíncrones al servidor, gestionant els resultats per afegir-los al final de la activitat visible.

### 7.2 VALORACIÓ PERSONAL

L'experiència de realitzar aquest projecte ha estat francament positiva. He pogut aprendre a treballar amb una tecnologia molt actual que es troba en ple procés d'expansió i que crec que és un coneixement extremadament útil i que ho serà més amb el temps, ja que les aplicacions per dispositiu mòbil només poden créixer en el futur.

El fet d'haver d'adaptar un producte que ja està en ús ha tingut aspectes positius i aspectes negatius. Com a punt negatiu, el fet de tenir una aplicació desenvolupada sobre la que treballar i amb la que he hagut de mantenir total integració ha significat en algun dels casos haver d'adaptar-me a funcionalitats que jo hauria plantejat d'una altra manera però que he hagut de mantenir. Com a punt positiu, al delegar les funcions complexes i de negoci de la granja de vaques he tingut més temps i llibertat per pensar en el millor aprofitament dels recursos que ofereixen els dispositius mòbils i això m'ha permès dissenyar unes interfícies que crec han quedat prou atractives.

### 7.3 FUTUR DEL PROJECTE I POSSIBLES MILLORES

Tal i com dèiem a l'inici del projecte, aquest és només el primer pas a tot un nou horitzó de possibilitats que a partir de la finalització d'aquest projecte es podran començar a plantejar. L'adaptació del simulador a Android significa que ara els professors i responsables de l'assignatura poden plantejar escenaris de simulació difícilment realitzables en una aplicació web ja que poden tenir en compte que l'alumne portarà sempre el mòbil amb ell i per tant podrà interactuar en qualsevol moment amb l'aplicació. A més, el fet que les aplicacions per a mòbil puguin rebre avisos i accions des de l'exterior fa que les accions no les hagi de començar sempre l'alumne si no que es puguin originar en el sentit invers. En aquesta direcció, per exemple, es podria fer que els esdeveniments com els naixements es generessin en temps real, en comptes de al fer la simulació i que, en el part s'enviés una alerta a l'alumne que tingués un temps limitat per realitzar les accions pertinents pel cuidat del vedell nounat.

En definitiva, el projecte realitzat ha assolit les fites plantejades, tant a nivell tècnic com a nivell funcional. A més hem pogut contrastar amb el client que la solució desenvolupada compleix satisfactòriament amb les necessitats de l'assignatura. Per tant podem concloure que el resultat del projecte ha sigut exitós.

## 8 BIBLIOGRAFIA

### 8.1 LLIBRES

- Naci Dai, Lawrence Mandel, Arthur Ryman: Eclipse Web tools platform: developing Java Web applications, Pearson Education, Inc. 2007
- Reto Meier: Professional Android Application Development, Wiley Publishing, Inc. 2009
- Jason Morrid: Android User Interface Development, Packt Publishing Ltd. 2011
- James Steele, Nelson To: The Android Developer's Cookbook, Pearson Education, Inc. 2010

### 8.2 REFERÈNCIES A INTERNET

- <http://stackoverflow.com>: (04/02/2012-16/02/2012) comunitat de desenvolupadors on es resolen dubtes sobre tot tipus de llenguatges de programació.
- <http://developer.android.com/guide/practices/>: (16/02/2012) guia oficial de bones pràctiques pel disseny d'aplicacions en dispositius Android.
- [http://www.androidpatterns.com/uap\\_pattern/](http://www.androidpatterns.com/uap_pattern/): (20/01/2012-26/01/2012) guia que conté els diferents tipus de patrons de disseny d'interfícies per Android amb els principals pros i contres de cadascun.
- <http://www.vogella.de/articles/AndroidListView/article.html>: (06/02/2012, 09/03/2012) tutorial per a la introducció en l'ús de ListViews.

-----  
Signat: **David Ramos Morera a juny de 2012**