Enquestes

Aplicació web per a la creació d'enquestes

Memòria del projecte
d'Enginyeria en Informàtica
realitzat per
David Eduard Espinosa Ruiz
i dirigit per
Anna Morajko

El sotasignat, ...

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

I per tal que consti firma la present.

Signat: .....
Bellaterra, ...de.....de 200.....

Agraïments:

Podria escriure 100 línies d'agraïments a aquelles persones que m'han ajudat constant o puntualment durant el camí transcorregut, podria agrair a aquells que s'esforcen per fer circular el coneixement, a aquells que fan possible un món de codi obert, a aquells que ajuden al altres a arribar als seus objectius, podria repartir el meu agraïment durant hores i encara això témer per oblidar-me un nom, una cara. Podria.....

A mis padres,
Antonio Espinosa Arroniz
Maria Rosa Ruiz Giménez.

Índex

1. INTRODUCCIÓ 9

1.1. MOTIVACIÓ 9
1.2. OBJECTIU 14
1.3. ORGANITZACIÓ 14

2. REQUERIMENTS 16

2.1. VIDA DE LES ENQUESTES 16
2.2. ACTORS 18
2.2.1. Client 18
2.2.2. Usuari 20
2.3. OBJECTES DE NEGOCI 21
2.3.1. Enquesta 21
2.3.2. Pàgina 25
2.3.3. Pregunta 25
2.3.4. Resposta 29
2.3.5. Format 30
2.4. CREACIÓ D'OPINIONS 33
2.4.1. Accés 33
2.4.2. Opinió 34
2.5. GENERACIÓ D'ESTADÍSTIQUES I RESULTATS 35

3. ARQUITECTURA 39

3.1. DESCRIPCIÓ GENERAL 39
3.1.1. J2EE 40
3.2. MODEL 42
3.3. ABSTRACCIÓ DE LA BASE DE DADES MITJANÇANT HIBERNATE 45
3.4. EL SERVIDOR 47

4. DISSENY I IMPLEMENTACIÓ 48

4.1. DESCRIPCIÓ 48
4.1.1. Struts 48
4.1.2. Estructura d'arxius 49
4.1.3. Classes 51
4.2. BASE DE DADES 52

5. CONCLUSIONS 56

5.1. LA APLICACIÓ, VALORACIONS 56
5.1.1. Anàlisis de requeriments i disseny de les estructures de dades necessàries 56
5.1.2. Construcció del framework 56
5.1.3. Implementació 57
5.1.4. Configuració servidor 58
5.1.5. Fase de proves 58
5.2. COMPARACIONS 59
5.3. POSSIBLES AMPLIACIONS I MILLORES 59

BIBLIOGRAFIA 61

1. Introducció

1.1. Motivació

*Una **enquesta** és un conjunt de preguntes normalitzades dirigides a una mostra representativa de la població o institucions, amb la finalitat de conèixer estats d'opinió o fets específics.*

Wikipedia [1]

*Una **enquesta** consisteix en reunir dades entrevistant a la gent.*

Stanton, Etzel y Walker [2]

*Una **enquesta** és un mètode per a obtenir informació sistemàticament dels enquestats a través de preguntes, ja siguin personals, telefòniques o per correu.*

Richard L. Sandhusen [3]

*Una **enquesta** és un conjunt d'entrevistes utilitzant un qüestionari dissenyat de forma prèvia.*

Naresh Malhotra [4]

Conèixer l'opinió general d'un conjunt de persones sobre un tema en concret cada vegada preocupa més. Aquesta és una preocupació lògica, doncs l'èxit del tema avaluat sol anar lligat de les consideracions que d'aquesta es deriven.

Són molts els escenaris en els que conèixer la opinió global són importants. A la política, en temps de eleccions, les enquestes d'intenció de vot apareixen en primera plana dels diaris i poden ser claus per a decantar a un cantó o a un altre el vot indecís. A la figura 1 podem veure un exemple d'enquesta de temàtica política. En el món del consum d'aliments, quant una marca comercial treu al mercat un nou producte, aquest primer es sotmès a varies fases de "test" per a conèixer l'opinió sobre molts i diversos aspectes d'aquest producte, així com per a conèixer el hàbits i costums d'un conjunt de la població com es pot veure en la figura 2. També a les universitats es succeeixen diverses enquestes, per conèixer els costums d'accés així com el vehicle utilitzat per els estudiants, l'opinió sobre els professors o sobre les instal·lacions. Un exemple és la última enquesta accessible des de el campus virtual com es pot veure en la figura 3.

És més, les enquestes són un dels mètodes més utilitzats en la investigació de mercats. De fet el Diccionari de Màrqueting de Cultural S.A., defineix el terme enquestar com el mètode de recollida de informació quantitativa que consisteix en interrogar als membre d'una mostra, sobre la base de un qüestionari perfectament estructurat.

Fig. 1
Enquesta sobre política.

ENCUESTA SOBRE ALIMENTOS

Por favor marque con una X el casillero que mejor lo describe a usted.	Nunca (1)	A veces/menos de la mitad de las veces (2)	La mitad de las veces (3)	La mayoría de las veces/más de la mitad de las veces (4)	Casi siempre (5)
(1) Planifico las comidas.					
(2) Comparo los precios de los alimentos y uso el precio unitario.					
(3) Me quedo sin alimentos, sin dinero para alimentos, o sin cupones para alimentos antes de fin de mes.					
(4) Uso una lista cuando salgo a comprar comestibles					

Fig. 2
Enquesta sobre aliments.

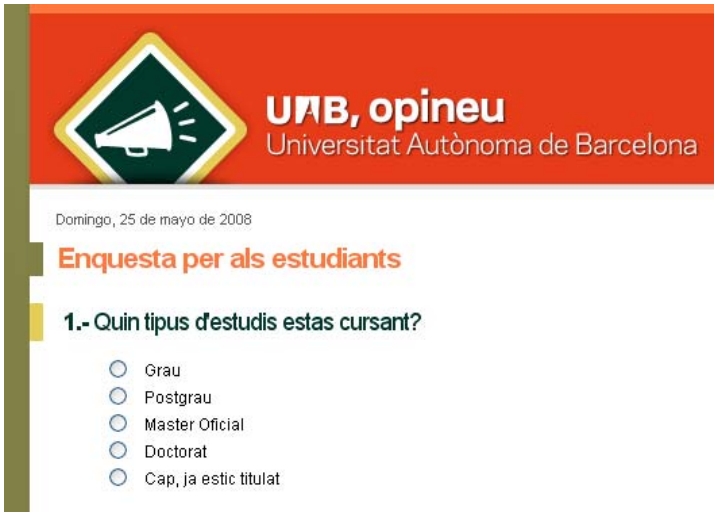


Fig. 3
Enquesta de l'Autònoma dirigida als estudiants.

Entre les avantatges de les enquestes, podem trobar:

- Permet obtenir informació de quasi qualsevol tipus de població.
- Permet obtenir informació exacte sobre fets passats dels enquestats.
- Gran capacitat per a estandarditzar dades, el que permet el seu tractament informàtic així com l'anàlisi estadístic.
- Bona relació entre el seu cost i la quantitat de la informació obtinguda.

Com a desavantatge podem destacar que no permet analitzar temes complexos, circumstàncies aquestes a les que hauríem de recórrer amb grups de discussió (una sèrie de enviaments sobre un tema en particular que son enviats a un servidor de notícies que els distribueix a d'altres servidors participants).

Fins a l'explosió de Internet, el medi per a les enquestes eren:

- Entrevista personal portada a terme per un entrevistador.
- Per correu, sent aquesta la solució econòmica però amb l'índex de no respostes més elevada.
- Qüestionaris telefònics, solució molt econòmica també però on no controlem la identitat (situació laboral, acadèmica...) de la persona enquestada.

Totes aquestes solucions, encara que pràctiques sempre acaben depenent de factors de disponibilitat del enquestat, i moltes vegades del enquestador. I a més, resulten tenir (i encara que antigament no es considerés així) una relació cost/informació molt pitjor que amb les noves solucions que aporta Internet. Així doncs la possibilitat de difondre les enquestes per la web ha estat vista per empresaris que veient aquesta possibilitat com una oportunitat de negoci (bé perquè poden cobrar al usuari enquestador o bé perquè poden inserir-hi publicitat):

Fig. 5
Survey Monkey [5]

Fig. 6
2ask [6]

Fig. 7
Survey pro [7]

Les figures 8 9 i 10 ofereixen captures de la aplicació *SurveyMonkey* [5] on podem veure el procés de creació d'una de les preguntes de l'enquesta. En les figures 11 i 12 podem veure el procés d'edició d'una pregunta de selecció única (accepta una única resposta) de l'aplicació *2ask*, més complerta que l'anterior.

Totes aquestes webs, ofereixen al usuari un servei de creació i enviament d'enquestes per la xarxa per a la recaptació posterior de opinions en forma de respostes. Aquesta informació resultant es mostrada al usuari en forma absoluta (total de respostes per pregunta i opció de resposta) així com estadística. Encara que diversa, el format de cobrament per a aquests serveis no sol ser per nombre de enquestes creades, sinó pel nombre de respostes rebudes, essent aquest l'objecte de negoci primordial.

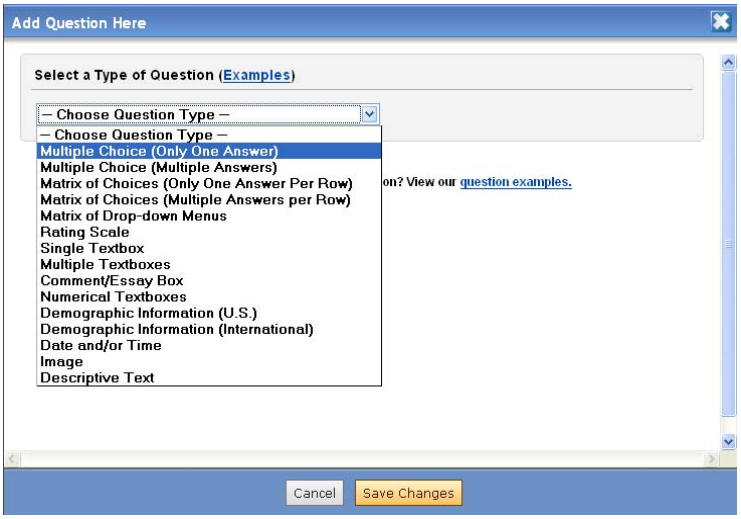


Fig. 8

Diàleg per a escollir el tipus de pregunta a inserir.

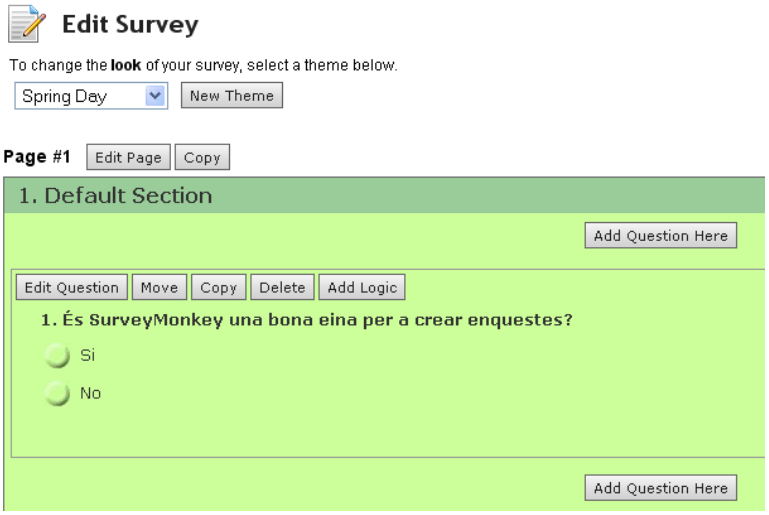


Fig. 9

Edició de pàgina.

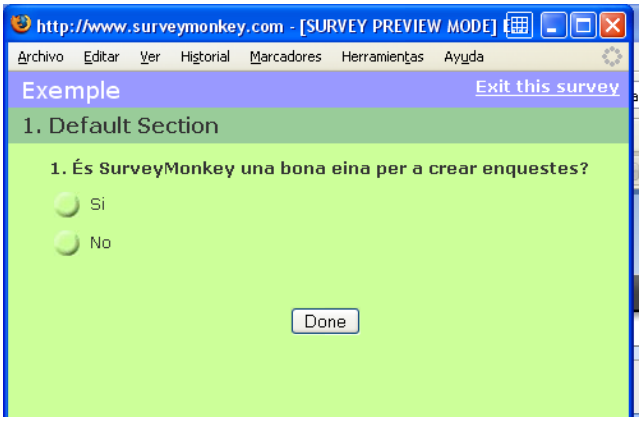


Fig. 10

Visualització de la nostre enquesta.

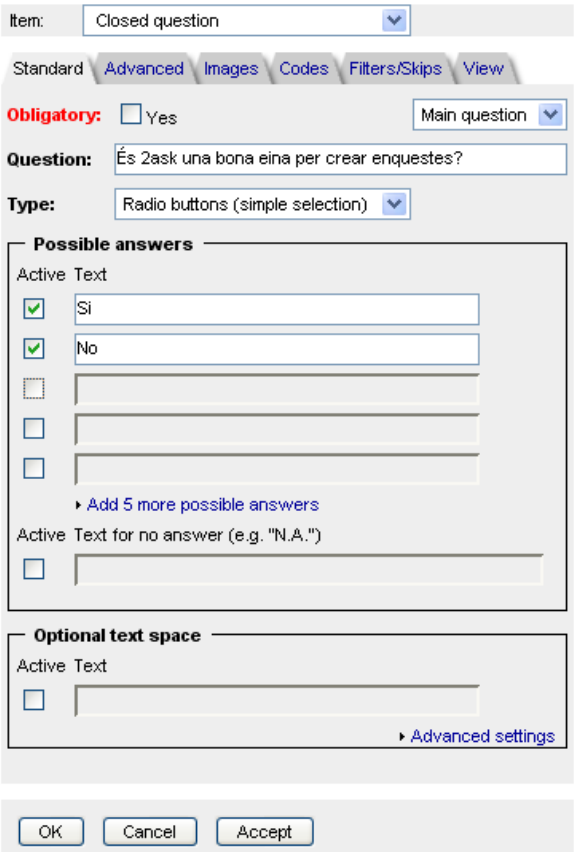


Fig. 11
Edició de pregunta.

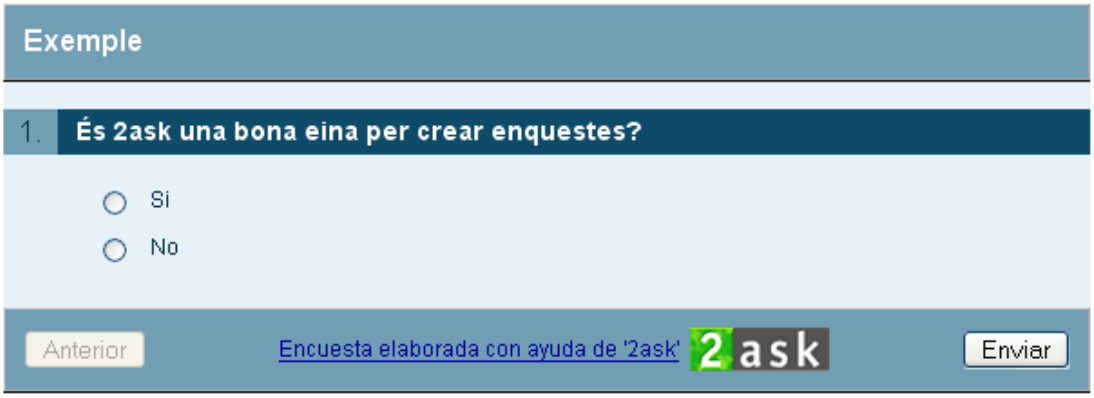


Fig. 12
Visualització de la nostre enquesta.

Per a apropar-se al usuari, una aplicació de creació d'enquestes online, ha d'oferir una interfície web "*friendly*" i intuïtiva, que permeti al usuari dedicar la seva atenció a la creació lògica de la seva enquesta i no al propi funcionament de l'eina. A més, algunes d'aquestes aplicacions, permeten formular diferents preguntes en funció de les respostes adquirides anteriorment. Per a les respostes aconseguides a través del usuari, aquestes solen ser obtingudes per l'usuari enquestador en varis formats. Però encara totes aquestes característiques, els usuaris enquestadors sempre troben a faltar alguna altre, que bé permetí personalitzar encara més les seves enquestes, o d'altres

aspectes relacionats amb l'adquisició de dades o bé com les enquestes arriben als enquestats. I es que com a la vida real, no és només important el *que* es pregunta sinó que també s'ha de tenir en compte el *com.*

Així doncs, sense oblidar la finalitat de l'enquesta, els esforços per a millorar aquesta s'orienten en:

1. Agilitzar la creació de l'enquesta per part de l'enquestador.
2. Dotar d'opcions el procés de creació per a personalitzar l'enquesta.
3. Procurar mètodes d'accés a l'enquesta creada.
4. Retornar les opinions dels enquestats al enquestador de manera pràctica.

1.2. Objectiu

L'objectiu de *enquestes* és la creació i distribució d'enquestes, de les que es deriven els resultats o opinions. És a dir e*nquestes es pot definir com* un portal web per a la generació i gestió d'enquestes. Aquests resultats són alhora l'objectiu del usuari creador, que anomenarem *client*, que vol conèixer l'opinió de l'enquesta't, que anomenarem *usuari*. Els resultats arriben al *client*, que els visualitza de forma directe, veient per pantalla el nombre de respostes donades per a cada opció, o bé gràficament, visualitzant les respostes en forma de gràfics.

Una vegada creades, aquestes han de poder ser publicades per diferents vies, retornar els resultats i poder-los tractar per a presentar estadístiques. El *client* ha de poder crear l'enquesta, atenent a diferents modalitats, estructures i opcions, per a que aquesta s'adapti al màxim a l'estètica que el creador li volguí donar. L'enquesta pot ser publicada en obert per a que qualsevol *usuari* hi pugui accedir des de la pàgina principal de l'aplicació, o bé ser enviada per mail així com per link a una llista restringida d'*usuaris*. Els resultats creats són inserits a una base de dades, que posteriorment és consultada per a poder crear estadístiques en funció de la variable d'*observació* indicada, presentant la informació en forma de gràfics de diferents tipologies.

1.3. Organització

L'organització d'aquest treball és divideix en cinc capítols:

Requeriments:

Aquest capítol descriu els elements que intervenen a la aplicació, així com la seva incidència en la aplicació. És defineix el cicle de vida de l'objecte lògic principal, la enquesta, i com l'usuari interactua amb la aplicació per a inicialitzar-la, construir-la i complementar-la. És presenten els components bàsics de l'enquesta: la pàgina, la pregunta i les respostes, així també com els actors: el client o enquestador, i el usuari o enquestat.

Arquitectura:

Enquestes és una aplicació web, en aquest capítol es presenten els components que formen la seva arquitectura. Es descriuen els aspectes i característiques principals amb la que es desenvolupada *Enquestes* i finalment s'introdueix el model MVC (model-vista-controlador), el model que descriu el funcionament d'aquesta aplicació així com de la majoria de projectes J2EE.

Disseny i implementació:

Com està organitzada la aplicació, com és controlat el flux de la aplicació per la entitat controladora ***Struts***, com s'encapsulen les dades o com està distribuïda la informació a la base de dades, tot això es troba descrit en aquest capítol.

Conclusions:

Comparacions entre *Enquestes* i altres aplicacions similars, quins són aquells complements que podrien millorar les prestacions de *Enquestes*.

2. Requeriments

2.1. Vida de les enquestes

Les *enquestes* tenen la propietat d'estar *tancades* o *obertes*. Quant un *client* construeix una *enquesta*, aquesta comença estant *oberta* i ho deixa de ser quant el *client* propietari de l'*enquesta* determina que ja està complerta i canvia el seu estat a *tancada*. Una vegada determinada com tancada, aquesta pot ser resposta pels *usuaris* i no podrà ser modificada pel *client*. Això succeeix no tindria sentit que haguessin *opinions* de preguntes que haguessin estat eliminades o bé modificades.

Una vegada l'enquesta està tancada (fins que no ho està, no podrà ser resposta), aquesta ja pot ser resposta pels clients. Per a que aquests hi puguin accedir, l'enquesta té dos camins diferents per a arribar-hi:

- Enviament directe d'enquesta a l'*usuari*.
- Mètode públic

El diagrama següent, figura 13, ens mostra les fases per les que una enquesta pot passar:

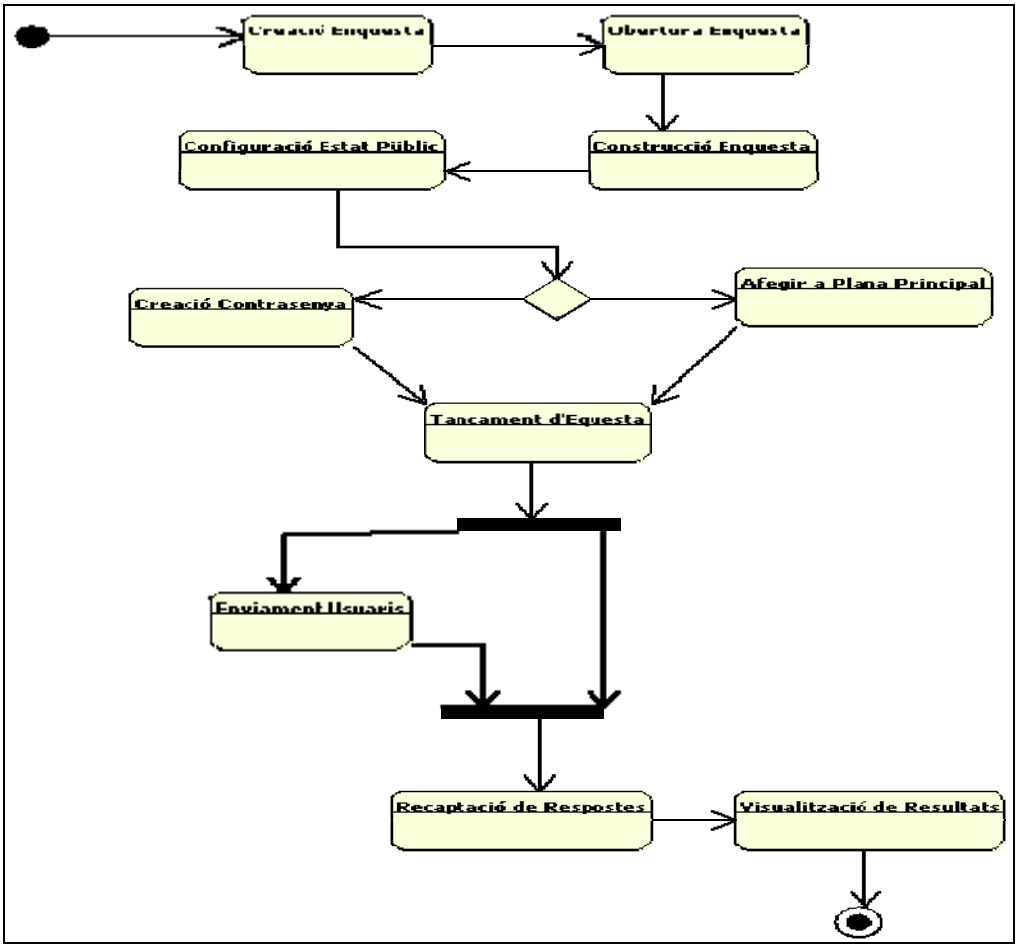


Fig. 13
Diagrama d'estats, vida d'una enquesta.

Quan el *client* vol que aquesta enquesta sigui resposta per una selecció determinada de *usuaris*, aquest generarà una llista de distribució, és a dur una sèrie de persones a les que se li farà arribar l'*enquesta*. El mètode de distribució de la enquesta, serà mitjançant l'enviament d'una *URL* per mail, des de la que podrà accedir l'*usuari* receptor introduint una determinada contrasenya (inclosa o no en el mail).

Quan la enquesta sigui classificada com *pública*, podrà ser accedida per mail com a l'opció anterior, però a mes, podrà ser accedida des de la plana principal sense que l'*usuari* hagi d'introduir cap contrasenya. Fet que no implica que el *client* no pugui enviar la URL de l'*enquesta* a una llista de distribució, sense que els components d'aquesta hagin d'introduir cap contrasenya. Per tant, la pàgina principal servirà d'entrada a qui vol crear *enquestes* com a qui només vol respondre'n.

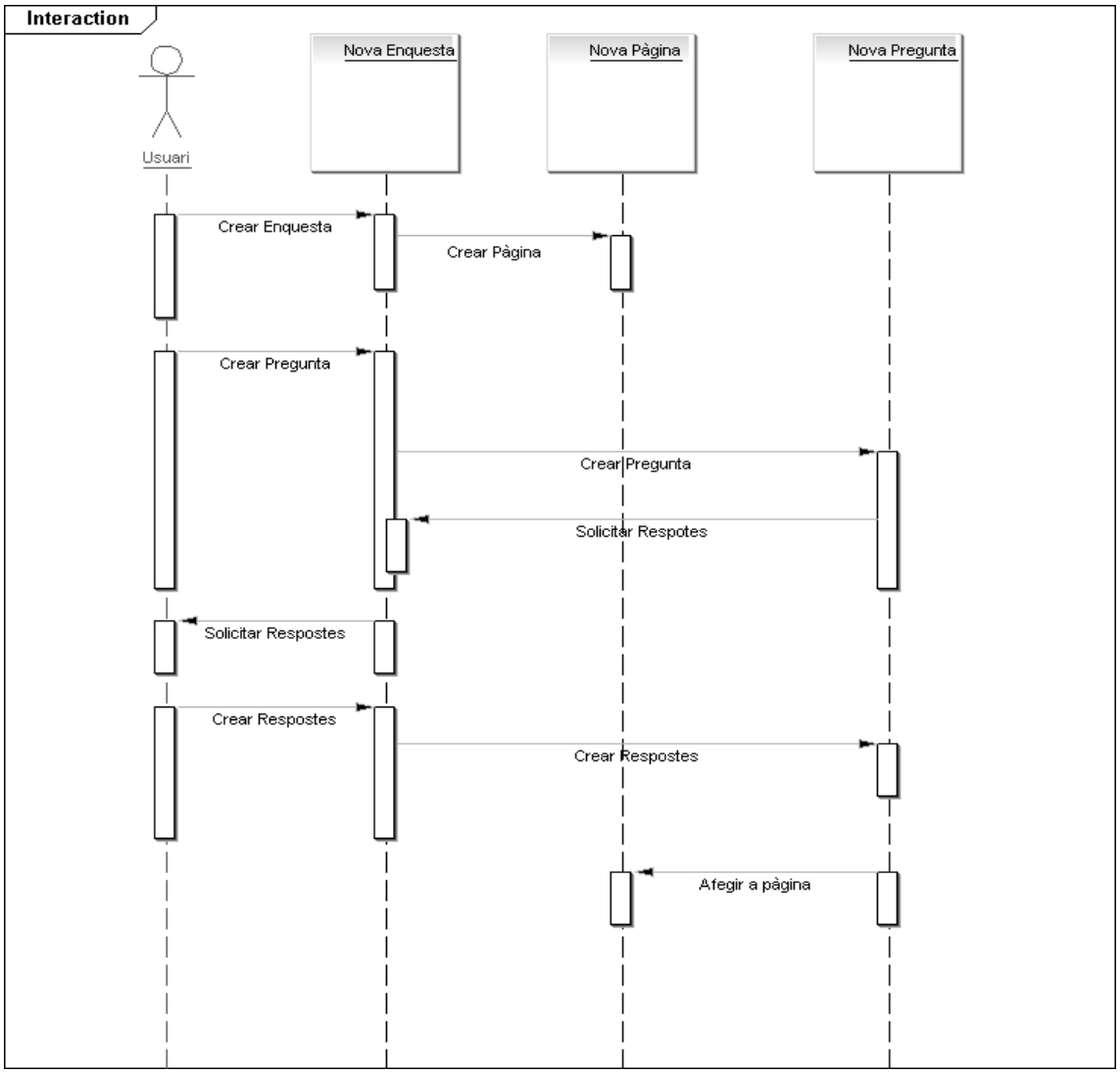


Fig. 14
Diagrama interacció creació enquesta

Al anterior diagrama, figura 14**,** d'interacció podem observar els passos que ha de seguir l'enquestador (l'usuari) per a la creació de l'enquesta.

Crear Enquesta: Una enquesta no és només l'objecte principal de la aplicació, sinó l'objecte que encapsularà tots els altres que s'afegeixin posteriorment.

Crear Pàgina: Una enquesta està formada per pàgines, que són les que contenen les preguntes i el principal mitjà per a distribuir aquestes a l'enquesta.

Crear Pregunta: Al igual que una pàgina està constituïda per preguntes, una pregunta ho està de respostes. El text de la pregunta és la qüestió que l'enquestat ha de plantejar-se.

Crear Resposta: Finalment, oferim al usuari enquestat les opcions a contemplar per a la resposta de la pregunta.

2.2. Actors

2.2.1. Client

Hi ha dos actors a *enquestes*, qui crea enquestes, els *clients* de l'aplicació, i els qui les responen, els *usuaris*. Aquesta aplicació està encarada als *clients*, a que aquests tinguin una aproximació de la opinió general sobre una determinada qüestió. Similar a d'altres pàgines d'enquestes, com *SurveyMonkey*, per a poder crear enquestes hem d'omplir un formulari amb la informació:

- Nom (obligatori)
- Cognom (obligatori)
- E-mail (serà el nostre identificador per a accedir-hi) (obligatori)
- Contrasenya (obligatori)
- Empresa (opcional)
- Telèfon (opcional)
- Ciutat (opcional)
- Codi postal (opcional)
- Idioma (no obligatori, per defecte Català)

Encara que tots aquests atributs descriuen el *client* que construirà les enquestes, n'hi ha que no tindran cap altre funció com **nom, cognom, telèfon, país**, **empresa** i **ciutat.** D'altres també permetran de configurar l'aplicació com el **idioma**, fent que l'aplicació s'hi pugui adaptar al idioma escollit per el *client.*

L'atribut **e-mail**, serà el nom d'usuari amb el que el *client* s'haurà de logar a la aplicació juntament amb la **contrasenya**. A més quant el *client* no recordi la seva contrasenya, s'enviarà un mail a la direcció **e-mail** per a recordar-li.

L'atribut **sector** pot resultar un criteri per a la cerca d'*enquestes*, per exemple, podríem buscar totes aquelles *enquestes* relacionades amb la informàtica, amb la economia o amb el sector automobilístic.

Així doncs, el *client* és l'actor més important de l'aplicació, doncs el fi d'aquesta és crear *enquestes* (per un *client*), per que siguin respostes, i poder així crear estadístiques que seran presentades al creador de la *enquesta* en qüestió, el *client* que la va crear.

Fig. 15
Pantalla de creació de nou client.

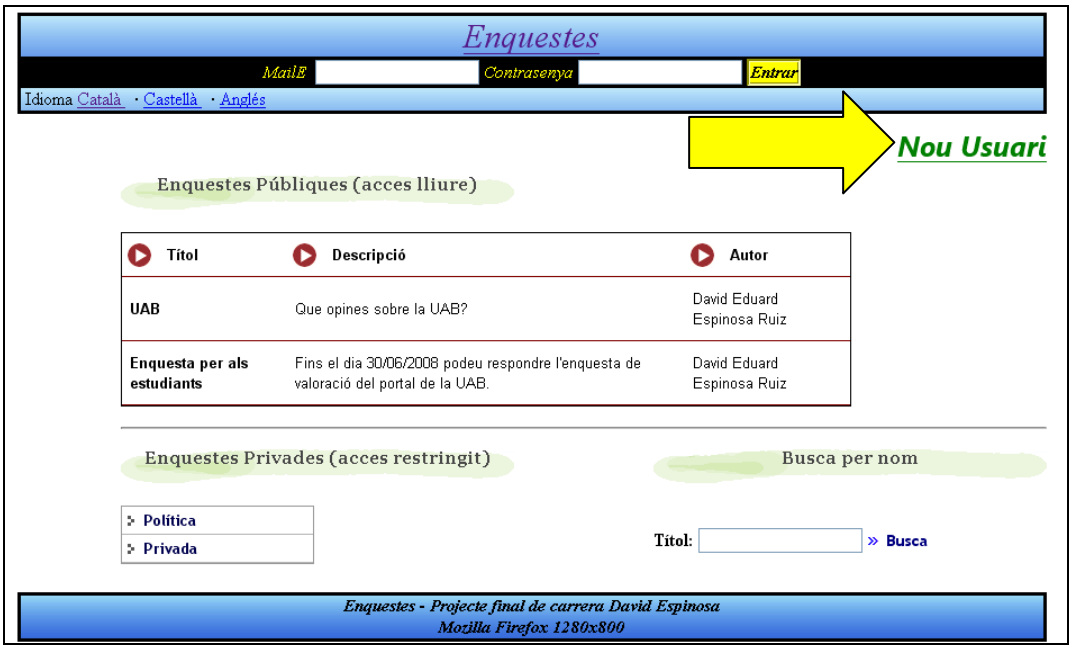


Fig. 16
Accés a la pantalla d'alta de client.

A la figura 15 podem observar la finestra per a la creació d'un nou usuari. Els camps amb * indiquen aquells necessaris per a completar l'alta d'un usuari. Els camps de *nom* i *cognom* identifiquen el creador de les enquestes que aquest usuari pugui crear posteriorment. El *correu electrònic* no només és el mitjà en el que l'aplicació podria comunicar-se amb el usuari, sinó que també és conjuntament amb el camp *contrasenya*, les dades necessàries per a logar-se a *Enquestes.*

L'accés a aquesta pantalla és pot identificar fàcilment a la pantalla inicial, com poder observar en la figura 16.

2.2.2. Usuari

Com ja hem dit, una *enquesta* serà creada per un *client* que s'hagi registrat a la aplicació, així doncs necessitem un altre actor que hi accedeixi per a respondre-la, és a dir, crear *opinions*. Aquest actor es el *usuari.*

Un *usuari* no ha d'estar registrat a la aplicació, ho pot ser qualsevol visitant del portal que decideixi respondre una enquesta, però també tindrà l'oportunitat d'inserir la seva informació personal mitjançant un registre. Aquest registre permetrà obtenir més informació estadística de les respostes al *client*. El registre d'un *usuari* si es portarà a terme (amb la mínima informació i nomes com a camp obligatori el e-mail) quant un *client* el vulgui inserir a la llista de distribució.

Per tant un *usuari* pot contenir les següents característiques:

- Nom
- Cognom
- E-mail (obligatori)
- Llista de distribució a la que pertany (taula)

La *llista de distribució* és un atribut que no podrà modificar l'*usuari*, ni tan sols el podrà veure. Aquest camp és omplert per un *client* que vol fer arribar la seva *enquesta* a un conjunt d'*usuaris*.

Existeixen dos camins que permeten crear una instància de *usuari*:

- **Per un visitant** de l'aplicació, quant aquest accedeixi expressament a la secció d'alta. Encara això, un *usuari* no tindrà més avantatges del que pot tindré un visitant. Una possible ampliació de l'aplicació podria permetre oferir serveis extres per a aquell visitant que volgués subscriure com a *usuari*, com per exemple l'enviament d'una enquesta.

- **Per un *client***, la raó de que l'únic camp necessari per a l'existència d'una instància de *usuari*, és degut a que es aquest el camp imprescindible per a entrar a formar part d'una llista de distribució.

2.3. Objectes de negoci

Els objectes de negoci (business objects) són aquells elements que creen els actors, i que representen el vehicle en el que les dades viatgen.

A *enquestes*, la creació d'aquests objectes es realitza d'una manera contenidora i jeràrquica. És a dir, un conjunt d'objectes del mateix tipus pertanyen a un altre de nivell més alt, i sense aquest, la seva existència no tindria sentit ni per l'objecte contenidor ni per el contingut. Per exemple, imaginem 4 respostes que no pertanyen a cap pregunta o bé una enquesta sense preguntes.

2.3.1. Enquesta

A nivell lògic, l'*enquesta* és el nivell jeràrquic superior, d'on deriven las resta d'elements que el componen. Com a símil, un bosc (enquesta) està format per arbres (pàgines) que contenen branques (preguntes) on trobem fulles (respostes). Aquest és l'element al que accedirà l'usuari, a la enquesta on podrà viatjar per les diferents pàgines que aquesta contingui, seleccionant les respostes dins de les preguntes contingudes a les pàgines.

El *client*, crea una *enquesta* que té associades les característiques:

- Estat (obert o no).
- Data d'obertura.
- Data de tancament.
- Títol
- Descripció
- Idioma
- Client propietari
- URL
- Pública.
- Contrasenya
- Format
- Auto numeració de pàgines
- Auto numeració de preguntes
- Auto numeració de respostes

Una *enquesta* esta descrita per una sèrie de cadenes de text, com el **títol** i la **descripció**, que ajudaran a la comprensió a prop de la finalitat d'aquesta. Estarà expressada en un **idioma** en concret (aquest és un registre de la base de dades, el que ens permetrà afegir-ne més d'una manera senzilla). És construïda i propietat de un client (**Client** propietari). Té una **URL** associada, per a poder accedir directament a respondre-la, una **contrasenya** per controlar l'accés a aquesta, la possibilitat de ser d'accés **públic** o no. És a dir, quant un **client** hagi creat una *enquesta*, podrà decidir si aquesta pot ser accedida per qualsevol *usuari*, cas en el que la *enquesta* seria considerada com pública, o bé si vol que aquesta sigui accedida utilitzant una **contrasenya** per a fer-ho, essent en aquest cas una *enquesta* privada.

Fig. 17

Menú d'accés a les enquestes.

A la figura 17 podem observar el llistat d'enquestes ja creades per el client, així com alguns detalls d'aquestes, com la descripció associada, la data de creació d'aquesta, l'estat. A més aquest llistat ens dona la possibilitat d'accedir a les respostes rebudes, la de eliminar l'enquesta o la de enviar-la per correu.

Les enquestes en les que apareix un forrellat al costat del nom de l'enquesta, són aquelles que tenen una contrasenya associada, i per tant ningú hi podrà accedir sense ella.

Una vegada creada l'*enquesta*, el *client* podrà decidir quant vol que els *clients* hi puguin accedir a ella mitjançant el camp **Estat**. Fins que el *client* no canviï l'**estat** a obert, cap *usuari* no hi podrà accedir, i per tant el *client* no podrà observar les estadístiques derivades de les opinions d'aquests.

Amb **data d'obertura** i **data de tancament** podrem saber el temps total que ha estat oberta una *enquesta* als *usuaris*. Tenim així un factor estadístic més. **Data d'obertura** s'inicialitzarà automàticament al passar l'**estat** de l'*enquesta* de tancada a oberta, mentre que la **data de tancament** la marcarà el *client* propietari al tancar-la expressament.

Com explicarem més endavant, una *enquesta* està formada per pàgines, les quals contenen preguntes que contenen respostes. Tots aquest objectes tenen un cert ordre per a presentar l'enquesta final a l'*usuari*. Amb els camps **Auto numeració de pàgines**, **Auto numeració de preguntes** i **Auto numeració de respostes** (booleans), donem la possibilitat de numerar aquests objectes automàticament. D'aquesta manera, tant *pàgines*, *preguntes* i *respostes* s'autoanomenaran automàticament al ser inserides a l'enquesta.

Per últim, ens trobem una sèrie de camps que incidiran a la presentació i construcció final de l'*enquesta*, i que afectaran als objectes que es deriven d'aquesta. Amb **format** determinarem la seva aparença estètica mitjançant un fitxer css (*Cascade*

Style Sheet). Aquest serà un camp emmagatzemat també a base de dades. A més, per a personalitzar la nostre *enquesta*, podrem incloure una imatge que la representi, és a dir, un logotip. Aquest últim es presentarà per pantalla quant es construeixi l'*enquesta*.

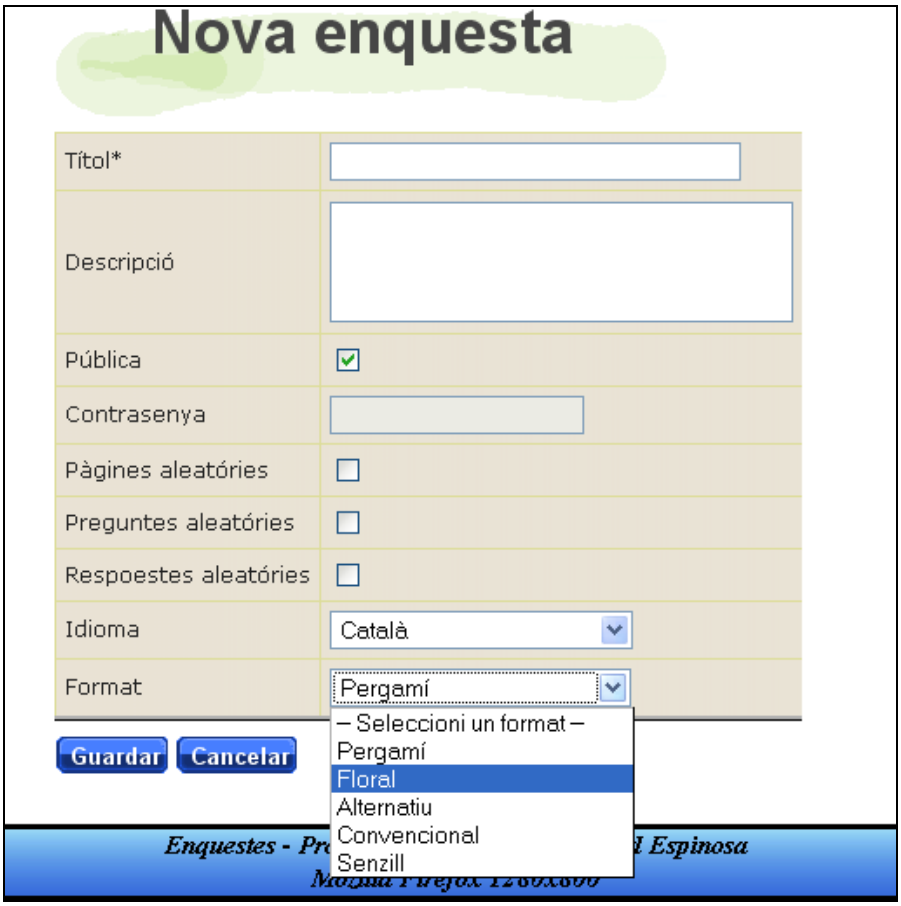


Fig. 18
Pantalla de creació d'enquesta.

A la figura 18 podem observa la pantalla de creació d'enquesta, on l'únic camp necessari és el del nom de l'enquesta, camp amb el que identificarem l'enquesta. La figura 19 podem observar la pantalla de construcció de l'enquesta, a ressaltar que totes les accions disponibles a la aplicació per a la modificació de l'estructura de l'enquesta són accessibles des de aquesta pantalla.

Títol: **Política**
Descripció: ***Quina opinió tens dels partits polítics?***
Data Creació: ***2008-05-06 21:34:25.0***
Estat: ***Tancada*** *Obrir*
Acces: ***Privada*** *Establir acces públic*
Url: ***http://enquestes.servehttp.com/enquestes/accedirEnquesta.do?enquesta=6***

Editar enquesta | Borrar Enquesta | Nova Pàgina | Previsualitzar

Pàgina:
1.- PSOE
Descripció: ***Partit socialista obrer espanyol***
» Editar » Borrar » Previsualitzar

» Afegir Pregunta Aquí

1.- Quin és el polític més simpàtic?:
- Felipe Gonzalez
- Zapatero
- Montilla
-

» Editar » Eliminar » Afegir Pregunta Aquí

2.- Quines comunitats autónomes tenen bona salut política?:
- Catalunya
- Andalusia
- Extremadura

» Editar » Eliminar » Afegir Pregunta Aquí

» Afegir Pregunta

» Nova Pàgina

Pàgina:
2.- PP
Descripció: ***Parit Popular***
» Editar » Borrar » Previsualitzar

» Afegir Pregunta Aquí

1.- Creus que Rajoy hauria de donar pas a un altre candidat?: **(*)**
Si No

» Editar » Eliminar » Afegir Pregunta Aquí

2.- Quí creus que podría ser el millor relleu per Rajoy?:

» Editar » Eliminar » Afegir Pregunta Aquí

» Afegir Pregunta

» Nova Pàgina

Tornar

Fig. 19
Pantalla de construcció de l'enquesta.

2.3.2. Pàgina

Les enquestes estan formades per pàgines, en les quals es distribueixen les diferents preguntes que creen els *clients* i a les que responen els *usuaris*. No té sentit tindre una enquesta sense pàgines, així que automàticament, la creació d'una enquesta comporta la creació d'una pàgina associada a ella. L'existència de varies pàgines en una enquesta, ajuda a classificar les diferents preguntes contingudes a l'enquesta, així com a la visualització d'aquesta. Per tant, una pàgina te associades les següents característiques:

- *Enquesta* propietària
- Títol
- Descripció
- Ordre de la pàgina dins del test

Títol i **descripció** són cadenes de text que permeten al *usuari* tindre una primera impressió sobre el contingut de les preguntes que la pàgina en qüestió contingui. I l'**orde de la pàgina** és un sencer que situarà la *pàgina* dins del recorregut que faci l'*usuari* a l'hora de respondre l'*enquesta*. A més, el *client* creador de l'*enquesta* podrà modificar aquest recorregut (l'ordre en que el *usuari* respon a les preguntes de l'enquesta) canviant aquesta característica. Com podem observar en la figura 20, els camps que descriuen una pàgina són el títol i descripció.

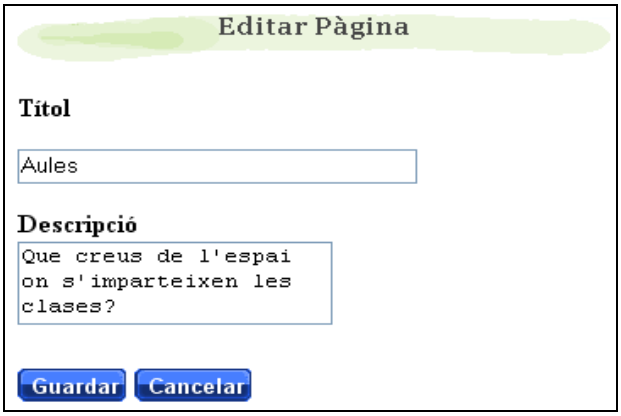


Fig. 20
Pantalla de edició de Pàgina.

2.3.3. Pregunta

Les pàgines, estan formades per preguntes, el nucli de l'aplicació, i per tant l'element més complex d'aquesta. Així que una pàgina estarà composta com a mínim per una pregunta sense que hi hagi un màxim d'aquestes. Hi han molts tipus de preguntes, però la propietat que els divideix d'una manera més absoluta és la seva finalitat. Tenim dos tipus de finalitats:

- Receptora de respostes

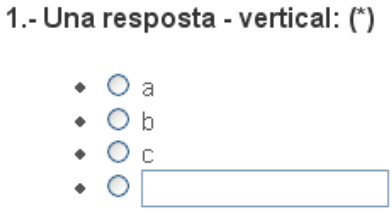
- De presentació (no te respostes associades)

Les preguntes ***De presentació*** no estan destinades a ser respostes, si no a presentar diferents tipus d'elements dins de la pàgina. Aquests tipus d'elements són:

- Text descriptiu
- Imatges
- Sons
- Separacions

D'aquesta manera el *client* té mes elements encarats a la presentació de l'enquesta. L'altre tipus de pregunta i que realment exerceix com a tal és divideix en un subconjunt més extens:

Tipus 1 i 2: Una resposta (presentació vertical i horitzontal)

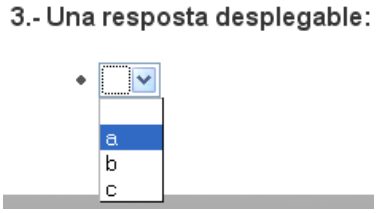


Tipus 1

Tipus 2

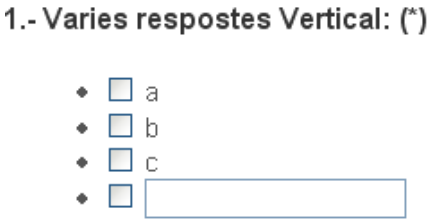
En les preguntes de Tipus 1 i 2, només una resposta és possible, raó per la qual les opcions apareixen precedides per un input tipus *radio*. Les respostes es presenten horitzontalment pel tipus 1, i verticalment pel tipus 2. A més com es veu a les imatges, es pot incloure una opció de resposta lliure.

Tipus 3: Una resposta (menú desplegable)



Pregunta d'una única resposta, aquesta vegada presentat com un *select*. Aquest tipus no presenta la opció de resposta lliure.

Tipus 4 i 5: Varies respostes (presentació vertical i horitzontal)

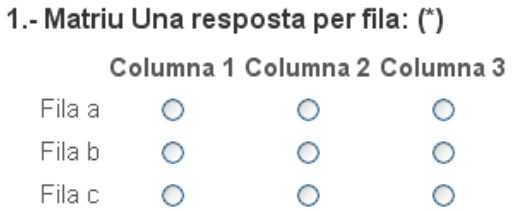


Tipus 4

Tipus 5

Similars a les preguntes tipus 1 i 2, però permetent múltiples respostes. La presentació de les respostes es realitza amb *checkbox* per a permetre les múltiples seleccions.

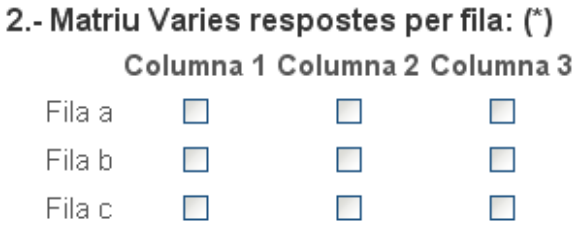
Tipus 6: Una resposta per fila (matriu)



Tipus 6

Les preguntes matricials responen a necessitats d'informació exhaustives, on no és només important conèixer la opinió d'un enquestat a una sola qüestió, si no conèixer-la en funció de varis condicionada. En el tipus 6, només podem escollir una opció per fila, raó per la qual es presenta en forma de "*radio buttons*".

Tipus 7: Varies respostes per fila (matriu)



Tipus 7

Nova pregunta matricial que es diferencia del tipus 6 en la possibilitat de realitzar seleccions múltiples. Totes les opcions de resposta poden ser seleccionades, raó per la qual la presentació es realitza en forma de *checkboxes*.

Tipus 8: Pregunta oberta (introducció d'un text).

Tipus 8

Les preguntes d'entrada lliure (no confondre amb l'opció de resposta lliure), plantegen una pregunta sense opcions de resposta a escollir i on és l'enquestat qui ha d'introduir manualment la seva resposta en un camp de text.

Tipus 9: Pregunta oberta amb 2 o més entrades (introducció de varis texts)

2.- Una o més línies: Resposta oberta:

Opció 1
Opció 2
Opció 3

Tipus 9

Similar al tipus 8, el tipus 9 ens dona a escollir el nombre de respostes lliures així com la seva temàtica que l'enquestat pot respondre.

Tipus 10: Varies respostes amb criteris (matriu de texts)

3.- Matriu: Resposta Oberta :

	Columna a	Columna b
Fila 1		
Fila 2		
Fila 3		

Tipus 10

El tipus de pregunta 10 és el més complex de tots, i mitjançant una matriu d'un nombre de files i columnes a escollir per l'enquestador, l'enquestat pot respondre a varies qüestions organitzades temàticament.

Amb tots aquest tipus de preguntes, el *client* té tota una amplia gamma per expressar la qüestió plantejada. A més, les respostes poden ser tant de tipus A/B (Si/No), com d'entrada d'opinió, amb una possibilitat o varies (és a dir, marcar més d'una resposta).

Per tant una pregunta estarà formada per les següents característiques:

- Pàgina propietària
- Text
- Tipus
- Ordre (posició dins de la pàgina)
- Pregunta obligatòria
- Opció "altres"
- Imatge
- Pes estadístic

Al ser continguda per una pàgina, un dels atributs que haurà d'emmagatzemar és la pàgina que la conté, gracies al camp **Pàgina propietària.** I si n'hi ha d'un camp més imprescindible que els altres, evidentment ha de ser el **text** de la pregunta.

Com veiem a les característiques, una pregunta pot ser presentada com de resposta obligada per l'*usuari* indicat per l'atribut **Pregunta obligatòria.** També tindrà l'*ordre* en el qual es presenta dins de la *pàgina*. L'opció **altres** donarà lloc, a que l'*usuari* pugui donar una resposta diferent a les plantejades, siguin aquestes d'introducció de text o no.

Per a la generació d'estadístiques, és molt probable que el *client* doni més importància a les respostes de certes preguntes més que d'altres, per a aquesta raó la característica **Pes estadístic** emmagatzemarà el factor de rellevància estadística.

De la mateixa manera que amb les **pàgines**, situarem les preguntes gracies a l'atribut **ordre**. Associant una **imatge** a la pregunta, podrem realitzar-ne del tipus: "¿Que opina d'aquesta imatge com a logotip de l'empresa?".

2.3.4. Resposta

Seguint amb la política d'elements "contenidors", una pregunta contindrà *respostes*, que es presentaran de diferents formes en funció del tipus de la pregunta. La manera en que el *client* les inserirà serà en la majoria de casos la mateixa, un text, que serà resposta en funció també del tipus de pregunta.

Però no nomes una pregunta pot contenir respostes, també una resposta pot contenir-hi. La raó es senzilla, quant una pregunta és de tipus matricial, necessitem un contenidor que contingui les preguntes d'una filera. Per tant, si tenim una matriu de 3x3, no tindrem 9 instàncies del objecte *resposta* relacionades amb la pregunta que les conté, si no 12. Nou de elles per a representar les 9 opcions possibles i 3 més per a contenir-les. Per tant, aquesta *pregunta* en concret contindria 3 *respostes*, i cada una d'aquestes en contindrien 3 més:

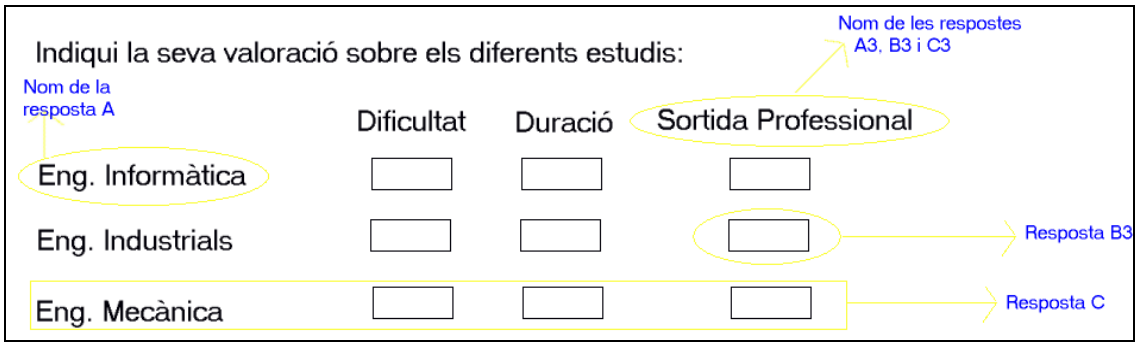


Fig. 21
Estructura Pregunta Matricial

A l'exemple que ens mostra la figura 21, podem veure una pregunta amb 3 respostes, que a la vegada contenen cadascuna 3 respostes més. És a dir, sigui la resposta A: *Eng. Informàtica*, aquesta conté 3 respostes més (com si d'un objecte Java es tractés) *Dificultat*, *Duració* i *Sortida Professional*. D'aquesta manera, el usuari actuarà sobre una de les 9 respostes a la vista, però també sobre les altres 3 que actuen

de *pares* (fet que dona més informació estadística a l'hora de processar les *opinions* de l'usuari).

Per tant una resposta estarà formada per les següents característiques:

- Pregunta propietària
- Resposta pare
- Ordre
- Text
- Imatge
- Es pare

Quant una *resposta* estigui continguda dins d'un altre, aquesta altre *resposta* la tindrem referenciada a la nostre instància gràcies al camp **Resposta pare** (quant això sigui així, tindrem una resposta d'una pregunta de tipus matriu). I sabrem si és, en canvi, pare d'alguna gracies al camp **Es pare**, actiu quant ho sigui. Encara que no és totalment necessària, aquest camp ens estalvia una pesada cerca per la base de dades.

Com a les *pàgines* o a les *preguntes*, les *respostes* també tenen un **ordre** per a situar-les dins de la *pregunta* que les conté. **Text** és la cadena de text que defineix la resposta. Aquesta a més pot anar acompanyada d'una imatge, o fins i tot no contenir **text** i tan sols la **imatge**. D'aquesta manera serà possible construir una pregunta del tipus "Quin logotip li sembla més apropiat per la campanya de Nadal?"), gracies al camp **Imatge**. Camp aquest que no serà una URL, sinó que físicament l'emmagatzemem a la base de dades.

2.3.5. Format

Al crear una *enquesta*, el *client* creador té l'opció d'escollir un format per a la presentació de la seva *enquesta*. Aquest *format* consistirà en triar una de les diverses opcions que permetrà l'aplicació que acabaran enllaçant un fitxer *css* dinàmicament quant es construeixi la pantalla de presentació. L'usuari no podrà crear el seu propi fitxer de *css*, però l'aplicació si que permetrà a l'administrador crear nous fitxers *css*, que al estar indexats en una taula, podran ser utilitzats per qualsevol client immediatament. La opció de canviar el format serà present a tota la vida de la *enquesta* doncs només modifica la seva presentació visual, sense produir cap canvi a qualsevol de les dades emmagatzemades i relacionades amb l'enquesta.

Per tant, un *format* te associades les següents característiques:

- Títol.
- Descripció.
- Fitxer css.

Títol i **descripció** són cadenes de text que permeten al *client* tindre una primera impressió sobre el tipus de presentació que adquirirà l'enquesta. *Fitxer css* serà el fitxer css (*Cascade Style Sheet*) que enllaçarà l'aplicació al crear la presentació.

Els formats que l'aplicació dona a escollir són cinc, els següents:

Tipus 1:

Nom: Pergamí.

Descripció: Presentació en pergamí.

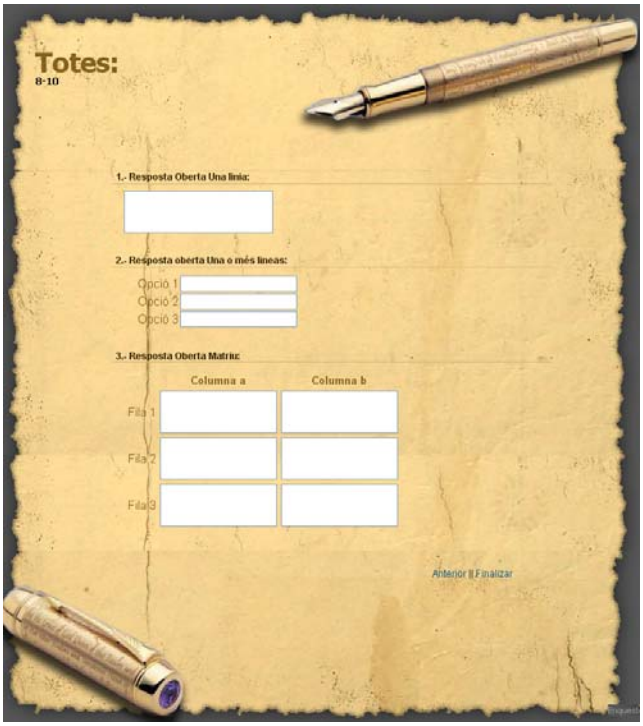


Fig. 22

Format Pergamí

Tipus 2:

Nom: Floral.

Descripció: Retocs naturals, sensació de camp.

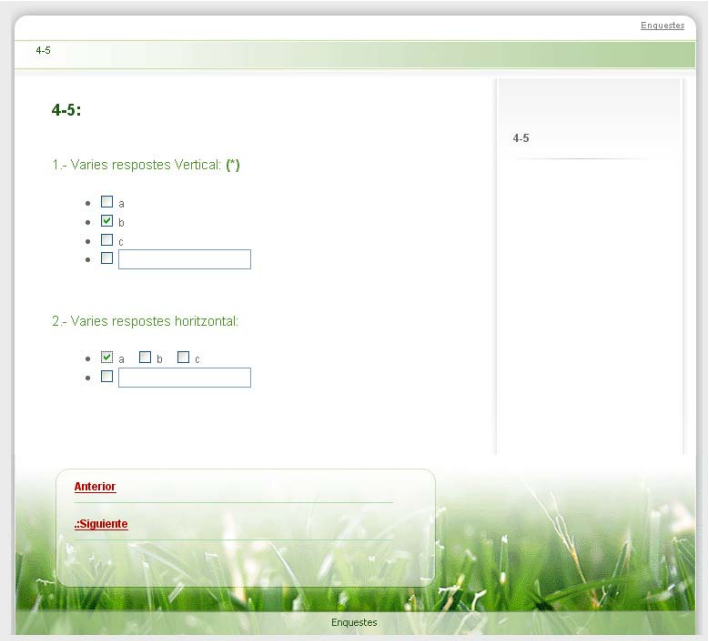


Fig. 23

Format Floral

Tipus 3:

Nom: Alternatiu.

Descripció: Dona un toc obscur a la teva enquesta.

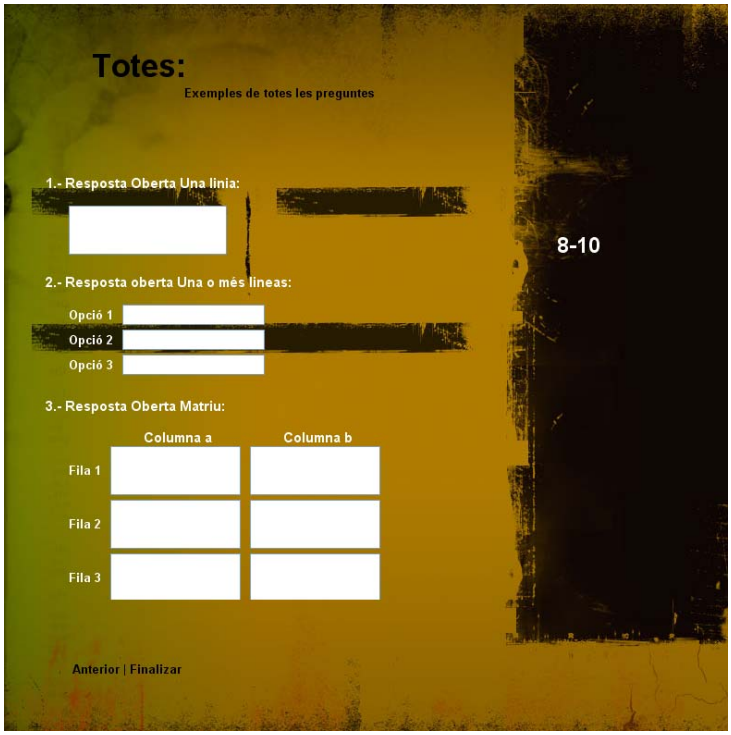


Fig. 24

Format Alternatiu

Tipus 4:

Nom: Convencional.

Descripció: Lleugera decoració per a acompanyar les teves enquestes:

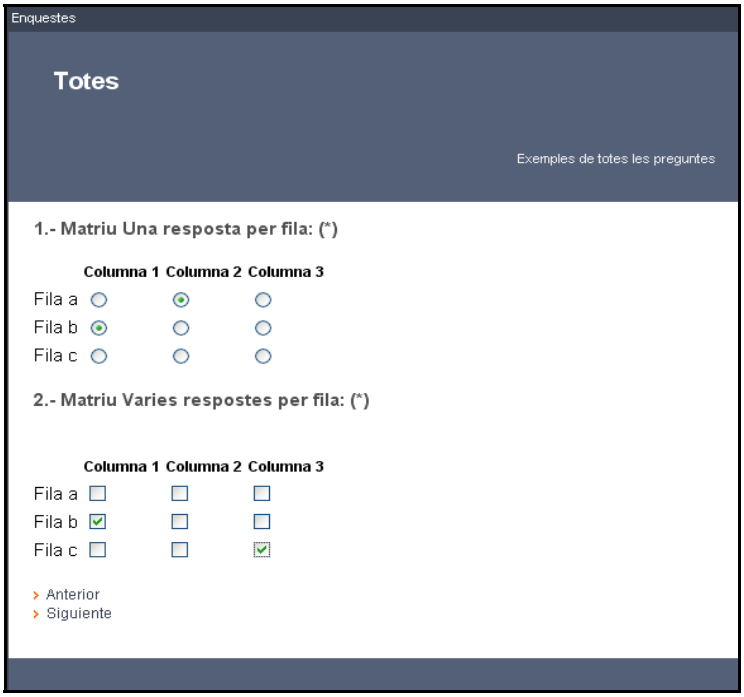


Fig. 26

Format convencional.

Tipus 5:

Nom: Senzill.

Descripció: Dona un retoc obscur a la teva enquesta.

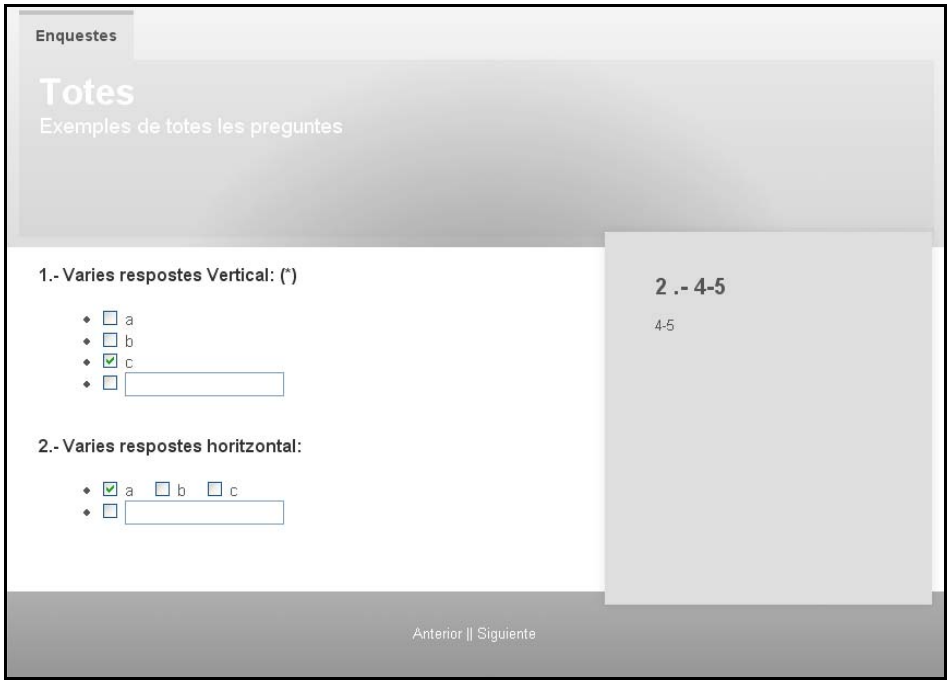


Fig. 27

Format senzill

2.4. Creació d'opinions

Les opinions representen les respostes donades per els enquestats a les enquestes, és a dir, és el retorn creat per el *usuari* a resposta del *client* de l'aplicació.

2.4.1. Accés

Cada vegada que un *usuari* respon una *enquesta*, es crea un objecte *acces*. Així que existiran tantes instàncies d'*acces* per a una *enquesta* en concret, com vegades algú hagi accedit a respondre-la. A més, aquest objecte ens servirà per a relacionar la resposta d'un *usuari* amb l'*enquesta* que la rep.

Per tant un *acces* té les següents característiques:

- Enquesta a la que es respon
- Usuari creador (l'*usuari* que ha accedit a l'*enquesta*)
- Data d'accés

Per a crear una instància de *acces* no es necessari ser un *usuari* registrat de l'aplicació, quant l'*enquesta* sigui pública, qualsevol visitant hi podrà accedir. Per tant el camp **usuari creador** sempre hi serà present, però en els casos que si hi sigui, això ens permetrà observar l'èxit obtingut d'un enviament. La informació referent al nombre

d'accessos rebuts en les nostres enquestes, es presentada en la barra d'eines, com podem observar en la figura 28:

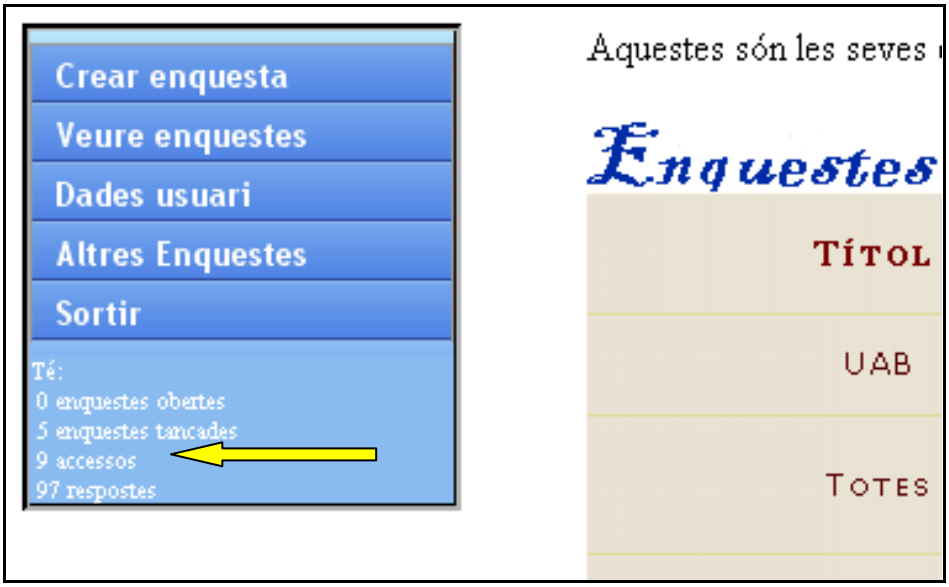


Fig. 28
La figura mostra el panell principal on l'enquestador pot veure el total d'accessos a les seves enquestes.

2.4.2. Opinió

Aquest és l'objecte que emmagatzemarà la resposta d'un *usuari* a una certa *pregunta* de l'*enquesta* a la que està accedint i estarà relacionada al objecte *acces* descrit anteriorment. Així com a mínim hi haurà una *opinió* per a cada *pregunta* obligatòria, sempre hi quant l'*usuari* hagi acabat de respondre a totes les preguntes de l'*enquesta*. Quant una pregunta sigui de resposta múltiple, es crearan tantes *opinions* com respostes hagi donat l'*usuari*, i seran els objectes que emmagatzemin l'entrada de text inserides com resposta.

Per tant una *opinió* té les següents característiques:

- Accés al que pertany
- Resposta referenciada
- Pregunta que conté la resposta
- Text introduït per l'usuari

Cal a dir, que no sempre trobarem text dintre d'una *opinió*, només quant la pregunta a la que es respon la sol·liciti, o bé quant pertanyi a un camp "otros". Cas aquest, en el que totes les dades d'una instància *opinio*, seran referències a altres objectes. La informació referent al nombre d'opinions rebudes en les nostres enquestes, es presentada en la barra d'eines, com podem observar en la figura 29:

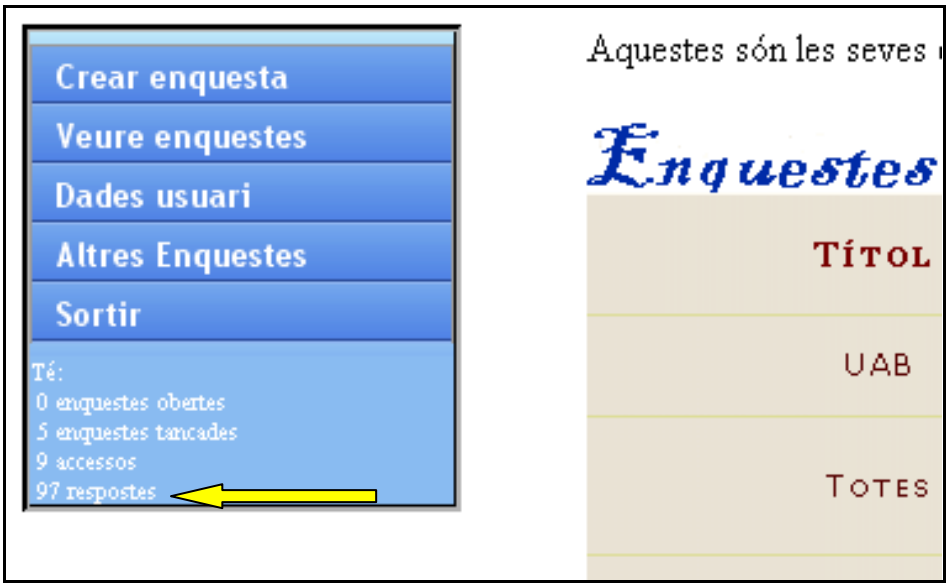


Fig. 29
La figura mostra el panell principal on l'enquestador pot veure el total de respostes que han rebut les seves enquestes.

2.5. Generació d'estadístiques i resultats

Per a generar estadístiques sobre una *enquesta*, és necessari que aquesta hagi rebut *opinions*. Si ens fixem en altres aplicacions de generacions d'enquestes com www.surveymonkey.com, veurem que qualsevol *client* pot generar enquestes lliurement, però que és la recepció de respostes (o la possibilitat de que aquestes es puguin generar) la part on és necessari una contractació prèvia. És el resultat i objectiu de qualsevol enquesta, la recepció de opinions o respostes.

Podem visualitzar les estadístiques com:

- Gràfics
- Taules de dades absolutes
- CSV

Sigui quin sigui el mode de representació escollit, el format de les dades dependrà absolutament del tipus de la pregunta a la qual s'està opinant. Amb el mode gràfic podem accedir a tota una gamma de gràfics de diferent tipus que ens permet observar les dades en funció de diferents variables. Les taules ens donen una visió més absoluta de les *opinions* obtingudes. I per últim, amb la modalitat de CSV (*Comma Separated Values*) obtindrem un fitxer amb una sèrie de característiques sobre les *opinions* així com dels *usuaris* creadors.

A la figura 31 podem observar l'informe de resultats de l'enquesta *UAB*. Aquest informe està compost per una opció d'impressió, una altre de previsualització i quadre de respostes obtingudes per cada pregunta de l'enquesta. Quant les respostes no són d'entrada lliure (l'usuari ha introduït una cadena de text) aquests quadres de respostes

mostren el nombre total de respostes rebudes dels usuaris, així com el tant per cent del total. A la part final del requadre podem veure el total de respostes rebudes a la pregunta (la suma dels quadres anteriors), i el nombre total d'accessos. En canvi, quan la resposta és d'entrada lliure també es podrà accedir al llistat de respostes mitjançant el link: Veure detall , donant lloc a la pantalla mostrada en la figura 30. En les respostes de preguntes matricials aquesta opció apareixerà només la lupa.

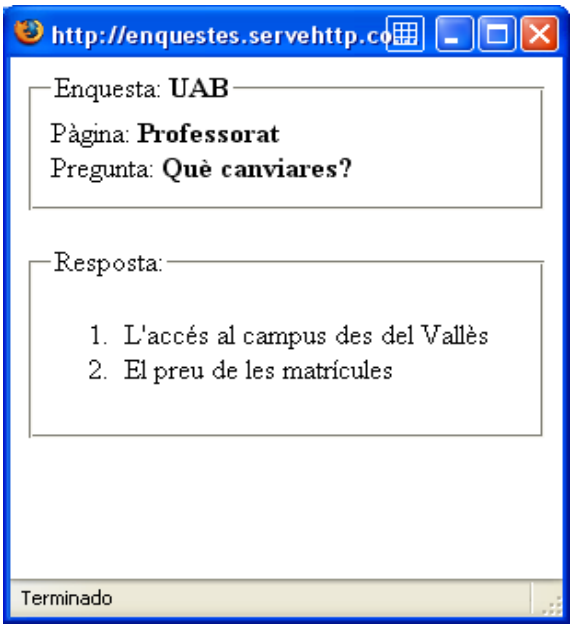


Fig. 30
Diàleg per mostrar les respostes de lliure entrada

Les opcions de visualització de gràfics, accessibles mitjançant el link de les figures i , mostraran un gràfic circular o de barres respectivament com es pot observar en les figures 32 i 33.

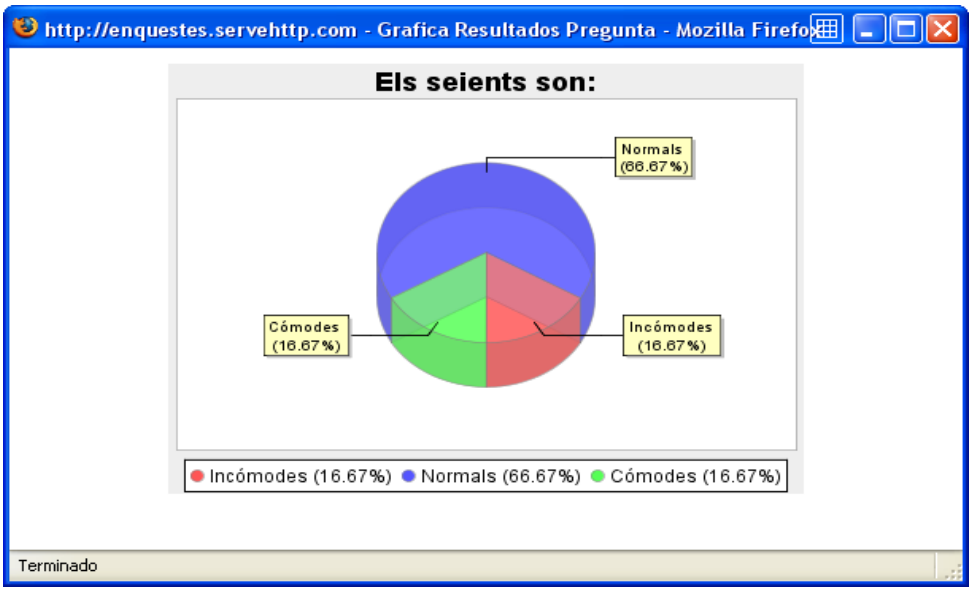


Fig. 32
Gràfic circular.

Informe de resultats: UAB

Imprimir

Pàgina: Aules

1. Els seients son:

		Percentatge	Total Respostes
Cómodes		25%	1
Normals		50%	2
Incómodes		25%	1
	Total respostes		4

2. El volúm d'alumnes per classe es

		Percentatge	Total Respostes
Baix		25%	1
Mitjà		75%	3
Alt		0%	0
Insostenible		0%	0
	Total respostes		4

3. A cuantes aules diferents acudeixes normalment?

		Percentatge	Total Respostes
1		25%	1
2		50%	2
3		25%	1
Més de 3		0%	0
	Total respostes		4

4. Que creus del material multimèdia de les aules?

Total respostes	Veure detall	2

Pàgina: Professorat

1. Que canviaries?

Total respostes	Veure detall	4

Fig. 31

Informe de resultats: Taula de dades absolutes.

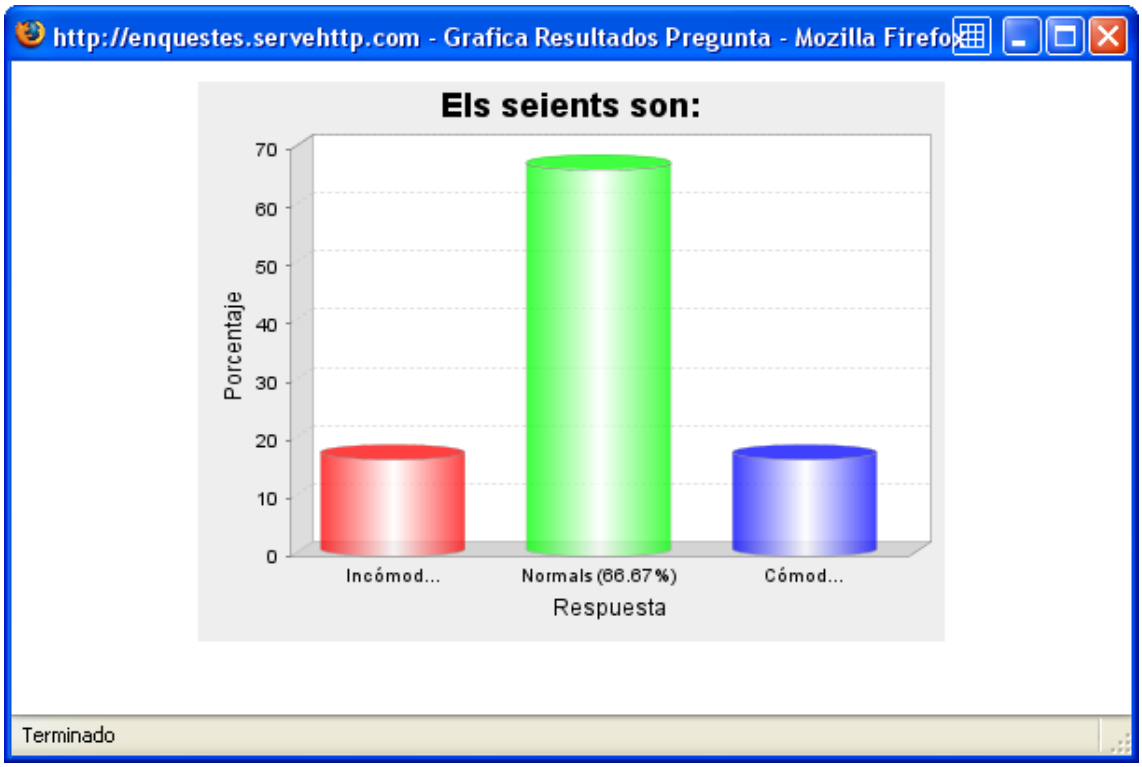


Fig. 33
Gràfic de barres.

3. Arquitectura

3.1. Descripció general

Com ja hem dit, *enquestes* és una aplicació web, tret que permet el seu ús en qualsevol dispositiu que tingui un navegador així com connexió a Internet o a Intranet, la seva estructura general la podem observar en la figura 34. L'accés d'una aplicació es realitza mitjançant el protocol TCP/IP, indistintament sigui aquest a través de Internet o una Intranet. La figura ens mostra les parts bàsiques de la interacció entre el usuari, que només interactua amb un navegador que pot residir en múltiples suports, com ordinadors de sobretaula, portàtils, terminals, pda's o fins i tot telèfons mòbils. Posteriorment, aquest navegador és comunica pel port 80 amb el servidor de la aplicació mitjançant Internet o Intranet, creant així una comunicació bidireccional de dades.

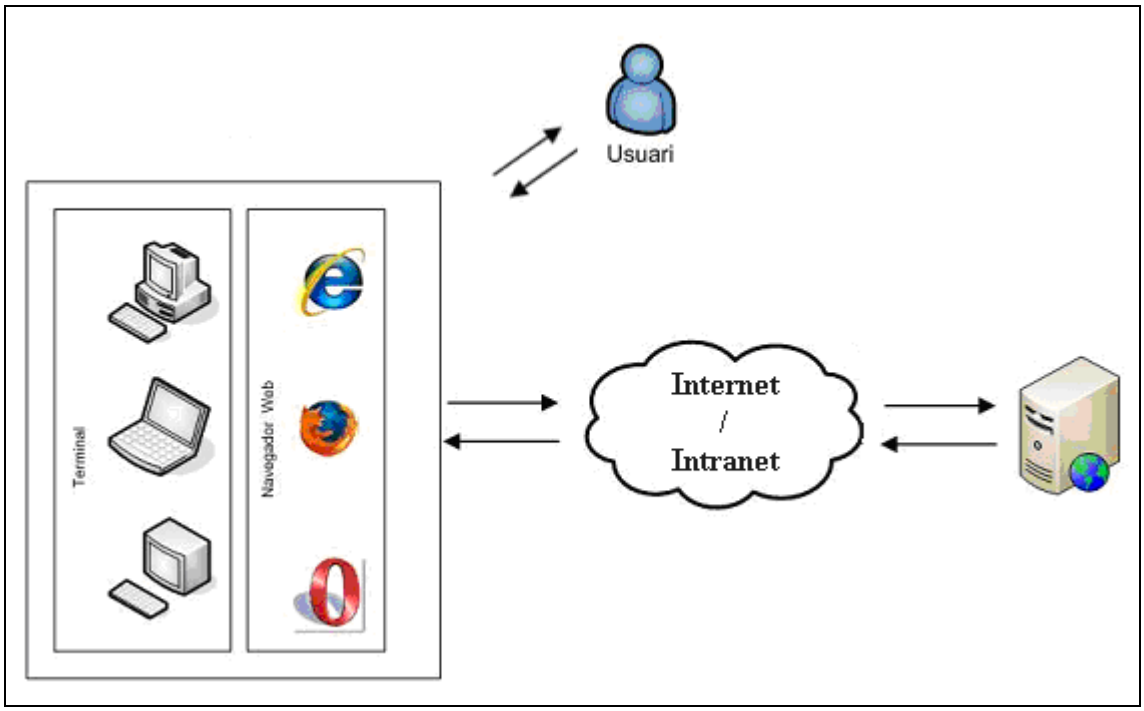


Fig. 34
Estructura bàsica d'una aplicació web

Els trets principals que diferencia una aplicació web a una convencional no és l'accés per web o pel navegador, sinó perquè una de web no necessita cap mena de instal·lació al PC client. Algunes aplicacions web si necessiten complements, com Flash Player o el Java Runtime de Sun, encara que no en *Enquestes*, on totes les dades que rep el client son en format Html.

En les aplicacions web ens podem trobar documents estàtics o bé dinàmic. Un document estàtic és aquell que sempre viatja del servidor en la mateixa estructura i dimensió (mai varia). Mentre que un document dinàmic és aquell que depèn de les entrades del client en forma de formularis o paràmetres POST/GET, així com del contingut de la base de dades relacionat amb l'usuari. Els documents que envia *Enquestes* són dinàmics en la seva totalitat.

Això pel que fa referència al client, en quant al servidor, *enquestes* té una arquitectura similar a la de la gran majoria de solucions web comercials. A més, la seva estructura modular permet adaptar-se a diferents plataformes i configuracions. Un servidor d'aplicacions, *Apache Tomcat* en aquest cas, atén la sol·licitud HTTP generada pel navegador del client. Remarcar en aquest punt, que qualsevol servidor preparat per aplicacions Java, podria servir per a substituir a la solució utilitzada en *enquestes*.

3.1.1. J2EE

Enquestes és desenvolupat en tecnologies Java web, com moltes altres aplicacions web en Internet on bàsicament complementen la oferta amb aquelles realitzades en tecnologies Microsoft, aplicacions J2EE o .Net respectivament.

Fig. 35
Logotips Sun i Java.

Java Platform, Enterprise Edition o Java EE [16] (anteriorment conegut com a Java 2 Platform, Enterprise Edition o J2EE fins la versió 1.4), es una plataforma de programació creada per Sun MicroSystems (figura 35)- part de la Plataforma Java - per a desenvolupar i executar software de aplicacions en Llenguatge de programació Java amb arquitectura de *n* nivells distribuïda, basant-se àmpliament en components de software modulars executant-les sobre un servidor d'aplicacions.

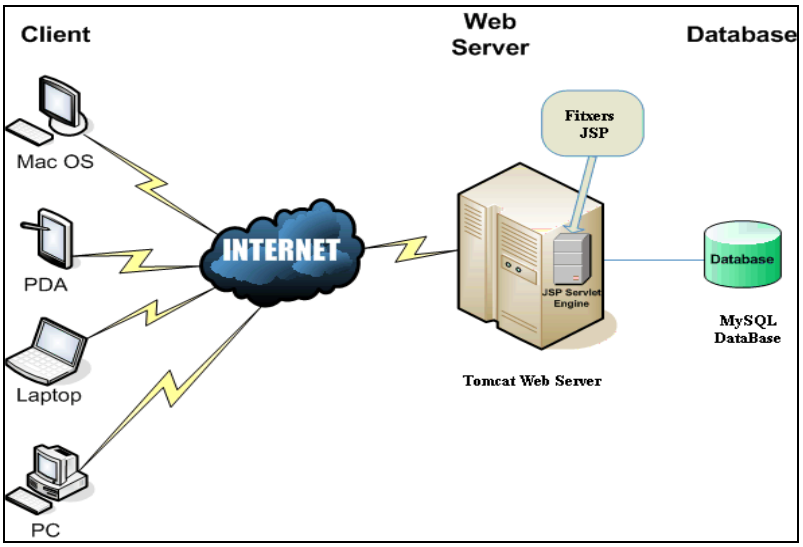


Fig. 36
Estructura general d'una aplicació web J2EE.

La plataforma Java EE està definida per una *especificació*. Similar a d'altres especificacions del *Java Community Process*, Java EE és també considerada informalment com a un estàndard *de facto* degut a que els subministradors han de complir amb certs requisits de conformitat per a declarar que els seus productes com a *conformes* amb Java EE.

A la figura 36 podem observar la estructura general d'una aplicació J2EE, on mitjançant un navegador, el client accedeix als recursos existents en el servidor d'aplicacions.

Java EE inclou varies especificacions de API, tals com *JDBC*, *RMI*, *e-mail*, *JMS*, Serveis Web, *XML*, etc així com defineix com coordinar-los. Java EE també configura algunes especificacions úniques Java EE per a components. Aquestes inclouen *Enterprise JavaBeans*, *servlets*, *portlets*, *JavaServer Pages* i varies tecnologies de serveis web. Això permet al desenvolupador crear una Aplicació de Empresa portable entre plataformes i escalable, a la vegada que integrable amb d'altres tecnologies.

D'altres beneficis afegits són, per exemple, que el servidor de aplicacions pot manegar transaccions, la seguretat, escalabilitat, concurrència i gestió dels components desplegats, significant que els desenvolupadors poden concentrar-se més en la lògica de negoci dels components en lloc de fer-ho en tasques de manteniment de baix nivell.

Juntament amb *J2EE* trobem les *JSPs*, acrònim de ***JavaServer Pages***. *JavaServer Pages* [17] és una tecnologia que permet als desenvolupadors de pàgines web, generar respostes dinàmicament a peticions HTTP. La tecnologia permet que codi Java i certes accions predefinides siguin incrustades en un context estàtic.

La sintaxi de JSP incorpora tags XML addicionals, anomenats accions de JSP, per ser usats per invocar altres funcions. Addicionalment, la tecnologia permet la creació de llibreries d'etiquetes que actuen com extensions de l'estàndard d'etiquetes *HTML* o *XML*. Les llibreries d'etiquetes aporten una forma multi plataforma d'ampliar les capacitats d'un servidor web. Els JSPs són compilats en Servlets per un compilador JSP. Aquest pot generar un servlet o generar bytecode directament.

Des del punt de vista de l'arquitectura, els JSP poden ser vistos com una abstracció, d'alt nivell, de servlets implementada com una extensió de l'API Servlet 2.1. Ambdues tecnologies van ser inicialment desenvolupades per Sun Microsystems. Començant per la versió 1.2 de l'especificació de JSP, JavaServer Pages van ser desenvolupats sota la *Java Community Process*. La JSR 53 defineix tant les especificacions 1.2 com la 2.3 i JSR 152 defineix la JSP 2.0. El maig de 2006 l'especificació 2.1 va ser llençada sota la JSR 245 com a part de Java EE 5.

Una pàgina JSP està dividida en les parts següents:

- dades estàtiques, com ara HTML o XML (no es recomana que els JSP generin respostes binàries)
- directives JSP, per exemple, els includes
- elements de scripting i variables

- accions JSP
- tags personalitzats

3.2. Model

En quant a la estructura que processa i respon a la petició HTTP, *enquestes* utilitza el framework ***Struts*** [9], que és un framework que implementa la arquitectura MVC (*Model-Vista-Controlador*) de tipus 2.

Fig. 37
Logotip Apache Foundation.

Struts és una eina per a la implementació d'aplicacions Web sota la plataforma **J2EE,** i es desenvolupa com a part del projecte *Jakarta* de la Apache Software Fundation (figura 37), encara que ara es un projecte independent conegut com *Apache Struts*. Gràcies al ús de *Struts***,** les aplicacions basades en el seu ús veuen com el temps necessari per al seu desenvolupament es retalla substancialment.

Struts implementa una capa intermèdia entre la aplicació i les peticions HTTP en forma de controlador que es qui determina el flux de la aplicació. *Struts* deriva la acció a una o d'altres parts de la aplicació en funció de les dades que rep del usuari contingut en la petició HTTP en forma de formularis, i quant aquestes han pogut dur a terme el seu propòsits, *Struts* avalua el resultat determinant el següent pas. És a dir, *Struts* implementa un controlador d'alt nivell per on passen totes les accions que es succeeixen entre la aplicació i el client. A més, també porta a terme la validació de formularis *on-server*, és a dir la correctesa de les dades no és examinada en el costat client mitjançant tecnologies secundàries com Javascript, sinó que són avaluades en el servidor abans de ser aquestes tractades per la acció específica.

Per tant, *Struts* representa una eina de suport per al desenvolupament d'aplicacions Web sota el patró MVC dins la plataforma J2EE (Java 2, Enterprise Edition). Struts es desenvolupa com a apart del projecte Jakarta de la Apache Software Foundation, encara que a hores d'ara es un projecte independent conegut com a Apache Struts.

Struts permet reduir el temps de desenvolupament. A més, el seu caràcter *open source* (software lliure) així com la seva compatibilitat amb totes les plataformes en les que Java Enterprise estigui disponible, el converteix en una eina amb un gran grau de disponibilitat.

Struts es basa en el patró MVC, Model-Vista-Controlador. Aquest és utilitzat àmpliament i es considerat com a base de solidesa. D'acord amb aquest model, el processament es separa en tres direccions, el model, la vista i el controlador.

Quant es programen aplicacions Web amb el patró MVC, sol aparèixer un *fat controller*, un controlador sobrecarregat amb la lògica de la aplicació, que acaba controlant totes les peticions. Amb Struts es troba solució a aquest problema, doncs aquest implementa un únic controlador dedicat a rebre les peticions del usuari mitjançant un fitxer con figurable (*struts-config.xml)*.

Normalment aquest model implementa accessos a Bases de Dades, o sistemes que funcionen independentment de la aplicació web. En el nostre cas, els components de model implementen una capa de abstracció de la base de dades mitjançant ***Hibernate***.

Els components de controls son els encarregats de coordinar les activitats de la aplicació, que van des de la recepció de dades del usuari, les verificacions de forma i la selecció de un component del model a ser cridat. Per la seva part els components del model envien al control els seus eventuals resultats o errors, per poder continuar així amb altres fases de la aplicació.

Aquesta separació simplifica enormement l'escriptura tant de vistes com de components del model: les pàgines JSP (Java Server Pages) no han d'incloure el control d'errors, mentre que els elements de control simplement decideixen el pas següent.

Aquestes són algunes de les característiques de *Struts* més importants:

- ✓ Configuració de control centralitzada.
- ✓ Interrelacions entre accions i pàgina o d'altres accions s'especifiquen per a taules XML en comptes de codificar-les en l'aplicació o pàgines.
- ✓ Components de la aplicació, que són el mecanisme per a compartir informació bi-direccionalment entre els usuaris de la aplicació i les accions del model.
- ✓ Llibreries de entitats per a facilitar la majoria de les operacions que generalment realitzen les pàgines JSP.
- ✓ Conté eines per a la validació de camps de plantilles sota varis esquemes que van des de les validacions locals en la pàgina (amb javascript) fins les validacions de fons realitzades en el costat del servidor.

Struts permet que el desenvolupador es concentri en el disseny de aplicacions complexes com una sèrie simple de components del Model i de la vista intercomunicats per un control centralitzat. Dissenyant aplicacions d'aquesta manera pot obtenir-se una aplicació més consistent i fàcil de mantenir.

A la arquitectura **MVC** de tipus 2, les diferents pàgines que l'usuari veu llencen accions utilitzant un únic controlador que s'encarrega d'atendre les diferents peticions a un conjunt d'accions prèviament registrades al controlador. Un únic servlet és cridat des de el client, romanent així visible un únic punt d'entrada al controlador. Aquesta arquitectura contrasta amb el tipus 1, on la lògica de control de la aplicació ve integrada a la capa de presentació. Aquesta arquitectura es la utilitzada en ASP.NET i en JSF (Java Server Faces). En aquest cas l'usuari genera esdeveniments que son tractats al controlador, associant accions als diferents esdeveniments.

El patró **MVC** és un patró de arquitectura de software que separa les dades de una aplicació, la interfície de usuari i la lògica de control en tres components diferents.

El patró **MVC** es molt sovint trobat en solucions web, on la vista és la pròpia pàgina HTML. El model és el sistema de gestió de bases de dades i el controlador representa la lògica de negoci. Més en profunditat podem definir les parts en que és compon el model MVC com:

- **Vista:** Representació del model en un format adequat per a interactuar. A *enquestes* no tenim pàgines pures de HTML, sinó pàgines **JSP**, que combinen lògica amb elements HTML així com inclouen fitxers *CSS* (*cascade style sheet*). Una interfície de usuari *friendly* ajuda a la millor navegació per l'aplicació.

- **Model:** Aquesta és la representació específica de la informació amb la que el sistema opera. La lògica de dades assegura la integritat de aquests i permet derivar noves dades; per exemple, no permetin-te realitzar una pregunta sense respostes associades. *Enquestes* implementa una abstracció de la base de dades mitjançant l'eina de mapatge objecte-relacional ***Hibernate***, amb la que accedim a la base de dades ***MySql*** com si de objectes i no pas taules es tractessin.

- **Controlador:** Aquest respon a esdeveniments, normalment accions del usuari així com invoca canvis en el model i la vista.

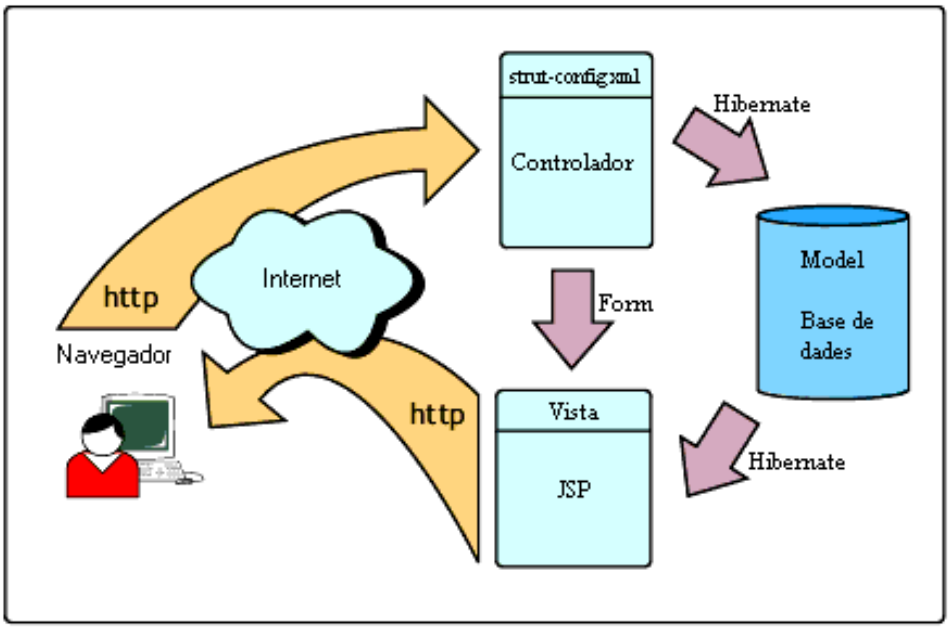


Fig. 38
Model MVC enquestes.

A la figura 38, podem observar els components del patró **MVC**, on la funció controladora la realitza *Struts*, derivant el flux dins de la aplicació. La funció model resideix en les dades emmagatzemades a la base de dades i posteriorment encapsulades en objectes Java. Finalment, la funció vista és implementada per els fitxers JSP, que representen visualment les dades rebudes del client.

Així doncs, a mode gràfic, podem observar l'estructura bàsica de ***enquestes*** en la següent figura 39:

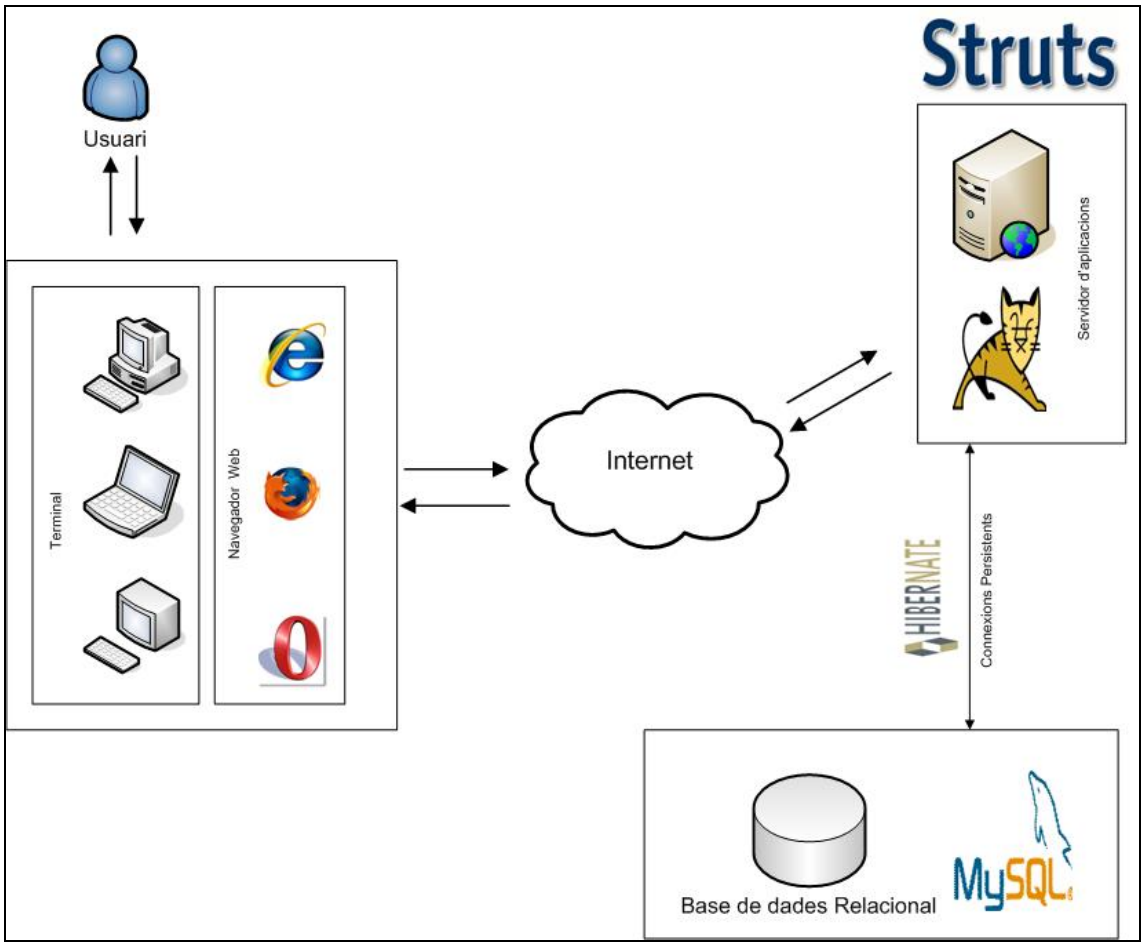


Fig. 39
Estructura Enquestes.

A l'anterior imatge s'observa la estructura bàsica de *Enquestes*, que segueix l'estàndard de moltes aplicacions web J2EE actuals. L'usuari accedeix mitjançant un navegador web (*Enquestes* està optimitzat per *Mozilla Firefox*), al servidor d'aplicacions *Apache Tomcat*, que retorna el *site* de *Enquestes*. També es pot observar la connexió persistent mitjançant *Hibernate*, de l'aplicació a la base de dades relacional *MySql* (està en el mateix servidor).

3.3. Abstracció de la base de dades mitjançant Hibernate

Per a realitzar la connexió persistent a la base de dades, *Enquestes* fa servir ***Hibernate*** [11], que actua com una capa intermèdia entre la aplicació i la base de dades, com es pot observar en la figura 40 . ***Hibernate*** és una solució implementada pel mapatge d'objectes relacionals (ORM) per aplicacions Java, sobre una base de dades relacional. Els seus propòsits bàsics són els d'alliberar el programador d'un seguit de tasques pròpies de la persistència de dades relacionals i dotar les aplicacions de portabilitat entre SGBDs diferents.

Seguint amb el caràcter general de *Enquestes*, Hibernate és lliure, de codi obert i està distribuït sota la GNU Lesser General Public License.

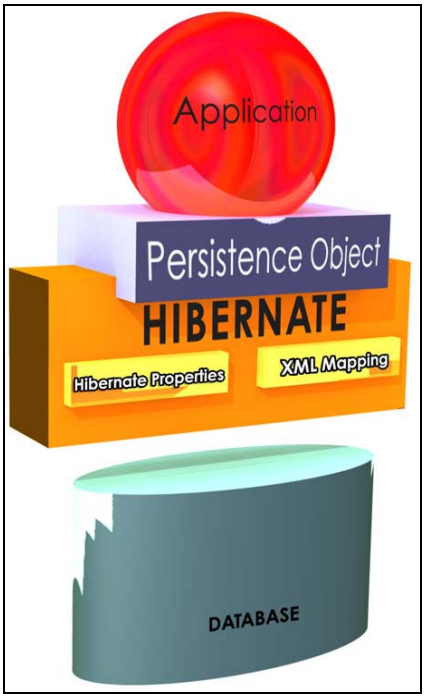


Fig. 40
Hibernate actua entre la base de dades i l'aplicació.

La funcionalitat bàsica de Hibernate és la del mapatge de classes Java en taules de Base de Dades i tipus de dades Java sobre tipus de dades d'SQL. Hibernate també proveeix un llenguatge de query (HQL) i facilitats per la recuperació de dades. Hibernate genera les crides SQL i delega al desenvolupador la gestió manual del resultat de la query i la conversió a objectes. Gràcies a això una aplicació pot ser portable a la majoria de bases de dades SQL, amb mínima càrrega addicional. El requeriment necessari perquè una aplicació basada en Hibernate pugui usar una base dades SQL és que existeixi la peça de software (driver) encarregada de la traducció d'HQL al dialecte SQL del SGBD sobre el que ha de córrer l'aplicació.

Hibernate proveeix persistència per "Plain Old Java Objects"; l'únic requeriment estricte per una classe persistent és el d'oferir un constructor públic i sense arguments. Hibernate pot ser usat tant en aplicacions "standalone" com en aplicacions Java EE que usen servlets o EJB session beans.

Gràcies a la ràpida expansió del seu ús, aquesta única implementació i la gran quantitat de desenvolupadors amb formació avançada en l'eina i documentació existent a Hibernate, són un complement molt potent al seu robust funcionament.

3.4. El servidor

El servidor d'aplicacions és un *Apache Tomcat*. Apache Tomcat [13] és un contenidor de servlets desenvolupat a l'Apache Software Foundation. Tomcat implementa les especificacions de servlet i de Java Server Pages (JSP) de Sun Microsystems, proporcionant un entorn per al codi Java a executar en cooperació amb un servidor web. Aquest afegeix eines per a la configuració i el manteniment, però també pot ser configurat editant els fitxers de configuració que normalment són en format XML. Tomcat inclou el seu propi servidor HTTP, per això també se'l considera un servidor web independent.

Fig. 41
Logotip Apache Tomcat.

Apache Tomcat, figura 41, es desenvolupat en un moviment de participació obert i distribuït sota la llicència Apache Software. Apache es promogut com a resultat del treball dels millors desenvolupadors d'arreu del món, aspecte que fa que sigui present en molts dels més ambiciosos projectes existents a l'actualitat.

Així doncs Tomcat és pot definir cim a un servidor web amb suport de servlets i JSPs. Inclou el compilador Jasper, que compila JSPs convertint-los en servlets. El motor de servlets de Tomcat sovint es presenta en combinació amb el servidor web Apache o altres servidors web. En els seus inicis, va existir la percepció que l'ús de Tomcat de forma autònoma era només recomanable per a entorns de desenvolupament i entorns amb requisits mínims de velocitat i gestió de transaccions. Avui dia ja no existeix aquesta percepció, i Tomcat és usat com servidor web autònom en entorns amb alt nivell de tràfic i alta disponibilitat.

Ja que Tomcat va ser escrit en Java, això fa que funcioni en qualsevol sistema operatiu que disposi de la JVM.

Un exemple de la importància d'aquest projecte el trobem a [8], l'agència catalana de notícies, que utilitza Tomcat com a servidor Web.

4. Disseny i implementació

4.1. Descripció

Enquestes segueix els estàndards J2EE, essent *Struts* la part controladora de l'aplicació.

4.1.1. Struts

L'aplicació segueix estrictament els estàndards *Struts* J2EE [12], on tot gira al voltant de l'arxiu de configuració *struts-config.xml.* Aquest fitxer XML, està compost majoritàriament per tags *action*. Un action defineix el path HTTP relatiu a la aplicació (l'extensió en la URL juntament amb el nom del servidor i de la aplicació), el formulari rebut, l'abast, l'acció a realitzar i els camins a seguir en el flux en funció del resultat.

Com hem dit, en el framework de *Struts*, el fitxer ***struts-config.xml*** defineix la lògica de la presentació i navegació de la aplicació. En aquest fitxer estan registrats:

- Beans de formulari, que son les dades que envia cada acció del usuari al controlador.
- Accions del formulari, que són les accions que poden ser cridades des de les pàgines del client. Cada acció rep informació mitjançant els *beans* del formulari i altres beans definits a la aplicació i respon al usuari generant diferents vistes mitjançant *JSP*. És en les accions on s'escriu la lògica de negoci i la gestió del model de dades així com la seva persistència.
- Regles de navegació per les diferents pàgines, mitjançant els mapatge d'accions (*action mappings*).
- Regles de validació de dades dels formularis, que permeten tant la validació al client con en el servidor.
- Regles de filtre de les peticions mitjançant cadenes de filtres.

El flux que segueix el control:

1. L'usuari interactua amb l' interfície d'usuari d'alguna manera (per exemple pitjant un link).
2. El controlador rep la notificació de l'acció sol·licitada per l'usuari. El controlador gestiona l'esdeveniment que arriba, a través del gestor de esdeveniments del framework ***Struts*** (en el fitxer ***struts-config.xml*** es troben les relacions esdeveniment-controlador).
3. El controlador accedeix al model, actualitzant-lo/modificant-lo en funció de l'acció sol·licitada per l'usuari. En aquest pas interactuen els ***beans*** de Java amb la persistència de objectes que proporciona ***Hibernate***.
4. El controlador delega als objectes de la vista la tasca de desplegar la interfície de usuari. La vista obté les seves dades del model per a generar la interfície apropiada per al usuari on es reflexa els canvis en el model (per exemple, el llistat de respostes associades a una determinada pregunta). Per a implementar l'estructura de dades que la vista rep, s'utilitzen els ***ValidatorForms*** de Java,

que permeten la validació al servidor dels diferents camps. El model no ha de tenir coneixement associat a la vista.

5. La interfície de usuari espera noves indicacions del usuari, començant el cicle novament.

Un exemple per detallar-ho millor:

```
<action
      path="/login"
      name="LoginForm"
      scope="request"
      type="com.enquestes.action.LoginAction">
      <forward name="success" path="/entrarMenu.do" />
      <forward name="error" path="/index.do" />
</action>
```

Aquest és un *tag action* del fitxer *struts-config.xml*, on:

- **Path:** És l'extensió URL, per exemple, si la aplicació estigués ubicada en un servidor nombrat ***www.enquestes.cat***, la url completa per a accedir al *path* login seria: http://www.enquestes.cat/login.
- **Name:** Nom del formulari adjunt a la petició HTTP i que correspon a una classe Java, en aquest cas la classe és LoginForm, compost per un nom d'usuari i una contrasenya, que és encapsulat en la classe Java Bean que finalment serà la que rebi la acció.
- **Scope**: Identificador de l'abast del bean, on els valor possibles son *request* (la vida del bean és la petició HTTP) o *session* (la vida del bean dura tota la sessió).
- **Type:** La acció a executar per a aquesta entrada. En aquest cas el flux executaria la acció *LoginAction* que seria la qui rebria el bean *LoginForm* i generaria el *forward*, el resultat.
- **Tag forward:** Una acció retorna una resposta al controlador S*truts* en forma de ***ActionForward***, que és la cadena de text que serveix al controlador per derivar el flux a una o d'altre acció. En aquest cas si *LoginAction* realitzés la seva funció (logar un usuari en la aplicació), la següent acció a realitzar seria *entrarMenu*, que correspon a una altre acció *Struts*.

4.1.2. Estructura d'arxius

La aplicació ha estat desenvolupada mitjançant el IDE (*Integrated Development Environment*) ***Eclipse*** [14]. ***Eclipse*** és un entorn de desenvolupament integrat de codi obert independent de la plataforma, i és el més utilitzat per a desenvolupaments Java a nivell professional.

Com es pot veure en la figura 42, **Eclipse** permet editar diferents tipus de formats alhora, com ara els fitxer de configuració *struts-config.xml*, les accions Java, els arxius de vista JSP o els de mapatge *Hibernate.*

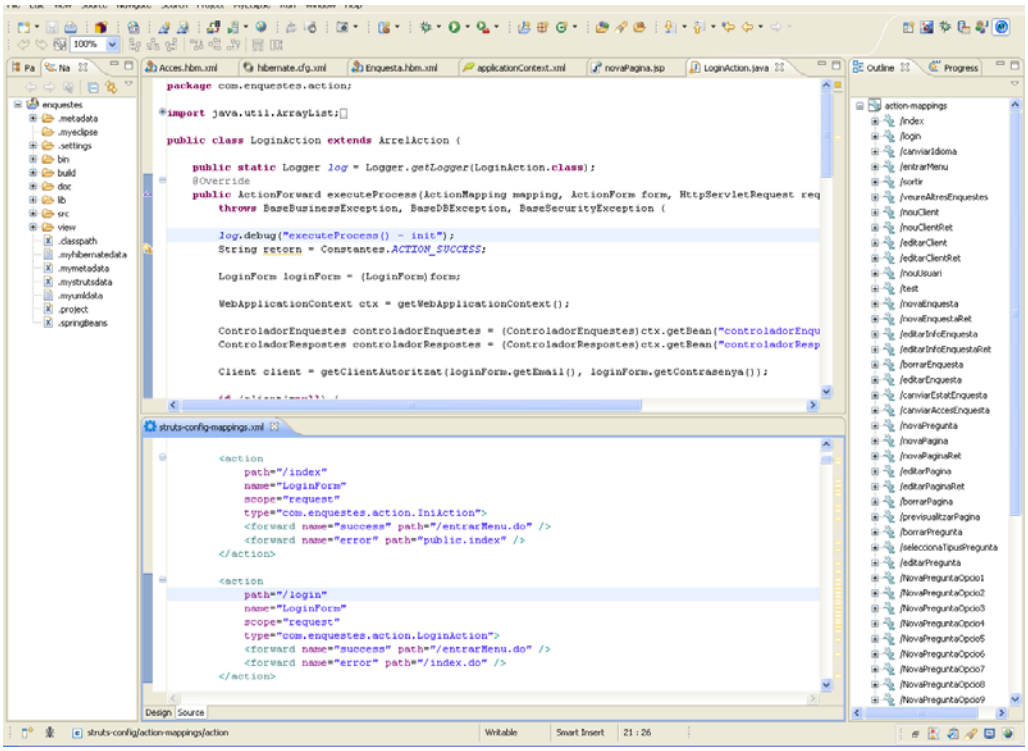


Fig. 42
Captura del IDE Eclipse.

La figura 43, ens mostra l'estructura interna de les carpetes:

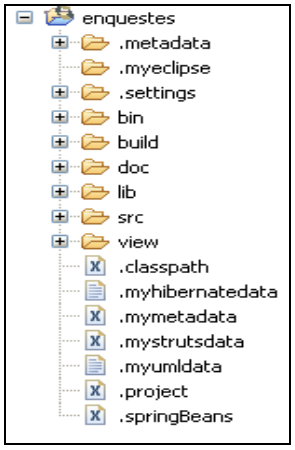


Fig. 43
Estructura fitxers.

bin: És la carpeta on romanen els binaris de la aplicació.
build: En aquesta carpeta trobem els scripts necessaris per a construir la base de dades, així com l'ant necessari per a realitzar el *deployment*.
doc: Fitxers de documentació de la aplicació.
src: Aquesta carpeta conté les classes Java que utilitza *enquestes*.
view: Aquesta és la carpeta principal de la aplicació Web. En aquesta carpeta s'han creat les següents carpetes:

META-INF: On resideix el fitxer *context.xml*, necessari per al servidor d'aplicacions.

WEB-INF: Aquesta és la carpeta que conté la informació interna del servidor Web. Aquesta carpeta té una gestió especial dins del *Tomcat*, per això creem noves carpetes dins de ella. També conté la carpeta *Classes*, en les que anirà el codi Java de la nostre aplicació.

4.1.3. Classes

Totes les classes presents a *Enquestes* que representen dades emmagatzemades a la base de dades (enquesta, pàgina, pregunta) són ***JavaBeans***. Els ***JavaBeans*** es defineixen com components de software reutilitzables que es poden manipular visualment en una eina de construcció. És a dir, gracies a tindre els objectes encapsulats com ***JavaBeans***., podem tractar-los tant en la capa de controlador (els ***Action Struts***), en la capa de model (***Hibernate*** realitza el mapatge directament sobre ells), o en la capa de vista (***JSP***).

Tots els ***JavaBeans*** han de complir certes característiques:

- La classe ha d'implementar la interfície *serializable* (gracies a la que posteriorment és capaç de ser persistent així com de restablir el seu estat).
- Ha de tenir un constructor sense arguments.
- Les seves propietats han de ser accessibles mitjançant mètodes *get* i *set*.
- Ha de contenir determinats mètodes de manipulació d'esdeveniments.

El *JavaBean Opinio* és un dels més senzills de la aplicació i on s'observen ràpidament els trets descrits anteriorment, tot tenint en compte, que a *Enquestes*, el beans no hereten directament de *Serializable*, sinó que ho fan de *ModelBO*, una classe pròpia que aquesta si hereta de *Serializable:*

```
package com.enquestes.model;

public class Opinio extends ModelBO {

	private Pregunta pregunta;
	private Resposta resposta;
	private Acces acces;
	private String text;

	public Opinio(){}

	public Acces getAcces() {
		return acces;
	}
	public void setAcces(Acces acces) {
		this.acces = acces;
	}
	public Pregunta getPregunta() {
		return pregunta;
	}
	public void setPregunta(Pregunta pregunta) {
		this.pregunta = pregunta;
	}
	public Resposta getResposta() {
```

```
            return resposta;
        }
        public void setResposta(Resposta resposta) {
            this.resposta = resposta;
        }
        public String getText() {
            return text;
        }
        public void setText(String text) {
            this.text = text;
        }
}
```

A la figura 44 podem observar el diagrama de classes de la aplicació. Podem observar com tot gira al voltant de la classe *Enquesta*, que és la classe principal. *Enquesta* conté classes *Pagina*, que a l'hora conté classes *Pregunta*, i aquesta repeteix la relació amb *Pregunta*. *Enquesta* referència a una classe *Format* que és la encarregada de determinar la visualització d'aquesta per pantalla. També conté una classe *Idioma* que és la encarregada de manegar els *locales* (fitxers d'idiomes) i per tant proporcionar el llenguatge corresponent.

Cal destacar la doble relació de *LlistaDistribucio* amb *Enquesta* i *Usuari*, on *Enquesta* pot contenir al igual que *Usuari*, una col·lecció de *LlistaDistribucio*. Degut a que existeixen molts tipus de preguntes, aquesta característica es reflecteix a la base de dades, i per tant, mitjançant *JavaBeans* també al model de classes. És per això que existeix la classe *Tipus* referenciada per *Pregunta*, i que determinarà el comportament d'aquesta última.

A destacar també, la relació de pertinença de *Resposta* no només amb *Pregunta*, sinó també amb si mateixa. Això succeeix degut a les preguntes matricials, on cada fila conté un determinat nombre de respostes, però que en si mateixa també representa una resposta.

Per altre banda, els actor més complert és el *Client*, és a dir, l'usuari que crea enquestes, per ho que evidentment contindrà classes de tipus *Enquesta*. A mode de configuració, també contindrà una classe de tipus *Sector*, *Pais* i *Idioma*.

Finalment, l'altre actor, el *Usuari* serà contingut per *LlistaDistribució* i podrà contenir instàncies *Acces* que a l'hora contindrà *Opinio*.

4.2. Base de dades

Per a poder mantenir les dades creades per *Enquestes*, aquestes han d'estar allotjades en una base de dades relacional. Aquesta base de dades és del tipus ***MySql*** [15], que és un sistema gestor de base de dades relacional multi-fil i multi-usuari amb més de 6 milions de instal·lacions en el món.

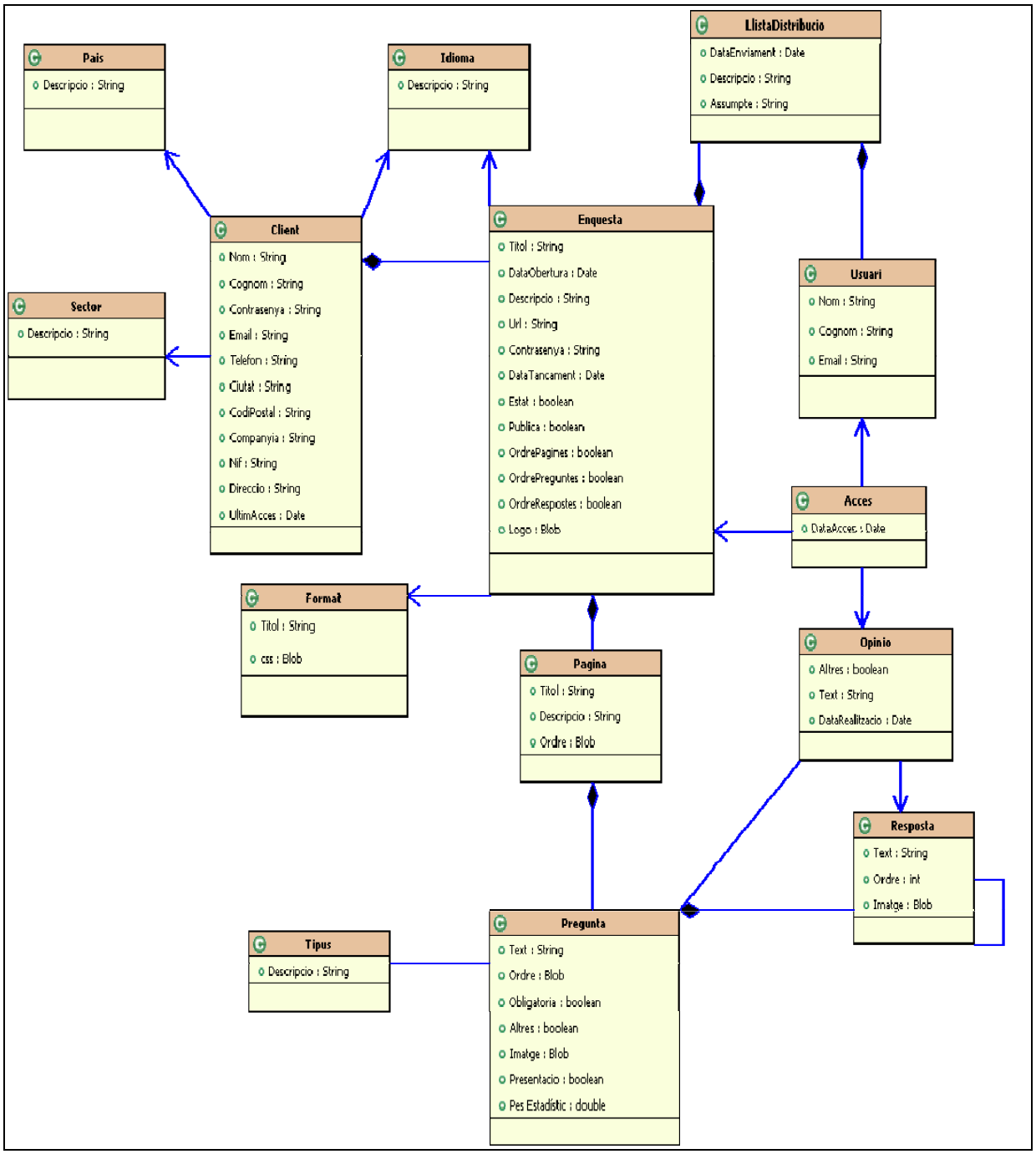


Fig. 44

Diagrama de classes de Enquestes.

MySQL, figura 45, és un sistema de gestió de bases de dades relacional (anglès RDBMS - Relational DataBase Management System) multi-fil (multithread), multi usuari, que usa el llenguatge SQL (Structured Query Language).

Al contrari d'altres projectes com el servidor d' HTTP Apache, MySQL pertany a, i està patrocinat per l'empresa sueca MySQL AB. La companyia desenvolupa i manté el programari en la versió lliure i la comercial. Ofereix suport als clients, assistència tècnica i documentació.

Fig. 45
Logotip MySql.

Es pot fer ús de MySQL en aplicacions de tota mena (web, d'escriptori o d'altres) de forma lliure i gratuïta sota les condicions de la llicència GPL (no comercial). Si MySQL forma part en la prestació d'un servei o producte amb lucre, les llicències d'ús s'han d'adquirir mitjançant la seva compra.

MySQL és un dels components de l'arquitectura LAMP (de Linux - sistema operatiu) i la WAMP (de Windows - sistema operatiu). Les tres sigles que segueixen (A,M i P) fan referència al conjunt format per Apache (servidor web), MySQL (base de dades) i PHP (llenguatge de programació). Una plataforma per a la construcció de llocs web utilitzant programari lliure.

MySql no és el SGBD més utilitzat empresarialment parlant, doncs sistemes com *Oracle* ofereixen millors prestacions, però per una aplicació d'aquests tipus, *MySql* esdevé immillorable, no només per les seves característiques tècniques, sinó també per la seva llicència *GNU GL*. *GNU GL* és l'acrònim de GNU General Public License i és una llicència creada per la *Free Software Foundation.* Gràcies a aquesta llicència, mentre la aplicació que la fa servir no sigui un producte privatiu, l'ús de *MySql* és lliure. A més *MySql* atorga a la aplicació una gran portabilitat així com facilitat i rapidesa en les tasques d'instal·lació.

Com es pot veure en la figura 46 en total existeixen 14 taules a *Enquestes*: *acces, client, enquesta, format, idioma, llistadistribucio, opinio, pagina, pais, pregunta, resposta, sector, tipuspregunta* i *usuari*. Algunes d'aquestes taules són d'òbvia necessitat, d'altres com *idioma* o *sector* poden ser de més dubtosa necessitat. Però la avantatge de tenir aquestes dades a la base de dades roman en la facilitat per ampliar les opcions. És a dir, si els tres idiomes existents ara mateix a *Enquestes* fossin "hard-coded", i volguéssim ampliar el nombre d'idiomes, hauríem de trobar totes aquelles parts de l'aplicació a on s'hi fa referència. En canvi romanent a base de dades, aquells escenaris on l'usuari ha d'escollir entre un d'ells reben el llistat d'idiomes disponibles de la base de dades, per tant si volguéssim afegir un idioma més, l'únic que hauríem de fer seria afegir un registre més a la base de dades i traduir un full de propietats amb les definicions del nou llenguatge. Similarment podríem extrapolar aquest concepte per *sector*, *format* o *tipuspregunta.*

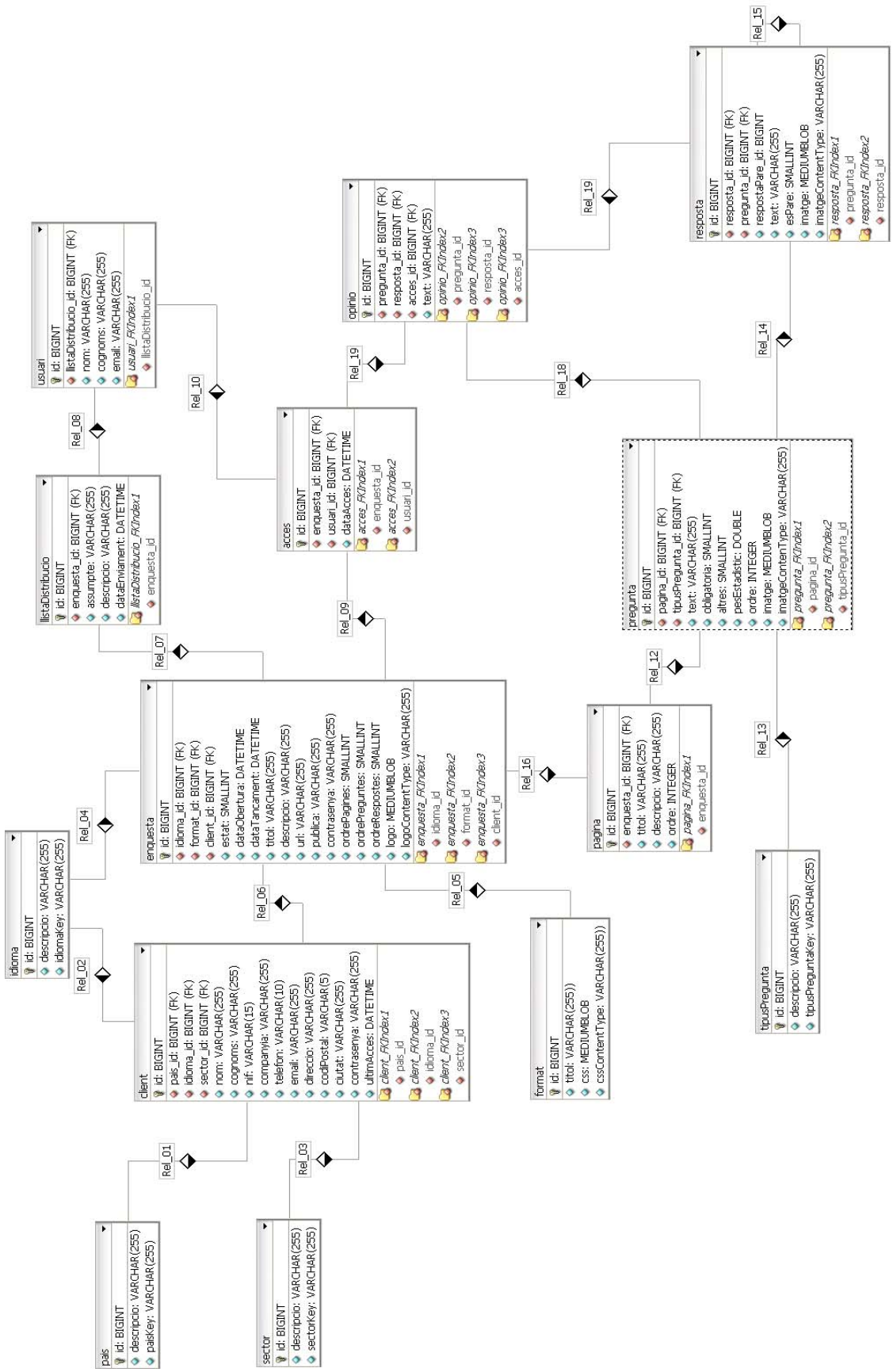


Fig. 46
Diagrama de entitat-relació.

5. Conclusions

A la fi d'aquest projecte, el repàs al cicle de treball seguit esdevé un motiu per a valorar la qualitat mateixa de resultat obtingut. La conclusió final més obvia possiblement sigui una dedicació desigual als dos pilars bàsics d'un projecte de final de carrera d'aquest tipus, el desenvolupament de la aplicació a avaluar i l'elaboració d'una documentació descriptiva d'aquesta, la memòria.

5.1. La aplicació, valoracions

L'elaboració de la aplicació s'ha dividit en 4 fases:

5.1.1. Anàlisis de requeriments i disseny de les estructures de dades necessàries

Una aplicació d'enquestes, havia de tenir un aspecte de manegament de dades, no podia pretendre ser una aplicació on els gràfics o altres components multimèdia tinguessin una gran presència, doncs bé tenia que respectar la finalitat final, la creació d'estructures d'informació organitzades, les enquestes. Així que si l'objectiu del *enquestador* o *client* de l'aplicació, era crear i manegar aquestes estructures, bé tenien que ser indexades i ser tractades per un sistema que prioritzés la seva administració per davant de qualsevol altre aspecte. Aquestes dades, havien d'estar correctament dissenyades amb l'objectiu de facilitar a l'aplicació el manegament de elles encapsulades en una estructura major, la *enquesta*.

Finalment, la aplicació si que va respectar aquestes premisses i des de el primer moment en que el *client*, era donat d'alta o logat a la aplicació, un panell de visualització de totes les enquestes era el primer pas dintre del recorregut que el *client* pogués realitzar dintre de la aplicació. A més, aquest panell mostra les principals accions aplicables a una enquesta, com la de ser modificada, eliminada o enviada, així com s'encarregava de mostrar l'estat d'aquesta o la seva data de creació. L'apartat de modificació i creació de l'enquesta és, sens dubte, la part més complexa i elaborada de la aplicació, on un gran nombre d'accions sobre l'enquesta i les seves part són accessibles ràpida i còmodament.

5.1.2. Construcció del framework

Si es cert, que totes les parts descrites i pertinents a la tecnologia J2EE, finalment faciliten i potencien el treball de desenvolupador, però la seva configuració i aplicació han de ser fortament considerades en el procés evolutiu d'una aplicació. Són moltes les parts en les que es divideix l'escenari de treball, i totes elles resulten de gran importància.

A l'hora d'escollir l'IDE amb el que treballar, va ser fàcil l'elecció, doncs Eclipse és pràcticament un estàndard al món comercial actual. I l'autor d'aquest

projecte ja tenia cert tractament amb l'Eclipse, doncs és l'eina habitual de treball en els últims temps. A notar que l'altre opció possible era el IDE NetBeans, més orientat al desenvolupament d'aplicacions d'escriptori.

Si Eclipse representa el IDE estàndard en aplicacions J2EE i fins i tot Java en general, enfocant d'una manera més acurada l'escenari de aplicacions web dinàmiques amb tractament de dades (base de dades), hem de identificar *Struts* igualment com un estàndard de facto. De fet, si l'autor hagués de ressaltar una característica per sobre de les altres, seria *Struts*. La seva adaptació a la aplicació no és obvia ni senzilla però les ajudes que aquest dona així com la seva potencia la fan una eina imprescindible.

Finalment, com a tasques més rellevants, la inclusió de llibreries i l'estructuració dels arxius que configuren finalment la aplicació, també son causa de reflexió i deteniment.

5.1.3. Implementació

Una vegada la taula de treball està llesta i les idees clares, és hora de entrar en acció i porta a terme els plantejaments realitzats. *Enquestes* conté 135 classes Java primordialment repartides en controlador, accions i beans. 67 arxius jsp, que determinen el model vista, i han estat optimitzats per una navegació còmoda i clara, agafant com a base el navegador *Mozilla Firefox* per a una configuració gràfica de 1280 x 800. En arxius de configuració trobem 23 arxius tipus XML o *properties*.

Així que una pregunta possible és: "Realment és necessari volum de desenvolupament per a una aplicació de creació d'enquestes?". La meva resposta és si, és si perquè com a futur enginyer no puc o no he de, entendre un desenvolupament com una solució orientada a resoldre una problemàtica puntual. És a dir, *Enquestes* és una aplicació estructurada i fortament extensible:

És extensible perquè l'abstracció de dades permet afegir, eliminar, o modificar l'estructura física de les dades sense alterar cap altre part. És extensible perquè el model vista implementat permet la reutilització dels recursos sense escriure una sola línia més, perquè l'ús dels tags JSP permet afegir qualsevol idioma addicional simplement omplint una fulla de propietats, perquè la presentació de la web resideix únicament en arxius CSS fent així possible alterar tot l'aspecte visual del site modificant una simple línia de codi. És extensible perquè segueix els estàndards J2EE, possibilitant així que un desenvolupador posterior sense coneixement previ de la aplicació, identifiqui ràpidament quins i on són les parts principals de la aplicació, facilitant la seva ràpida familiarització amb l'escenari.

Com a apunt individual de l'autor d'aquest projecte, amb una breu experiència de 3 anys en el desenvolupament d'aplicacions web comercial, estructurar, seguir estàndards i respectar l'ordre no és la millor manera de realitzar una aplicació, sinó que és l'única. Són molts els casos en el món laboral, en el que el moviment de desenvolupadors, implica per una aplicació passar de mà en mà constantment, i a la meva visió, només aquelles aplicacions realitzades amb les característiques esmentades, són capaces de sobreviure.

5.1.4. Configuració servidor

Aquesta fase representa una peça clau en el resultat final d'aquet projecte, però mitjançant eines potents i configurables com ho són *MySql* i *Apache Tomcat*, un altre característica que resultaria bàsica en una aplicació comercial, és la **portabilitat**. Tant en sistemes Windows com like-unix, la instal·lació de un servidor Tomcat és molt intuïtiva, hi encara que no té una panell de control tant visual com la de IIS de Microsoft, els seus fitxers de configuració permeten una edició ràpida i directe. A ressaltar que la convivència entre servidors Apache (aquest per si sol no suporta Java) i Apache Tomcat és molt habitual, essent el Apache el servidor que escolta les peticions en primer lloc, i redirigint la acció al Tomcat en cas de que les peticions rebudes impliquessin l'acció de components J2EE. Aquest factor explica els ports per defecte d'aquests dos servidors, on el 80 (l'utilitza't en les peticions HTTP) és el trobat a el Apache i el 8080 en el Tomcat. *Enquestes* disposa actualment d'un servidor publicat a Internet, http://enquestes.servehttp.com/enquestes , que com es pot observar no precisa el port 8080. Això no succeeix per cas esmentat anteriorment on dos servidors conviuen juntament, sinó a la capacitat de configuració del Tomcat que en aquest cas escolta pel port 8080. Remarcar que aquest servidor no a precisat de cap mena d'inversió, doncs es tracta d'un PC convencional connectat a un ISP convencional que disposa d'una redirecció DNS dinàmica, gràcies a la aplicació en modalitat gratuïta NO-IP.

L'aplicació disposa de fitxers Ant (fitxers de comandes automàtiques) que comprimeixen tot el codi de la aplicació en un fitxer **jar (**similar a un rar destinat a aplicacions Java**).** Al incialitzar-se el Tomcat la carpeta *webapps* continguda en home folder de la aplicació, és analitzada en busca d'aquesta mena d'arxiu i per cada un dels trobats, una aplicació web és 'deployada' automàticament.

Un arxiu sql resideix en una carpeta de documentació de l'aplicació, que conté totes les comandes de construcció de la base de dades en funció d'un nom d'usuari, una contrasenya i un host. Així que per instal·lar *Enquestes* en un servidor amb sortida a Internet, les passes a realitzar són les següents.

1. Instal·lar Java jdk.
2. Instal·lar Apache Tomcat.
3. Utilitzar una aplicació d'accés a bases de dades MySql per executar el script de construcció sql.
4. Copiar el jar de l'aplicació a la carpeta *webapps* del Tomcat.
5. Engegar el *Tomcat.*

Les instal·lacions realitzades seguint aquestes passes han durat aproximadament 10 minuts, tret que fa indicar que la majoria de perfils informàtics no superarien els 30 minuts. Aquest factor indica un alt nivell de **portabilitat**.

5.1.5. Fase de proves

La fase de proves ha estat exhaustiva i complerta, certificant així un ús de la aplicació sense errors. Aquesta s'ha executat a la fi de totes les implementacions modulars de manera unitària, i global a la fi del desenvolupament total.

5.2. Comparacions

És complicat comparar un projecte de final de carrera amb una aplicació comercial desenvolupada per tenir èxit comercial. Totes les aplicacions web que he trobat amb finalitats similars a *Enquestes*, donen la possibilitat de provar la aplicació creant una o enquestes, per a posteriorment o bé no permetre ser resposta, o bé limitar la quantitat de accessos que altres usuaris poden realitzar per a respondre-la. Posteriorment proposen al usuari una subscripció o bé un paquet de respostes. Aquelles aplicacions que realitzen una venda de subscripció donen accés il·limitat a la seva aplicació (creació d'enquestes, resultats, estadístiques..) durant un cert període de temps. Aquelles altres que proposen l'adquisició de paquets de respostes al usuari, no impedeixen que aquest pugui crear enquestes, modificar-les o publicar-les, però si accedir a les respostes dels altres usuaris, i és on el tipus de paquet adquirit determina el nombre de respostes a observar per part de l'enquestador.

Així doncs, aquestes aplicacions esmentades, tenen la possibilitat de donar un suport constant a la seva aplicació i així poder ampliar les seves capacitats i funcions. Però l'objectiu principal de totes aquestes aplicacions com la de Enquestes és la creació d'enquestes per a la seva posterior resposta per part d'un conjunt enquestat, amb un final objectiu per al enquestador que és el de observar els resultats. Des d'aquest prisma, Enquestes resol la mateixa funció principal que la resta d'aplicacions. Però per complementar aquesta funció principal, es pot complementar l'aplicació amb tres tipus de complements:

- **De estructura de enquestes:**
  Altres tipus de preguntes o bé complements per a elles, com poden ser elements multimèdia (imatges, vídeos...) que puguin enriquir la pregunta.

- **De enviament de enquestes:**
  Serveis de missatgeria tant per mail com per sms, als enquestador donant-li a conèixer l'estat de les seves enquestes. Serveis d'enviament massius d'enquestes per mail.

- **De recepció de respostes i estadístiques:**
  Obtenir les respostes en altres formats com CSV o XML, així com altres tipus de gràfics.

5.3. Possibles ampliacions i millores

Algunes previsions inicials de l'aplicació no han estat portades a terme, bé per adequació al disponibilitat del autor, o bé per la seva dificultat tècnica. Una de les previsions, indicades en aquesta memòria i no realitzada, ha estat la implementació de dos tipus d'actors diferents, el *client* i el *usuari*. A l'aplicació només existeix el *usuari*, que és el creador de les enquestes. L'objectiu de l'existència de l'usuari era donar més capacitat estadística a les enquestes mitjançant una breu i optativa descripció a l'enquestat. D'aquesta manera, no només obtindríem un resultat absolut de les respostes obtingudes, sinó que aquestes podrien ser classificades en variables d'observació en funció del perfil dels enquestats. Aquesta és una implementació que no es troba a les

altres aplicacions web observades, però que en el tipus d'enquesta persona-persona si són contemplades. Un exemple d'aquesta característica seria una simple diferenciació de sexe, entre si l'enquestat és home o dona.

Una possible millora consistiria en emmagatzemar un camp de Finalitzat a les enquestes. D'aquesta manera quant un *usuari* hagués accedit a una *enquesta* per a respondre-la, però no l'hagués acabat (no hauria respost a totes les preguntes de l'*enquesta*), l'*acces* creat emmagatzemaria les respostes de l'*usuari* però emmagatzemaria també que aquesta no està acabada. Gràcies a aquesta característica, un *usuari* podria recuperar aquest *acces*, i acabar de respondre a l'*enquesta*.

Una altre funcionalitat prevista que en aquest cas si s'ha portat a terme però de manera no satisfactòria, és l'enviament d'enquestes per mail. Per a realitzar enviaments és necessari un servei d'enviament SMTP/POP3, i les aplicacions gratuïtes trobades a la web com MiniRelay [19], no aconseguien traspassar els filtres de *spam* dels servidors de correu. En les proves realitzades, només 1 de 5 servidors testejats va acceptar com vàlid el correu.

Conjuntament amb la opció d'enviar els correus amb les direccions de les enquestes, una altre ampliació possible seria la d'enviar un panell d'informació sobre els enviaments realitzats, indicant informació a l'estat de la consulta per part del usuari enquestat.

Bibliografia

[1] Wikipedia. http://www.wikipedia.org

[2] [Mc Graw Hill, 200] Stanton, Etzel, Walker. Fundamentos de Marketing.

[3] [Continental, 2002] Richard L. Sandhusen. Mercadotecnia.

[4] [FT Prentice Hall, 2006] Naresh Malhotra.

[5] http://*www.surveymonkey.com*
Aplicació web per a la creació d'enquestes. Survey Monkey va ser creada en el 1999, i les seves oficines estan situades en Portland, Oregón, Estats Units. Ofereix una subscripció per 19.95 dòlars al mes amb accés il·limitat a aplicació.
Idiomes disponibles: Anglès.

[6] http://*www.2ask.net*
Aplicació web per a la creació d'enquestes. Ofereix paquets de creació d'enquestes i recepció de respostes entre 398,00 € i 998,00 €
Idiomes disponibles: Anglès\Alemany.

[7] http://*www.esurveyspro.com*
Aplicació web per a la creació d'enquestes. *eSurveysPro.com* és una aplicació web creada per *Outside Software Inc.*, amb seu a Bucarest, Romania. És l'única aplicació que ofereix una modalitat de creació i recepció de respostes gratuïta sense limitacions. Paral·lelament ofereix altres modalitats de subscripció entre 100€ i 200€, amb eines addicionals com suport telefònic o configuració de *look and feel*.
Idiomes disponibles: Anglès.

[8] http://www.acn.cat
Agència Catalana de Notícies. Aplicació web construïda en J2EE.

[9] [SourceBeat, 2004] Jakarta Struts Live, Rick Hightower. Website: http://www.theserverside.com/tt/books/sourcebeat/JakartaStrutsLive/index.tss

[10] Spring Framework home. WebSite: http://www.springframework.org/

[11] Hibernate Reference Documentation, version 3.2.1. WebSite: Http://www.hibernate.org

[12] The Apache Software Foundation. WebSite: http://struts.apache.org/

[13] Apache Tomcat User Guide. WebSite: http://tomcat.apache.org/tomcat-5.5-doc/index.html

[14] Eclipse home. WebSite: http://www.eclipse.org/

[15] MySql Enterprise. WebSite: http://www.mysql.com

[16] Java Platform SE 6 API. WebSite: http://java.sun.com/javase/6/docs/api/

[17] Java JSTL 1.1 API. WebSite:
http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html

[18] JavaServer Pages Standard Tag Library 1.1. WebSite:
http://java.sun.com/products/jsp/jstl/1.1/docs/api/index.html

[19] MiniRelay. Servidor de Mails per Apache. Download WebSite:
http://blog.almadark.com/2005/09/05/minirelay-servidor-de-mail-en-apache/

Firmat: David Eduard Espinosa Ruiz.

Bellaterra, Juny de 2008

Resum

Les enquestes són un mètode excel·lent per a extreure informació relativa a conjunts de població específics, amb la finalitat de conèixer l'opinió d'aquests sobre l'objectiu de la enquesta. En aquest treball es proposa una implementació d'una aplicació que té com objectiu reduir l'esforç associat a la creació d'enquestes, recepció de respostes i visualització de resultats. Totes les tecnologies utilitzades són gratuïtes, i la implementació es porta a terme amb els estàndards de J2EE, prioritzant la extensibilitat i la portabilitat per sobre altres característiques.

Resumen

Las encuestas son un método excelente para la extracción de información relativa a conjuntos de población específicos, con la finalidad de conocer la opinión de estos respecto al sujeto de la encuesta. En este trabajo se propone una implementación de una aplicación que tiene como objetivo reducir el esfuerzo asociado a la creación de encuestas, recepción de opiniones y visualización de resultados. Todas las tecnologías utilizadas son gratuitas, y la implementación se lleva a cabo con los estándares J2EE, priorizando la extensibilidad y la portabilidad por encima de los demás factores.

Abstract

Surveys are an excellent method for extracting relative information from specifics population groups, where the purpose is to know the opinion of the survey's subjects. This assignment has been designed specifically for an application development that expects to reduce the necessary energy associated with survey's creation, information acquiring and result viewing. All the technologies used are free, and the application is developed with J2EE standards, with extensibility and portability above all other factors.