ContrlD

Aplicació Web de seguiment remot dels autocontrols de malalts diabètics.

Fabià Vilar Garcia
ETIS

Salvador Campo Manzarico

25 de juny de 2008

Dedicatòria

A la meva petita Alba i la meva dona Espe, per les meves absències.

RESUM

Aquest document es la memòria del meu Treball de Fi de Carrera, centrat a l'àrea d'aplicacions J2EE.

Dins, trobem la descripció del problema inicial que es proposa, el qual pretenem minimitzar amb la implantació del producte final del projecte.

També trobem un resum de les funcionalitat que ofereix així com la descripció dels requisits, l'anàlisi, el disseny i detalls de la implementació, junt amb als diagrames UML més significatius. També s'adjunten captures de les principals pantalles de la interfície gràfica.

Igualment s'indiquen els productes utilitzats per a la seva implementació: ides, frameworks, servidors, bases de dades.

Cal destacar, a la fase d'implementació, la utilització d'Struts, una framework molt utilitzat en el desenvolupament d'aplicacions Web, que facilita la implementació del patró arquitectònic MVC (Model Vista Controlador) i d'Hibernate, un motor de persistència ORM (Object/Relational Mapping) que facilita l'accés a les taules de les bases de dades que permet implementar la persistència treballant directament amb objectes.

1. Index

1. Index. 4
2. Index de figures. 5
3. Introducció. 6
    1. Justificació 6
    2. Context. 6
    3. Objectius. del projecte. 7
    4. Enfocament i mètode seguit. 7
    5. Planificació. 7
    6. Productes obtinguts. 8
    7. Descripció de la resta de capítols. 8
4. Requisits i anàlisi. 9
    1. Requisits funcionals demanats. 9
    2. Identificació de rols i funcionalitats assignades. 9
    3. Casos d'us. 10
        1. Descripció. 10
        2. Diagrames de casos d'ús. 21
    4. Classes derivades de l'anàlisi. 24
5. Disseny. 25
    1. Disseny de la interfície d'usuari. 25
    2. Disseny de la persistència. 32
    3. Subsistemes. 35
6. Arquitectura. 37
    1. Aplicacions en capes. 37
    2. Patró MVC 2 38
7. Implementació 41
    1. Característiques. 41
    2. Implementació de la Capa Vista. 43
    3. Implementació de la Capa Model. 46
    4. Implementació de la Capa Controlador. 48
    5. Paquets. 50
    6. Funcionalitats implementades. 51
8. Millores. 52
9. Programari utilitzat. 52
10. Valoració econòmica. 53
11. Conclusions. 53
12. Glossari. 54
13. Bibliografia. 54

2. Index de figures

Figura 1. Diagrama de Gantt. 7
Figura 2. Diagrama de casos d'us administrador 21
Figura 3. Diagrama de casos d'us cap 22
Figura 4. Diagrama de casos d'us pacient 22
Figura 5. Diagrama de casos d'us especialista. 23
Figura 6. Diagrama de classes. 25
Figura 7. Pantalla Login 25
Figura 8. Pantalla menú administrador. 26
Figura 9. Pantalla menú pacient. 26
Figura 10. Pantalla menú especialista. 27
Figura 11. Pantalla menú cap. 27
Figura 12. Pantalla Alta Usuari 28
Figura 13. Pantalla Consultar Controls 28
Figura 14. Pantalla Estadístiques 29
Figura 15. Pantalla Assignar Especialista 29
Figura 16. Pantalla Introduir Glucèmies 30
Figura 17. Pantalla Introduir Dosis Oiaries. 30
Figura 18. Pantalla Consultar Dosis Base 31
Figura 19. Diagrama ER 33.
Figura 20. Diagrama de subsistemes. 36
Figura 21. Aplicació per capes. 37
Figura 22. Patró MVC. 38
Figura 23. Patró Front Controller. 39
Figura 24. Diagrama seqüència Front Controller. 40
Figura 25. Diagrama aplicació Struts. 41
Figura 26. Diagrama aplicació Hibernate 42

3. Introducció

3.1. Justificació.

El projecte parteix de la meva experiència personal com a diabètic i animat per la escassa o nul·la aplicació que hi ha de les noves tecnologies en la prestació d'assistència remota sanitària als malalts en general.
Malgrat que, amb tota seguretat, aquesta aplicació no s'implantarà al món real, he cregut interessant basar el projecte en aquesta idea.

3.2. Context del projecte.

El Servei d'Endocrinologia i Nutrició de l'Hospital de Bellvitge s'encarrega de donar atenció especialitzada a pacients amb malalties endocrinològiques i nutricionals, tant dels ingressats a l'Hospital com als atesos a Consultes Externes.
Aquest servei cobreix tota la regió sanitària anomenada “Costa de Ponent”, que està formada per la ciutat de l'Hospitalet de Llobregat i les comarques d'El Baix Llobregat, El Garraf i l'Alt Penedes.

D'entre els malalts atesos a aquest servei hi ha el grup de diabètics tipus 1. Aquests malalts realitzen visites periòdiques als metges endocrinòlegs que atenen a Consultes Externes.
En aquestes visites, els malalts, porten el valors de les glucèmies practicades des de la darrera visita anotats en una llibreta, proporcionada pel mateix hospital. En aquesta llibreta també apunten les dosis d'insulina administrades durant aquest període. Entre visites, poden passar 4 o 6 mesos.

Durant la visita, el metge revisa aquests valors i els contrasta amb les dades obtingudes amb anàlisis practicades amb anterioritat. Si els valors dels anàlisis no son adequats, o empitjoren respectes els anteriors, revisa detalladament els valors anotats a la llibreta i fa canvis a les dosis base que dona al pacient.

Es proposa, amb aquest projecte, dissenyar i implementar un aplicació en la que els pacient puguin introduir aquests valors en un ordinador des de la seva, accedint, per fer-ho, a una web per a tal efecte. Aquesta mateixa aplicació ha de permetre als metges consultar aquests controls i tenir una visió més actualitzada dels controls que es fan els pacients entre visites, i si es necessari, canviar les dosis d'insulina.

3.3. Objectius.

El principal objectiu del projecte és el desenvolupament d'una aplicació Web amb tecnologia J2EE, des del plantejament del problema fins la entrega d'un producte funcional, passant per les fases d'anàlisi i disseny.

Com a objectiu secundari, no per això menys important, es la adquisició de coneixements i experiència en aquesta tecnologia, la qual cosa em permetrà desenvolupar posteriorment aplicacions més elaborades.

3.4. Enfocament i mètode seguit.

La metodologia utilitzada pel desenvolupament de la aplicació es la del cicle de vida clàssic. En aquest, dividim el projecte en diferents fases: recollida de requisits, anàlisi, disseny, implementació i proves.

3.5. Planificació.

La primera versió de la planificació no era molt comprometedora. Posteriorment es va tornar a fer amb més detall, tal i com es mostra al diagrama de Grantt adjunt.

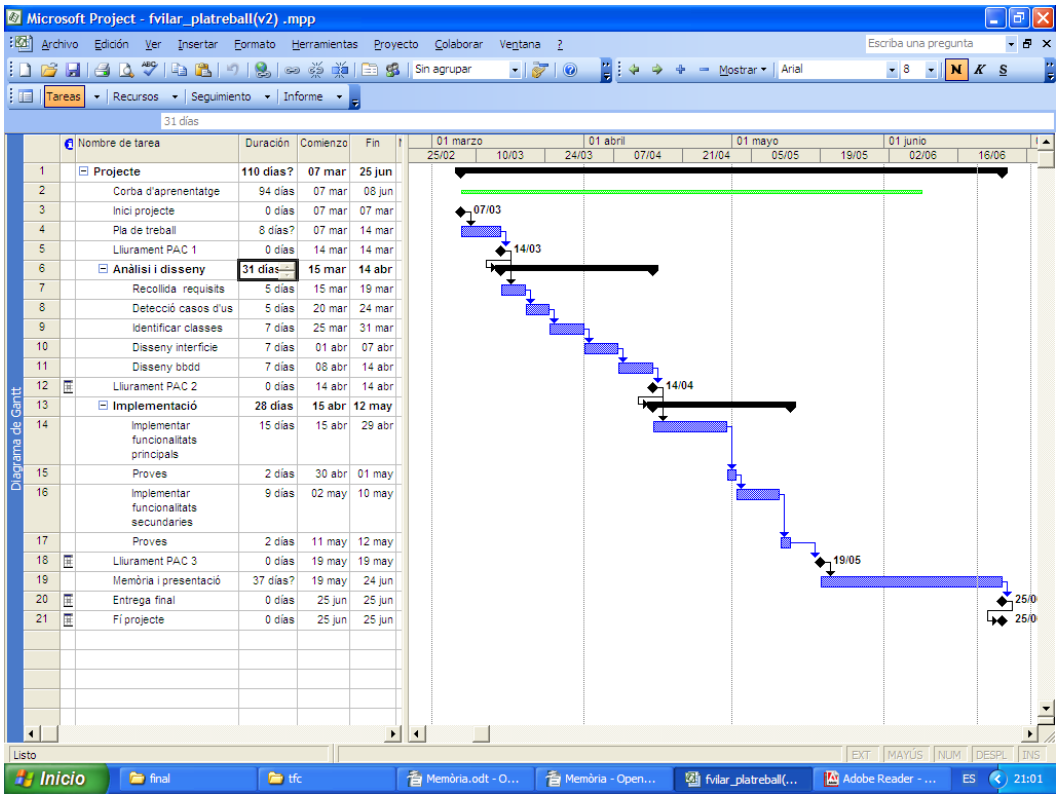


Figura 1

Destaquem, com a fites importants dins de la planificació, la entrega de les pacs.

Tot i això, la corba d'aprenentatge ha incidit molt en el compliment del pla. S'ha de tenir en compte que abans de l'inici del projecte, el coneixement de les tecnologies i productes utilitzats en el desenvolupament era molt baix o nul.

3.6. Productes obtinguts.

S'han obtingut els següents productes:

- Codi font: que inclou el fitxers java, xml i jsp.
- Fitxer war pel desplegament a un servidor.
- Doc. del projecte.
- Script de creació i populació de la bbdd.
- Memòria del projecte.
- Document de presentació virtual.

3.7. Descripció de la resta de capítols de la memòria.

- Requisits i anàlisi.
  Anàlisi del requisits i detecció dels casos d'ús i actors que hi participen així com les classes.

- Disseny de la interfície gràfica.
  Disseny de la interfície que interacciona amb els diferents usuaris de la aplicació.

- Disseny de la persistència.
  Disseny de les taules de la base de dades de la aplicació segons les especificacions.

- Arquitectura.
  Justificació de la selecció del tipus d'arquitectura de la aplicació.

- Implementació.
  Detalls de la implementació de les diferents parts(frameworks, llibreries, etc)

- Programari utilitzat.
  Descripció del programari utilitzat en el projecte: ides, frameworks, editors uml, bases de dades.

- Valoració.
- Conclusions

4. Requisits i anàlisi.

4.1 Descripció dels requisits funcionals demanats.

Els pacients han de poder accedir a traves d'Internet, amb un ordinador amb navegador.

El personal de l'Hospital (metges, cap, administrador de la aplicació) accedirà a la aplicació a traves d'Internet o de la intranet corporativa, mitjançant, també, un ordinador que disposi d'un navegador.

Els pacients han de poder introduir el valors les glucèmies practicades i les dosis administrades de forma individual o diària. També han de poder consultar les dosis base en qualsevol moment i els controls des d'una data donada

El metge ha de poder consultar els controls dels pacients i de les dosis administrades així com tenir dades estadístiques de cadascú: mitjana de valors, valor màxim i quantitat de controls efectuats. Des d'una data donada.

El metge cap ha de poder assignar i canviar un metge a un pacient. També ho ha de poder canviar, si cal.

L'administrador s'encarrega de donar d'alta i baixa els diferents usuaris del sistema. També ha de poder modificar les dades dels usuaris.

4.2. Identificació dels rols i funcionalitats de cadascú.

Del anàlisi dels requisits detectem els actors que intervenen en el sistema i les seves responsabilitats.

Administrador.

Altes d'usuari.
Baixes d'usuari.
Modificar les dades dels usuaris.

Cap.

Assignar especialista als pacients.

Especialista.

Consultar controls dels pacients.
Consultar dosis dels pacients.
Consultar estadístiques dels controls dels pacients.
Consultar gràfics de les dades dels controls dels pacients.
Canviar dosis base del pacient.

Pacient.

Consultar controls .
Consultar dosis .
Consultar dosis base.
Introduir controls (diari o individualment).
Introduir dosi (diari o individualment).

La descripció d'aquestes funcionalitats es la següent:

-Login:

La aplicació ens demanarà el nom d'usuari i la clau. Una vegada confirmada surt el menú amb les funcionalitats de la aplicació, que seran diferents per a cada rol.

-Alta d'usuari:

Dona d'alta un usuari. S'introdueixen les dades de l'usuari (nom, cognoms, clau, rol...) i son emmagatzemades a la base de dades.

-Baixa d'usuari:

Dona de baixa un usuari. Es cerca a la base de dades i s'elimina.

-Assignar metge:

Assigna metge a un pacient. Es tria el pacient i el metge assignat.

-Canviar metge:

Canvia el metge assignat al pacient.

-Consultar estadístiques:

Consulta les estadístiques generades amb les dades dels pacients.

-Canviar dosi

Canviar les unitats d'insulinaanvia aquest valor

-Introduir glucèmies.

El pacient introdueix els valors de les glucèmies obtingudes

-Introduir dosis.

El pacient introdueix els valors de les dosis administrades.

4.3. Casos d'us..

4.3.1. Descripció.

Cada taula descriu un cas d'ús i les seves propietats:

Cas d'ús:	Login
Descripció	Identifica i valida l'usuari per utilitzar el sistema
Actors	Administrador, Metge, Metge Cap, Pacient
Casos d'us relacionats	
Precondició	L'usuari ha accedit a la pàgina de login
Flux principal	Seqüència: - Apareix una pàgina amb una caixa de text pel login i altra pel password. - L'usuari introdueix el seu login. - Introdueix el password en l'altra caixa. - Fa clic al botó d'acceptar. - El sistema comprova les dades i si son correctes s'accedeix a la pàgina assignada segons el tipus d'usuari.
Flux alternatiu	Si les dades no son correctes el sistema dona un missatge d'error i dona altra oportunitat per introduir les dades correctament

Postcondició	L'usuari està validat al sistema i ha accedit a la pàgina que te assignada.

Cas d'ús:	Alta usuari
Descripció	Es dona d'alta al sistema a un nou usuari afegint les dades personals i de validació.
Actors	Administrador
Casos d'us relacionats	
Precondició	L'usuari no existeix al sistema i l'administrador està a la pàgina d'administració de la aplicació.
Flux principal	Seqüència: - L'administrador fa clic a l'enllaç “Alta”. - Apareix un formulari amb camps per omplir: login password dni cognom1 cognom2 nom tipus d'usuari (desplegable amb els tipus d'usuari possibles i un camp diferent per a cada tipus d'usuari) - Una vegada omplerts fa clic al botó “Acceptar”. - Si tot es correcte el sistema emmagatzema les dades i dona el missatge “Dades emmagatzemades”
Flux alternatiu	Si les dades no son correctes el sistema dona un missatge d'error i dona altra oportunitat per introduir les dades correctament
Postcondició	Les dades del usuari han estat validat i han estat emmagatzemades al sistema.

Cas d'ús:	Baixa usuari
Descripció	Es dona de baixa al sistema a un usuari eliminant les dades del sistema
Actors	Administrador
Casos d'us relacionats	Cercar usuari.
Precondició	L'usuari està donat d'alta al sistema i l'administrador està a la

	pàgina d'administració de la aplicació.
Flux principal	Seqüència: - L'administrador fa clic a l'enllaç de baixa d'usuari. - Apareix un formulari amb camps per omplir els criteris de cerca: login cognom1 cognom2 nom - Una vegada omplerts fa clic a “Acceptar”. - Si tot es correcte el sistema mostra l'usuari coincident amb les dades del formulari. - Es marca la casella de verificació d'eliminació. - Es marca el *check* de “Confirmar”. - El sistema dona un missatge confirmant la eliminació de les dades de l'usuari
Flux alternatiu	Si no es troba cap coincidència en al cerca de l'usuari, el sistema avisa i dona alta oportunitat per introduir les dades
Postcondició	L'usuari ha estat donat de baixa del sistema i les seves dades han estat eliminades.

Cas d'ús:	Baixa usuari
Descripció	Es dona de baixa al sistema a un usuari eliminant les dades del sistema
Actors	Administrador
Casos d'us relacionats	Cercar usuari.
Precondició	L'usuari està donat d'alta al sistema i l'administrador està a la pàgina d'administració de la aplicació.
Flux principal	Seqüència: - L'administrador fa clic a l'enllaç de baixa d'usuari. - Apareix un formulari amb camps per omplir els criteris de cerca: login cognom1 cognom2 nom - Una vegada omplerts fa clic a “Acceptar”.

	- Si tot es correcte el sistema mostra l'usuari coincident amb les dades del formulari. - Es marca la casella de verificació d'eliminació. - Es marca el *check* de “Confirmar”. - El sistema dona un missatge confirmant la eliminació de les dades de l'usuari
Flux alternatiu	Si no es troba cap coincidència en al cerca de l'usuari, el sistema avisa i dona alta oportunitat per introduir les dades
Postcondició	L'usuari ha estat donat de baixa del sistema i les seves dades han estat eliminades.

Cas d'ús:	Modificar usuari
Descripció	Modificar les dades del usuari.
Actors	Administrador
Casos d'us relacionats	Cercar usuari
Precondició	Es vol modificar les dades d'un usuari.
Flux principal	Seqüència: - L'administrador fa clicka l'enllaç de modificació d'usuari. - Apareix un formulari amb camps per omplir els criteris de cerca: login dni - Una vegada omplert qualsevol dels 2 es fa clic a “Acceptar”. - Si tot es correcte el sistema mostra l'usuari coincident amb les dades del formulari amb els camps modificables. - Es modifica el camp que sigui necessari. - Es clica “Acceptar”. - Si tot es correcte el sistema ho confirma.
Flux alternatiu	Si no es troben coincidències al sistema es mostra un missatge i dona altra oportunitat per corregir les dades introduïdes al formulari.
Postcondició	Les dades de l'usuari han estat modificades a la base de dades del sistema.

Cas d'ús:	Assignar especialista
Descripció	El metge cap pot assignar o canviar un metge especialista a cada pacient.
Actors	Metge Cap
Casos d'us relacionats	Cercar pacient. Seleccionar especialista.
Precondició	El pacient no te assignat cap especialista o es vol assignar un altre especialista. El cap es troba a la pàgina assignada.
Flux principal	Seqüència: - Es fa clic a l'enllaç “Assignar especialista”. - Apareix una pàgina amb diferents camps per cercar un pacient. (Cas d'us “Cercar Pacient”) - S'omplen el camps segons el criteris de cerca. - El sistema torna les dades del pacient i un camp desplegable buit si no té un metge assignat o amb el metge assignat si ja ho te. - Es fa clic en aquest camp i es desplega amb la llista de metges.(Cas d'us “Seleccionar especialista”) - Selecciona un especialista. - Es fa clic al botó “Acceptar”. - El sistema actualitza la base de dades i confirma la acció.
Flux alternatiu	Si el sistema no troba el pacient dona un missatge d'error i dona altra oportunitat per corregir les dades introduïdes en la cerca.
Postcondició	El pacient te assignat un especialista

Cas d'ús:	Seleccionar especialista.
Descripció	Es selecciona un metge de la base de dades
Actors	Cap
Casos d'us relacionats	Assignar metge
Precondició	Es vol assignar un metge a un pacient.
Flux principal	- Es fa clic a un camp desplegable amb els metges. - Es tria un metge.
Flux alternatiu	

Postcondició	El metge ha estat seleccionat

Cas d'ús:	Cercar pacient.
Descripció	Per cercar un pacient a la base de dades. S'utilitza quan es vol assignar o canviar un metge a un especialista o consultar els seus controls.
Actors	Especialista, Cap
Casos d'us relacionats	Assignar metge. Consultar controls.
Precondició	Es volen consultar el controls del pacient o assignar un metge a un pacient.
Flux principal	Seqüència: - Apareix un camp per omplir amb el nº de targeta sanitaria del pacient. - Una vegada omplert es fa clic a “Cercar”. - Si tot es correcte el sistema mostra les dades del pacient.
Flux alternatiu	Si el sistema no troba el pacient dona un missatge d'error i dona altra oportunitat per corregir les dades introduïdes en la cerca.
Postcondició	Es mostren les dades del pacient i un link a “Consultar controls” (si es l'especialista)

Cas d'ús:	Consultar controls setmanals
Descripció	Per visualitzar els els controls introduïts pel pacient en una setmana concreta.
Actors	Especialista, Pacient
Casos d'us relacionats	Cercar pacient. Canviar dosis base.
Precondició	Es volen consultar els controls setmanals d'un pacient
Flux principal	Seqüència: - Es fa clic al link opció “ Consultar controls”. - Apareix una pantalla en el frame on es pot triar en un camp desplegable la setmana que es vol visualitzar. - Es tria la setmana. - Es fa clic al botó “Acceptar” - El sistema torna les dades demanades
Flux alternatiu	Si no s'han introduir valors durant la setmana triada el sistema dona un missatge d'error.
Postcondició	El sistema presenta el controls i dosis introduïdes durant la

	setmana triada i un link a “Canviar dosis base” si el consultant es l'especialista.

Cas d'ús:	Canviar dosis base.
Descripció	El metge canvia el valors de les dosis base del pacient.
Actors	Especialista
Casos d'us relacionats	Consultar estadístiques, Consultar controls setmanals.
Precondició	S'estan consultant els controls setmanals del pacient.
Flux principal	Seqüència: - Es fa clic a l'enllaç “Canviar dosis base” - Surten 3 camps amb els valors i tipus de la insulines assignades. - Es corregeixen els valor. - Es fa clic a “Acceptar”. - Si tot es correcte el sistema dona un missatge confirmant l'acció.
Flux alternatiu	
Postcondició	El valors de les dosis base del pacient han estat actualitzats a la base de dades del sistema.

Cas d'ús:	Consultar estadístiques.
Descripció	Es vol tenir el valor de la mitjana dels valors de les glucèmies dels últim 15 dies, per franja del dia i total. i visualitzar gràfics de la evolució d'aquests valors també per franja del dia.
Actors	Especialista
Casos d'us relacionats	Consultar estadístiques, Canviar dosis base
Precondició	S'estan consultant un pacient després de cercar-lo a la base de dades.
Flux principal	Seqüència: - El metge clica a l'enllaç “Consultar estadístiques “ - Es presenten les dades dades estadístiques calculades pel sistema.
Flux alternatiu	
Postcondició	Les dades estadístiques han estat calculades i presentades.

Cas d'ús:	Consultar gràfic.
Descripció	Es vol tenir un gràfic amb la evolució dels valors de les glucèmies per franja del dia
Actors	Especialista
Casos d'us relacionats	Consultar estadístiques, Canviar dosis base
Precondició	S'estan consultant les estadístiques de les glucèmies d'un pacient.
Flux principal	Seqüència: - L'especialista clica a l'enllaç “Gràfic últims 15 dies” - Es presenta el gràfic calculat pel sistema.
Flux alternatiu	
Postcondició	El gràfic ha estat calculat i presentat.

Cas d'ús:	Introduir controls
Descripció	El pacient vol introduir els valors de les glucèmies practicades.
Actors	Pacient
Casos d'us relacionats	
Precondició	
Flux principal	Seqüència: El pacient clicka l'enllaç “Introduir controls”. Apareix altre menú amb 2 opcions més per introduir-los de forma individual o diaria. - Si tria de forma diaria: surt un camp desplegable per triar el dia i surten 4 camps més per introduir el valors de les glucèmies al llarg del día amb 4 camps més per introduir la hora. - Si tria “individual”: surt un camp per introduir el control, un camp per introduir la hora i un camp desplegable per triar el dia. - Una vegada introduïts clicka “Acceptar” - Si tot es correcte el sistema dona un missatge de confirmació
Flux alternatiu	

Postcondició	El valors del les glucèmies han estat emmagatzemats a la base de dades del sistema.

Cas d'ús:	Introduir dosis
Descripció	El pacient introdueix el valors de les dosis administrades pel pacient i la data i hores a les que ho fà.
Actors	Pacient.
Casos d'us relacionats	
Precondició	El pacient vol introduir les dosis.
Flux principal	Seqüència: El pacient clicka l'enllaç “Introduir dosis”. Apareix altre menú amb 2 opcions més per introduir-las de forma individual o diaria. - Si tria de forma diària: surt un camp desplegable per triar el dia i surten 4 camps més per introduir el valors de les dosis administrades al llarg del dia amb 4 camps més per introduir la hora d'administració. - Si tria “individual”: surt un camp per introduir la dosi, un camp per introduir la hora i un camp desplegable per triar el dia. - Una vegada introduïdes clicka “Acceptar” - Si tot es correcte el sistema dona un missatge de confirmació
Flux alternatiu	
Postcondició	El valors del les dosis administrades han estat emmagatzemats a la base de dades del sistema.

Cas d'ús:	Consultar dosi base.
Descripció	El pacient pot consultar a la base de dades la dosi base assignada per l'especialista.
Actors	Pacient.
Casos d'us relacionats	
Precondició	El pacient vol consultar la dosi base assignada pel especialista i es troba a la pàgina assignada.
Flux principal	Seqüència: - Clica “Consultar dosi base” - El sistema retorna les unitats de les dosis base, el tipus d'insulina de cadascuna i la pauta d'utilització d'aquesta.
Flux alternatiu	
Postcondició	Les dosis base del pacient son mostrades per pantalla.

4.3.2. Diagrames de casos d'us.

-Diagrama de casos d'us de l'administrador:

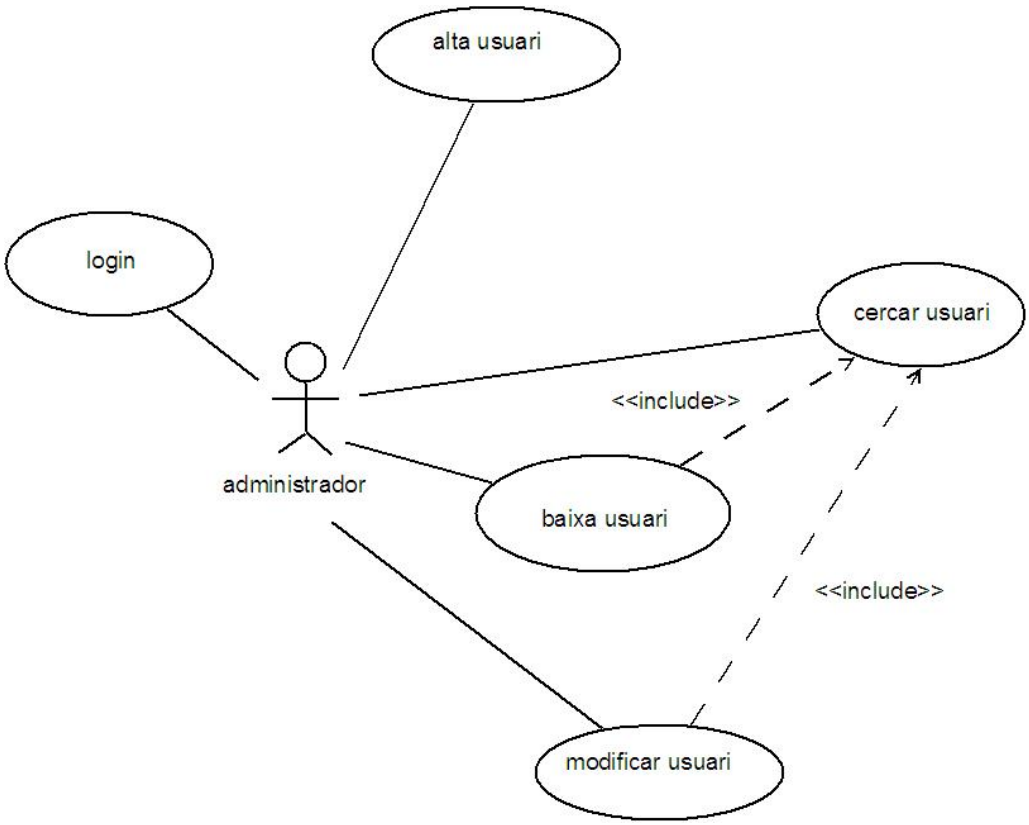


Figura 2

-Diagrama de casos d'us de cap.

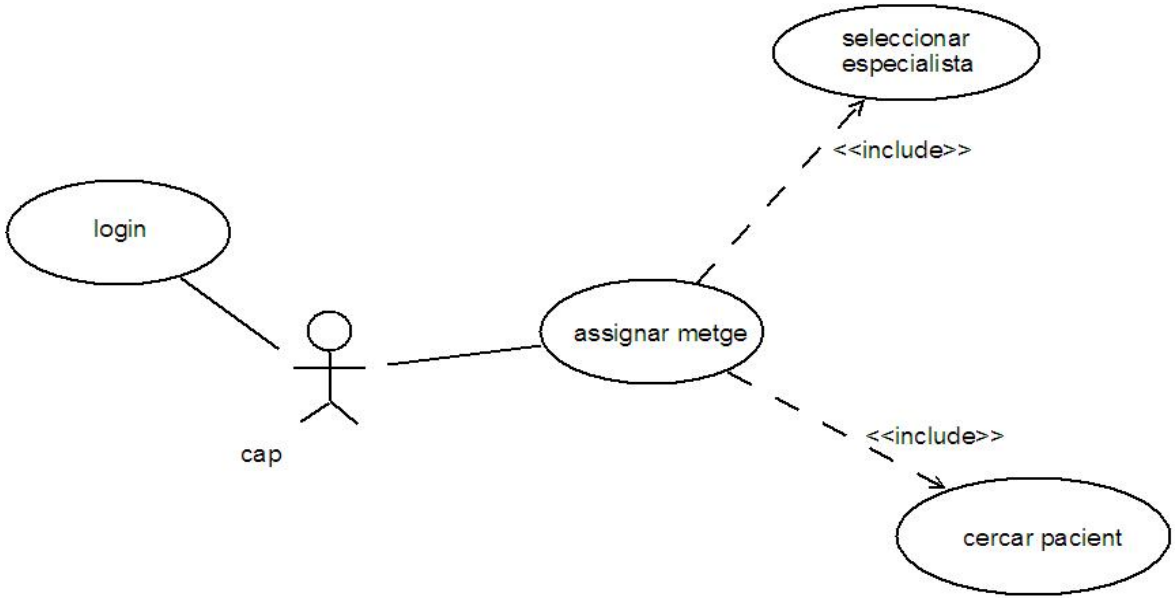


Figura 3

-Diagrama de casos d'us del pacient.

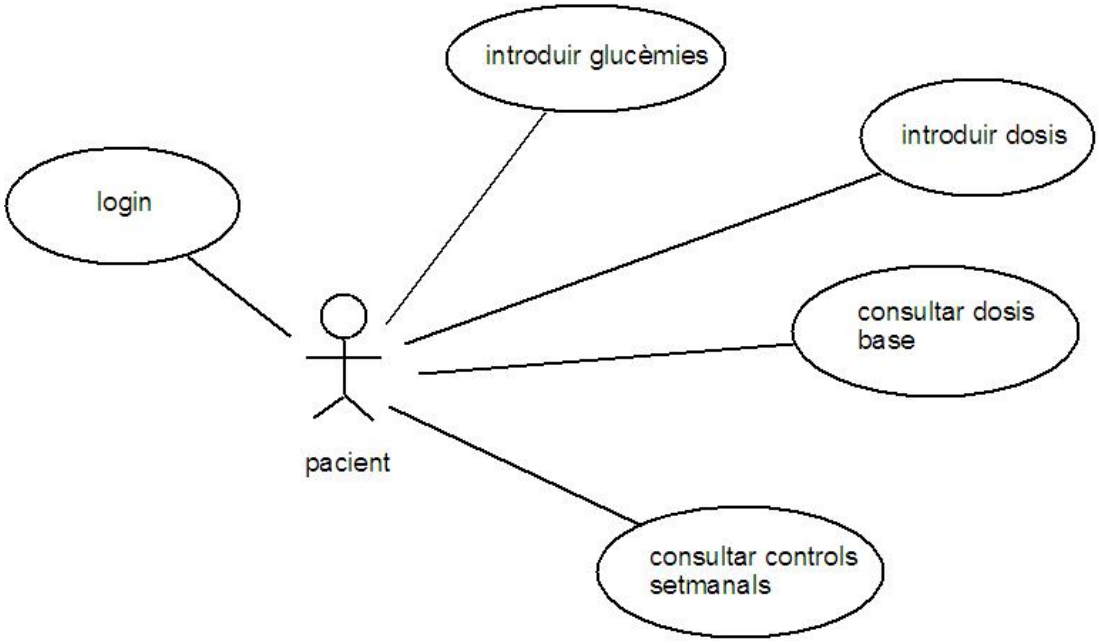


Figura 4

-Diagrama de casos d'us de especialista

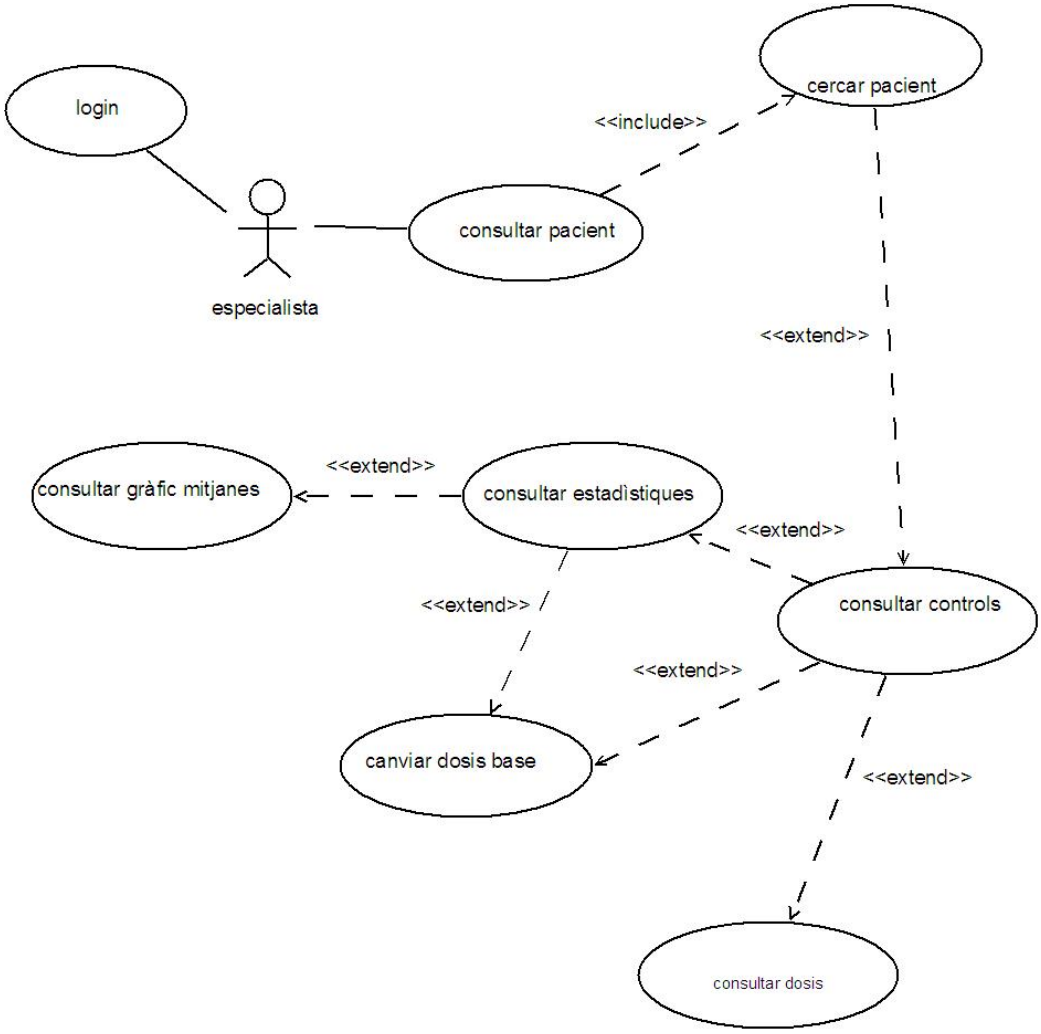


Figura 5

4.4. Classes derivades del anàlisi..

De l'anàlisi dels requisits i dels casos d'us detectem les entitats participants.

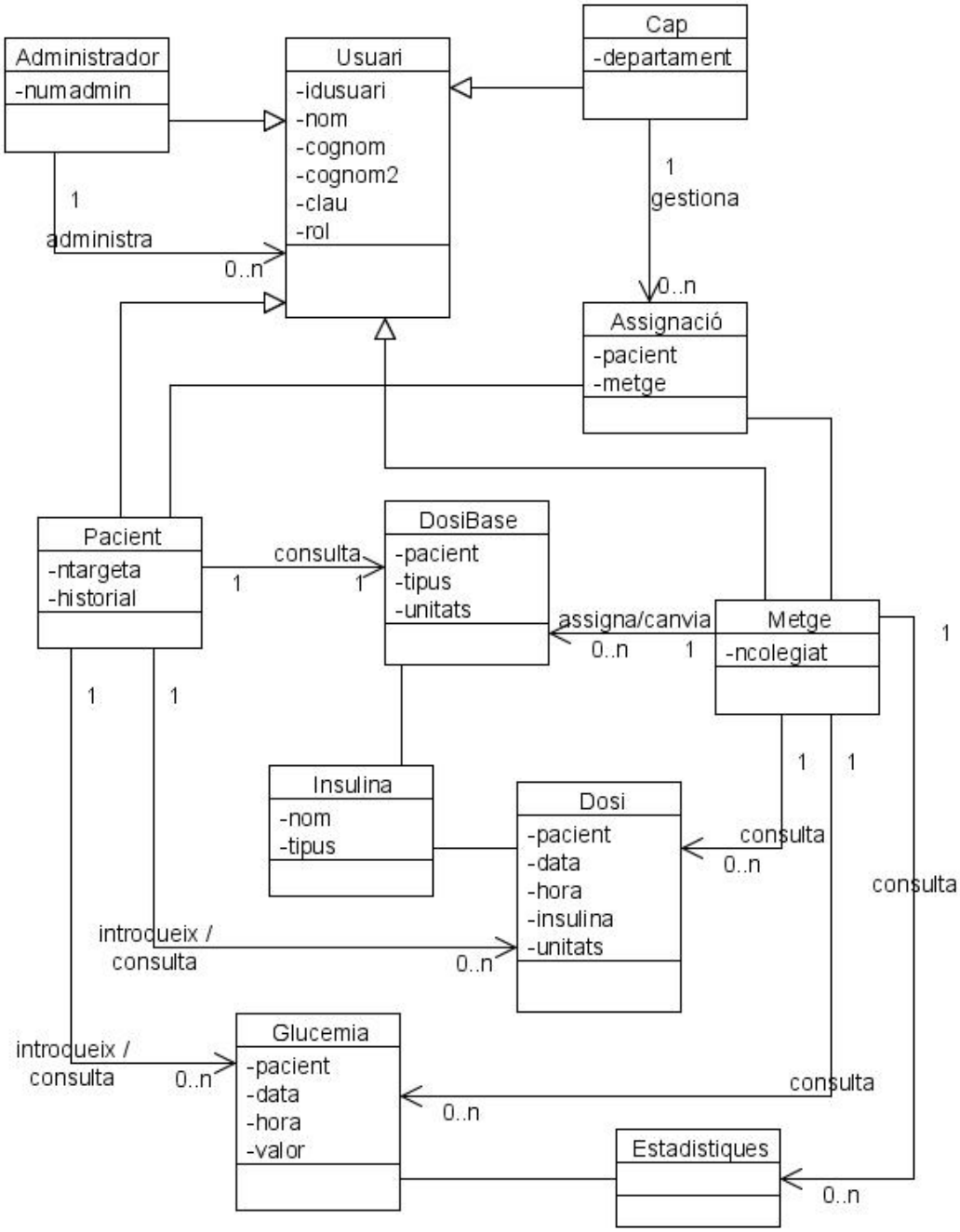


Figura 6 Diagrama de classes

5. Disseny.

5.1. Disseny de la interfície d'usuari..

Captures de les principals pantalles de la interfície gràfica.

Pantalla Login:

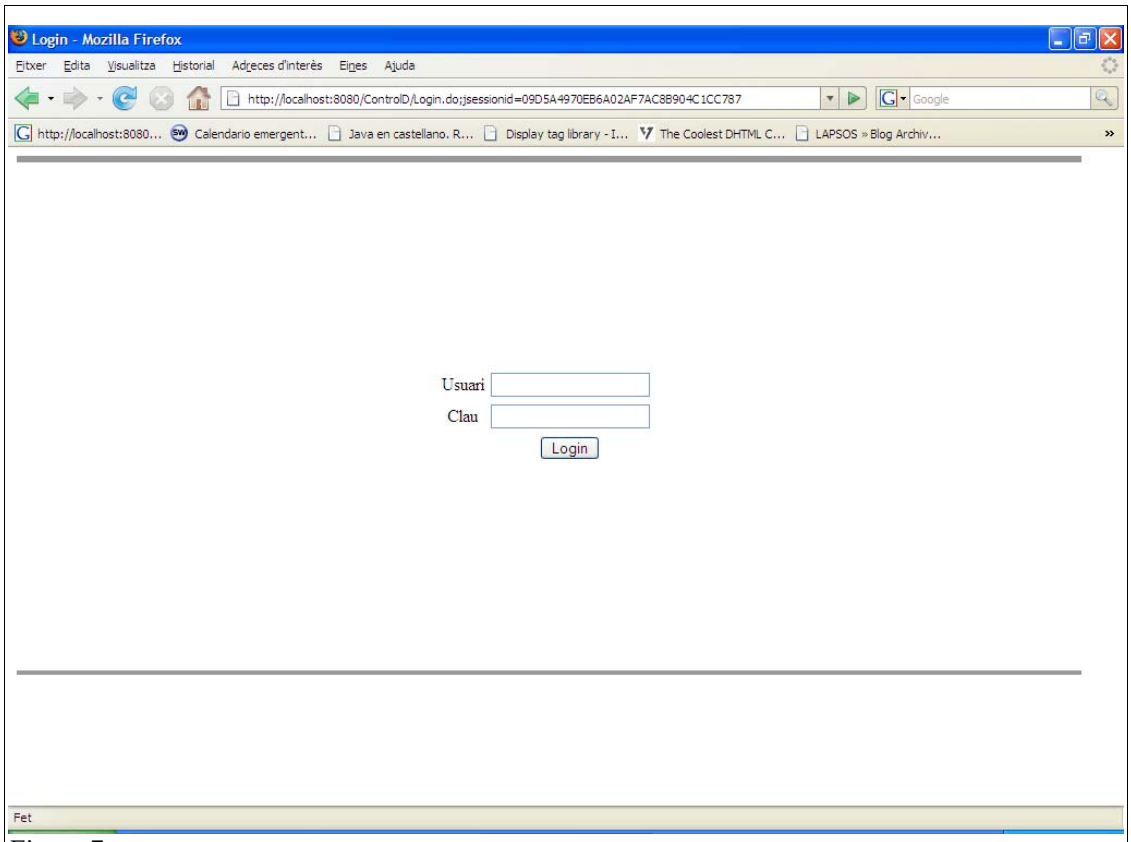


Figura 7

Menu Administrador:

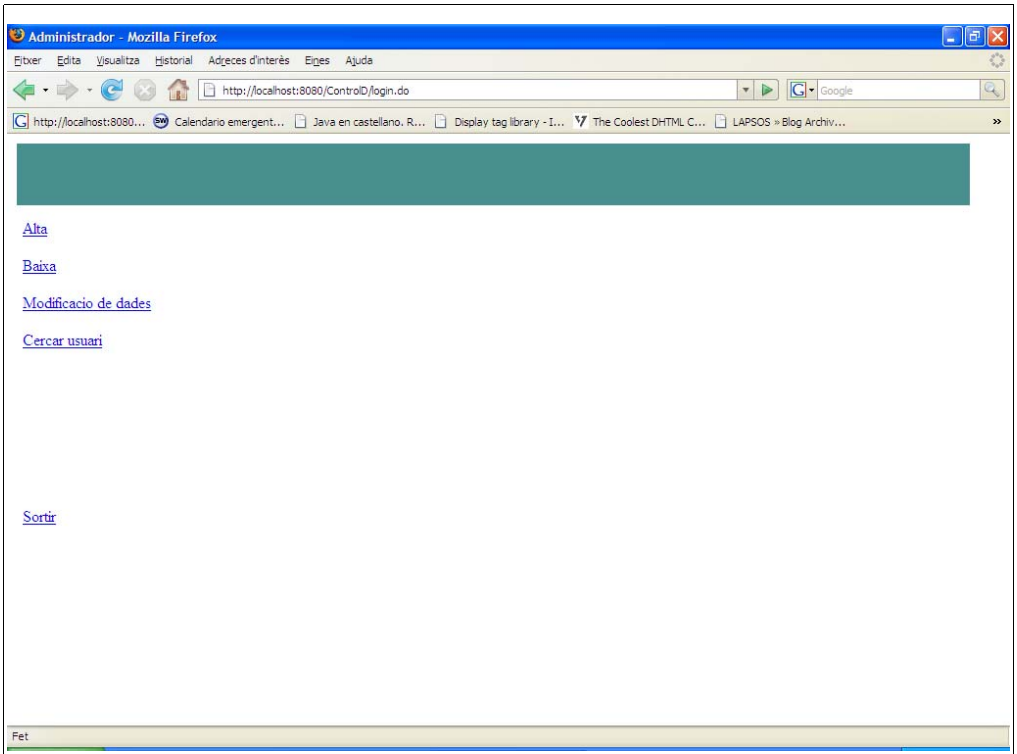


Figura 8

Menu Pacient:

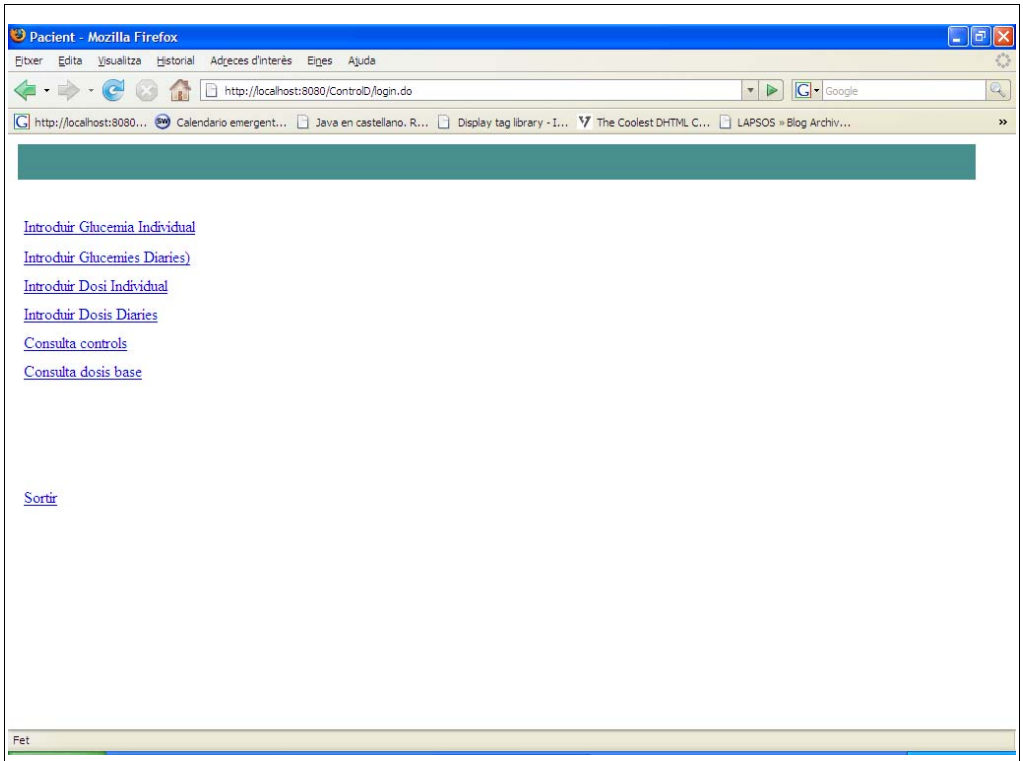


Figura 9

Menú Especialista:

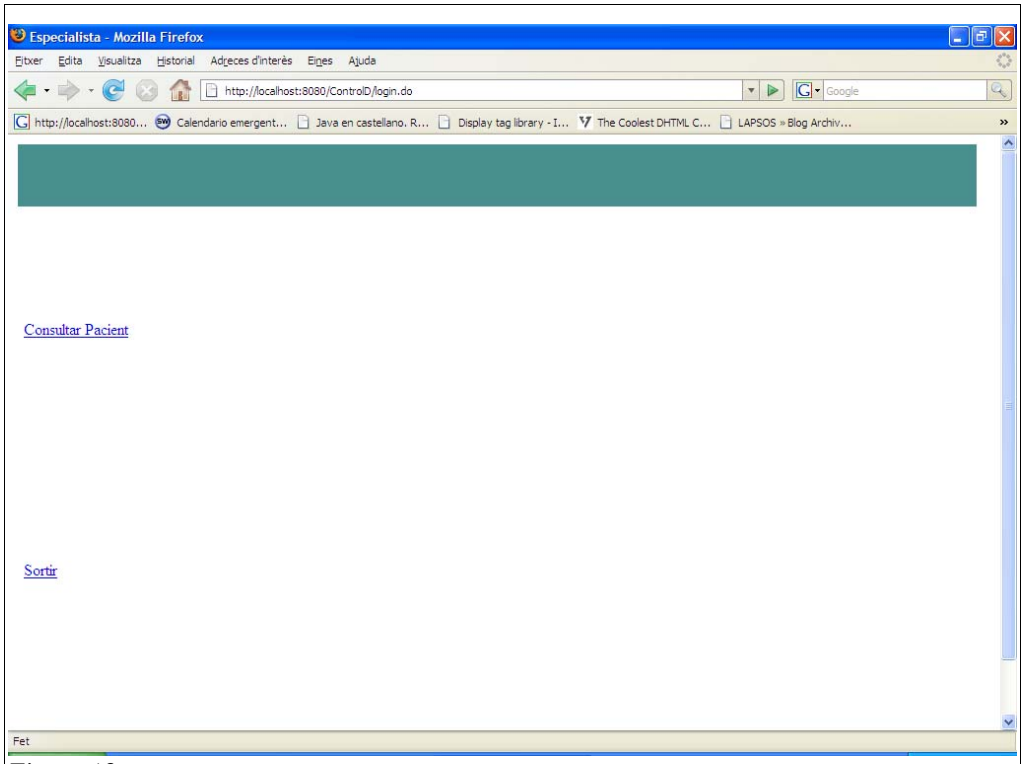


Figura 10

Menú Cap:

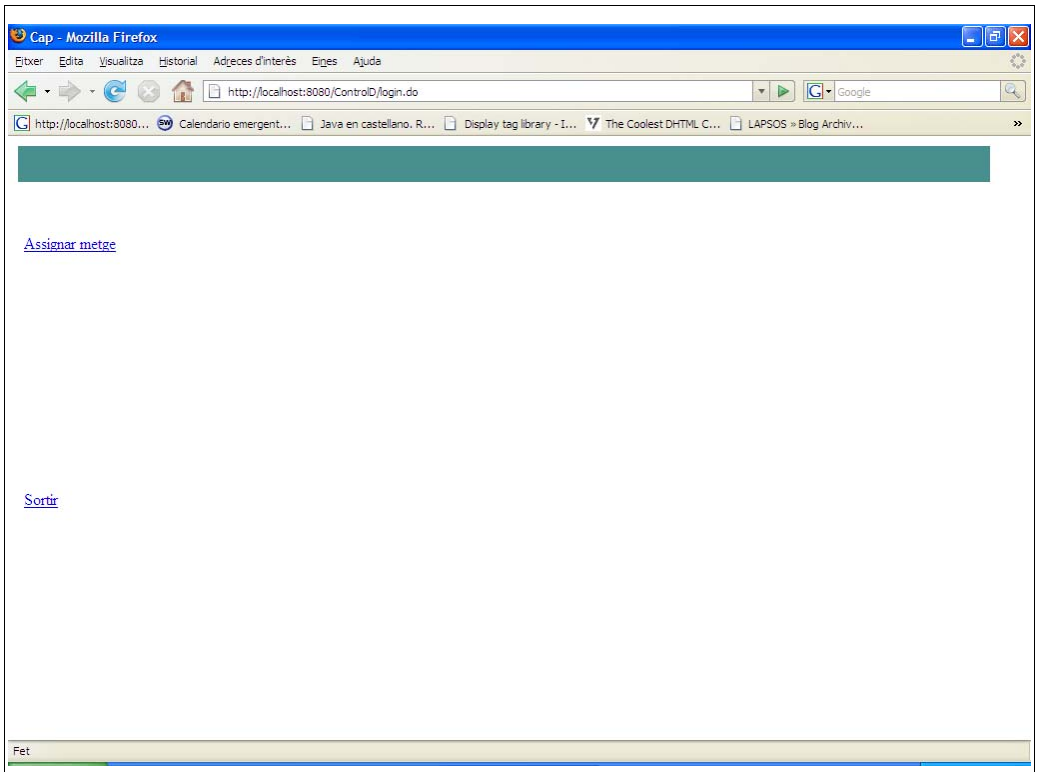


Figura 11

Pantalla d'alta d'usuari:

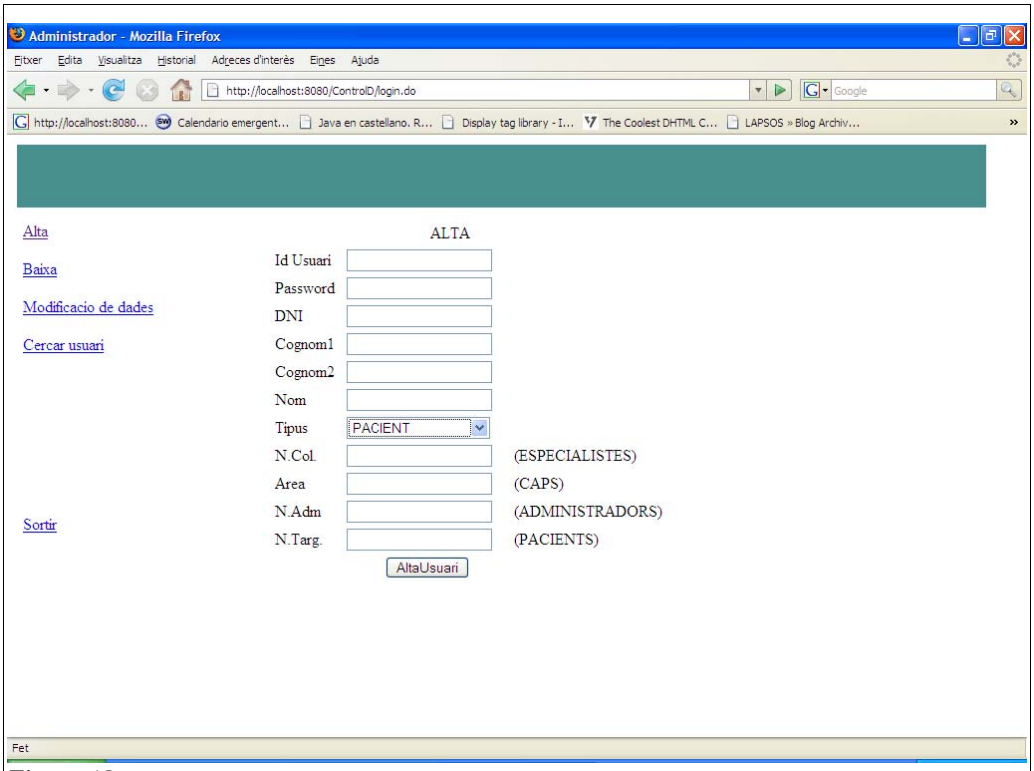


Figura 12

Pantalla de consulta de controls (Especialista)

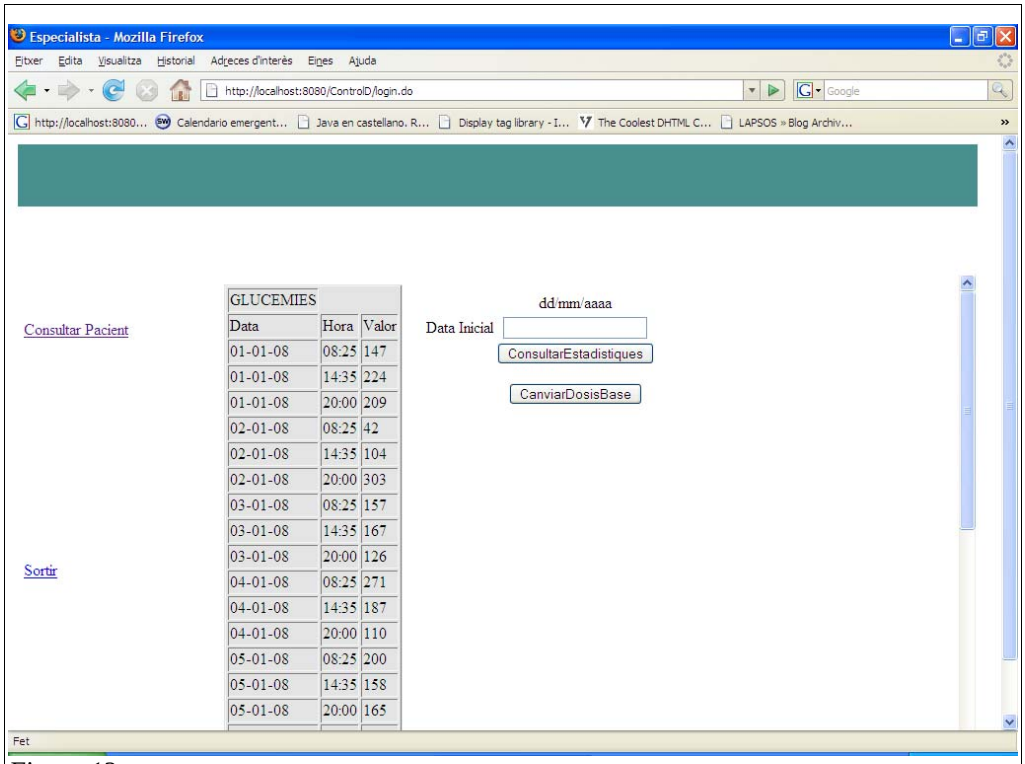


Figura 13

Pantalla d'estadístiques:

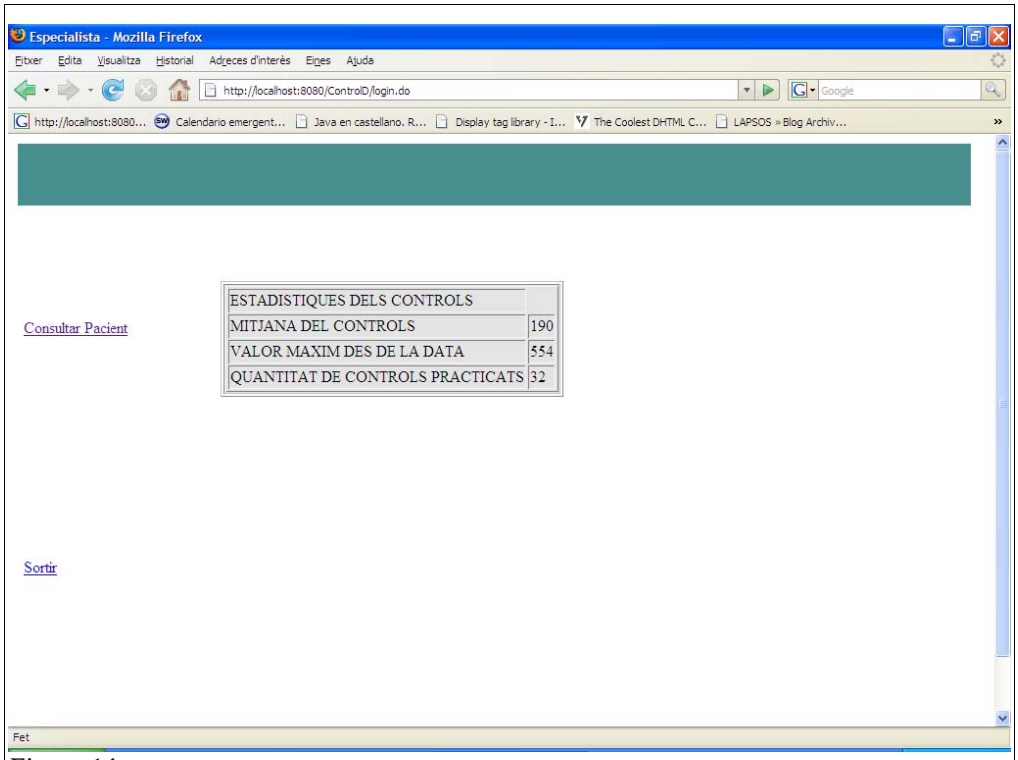


Figura 14

Pantalla d'assignació d'especialista:

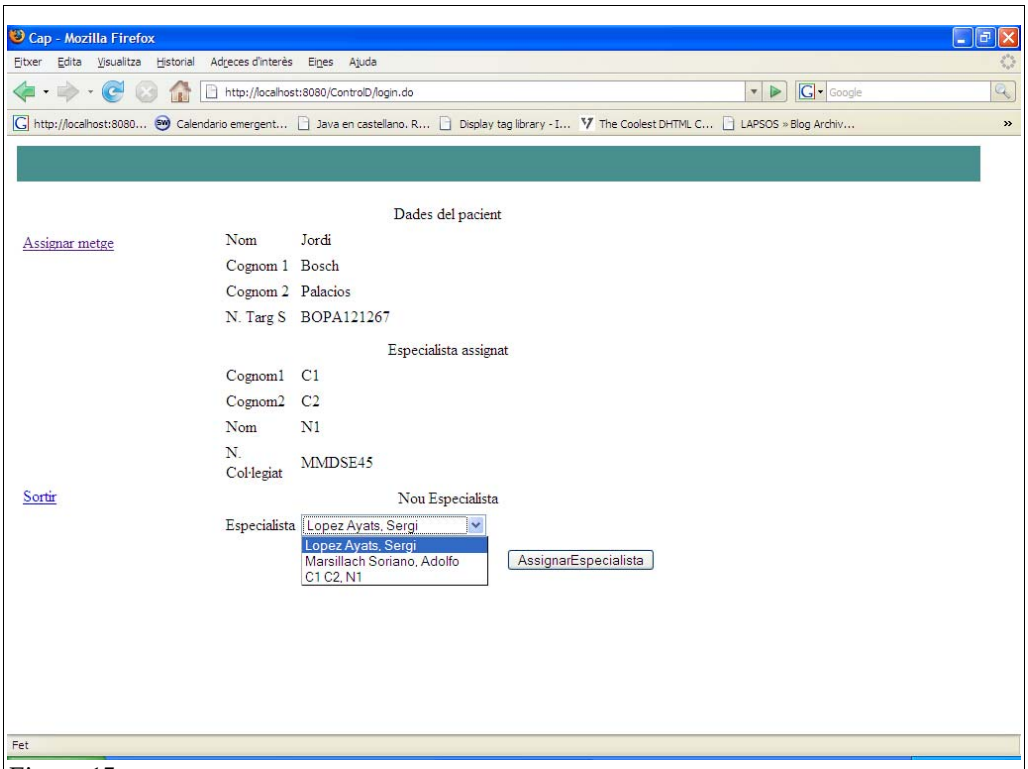


Figura 15

Pantalla d'introducció de glucèmies diàries

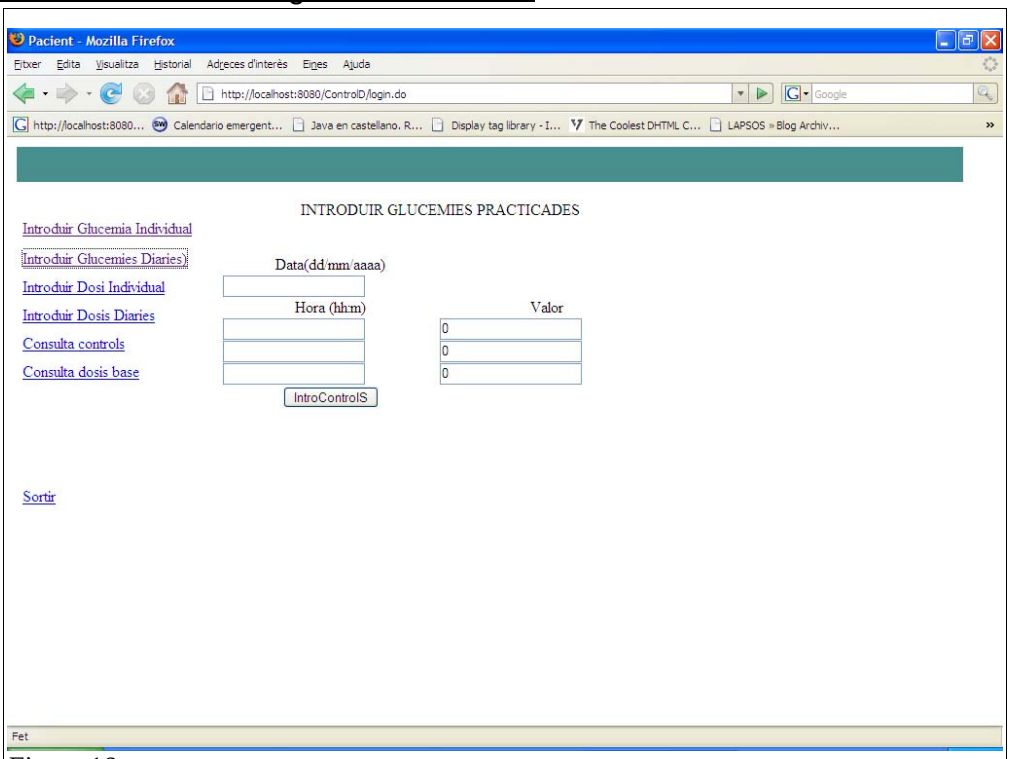


Figura 16

Pantalla d'introducció de dosis diàries:

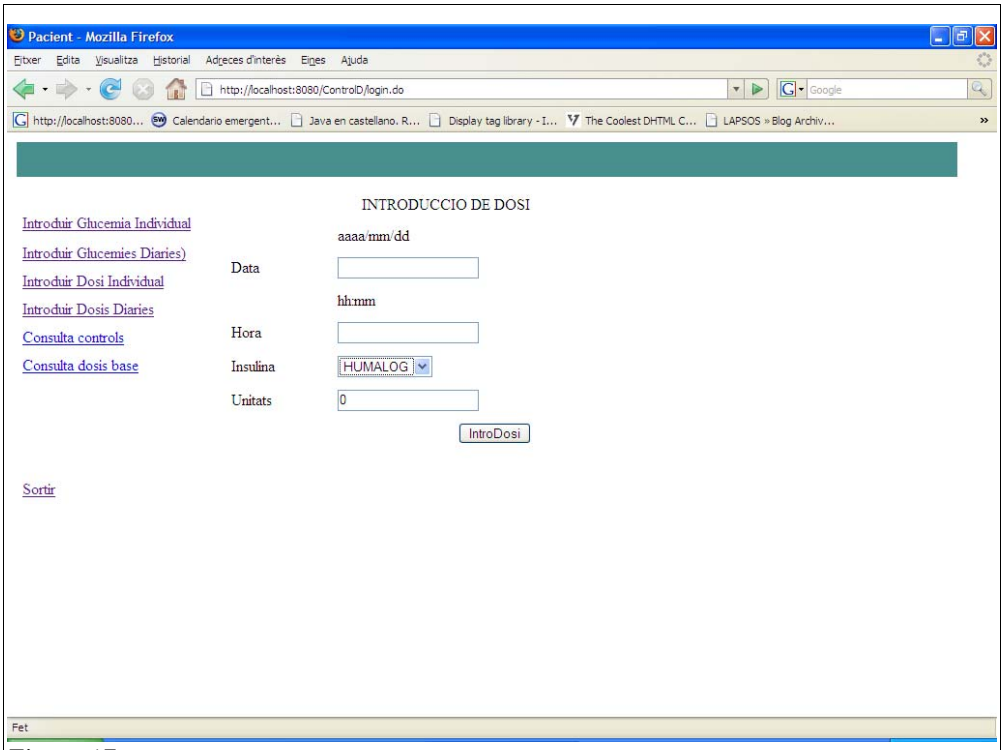


Figura 17

Consultar dosis base:

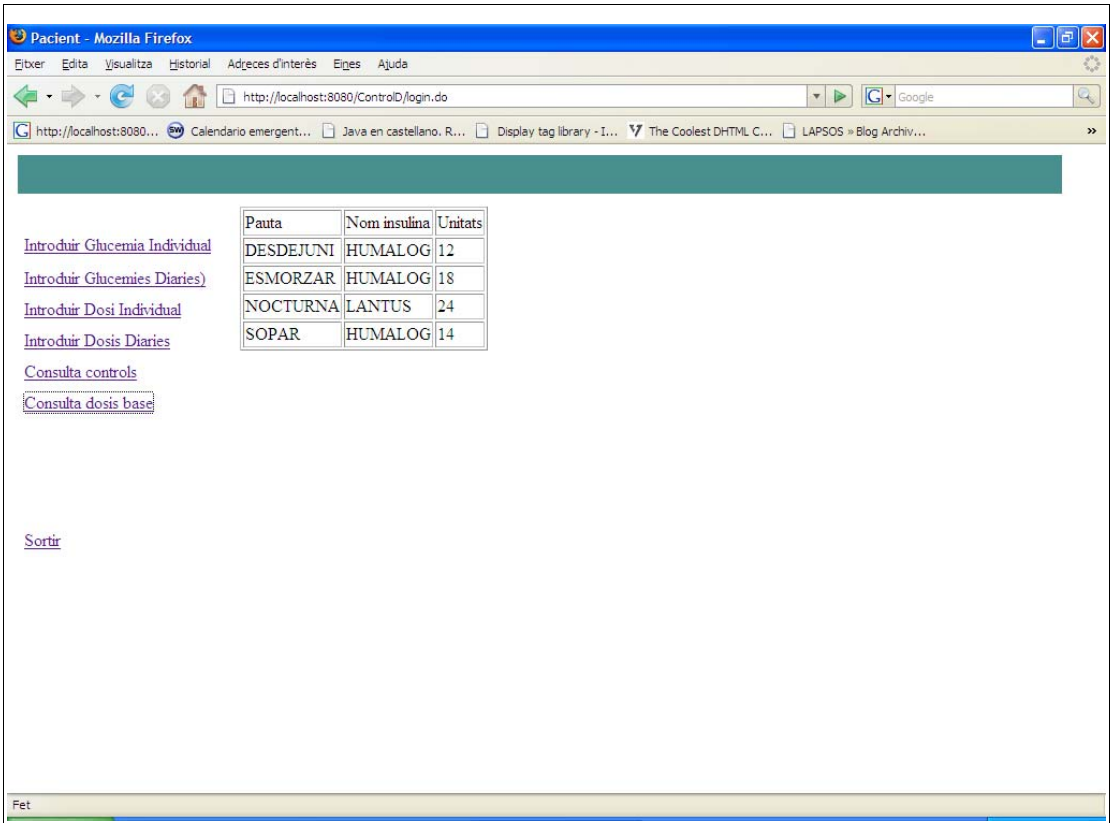


Figura 18

5.2. Disseny de la persistència.

De l'anàlisi del diagrama de classes detectem les entitats que han de ser persistents:

USUARI(idusuari, dni, cognom1, cognom2, nom,, password)

ADMINISTRADOR(numAd)
Entitat subclasse d'usuari.

METGE(numeroColegiat)
Entitat subclasse de metge.

CAP(area)
Entitat subclasse d'usuari.

PACIENT(numtargeta, numhistorial)
Entitat subclasse d'usuari.

INSULINA(nom, tipus)

GLUCEMIA(data, hora, valor)

DOSI(data, hora, unitats)

DOSISBASE(data, hora, unitats)

DATA()

HORA()

El diagrama entitat-relació resultant es el següent:

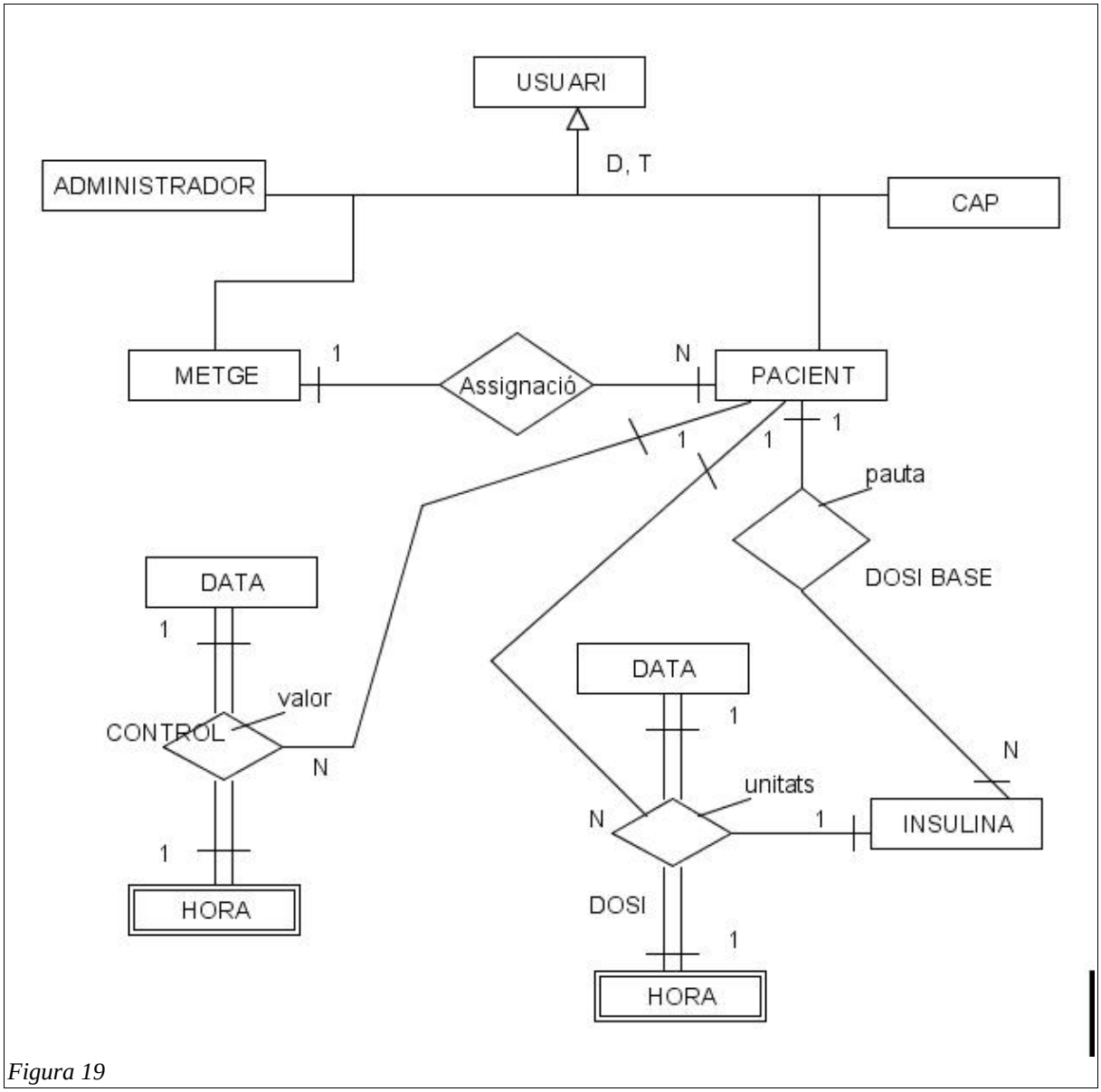


Figura 19

Transformació al model relacional:

USUARI(dni, cognom1, cognom2, nom, <u>idusuari</u>, password)

ADMINISTRADOR(<u>idusuari</u>, numeroAdmin)

on idusuari referència USUARI

METGE(idusuari, numeroColegiat)

on idusuari referència USUARI

METGE-CAP(idusuari, l area)

on idusuari referència USUARI

METGE(idusuari, especialitat)

on idusuari referència USUARI

PACIENT(idusuari, ntarg)

on idusuari referència USUARI

DOSIBASE(nom, ntarg, pauta, unitats)

on nom referència nom a INSULINA
on ntarg referència ntarg a PACIENT

GLUCEMIA(dni,data,hora,valor)

on dni referència PACIENT
on data referència data
on hora referència hora

INSULINA(nom, tipus)

DOSI(ntarg,nom,data,hora,unitats,)

on nom referència nom a INSULINA
on ntarg referència ntarg a PACIENT
on data referència data
on hora referència hora

Subsistemes.

Dividim el sistema en subsistemes tenint en compte la proximitat de les funcionalitats segons els rols.

- Subsistema de connexió.

  Aquest subsistema te les funcionalitats necessàries per identificar i validar l'usuari dins el sistema i posteriorment posar a la seva disposició els recursos del sistema assignats segons el seu rol dins d'aquest.

- Subsistema d'administració.

  Les funcionalitats incloses dins aquest subsistema son les de gestió dels usuaris del sistema: altes i baixes d'usuaris i modificació de les dades bàsiques.

- Subsistema d'estadistiques.

  Creació i consulta de les estadístiques, gràfics i informes que es poden generar amb les dades dels pacients.

- Subsistema de gestió.

  Assignació o canvi del metges als pacients per part del cap del servei.

- Subsistema de dades.

  Totes les funcionalitats relacionades amb la introducció i consulta de les dades introduïdes pels pacients i el metges.

Diagrama amb els susbsistemes:

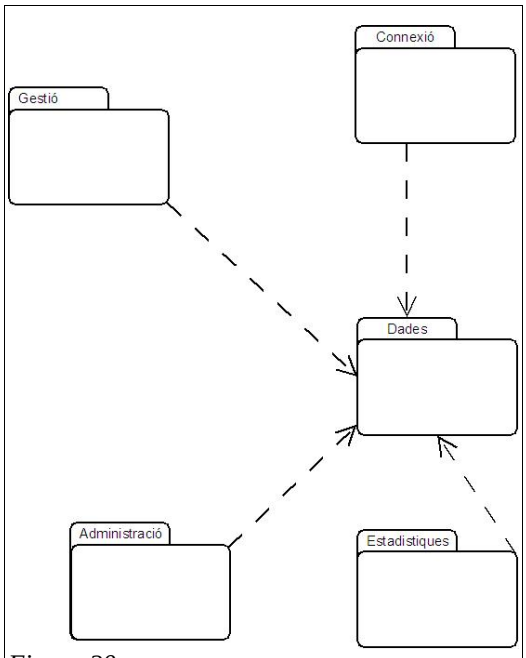


Figura 20

Tots aquest subsistemes formaran packages, a la fase d'implementació.

6. Arquitectura.

6.1. Aplicació en 3 capes.

Aquest plantejament divideix la aplicació en tres parts: una capa per guardar les dades, una que centralitza tota la lògica de negoci i un altra que conté la interfície gràfica que interacciona amb els usuaris.

La utilització d'aquesta arquitectura per a la construcció d'aplicacions web aporta molts avantatges entre el quals podem destacar el següents:

- Senzillesa dels clients, que normalment son navegadors web.
- Major seguretat, ja que les dades son manejades per la capa de negoci (model), descarregant els navegadors d'aquesta responsabilitat.
- Al tenir la lògica de negoci centralitzada, les possibles modificacions o millores sobre aquesta nomes cal fer-les una sola vegada.
- Facilitar la implementació de la interfície sense tenir coneixement de la resta de la aplicació.

L'esquema gràfic es el següent:

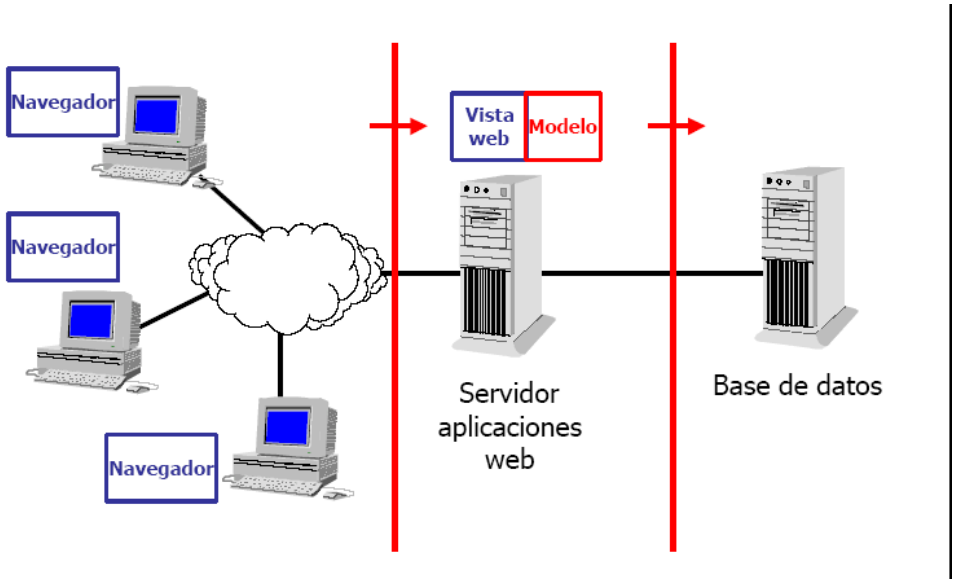


Figura 21

En aquest cas la capa model queda integrada en al mateix servidor que la capa vista però funcionalment son capes separades. HI ha un arquitectura alternativa que separa aquestes capes en diferents servidors.

6.2. Patró MVC .

El patró arquitectònic MVC es un dels mes utilitzats en la construcció d'aplicacions Web. Aquest patró permet la implementació d'aplicacions en 3 capes, separant el codi de cadascuna, facilitant el seu disseny, desenvolupament, modificació i posterior manteniment.

Una aplicació que segueix el patró MVC es composa de 3 parts fonamentals:

-Vista. Es la part responsable de la presentació de les dades i que interacciona amb l'usuari.

-Controlador. Es la part que rep els events d'entrada des de la Vista i encarregada d'executar les accions associades a aquests events, actuant, si es necessari, sobre el model.

-Model. Es la part encarregada de la lògica de negoci i d'accedir a les dades.

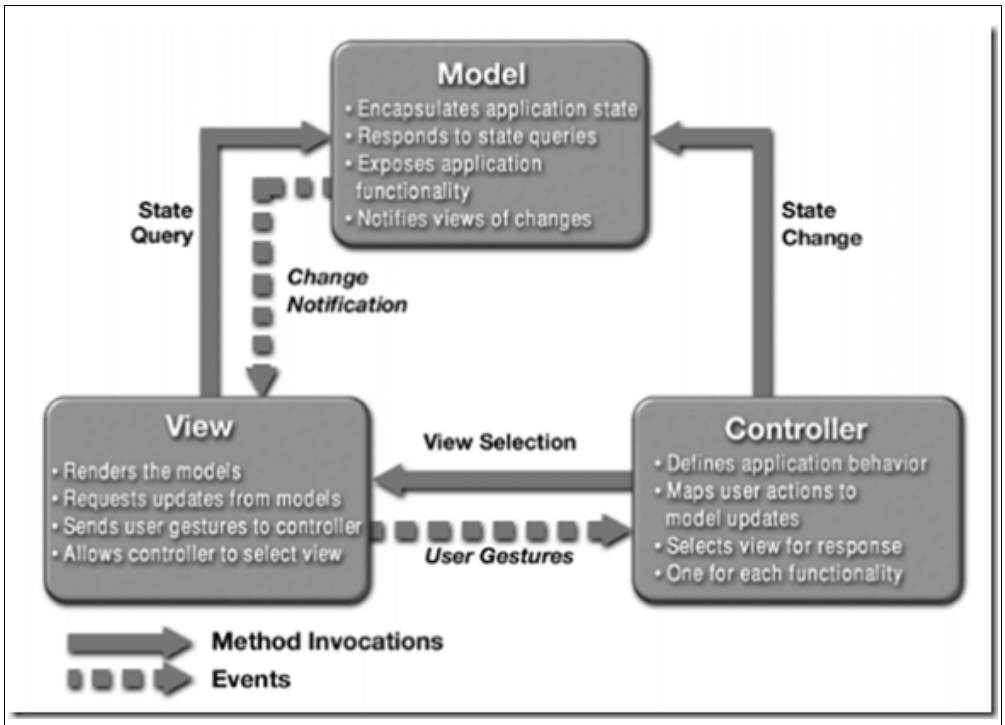


Figura 22

Amb aquest esquema podem aïllar els canvis que es poden fer posteriorment a la aplicació, com poden ser ampliacions de funcionalitats, actualitzacions de programari o maquinari, etc., degut al baix acoblament que hi ha entre la lògica de negoci i les dades, de la lògica de presentació.

Aquesta arquitectura es ideal pel desenvolupament d'aplicacions web per capes. També es de fàcil integració amb tecnologies J2EE.

El model que recomana Sun per a la implementació d'aplicacions web en plataformes J2EE s'anomena Model 2. En aquest s'especifica la implementació del controlador en un únic servlet, que centralitzi totes les peticions fetes des dels client. El disseny d'aquest controlador segueix el patró *Front Controller*.

Patró Front Controller.

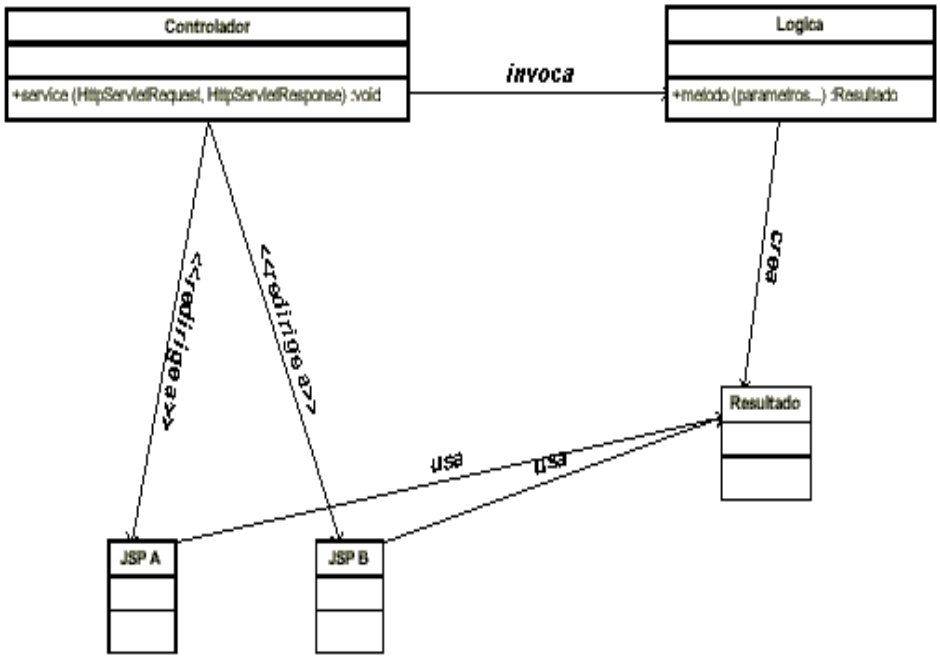


Figura 23

Aquest patró centralitza totes les peticions fetes pels clients, comprovacions de seguretat, logs. També evitar tenir que repetir aquest codi per a cada acció.

Diagrama de seqüència del patró *Front Controller* :

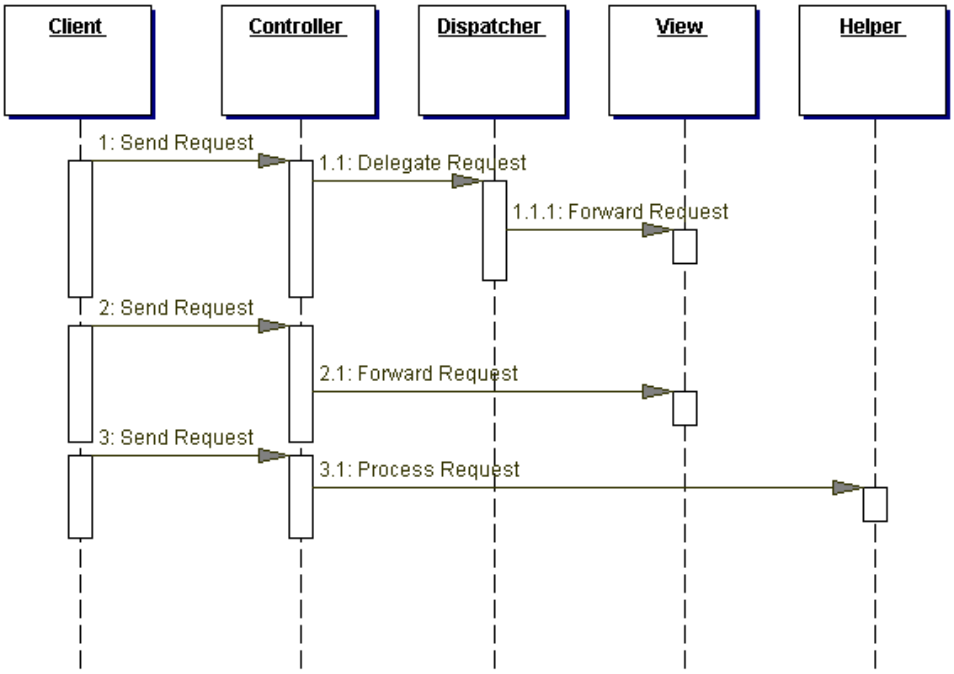


Figura 24

7. Implementació.

7.1. Característiques.

La implementació es basa fonamentalment en aquest trets:

a) Utilització del framework Struts per a la implementació del patró arquitectònic MVC. Hi han molts framework que implementen aquest patró (Spring, Struts, Tapestry,..) però Struts és el més utilitzat i del que hi ha més documentació i exemples.

Diagrama d'aplicació amb Struts.

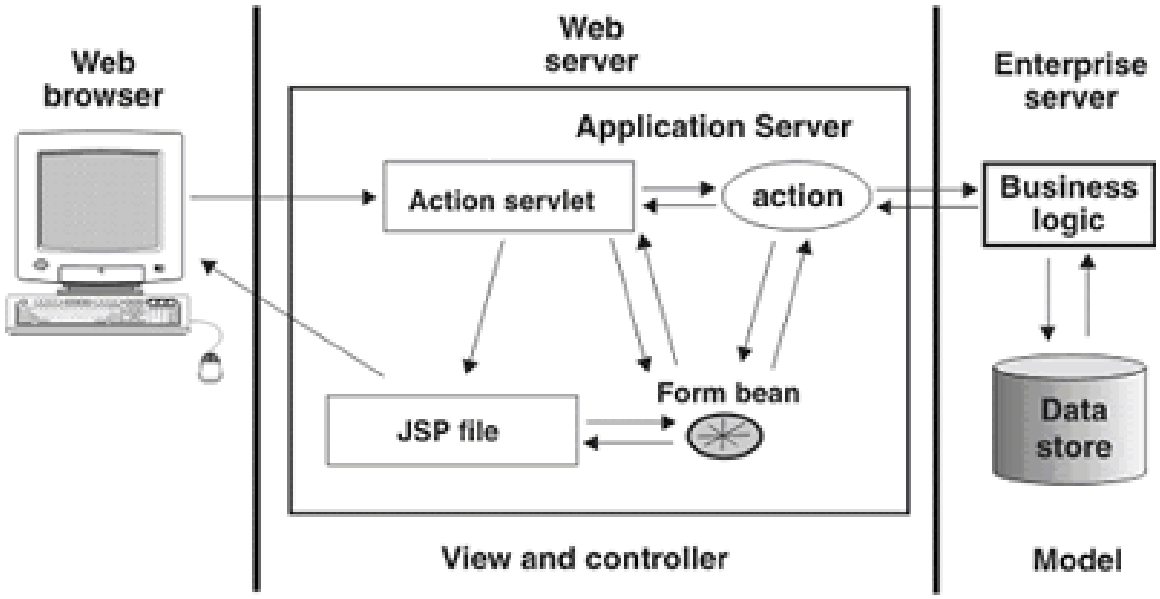


Figura 25

b) Utilització d'Hibernate com a motor de persistència. Hibernate es un ORM (object-relational mapping) que ens permet fer una abstracció de la implementació de la base de dades i treballar la persistència directament amb objectes, conservant totes les característiques del paradigma OO. Es fàcilment configurable mitjançant els arxius xml configuració i de mapeig.

Esquema d'aplicació amb Hibernate:

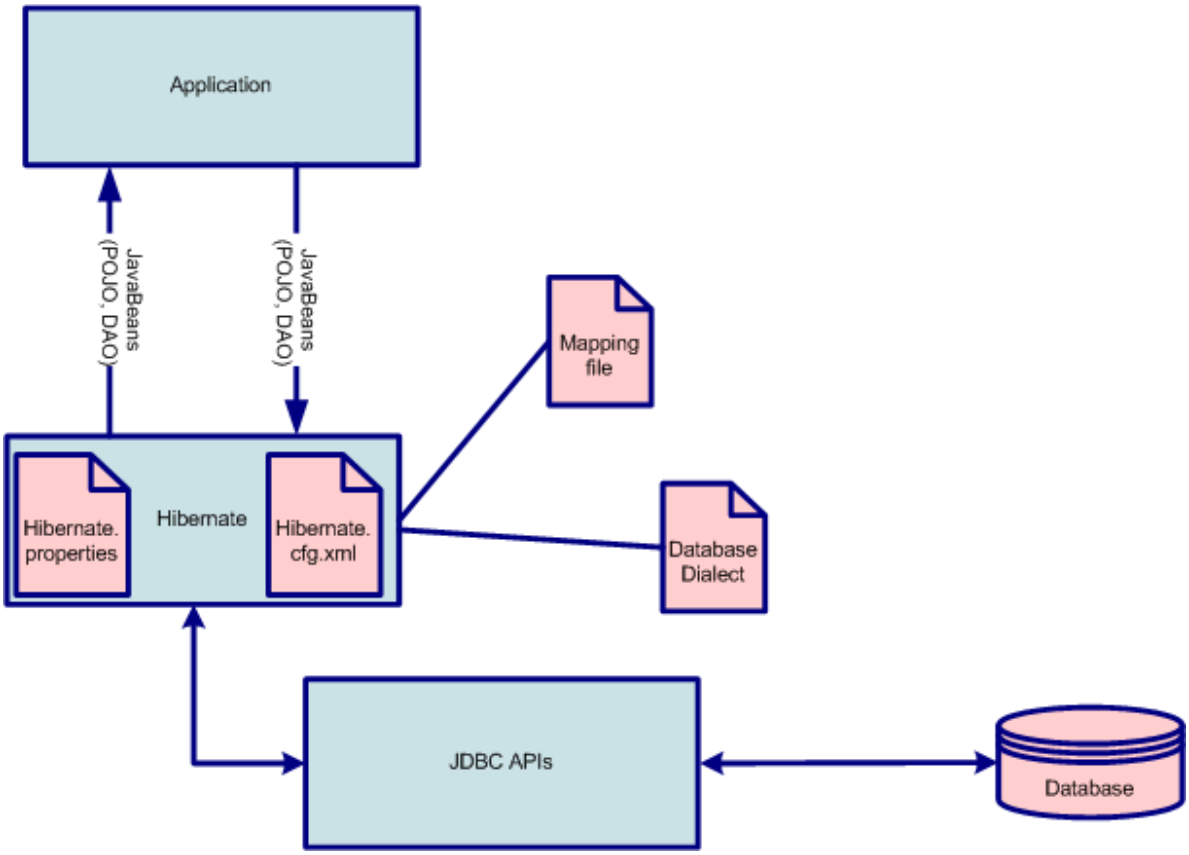


Figura 26

c) Utilització de pàgines JSP amb HTML, XML i tags de les llibreries d'Struts per a la creació dels formularis.

d)Utilització de classes simples, POJO (Plain Old Java Object), com a objectes de negoci. Aquest, a diferencia de EJB, no implementen cap interface, nomes porten definits els seus atributs i el corresponents mètodes getters/setters i faciliten la definició del mapeig amb Hibernate.

7.2. Implementació del la Vista..

La principal part de la Vista son les pàgines JSP, aquestes estan implementades amb les llibreries de tags JSP, tant les estàndard com les proporcionades pel mateix Struts, html i codi java.

També son part molt important d'aquesta capa els Form Beans o formularis, aportats per Struts, els quals la seva missió es la de transferir les dades entre la Vista i el Controlador, fent de buffer entre elles. També permeten validar les dades d'entrada abans de que arribin a la corresponent acció i fer reset sobre els camps del formulari.
Les classes que implementes el Form Beans hereten d'ActionForm. Aquestes s'implementen declarant el atributs que es volen transferir entre la vista i el controlador i els corresponents mètodes getters/setters.

Exemple d'ActionForm:

```
package cd.administracio;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;


public class CercarUsuariForm extends ActionForm{


		private String idusuari = null;
		private String dni = null;

		public String getDni() {
				return dni;
		}

		public void setDni(String dni) {
				this.dni = dni;
```

```
		}

		public String getIdusuari() {
			return idusuari;
		}

		public void setIdusuari(String idusuari) {
			this.idusuari = idusuari;
		}

		public void reset(ActionMapping mapping, HttpServletRequest request){

			this.idusuari = null;
			this.dni = null;
		}

		public ActionErrors validate(ActionMapping mapping,
						HttpServletRequest request){

			ActionErrors errors=new ActionErrors();

			if((idusuari == null) && (dni == null))
						errors.add("usuari", new ActionMessage("error.usuari.requerit"));


			return errors;
		}


}
```

Y els corresponents camps html:

```
<%@ taglib uri="/tags/struts-bean" prefix="bean"%>
<%@ taglib uri="/tags/struts-html" prefix="html"%>
<%@ taglib uri="/tags/struts-logic" prefix="logic"%>
<html:html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Cercar Usuari</title>
</head>
<body>
<table style="text-align: left; width: 463px; height: 270px;" border="0" cellpadding="2"
cellspacing="2">
  <tbody>
    <tr>
      <td style="text-align: center;" colspan="3" rowspan="1"> CERCAR USUARI</td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
   <html:form action="/cercarUsuari.do">
    <tr>
      <td style="width: 10%;"></td>
      <td>Id Usuari</td>

      <td><html:text property="idusuari" /></td>


    </tr>
```

```
    <tr>
      <td></td>
      <td>DNI</td>

      <td><html:text property="dni" /></td>

    </tr>
    <tr>
      <td></td>
      <td></td>
      <td style="text-align: center;"><html:submit value="CercarUsuari" /></td></td>
    </tr>
   </html:form>
  </tbody>
</table>
<br>
</body>
</html:html>
```

Struts defineix 2 tipus de formularis: ActionForm i DinactionForm. El primer es una representació de les dades que es transfereixen des de o cap formulari html. El segon permet declarar de form dinàmica els ActionForm. En aquest projecte només s'han utilitzat del primer tipus.

Un ActionForm té 2 tipus d'abast: petició i sessió (request / session). Si l'abast és de petició ,els form-bean, que son la instància dels ActionForm, estarà disponible fins el final del cicle de petició/resposta HTTP. Si l'abast és de sessió, estaran disponibles durant tota la sessió i per tant per a més d'una petició.

Exemple de declaració de l'abast:

```
<!-- CONSULTAR ELS CONTROLS D'UN PACIENT -->
      <action
          path="/consultarControls"
          input="/WEB-INF/pages/especialista/DadesPacient.jsp"
          name="consultarControlsForm"
          scope="request"
          type="cd.dades.especialista.ConsultarControlsAction"
          validate="false">
          <forward name="mostrarControls" path="/WEB-    INF/pages/especialista/
ResultConsultarControls.jsp" />
</action>
```

Declaració dels form-beans al arxiu struts-config.xml:

```
<form-beans>
           <form-bean name="loginForm" type="cd.connexio.LoginForm" />
           <form-bean name="cercarUsuariForm" type="cd.administracio.CercarUsuariForm" />
           <form-bean name="altaUsuariForm" type="cd.administracio.AltaUsuariForm" />
           <form-bean name="cercarPacientForm" type="cd.gestio.CercarPacientForm" />
           <form-bean name="assignarEspecialistaForm" type="cd.gestio.AssignarEspecialistaForm" />
           <form-bean name="cercarPacientEForm" type="cd.dades.especialista.CercarPacientEForm" />
```

```
            <form-bean name="consultarDosisBaseForm"
type="cd.dades.pacient.ConsultarDosisBaseForm" />
            <form-bean name="consultarControlsForm"
type="cd.dades.especialista.ConsultarControlsForm" />
            <form-bean name="consultarControlsPForm"
type="cd.dades.pacient.ConsultarControlsPForm" />
            <form-bean name="introControlForm" type="cd.dades.pacient.IntroControlForm" />
            <form-bean name="introControlSForm" type="cd.dades.pacient.IntroControlSForm" />
            <form-bean name="introDosiForm" type="cd.dades.pacient.IntroDosiForm" />
            <form-bean name="introDosiSForm" type="cd.dades.pacient.IntroDosiSForm" />
            <form-bean name="modificarUsuariForm" type="cd.administracio.modificarUsuariForm" />
            <form-bean name="consultarEstadistiquesForm"
type="cd.estadistiques.ConsultarEstadistiquesForm" />
            <form-bean name="canviarDosisBaseForm" type="cd.dades.especialista.CanviarDosisBaseForm"
/>
</form-beans>
```

7.2. Implementació del Model.

Com ja s'ha comentat abans, hem utilitzat Hibernate, un framework ORM (Object-Relational Mapping), que facilita treballar la persistència directament amb objectes. Hibernate fa un mapeig entre els objectes de negoci i les taules de la base de dades, suportant totes les característiques del paradigma OO: herència, polimorfisme,...

La configuració d'Hibernate es fa mitjançant el fitxer *Hibernate.properties* , o com es el cas d'aquest projecte *Hibernate.cfg.xml*.

Contingut del arxiu *Hibernate.cfg.xml* :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
                      "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
                      "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory name="SF_01">
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.password">00000000</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost/bdsugar</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>


        <mapping resource="cd/dades/pojos/Usuari.hbm.xml"/>
        <mapping resource="cd/dades/pojos/DosisBase.hbm.xml"/>
        <mapping resource="cd/dades/pojos/Assignacio.hbm.xml"/>
        <mapping resource="cd/dades/pojos/Dosis.hbm.xml"/>
        <mapping resource="cd/dades/pojos/Insulina.hbm.xml"/>
        <mapping resource="cd/dades/pojos/Glucemia.hbm.xml"/>


    </session-factory>
```

```
</hibernate-configuration>
```

També es pot configurar un pool de connexions, si la concurrència a la bbdd es molt alta.

El mapeig entre els objectes de negoci i les taules de la base de dades també es configura fàcilment. Una vegada declarat el mapeig al arxiu de configuració, com es pot comprovar, es crea un arxiu, per a cada objecte, on queden indicades les correspondències entre el atributs del objecte i les columnes de la taula.

Exemple de mapeig:

Classe Insulina.java

```
package cd.dades.pojos;
public class Insulina implements Serializable {

            /**
             * El nom de la insulina
             */
            private String nom;

            /*
             * El tipus d'insulina
             */
            private String tipus;

            /**
             * @return Returns the nom.
             */
            public String getNom() {
                        return nom;
            }

            /**
             * @return Returns the tipus.
             */
            public String getTipus() {
                        return tipus;
            }

            /**
             * @param nom The nom to set.
             */
            public void setNom(String nom) {
                        this.nom = nom;
            }

            /**
             * @param tipus The tipus to set.
             */
            public void setTipus(String tipus) {
                        this.tipus = tipus;
            }
}
```

Arxiu de mapeig Insulina.hbm.xml:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
            "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
            "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="cd.dades.pojos">
  <class name="Insulina" table="insulina">
            <id name="nom" column="nominsulina" type="string">
                                    <generator class="assigned"/>
                        </id>
                        <property name="tipus" column="tipus" type="string" />

  </class>
</hibernate-mapping>
```

La lògica de negoci s'ha encapsulat dins una única classe, *GestorDades.java*, Aquesta classe conté tots els mètodes de cerca, actualització, eliminació i càlculs estadístics que son utilitzats des d'els Action de la capa Controlador.

Dins aquesta classe s'ha utilitzat la classe HibernateSessionFactory(1), que millora, utilitzant-la, la classe SessionFactory, proveïda per Hibernate. Aquesta classe assegura que només sigui utilitzada una instància de sessió per a cada thread en execució.

(1)Obtingut a http://www.lalilluna.de/

7.3. Implementació de la capa Controlador.

Struts ja porta implementat un controlador, ActionServlet. Es l'encarregat de rebre les peticions del client i redirigir-les adequadament. Aquest està implementat seguint el patró de disseny Front Controller, comentat anteriorment. ActionServlet centralitza totes les peticions que provenen del client. El servlet rep una serie de mapeigs, ActionMapping, els quals defineixen la acció a executar per a cada petició. Aquests mapeig es declaren a arxiu struts-config.xms

Exemple de mapeig:

```
<!-- CONSULTAR ELS CONTROLS D'UN PACIENT -->
      <action
          path="/consultarControls"
          input="/WEB-INF/pages/especialista/DadesPacient.jsp"
          name="consultarControlsForm"
          scope="request"
          type="cd.dades.especialista.ConsultarControlsAction"
          validate="false">
```

```
        <forward name="mostrarControls" path="/WEB-     INF/pages/especialista/
ResultConsultarControls.jsp" />
</action>
```

El codi principal d'una aplicació amb Struts es desenvolupa dins de les classes Action. Es des d'aquestes classes on s'accedeix a la lògica de negoci.

Exemple de classe Action:

```
package cd.dades.especialista;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import cd.dades.GestorDades;
import cd.dades.pojos.Pacient;

/**
 * Action de consultar els controls
 * @author F. Vilar
 *
 */
public class ConsultarControlsAction extends Action {

        public ActionForward execute(
                        ActionMapping mapping,
                        ActionForm form,
                        HttpServletRequest request,
                        HttpServletResponse response)
                throws Exception{

                ConsultarControlsForm consultarControlsForm = (ConsultarControlsForm) form;
                String data = consultarControlsForm.getData();

                String ntarg = consultarControlsForm.getNtarg();

                GestorDades gd = new GestorDades();

                consultarControlsForm.setGlucemies(gd.getGlucemies(ntarg,data));
                Pacient p = gd.getPacient(ntarg);

                request.setAttribute("pacient",p);

                return mapping.findForward("mostrarControls");
        }
}
```

Struts ofereix diverses classes derivades d'Action, a utilitzar segons el tipus d'acció es vol fer. Es aquests projecte només s'han utilitzat de dos tipus: Action i ForwardAction. La primera la em utilitzat en les accions amb formularis, com es veu a l'últim exemple.
La segona per realitzar el forward del links d'accés a les funcionalitats sense eludir la responsabilitat del controlador, com es veu al següent exemple:

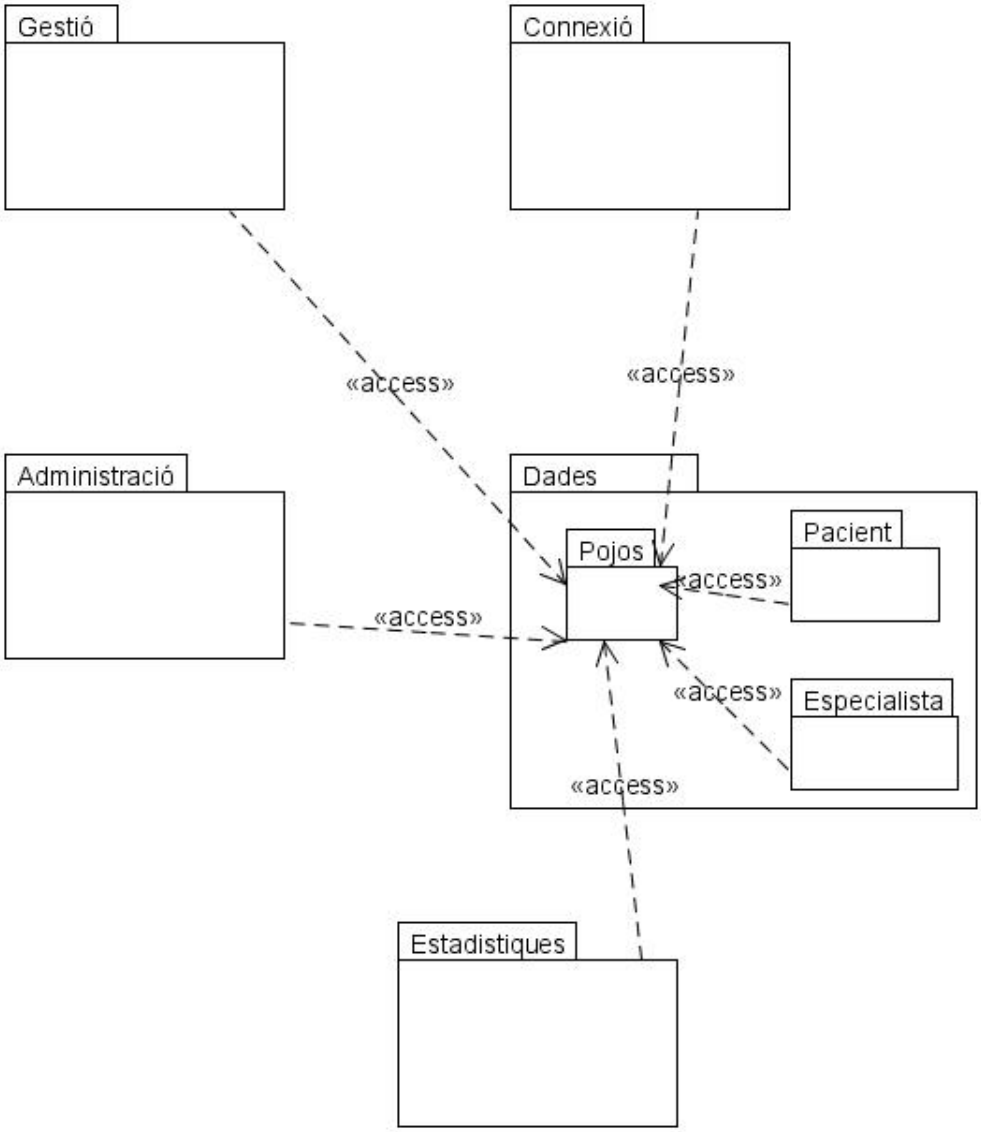
```
<!-- LINK A LA PAGINA D'ALTA D'USUARI -->
                <action
                        path="/linkAltaUsuari"
                        type="org.apache.struts.actions.ForwardAction"
                        parameter="/WEB-INF/pages/administrador/AltaUsuari.jsp"
                        input="/WEB-INF/pages/administrador/Administrador.jsp"
                        scope="request"
                        validate="false">
```

```
</action>
```

Des dels Action es guarden i recuperen objectes de negoci, form-beans i atributs en *session* o en *request* per recuperar-los als jsp. Com ja s'ha comentat abans, el session romandran disponibles durant tota la sessió i els request només durant la petició

7.4. Paquets

Les diverses classes generades durant la implementació s'han distribuït en paquets, per facilitar les tasques de localització, modificació, i manteniment, segons el següent diagrama:



7.5. Funcionalitats implementades

S'han implementat les següents funcionalitats:(classificades per actor):

Comuns:

- Login.

Administrador:

-Alta Usuari.
-Cercar usuari.

Cap:

-Assignar Metge.

Especialista:

-Consultar Pacient --> Consultar Controls --> Consultar Estadístiques

Pacient:

-Introduir Glucèmia Individual.
-Introduir Glucèmies Diàries.
-Introduir Glucèmia Individual.
-Introduir Glucèmies Diàries.
-Consultar Controls.
-Consultar Dosis Base

Cal indicar que les dates s'han d'introduir am el format dd/MM/yyyy, ja que aquesta es una part a millora, per exemple, introduint calendaris desplegables i que no s'han implementat totes les validacions dels camps HTML.

Per altre costat es poden obtenir usuaris i dades valides dins l'script de creació i populació de la base de dades.

8. Millores.

La part mes millorable del projecte es la Vista. Les pàgines han estat implementades de forma funcional però son poc estètiques. L'ús de Tiles, JasperReport i Displaytag pot afavorir enormement la seva presentació, donant-li uniformitat a les pàgines i un aspecte més professional als resultats.

Struts ha estat utilitzat de forma bàsica. Només s'han utilitzat 2 classes derivades d'Action. Hi han més que es poden adaptar més a les necessitat de cada funció. També es pot implementar la validació amb el plug-in inclòs i configurant-lo al arxius xml corresponent. Struts també ofereix facilitats per implementar la internacionalització d'aplicacions (i18n).

També es pot millorar la persistència, aprofundint en la utilització d'Hibernate.

Altra millora, per al producte final, seria la utilització d'un framework per fer proves unitàries, com per exemple JUnit.

En principi, aquesta aplicació només es farà per a malalts diabètics però no es descarta la seva ampliació a altres malalties que requereixin controls constants per part del pacient.

Un aspecte a desenvolupar, no plantejat inicialment es el de la seguretat, tenint en compte que estem tractant amb dades mèdiques personal afectades per la LOPD. (Ley Orgánica de Protección de Datos). Java ofereix serveis i llibreries per afegir seguretat a les aplicacions (JCA, JCE, JAAS).

També s'ha pensat, com a projecte futur, la ampliació de la aplicació per a donar suport als aparells de control (glucòmetre en el cas del diabètics) amb port per a la connexió directa a l'ordinador, per estalviar la introducció manual de les dades i l'ús de tecnologia mòbil.

9. Programari utilitzat.

Durant les diferents fases del projecte s'han utilitzat els següent programari:

JavaEE SDK 5.0.4 (jse 1.6.0)

Per a la implementació del patró MVC s'ha utilitzat Struts 1.2.7

Com a motor de persistència Hibernate 3.1

MySql Server 5.0, per a la base de dades.

Jboss 4.0.2 com a servidor d'aplicacions, que porta incorporat Tomcat..

NVU com a editor de planes html.

ArgoUml, per a la edició de diagrames UML.

UMLet, per a la edició de diagrames UML.

Eclipse Jboss IDE 1.6

Struts Console, durant la fase d'aprenentatge d'Struts

Microsoft Project 2003

10. Valoració econòmica

Tot i que no s'ha fet cap seguiment exhaustiu del temps invertit en el desenvolupament del projecte cal tenir en compte que totes les eines i complements utilitzats son programari lliure o gratuït (menys MS Project) la qual cosa incideix directament en el cost final del producte.

11. Conclusions

Aquest projecte es la primera aplicació seriosa que he fet. Personalment no treballo al sector de les TIC i no tinc cap experiència en projectes informàtics, la qual cosa ha fet d'aquest projecte un repte difícil, i laboriós, per mi.

El desenvolupament del projecte m'ha fet valorar la importància d'una bona planificació a l'inici d'aquest i de la necessitat d'un anàlisi encertat junt a un bon disseny. Això, suposo, només s'obté amb experiència en projectes de similar o major envergadura.

He pogut comprovar que existeixen molt frameworks, el comportament dels quals es pot configurar través d'un fitxer xml, que si son utilitzats en la construcció d'aplicacions, estalvien l'escriptura d'una gran quantitat de codi (hardcode) i per tant temps i cost. Aquest frameworks, tenen, per contra una corba d'aprenentatge pronunciada.

Trobar solucions, principalment a Internet, als problemes que sorgien durant el

desenvolupament del projecte, em fan pensar que molts desenvolupadors d'aplicacions tenen molts problemes en comú. Pot ser que calgui analitzar l'origen i corregir-los.

Encara que el projecte no està 100% acabat em satisfà molt veure els resultats, bàsics, de les funcionalitats implementades.

Ha estat una experiència sens dubte positiva

12. Glossari.

- Administrador. Encarregat de mantenir el sistema
- Metge. Encarregat de controlar els pacients
- Cap. Metge que coordina els altres metges.
- Pacient. Malalt que introduieix principalment els controls.
- Dosi. Unitats d'insulina que s'administra el malalt.
- Dosi base. Unitats d'insulina que s'hauria d'administrar el pacient si els control son correctes.
- Glucèmia: Valor obtingut amb el mesurador de glucosa, equivalent a control.
- Control: Valor obtingut amb el mesurador de glucosa, equivalent a glucèmia.

13. Bibliografia.

IVAR JACOBSON; GRADY BOOCH; JAMES RUMBAUGH.EL LENGUAJE UNIFICADO DE MODELADO. Addison Wesley.

IVAR JACOBSON; GRADY BOOCH; JAMES RUMBAUGH.EL LENGUAJE UNIFICADO DE MODELADO.MANUAL DE REFERENCIA. Addison Wesley.

IVAR JACOBSON; GRADY BOOCH; JAMES RUMBAUGH.EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE. Addison Wesley.

JAMES HOLMES. The Complete Reference. Struts. Second Edition. Osborne -Mc Graw Hill.

FELIPE LIMA DIAZ. Java 2 v5.0. Manual Avanzado. Anaya

Material UOC.

http://www.hibernate.org/hib_docs/reference/en/html/index.html
abril-08

http://www.codejava.org/?idxpagina=9&idxnota=32795&destacada=1
marzo-08

http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml
marzo-08

http://www.tic.udc.es/~fbellas/teaching/is-2001-2002/Tema4Apartado4.4.pdf
marzo

http://www.mindfiresolutions.com/mindfire/Java_Hibernate_JDBC.pdf
abril-08

http://www.di.uniovi.es/~dflanvin/docencia/dasdi/teoria/Transparencias/06.%20Arquitectura%20Web.pdf
abril-08

http://www.dei.inf.uc3m.es/docencia/p_s_ciclo/pa4/practicas/mvc.pdf
abril-08

http://petra.euitio.uniovi.es/~i6950404/wiki/pmwiki.php?n=Tema5.PatronFront-Controller
abril-08

http://www.lalilluna.de/assets/tutorials/struts-hibernate-integration-tutorial-es.pdf
abril-08

Projecte ControlD

Anàlisi, disseny i implementació d'una aplicació Web per a la supervisió remota d'autocontrols de malalts diabètics.

Fabià Vilar Garcia
ETIS
Universitat Oberta de Catalunya

25 de juny de 2008
Consultor: Salvador Campo Manzarico

OBJECTIUS I

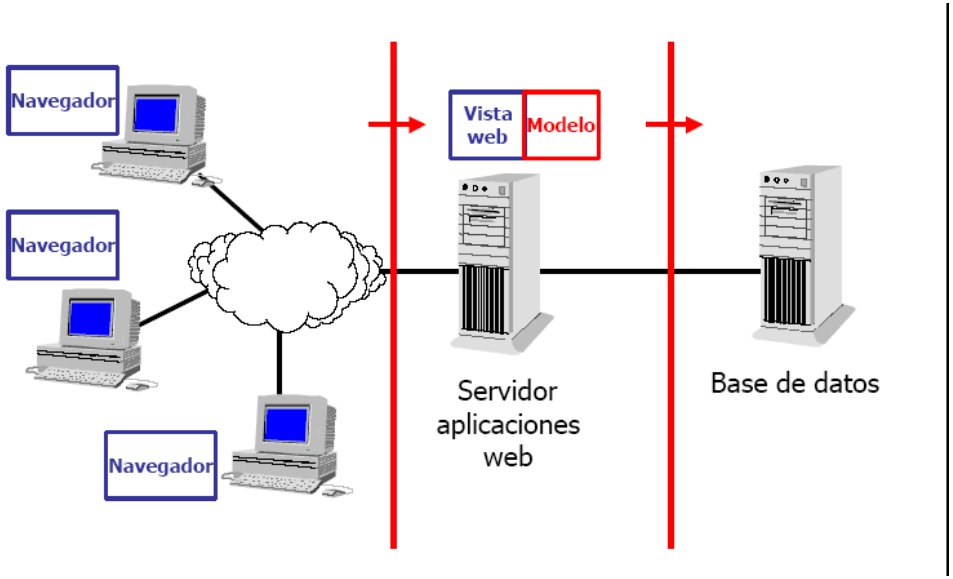
L'objectiu d'aquesta aplicació es la implementació d'un sistema que faci possible fer un seguiment, per part dels metges especialistes, de la evolució dels autocontrols que se efectuen diàriament els malalts diabètics.

OBJECTIUS II

Com a objectiu addicional, hi ha el d'aprofundir en el coneixement de la tecnologia J2EE, i del desenvolupament de projectes amb aquesta, utilitzant ides, frameworks i eines de suport

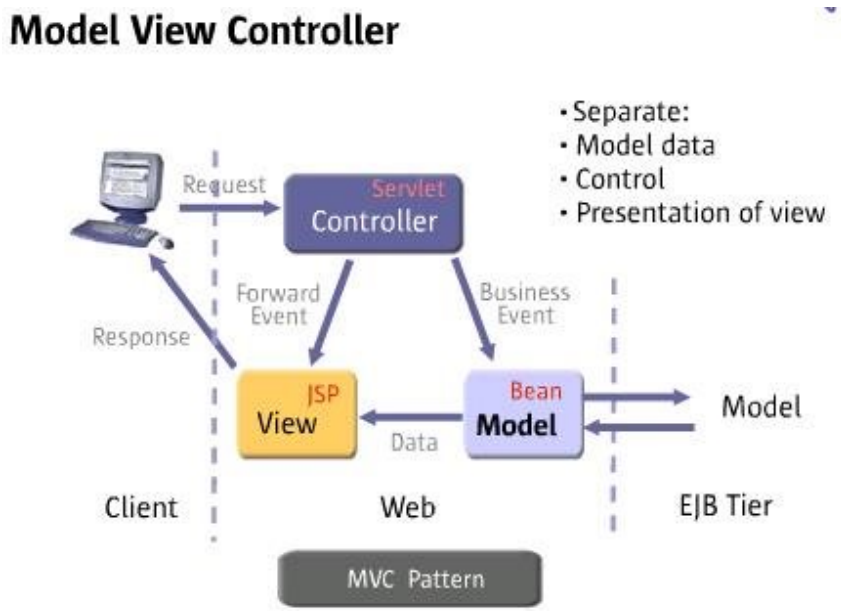
ARQUITECTURA I

La arquitectura de la aplicació segueix el model per capes:



ARQUITECTURA II

Pel seu disseny s'ha utilitzat el patró MVC . Aquest patró separa el model de la vista i el controlador.



TECNOLOGIA

Pel desenvolupament s'han utilitzat elements de la tecnologia J2EE i també complementaris d'aquesta, que s'integren perfectament entre ells:

- Java
- JSP
- POJO
- Struts
- Hibernate
- Xml

IMPLEMENTACIO I

Per a la implementació de la capa Vista s'han utilitzat pàgines JSP. Aquestes estan formades per codi Java, HTML i XML i permeten crear contingut dinàmic dins una web.

IMPLEMENTACIO II

La Capa Controlador està formada per ActionServlet, el nucli d'Struts, que està dissenyat seguint el patró Front Controller, i que centralitza totes les peticions dels clients en un sol punt. El seu comportament es configura a l'arxiu struts-config.xml

IMPLEMENTACIO III

La Capa Model està implementada amb Hibernate, un ORM que fa un mapeig entre el objectes de negoci i les taules de la base de dades, declarant un arxiu que relaciona cada atribut del objecte a una columna de la taula.

IMPLEMENTACIO IV

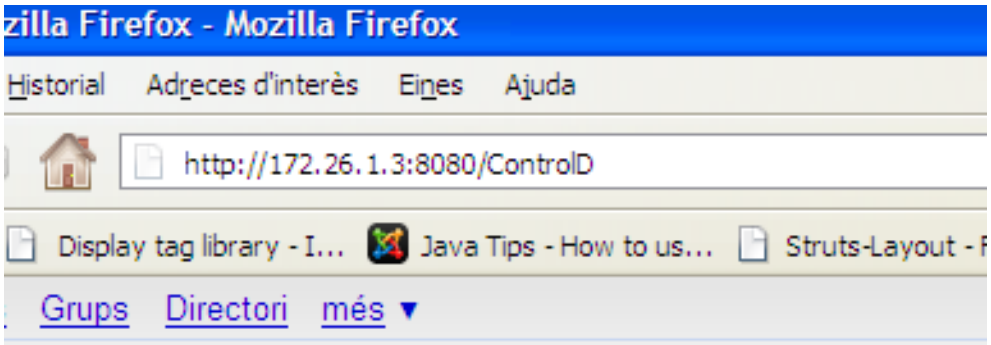
Com a objectes de negoci s'han utilitzat POJO (Plain Old Java Object). Aquestes son classes simples que només tenen declarat els atributs i el mètodes getter i setters d'aquests. Aquest s'integren fàcilment amb Hibernate.

DESPLEGAMENT

Pel desplegament de la aplicació es necessari un contenidor de servlets, com es Tomcat, o un servidor d'aplicacions com Jboss, que porta incorporat Tomcat.

ACCES A LA APLICACIO

Per accedir a la aplicació només cal escriure la direcció a la barra de qualsevol navegador:



CONCLUSIONS

La utilització de frameworks opensource per a la construcció d'aplicacions web facilita la implementació d'aquestes, reduint les línies de codi, el temps invertit i per tant el cost final.

FI DE LA PRESENTACIO