

# WEB PER A PSICOLOGIA I TERAPIA ONLINE



**TFC - J2EE**

**Enginyeria Tècnica d'Informàtica de Sistemes**

**Autor: Roberto González Méndez**

**Consultor: Joan Vicent Orença Serisuelo**

**Data presentació: 23/06/2011**

## 1. Dedicatòria

A Susanna,  
per la teva infinita paciència  
en tot aquest temps dedicat,  
i sobretot,  
pel teu amor incondicional.

## 2. Agraïments:

A la comunitat UOC,  
per fer possible uns estudis universitaris a distància,  
i per la qualitat de l'ensenyament.  
A tots, alumnes, consultors, tutor,

A Alberto Garulo,  
per donar-me l'empenta per començar aquesta enginyeria,  
per ser un bon company.

A la meva mare,  
per la educació que m'has donat ,  
per les teves monedes...

Gràcies!

### 3. Resum

Aquest Treball de Final de Carrera engloba l'anàlisi, el disseny i la implementació d'una aplicació web per a psicologia i teràpia online.

El projecte sencer és de gran envergadura i molt ambiciós. Degut a la càrrega de treball i al temps limitat per a la seva realització, he desenvolupat una petita part de les funcionalitats, amb les que es pot obtenir una visió generalitzada i clara del seu funcionament.

En aquest sentit, i dins d'aquest grup de funcionalitats, també hi han restriccions en determinades funcions que aportaven repetició més que no pas informació significativa, i que aniré especificant en cada cas que sigui necessari.

L'enginyeria d'aquest programari està basada en la tècnica d'orientació a objectes, dins l'estàndard UML.

Els aspectes generals de l'anàlisi i disseny s'han desenvolupat amb un cicle de vida en cascada, per tenir una bona base de partida i poder confeccionar una planificació en el temps. La fase de implementació, està basada en un cicle de vida iteratiu e incremental, implementant a cada iteració una petita part amb autonomia que correspon a un cas d'ús. A mesura que la implementació aprofundeix en detall, ha fet modificar aspectes inicials en les etapes d'anàlisi i disseny. Així mateix a succeït amb l'etapa de proves. Per tant, globalment es tracta d'un model de cicle de vida iteratiu e incremental basat en el cicle de vida en cascada.

Com a llenguatge de desenvolupament he escollit Java, i com a arquitectura de l'aplicació J2EE, degut a la seva robustesa i a que en l'actualitat, té un fort posicionament en aplicacions web i en xarxa, arribant a ser un estàndard en l'entorn distribuït d'aplicacions empresarials a Internet.

En l'estratègia en el disseny i per donar solucions efectives a problemes tipificats, he fet servir el patró MVC, que a més, ha incrementat considerablement la reusabilitat i efectivitat del codi i estructura de la programació.

Per a la implementació he incorporat el framework Struts2, que basa la seva arquitectura en el patró MVC, i que ha facilitat molt el treball ja que ha donat solucions a problemes generals estàndard i altres de baix nivell, i ha permès focalitzar els esforços en donar solució a qüestions més particulars i específiques del projecte.

En l'accés transparent a les dades he optat per Hibernate3, una poderosa eina que enllaça el món relacional de les BBDD amb el món de l'orientació a objectes de les classes de les aplicacions. I com a SGBD per a la persistència de dades, he fet servir Oracle 10g XE, també tot un referent en la indústria, i un dels més complets.

## 4. Índex

<b>1. Dedicatòria.....</b>	<b>2</b>
<b>2. Agraïments.....</b>	<b>2</b>
<b>3. Resum.....</b>	<b>3</b>
<b>4. Índex.....</b>	<b>4</b>
4.1. Índex de il·lustracions.....	5
<b>5. Introducció.....</b>	<b>6</b>
5.1. Justificació del TFC.....	6
5.2. Objectius del TFC.....	7
5.3. Descripció del producte final.....	7
5.4. Funcionalitats del producte.....	8
5.5. Enfocament i mètode seguit.....	10
5.6. Planificació del projecte.....	11
<b>6. Anàlisi.....</b>	<b>12</b>
6.1. Actors i casos d'ús.....	12
6.1.1. Actors.....	12
6.1.2. Casos d'ús.....	12
6.1.3. Fitxes dels casos d'ús més significatius.....	13
6.1.4. Resta de casos d'ús.....	20
6.2. Diagrama de classes.....	28
6.3. Diagrama de seqüència.....	29
<b>7. Disseny.....</b>	<b>30</b>
7.1. Disseny de l'arquitectura.....	30
7.1.1. Què és la plataforma J2EE?.....	30
7.1.2. Què és un patró de disseny de software?.....	31
7.1.3. Què és un framework?.....	32
7.1.4. Elecció de l'arquitectura del TFC.....	35
7.1.5. Diagrama de components.....	38
7.2. Disseny de la persistència.....	39
7.2.1. Diagrama E/R de la BBDD.....	39
7.2.2. Disseny relacional de la BBDD.....	40
7.3. Disseny de la interfície d'usuari.....	41
<b>8. Implementació.....</b>	<b>48</b>
8.1. Instal·lació del software.....	48
8.1.1. Instal·lació de Java (JDK).....	48
8.1.2. Instal·lació de l'IDE Eclipse i del contenidor web Tomcat....	48
8.1.3. Instal·lació de Struts2.....	49
8.1.4. Instal·lació de Hibernate3.....	51
8.1.5. Instal·lació de Oracle 10g XE.....	53
8.2. Desenvolupament de l'aplicació.....	53
8.2.1. Implementació de la capa de presentació d'Espai Nirvana.....	53
8.2.2. Implementació de la capa de presentació d'Espai Nirvana.....	54
8.2.3. Implementació de la capa de presentació d'Espai Nirvana.....	54
<b>9. Resultats.....</b>	<b>55</b>
<b>10. Conclusions.....</b>	<b>56</b>
<b>11. Bibliografia.....</b>	<b>57</b>

## 4.1. Índex de il·lustracions

Il·lustració 1. Planificació final del projecte.....	11
Il·lustració 2. Actors i casos d'ús.....	13
Il·lustració 3. Diagrama de classes.....	28
Il·lustració 4. Diagrama de seqüència.....	29
Il·lustració 5. Lògica de Struts2.....	34
Il·lustració 6. Arquitectura amb els frameworks Struts2 i Hibernate.....	37
Il·lustració 7. Diagrama de components de l'aplicació.....	38
Il·lustració 8. Diagrama E/R.....	39
Il·lustració 9. Disseny relacional de la BBDD.....	40
Il·lustració 10. Pàgina de presentació.....	41
Il·lustració 11. Entrada al sistema.....	42
Il·lustració 12. Pàgina personal.....	42
Il·lustració 13. Consulta de dades.....	43
Il·lustració 14. Consulta Pla de tractament.....	43
Il·lustració 15. Consulta Diagnòstic.....	44
Il·lustració 16. Consulta Avaluació.....	44
Il·lustració 17. Omplir Exercicis Test.....	45
Il·lustració 18. Consultar Pacient.....	45
Il·lustració 19. Modificar Diagnòstic.....	46
Il·lustració 20. Modificar dades pacient.....	47
Il·lustració 21. Assignar Mòdul tractament.....	47

## 5. Introducció

### 5.1. Justificació del TFC

L'origen d'aquest TFC sorgeix de la idea inicial de donar solució a unes necessitats no cobertes en l'atenció psicològica i terapèutica a persones. Es tracta de l'atenció psicològica i terapèutica "online" (en línia). Hi ha un fort forat de mercat en aquesta vessant, en la que les noves tecnologies li poden donar un gran impuls, i que ja son utilitzades des de fa temps en altres camps, com l'educació.

El producte final, està pensat per a persones que han d'afrontar un problema o tenen un malestar psíquic, i no s'atreveixen o bé no poden anar a un psicòleg o terapeuta presencialment. Ens trobem davant un sistema de tractament, que serà efectiu sempre i quan tracti amb problemes de curta o mitjana duració o d'una complexitat moderada. En qualsevol altre cas, la teràpia a distància no serà el model més adequat.

L'objectiu doncs, és donar la orientació i les eines necessàries perquè aquestes persones puguin superar els seus problemes, de manera no presencial.

Les TIC fan possible aquesta tasca, el vehicle és Internet, que dóna pas a la teràpia en línia. Així i dins d'aquest paradigma neix Espai Nirvana, una web per a psicoteràpia online.

Des de casa, amb el ritme personal que pot assolir cada persona i amb la planificació més adient, els pacients poden seguir un tractament personalitzat amb un contacte continu amb els professionals. La confidencialitat també queda garantida amb un accés segur a la web.

Els requisits de coneixements informàtics son mínims: habilitats de navegació bàsica per Internet. A més, l'estructura de la web és molt intuïtiva i usable.

## 5.2 Objectius del TFC

Desenvolupar una aplicació web per a la psicoteràpia online, usable e intuïtiva, que serveixi com a eina facilitadora del procés de tractament dels pacients, i com a eina per als professionals per portar a terme la seva tasca.

Com a segon objectiu, i no menys important, la creació d'un projecte sencer des de la idea inicial fins al producte final, passant per totes les etapes d'anàlisi de requeriments, disseny, implementació i proves.

Inherent a totes aquestes fases d'elaboració del producte, l'aprenentatge i coneixement de l'arquitectura J2EE, amb la que no havia treballat fins ara. Així mateix la utilització de patrons de disseny com MVC, l'aplicació dels frameworks Struts2 i Hibernate, i l'aprofundiment en el llenguatge Java, del que tenia coneixements molt bàsics.

## 5.3 Descripció del producte final

En aquest TFC desenvoluparé parcialment la web per a realitzar sessions de teràpia online. La web està orientada al tractament psicològic a persones amb patologies o problemes psíquics, d'una complexitat moderada.

Ofereix uns tractaments personalitzats en diversitat de malalties i addiccions, com la depressió, l'alcoholisme, problemes d'autoestima, etc.

La seqüència de funcionament de l'aplicació és la següent:

- Inicialment, i fora del funcionament propi de l'aplicació, un usuari contacta via e-mail amb els professionals del lloc web. Després d'omplir una sèrie de qüestionaris de dades personals i explicació general del seu problema, aquest rep les dades necessàries per entrar al sistema. A partir d'aquest moment, en que l'usuari entra al sistema, és quan comença el procés del seu tractament terapèutic..
- L'usuari, des d'ara "pacient", comença el procés de tractament, omplint un formulari per poder-li diagnosticar amb exactitud el seu problema.
- El terapeuta assignat estudia el cas i emet un diagnòstic que envia al pacient. Juntament, aplica un pla de tractament i uns mòduls per a començar la teràpia de tractament. Aquests mòduls consten d'uns exercicis a realitzar, i que avaluarà el terapeuta una vegada finalitzats.
- El pacient realitza els exercicis i quan rep l'avaluació pot saber com està evolucionant i en quins aspectes ha de millorar.

Com podem extreure de la seqüència de funcionament, el tractament consta de tres fases: diagnòsis del problema, tractament, i avaluació.

○ **Diagnòsis:**

Es recull la informació necessària per comprendre el problema del pacient, mitjançant un formulari. Aquest formulari consta d'una llista de preguntes test amb possible resposta SI / NO, i una pregunta oberta perquè el pacient pugui explicar amb detall el seu problema. El pacient omple les caselles i quan finalitza ho envia.

El terapeuta una vegada ho ha analitzat, respon amb un diagnòstic. El diagnòstic serà una explicació dins una taula, de la patologia o problema del pacient, i que aquest pot consultar en qualsevol moment des de la seva àrea privada. Per notificar al pacient el diagnòstic, el terapeuta ho comunica via e-mail.

○ **Tractament:**

Consta d'uns mòduls que assigna el terapeuta al pacient, en les que hi ha un pla de tractament, que és una guia adequada i personalitzada de la metodologia a seguir i explicació del dubtes més freqüents. I uns exercicis per realitzar.

Aquest exercicis es divideixen en dos tipus:

- Exercicis test, que es basen preguntes amb resposta SI o NO, més una pregunta oberta, per resposta explicativa.
- Exercicis formulari, que incorporen preguntes obertes amb espai per respostes raonades i explicatives.

Fins que el terapeuta no assigna un nou mòdul de tractament, el pacient no pot accedir a ell.

○ **Avaluació:**

Fase en la que el terapeuta emet una valoració del treball realitzat pel pacient. Es tracta d'un informe presentat en una taula, que el pacient podrà consultar també en qualsevol fase de la seva teràpia. L'avís al pacient per confirmar que ja té confeccionada una avaluació, es farà també via e-mail.

L'objectiu de l'avaluació és que el pacient sàpiga com esta evolucionant, en quins aspectes pot millorar i obtenir pautes i



consells per encaminar correctament el tractament i així assolir amb garanties d'èxit la teràpia.

## 5.4 Funcionalitats del producte

Trobem una pantalla inicial de presentació en la que hi ha informació diversa relativa a l'empresa, els seus professionals, què ofereix, i en què consisteixen els serveis.

Un accés directe del menú principal, dona pas a la pantalla de entrada al sistema, mitjançant un NIF i una contrasenya.

L'aplicació llavors, proporciona una sèrie de funcionalitats segons el rol de l'usuari que accedeix , si és un terapeuta o un pacient.

### Funcionalitats del pacient

Dins la seva àrea privada, el pacient té una sèrie de funcionalitats al seu abast:

- Omplir el formulari per al diagnòstic
- Consultar i realitzar els exercicis tipus test i formulari.
- Consultar les dades personals
- Consultar el pla de tractament , el diagnòstic, i l'avaluació.

### Funcionalitats del terapeuta

Dins la seva àrea privada, el terapeuta disposa també d'un ventall de funcionalitats al seu abast:

- Consultar, editar o modificar el diagnòstic, l'avaluació i el pla de tractament.
- Consultar i modificar els exercicis tipus test i formulari.
- Consultar i modificar el formulari per al diagnòstic.
- Consultar, editar o modificar les dades dels pacients.
- Assignar els mòduls de tractament als pacients.

## 5.5 Enfocament i mètode seguit

Per a portar a terme el desenvolupament del TFC he hagut de realitzar dues tasques paral·lelament. D'una banda, treballar en les diverses etapes del cicle de vida del programari. D'altra banda, l'estudi de l'arquitectura J2EE, dels diversos patrons de disseny, i dels frameworks que poden facilitar la feina.

Una vegada analitzats els diversos patrons i tecnologies de frameworks, he hagut d'escollir els més adients per les seves funcionalitats, corbes d'aprenentatge i temps per finalitzar el TFC.

Amb aquestes directrius he escollit el patró MVC, el framework Struts2, i el ORM Hibernate3.

L'enfocament és el de qualsevol empresa de desenvolupament de software. Seguint totes les etapes, en el que és tant important el resultat final com la facilitat de manteniment, amb la reusabilitat i claredat del codi, i la documentació. Tot intentant realitzar-ho amb bones pràctiques de programació. L'objectiu és obtenir un producte eminentment pràctic, amb unes funcionalitats de gestió per a accés i manipulació de dades, i amb altres funcionalitats per a facilitar una navegació intuïtiva i clara, que ajudi a complir amb la finalitat de la web.

## 5.6 Planificació del projecte

Per portar a bon terme el projecte, es necessari una acurada planificació en el temps, amb una descomposició en les diferents fases i en les diverses tasques ha realitzar. Un seguiment rigorós ens farà portar un ritme adequat i adaptar-nos als canvis, inevitables, en l'estratègia del desenvolupament.

Fase	Dies	Data inici	Data fi
<b>Anàlisi</b>			
Proves amb la tecnologia	9	17/03/2011	25/03/2011
Estudi dels casos d'ús	5	26/03/2011	30/03/2011
Anàlisi de les funcionalitats	3	31/03/2011	02/04/2011
Definició de BBDD	3	03/04/2011	05/04/2011
	<b>Total: 20</b>		
<b>Disseny</b>			
Diagrama de classes i seqüències	2	06/04/2011	07/04/2011
Disseny de les pàgines web	4	08/04/2011	11/04/2011
Disseny BBDD	3	12/04/2011	14/04/2011
	<b>Total: 9</b>		
<b>Implementació i proves</b>			
Creació de la BBDD	4	15/04/2011	18/04/2011
Estructura web	5	19/04/2011	23/04/2011
Pàgina de inici de funcionalitats	7	24/04/2011	30/04/2011
Funcionalitats pacient	7	01/04/2011	07/05/2011
Funcionalitats terapeuta	8	08/05/2011	15/05/2011
Proves	8	16/05/2011	23/05/2011
	<b>Total: 39</b>		
<b>Memòria i presentació</b>			
Redacció de la memòria	18	24/05/2011	10/06/2011
Creació de la presentació virtual	10	11/06/2011	20/06/2011
	<b>Total: 28</b>		

Il·lustració 1. Planificació final del projecte

Paral·lelament a aquesta planificació, he investigat patrons de disseny, frameworks i diverses tecnologies que formen el conjunt de la plataforma J2EE, en quant les seves característiques i funcionalitats, per poder utilitzar-les al projecte i tenir els conceptes clars a l'hora de la seva implementació.

## 6. Anàlisi

En aquesta etapa, es descriu el futur programari, sense tenir en compte els mitjans tècnics amb que es durà a terme, com en quin llenguatge s'implementarà, quina serà la base de dades, etc.

### 6.1 Actors i casos d'ús

Els actors son la visió que té el programari d'una entitat exterior. Cada actor forma un conjunt de papers en relació amb aquest. Per ser considerat entitat exterior, ha de ser autònom respecte el programari, i tenir relació directa amb ell.

Un cas d'ús documenta la interacció entre el programari i un o més actors. Aquesta ha de ser una funció autònoma dins el programari.

#### 6.1.1 Actors

Ens trobarem dos perfils d'usuaris del sistema:

**Pacient:** usuari que rebrà la teràpia de tractament

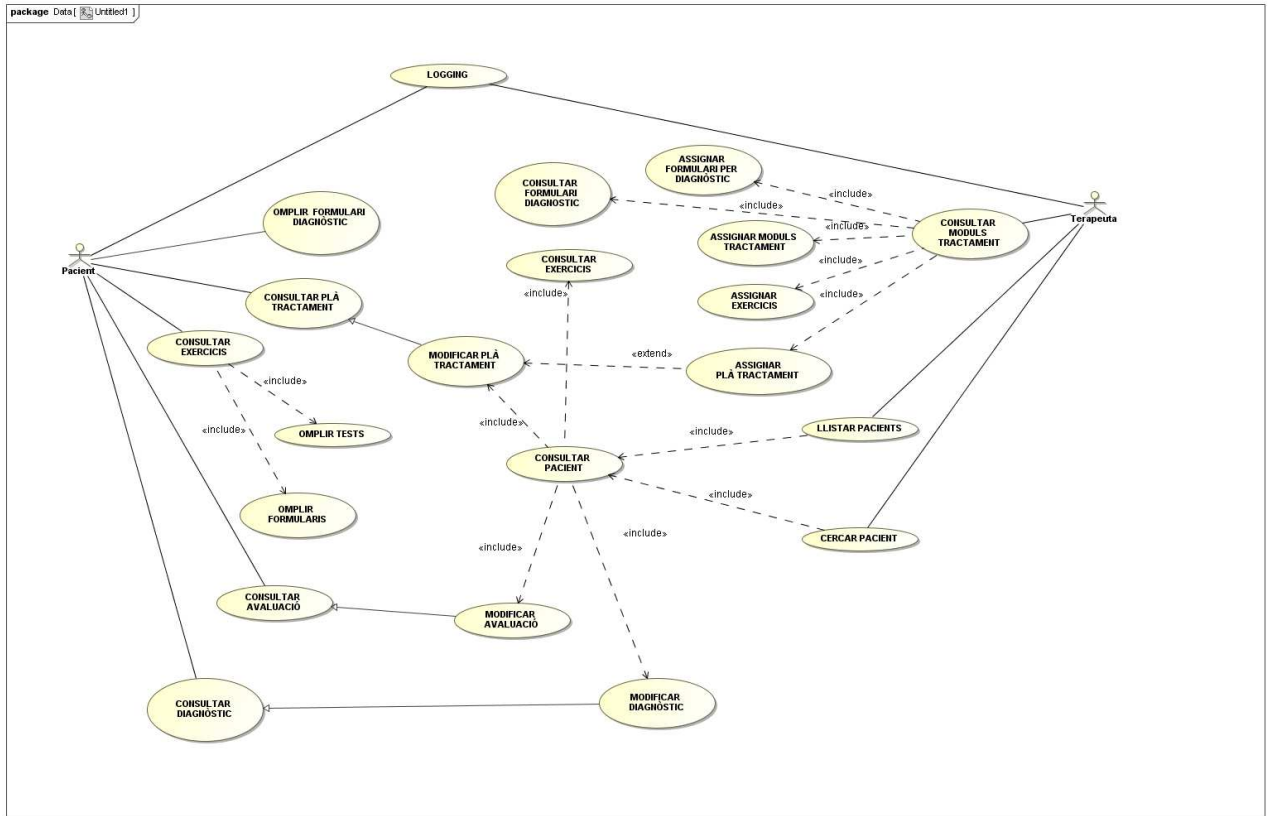
- Podrà consultar i omplir el formulari per al diagnòstic.
- Consultar i realitzar els exercicis tipus tests i tipus formulari.
- Consultar el pla de tractament, l'avaluació i el diagnòstic.

**Terapeuta:** responsable de tot el procés de tractament del pacient

- Podrà consultar i assignar el formulari pel diagnòstic.
- Assignar, consultar i modificar els exercicis tipus tests i formularis.
- Assignar, consultar i modificar el pla de tractament.
- Assignar, consultar i modificar l'avaluació del pacient.
- Assignar, consultar i modificar el diagnòstic del pacient.
- Assignar, consultar i modificar els mòduls sencers del pacient.
- Llistar, cercar i consultar dades pacients.

#### 6.1.2 Casos d'ús

En l'esquema següent es mostren les relacions entre actors i casos d'ús del projecte.



Il·lustració 2. Actors i casos d'ús

### 6.1.3 Fitxes dels casos d'ús més significatius:

Documentació textual dels casos d'ús:

#### Cas d'ús 1: Accés al sistema - Login

Resum de la funcionalitat: validar un usuari per comprovar que està registrat al sistema.

Paper dins del treball de l'usuari: es utilitza per tots els usuaris per entrar al sistema. Donem per fet que les dades dels usuaris ja han estat entrades al sistema. Quan un usuari es vol registrar, ha de contactar via e-mail a l'adreça que informa la web. Una vegada li envien les dades per entrar al sistema, aquest ja pot navegar per la seva àrea privada.

Actors: Pacient i Terapeuta

Precondició: l'usuari serà vàlid si està el seu NIF d'usuari i la contrasenya a la base de dades.

Postcondició: el sistema valida l'usuari, i si és satisfactori el resultat, segons el perfil, li mostra un determinat menú i la pàgina principal de benvinguda, i tindrà accés a les seves funcionalitats particulars.

Flux d'esdeveniment principal: l'usuari entra a la pantalla de Login, mitjançant el botó de funció "Accés usuaris" de la pantalla inicial. El sistema presenta dos camps de text, per demanar un NIF d'usuari i una contrasenya i així poder comprovar que l'usuari és qui diu que és.

Flux d'esdeveniments alternatius: si no és correcte la validació, es mostren missatges d'error. Específicament, en el cas que no s'introdueixi cap dada en els camps mencionats, ens donarà un missatge conforme ha d'introduir dades en el camp que li demana. Així mateix si el NIF de l'usuari o la contrasenya no son al sistema, ens tornarà un missatge de error.

## **Cas d'ús 2: Omplir Formulari per Diagnòstic**

Resum de la funcionalitat: el pacient omple el Formulari perquè quan el terapeuta el rebí, pugui realitzar el diagnòstic del pacient.

El formulari per al diagnòstic consta d'una llista de 9 preguntes tipus test. És a dir, amb possible resposta SI o NO, que l'usuari escull d'un combobox. A més hi ha una pregunta oberta amb una resposta explicativa que pot ser extensa, i que s'omple dins una petita àrea de text de 1000 caràcters.

Paper dins del treball de l'usuari: es tracta del primer Formulari que ha de realitzar el pacient, per a que el terapeuta confeccioni el diagnòstic i determini el tractament adient.

Actors: Pacient

Precondició: el terapeuta ha assignat el Formulari del Diagnòstic al pacient. Aquest ha entrat al sistema i consulta el Formulari .

Postcondició: el pacient ha omplert el Formulari i ho ha enviat.

Flux d'esdeveniment principal: en el cas que el terapeuta hagi assignat el Formulari al pacient, aquest podrà accedir-hi i omplir els camps de dades responent a les preguntes. Una vegada ha acabat, ho envia per poder obtenir un diagnòstic del seu terapeuta i el tractament necessari, i així poder començar a realitzar els exercicis dels mòduls del seu tractament.

Flux d'esdeveniments alternatius: En el cas que el Formulari encara no estigui assignat al pacient, aquest quan intenti accedir-hi, rebrà un missatge d'avís conforme encara no pot omplir-lo.

Una vegada fet el diagnòstic per part del terapeuta, el pacient no pot modificar les respostes del Formulari, i rebrà un missatge d'avís explicatiu de la situació. Mentrestant pot modificar-lo per canviar alguna qüestió o afegir alguna dada que cregui rellevant.

A l'hora de realitzar el Formulari, si no respon a cap o a alguna de les preguntes de qualsevol dels tipus, no podrà enviar-lo, i el sistema mostrarà els avisos en els camps que no hagin estat omplerts.  
En el cas que l'usuari surti de l'aplicació, i no hagi enviat les dades, aquestes es perdran.

### **Cas d'ús 3: Omplir Exercicis tipus Tests**

Resum de la funcionalitat: el pacient realitza els exercicis tipus test del seu mòdul de tractament.

Un exercici tipus test, consta d'una llista amb 10 preguntes amb possible resposta SI o NO dins d'un combobox.

Paper dins del treball de l'usuari: els test son un dels dos tipus d'exercici de que consten els mòduls del tractament. Aquest cas d'ús, permet al pacient realitzar unes activitats que formen una part del treball del seu tractament. Únicament implementaré un exercici tipus test.

Actors: Pacient

Precondició: el terapeuta ha assignat els mòduls de tractament al pacient. Aquest ha entrat al sistema, i dins l'apartat tractament, consulta els exercicis tipus test.

Postcondició: el pacient ha realitzat l'exercici tipus test i ho ha enviat.

Flux d'esdeveniment principal: després que el terapeuta ha assignat els mòduls de tractament al pacient, aquest pot entrar, mitjançant l'accés "tractament", i dins l'apartat exercicis tests, pot obrir un test per omplir-ho i enviar els resultats.

Flux d'esdeveniments alternatius: En el cas que els mòduls de tractament encara no estiguin assignats al pacient, aquest quan intenti accedir a l'apartat "tractament", rebrà un missatge d'avís conforme encara no té cap tractament.

Una vegada fet el informe d'avaluació per part del terapeuta, el pacient no pot modificar les respostes dels tests, i rebrà un missatge d'avís explicatiu de la situació. Mentrestant pot refer-los per canviar alguna qüestió o afegir alguna resposta que cregui necessària.

A l'hora de omplir els tests, si no respon a cap o alguna de les preguntes, no podrà enviar-lo, i el sistema mostrarà els avisos en els camps que no hagin estat omplerts.  
En el cas que l'usuari surti de l'aplicació, i no hagi enviat les dades, aquestes es perdran.

#### **Cas d'ús 4: Consultar Avaluació**

Resum de la funcionalitat: permet al pacient consultar l'avaluació confeccionada pel terapeuta, del treball realitzat.

L'avaluació és un informe que es presenta dins una taula, amb una extensió màxima de 1000 caràcters.

Paper dins del treball de l'usuari: hi ha una avaluació per cada mòdul realitzat i una avaluació global final del tractament. Es un apartat que pot consultar el pacient en qualsevol moment dins el seu procés de tractament.

A més és un cas d'ús que pot accedir el terapeuta com a pas previ per a modificar o confeccionar l'avaluació. No entrem en més detall per la importància secundària que té per al terapeuta, i que a més quedarà detallada en el cas d'ús "Modificar avaluació". Així ens centrem solament en la funcionalitat per al pacient.

Actors: Pacient.

Precondició: el pacient ha realitzat els exercicis del mòdul, els ha enviat, i el terapeuta ha omplert el informe d'avaluació.

Postcondició: el pacient ha pogut veure el informe d'avaluació.

Flux d'esdeveniment principal: després que el terapeuta hagi realitzat la avaluació, el pacient pot veure el seu informe d'avaluació, i així poder valorar personalment la seva evolució i seguir les orientacions donades per avançar en el seu procés.

Flux d'esdeveniments alternatius: Si el pacient encara no té cap avaluació, simplement veurà el informe buit. A mesura que vagi avançant en els mòduls, anirà tenint les avaluacions corresponents.

#### **Cas d'ús 5: Cercar Pacient**

Resum de la funcionalitat: el terapeuta introdueix el NIF del pacient, i rep les dades personals del pacient.

Paper dins del treball de l'usuari: permet al terapeuta cercar un pacient per poder gestionar les seves dades i el seu procés de teràpia. Es un cas d'ús necessari per poder accedir a qualsevol apartat del tractament del pacient.

Actors: terapeuta.

Precondició: el terapeuta ha entrat al sistema, i vol cercar un pacient.

Postcondició: el terapeuta rebrà totes les dades personals del pacient i l'accés directe "Consultar".



Flux d'esdeveniment principal: el terapeuta entra a l'apartat Pacients, i cerca un pacient introduint el NIF d'aquest. Rebrà del sistema una taula amb totes les dades personals del pacient i un accés directe que li permetrà entrar a una pàgina per gestionar aquestes dades i les dades del diferents apartats del tractament.

Flux d'esdeveniments alternatius: el terapeuta pot també introduir una part del NIF del pacient. Llavors rebrà del sistema tots els pacients que coincideixin amb la part del NIF introduïda. En cas que no introdueixi cap, obtindrà la llista de tots els pacients ordenada pel número de pacient, (cas d'ús 13).

Si el pacient no és al sistema, rebrà un missatge avisant d'aquesta situació.

### **Cas d'ús 6: Consultar Pacient**

Resum de la funcionalitat: el terapeuta accedeix a la pàgina on pot consultar i modificar les dades personals, diagnòstic, pla de tractament, avaluació i exercicis d'aquest pacient. També pot eliminar del sistema qualsevol apartat, inclòs al pacient mateix

Paper dins del treball de l'usuari: és un cas d'ús necessari per poder accedir a qualsevol apartat del tractament del pacient, per gestionar el procés terapèutic del pacient. Es la part principal de treball del terapeuta.

Actors: terapeuta.

Precondició: el terapeuta ha fet una cerca del pacient mitjançant un llistat o una cerca directa, i ha pitjant en l'accés directe "Consultar".

Postcondició: el terapeuta pot gestionar el procés terapèutic del pacient mitjançant els accessos directes als diferents apartats del tractament del pacient i consultar les dades personals.

Flux d'esdeveniment principal: el terapeuta obté una sèrie d'accessos directes amb els que pot realitzar determinades funcionalitats. Consultar els dades personals del pacient. Consultar, editar/modificar i eliminar els mòduls de tractament, el formulari per al diagnòstic, el pla de tractament, exercicis tipus test i tipus formulari, avaluacions o diagnòstic del pacient. O bé, confeccionar un diagnòstic, avaluació, o un pla de tractament.

Flux d'esdeveniments alternatius: cadascuna de les funcionalitats de cada accés directe, és un nou cas d'ús.

### **Cas d'ús 7: Modificar Diagnòstic**

Resum de la funcionalitat: el terapeuta modifica el diagnòstic d'un pacient, o edita el diagnòstic sencer en cas que encara no ho hagi confeccionat. El diagnòstic és un informe que es presenta dins una taula, amb una extensió màxima de 1500 caràcters.

Paper dins del treball de l'usuari: és un cas d'ús al que accedeix el terapeuta quan vol confeccionar el diagnòstic, i molt esporàdicament si vol modificar algun aspecte del informe.

Actors: terapeuta.

Precondició: el terapeuta, després d'haver realitzat una cerca i una consulta del pacient, ha realitzat una consulta del diagnòstic, i ha pitjant en l'accés directe "Editar diagnòstic" dins la taula de dades que rep de la consulta feta.

Postcondició: el terapeuta ha modificat o editat sencer el diagnòstic del pacient i ho ha introduït al sistema per a que el pacient ho pugui consultar.

Flux d'esdeveniment principal: el terapeuta pot modificar i/o eliminar el diagnòstic. En cas que no hi hagi, perquè es tracta d'un nou pacient, pot confeccionar un primer diagnòstic. Una vegada lliurat al pacient, no té sentit modificar-lo a excepció que trobi alguna errada, cap qüestió important que no havia inclòs o explicar-ho d'una altre manera.

Flux d'esdeveniments alternatius: el terapeuta avisa al pacient mitjançant e-mail de la modificació o confecció del diagnòstic.

### **Cas d'ús 8: Consultar Dades Exercicis**

Resum de la funcionalitat: permet al terapeuta consultar el resultat dels exercicis tipus test i tipus formulari que ha realitzat el pacient per poder tenir la informació necessària per realitzar l'avaluació corresponent. (Aquesta última funcionalitat pertany a un altre cas d'ús ).

Paper dins del treball de l'usuari: es part del treball de terapeuta. Bàsicament és la recollida d'informació que donarà al professional una idea de com respon el pacient i com evoluciona, i així al final de tots els exercicis del mòdul, realitzar la avaluació.

Actors: terapeuta

Precondició: el terapeuta ha realitzat una cerca i una consulta del pacient. Ha fet una consulta dels exercicis tipus test, o bé una consulta dels exercicis tipus formulari que ha realitzat el pacient.

Postcondició: el terapeuta ha consultat els exercicis del pacient i està preparat amb la informació necessària per la realització d'una avaluació dels resultats si escau.

Flux d'esdeveniment principal: es tracta de dues funcionalitats, que degut a la seva semblança, les incloem en un sol cas d'ús. Consultar els exercicis tipus test, i consultar els exercicis tipus formulari. El terapeuta recull la informació per a més endavant realitzar l'avaluació dels resultats, per a que el pacient tingui accessible la valoració del seu progrés en el tractament

Flux d'esdeveniments alternatius: En el cas que el pacient no hagi realitzat els exercicis, quan el terapeuta faci la consulta rebrà els exercicis buits. No hi haurà cap missatge d'avís en aquest cas. Si ho creu convenient pot comunicar-se via e-mail amb el pacient per aclarir qualsevol qüestió.

### **Cas d'ús 9: Assignar Mòduls Tractament**

Resum de la funcionalitat: permet al terapeuta assignar els mòduls de tractament al pacient.

Paper dins del treball de l'usuari: el terapeuta accedeix quan el pacient ha de començar un nou mòdul. Es tracta d'una funcionalitat bàsica perquè el pacient pugui començar o continuar el seu procés de tractament. Únicament implementaré un mòdul de tractament.

Actors: terapeuta

Precondició: el terapeuta ha entrat al sistema, i ha entrat dins l'apartat Mòduls.

Postcondició: el terapeuta ha assignat els mòduls necessaris al/als pacient/s .

Flux d'esdeveniment principal: el terapeuta ha entrat a l'apartat Mòduls. El terapeuta pot introduir el NIF d'un pacient per poder assignar-li o treure-li el mòdul de tractament.

Flux d'esdeveniments alternatius: el terapeuta pot també des de aquest cas d'ús, assignar o treure el Pla de tractament del pacient.

#### **6.1.4 Resta de casos d'ús:**

No es tracta de casos d'ús menys importants, si no que son molt semblants als considerats principals o bé formen part d'aquest com a extensió de

funcionalitat. Els casos d'ús 14, 15, i 16, són purament explicatius, no quedaran implementats en aquest TFC.

### **Cas d'ús 10 : Assignar Pla de Tractament**

Resum de la funcionalitat: el terapeuta pot assignar el Pla de Tractament al pacient, des de l'apartat Mòduls.

El pla de tractament és un informe que es presenta dins una taula, amb una extensió màxima de 1000 caràcters.

Paper dins del treball de l'usuari: forma part del cas d'ús 9 (Assignar Mòduls Tractament). Quan el terapeuta assigna els mòduls al pacient, assigna tot el seu contingut. En aquest cas específicament assigna sols el pla de tractament. Es tracta d'una funcionalitat d'ús molt esporàdic.

Actors: terapeuta

Precondició: el terapeuta a entrat al sistema i ha entrat a l'apartat Mòduls. Ha consultat els mòduls d'un tractament i ha consultat el contingut del mòdul.

Postcondició: el terapeuta ha assignat el pla de tractament al pacient, i aquest ho pot consultar des d'aquest moment.

Flux d'esdeveniment principal: el terapeuta dins un mòdul, fa una consulta del pla de tractament que vol assignar al pacient. Rep les dades i introdueix el NIF del pacient a qui vol assignar el pla. Finalment acciona el botó "Assignar".

Flux d'esdeveniments alternatius: si no existeix el pacient, el sistema ens donarà l'avís de introduir NIF.

### **Cas d'ús 11 : Consultar Pla de Tractament**

Resum de la funcionalitat: terapeuta i pacient, poden consultar el Pla de Tractament d'un mòdul.

Paper dins del treball de l'usuari: per al cas del terapeuta, es el pas previ per assignar el pla de tractament al pacient (cas d'ús 10).

Per al cas del pacient, es el pas previ del començament del mòdul, en el que pot llegir la guia i consell de realització.

Actors: terapeuta i pacient

Precondició: l'usuari (terapeuta o pacient), han entrat al sistema. El pacient ha entrat a l'apartat tractament i ha accionat el botó "Pla tractament".

El terapeuta ha realitzat una consulta del tractament, seguidament una consulta del mòdul, ha accionat el botó “Pla tractament” i dins la taula de dades rebuda, ha pitjat l’ accés directe “Consultar”.

Postcondició: l’usuari ha rebut les dades del pla de tractament.

Flux d’esdeveniment principal: el pacient consulta el pla accionant el botó “Pla de tractament”, i rep el pla que ha de llegir detingudament per a la correcta realització dels exercicis del mòdul i per obtenir una orientació sobre els objectius a aconseguir amb el mòdul. En el cas del terapeuta, consulta el pla per a poder assignar-lo al pacient o modificar-lo.

Flux d’esdeveniments alternatius:

## **Cas d’ús 12 : Modificar Pla de Tractament**

Resum de la funcionalitat: el terapeuta pot modificar el contingut del pla de tractament

Paper dins del treball de l’usuari: funcionalitat necessària per modificar o confeccionar el pla de tractament d’un mòdul.

Actors: terapeuta

Precondició: el terapeuta ha realitzat una cerca i després una consulta del pacient.

Postcondició: el sistema mostra les dades del pla de tractament

Flux d’esdeveniment principal: el terapeuta acciona l’ accés directe “Consultar” dins la taula de dades del pacient, corresponent al pla de tractament. Visualitza les dades i edita/modifica el pla, per a finalment guardar les dades al sistema. A partir d’aquest moment el pacient consultarà el pla modificat.

Flux d’esdeveniments alternatius: el terapeuta avisa al pacient mitjançant e-mail de la modificació o confecció del pla de tractament.

## **Cas d’ús 13 : Llistar Pacients**

Resum de la funcionalitat: el terapeuta obté un llistat de tots els pacients dins l’ apartat “Pacients”.

Paper dins del treball de l'usuari: com en el cas d'ús de cercar pacient, és un pas previ per a la resta de casos d'ús en que es vol consultar/modificar dades personals del pacient i dades dels diferents apartats del seu tractament.

Actors: terapeuta

Precondició: el terapeuta ha entrat al sistema i ha entrat dins l'apartat "Pacients".

Postcondició: el sistema mostra el llista de tots els pacients. Mostra el seu nom, cognoms i NIF, més un accés directe "Consultar".

Flux d'esdeveniment principal: dins l'apartat pacients el terapeuta no introdueix cap NIF, i acciona el botó "Cercar", obtenint la llista de tots els pacients ordenada pel número de pacient.

Flux d'esdeveniments alternatius:

#### **Cas d'ús 14 : Assignar Formulari per al Diagnòstic**

Resum de la funcionalitat: el terapeuta pot assignar el Formulari per al Diagnòstic al pacient, des de l'apartat Mòduls.

Paper dins del treball de l'usuari: el terapeuta assigna el Formulari per al Diagnòstic al nou pacient que vol iniciar un procés de teràpia. Es tracta d'una funcionalitat només utilitzada al començament de tot el tractament.

Actors: terapeuta

Precondició: el terapeuta a entrat al sistema, ha entrat a l'apartat Mòduls, i després a l'apartat Formularis per al diagnòstic, i ha consultat un d'ells

Postcondició: el terapeuta ha assignat el Formulari per al Diagnòstic, i aquest ho pot omplir des d'aquest moment.

Flux d'esdeveniment principal: el terapeuta dins l'apartat Formularis per al diagnòstic, fa una consulta del Formulari per al Diagnòstic que vol assignar al pacient. Rep les dades i introdueix el NIF del pacient a qui vol assignar el Formulari. Finalment acciona el botó "Assignar".

Flux d'esdeveniments alternatius: prèviament quan el pacient ha estat donat d'alta en el sistema, el terapeuta ha tingut contacte via e-mail amb el pacient, i ja té una idea orientativa del seu problema o preocupació. Dins aquesta situació, el terapeuta pot encaminar el Formulari més adient per especificar el grau i determinar exactament el tipus de problema, per realitzar un diagnòstic precís.

### **Cas d'ús 15 : Consultar Formulari per al Diagnòstic**

Resum de la funcionalitat: el terapeuta pot consultar el Formulari per al Diagnòstic.

És un pas previ necessari per a poder confeccionar un diagnòstic per al pacient

Paper dins del treball de l'usuari: cas d'ús bàsic per a començar el procés de tractament. És la base sobre la que es confecciona el diagnòstic i assignarà el terapeuta els mòduls adients.

També es utilitza per el terapeuta just abans d'assignar un Formulari per al Diagnòstic.

Actors: terapeuta.

Precondició: el terapeuta ha entrat al sistema, ha realitzat una cerca i una consulta del pacient.

O d'altra banda, el terapeuta ha entrat al sistema, ha entrat a l'apartat Mòduls, i després a l'apartat Formularis per al diagnòstic.

Postcondició: el sistema mostra les dades del Formulari omplert pel pacient.

Flux d'esdeveniment principal: dins la taula de dades del pacient, el terapeuta accedeix a l'accés directe "Consultar" en l'apartat "Formulari Diagnòstic".

Flux d'esdeveniments alternatius: també és utilitzat pel terapeuta dins de l'apartat Formularis per al Diagnòstic, i pitjant sobre un d'ells obté les dades del Formulari.

Com queda explicat en el cas d'ús 2, una vegada fet el diagnòstic per part del terapeuta, el pacient no pot modificar les respostes del Formulari Diagnòstic, i rebrà un missatge d'avís explicatiu de la situació. Mentrestant pot refer-los per canviar alguna qüestió o afegir alguna resposta que cregui necessària.

### **Cas d'ús 16 : Assignar exercicis**

Resum de la funcionalitat: permet al terapeuta, assignar al pacient els exercicis tipus tests i exercicis tipus formularis individualment del mòdul sencer, des de l'apartat Mòduls.

Paper dins del treball de l'usuari: forma part del cas d'ús 9 (Assignar Mòduls Tractament). Quan el terapeuta assigna els mòduls al pacient, assigna tot el seu contingut. En aquest cas específicament assigna els exercicis de test i de formulari. Es tracta d'una funcionalitat d'ús molt esporàdic.

Actors: terapeuta

Precondició: el terapeuta ha entrat al sistema i ha entrat a l'apartat Mòduls. Ha consultat els mòduls d'un tractament i ha consultat el contingut del mòdul.

Postcondició: el terapeuta ha assignat els exercicis al pacient, i aquest els pot consultar i realitzar des d'aquest moment.

Flux d'esdeveniment principal: el terapeuta dins un mòdul, fa una consulta dels exercicis que vol assignar al pacient. Realment es tracta de dues consultes una per cada tipus d'exercici. En qualsevol dels dos casos, rep les dades i introdueix el NIF del pacient a qui vol assignar l'exercici. Finalment acciona el botó "Assignar".

Flux d'esdeveniments alternatius: si no existeix el pacient, el sistema ens donarà l'avís de introduir NIF.

### **Cas d'ús 17 : Consultar exercicis**

Resum de la funcionalitat: pacient i terapeuta poden consultar els exercicis test i formulari.

Paper dins del treball de l'usuari: té dues funcionalitats, una pel pacient, i altre pel terapeuta.

En el cas del pacient, aquest pot consultar un exercici una vegada ho ha omplert i enviat i vol revisar les respostes abans que el terapeuta ho hagi avaluat.

En el cas del terapeuta accedeix just abans d'assignar un exercici al pacient.

Actors: terapeuta i pacient.

Precondició: el terapeuta ha entrat al sistema, ha realitzat una cerca i una consulta del pacient.

O d'altra banda, el pacient ha entrat al sistema, i dins l'apartat Tractament, pot accedir als exercicis tipus test o als tipus formulari, que hagi realitzat i/o que estiguin assignats i encara no omplerts.

Postcondició: l'usuari (terapeuta o pacient), rep en una taula les dades de l'exercici consultat.

Flux d'esdeveniment principal: el pacient, mitjançant l'accés "tractament", pot entrar dins l'apartat exercicis tests o exercicis formulari, per consultar les dades d'aquests.

Flux d'esdeveniments alternatius: en el cas que el terapeuta no hagi assignat cap exercici, el pacient simplement no veurà cap dada a la taula.



## Cas d'ús 18 : Omplir Exercicis tipus Formulari

Resum de la funcionalitat: permet al pacient omplir els exercicis tipus formulari del seu mòdul de tractament.

Un exercici tipus formulari consta de 2 preguntes obertes, amb respostes explicatives que poden ser extenses, i que s'omplen dins de dues petites àrees de text de 1000 caràcters cadascuna.

Paper dins del treball de l'usuari: els formularis son un dels dos tipus d'exercici de que consten els mòduls del tractament. Aquest cas d'ús, permet al pacient realitzar unes activitats que formen una part del treball del seu tractament. És una funcionalitat molt semblant al cas d'ús 3 (omplir exercicis tipus test).

Únicament implementaré un exercici tipus formulari.

Actors: pacient

Precondició: el terapeuta ha assignat els mòduls de tractament al pacient. Aquest ha entrat al sistema, i dins l'apartat tractament, consulta els exercicis tipus formulari.

Postcondició: el pacient ha realitzat l'exercici tipus formulari i ho ha enviat.

Flux d'esdeveniment principal: després que el terapeuta ha assignat els mòduls de tractament al pacient, aquest pot entrar, mitjançant l'accés "tractament", i dins l'apartat exercicis formulari, pot obrir un test per omplir-ho i enviar els resultats.

Flux d'esdeveniments alternatius: en el cas que els mòduls de tractament encara no estiguin assignats al pacient, aquest quan intenti accedir a l'apartat "tractament", rebrà un missatge d'avís conforme encara no té cap tractament.

Una vegada fet el informe d'avaluació per part del terapeuta, el pacient no pot modificar les respostes dels formularis, i rebrà un missatge d'avís explicatiu de la situació. Mentrestant pot refer-los per canviar alguna qüestió o afegir alguna resposta que cregui necessària.

A l'hora de omplir els formularis, si no respon a cap o alguna de les preguntes, no podrà enviar-los, i el sistema mostrarà els avisos en els camps que no hagin estat omplerts.

En el cas que l'usuari surti de l'aplicació, i no hagi enviat les dades, aquestes es perdran.

## **Cas d'ús 19 : Consultar Diagnòstic**

Resum de la funcionalitat: pacient i terapeuta poden consultar el diagnòstic realitzat.

Paper dins del treball de l'usuari: hi ha un diagnòstic per a tot el procés de tractament. Es un apartat que pot consultar el pacient en qualsevol moment dins el seu procés de tractament.

També ho pot consultar el terapeuta per revisar-ho, o com a pas previ a la seva modificació.

Actors: terapeuta i pacient.

Precondició: en el cas del pacient, aquest ha omplert el Formulari Diagnòstic, ho ha enviat, i el terapeuta ha confeccionat el diagnòstic. En el cas del terapeuta, ha realitzat una cerca i consulta del pacient i ha accedit a l'accés directe "Diagnòstic", de les dades rebudes

Postcondició: l'usuari (terapeuta o pacient) ha pogut veure el informe del diagnòstic.

Flux d'esdeveniment principal: després que el terapeuta hagi confeccionat el diagnòstic, el pacient pot veure el informe, i així conèixer quin és exactament el problema que té, valorat per un professional. Per aquesta tasca, accedeix a l'apartat "Diagnòstic" de la seva pàgina personal, i rep les dades del informe del diagnòstic.

A més és un cas d'ús que pot accedir el terapeuta com a pas previ per a modificar o confeccionar el diagnòstic.

Flux d'esdeveniments alternatius: si el pacient encara no té el diagnòstic, simplement veurà el informe buit.

## **Cas d'ús 20 : Modificar Avaluació**

Resum de la funcionalitat: el terapeuta modifica l'avaluació d'un pacient, o la edita sencera en cas que encara no la hagi confeccionat.

Paper dins del treball de l'usuari: és un cas d'ús al que accedeix el terapeuta quan vol realitzar l'avaluació, o a vegades modificar algun aspecte del informe. Es un cas d'ús fonamental pel desenvolupament del procés de teràpia, ja que recopila informació i fa una valoració en el moment present de quin és l'estat i evolució en que es troba el pacient.

Actors: terapeuta.

Precondició: el terapeuta, després d'haver realitzat una cerca i una consulta del pacient, ha realitzat una consulta de l'avaluació, i ha pitjant en l'accés directe "Editar avaluació" dins la taula de dades que rep de la consulta feta.

Postcondició: el terapeuta ha modificat o editat sencer l'avaluació del pacient i l'ha introduït al sistema per a que el pacient la pugui consultar.

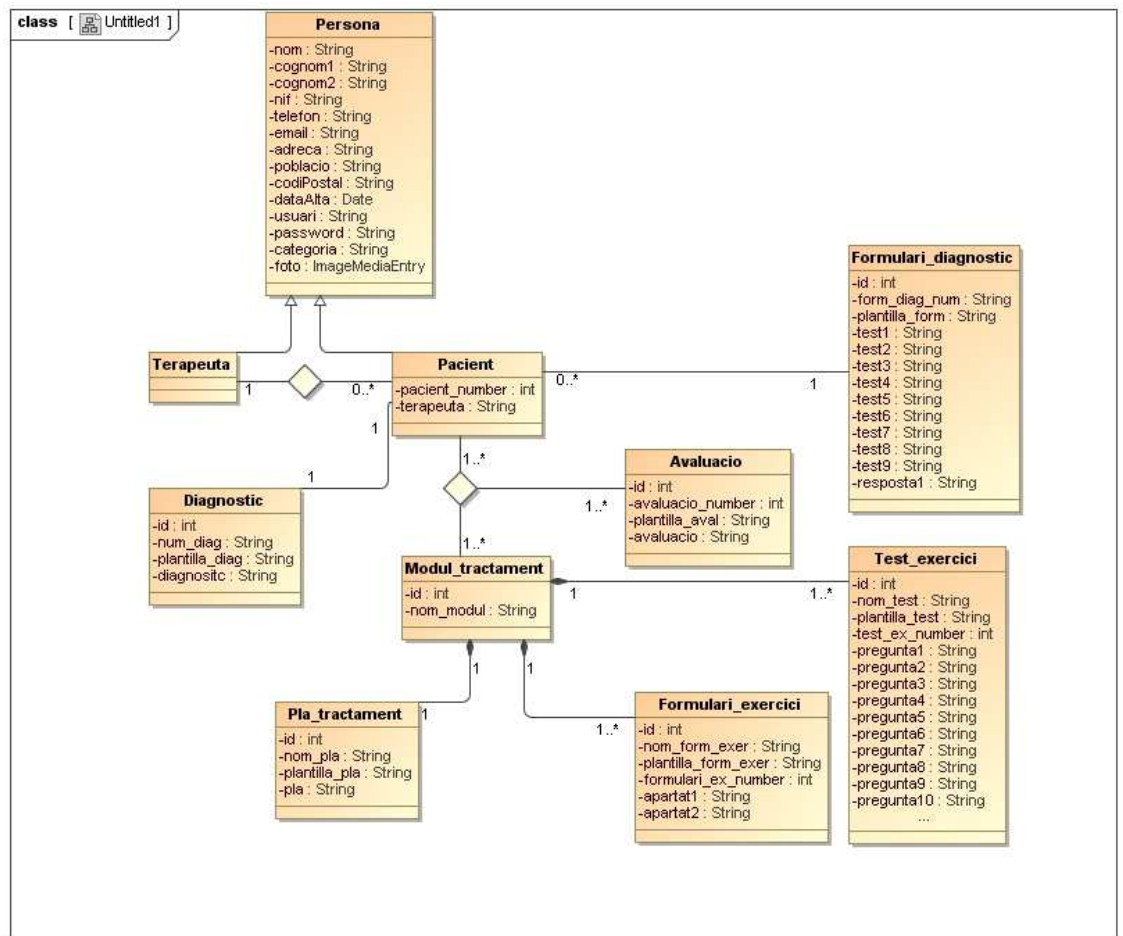
Flux d'esdeveniment principal: el terapeuta pot modificar i/o eliminar l'avaluació. En cas que no hi hagi, perquè es tracta de la primera, pot confeccionar-la sencera. Una vegada lliurada al pacient, no té sentit modificar-la a excepció que trobi alguna errada, o cap qüestió important que no havia inclòs.

Flux d'esdeveniments alternatius: el terapeuta avisa al pacient mitjançant e-mail de la modificació o confecció l'avaluació.

## 6.2 Diagrama de classes

El diagrama resultant és el següent: podem apreciar la generalització de les classes Pacient i Terapeuta amb la classe Persona.

També es pot ressaltar la classe Avaluació, que recull l'avaluació dels mòduls de tractament per cada pacient.

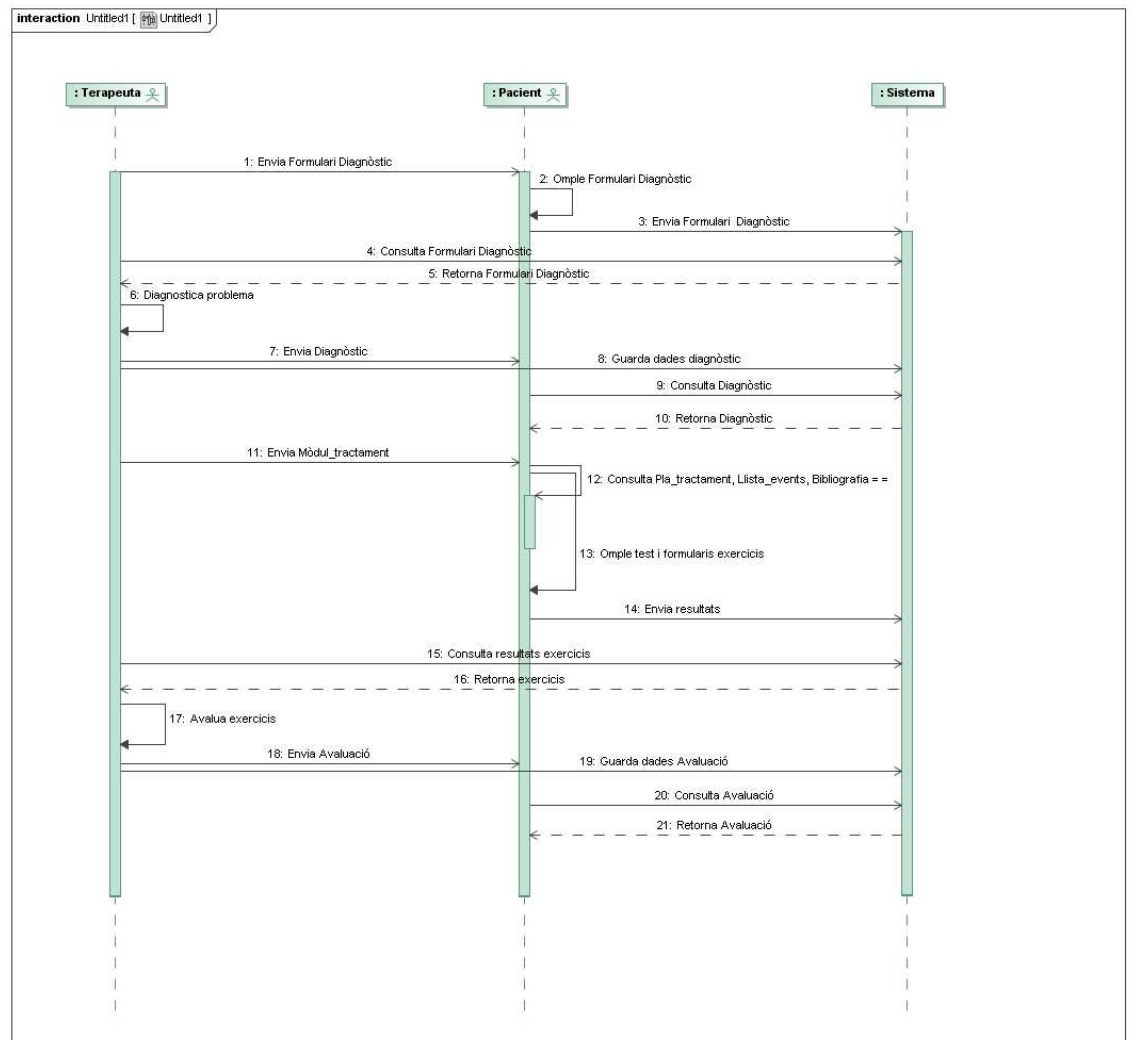


Il·lustració 3. Diagrama de classes

### 6.3 Diagrama de seqüència:

Per al bon enteniment de la seqüència de funcionament global entre un pacient i el terapeuta, serà necessari un sol diagrama, donat que els processos parcials son molt senzills i no requereixen entrar en més detall.

En el diagrama incloem l'objecte sistema que simbolitza els processos que es realitzen automàticament, sense l'acció directa dels actors terapeuta o pacient.



Il·lustració 4. Diagrama de seqüència

## 7. Disseny

### 7.1 Disseny de l'arquitectura

En aquesta fase es prenen les decisions sobre la utilització de les tecnologies i les seves configuracions.

#### 7.1.1 Què és la plataforma J2EE?

Java Platform Enterprise Edition o J2EE és una plataforma de codi obert, per desenvolupar i executar software d'aplicacions en llenguatge Java, amb una arquitectura multicapa i distribuïda, que es recolza en components modulars efectuant-se sobre un servidor d'aplicacions. És un estàndard, i esta definida per una especificació.

J2EE està orientat a la implementació d'arquitectures distribuïdes de grans dimensions, com son les aplicacions empresarials.

Aquest model multicapa que proposa, està dividit en 3 capes de n-nivells a la capa intermèdia:

- **Capa client.** Conté els components que s'executen a les màquines client, com les pàgines HTML en un navegador, o els applets.
- **Capa intermèdia.** Engloba els components que processen la lògica de negoci. Aquests s'executen dins de contenidors, dins del servidor d'aplicacions. Aquesta capa es subdivideix en tres nivells:
  - Capa de presentació: conté la lògica de la interacció entre l'usuari i l'aplicació, i la forma en què es presenta la informació. Els seus components principals son Servlets i JSP's. Es pot executar en un contenidor web, o en el client en les aplicacions d'escriptori.
  - Capa de negoci: conté les regles de negoci de l'aplicació i els components de negoci, com son els EJB's o com les classes POJO.
  - Capa d'integració: on s'integra l'aplicació amb els diferents sistemes externs, o gestors de la informació a interactuar. Conté components com EJB's d'entitat, classes POJO , o Mapatge O-R.
- **Capa Eis.** Conté els components de la capa de dades persistents, que resideixen als servidors de BBDD.

Aquesta separació entre capes, ofereix una delimitació de responsabilitats, aportant escalabilitat, extensibilitat i flexibilitat.

## 7.1.2 Què és un patró de disseny de software?

Un patró de disseny es una solució a un problema de disseny, que conté unes característiques tals com efectivitat en la resolució de problemes similars en repetides ocasions, i reusabilitat per ser aplicable a problemes de disseny en diverses circumstàncies.

Per tant, és aconsellable l'ús de patrons per evitar cercar solucions a problemes ja coneguts i solucionats anteriorment. Els patrons, ens donen unes garanties d'èxit, degut a que faciliten una solució provada exhaustivament, i estandarditzen el nostre disseny. Tot i així, no sempre serà el seu ús la solució més adequada entre totes les que existeixen.

Hi han diversos tipus segons l'escalabilitat o nivell d'abstracció:

- **Patrons d'arquitectura.** Contenen un esquema organitzatiu estructural fonamental per a sistemes de software.
- **Patrons de disseny.** Contenen esquemes per definir estructures de disseny, o relacions amb les que construir sistemes de software.
- **Dialectes.** Patrons de baix nivell específics per a un llenguatge de programació o entorn concret.

### Patrons per a la capa de presentació:

#### MVC (Model-Vista-Controlador)

És un patró d'arquitectura, que separa el sistema en tres components diferents. El codi de la presentació, el codi de la lògica de negoci, i el codi d'accés a les dades.

Aquests tres components son:

- **Model:** correspon a la lògica de negoci. Es la representació específica de la informació amb la que opera el sistema.
- **Vista:** correspon a la capa de presentació. És la interfície amb l'usuari de l'aplicació i tots els components de presentació.
- **Controlador:** correspon a l'accés a les dades. Comunica vista i model. Respon o processa les entrades de l'usuari, actualitza el model i modifica les vistes.

La utilització d'aquest patró, permet una major reutilització de codi ja que una modificació en un component no afecta als altres. A més, permet una

major especialització en els desenvolupadors, donat que les seves competències es poden separar en àrees concretes (separació de tasques i responsabilitats).

Com a inconvenient trobem que hi ha una certa corba d'aprenentatge per a la seva implantació, i que no sempre es poden desacoblar completament les vistes del model.

### **Patrons per a la capa d'integració:**

#### **DAO (Data Access Object)**

Mitjançant la utilització d'aquest patró de disseny es pretenen facilitar mètodes d'abstracció i encapsulament d'accés a dades.

DAO ofereix una solució per donar la màxima independència a una aplicació respecte el SGBD que hi ha al darrera. Amb aquest patró, podem encapsular tots els accessos a l'origen de dades per obtenir i emmagatzemar les dades. O sigui, DAO encapsula tant la font de les dades com la manera d'accedir a aquestes, dotant a les aplicacions d'una gran portabilitat.

### **7.1.3 Què és un framework?**

Un framework és la implementació de diferents patrons de disseny, que facilita la reutilització de disseny i del codi, i agilitza el desenvolupament d'aplicacions.

#### **Frameworks per a la capa de presentació:**

Hi han diversos frameworks per a aquesta capa: Struts2, JSF(Java Server faces), Spring MVC, etc.

#### **Struts2**

És un framework open-source basat en el patró MVC. Aquest patró, com he comentat anteriorment, té com a finalitat reduir l'acoblament, dividint les responsabilitats en 3 capes.

Actualment és un referent en el desenvolupament d'aplicacions web, especialment indicat per a aplicacions amb un nombre elevat de pàgines i que necessitin un manteniment fàcil al llarg del temps.

Struts2 s'encarregarà d'obtenir i transformar els paràmetres de les peticions del client, validar les dades, cercar quina acció ha de realitzar en el model segons la petició, cercar la pàgina per mostrar a l'usuari segons la resposta del model, proporcionar les dades del model a la vista; s'ocuparà de la internacionalització, etc.



Struts2 posseeix extensions com **Validator**, que s'utilitza per la validació de la informació que un usuari introdueix via web.

Així podem tenir un control centralitzat de la validació de la web, sense tenir en compte la tecnologia de la vista.

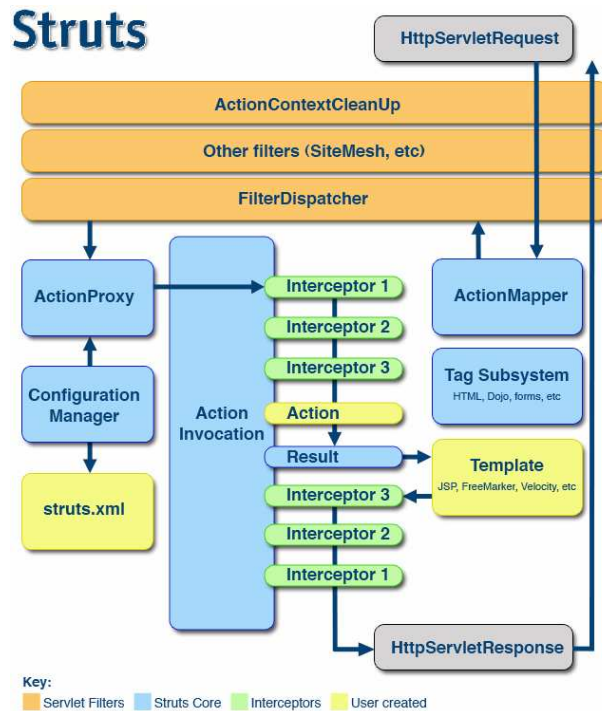
Per al model, Struts2 implementa els Actions. Per a la vista, els Results. I per al controlador, el filtre FilterDispatcher. Aquest seria un resum del cicle de vida d'una petició:

1. Un usuari envia una petició.
2. L'element FilterDispatcher determina l'acció que haurà de respondre: el framework disposa dels elements perquè el Dispatcher sigui capaç de determinar què Action és el responsable de rebre la petició i processar-la.
3. S'apliquen els interceptors definits: Hi ha diferents interceptors que es poden configurar perquè executin diferents funcionalitats com workflows, validacions, upload de fitxers, etc.
4. S'executa el Action: després de l'execució dels diferents interceptors el mètode específic del Action és executat, realitzant les operacions i accions que s'hagin definit. El action acaba retornant un resultat que s'utilitza per determinar la pàgina a tornar.
5. Es renderitza la sortida: després de l'execució de l'action es determina quina és la pàgina que es retorna i s'executa el redireccionament a aquesta pàgina.
6. Es retorna la petició: per a realitzar la devolució s'executen els interceptors que corresponguin i es procedeix a retornar la petició al client.
7. Es mostra el resultat al client final: el control és retornat al client, que podrà visualitzar el resultat al navegador.

En resum, els punts a destacar d'Struts2 son:

- Framework robust amb àmplia experiència en aplicació de diferents projectes, i amb extensa documentació i exemples.
- Orientat a la petició/resposta de la web.
- La corba d'aprenentatge no és elevada.
- Disposa d'una llibreria de tags de suport per a la capa de presentació.
- Ofereix serveis de validació, i de internacionalització amb (i18n).

La lògica general d'Struts2 es pot resumir en el següent esquema:



**Il·lustració 5. Lògica de Struts2**

**Frameworks per a la capa d'integració:**

Hi han diversos frameworks per a la persistència de les dades. Entre ells, el més usat i estès es Hibernate.

**Hibernate**

És una eina poderosa, és un ORM (Object Relational Modeler), o sigui una eina de mapeig d'atributs entre una base de dades relacional i el model d'objectes d'una aplicació. Utilitza arxius XML o anotacions en los beans de les entitats per a establir aquestes relacions.

Implementa automàticament el patró DAO, explicat anteriorment.

Hibernate soluciona el problema que existeix de la diferència entre dos models de dades coexistents en una aplicació: el que es usat a la memòria de la computadora (orientació a objectes) i el utilitzat en las bases de dades (model relacional).

Hibernate convertirà les dades entre els tipus utilitzats per Java i els definits per SQL. Hibernate genera les sentències SQL i allibera al desenvolupador del maneig manual de les dades que resulten de l'execució d'aquestes sentències, mantenint la portabilitat entre tots els motors de bases de dades amb un lleuger increment en el temps d'execució.

També té la funcionalitat de crear la base de dades a partir de la informació disponible.

Hibernate ofereix un llenguatge de consulta de dades anomenat HQL

(Hibernate Query Language).

Avui dia és un estàndard i un referent entre els diferents ORM.

### **7.1.4 Elecció de l'arquitectura del TFC**

Atès que la separació entre capes aporta una delimitació de responsabilitats i ofereix un alt grau d'escalabilitat, extensibilitat i flexibilitat, he volgut aprofitar aquests avantatges de l'arquitectura J2EE i del patró MVC, per crear un projecte modular, adaptable i amb la major portabilitat possible.

#### **Arquitectura per a la capa de presentació:**

He escollit el framework Struts2 enfront altres com JSF, o Spring MVC, perquè el projecte no requereix el disseny de pantalles de gran complexitat, i perquè la cova d'aprenentatge es assequible. A més utilitzaré el servei de validació de dades, propi d'Struts2.

Struts2 elimina de les JSP's tota la lògica de procés que no vagi encaminada exclusivament a la presentació. El benefici que obtenim és la separació entre la presentació i el contingut de les dades.

#### **Arquitectura per a la capa de negoci:**

La utilització de POJO's és la meua proposta, ja que la lògica de negoci és més senzilla d'implementar. Donat que la meua experiència prèvia en aquestes tecnologies és nul·la, i que el temps per el desenvolupament del TFC és molt limitat, sóc conscient que amb aquesta decisió, no aprofito els avantatges que ofereix J2EE en quant a transaccions, seguretat, etc.

#### **Arquitectura per a la capa d'integració:**

He optat per Hibernate3 com la millor solució per a l'encapsulament de l'accés a les dades. D'una banda per tot el que ofereix com a potent ORM, i per la seva estandardització i implantació entre la comunitat de desenvolupadors. D'altra banda, la meua intenció d'aprofundir coneixements en el treball amb arxius XML, tot i que podria haver utilitzar les annotations pròpies de Hibernate per estalviar-me aquesta tasca. En el cas d'utilitzar les annotations, cal tenir en compte que hem de tenir una versió de java 1.4 o superior.

## Requeriments de programari :

- **En el client:**  
Únicament un navegador web
- **En el servidor:**  
Com a contenidor web, **Tomcat 6**, de Apache Software Foundation. Implementa les especificacions dels servlets i de les JSP. Tomcat és open-source i ha estat programat en Java, per tant funciona sobre qualsevol SO que disposi de la màquina virtual Java.

En quant a la decisió de l'SGBD per a la persistència de les dades, m'he decantat per **Oracle 10g XE**, que a més és una versió gratuïta. Oracle és un sistema de gestió de base de dades objecte-relacional, i es considera com un dels sistemes de bases de dades més complets, destacant el suport a transaccions, escalabilitat, estabilitat i suport multiplataforma. El seu domini en el mercat de servidors d'empresa ha estat gairebé total fins fa poc temps amb la competència d'altres com Microsoft SQL Server, o com MySQL i PostgreSQL de llicència lliure.

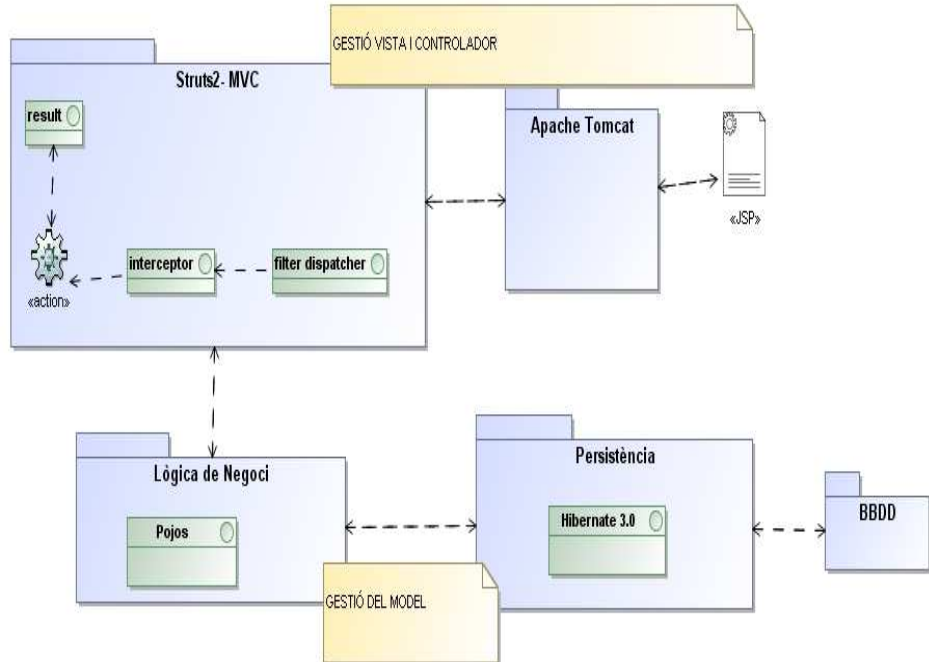
Com a entorn de desenvolupament, encara que no és necessari, és molt recomanable utilitzar-lo, he escollit **Eclipse**, en la seva versió Ganymede.

Es tracta d'un entorn de desenvolupament integrat (IDE, Integrated Development Environment) que facilita enormement les tasques de edició, compilació i execució de programes durant la seva fase de desenvolupament. Encara que Eclipse pretén ser un entorn versàtil suportant diversos llenguatges de programació, és amb el llenguatge Java amb el que millor s'integra i amb el qual ha guanyat la seva popularitat.

Eclipse és una aplicació gratuïta i de codi obert, del que ja estic una mica familiaritzat i la meva intenció és aprofundir les característiques i possibilitats que ofereix.

Per al disseny de codi HTML i utilització de plantilles e imatges, he obtingut ajuda amb el software de creació de pàgines web Adobe Dreamweaver CS3, donat que la meva experiència i coneixements d'aquest llenguatge de marques era molt precari.

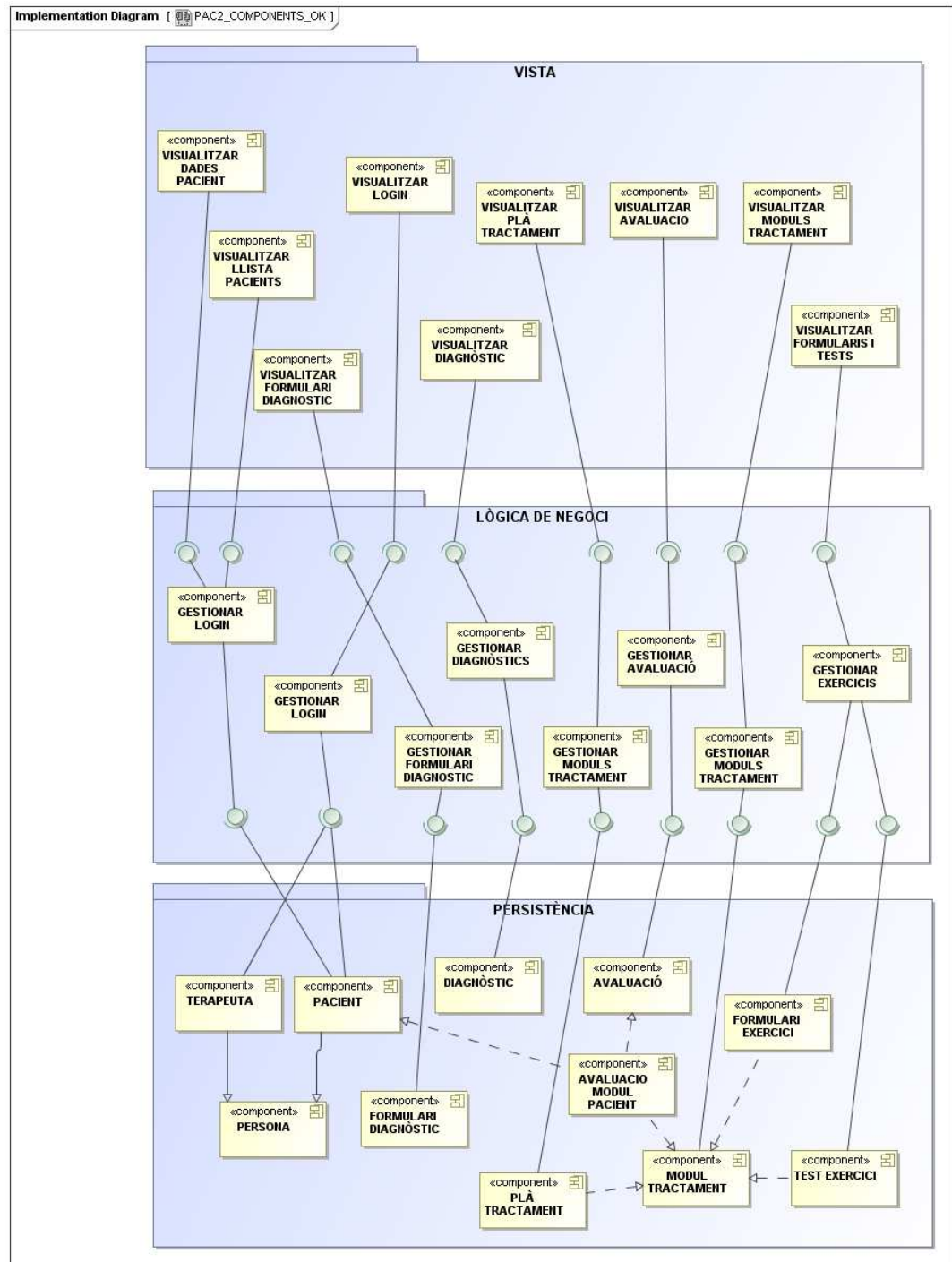
En l'esquema següent, podem observar una visió general de l'arquitectura del sistema amb la utilització dels frameworks Struts2 i Hibernate3, i en el que es denota clarament la separació de capes:



**Il·lustració 6. Arquitectura amb els frameworks Struts2 i Hibernate**

### 7.1.5 Diagrama de components:

En aquest diagrama identifiquem els objectes físics que hi han a la nostra aplicació, i podem veure la separació en les 3 capes. Els components són un conjunt de classes agrupades dins un paquet. Les relacions entre components de diferents capes estan connectades mitjançant interfícies que implementen aquesta independència de capes.



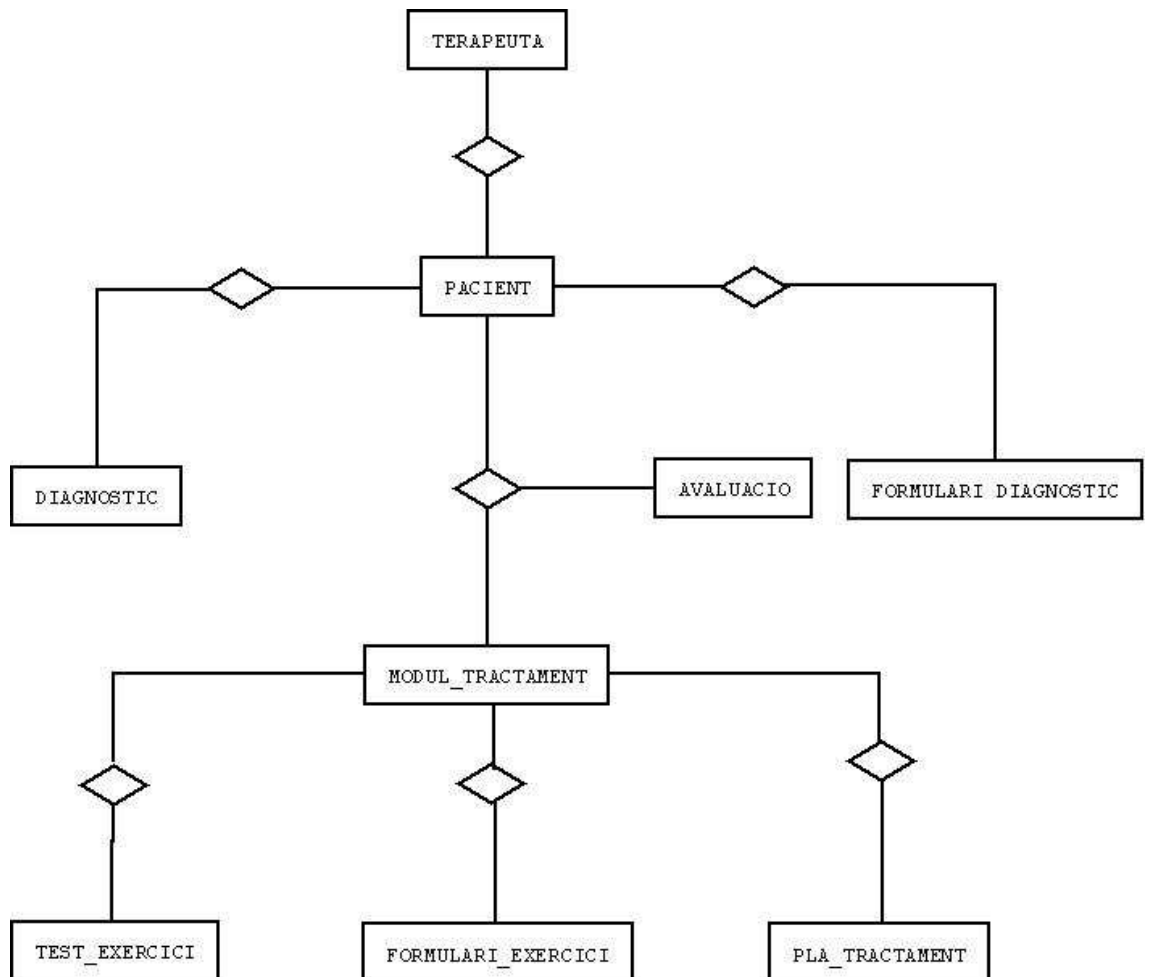
Il·lustració 7. Diagrama de components de l'aplicació

## 7.2 Disseny de la persistència

Quan un procés finalitza i s'allibera la memòria, les dades d'aquest es perden. Per poder mantenir-les, hem d'utilitzar sistemes d'emmagatzemament permanent.

### 7.2.1 Diagrama E/R de la BBDD

El diagrama E/R de la base de dades resultant serà el següent:

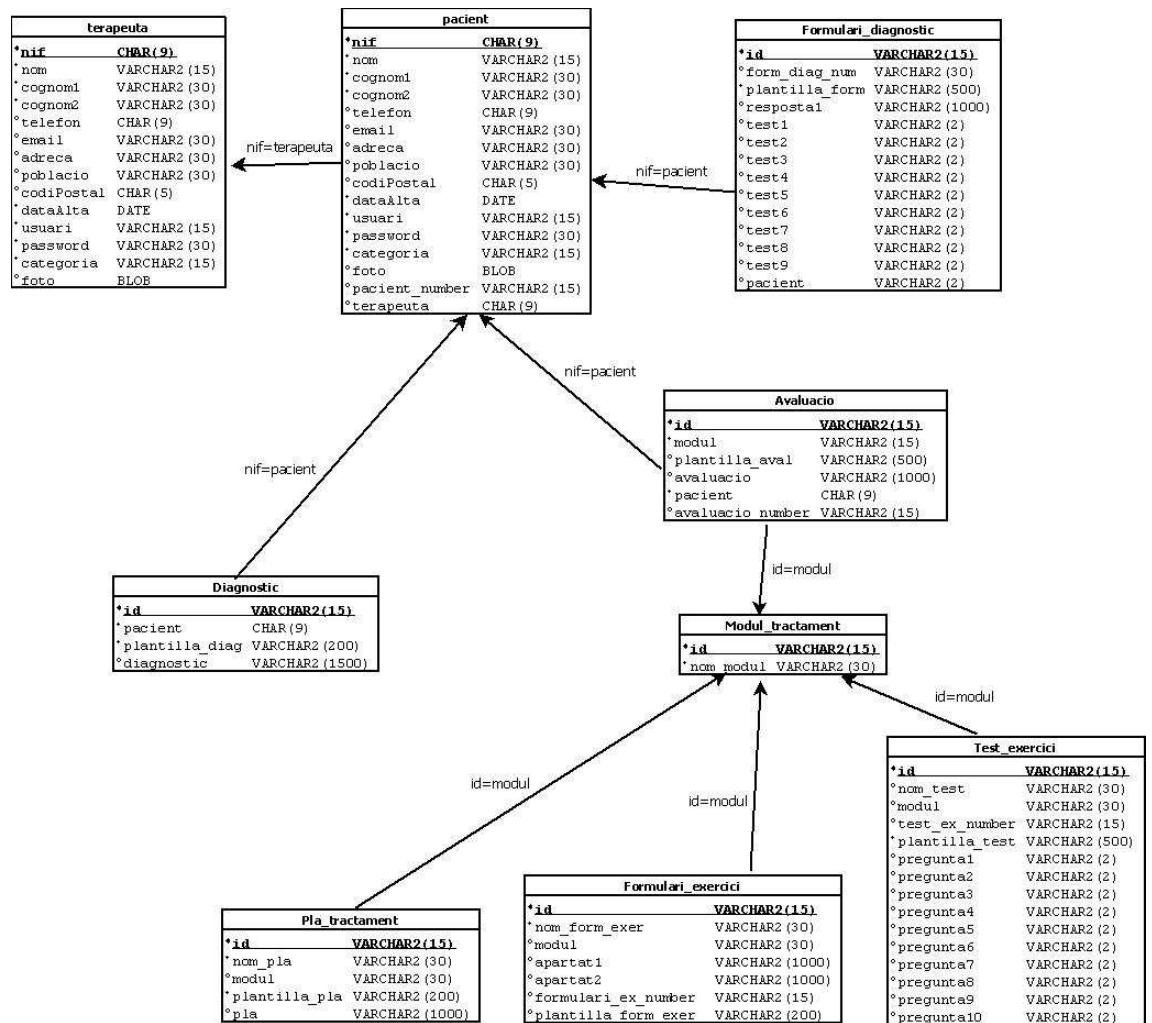


Il·lustració 8. Diagrama E/R

Es pot destacar del diagrama, la relació ternària Pacient-Avaluació-Modul\_tractament, necessària per la persistència de dades referents a l'avaluació d'un mòdul de tractament per a un pacient.

## 7.2.1 Disseny relacional de la BBDD

En aquest esquema podem veure el detall i tipus dels atributs tal i com seran a la BBDD. Declarant les claus primàries i forànies.



Il·lustració 9. Disseny relacional de la BBDD



## 7.3 Disseny de la interfície d'usuari

A continuació exposaré una selecció de la implementació final de les pàgines més significatives, tal i com les veuran els usuaris de l'aplicació.

### Pàgina de presentació

En aquesta pàgina, tenim la descripció de què ofereix l'empresa, qui forma part en ella, i tota la informació necessària perquè un usuari tingui una visió clara dels serveis a que pot optar.



Il·lustració 10. Pàgina de presentació

## Entrada al sistema

Pàgina per l'accés al sistema mitjançant el NIF de l'usuari i una contrasenya. Si succeeix algun error, ens mostrarà els avisos adients especificant el motiu de la seva aparició.



Il·lustració 11. Entrada al sistema

## Pàgina personal

Pàgina de benvinguda a l'usuari. Podem veure el menú de funcionalitats a l'esquerra que es manté al llarg de la navegació dins l'aplicació, i que varia en el seu contingut segons el tipus d'usuari que ha entrat al sistema.



Il·lustració 12. Pàgina personal

## Consultar dades (pacient)

El pacient pot consultar les seves dades personals.



Il·lustració 13. Consultar dades

## Consultar Pla de tractament (pacient)

El pacient pot consultar el seu Pla de tractament.



Il·lustració 14. Consultar Pla de tractament

## Consultar Diagnòstic (pacient)

El pacient pot consultar el Diagnòstic que ha elaborat el seu terapeuta.



Il·lustració 15. Consultar Diagnòstic

## Consultar Avaluació (pacient)

El pacient pot consultar l'Avaluació de cada mòdul que ha realitzat el seu terapeuta a partir del treball realitzat amb els exercicis.



Il·lustració 16. Consultar Avaluació

## Omplir exercicis tests (pacient)

El pacient pot omplir els exercicis tipus test del seu Mòdul de tractament.



Il·lustració 17. Omplir Exercicis Test

## Consultar pacient (terapeuta)

El terapeuta realitza consultes de les dades dels pacients.



Il·lustració 18. Consultar pacient

## Modificar diagnòstic (terapeuta)

El terapeuta modifica el diagnòstic que havia confeccionat per al pacient..



Il·lustració 19. Modificar diagnòstic

## Editar Avaluació (terapeuta)

El terapeuta realitza l'avaluació del treball del pacient



Il·lustració 20. Modificar Avaluació

## Modificar dades pacient (terapeuta)

El terapeuta pot modificar les dades dels pacients cercats.



Il·lustració 21. Modificar dades pacient

## Assignar mòduls de tractament (terapeuta)

El terapeuta assigna el mòdul de tractament al pacient.



Il·lustració 22. Assignar Mòdul tractament

## 8. Implementació

### 8.1. Instal·lació del software

#### 8.1.1 Instal·lació de Java (JDK)

Necessitem la *JRE (Java Runtime Environment)* per a executar aplicacions Java, i també per l' IDE *Eclipse*. El paquet *JDK* es pot baixar de: <http://www.oracle.com/technetwork/java/javase/downloads/index.html> on hem d'escollir a més el tipus de paquet adequat segons el nostre SO.

#### 8.1.2 Instal·lació de l'IDE Eclipse, i del contenidor web Tomcat:

Vegem els passos a seguir per aconseguir aquesta interacció.

Actualment, a partir de la versió *Eclipse 3.4, Eclipse Ganymede*, podem iniciar i aturar *Tomcat* des de *Eclipse* sense cap Plug-in afegit.

Descarreguem l'entorn de desenvolupament [Eclipse](#) des de la seva pàgina web. La versió que ens interessa, ja que anem a desenvolupar amb *J2EE*, és l'anomenada *Eclipse IDE for Java EE Developers*.

Descarreguem i descomprimim el zip de l'última versió core de [Tomcat](#).

#### Configurar Servidor Tomcat en Eclipse:

Iniciem *Eclipse*. Ens dirigim a Window -> Preferences -> Server -> Runtime Environment, premem Add i seleccionem la nostra versió de *Tomcat*.

Marquem Create a new local server si no està seleccionat.

Premem Next i busquem el directori on instal·lem *Tomcat*. Premem Finish i OK. El nou servidor hauria de mostrar-se en la pestanya Servers. També s'haurà creat un projecte nou Servers amb els arxius de configuració de la nostra instància de *Tomcat*.

Amb això ja tenim integrat el nostre servidor *Tomcat* per al desenvolupament de projectes web.



### Creació d'un projecte Web:

Ara hem de crear un projecte web, per això fem:

*File -> New -> Other ... -> Web -> Dynamic Web Project*

Aquí assignem el nom del nostre projecte, i veiem que els camps de "Target Runtime" i "Configuration" ja estan configurats amb la informació del nostre servidor *Tomcat*. Finalment premem a "Finalitzar" i es crea el nostre projecte. És destacable en aquest punt que *Eclipse* ens proposarà canviar la perspectiva a "*Java EE*" la qual cosa és necessari per continuar treballant. Les perspectives de eclipsi es poden canviar fàcilment després en:

*Windows -> Open Perspective*

Un cop acabada l'aplicació només resta crear un arxiu WAR per poder distribuir fàcilment. Fem clic amb el botó dret sobre el nostre projecte, seleccionem *Export -> WAR File*, i introduïm el nom a utilitzar i la ruta en què volem guardar el WAR.

### 8.1.3 Instal·lació de Struts2

Descarregar les llibreries del Framework [Struts2](#) necessàries.

#### **Configurar Struts2 manualment:**

Ara que el nostre projecte està creat i el nostre servidor està configurat, només falta configurar *Struts2*.

El primer que hem de fer és incloure les llibreries en el nostre projecte.

Hem descarregat "struts-2.1.8.1-all.zip" de

<http://struts.apache.org/download.cgi>. Si el descomprimim ens trobem amb la carpeta "Lib", on es troben totes les llibreries de *Struts2*, i les més comuns o bàsiques son les següents:

struts2-core-2.1.6.jar  
xwork-2.1.2.jar  
ognl-2.6.11.jar  
freemarker-2.3.13.jar  
commons-logging-1.0.4.jar  
commons-fileupload-1.2.1.jar

Dins de la mateixa carpeta es troben moltes més llibreries que es poden incorporar a mesura que les anem requerint. A més en el paquet que baixem hi ha la documentació oficial, aplicacions d'exemples i el codi font.

Aquestes llibreries les afegim al nostre projecte simplement arrossegant a:

*"nomProjecte" -> WebContent -> WEB-INF -> Lib*

Modificar web.xml:

El projecte té un arxiu descriptor encarregat de la configuració de l'aplicació web, aquest és l'arxiu web.xml. Hem de modificar aquest fitxer afegint el filtre que gestiona les peticions a Struts2.

L'arxiu es troba a: "*nomProjecte*" -> *WebContent* -> *WEB-INF*  
i hem d'afegir les següents línies:

```
<filter>
  <filter-name> struts2 </ filter-name>
  <filter-class> org.apache.struts2.dispatcher.FilterDispatcher </ filter-class>
</ Filter>
<filter-mapping>
  <filter-name> struts2 </ filter-name>
  <url-pattern> / * </ url-pattern>
</ Filter-mapping>
```

Amb això el que estem fent és indicar a *Apache Tomcat* que tota petició ha de ser interceptada i gestionada pel filtre "*FilterDispatcher*", que és el controlador de la nostra aplicació.

Crear struts.xml:

*struts.xml* és un dels arxius més importants en el framework, ja que és l'arxiu de configuració de *Struts2*. Aquest s'ha de crear a:

"*nomProjecte*" -> *Java Resources: src*

```
<! DOCTYPE struts PUBLIC
"- // Apache Software Foundation // DTD Struts Configuration 2.0 // EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
  <constant name="struts.enable.DynamicMethodInvocation" value="false" />
  <constant name="struts.devMode" value="false" />

  <package name="default" namespace="/" extends="struts-default">

    // Declaració d'Accions

  </ Package>
</ Struts>
```

En aquest arxiu es defineixen les Accions de *Struts2* definint així el mapping de l'aplicació, amb això l'aplicació sap a què pàgina redirigir en resposta a una acció i que classe s'encarrega de realitzar l'acció. Això es detallarà més endavant, ara cal saber només que en aquest arxiu es configura la nostra aplicació.

**Configurar Struts2 [usant struts2-blank-2.1.8.1.war]:**

Una altra forma de configurar struts2 és usant "*struts2-blank-2.1.8.1.war*" que és una aplicació d'exemple que ve en el paquet "*struts-2.1.8.1-all.zip*".

En descomprimir, aquesta aplicació es troba a:

*Struts-2.1.8.1 -> apps*

Per crear el nostre projecte simplement importem aquest fitxer a:

*File -> Import ... -> Web -> WAR file*

Ara a "WAR file" seleccionem l'arxiu "*struts2-blank-2.1.8.1.war*", i en "Web project" donem el nom del nostre projecte: "*nomProjecte*".

**Què son les Accions?:**

Les accions són les encarregades de definir quina serà la classe que s'encarregarà de processar la petició sol·licitada i quina serà la resposta depenent del resultat. Aquestes accions es registren dins *struts.xml*. L'estructura bàsica d'una acció és la següent:

```
<action name="Nombre_Accion" class="Paquete.Nombre_Clase">
  <result name="error"> / pagina_error.jsp </ result>
  <result name="success"> / pagina_exito.jsp </ result>
</ action>
```

Nombre\_Accion: identificador pel qual es crida a l'acció des de la "Vista" (fitxers JSP).

Paquete.Nombre\_Clase: Controlador encarregat de realitzar la lògica de l'acció. És una classe java, situada en algun paquet.

pagina\_exito.jsp / pagina\_error.jsp: Són jsp's que depenent si la resposta del controlador és èxit o error seran visualitzats.

**8.1.4 Instal·lació de Hibernate3**

Ens dirigim a la següent pàgina web de SourceForge: [Hibernate3](#), i llavors seleccionem per descarregar-la, el vincle amb la versió 3.6.0.Final.

En aquest exemple de descàrrega d'*Hibernate* farem clic sobre el lligam *hibernate-distribution-3.6.0.final-dist.zip*.

Aquest fitxer el podem extreure en el directori arrel del IDE *Eclipse*.

Aquí tenim la relació de fitxers .jar que necessitarem:

- hibernate-distribution-3.6.0.Final
  - hibernate3.jar
  
- hibernate-distribution- 3.6.0.Final/lib/required
  - antlr-2.7.6.jar
  - commons-collections-3.1.jar
  - dom4j-1.6.1.jar
  - javassist-3.12.0.GA.jar
  - jta-1.1.jar
  - slf4j-api-1.6.1.jar
  
- hibernate-distribution-3.6.0.Final/lib/jpa
  - hibernate-jpa-2.0-api-1.0.0.Final

### **La sessió Hibernate:**

Una càrrega de l'objecte mapejat (de persistència) de la base de dades o l'actualització (update, inserció o eliminació) només es pot fer si una sessió *Hibernate* és instanciada. La sessió és una capa representada per la interfície Java org. hibernate.Session  
Per a un desenvolupador d'una aplicació Java, l'activitat *Hibernate* comença després de la instrucció:

```
session = sessionFactory.openSession ()
```

En el que *SessionFactory* és un objecte creat per l'aplicació i utilitza la connexió JDBC (o JNDI).

L'activitat *Hibernate* acaba després de l'anomenat del mètode:

```
session.close ()
```

Entre aquests dos mètodes esmentats anteriorment, una transacció pot ser executada mitjançant:

```
session.beginTransaction ()
```

La classe org.hibernate.cfg.Configuration: el constructor de Configuració utilitza el fitxer de configuració d'*Hibernate* (*hibernate.cfg.xml*).

La classe org.hibernate.HibernateException: pràcticament tots els mètodes *Hibernate* poden provocar aquesta excepció.

### 8.1.5 Instal·lació de Oracle 10g XE

Descarreguem l'SGBD [Oracle](#) i escollim el paquet adequat al nostre SO i amb la versió “universal”.

Executem l'arxiu .exe descarregat, i seguim els passos tal i com s'indica a la guia oficial d'instal·lació d'Oracle 10:

[http://download.oracle.com/docs/cd/B25329\\_01/doc/install.102/b25143/toc.htm](http://download.oracle.com/docs/cd/B25329_01/doc/install.102/b25143/toc.htm)

En aquesta pàgina s'indiquen els requeriments mínims per a l'instal·lació i les incompatibilitats detectades amb software de seguretat com antivirus i firewalls.

## 8.2 Desenvolupament de l'aplicació

El desenvolupament ha estat un procés amb cicle de vida iteratiu e incremental basat en el cicle de vida en cascada, tal com queda explicat al començament en el Resum. Així amb aquest model, he anat implementant cada cas d'ús individualment, i perfeccionant-los a cada iteració. En els següents apartats, explicaré breument el guió de l'estratègia en la implementació de les 3 capes.

### 8.2.1 Implementació de la capa de presentació d'Espai Nirvana

Realització de la implementació de la capa de presentació fent ús del Framework *Struts2*, i els serveis d'internacionalització i18n de *Struts2*.

- Modificar l'arxiu *web.xml*, per afegir el filtre que gestiona les peticions a *Struts2*.
- Creació de les pàgines JSP.
- Definir les accions de *Struts2*, els Actions.
- Crear les validacions dels formularis.
- Fer ús dels serveis i18n de internacionalització per al desenvolupament. I crear l'arxiu amb els missatges d'error.
- Desenvolupar les classes (POJO) necessàries corresponents als Actions.

## 8.2.2 Implementació de la capa de negoci d'Espai Nirvana

En aquesta capa he implementat POJO's, per la seva senzillesa i perquè no necessitem més funcionalitats en aquesta capa de l'aplicació.

- Desenvolupar les classes per la comunicació entre la capa de negoci i la capa d'integració.
- Utilització del llenguatge HQL d'*Hibernate*, per a la realització de consultes a la BBDD.
- Utilitzar un mapatge O-R mitjançant fitxers XML (*nomFitxer.hbm.xml*) entre les taules i les classes, per tal de tractar els valors de forma persistent . La cardinalitat de les associacions entre classes també queda implementada en aquests fitxers XML, tal com son relacions many-to-many, one-to-one, etc.

## 8.2.3 Implementació de la capa d'integració d'Espai Nirvana

En aquesta capa he usat *Hibernate* com a framework d'integració.

- Desenvolupar les classes (POJO) necessàries que corresponen a les taules de la BBDD.
- Modificar el fitxer de configuració d'*Hibernate* (*hibernate.cfg.xml*) amb els paràmetres, dins de *session-factory*, de connexió a la BBDD juntament amb els drivers, el tipus de dialecte SQL, etc . També hem d'indicar en aquest fitxer de configuració, els diferents fitxers de mapatge classe-taula, que son els (*nomFitxer.hbm.xml*), definits a la capa de negoci.

## 9. Resultats

El resultat obtingut és un producte amb un conjunt de funcionalitats ben definides i acabades, dissenyades amb la consigna d'obtenir senzillesa en l'ús, claredat en l'exposició, i usabilitat en les seves funcions.

Gràcies a un treball ben planificat, un seguiment estricte en el temps, i una forta dedicació en hores, ha quedat finalitzat al 100%.

El projecte original és molt més ambiciós, i per adaptar-me al temps disposat per a la realització del TFC, he hagut d'escollir un conjunt de funcionalitats que permetin donar una visió general de la dinàmica del funcionament de l'aplicació.

Incorporació de millores en el futur:

- Donat que el projecte sencer no queda completat en aquest TFC, en una segona fase, afegiré les funcionalitats que resten, i segurament implementaré el patró DAO per a augmentar encara més la portabilitat de la capa de negoci respecte l'accés a les dades.
- Utilitzaré el component *Tiles* d'*Struts2*, per a al disseny de pàgines JSP amb plantilles, que donaran reutilització a aquestes en altres parts del desenvolupament.  
Així com també l'ús del component de internacionalització *i18n*, per a obtenir una interfície amb diversos idiomes ( català, castellà, i anglès).
- En quant a la navegació en la interfície d'usuari, afegiré informació i mecanismes per conèixer en tot moment la ubicació dins la web i crearé uns accessos millorats.  
Respecte a la presentació de les dades, es poden refinar alguns detalls per millorar la seva visualització, com per exemple, separar en diverses files els resultats d'una consulta de les dades d'un pacient.
- Una vegada finalitzat, tinc el propòsit de provar altres tecnologies que he descartat en aquest Treball, algunes per la seva corba d'aprenentatge, i altres perquè he cregut que no s'adaptarien bé en aquest disseny.

## 10. Conclusions

La realització d'aquest projecte m'ha donat una perspectiva real del que consisteix el desenvolupament de software. M'he adonat de la dificultat de portar a terme una idea inicial que es vol convertir en un producte acabat, seguint un procés d'anàlisi, disseny i implementació, amb una planificació en el temps.

He hagut d'aplicar coneixements adquirits al llarg de la carrera, i sobretot he hagut d'aplicar la tècnica d'aprendre a investigar noves tecnologies. Penso que és fonamental aprendre a aprendre. Potenciar les capacitats de ser autodidacta, desenvolupar un criteri propi, i prendre decisions fonamentades i raonades, en quant a quina tecnologia escollir, analitzant les seves característiques per saber quina s'adapta més al nostre treball.

Aquest aspecte ha estat propiciat encara més perquè no tenia quasi coneixements sobre la tecnologia J2EE, ni tampoc sobre patrons de disseny, ni sobre frameworks (bastiments). Desconeixia la dificultat de la implementació de funcionalitats aparentment senzilles, i de com aquests patrons i frameworks ens donen un ajut considerable, no sols en problemes ja coneguts i resolts, si no que també generen bones pràctiques.

He vist la necessitat de la fase d'anàlisi per estructurar correctament el projecte en les seves funcionalitats, i con una manera de facilitar la incorporació de les modificacions que van sorgint. Tot seguint una planificació de tasques que ens assegura un seguiment i dedicació adient en el temps.

He descobert un món apassionant del que sols conec ara una molt petita part, i que amb la confecció del projecte he anat adquirint molta curiositat per saber més, i per provar noves fórmules de conjuntar tipus d'arquitectures i patrons.

L'objectiu de realitzar aquest TFC amb J2EE era aprendre a crear una web dinàmica, i conèixer les funcionalitats bàsiques de les tecnologies utilitzades. Crec que amb el treball realitzat, aquest objectiu ha estat aconseguit. D'ara en davant manca molt camí per recórrer, en conèixer funcionalitats avançades d'aquestes tecnologies i adquirir molta més experiència i habilitats en la construcció d'aplicacions J2EE.



## 11. Bibliografia

### Llibres:

**Carles Mateu** (2006). *Desenvolupament d'aplicacions web.*(1ª ed.).  
Material UOC. (FUOC • XP06/M2008/01165)  
ISBN: 84-9788-346-2

**Benet Campderrich Falgueras / Recerca informàtica, SL** (2004). *Enginyeria del programari.* (1ª ed.). Material UOC. (FUOC • XP03/05060/02078)  
ISBN: 84-9788-065-X

**Camps i Riba, Josep Maria.** *J2EE. Una plataforma de components distribuïda.*  
Material UOC. (FUOC • P06/11059/01150)

**Pradel i Miquel, Jordi / Raya Martos, José Antonio:** *Catàleg de patrons.*  
Material UOC. (Universitat Oberta de Catalunya • P05/11058/00504 • Mòdul 2)

**Patrick Peak / Nick Heudecker** (2006). *Hibernate Quickly.*  
Greenwich: Manning Publications Co.  
ISBN 1932394419

### Llocs web:

Oracle (darrera consulta juny 2011). *J2EE Patterns Catalog.*  
URL: <http://www.oracle.com/technetwork/java/catalog-137601.html>

Autentia Formación y desarrollo (Java, J2EE, UML, XML,etc.). *Tutoriales sobre nuevas tecnologías.*  
URL: <http://www.adictosaltrabajo.com/index.php?pagina=home>

*The Struts Taglib*  
<http://struts.apache.org/1.x/struts-taglib/>

*Diseño de aplicaciones internet usando los Patrones de diseño J2EE*  
[www.moisesdaniel.com](http://www.moisesdaniel.com)

*J2EE* (diversitat de temes relacionats amb la plataforma i forums)  
[www.javahispano.org](http://www.javahispano.org)

Joel Armando Jil Garay (31/05/2010). “Comprendiendo Struts2 [Parte 1]”.  
*El Blog del Joel*  
<http://joeljil.wordpress.com/>

Raúl González Duque. *Etiquetas Struts2*  
<http://mundogeek.net/archivos/2009/02/08/struts-2/>

### **Versions de Programari:**

Servidor Web i contenidor de servlets i JSP Tomcat 6.0  
<http://tomcat.apache.org/download-60.cgi>

Versió de Java JDK  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Entorn de desenvolupament (IDE) Eclipse Ganymede  
<http://www.eclipse.org/downloads/>

SGBD Oracle 10g XE  
[http://download.oracle.com/docs/cd/B25329\\_01/doc/install.102/b25143/toc.htm](http://download.oracle.com/docs/cd/B25329_01/doc/install.102/b25143/toc.htm)

### **Lliberies del Projecte:**

El projecte requereix incloure aquestes lliberies a la carpeta WEB-INF/lib del Nostre projecte:

La versió 3.6.0 de la llibreria de Hibernate "*hibernate-distribution-3.6.0.final-dist.zip*"  
<http://sourceforge.net/projects/hibernate/files/hibernate3/>

La versió 2.1.8.1 d'Struts2 "*struts-2.1.8.1-all.zip*"  
<http://struts.apache.org/download.cgi>