

Treball de fi de carrera J2EE: Club d'escacs postal

José Antonio Seguro Castro
ETIS

Albert Grau Perise

25/6/2008

Dedicatòria

*Mentre els rius corrin cap al mar, facin ombra les muntanyes a les valls i hi hagi estels al cel, ha de durar la memòria del benefici rebut a l'home agraït, **Publio Virgilio, 70 AC-19 AC, poeta romà***

Es just i necessari que les primeres paraules d'agraïment vagin cap a la persona que m'aguanta cada dia. Gràcies Lucia, per estimar-me, per recolzar-me, i per haver insistit fins a l'extenuació per què em matriculés a l'UOC i així poder optar a finalitzar els meus estudis. Aquest projecte te'l dec a tu. Gràcies per ser la meva companya. T'estimo.

També es molt normal que me'n recordi d'aquells que ho han donat tot per mi. Gracias a mis padres, por dárme lo todo sin pedirme nada a cambio, y por todos los sacrificios que tuvisteis que hacer para sacarme adelante. Me enseñasteis muchas cosas pero sobre todo me enseñasteis a pensar. Os quiero.

Crec que també es de calaix que me'n recordi del meu "sherpa" en la pujada d'aquesta muntanya. Moltes gràcies Albert per tres raons. La primera per fer-me de guia, has posat a la meva disposició quantitat ingent d'informació que m'ha vingut molt bé, no he tingut gaires dubtes així que no t'he molestat gaire però tot i així has estat un consultor excepcional. La segona per les mostres d'ànim, això ho fas molt bé i es molt necessari, tot i que no ho sembli, així que si us plau no canviïs, els que vinguin darrera meu ho agrairan. I la tercera per la teva santa paciència amb un alumne que mai va entendre gaire bé que era allò de l'avaluació continuada.

Penso que es important fer una menció a dos persones excepcionals que em van ajudar i que, amb el pas dels anys, s'han convertit en part important de la meva vida. Gracias Raquel y Ana, por seguir ahí después de tanto tiempo.

A l'hora vull recordar-me'n del meu cosí José Manuel, i per extensió, dels seus dos germans Cristina i Jesús. Aunque no lo lleguéis a entender, el hecho de que este proyecto haya tomado forma os lo debo a vosotros. Durante su realización hubo un momento muy difícil en que me planteé dejarlo. Entonces San Google me mostró a Jose (más bien una cosa escrita por Jose) y me hizo darme cuenta de algo que se me había olvidado y que necesitaba que se me fuera recordado. Así que gracias por la parte que os toca.

Vull dedicar aquest projecte a la memòria i el record d'una persona molt especial per mi. Manuel Castro Veloso. Ojalá que una parte de tu espíritu se impregne en aquellos que vendrán detrás de ti.

Caure's es permès, aixecar-se es obligatori.
Proverbi rus.

Resum

*En dues paraules puc resumir quant he après sobre la vida: continua endavant. **Robert Lee Frost, 1874-1963, poeta americà***

Aquest treball de fi de carrera es basa en el desenvolupament d'una aplicació web per tal de que jugadors de tot el mon puguin jugar a escacs on-line.

L'aplicació funcionarà de manera molt simple, l'usuari es donarà d'alta, i podrà començar a jugar-hi contra els altres usuaris ja presents, participar en torneigs i fins i tot crear-ne de propis. Cal tenir en compte que el sistema de joc que s'utilitzarà serà el d'escac per correspondència, no el d'escac en temps real. L'usuari mourà, el seu moviment quedarà enregistrat, i, més tard, el seu rival es connectarà i mourà. Aquest sistema dona més temps a l'usuari per fer el seu moviment i permet que pugui dur moltes partides al mateix temps. El més normal es que un usuari d'aquest tipus de sistema dugui al voltant de trenta o quaranta partides a l'hora, cosa que el fa ideal per aprendre'n i millorar-ne el joc.

La idea bàsica es que l'aplicació sigui molt simple d'utilitzar i que l'usuari pugui fer-la servir de manera molt intuïtiva sense necessitar gaire temps per aprendre'n, després de tot els escacs sols són un joc. No ho hem d'oblidar.

Lògicament una aplicació com aquesta, tot i semblar molt simple a priori, es veu d'una hora lluny que es desproporcionada per un projecte de fi de carrera. Quan vaig començar a desenvolupar-la ho tenia molt clar. En cap moment ha estat la meva intenció acabar tots els casos d'ús presents, tot i que m'agradaria continuar treballant-hi després de la finalització del present projecte. La meva idea es dedicar-hi una mica de temps cada dia, i, amb una mica de sort, poder-la fer entrar en fase de explotació dintre d'un temps.

Índex

Índex.....	3
Índex d'imatges i taules.....	5
1 Introducció.....	6
1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.....	6
1.2 Objectius del TFC.....	6
1.3 Enfocament i mètode seguit.....	7
1.4 Planificació del projecte	7
1.5 Productes obtinguts.....	9
1.6 Breu descripció dels altres capítols de la memòria.....	9
2 Anàlisi de requeriments.....	10
2.1 Descripció més profunda del projecte.....	10
2.2 Actors del sistema.....	11
2.3 Casos d'ús	12
2.4 Interfície gràfica.....	19
2.4.1 Breu explicació textual de la interfície gràfica.....	19
2.4.2 Captures de pantalla de la interfície gràfica.....	19
2.5 Actors i casos d'ús de prioritat 1.....	22
2.5.1 Casos d'ús d'Usuari no registrat.....	22
2.5.2 Casos d'ús d'Usuari Novell.....	22
2.6 Descripció textual dels casos d'ús de prioritat 1.....	23
2.6.1 Acceptar partida oberta.....	23
2.6.2 Acceptar/Rebutjar taules.....	23
2.6.3 Anar a la següent partida.....	23
2.6.4 Llistar jugadors – Filtrar jugadors (antigament buscar jugadors).....	23
2.6.5 Llistar partides obertes / actives / finalitzades.....	24
2.6.6 Llistar partides (Buscar partides).....	24
2.6.7 Crear una partida oberta.....	24
2.6.8 Demanar suggeriment de rivals.....	24
2.6.9 Donar-se d'alta al sistema.....	24
2.6.10 Eliminar partida per jugador / eliminar partida oberta.....	25
2.6.11 Fer moviment	25
2.6.12 Identificar-se al sistema.....	25
2.6.13 Modificar dades personals.....	25
2.6.14 Navegar per la partida.....	25
2.6.15 Oferir taules.....	26
2.6.16 Reclamar partida.....	26
2.6.17 Rendir-se	26
2.6.18 Reptar a jugador.....	26
2.6.19 Saltar-se partida.....	27

2.6.20	Sortir.....	27
2.6.21	Veure partida.....	27
2.6.22	Veure partides actives.....	27
2.6.23	Veure partides finalitzades.....	27
2.6.24	Llistar partides actives.....	28
2.6.25	Llistar jugadors.....	28
2.6.26	Buscar usuari.....	28
2.7	Diagrama estàtic de l'anàlisi.....	29
2.8	Diagrames d'estat.....	30
2.8.1	Classe Jugador.....	30
2.8.2	Classe Partida.....	30
3	Disseny.....	31
3.1	Diagrama estàtic de Disseny.....	31
3.2	Diagrames de seqüència.....	32
3.2.1	Llistar partides + Veure partida + Navegar Partida.....	32
3.2.2	Alta Sistema + Buscar usuari + Identificar-se al sistema.....	33
3.2.3	Llistar jugadors + Reptar jugadors + Llistar-Filtrar jugadors.....	34
3.2.4	Veure part. actives + Llistar partides actives + Veure finalitzades.....	35
3.2.5	Crear partida oberta + Buscar partides obertes + Acceptar partida oberta.....	35
3.2.6	Oferir taules + Acceptar/Rebutjar taules.....	36
3.2.7	Fer moviment (també Rendir-se).....	36
3.3	Disseny de la persistència.....	37
3.4	Arquitectura del sistema.....	38
3.4.1	Patrons de disseny.....	39
3.4.1.1	El patró de disseny Model – View – Controller.....	39
3.4.1.2	El patró Composite view.....	40
3.4.1.3	El patró DAO.....	41
4	Implementació.....	42
4.1	Productes que s'han fet servir.....	42
4.2	Patrons.....	42
4.2.1	MVC, mitjançant Jakarta Struts.....	42
4.2.2	El patró composite view a l'implementació.....	43
4.2.3	El patró DAO: Hibernate.....	43
4.2.3.1	Asociacions not lazy.....	44
4.2.3.2	DataUtils: ¿Per qué una capa més entre la capa de negoci i Hibernate?.....	45
4.2.4	Llibreria opensource sobre el joc d'escacs: Internet Chess Toolkit.....	45
4.3	Estructura dels fonts.....	45
4.4	Estructura del WAR.....	47
4.5	Proves.....	47
5	Conclusions.....	48
6	Glossari.....	50
7	Bibliografia.....	52
7.1	Llibres.....	52

7.2 Pàgines web.....	52
8 Annex A: Sistema de puntuació.....	54
8.1 Sistema de puntuació durant les 30 primeres partides	54
8.2 Sistema de puntuació a partir de la partida 31	55
8.3 Consideracions sobre les puntuacions dins el sistema.....	57
9 Annex B: Deploy del projecte.....	59

Índex d'imatges

Imatge 1: Pantalla de benvinguda.....	20
Imatge 2: Pantalla d'introducció de dades.....	20
Imatge 3: Configuració del tauler.....	21
Imatge 4: Llistat de partides.....	21
Imatge 5: Casos d'us d'usuari no registrat	22
Imatge 6: Casos d'ús d'usuari novell.....	22
Imatge 7: Diagrama estàtic d'anàlisi	29
Imatge 8: Diagrama d'estat classe jugador.....	30
Imatge 9: Diagrama d'estats classe partida	30
Imatge 10: Diagrama de disseny.....	31
Imatge 11: Diagrama de seqüència: Llistar partides + Veure partida + Navegar partida ...	32
Imatge 12: Diagrama seq: Alta sistema + buscar usuari + identificar-se al sistema.....	33
Imatge 13: Diagrama seq: Llistar jugadors + Reptar jugadors + Llistar-Filtrar jugadors	34
Imatge 14: Diag seq: Veure partides actives + Llistar partides actives + Veure finalitzades	35
Imatge 15: Diag seq: Crear partida oberta + Buscar p. obertes + Acceptar partida oberta..	35
Imatge 16: Diagrama de seqüència: Oferir taules + Acceptar / Rebutjar taules.....	36
Imatge 17: Diagrama de seqüència: Fer moviment - Rendir-se.....	36
Imatge 18: Disseny de la persistència.....	37
Imatge 19: El patró MVC.....	39
Imatge 20: El patró composite view	40
Imatge 21: El patró DAO.....	41
Imatge 22: Diagrama de seqüència patró DAO	41
Imatge 23: Llibreries utilitzades	60

Índex de taules

Taula 1: Planificació.....	8
Taula 2: Casos d'us.....	18
Taula 3: Taula de puntuació.....	54

1 Introducció

Començar bé no es poc, però tampoc es molt, Sòcrates, 470 AC-399 AC, filòsof grec

1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.

El present projecte de final de carrera té com a objectiu la creació d'un portal web que permeti jugar a escacs per correspondència.

No es un tema nou, existeixen varies aplicacions d'aquest estil a la xarxa, però a totes els hi falta alguna funcionalitat important, i els usuaris sempre es troben a disgust amb algunes coses. L'esperit que rau darrera de la realització d'aquest sistema es agafar tot el bo de cadascun d'aquests sistemes ja consolidats per crear un de nou que realment convenci als usuaris, i que sigui totalment gratuït per tal de que la gent, que com jo, estima els escacs tingui un lloc de reunió on poder desenvolupar les seves partides.

Aquest es un tema que m'apassiona, des de que era molt petit. Per mi es una oportunitat i un desafiament important poder barrejar sota un mateix projecte les meves dos grans passions: els escacs i la programació.

A part de jugar -hi, el sistema permetrà als usuaris anar guanyant punts i experiència. Amb el pas del temps, les partides finalitzades els permetran accedir-hi a més funcionalitats que en principi no posseïen, com la capacitat de crear torneigs. En finalitzar l'aplicació existiran quatre tipus d'usuaris: L'usuari normal, l'experimentat, l'expert i l'administrador, que disposaran de diferents funcionalitats.

1.2 Objectius del TFC

Els objectius principals del present projecte son:

1. Permetre als usuaris disputar les seves partides de manera fàcil i intuïtiva
2. Fer que els jugadors tinguin coneixement dels altres jugadors informant-los amb estadístiques dels punts forts i febles dels seus rivals
3. Permetre crear fòrums on els jugadors puguin compartir la seva experiència amb els escacs i fer totes les preguntes que vulguin a altres jugadors, així com contestar les qüestions formulades per altres companys.
4. Incentivar la competitivitat amb un sistema de punts que informi al jugador de quina es la seva força relativa respecte als altres jugadors, així com otorgant medalles en guanyar campionats.

Cal tenir en compte que molts d'aquests objectius, com es veurà més endavant no han estat possibles de aconseguir, degut a l'enorme quantitat de feina que aquest projecte sencer demanava, feina que, de totes, totes, superava el temps que hom normal, pot dedicar a un projecte de fi de carrera.

1.3 Enfocament i mètode seguit

S'han seguit, en el desenvolupament del present projecte, una sèrie de passos destinats a distribuir la feina eficientment en el temps.

Primer s'ha fet una planificació del mateix destinada a guiar cadascun dels passos de la realització del treball.

Un cop fet això s'ha començat a fer l'anàlisi de requeriments per tal de poder fer-se una idea de les funcionalitats necessàries d'implementar. Més tard s'ha passat a realitzar el disseny per tal de clarificar les coses de cara a la següent fase.

Ja per acabar s'ha passat a dur a terme la implementació final del producte, aquesta fase es la que s'ha endut més temps, com en un principi era d'esperar.

Per últim s'ha fet l'últim joc de proves destinat a crear una base de dades suficientment gran com perquè el sistema sigui provat a fons pel tribunal.

1.4 Planificació del projecte

*Planejar: preocupar-se per trobar el millor mètode per obtenir un resultat accidental, **Ambrose Bierce, 1842-1914, escriptor americà***

Com tota bona planificació aquesta es sempre una utopia. En el mon real l'home proposa, i el destí disposa. Penso que en aquest punt de la memòria caldria plasmar la planificació original del projecte, i explicar per què aquesta no s'ha pogut dur a terme com estava planejat.

La idea original era fer el projecte de manera iterativa e incremental segons la prioritat. Com es veurà més endavant l'anàlisi de requisits va donar una quantitat immensa de casos d'ús, aquests es van dividir per prioritats, de l'1 al 5. En un principi primer s'escolliran els casos d'ús de prioritat 1 i es començaria a treballar amb ells, un cop els casos de prioritat 1 estiguessin especificats, dissenyats, implementats, i provats es passarà als casos d'ús de prioritat 2, i així. La idea era assegurar-se de que cada filera de totxanes està ben col·locada abans de començar la següent.

Així doncs la planificació original va quedar així:

TASCA	Dies durada	Dates
<i>Prioritat 1</i>		
Especificació		
Especificació diagrama de classes	1	15/3
Especificació de casos d'ús	11	16/3 – 26/3
Disseny		
Disseny de casos d'ús	7	27/3 – 2/4
Revisió diagrama estàtic de Disseny	1	3/4
Disseny de la persistència	3	4/4 – 6/4
Disseny/Implementació de la interfície d'usuari	5	7/4 – 11/4
Implementació		
Implementació i proves	16	15/4 – 30/4
Ajuda		
Creació de l'ajuda a pantalla	2	1/5 – 2/5
<i>Prioritat 2</i>		
Especificació		
Modificació diagrama de classes	1	3/5
Especificació de casos d'ús	8	4/5 – 11/5
Disseny		
Disseny de casos d'ús	4	12/5 – 15/5
Revisió diagrama estàtic de Disseny	1	16/5
Disseny de la persistència	3	20/5 – 22/5
Disseny/Implementació de la interfície d'usuari	5	12/5 – 16/5
Implementació		
Implementació i proves	15	16/5 – 30/5
Ajuda		
Creació de l'ajuda a pantalla	2	31/5 – 1/6
<i>Temps reserva</i>		
Temps de reserva	19	2/6 – 20/6
<i>Traducció</i>		
Traducció al castellà i si dona temps a l'anglès	5	16/6 – 20/6
<i>Memòries diverses</i>		
Creació documentació PAC2	3	12/4 – 14/4
Creació documentació PAC3	3	17/5 – 19/5
Creació memòria final	2	21/6 – 22/6
Creació presentació final	2	23/6 – 24/6

Taula 1: Planificació

Com es veu ja s'havia considerat el fet de que no es podrien dur a terme tots els casos d'ús del sistema. Ja que sols s'hi dedicava temps per les prioritats 1 i 2, mentre, com he dit abans, n'hi havia 3,4 i 5. Sigui com sigui aquesta planificació, per motius personals, de feina,

i familiars, no s'ha pogut dur a terme d'aquesta manera. El resultat es que no estan acabats els casos d'ús de nivell 1, falta calcular la puntuació, per exemple. A la partida no està contemplada la possibilitat d'enrocar ni la promoció per una cosa diferent a dama. La traducció està feta però no hi hagut temps de implementar-la adequadament al sistema, i l'ajuda en pantalla tampoc s'ha pogut fer.

1.5 Productes obtinguts

Fonamentalment s'han obtingut tres productes de la realització d'aquest projecte:

1. L'aplicació web en si, el leitmotiv del projecte, es lliura junt amb aquest document per tal de que pugui instal·lar-se en un contenidor de servlets, com pot ser Tomcat.
2. La memòria que esteu llegint ara mateix, document on s'expliquen les diverses fases de la feina feta
3. La presentació, que resumeix la memòria mostrant i resumint els punts més importants d'aquesta.

1.6 Breu descripció dels altres capítols de la memòria.

En els següents apartats es podrà trobar el següent:

- Anàlisi de requeriments, es un apartat en el qual hom podrà veure les diverses funcionalitats del projecte en si, així com una explicació no formal de cadascuna de elles
- Disseny, un cop sabem que farem, en aquest apartat s'explicarà com ho farem, es mostraran com seran les dades i diverses imatges i gràfics destinats a fer mes entenedora l'explicació
- Implementació, aquest serà un punt en que es tractarà com s'ha implementat l'aplicació, quines tecnologies hem utilitzat així com quina es l'estructura d'aquesta.
- Conclusions, part dedicada íntegrament a valorar la realització del projecte, i que m'ha aportat individualment
- Glossari, un parell de pàgines amb mots de difícil comprensió per qui no tingui familiaritat amb els escacs.
- Bibliografia, una llista tant de llibres com de pàgines webs fetes servir durant la realització del projecte,
- Annexos, un annex sobre el sistema de puntuació i un altre sobre com fer la instal·lació de la aplicació.

2 Anàlisi de requeriments

2.1 Descripció més profunda del projecte

L'objectiu del present projecte es la realització d'una aplicació web que simuli un club d'escacs per correspondència. La principal funcionalitat serà permetre que els usuaris puguin jugar escacs on-line, guardant totes les partides jugades i que aquestes siguin accessibles per qualsevol usuari.

Primer l'usuari s'haurà de donar d'alta al sistema omplint un formulari. Un cop estigui donat d'alta podrà entrar i començar a jugar amb altres usuaris que hi siguin al sistema.

El sistema no sols permetrà jugar partides entre jugadors si no que també permetrà que el jugador s'hi inscrigui a torneigs, i, fins i tot que els creï (El sistema també en crearà periòdicament). L'usuari podrà veure estadístiques dels altres usuaris del sistema, podrà accedir a un llibre d'apertures, podrà inscriure's a un equip i fins i tot crear-ne un, així com jugar partides i torneigs en equip. Al mateix temps que juga també podrà comunicar-se amb els contrincants deixant-los missatges. I participar en fòrums oberts i crear-ne de propis.

El tipus de escacs que permetrà el sistema serà el de partides per correspondència. Aquest tipus de partides es diferencien de les partides normals en que el jugador té més temps per fer els moviments, cosa que fa que els usuaris no hagin d'estar al mateix temps on-line per tal de jugar-hi una partida. A més, a més, les partides per correspondència sempre han tingut unes normes diferents a les partides d'escacs presencials. Entre aquestes destaca el fet que el jugador pot accedir a llibres d'apertures i a documentació sobre altres partides que haguessin jugat altres jugadors. Això continua ben present a l'escac per e-mail o per servidor (com aquest cas). És per això que el sistema permet l'accés a altres partides i a un llibre d'apertures.

Sigui com sigui, molts usuaris gaudeixen competint sense esperes, per tant el sistema informarà visualment a l'usuari que competidors es troben on-line en cada moment. I tot i que els controls de temps seran el de l'escac per correspondència, res impedirà que dos rivals que es vegin on-line decideixin accelerar la seva partida, tot i que cap dels dos tindrà l'obligació de finalitzar-la en aquell moment.

Cada usuari tindrà una puntuació que reflectirà la fortalesa del seu joc. El sistema concret de puntuació serà el [sistema Elo](#) a partir de la partida 31 de l'usuari. Durant les 30 primeres partides s'aplicarà una modificació del sistema Elo basada en una altra fórmula i una taula de conversió, que es troben a l'Annex A del present document, junt amb les fórmules utilitzades per calcular la puntuació Elo. Aquesta taula i fórmula, es poden també trobar a [l'Associació Espanyola d'escacs per correspondència](#) (AEAC), tot i que el sistema Elo descrit a aquest lloc es diferencia una mica del que farem servir al nostre sistema.

Una altra consideració a tenir en compte es que les partides que no formin part d'un torneig podran ser amb puntuació o sense, segons acordin els jugadors.

2.2 Actors del sistema

En principi els actors principals del sistema seran jugadors, de totes maneres, però, no tots els jugadors tindran les mateixes funcionalitats, aquestes funcionalitats dependran de les partides que el jugador hagi completat. Per tant ens trobarem amb aquests actors:

- **Usuari no registrat** Es un usuari que encara no s'ha registrat al sistema. Pot veure partides que s'hagin jugat o s'estiguin jugant, així com accedir al llibre d'apertures.
- **Usuari novell:** Es un usuari que ha jugat 30 o menys partides amb puntuació. Aquest usuari no pot crear torneigs ni inscriure's a cap equip. Les partides que jugui aquest tipus d'usuari contra usuaris que no siguin novells no suposaran pèrdua de punts pels seus rivals, en cas de taules o victòria de l'usuari novell.
- **Usuari experimentat** Es un usuari que ha jugat 31 o més partides amb punts, fins a un màxim de 100. Aquest usuari pot inscriure's a equips, però no pot crear torneigs, ni crear equips, ni ser escollit capità d'equip.
- **Usuari expert** Es aquell usuari que hagi jugat 101 o més partides amb punts, no tindrà cap mena de limitació.
- **Capità:** Capità d'un equip, ha de ser un usuari expert i les seves tasques tindran que veure amb la inscripció d'equips a torneigs i la seva administració.
- **Administrador:** La funció principal de l'administrador serà la de configurar el sistema per tal de que hi hagi eliminacions automàtiques (de jugadors no actius, de partides molt antigues, etc.). També podrà fer perdre a un jugador una partida si rep queixes sobre ell, per exemple perquè insulta a un rival, i fins i tot podrà prohibir l'accés a un jugador si la seva actitud es negativa. Un Administrador tindrà el rang d'usuari expert, es a dir que a més de les opcions d'administrador, tindrà les opcions de menú d'un jugador expert i podrà ser capità també si vol.

2.3 Casos d'ús

Aquesta es la llista de casos d'ús del sistema, com es pot veure n'hi ha bastants, més de 70, segurament, i valorant la meua experiència, la majoria no podran ser portats a terme durant la realització del present projecte. Cal tenir en compte que molts són molt simples, tot i així, tot i que estic molt animat i motivat per fer una web funcional sobre un tema que m'apassiona, me n'adono que el número es òbviament excessiu per un projecte de fi de carrera, per tant s'han dividit els casos d'us per prioritats. Aquestes venen indicades per un número de l'1 al 5, sent 1 els casos d'ús de prioritat màxima i 5 els de prioritat més baixa. Gaire bé segur, per no dir segur, que els casos d'us de prioritat 4 o 5 no seran implementats, ni, potser, especificats, els de 3, dubtosament i els de 2, sent optimistes. Com es pot veure:

- Tot el que està relacionat amb les partides i amb la identificació de l'usuari té una prioritat màxima.
- Segueixen els torneigs i l'accés al llibre d'apertures.
- Després me n'ocuparé dels missatges, de les tasques pròpies de l'administració i de la cerca per posicions.
- Després, si dona temps, els fòrums i els moviments condicionals
- I per finalitzar els equips.

Cas d'ús (entre parèntesis l'actor que en fa servei)	Pr.	Descripció
1. Acceptar partida oberta (Usuari novell o més)	1	Accepta la partida oberta convertint-se en el rival buscat.
2. Acceptar/Rebutjar taules (Usuari novell o més)	1	Accepta o rebutja un oferiment de taules
3. Anar a la següent partida (Usuari novell o més)	1	Del llistat de partides en que sigui el torn de l'usuari, el sistema mostrarà aquella en que a l'usuari li quedi menys temps per fer el seu moviment.
4. Crear una partida oberta (Usuari novell o més)	1	Crea una partida nova sense contrincant per tal de que algun altre usuari l'accepti, podrà tenir una posició inicial.
5. Demanar suggeriment de rivals (Usuari novell o més)	1	Llista un conjunt de jugadors que per la seva experiència i puntuació son adients per enfrontar-se a nosaltres.
6. Donar-se d'alta al sistema (Usuari no registrat)	1	A partir d'unes dades personals de l'usuari, el dona d'alta al sistema
7. Eliminar partida oberta (Usuari novell o més)	1	Elimina una partida oberta creada pel mateix jugador

Cas d'ús (entre parèntesis l'actor que en fa servei)	Pr.	Descripció
8. Eliminar partida per jugador (Usuari novell o més)	1	Abans de que les blaques hagin fet el seu tercer moviment, l'usuari reptat o reptador (si es tracta d'un repte) o bé l'usuari que havia creat la partida oberta (si es tracta d'una partida oberta) podran eliminar la partida i aquesta serà com si no hagués existit.
9. Fer moviment (Usuari novell o més)	1	El sistema comprovarà que el moviment sigui correcte i l'enregistrarà. Si es escac i mat, donarà la partida al jugador. Si aquest escac i mat significa la victòria/finalització d'un torneig, el sistema donarà el premis.
10. Identificar-se al sistema (Usuari no registrat)	1	A partir del nom de l'usuari i la seva contrasenya, valida aquest al sistema
11. Llistar jugadors - Filtrar jugadors (Usuari novell o més)	1	Llista els diferents jugadors que formin part del sistema permetent anar filtrant-los fent-hi click, per exemple, fent click a la ciutat del jugador, obtindrem els jugadors de la mateixa ciutat.
12. Llistar partides actives d'altres jugadors (Tots)	1	Llista les partides que actualment s'estan disputant entre altres jugadors
13. Llistar partides finalitzades d'altres jugadors (Tots)	1	Permet llistar les partides finalitzades d'altres jugadors. Estarà format per un conjunt de casos d'ús que aniran des de llistar les guanyades, a llistar-ne sols les perdudes amb blanques, per exemple
14. Llistar partides obertes (Usuari novell o més)	1	Llista les partides obertes que s'hagin creat al sistema.
15. Modificar dades personals (Usuari novell o més)	1	Permet a l'usuari modificar les seves dades. Estarà format per tres subcasos d'ús. L'usuari podrà modificar les dades personals, les relacionades amb el tauler, així com canviar la contrasenya.
16. Navegar per la partida (Tots)	1	Permet moure's pels moviments de la partida i anar a una posició concreta d'aquesta, quant hom estigui a una posició el sistema facilitarà la descripció textual d'aquesta amb notació FEN.
17. Oferir taules (Usuari novell o més)	1	Ofereix taules al contrari

Cas d'ús (entre parèntesis l'actor que en fa servei)	Pr.	Descripció
18. Reclamar partida (Usuari novell o més)	1	Si el temps del contrincant s'ha esgotat, ens dona la victòria.
19. Reclamar taules (Usuari novell o més)	1	En cas de repetició de moviments, per exemple, hom pot reclamar taules.
20. Rendir-se (Usuari novell o més)	1	Dona la partida per finalitzada, otorgant la victòria al contrincant
21. Reptar a jugador (Usuari novell o més)	1	Crea una partida nova tenint com a contrincant el jugador escollit, pot tenir una posició inicial.
22. Saltar-se partida (Usuari novell o més)	1	Fer que la partida mostrada es col·loqui a la cua de les partides independentment del temps que li quedi a l'usuari per moure.
23. Sortir (Usuari novell o més)	1	Farà el logout de l'usuari
24. Veure partida (Tots)	1	Accedeix a una partida presentant el tauler
25. Veure partides actives (Usuari novell o més)	1	Presenta una llista amb les partides que està jugant l'usuari, per tal de que aquest n'esculli una, si vol.
26. Veure partides finalitzades (Usuari novell o més)	1	Presenta una llista amb les partides que ha finalitzat l'usuari, per tal de que aquest pugui veure la que vulgui.
27. Accedir a llibre d'apertures (Tots)	2	Presenta el tauler i un llistat amb els primers moviments de les diferents apertures.
28. Afegir usuari a la llista d'ignorats (Usuari novell o més)	2	Segons hom vagi jugant es trobarà amb usuaris que, pel que sigui, no respecten les normes d'una partida entre cavallers, que es el que hauria de ser una partida d'escacs. Si l'usuari vol podrà afegir a aquest usuari a la llista d'ignorats de manera que l'usuari afegit no podrà reptar-nos ni acceptar cap partida oberta per nosaltres. Si però ens el podem trobar, eventualment, a un torneig.

Cas d'ús (entre parèntesis l'actor que en fa servei)	Pr.	Descripció
29. Baixar PGN de partida (Usuari novell o més)	2	El PGN es un tipus de fitxer de text amb totes les dades d'una partida. Molts jugadors d'escacs on-line l'utilitzen per tal d'enregistrar les seves partides i les d'altres a bases de dades pròpies. Un jugador podrà baixar qualsevol partida, no sols les seves, però s'implementarà algun sistema de crèdits per tal de que no hi hagi abusos que col·lapsin el servidor.
30. Buscar torneig (Usuari novell o més)	2	A partir d'uns paràmetres de cerca, el sistema buscarà els torneigs que compleixin els requisits, informant a l'usuari de a quins pot inscriure's i a quins no.
31. Configurar auto-esborrat (Administrador)	2	Permetrà a l'administrador configurar com s'ha de comportar el sistema en els esborrats automàtics
32. Crear torneig (Usuari expert)	2	Es permetrà a l'usuari la creació d'un torneig.
33. Esborrar-se de torneig (Usuari novell o més)	2	Un usuari inscrit a un torneig s'esborrarà d'aquest.
34. Inscriure's a torneig (Usuari novell o més)	2	L'usuari s'inscriurà a un torneig.
35. Navegar pel llibre d'apertures (Tots)	2	Permet navegar pels diferents moviments i veure'n les diferents variants.
36. Programar vacances (Usuari novell o més)	2	L'usuari podrà escollir els dies de vacances que voldrà tenir, i modificar-ne programacions anteriors que hagi fet, de dies futurs, obviament. No podrà canviar de dia vacances ja gaudides.
37. Treure usuari de la llista d'ignorats (Usuari novell o més)	2	Treure un usuari de la llista d'ignorats, a partir de llavors aquest podrà reptar-nos i acceptar partides obertes per nosaltres.
38. Veure dades de jugador (Usuari novell o més)	2	Mostra una pàgina amb les dades públiques del jugador, incloent-hi estadístiques sobre les seves partides i els seus punts, així com les medalles aconseguides als diferents torneigs.
39. Veure llista d'ignorats (Usuari novell o més)	2	Mostra la llista amb tots els usuaris que hem ignorat, permetent treure'n de la llista als usuaris que vulguem.

Cas d'ús (entre parèntesis l'actor que en fa servei)	Pr.	Descripció
40. Veure torneig (Usuari novell o més)	2	El sistema mostrarà el torneig, si aquest no ha començat es mostraran els usuaris inscrits, si ha començat es mostrarà la classificació / creuaments del torneig.
41. Veure vacances (Usuari novell o més)	2	Veure el temps de vacances que l'usuari ha esgotat i el que li queda, així com el calendari amb les vacances programades.
42. Buscar posició a llibre d'apertures (Usuari novell o més)	3	A partir d'una posició a una partida permet buscar aquesta al llibre d'apertures, per tal de veure quines son les continuacions més normals.
43. Buscar posició a partides jugades (Usuari novell o més)	3	Igualment però en lloc de ser al llibre d'apertures, la cerca es fa a totes les partides jugades fins llavors, per veure com van jugar altres jugadors a aquesta posició.
44. Decidir resultat de partida (Administrador)	3	Permetrà a l'administrador decidir el resultat d'una partida de manera justificada.
45. Enviar missatge a partida (Usuari novell o més)	3	Durant el transcurs d'una partida els usuaris podran intercanviar-se missatges que es mostraran al costat del tauler.
46. Enviar missatge a usuari / Respondre a missatge (Usuari novell o més)	3	En qualsevol moment un usuari podrà enviar un missatge a un altre usuari, inclòs l'administrador.
47. Esborrar missatge d'usuari (Administrador, Usuari novell o més)	3	L'usuari podrà esborrar un missatge que l'hagin enviat, també ho podrà fer l'administrador.
48. Esborrar missatge de partida (Administrador)	3	Si l'administrador considera que un missatge de partida es inadequat, previ avís de un usuari, podrà esborrar-lo
49. Expulsar a jugador de torneig (Administrador)	3	Permetrà a l'administrador expulsar a un jugador del torneig si rep justificades queixes sobre ell
50. Expulsar a jugador del sistema (Administrador)	3	També permetrà expulsar a un jugador del sistema per un temps si rep moltes queixes o el jugador la fa molt grossa.
51. Llegir missatge (Usuari novell o més)	3	Mostrarà el contingut d'un missatge
52. Veure llista de missatges rebuts (Usuari novell o més)	3	Un usuari podrà veure la llista de missatges rebuts

Cas d'ús (entre parèntesis l'actor que en fa servei)	Pr.	Descripció
53. Crear fòrum (Administrador)	4	L'administrador tindrà la potestat de crear els diversos fòrums de participació que tindrà el sistema.
54. Crear/Modificar moviment condicional (Usuari experimentat o més)	4	Tractant-se d'escac postal, en que els usuaris no han de estar forçosament on-line al mateix temps, el sistema permetrà al jugador avançar-se a un possible moviment del seu contrincant i crear una / varies clàusules <i>si-llavors</i> amb un possible moviment del seu contrincant i una resposta seva. Els moviments condicionals podran ser anidats, es a dir el jugador podrà avançar-se a un conjunt de moviments del seu contrincant. No es permetrà introduir un moviment condicional que signifiqui fer escac i mat, degut a que cal donar al rival l'oportunitat d'abandonar.
55. Enviar missatge a fòrum (Usuari novell o més)	4	L'usuari podrà participar al fòrum escrivint un missatge.
56. Esborrar fòrum (Administrador)	4	També podrà esborrar-los quan li convingui.
57. Esborrar missatge de fòrum (Administrador, Usuari novell o més)	4	Tant el creador del missatge com l'administrador podran esborrar un missatge del fòrum, el creador el podrà esborrar sempre que no hi hagin respostes.
58. Esborrar moviment condicional (Usuari experimentat o més)	4	Esborra un moviment condicional de la llista.
59. Llistar fòrums (Usuari novell o més)	4	El sistema tornarà un llistat amb tots el fòrums
60. Llistar moviments condicionals¹ (Usuari experimentat o més)	4	Mostra per pantalla la llista de moviments condicionals que te creats un jugador.
61. Modificar fòrum (Administrador)	4	Així com modificar les dades del fòrum.
62. Modificar missatge de fòrum (Administrador, Usuari novell o més)	4	Igual que a l'anterior però modificant el missatge

¹Cal tenir clar que els moviments condicionals estaran associats a un moviment concret. Es a dir si un contrincant fa un moviment no reflexat a la llista de moviments condicionals, la llista sencera serà esborrada, perquè ja no tindria sentit. Igualment si un contrincant fa un moviment que el fa entrar a un segon nivell de moviments condicionals, tot el primer nivell serà eliminat per la mateixa raó.

Cas d'ús (entre parèntesis l'actor que en fa servei)	Pr.	Descripció
63. Veure fòrum (Usuari novell o més)	4	El sistema mostrarà a l'usuari el contingut d'un fòrum
64. Veure missatge de fòrum (Usuari novell o més)	4	L'usuari podrà veure el contingut complet d'un missatge d'un fòrum
65. Acceptar invitació entrar a equip (Usuari experimentat o més)	5	Acceptar una invitació, equivalent a inscriure's a equip.
66. Convidar a entrar a equip (Capità)	5	Enviar una invitació a un jugador per pertànyer al nostre equip. L'envia el capità.
67. Crear equip (Usuari expert o més)	5	Permetre a l'usuari la creació d'un equip, l'usuari passarà a ser el capità.
68. Esborrar equip de torneig (Capità)	5	El capità podrà esborrar a l'equip d'un torneig per equips.
69. Esborrar-se d'equip (Usuari experimentat o més)	5	Permet a l'usuari esborrar-se d'un equip. No es pot fer a meitat d'un torneig on estigui jugant aquest equip.
70. Escollir capità d'equip (Capità)	5	Assignar a un usuari el rol de capità. Sols ho pot fer l'actual capità, que deixarà de ser-ho.
71. Expulsar a equip de torneig (Administrador)	5	L'administrador podrà expulsar a un equip d'un torneig si ho considera necessari.
72. Inscriure equip a torneig (Capità)	5	El capità podrà inscriure l'equip a un torneig confeccionant una alineació dels jugadors que hi participaran.
73. Inscriure's a equip (Usuari experimentat o més)	5	Permet a un usuari inscriure's a un equip. Un usuari sols podrà està inscrit a un equip.
74. Modificar alineació d'equip (Capità)	5	El capità podrà modificar l'alineació del seu equip abans però que el torneig hagi començat.

Taula 2: Casos d'us

2.4 Interfície gràfica

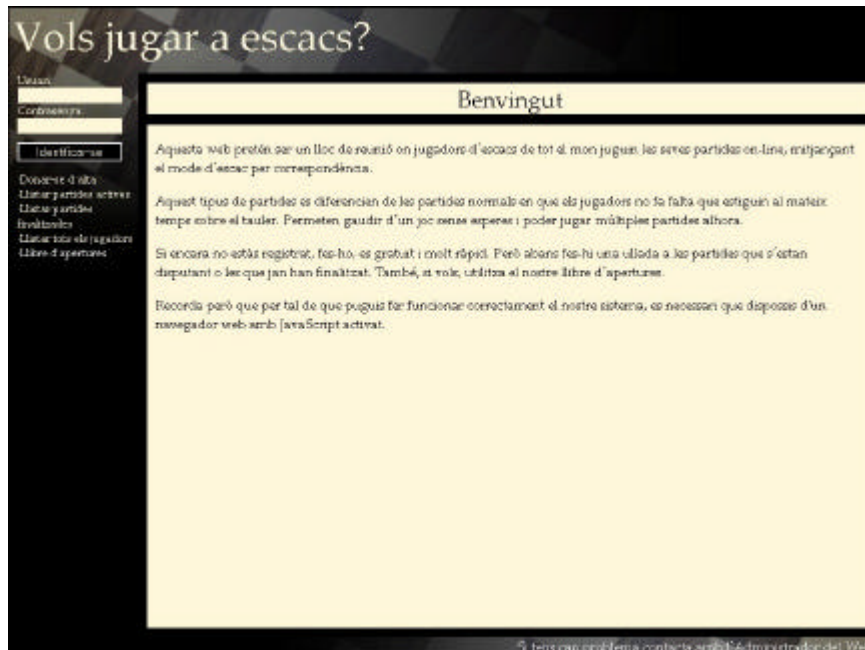
2.4.1 Breu explicació textual de la interfície gràfica

La interfície gràfica estarà dividida en cinc seccions, una de les quals, la dreta, solament sortirà quan s'estigui jugant o veient una partida. S'ha dissenyat per una resolució de 1024x768, de manera que si la pantalla es mes petita sortirà una barra de desplaçament horitzontal, sigui com sigui, els millors resultats s'obtenen amb una resolució de 1280.

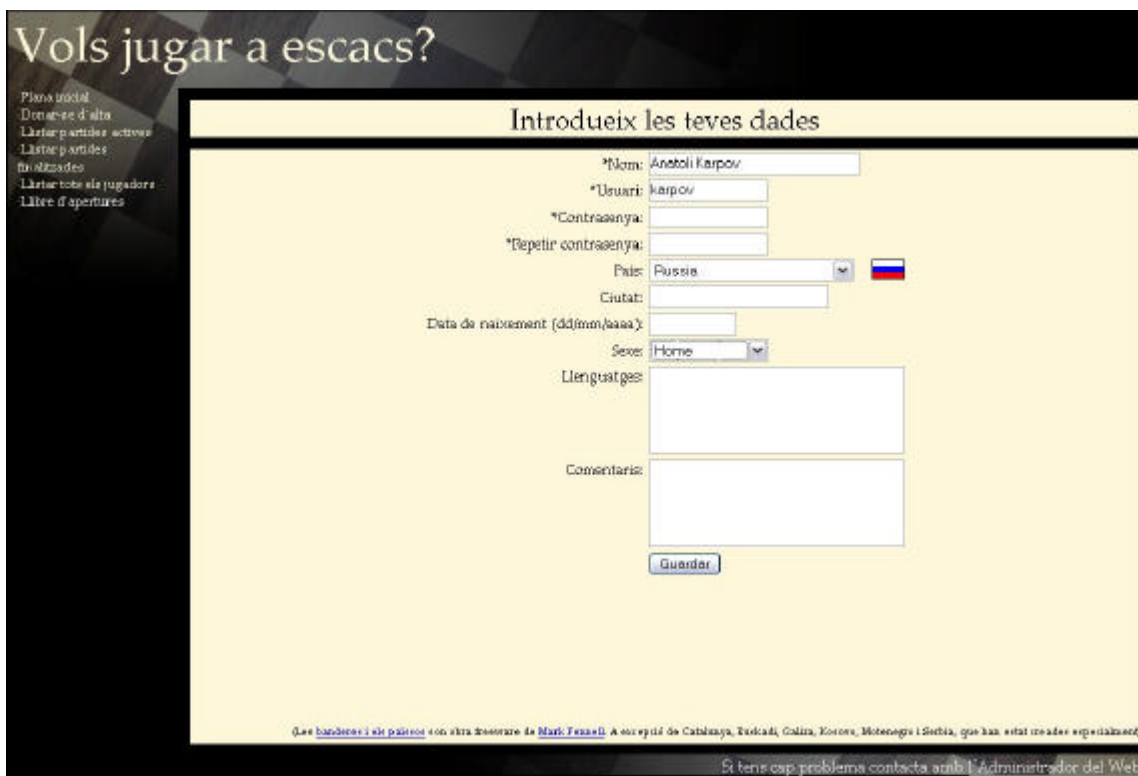
Capcelera amb alguna imatge propia sobre els Escacs		
Formulari per la identificació i/o Menu Javascript amb les opcions que te l'usuari	<p>Quan s'estigui jugant o veient una partida aquí sortirà el tauler, i el jugador podrà fer els moviments des de aquí.</p> <p>Quan no s'estigui jugant o veient una partida aquest espai serà usat per presentar la informació, tant llistes, com classificació dels torneigs, informacions sobre els jugadors, etc...</p> <p>Quan l'usuari no s'hagi identificat al sistema aquí sortiran uns mini-taulers amb els últims moviments fets pels usuaris del sistema, de manera que, sense identificar-se, un usuari tingui accés per veure aquestes partides, fent simplement click sobre el mini-tauler que vulgui.</p>	<p>Quan s'estigui jugant o veient una partida en aquest costat sortiran els moviments fets i altres aspectes relacionats amb la partida, serà com la planilla d'una partida d'escacs real. Permetrà la navegació per la partida fent click a cadascun dels moviments</p> <p>Quan no s'hi jugui o s'hi vegi una partida aquest espai desapareixerà.</p>
Peu		

2.4.2 Captures de pantalla de la interfície gràfica

Tot seguit es poden veure algunes captures de pantalla de la interfície gràfica per que el lector es pugui fer una idea:



Imatge 1: Pantalla de benvinguda



Imatge 2: Pantalla d'introducció de dades



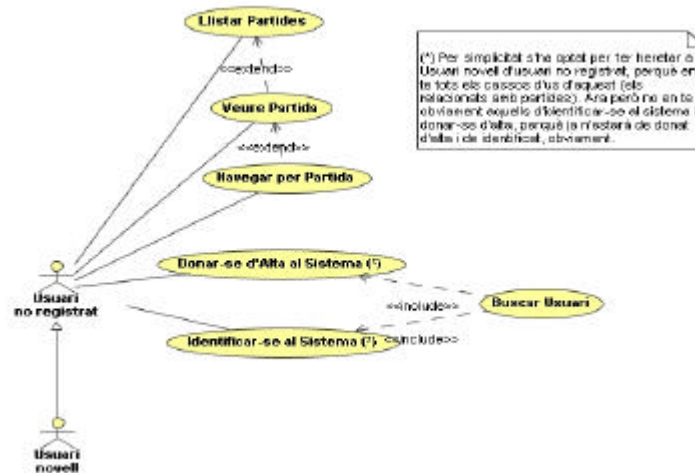
Imatge 3: Configuració del tauler



Imatge 4: Llistat de partides

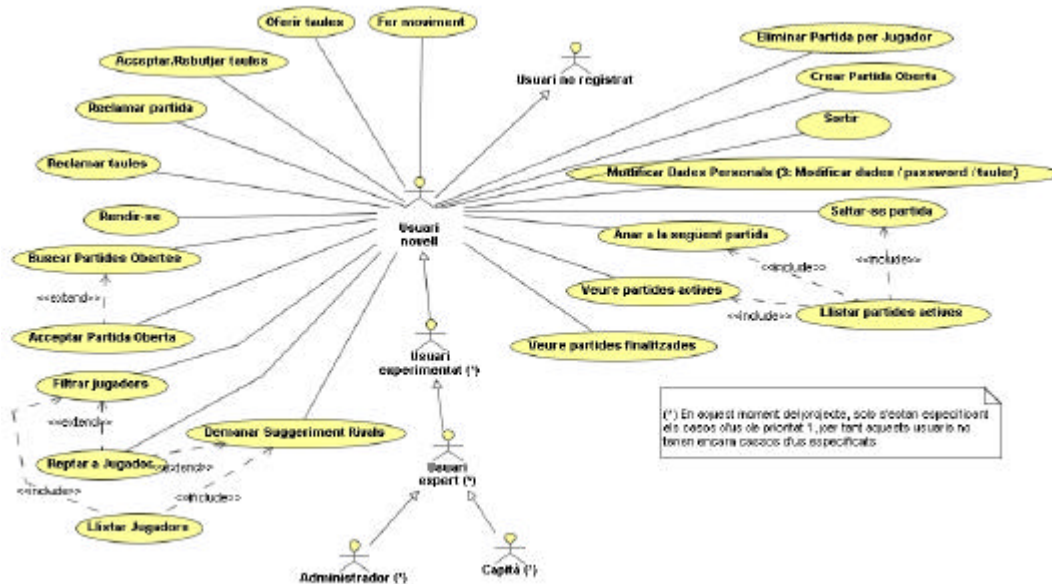
2.5 Actors i casos d'ús de prioritat 1

2.5.1 Casos d'ús d'Usuari no registrat



Imatge 5: Casos d'us d'usuari no registrat

2.5.2 Casos d'ús d'Usuari Novell



Imatge 6: Casos d'ús d'usuari novell

2.6 Descripció textual dels casos d'ús de prioritat 1

2.6.1 Acceptar partida oberta

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: A partir de la llista de partides obertes, l'usuari podrà escollir-ne una per començar a jugar-hi.	
Precondició: La partida oberta existeix a la base de dades i no està creada pel mateix usuari.	
Postcondició: La partida oberta ha deixat de ser una partida oberta per convertir-se en una partida normal on l'usuari es un dels rivals.	

2.6.2 Acceptar/Rebutjar taules

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: Enregistra l'acció de l'usuari respecte a un oferiment de taules per part del rival de la partida activa en aquell moment.	
Precondició: Existeix un oferiment de taules per part del rival de l'usuari a la partida activa.	
Postcondició: La resposta es enregistra a la base de dades. Si ha estat afirmativa la partida estarà marcada com finalitzada	

2.6.3 Anar a la següent partida

Actor(s): Usuari novell o més	Casos d'ús relacionats: Llista partides actives
Descripció: Passa a la següent partida de l'usuari, basat en el temps de resposta de queda, primer les partides que necessiten atenció més urgent perquè queda menys temps de resposta.	
Precondició: L'usuari té almenys una partida que espera resposta.	
Postcondició: Cap	

2.6.4 Llistar jugadors – Filtrar jugadors (antigament buscar jugadors)

Actor(s): Usuari novell o més	Casos d'ús relacionats: Reptar jugador, Llistar jugadors
Descripció: L'usuari pot llistar i filtrar els jugadors	
Precondició: Cap	
Postcondició: Cap	

2.6.5 Llistar partides obertes / actives / finalitzades

Actor(s): Usuari novell o més	Casos d'ús relacionats: Acceptar partida oberta
Descripció: Mostra totes les partides obertes que esperen un rival i a les quals l'usuari s'hi pot inscriure. o actives o finalitzades,	
Precondició: Cap	
Postcondició: Cap	

2.6.6 Llistar partides (Buscar partides)

Actor(s): Usuari no registrat o més	Casos d'ús relacionats: Veure partida
Descripció: Semblant a buscar jugadors però amb partides.	
Precondició: Cap	
Postcondició: Cap	

2.6.7 Crear una partida oberta

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: L'usuari crea una partida oberta que s'afegeix a la llista de partides obertes del sistema, amb una sèrie de paràmetres, posició inicial, puntuació del rival buscat, etc...	
Precondició: Cap	
Postcondició: La partida oberta creada per l'usuari es troba enregistrada al sistema	

2.6.8 Demanar suggeriment de rivals

Actor(s): Usuari novell o més	Casos d'ús relacionats: Llistar jugadors, Reptar jugador
Descripció: A partir de la puntuació i l'experiència de l'usuari, el sistema li mostra una sèrie de jugadors que s'ajustin a les característiques de l'usuari.	
Precondició: Cap	
Postcondició: Cap	

2.6.9 Donar-se d'alta al sistema

Actor(s): Usuari no registrat	Casos d'ús relacionats: Buscar usuari
Descripció: A partir d'unes dades que haurà d'introduir l'usuari, el sistema el dona d'alta.	
Precondició: L'usuari no existeix al sistema	
Postcondició: L'usuari es troba present a la base de dades	

2.6.10 Eliminar partida per jugador / eliminar partida oberta

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: L'usuari esborra del sistema una partida que ell mateix ha creat o un repte que ha fet o acceptat. Així com una partida oberta	
Precondició: La partida existeix i no s'ha jugat el tercer moviment de les blanques	
Postcondició: La partida ja no es troba al sistema.	

2.6.11 Fer moviment

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: Després de comprovar que el moviment es correcte, el sistema l'enregistrarà. Si el moviment defineix el resultat de la partida, el sistema la finalitzarà.	
Precondició: La partida per aquest moviment existeix i li toca a l'usuari que ha fet el moviment.	
Postcondició: El moviment ha estat enregistrat a la partida, si aquesta ha finalitzat estarà marcada com finalitzada	

2.6.12 Identificar-se al sistema

Actor(s): Usuari no registrat	Casos d'ús relacionats: Buscar usuari
Descripció: El sistema comprova que l'usuari existeix al sistema i li mostra la partida activa que necessita atenció més urgent. Si no en té cap, li mostra la llista de partides actives.	
Precondició: L'usuari existeix al sistema.	
Postcondició: Cap	

2.6.13 Modificar dades personals

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: El sistema mostra les dades que té enregistrades per l'usuari, permetent la seva modificació. Permet modificar les dades / canviar la contrasenya i canviar l'estil del tauler.	
Precondició: L'usuari existeix al sistema.	
Postcondició: Cap	

2.6.14 Navegar per la partida

Actor(s): Usuari no registrat o més	Casos d'ús relacionats: Veure partida
Descripció: El sistema mostrarà una posició concreta de la partida, aquella que l'usuari	

hagi escollit.
Precondició: La partida existeix
Postcondició: Cap

2.6.15 Oferir taules

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: El sistema enregistra un oferiment de taules fet per l'usuari al seu rival a la partida activa.	
Precondició: L'usuari i la partida existeixen i aquesta no està finalitzada.	
Postcondició: L'ofertament de taules està enregistrat	

2.6.16 Reclamar partida

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: El sistema dona com a guanyador (o eventualment taules) a l'usuari de la partida activa, enregistrant la partida com a finalitzada. Si l'usuari té material suficient com per, amb les pitjors jugades del rival, fer mat, llavors se li dona la partida, si no té material suficient per fer mat, encara que el rival faci les pitjors jugades possibles, es taules.	
Precondició: La partida i l'usuari existeixen. L'usuari juga a aquesta partida. Li toca moure al rival de l'usuari. El rival ha exhaurit el seu temps.	
Postcondició: La partida està enregistrada com a finalitzada donant com a guanyador a l'usuari o bé donant-la com a taules, segons el material que aquest tingui.	

2.6.17 Rendir-se

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: El sistema dona com a guanyador al rival, enregistrant la partida com a finalitzada.	
Precondició: La partida i l'usuari existeixen. L'usuari juga a aquesta partida.	
Postcondició: La partida està enregistrada com a finalitzada donant com a guanyador al rival.	

2.6.18 Reptar a jugador

Actor(s): Usuari novell o més	Casos d'ús relacionats: Demanar suggeriment de rivals, Buscar jugadors
Descripció: A partir d'una llista de jugadors cercats o suggerits el sistema permet escollir-ne un per reptar-lo a una partida, la qual pot tenir una posició inicial.	
Precondició: L'usuari reptador i el reptat existeixen.	
Postcondició: La partida es troba enregistrada al sistema.	

2.6.19 Saltar-se partida

Actor(s): Usuari novell o més	Casos d'ús relacionats: Llistar partides actives
Descripció: La partida activa passa a la cua de les partides actives independentment del temps que resti per fer el moviment.	
Precondició: L'usuari i la partida existeixen i l'usuari té més d'una partida activa.	
Postcondició: La partida es troba a l'última posició de la llista de partides actives de l'usuari.	

2.6.20 Sortir

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: L'usuari surt del sistema.	
Precondició: L'usuari es troba actiu al sistema.	
Postcondició: L'usuari no es troba actiu al sistema.	

2.6.21 Veure partida

Actor(s): Usuari no registrat o més	Casos d'ús relacionats: Navegar per partida, Buscar partides
Descripció: El sistema presenta el tauler de la partida que s'ha escollit, segons si a l'usuari li toca moure, si l'usuari ha mogut o si l'usuari no participa en la partida es mostren unes opcions o unes altres.	
Precondició: La partida existeix	
Postcondició: Cap	

2.6.22 Veure partides actives

Actor(s): Usuari novell o més	Casos d'ús relacionats: Llistar partides actives
Descripció: A partir de la llista de partides actives per l'usuari, el sistema les mostra ordenades per temps que queda per fer moviment.	
Precondició: L'usuari existeix	
Postcondició: Cap	

2.6.23 Veure partides finalitzades

Actor(s): Usuari novell o més	Casos d'ús relacionats: Cap
Descripció: Mostra la llista de partides finalitzades, ordenades segons la data de finalit-	

zació.
Precondició: L'usuari existeix
Postcondició: Cap

2.6.24 Llistar partides actives

Actor(s): Usuari novell o més	Casos d'ús relacionats: Saltar-se partida, Anar a la següent partida, Veure partides
Descripció: Accedeix a la base de dades obtenint la llista de les partides on l'usuari està jugant actualment.	
Precondició: L'usuari existeix.	
Postcondició: Cap	

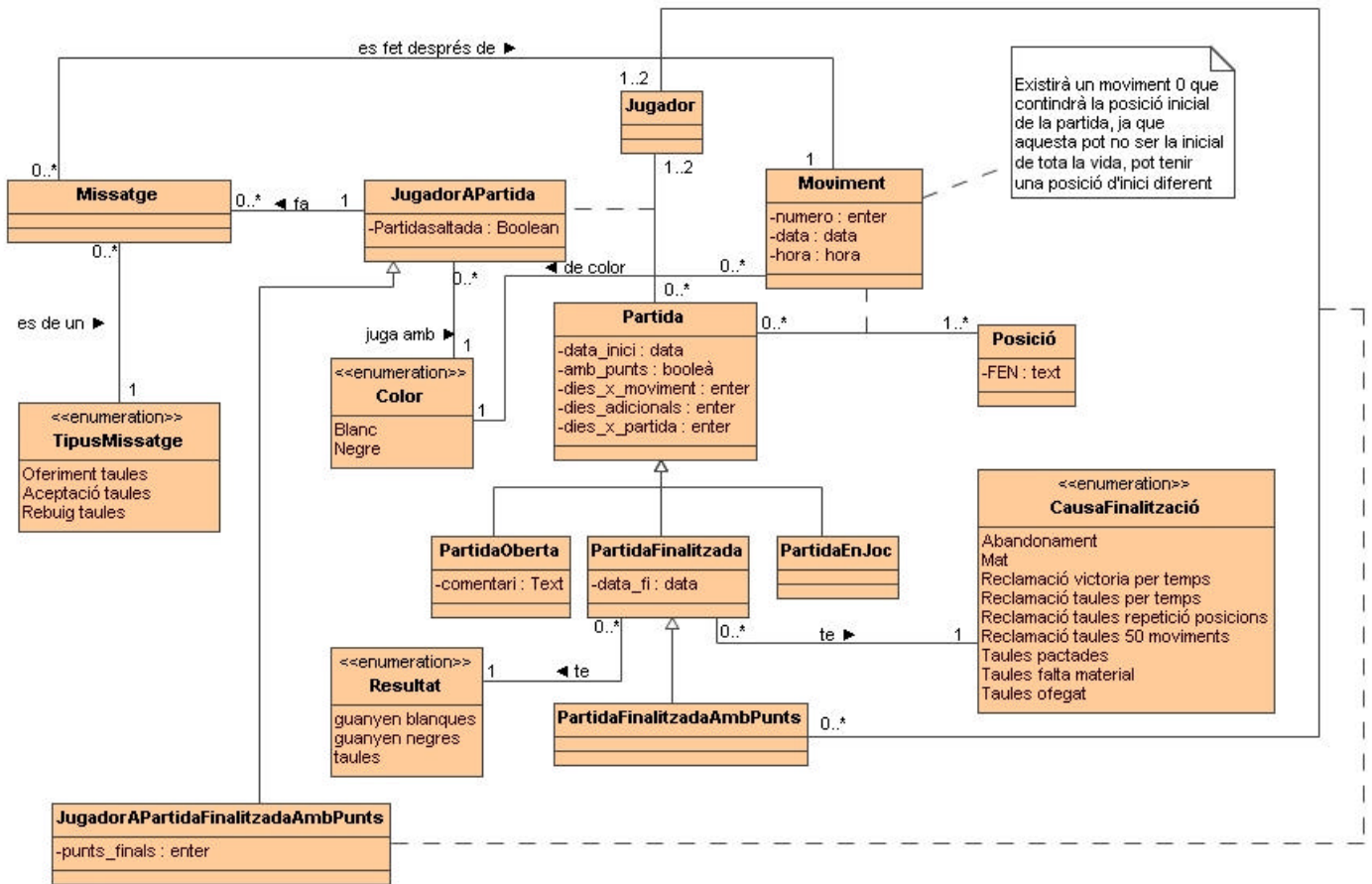
2.6.25 Llistar jugadors

Actor(s): Usuari novell o més	Casos d'ús relacionats: Buscar jugadors, Demanar suggeriment de rivals
Descripció: Accedeix a la base de dades obtenint una llista de jugadors segons uns paràmetres	
Precondició: Cap	
Postcondició: Cap	

2.6.26 Buscar usuari

Actor(s): Usuari no registrat	Casos d'ús relacionats: Donar-se d'alta al sistema, Identificar-se al sistema
Descripció: Buscar l'username d'un usuari al sistema i dona les seves dades.	
Precondició: Cap	
Postcondició: Cap	

2.7 Diagrama estàtic de l'anàlisi



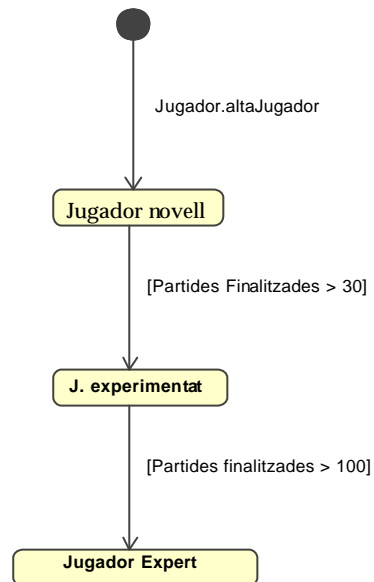
Imatge 7: Diagrama estàtic d'anàlisi

Els atributs del Jugador no s'han posat per tal de no complicar el diagrama, però serien: nom (text), usuari (text), contrasenya (text), pais (text), ciutat (text), punts (que seria un atribut derivat cercant punts_finals de JugadorAPartidaFinalitzadaAmbPunts), sexe (caràcter), data_naixement (data), idiomes parlats (text), membre_desde (data), ultima_visita (data), i comentaris (text).

En aquests moments els missatges no estan especificats, i aquí sols funcionen com a receptors dels oferiments, acceptacions i rebuigs de taules. Quan s'entri en el nivell 2, aquests missatges podran contenir a més, a més text.

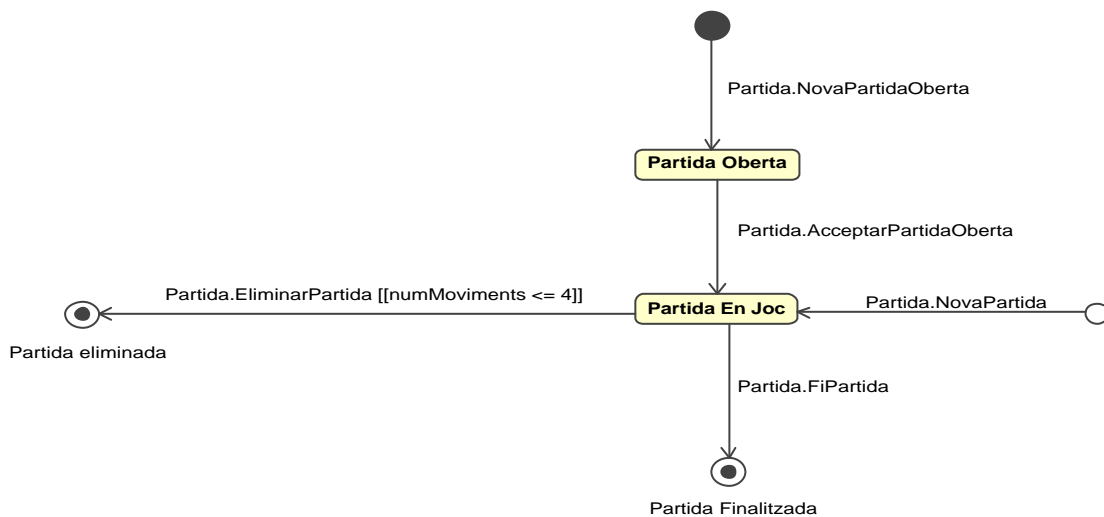
2.8 Diagrames d'estat

2.8.1 Classe Jugador



Imatge 8: Diagrama d'estat classe jugador

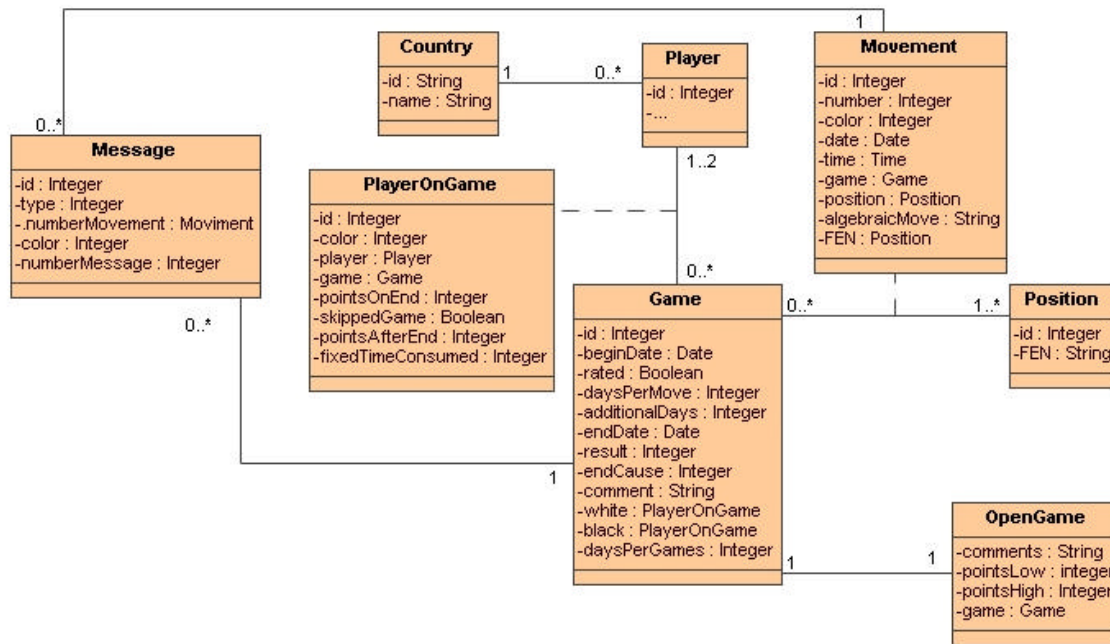
2.8.2 Classe Partida



Imatge 9: Diagrama d'estats classe partida

3 Disseny

3.1 Diagrama estàtic de Disseny



Imatge 10: Diagrama de disseny

Aquest serà el diagrama de classes que s'utilitzarà al desenvolupament de l'aplicació, lògicament no s'hi troben totes les classes, sols les del model de dades.

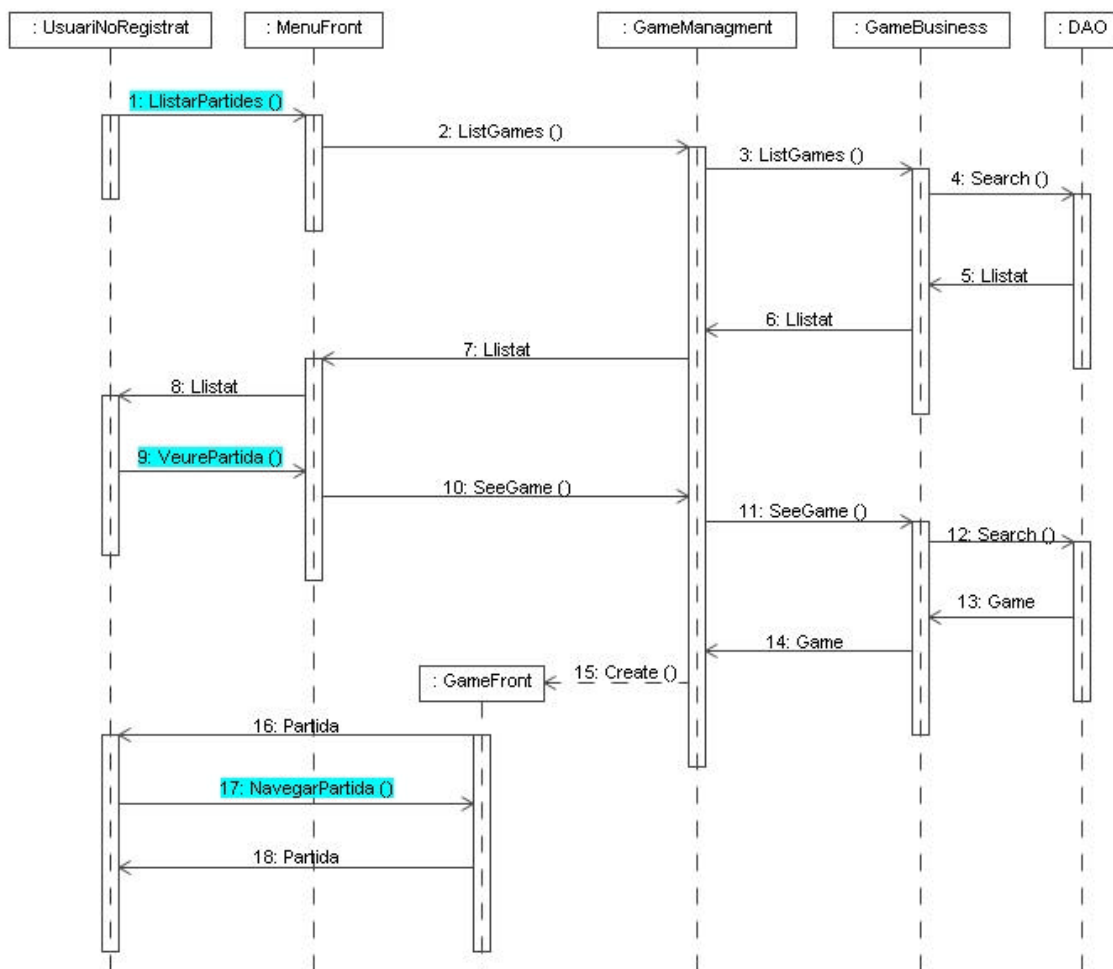
S'han afegit identificadors a cada classe, a més de atributs de les relacions que no son llistes. També s'ha traduït a l'anglès seguint els consells de Sun, degut a que l'aplicació s'implementarà en Java. Els atributs també han agafat l'estic característic de Java començant amb minúscula i separant les paraules amb majúscules. S'han eliminat les enumeracions que passen a ser enters, a més a més la Partida (Game) conté tota la informació necessària segons el tipus de partida que sigui, per tant ja no son necessàries les herències, perquè s'hi han generalitzat a la classe Game. Igual passa amb el JugadorAPartida (PlayerOnGame) que conté els punts al finalitzar (pointsOnEnd) de JugadorAPartidaFinalitzada. OpenGame es una excepció, s'ha considerat prou important, degut a que la majoria de partides no seran d'aquest tipus i generalitzar-lo hagués estat costós.

La classe Player contindrà tots els atributs especificats al diagrama estàtic de l'anàlisi, traduïts a l'anglès i amb la nomenclatura que s'hi ha utilitzat als altres atributs, a més d'alguns altres que es creguin convenient, com colors del tauler, etc...

3.2 Diagrames de seqüència

Existeixen casos d'ús (Sortir, Saltar partida, Anar a la següent partida etc...) que els he considerat massa simples com per fer un diagrama de seqüència, dels altres, he pensat que la millor manera de fer-ho era ajuntant varis casos d'ús relacionats en un sol diagrama, ja que d'aquesta manera la informació donada sembla més consistent i pot entendre's millor.

3.2.1 Llistar partides + Veure partida + Navegar Partida



Imatge 11: Diagrama de seqüència Llistar partides + Veure partida + Navegar partida

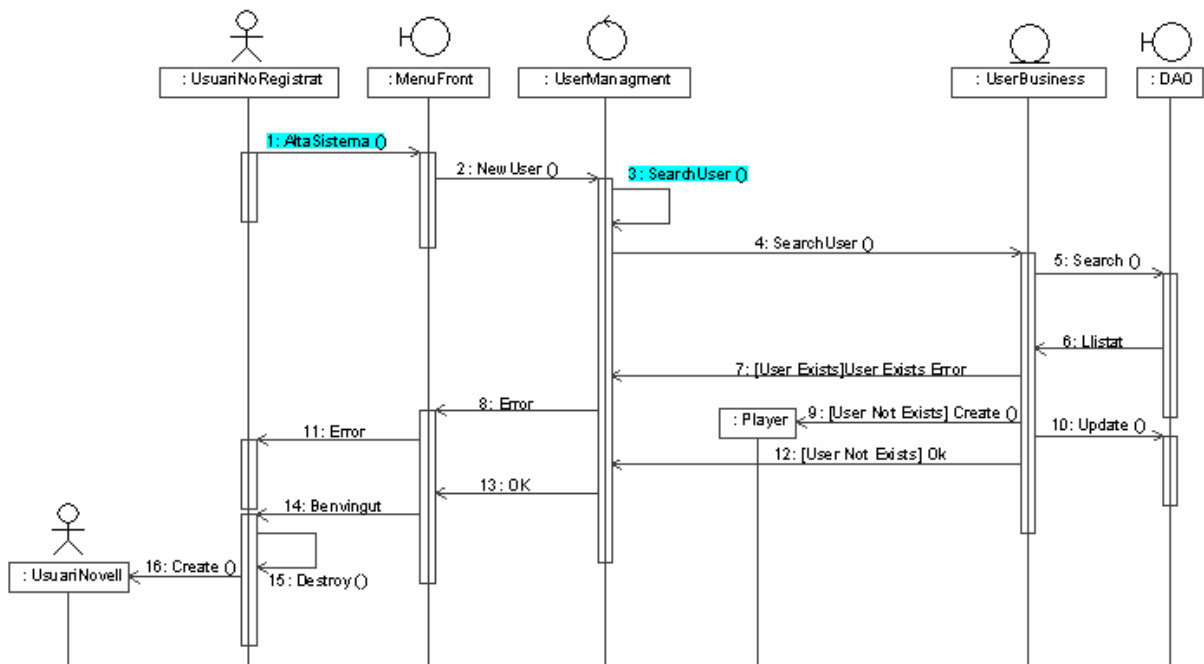
Cal observar dos coses:

1. La primera es que la classe Game no apareix com a tal. De fet la classe Game es trobarà dividida en dues: Game pròpiament dita, que contindrà les dades de la partida, que des de el punt de vista del patró DAO, serà el TransferObject, i la classe GameManagment, que

serà una classe sense estat, que contindrà la lògica de negoci relacionada amb la classe Game. Això s'ha fet així perquè m'ha semblat que serà molt més fàcil de desenvolupar i de mantenir que fer una classe Game que contingués tota la lògica de negoci.

2. La segona cosa es que per Navegar per la Partida no serà necessària cap petició al servidor, el tauler (GameFront) contindrà totes les dades de la partida, i per navegar per aquesta l'usuari no haurà de sortir del seu navegador. Això s'ha dissenyat així per tal de no sobrecarregar el servidor amb peticions de navegació.

3.2.2 Alta Sistema + Buscar usuari + Identificar-se al sistema

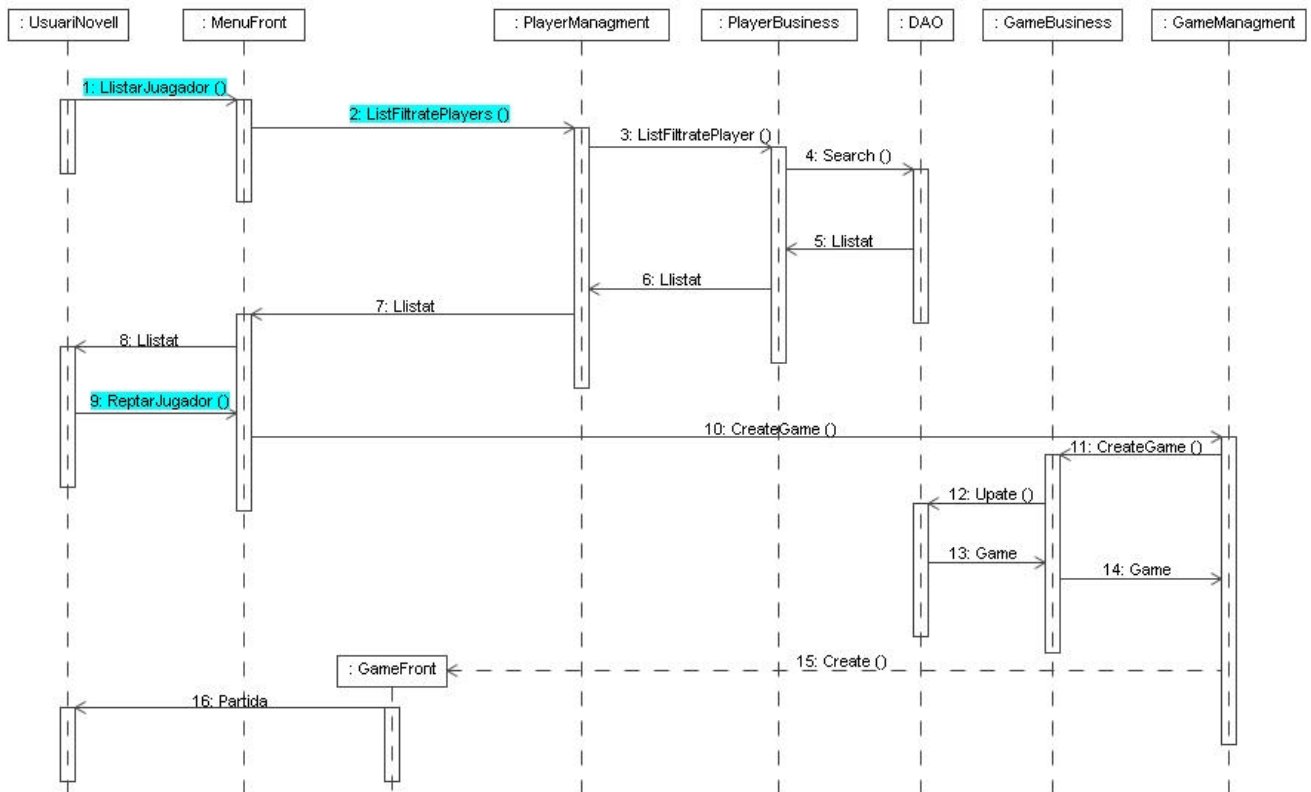


Imatge 12: Diagrama de seqüència: Alta sistema + buscar usuari + identificar-se al sistema

Quan un UsuariNoRegistrat es registra al sistema, deixa de ser un UsuariNoRegistrat per convertir-se en un UsuariNovell.

No s'ha dibuixat el fet de identificar-se al sistema. Es semblant a Alta Sistema, sols que amb les guardes canviades, també caldrà presentar-hi a l'usuari les partides actives, cosa que es pot veure en el diagrama de seqüència 4.

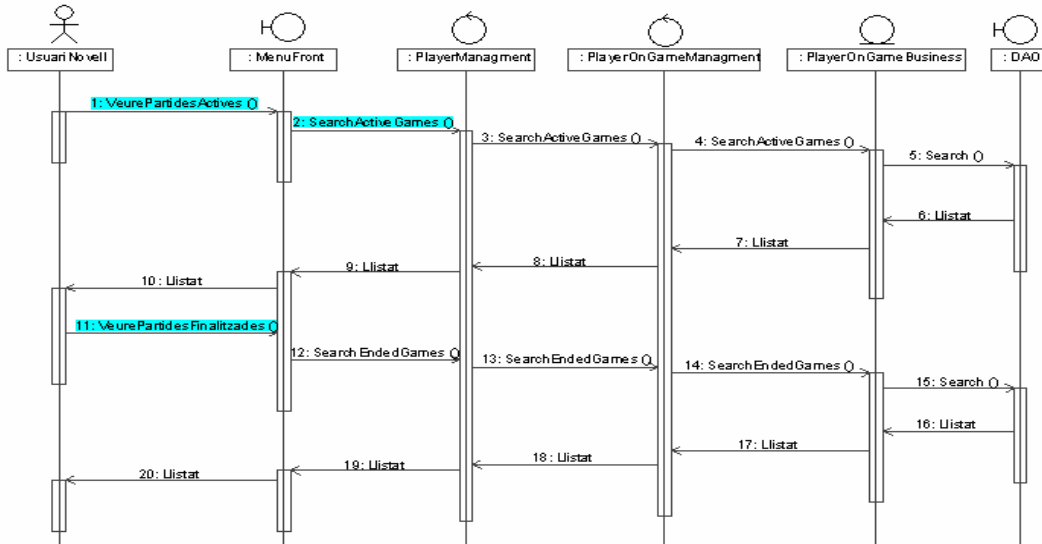
3.2.3 Llistar jugadors + Reptar jugadors + Llistar-Filtrar jugadors



Imatge 13: Diagrama de seqüència: Llistar jugadors + Reptar jugadors + Llistar-Filtrar jugadors

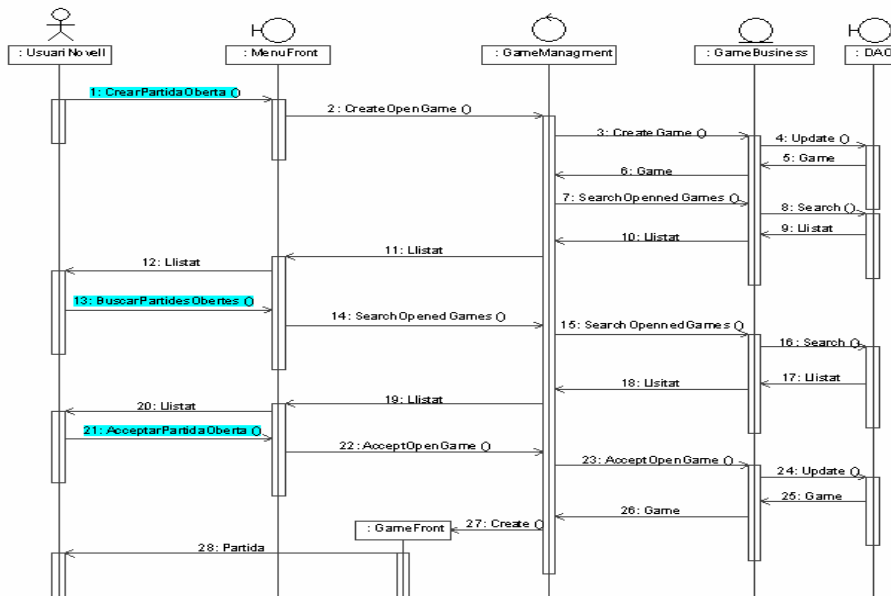
Quan GameBusiness crea una partida, a partir de DAO, cal adonar-se, que no sols es crea un Game, a més a més cal obtenir una Position i crear un Movement (l'inicial). No s'ha dibuixat perquè hagués quedat un diagrama molt dens.

3.2.4 Veure part. actives + Llistar partides actives + Veure finalitzades



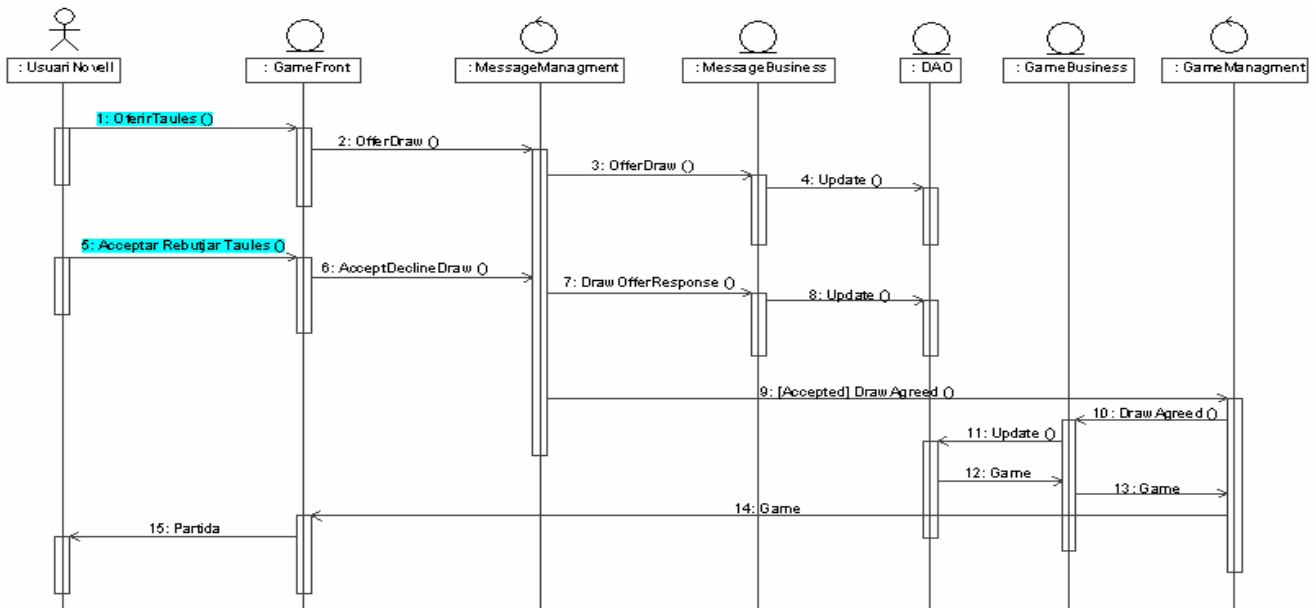
Imatge 14 Diagrama de seqüència: Veure partides actives + Llistar partides actives + Veure finalitzades

3.2.5 Crear partida oberta + Buscar partides obertes + Acceptar partida oberta



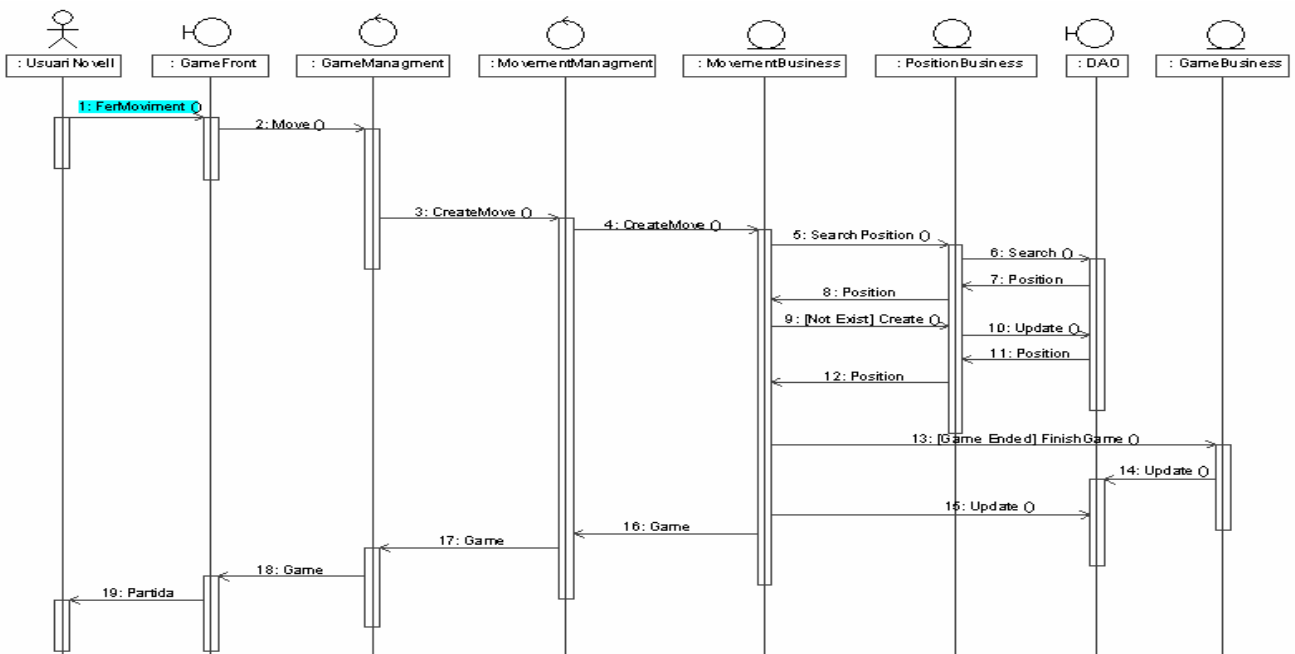
Imatge 15 Diagrama de seqüència Crear partida oberta + Buscar partides obertes + Acceptar partida oberta

3.2.6 Oferir taules + Acceptar/Rebutjar taules



Imatge 16 Diagrama de seqüència: Oferir taules + Acceptar / Rebutjar taules

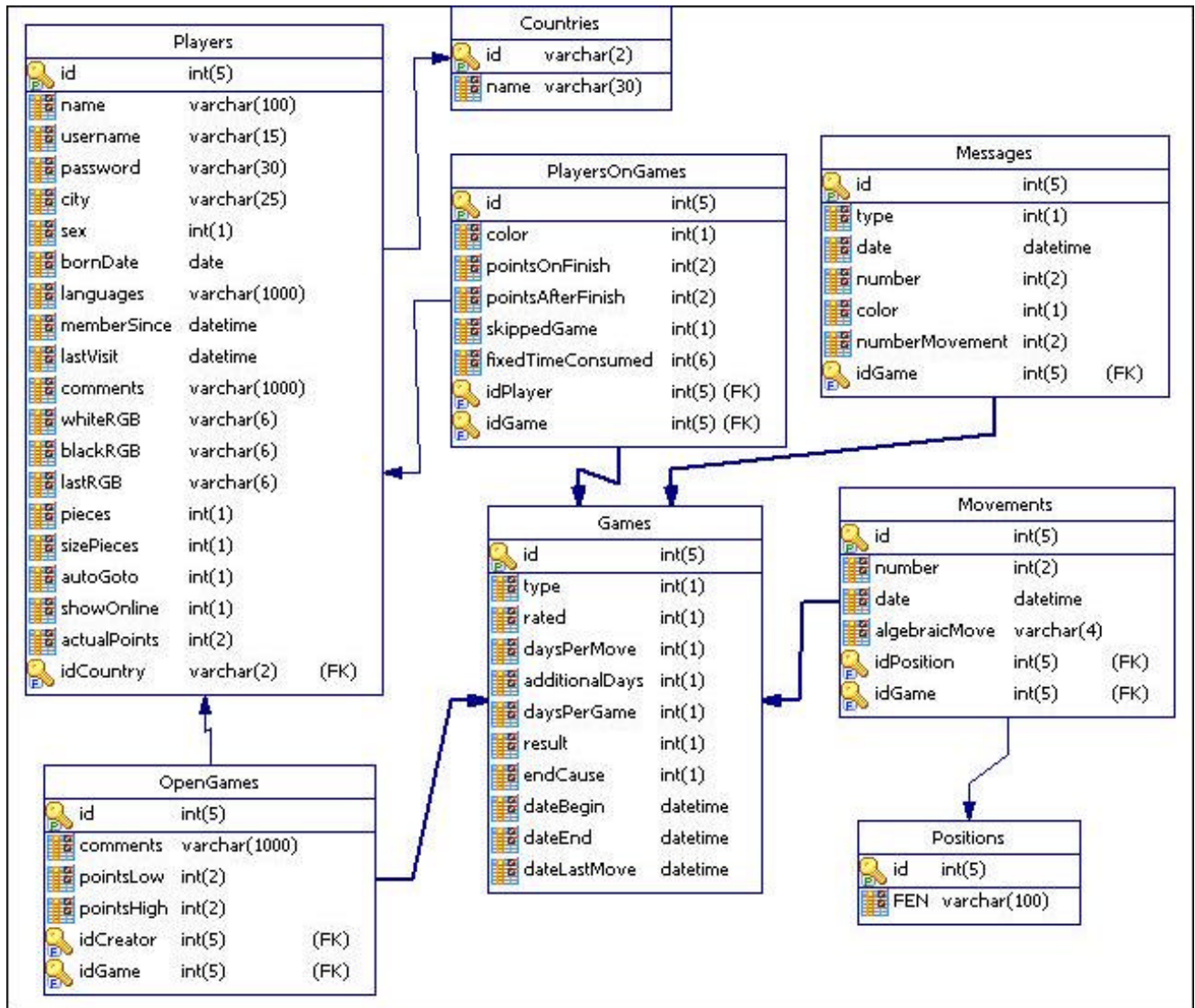
3.2.7 Fer moviment (també Rendir-se)



Imatge 17: Diagrama de seqüència: Fer moviment - Rendir-se

3.3 Disseny de la persistència

El SGBD que s'utilitzarà serà MySQL per un qüestió de coneixença i familiaritat. La Base de Dades es dissenyarà fent servir MicroOlap Database Designer for MySQL.



Imatge 18: Disseny de la persistència

Per tal de implementar la persistència s'utilitzarà DAO, implementat pel framework Hibernate. El patró DAO s'explicarà al següent apartat d'arquitectura.

3.4 Arquitectura del sistema

El sistema es desenvoluparà sota la plataforma J2EE. El fet de desenvolupar el projecte en Java, a més de ser obligatori perquè precisament aquesta es la branca d'estudi del projecte, fa que aquest sigui transportable entre diverses plataformes, i fàcil de programar, degut a que entre altres coses, no cal preocupar-se per alliberar memòria. També ens permet treballar amb el paradigma de l'orientació a l'objecte, que fa que ens podem beneficiar de les característiques d'aquest paradigma en quant a encapsulament, polimorfisme, abstracció, herència, etc...

El client hi accedirà mitjançant l'explorador, degut a que això permet més llibertat per les màquines dels usuaris, que sols necessiten un navegador per accedir-hi. En ser una aplicació web es necessitarà un servidor web amb servlets i JSP. S'ha optat pel Tomcat 6.0 per una qüestió de familiaritat, per què es gratuït i open-source. A més la seva gran utilització fa que hi hagi molta documentació associada.

S'utilitzarà el patró MVC, explicat més endavant, per tal de minimitzar el cost de futures ampliacions i manteniment. Per tal d'aplicar aquest patró s'optarà per Apache Struts. S'ha optat per Struts perquè al igual que amb el Tomcat, el seu us està bastant estès i es possible trobar bastant informació relacionada.

La Base de Dades relacional que s'utilitzarà per tal de fer la persistència serà MySQL, com s'ha dit a l'apartat de Disseny de la persistència, també per una qüestió de coneixença. Per tal de crear la capa de persistència l'objectiu es fer servir el framework Hibernate, per tal de facilitar l'implementació de l'accés a la BBDD.

L'aplicació s'implementarà amb l'ajuda d'Eclipse com a RAD, degut a que es gratuït, potent i el conec bastant bé.

L'arquitectura de la aplicació estarà dividida en capes, tal i com J2EE promou:

- **Capa client:** La qual estarà formada pel navegador.
- **Capa web:** Es on farem servir el patró MVC per tal de general el contingut per l'usuari. Estarà formada per Servlets + JSP's.
- **Capa de la lògica de negoci:** Aquesta capa es l'encarregada d'implementar les classes del nostre sistema. Sun recomana l'utilització d'EJB per aquesta capa. De totes maneres, sota el meu punt de vista, això suposaria la complicació excessiva d'un sistema que tot i que té molts casos d'ús no es excessivament complex. Per tant per tal de fer aquesta capa es faran servir POJOs (Plain Old Java Objects), o el que es el mateix objectes java normals i corrents.
- **Capa de persistència:** Per tal de fer aquesta capa s'utilitzarà el patró DAO, el qual està implementat al framework Hibernate.

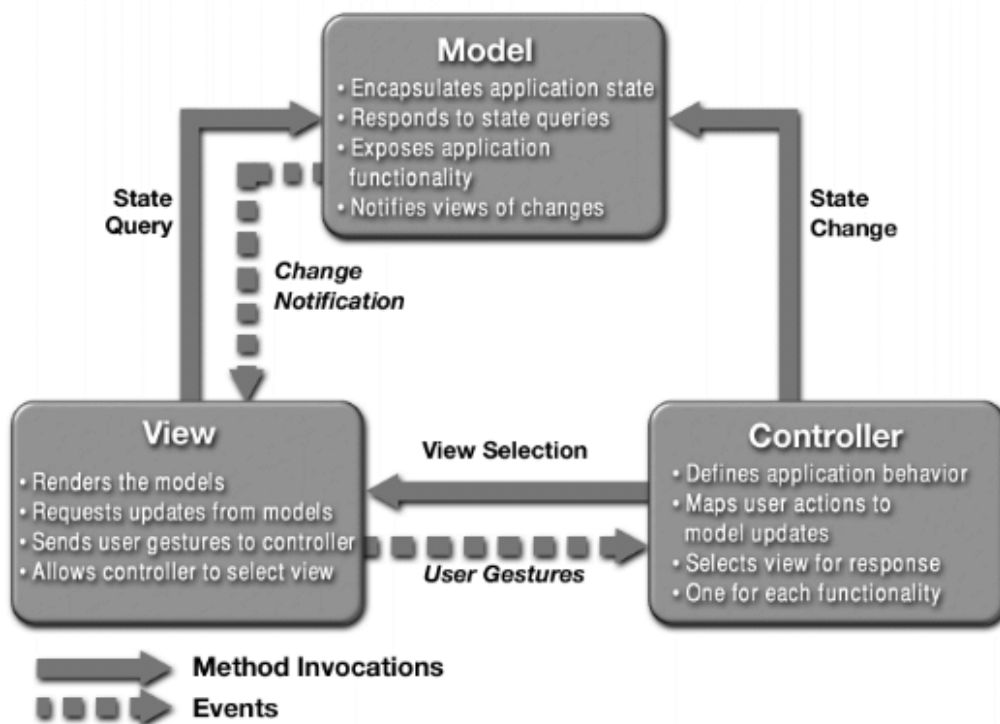
3.4.1 Patrons de disseny

Un patró de disseny es una solució acceptada com a correcta per la solució d'un determinat problema de disseny concret. A aquest projecte n'utilitzarem tres: MVC, DAO i Composite View. Tot seguit explicarem MVC.

3.4.1.1 El patró de disseny Model – View – Controller

El patró MVC, Model – View – Controller, o en català Model – Vista - Controlador, prova de crear un marc on l'aplicació es pugui desenvolupar de manera que el seu manteniment sigui molt més senzill.

La idea que hi ha sota aquest patró es separar les dades de l'aplicació, de la interfície d'usuari i del controlador de l'aplicació. Mitjançant aquesta divisió, com s'ha dit, es pretén que tant el desenvolupament com el manteniment sigui més simple i amb menys probabilitats de donar errors.



Imatge 19: El patró MVC

Anem a veure per sobre cadascún d'aquests components:

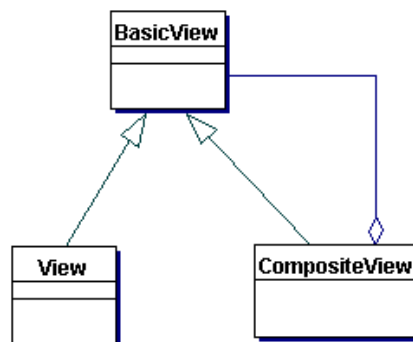
- **Model:** Es tractaria de les dades pròpiament dites de l'aplicació i de la funcionalitat d'aquesta. En el nostre cas les classes del diagrama estàtic de disseny estarien dins d'aquest model. El model no es preocupa pas de com s'enregistren les dades físicament. Tan sols es preocupa de com funcionen.
- **Controlador:** El controlador respon a les accions dels usuaris, interactua amb el model, actualitzant-lo si cal, determina quina es la vista a utilitzar per presentar la informació als usuaris.
- **Vista:** Es la encarregada de presentar físicament la informació a l'usuari.

El funcionament seria el següent:

1. L'usuari fa anar la interfície d'usuari (en el nostre cas un navegador web) d'alguna manera generant una petició.
2. El controlador rep aquesta petició i decideix com ha de interactuar amb la lògica de negoci per tal de realitzar la petició de l'usuari.
3. El controlador passa el control a la vista adequada.
4. La vista pot accedir al model per tal de presentar les dades que necessiti. Tot i així el model MAI té coneixement de la vista.
5. La vista mostra la informació a l'usuari

3.4.1.2 El patró Composite view

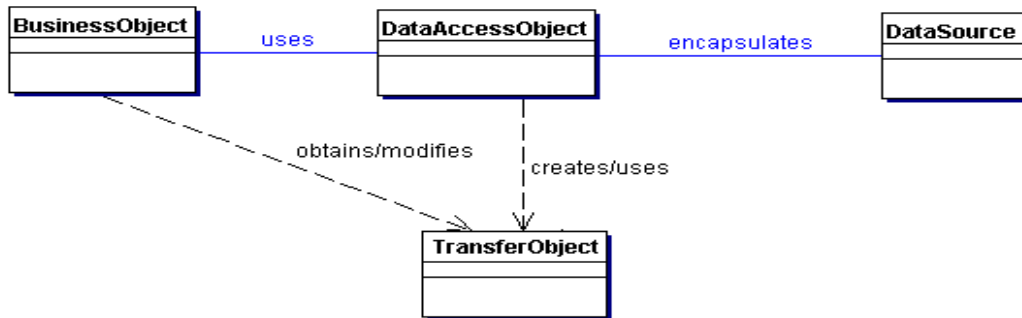
La idea darrera del patró Composite view es tractar les pàgines que es mostren a l'usuari com un conjunt de subpàgines independents entre elles, unides per formar un tot. Es a dir en lloc de veure la pàgina mostrada com una unitat, aquesta està formada per subpàgines reutilitzables, cosa que fa que el desenvolupament sigui molt més simple, fàcil de mantenir i menys propens a errors:



Imatge 20 El patró composite view

3.4.1.3 El patró DAO

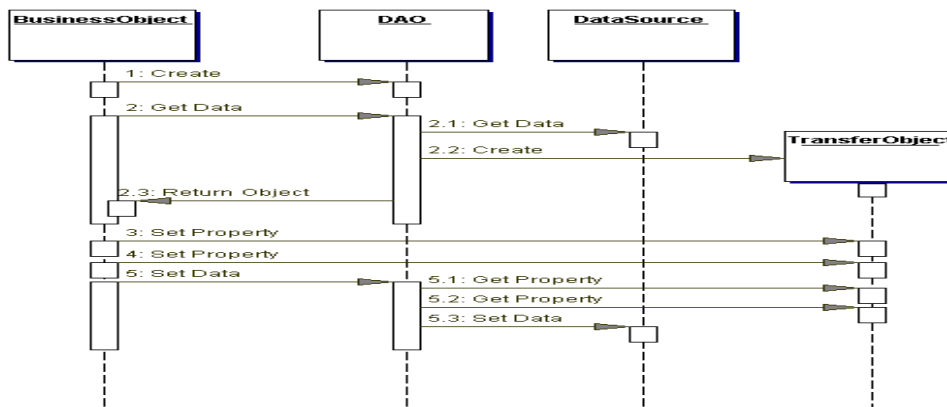
El patró DAO serveix per encapsular i abstraure l'accés a les dades pròpiament dites. DAO s'encarrega de la connexió física amb les dades i d'obtenir i guardar les dades. En el nostre cas aquest patró estarà implementat pel framework Hibernate.



Imatge 21: El patró DAO

El patró DAO està format per quatre participants:

- **BusinessObject**: Es l'encarregat de contenir la lògica de negoci de la aplicació, es l'objecte que necessita obtenir l'accés a les dades.
- **TransferObject**: Son les dades transferides pròpiament dites, tant per llegir-les, com per enregistrar-les.
- **DataAccessObject (DAO)**: Serveix per encapsular l'accés a les dades per tal de que l'objecte que contè la lògica de negoci tingui un accés transparent a aquestes, sense preocupar-se per com estan implementades.
- **DataSource**: Representa a les dades com a tals en el seu contingut físic. Pot ser una base de dades, un fitxer...



Imatge 22: Diagrama de seqüència patró DAO

4 Implementació

*El geni comença les grans obres, però només el treball les acaba, **Joseph Joubert, 1754-1824, escriptor francès***

Per tal de desenvolupar l'aplicació s'han fet servir els següents productes:

4.1 Productes que s'han fet servir

- **Tomcat 6.0**, contenidor de servlets, es on es desplega la aplicació per tal de que el navegador hi pugui accedir, s'ha escollit Tomcat per familiaritat i per què en no fer servir EJB, tampoc no era necessari un servidor d'aplicacions com ara JBoss
- **Eclipse Europa**, el RAD que s'ha fet servir. M'ha ajudat molt tot i que penso que no l'hi he tret tot el partit que podia, si l'hagués conegut millor segur que l'aplicació s'hagués desenvolupat més ràpidament
- **Struts 1.1**, del qual parlem més endavant
- **Hibernate 3.2.4.sp1**, igualment parlarem d'ell dintre d'algunes planes
- **Log 4J 1.2.11**, el sistema per tal de fer el seguiment dels esdeveniments que van succeint durant l'execució, inclòs amb l'Hibernate
- **JRE 1.6.0_02**, el Java Runtime Enviroment instal·lat al meu sistema.
- **Internet Chess Toolkit**, lliberia open source amb les regles dels escacs
- **MySQL Server 5.0**, el SGBD que utilitzarem, també s'ha escollit per familiaritat i per què s'hi pot trobar molta informació associada.
- **MicroOlap Database Designer for MySQL**, per tal de dissenyar la base de dades
- **MagicDraw™ UML 12.0**, per tal de fer els diferents gràfics d'aquesta documentació
- **Suite Office XP**, per tal de crear aquest arxiu, els dels diferents lliuraments i la presentació associada.

L'objectiu final de la aplicació es un fitxer war que pugui ser desplegat en un servidor, com per exemple el Tomcat, del que hem parlat. Ara es mostrarà l'estructura de carpetes de l'aplicació, primer de les fonts i després del war. Abans d'explicar -hi la estructura anem a aprofundir sobre els tres patrons de disseny, explicant com es durant a terme durant la implementació

4.2 Patrons

4.2.1 MVC, mitjançant Jakarta Struts

Per tal d'implementar el patró MVC s'utilitzarà, en aquest projecte, el framework Apache Struts. Aquest framework té una sèrie de components que son:

- **Action Servlet:** Es el cor del Controlador, s'encarrega de processar la sol·licitud de l'usuari, de decidir quina Action d'executarà i quina pàgina JSP s'encarregarà de mostrar la resposta a l'usuari.
- **Classes Action:** Son les classes Controladores creades pel programador, s'encarreguen d'accedir a les lògica de negoci de l'aplicació i realitzar la sol·licitud. Representen la part Controlador del model MVC
- **Planes JSP:** Son la part de la vista, encarregada de mostrar les dades a l'usuari. Representen la part Vista del model.

El funcionament d'Struts s'explica tot seguit, per tal de veure com interactuen aquests components:

1. L'ActionServlet rep una sol·licitud i decideix a quina classe Action ha de ser passada.
2. La classe Action processa la sol·licitud i interactua amb el Model (les classes Java programades pel programador per tal de fer anar la lògica de negoci), obtenint uns resultats.
3. L'ActionServlet es avisa que el resultat ja hi son i decideix quina plana JSP s'encarregarà de mostrar aquestes dades.
4. La plana JSP accedeix a les dades mostrant la resposta a l'usuari.

Tot això es planificat gràcies al fitxer xml: `struts-config.xml`, en el qual s'especifiquen les diverses decisions que cal que pregui el sistema segons les peticions que hi arriben.

La versió d'Struts que s'ha utilitzat es la 1.1, la qual es bastant vella, s'ha fet servir però perquè disposava de bastant documentació i llibres sobre ella, i això facilitava molt el seu aprenentatge.

4.2.2 El patró composite view a l'implementació

Aquest patró ha estat implementat mitjançant l'estructura `<jsp:include>`. Tota pàgina mostrada es en realitat un conjunt de pàgines que juntes formen la plana, i gaire bé totes comparteixen la majoria de parts (el menu, el peu, la capçalera), cosa que fa que sigui molt fàcil de mantenir. Si volguéssim, per exemple, canviar la capçalera, sols hauríem de canviar una plana (aquella que conté sols la capçalera) i totes les planes quedarien modificades quan la hi importessin.

4.2.3 El patró DAO: Hibernate

Hibernate es l'encarregat d'encapsular l'accés a les dades. Jo l'utilitzo a partir d'un objecte creat per mi anomenat `DataUtils`, que s'encarrega de cridar a les operacions d'Hibernate ocultant-les a la capa de negoci. El funcionament d'Hibernate es molt simple. Primer cal

fer un arxiu hbm.xml amb el mapeig dels diversos camps de la base de dades en membres de la classe. La classe en concret ha de tenir mètodes get i set adequats per tal de que Hibernate pugui fer la seva feina. Es necessari també un fitxer de configuració Hibernate.cfg.xml, que ens indica quins arxius hbm.xml s'han d'utilitzar i far la connexió a la base de dades amb usuari i contrasenya.

El funcionament d'Hibernate es el següent:

1. Es crea una sessió
2. S'obre una transacció
3. Es realitza l'operació, una lectura, una escriptura.
4. Si es una escriptura es crida al mètode commit de la transacció el qual fa els canvis permanents
5. En cas de que hi hagi algun problema es crida al mètode rollback que torna a deixar tot com estava per evitar incongruències.
6. Es tanca la sessió

La versió d'Hibernate que s'ha fet servir es la 3.2.4.sp1, per la mateixa raó que amb Struts, disposar de documentació sobre ella.

4.2.3.1 Associacions not lazy

Hibernate disposa d'un atribut que es fa servir als fitxers .hbm.xml, per indicar com s'ha de comportar el framework quan recuperi associacions de la base de dades. Es l'atribut lazy (mandrós).

Anem a entendre el seu funcionament amb un exemple. Suposem que la classe Game té un atribut que es una col·lecció anomenat Movements, que son els moviments de la partida. Si la relació entre totes dues tables es defineix com a lazy, quan es carregui un Game no es carregaran els seus moviments fins que no s'hi accedeixi. Això es fa així perquè en una base de dades relacional una sola lectura podria gaire bé significar carregar tota la base de dades en memòria.

Per tal de que això funcioni es necessari tenir oberta la sessió d'Hibernate i no tancar-la. Això pot complicar molt la programació, sobre tot, si com es el meu cas, les funcions de lectura / escriptura de la base de dades obren y tanquen la sessió just en el moment d'accedir a les dades. Després de pensar-ho molt i veure que realment no comportava cap perill, degut a l'estructura de la base de dades en que no hi ha associacions recursives, es va decidir marcar totes les associacions com a not lazy, de manera que quan es carregui un Game, per exemple, es carregaran els seus Movements. D'aquesta manera no es necessari

deixar oberta la sessió, i l'accés a la base de dades es més transparent per les funcions de la lògica de negoci que no s'han de preocupar per obrir i tancar sessions.

4.2.3.2 DataUtils: ¿Per qué una capa més entre la capa de negoci i Hibernate?

La explicació a això es bastant simple. La meua idea es que aquest projecte no es quedi solament en un TFC, vull que es converteixi en una realitat a la web, i, sent ambiciós, vull que, amb el temps, pugui arribar a ser uns dels millors llocs d'escacs que hi hagi a internet, agafant el bó i el millor de cada un dels que hi son presents. Hibernate simplifica molt les coses, el seu problema es que segons he llegit no està pensat per funcionar adequadament en entorns de producció. He llegit opinions que diuen, per exemple, que si el deixen funcionant tota la nit, s'acaba penjant, i, fins i tots, des de la propi lloc web dels creadors s'avisava que no està pensat per fer-lo servir de manera seriosa. Per això vaig decidir crear una classe que fes de frontera entre les classes de negoci e Hibernate, de manera que les primeres no sabessin res d'ell, i en cas de haver, en un futur, de canviar Hibernate per un altre sistema, fins i tot per SQL, o per una altre versió, poder-lo fer sense necessitat d'estar tocant les classes de la lògica de negoci. En cas de canviar cap a un altre framework que implementi el patró DAO, simplement es modificarà DataUtils, funcionant com funcionava fins llavors.

4.2.4 Llibreria opensource sobre el joc d'escacs: Internet Chess Toolkit

Degut a la falta de temps m'ha resultat impossible implementar les regles dels escacs per mi mateix, de fet n'hi ha un intent (board.java, inclosa al lliurament) però no controlava les taules i me'n vaig adonar que tenia uns quants bugs. Per tant vaig decidir utilitzar una llibreria externa de codi lliure d'escacs per comprovar si el moviment era correcte o no, així com per treballar amb el FENs.

4.3 Estructura dels fonts

En total n'hi ha set paquets:

- **chess.business**, aquí es troben les classes de la lògica de negoci que s'encarreguen de realitzar les operacions del sistema pròpiament dites. Son totes classes sense atributs (sense estats) amb mètodes estàtics que s'encarreguen de accedir a la capa d'accés a les dades i retornar la informació a les classes controladores.
- **chess.business.exceptions**, dintre d'aquest paquet es troben les excepcions personals que creen les classes anteriors, segons les situacions que es donin. Per exemple si en acceptar una partida oberta resulta que el creador i la persona que l'accepta son la mateixa persona doncs la classe encarregada gen erarà una excepció.

- **chess.business.utils**, en aquest paquet sols es troben dos classes: DataUtils i DataListResult. Deixant de banda que Hibernate es el veritable objecte DAO, aquests dos objectes representarien, per la nostra aplicació, el primer, una cosa semblant a l'objecte DAO i el segon una llista amb DTO's del mateix patró. El funcionament d'aquest objecte (DataUtils) es senzill, fa de capa entre la lògica de negoci i Hibernate que es l'encarregat d'accedir físicament a les dades, a les consultes, dona com a resultat un DataListResult, que es pasat a la lògica de negoci i aquesta el passa a la capa controladora. S'ha pensat en fer-lo així perquè d'aquesta manera no carreguem la lògica de negoci amb sentències de consulta.
- **chess.controller**, dintre d'aquesta classe es troben les classes Action de l'.struts. Son classes DispatchAction que permeten tenir tots els mètodes definits en una sola classe, cosa que fa que el número de fitxers no creixi de manera desorbitada. Aquestes classes son les encarregades de rebre la petició de l'ActionServlet de l'.struts, i decidir que cal fer per satisfer-la, utilitzant la lògica de negoci. No accedeixen al DAO del paquet chess.business.utils, si però als DTO's perquè necessiten mostrar les dades que recuperen. Un cop obtingudes les dades pasen el control a l'ActionServlet dient quina mesura ha de prendre'n, quina pàgina ha de mostrar.
- **chess.controller.utils**, aquí es troben una serie de classes utilitzades per motius diversos. La seva funció es ajudar als controladors. Les classes que es troben son els següents:
 - Action, aquesta classe s'encarrega de ajudar a crear el menu d'usuari.
 - ChessUocRequestProcessor, aquesta classes es encarregada d'observar cada petició i veure si te permís per fer allò que vol fer. Per exemple: una petició de crear partida provinent d'un usuari no identificat al sistema serà filtrada per aquesta classe. No hi hagut temps d'implementar-la adequadament.
 - Option, es encarregada de crear opcions per ser mostrades a llistes desplegable.
 - PlayerActions, es l'encarregada utilitzant la classe Action de crear el menu d'usuari.
 - PlayerLogged, es tracta d'una classe utilizada per tal d'encapsular l'usuari actiu. Implementa HttpSessionBindingListener per tal de capturar els fins de sessió sense logout
 - StartUp, es una classe que s'executa al inici de l'aplicació per tal d'actualitzar una sèrie de dades inicials, dedicades a tenir objectes que populeixin formularis.
 - Table, TableLine, TableElement, aquestes tres classes formen les taules. S'ha decidit que la pàgina que mostri les taules del sistema sigui totalment cega, es a dir, no sabrà res del que mostra, l'objecte Table crea les diverses línies que després llegirà la plana que s'encarregarà de mostrar les taules. D'aquesta manera penso que son més fàcil de desenvolupar les taules i donen més joc.

- **chess.data**, aquí es troba el model de dades. Les dades de l'aplicació pròpiament dites. Son objectes amb els diversos camps que formen l'aplicació. Els objectes `DataListResult` son llistes d'aquests objectes
- **chess.utils**, son dos objectes, un encarregat de calcular la diferència entre dues dates, i un altre per tal de mostrar missatges, tant d'error com d'informació.

A part ens troben amb el paquet **resources** que dins te l'arxiu `application.properties` que conté tot el text que l'aplicació mostra a l'usuari de manera que sigui fàcil utilitzar un altre idioma, tot i que aquesta funcionalitat no ha donat temps d'implementar-la, tot i així tot el text està traduït a l'anglès i al castellà.

4.4 Estructura del WAR

El fitxer `war` es l'objectiu del nostre projecte. Defineix l'aplicació web que s'haurà de desplegar al servidor. Conté quatre carpetes importants:

- **images**, aquesta carpeta es molt important per l'aplicació, conté gaire bé 700Kb d'imatges utilitzades per diversos motius, dins d'`images/icons` es motren icones que son utilitzades a les llistes per simplificar la seva estructura i fer-la més atractiva a l'usuari. Dintre d'`images/flags` es troben les banderes utilitzades per mostrar el país de l'usuari. I dintre de `images/pieces` es troben diversos tipus de peces i tamanys perquè l'usuari pugui customitzar el seu tauler com vulgui.
- **js**, dintre d'aquesta carpeta es troben varios `javascripts`, creats expressament per l'aplicació, el més important es l'encarregat de mostrar el menu de l'aplicació, n'hi ha un altre encarregat de crear cookies per tal de que el menu mantigui l'estat, altre que dibuixa el tauler i permet a l'usuari moure les peces i altres menys importants.
- **pages**, dins es troben les planes `jsp` que utilitza l'aplicació. A l'arrel es troben les planes principals, que com s'ha dit estan formades per un conjunt de subplanes. Te una estructura dividida en carpetes amb aquestes subplanes.
- **WEB-INF**, dintre d'aquesta carpeta hi ha dos arxius importants: `struts-config.xml`, que com s'ha explicat es l'encarregat de decidir el funcionament de l'aplicació i `validation.xml`, utilitzat per tal de validar els formularis, que es creen dinamicament (`DynaValidatorForms`) en lloc de crear-los del tipus `ActionForm`. Es va pensar que tenir definides les Forms en un fitxer XML era més simple que tenir que crear un objecte per cada una. Dintre d'aquesta carpeta es troba la carpeta **lib** on es troben les lliberies de l'aplicació i la carpeta **classes** on es troben els fitxers compilats dels fonts. També dintre de classes (no he pogut col·locar-los a un altre lloc) es troben els fitxers d'Hibernate (un per cada taula-classe) i el fitxer de configuració de l'Hibernate, així com el fitxer de configuració del `log4j`, sistema utilitzat per tal de fer els diversos logs de l'aplicació.

4.5 Proves

Durant la realització del projecte s'han anat fent proves per tal de comprovar que les funcionalitats funcionaven com estava previst. Però hi hagut una part, tota la de la partida que no s'ha pogut provar per falta de temps, el front d'usuari de la partida em va resultar molt més costós de previst i ha resultat impossible testar la partida i els seus casos d'ús relacionats.

5 Conclusions

Només sé que no sé res, Sòcrates, 470 AC-399 AC, filòsof grec

Tal i com es va explicar a l'apartat de planificació la part de la que em sento menys orgullós d'aquest projecte es precisament aquesta. No s'ha pogut dur a terme tal i com estava planificat. La meua idea original era que es poguessin implementar tots els casos d'ús de nivell 2, però això, tal i com s'ha vist ha estat impossible.

En no arribar al nivell 2, lògicament la majoria d'actors presents en els requisits no participen al sistema, sols n'hi ha dos: Usuari no registrat, i usuari novell, tot i no canviar cap funcionalitat, sí que quan un novell finalitza la seva partida 31 passa a un altre sistema de puntuació, i llavors sí que es pot considerar experimentat, tot i que les funcions que té disponibles no canviïn.

Sigui com sigui, tot i que no puc estar content del resultat obtingut, sí que ho estic de tot el que he après. Tot i les dificultats atravesades, i potser precisament per aquestes, aquest projecte m'ha permès reconciliar-me amb mi mateix i em sento orgullós d'haver-lo finalitzat tot i que no hagi estat tal i com hagués volgut.

Em queda molt per aprendre'n, de fet, sols he pogut ratllar la superfície de J2EE, i el meu coneixement tant d'Struts com d'Hibernate encara es pot considerar el d'un principiant, però tot i això aquest projecte m'ha obert les portes a un món, la programació web, que m'era bastant desconegut, i m'ha agradat força el que he vist.

Lògicament, tal i com vaig dir, el projecte del club d'escacs no s'acaba aquí. La meua idea es dur-lo a fase d'exploració, tot i que per tal de fer això sé que caldrà canviar algunes coses, com ara fer la migració cap a una versió d'Struts més moderna, així com llegir més sobre Hibernate, perquè pel que he pogut esbrinar, els seus mateixos creadors expliquen que no està creat per tal de fer-lo servir en un entorn real, al menys tal i com jo l'he fet servir, així que encara cal documentar-se més sobre aquest tema.

Existeixen però apartats que cal millorar, a part de fer créixer les funcionalitats del sistema. Una cosa que m'ha dut molts problemes i que al final no he pogut arreglar es la gestió dels caràcters. Tot i haver utilitzar Unicode UTF-8, tant a la web, com a la base de dades, alguna cosa estranya falla amb la comunicació entre totes dues, suposo que degut a la versió

antiga d'Struts que faig servir, quan s'utilitzen accents i caràcters no normals. També es necessària la implementació de la traducció tant a anglès com a castellà, i més estant traduït com està.

En resum puc afirmar, que en línies generals, tot i no haver complert les fites que jo mateix em vaig marcar a principi del semestre i no estar content al cent per cent, he après força. He vist unes eines molt potents, tant Struts, com Hibernate, fan que la creació d'una aplicació com aquesta, tot i no ser una tarea senzilla, sigui molt més simple del que sembla a priori. Això em fa venir moltes ganes de continuar el seu aprenentatge i millorar el meus coneixements. En aquest sentit la continuació amb el projecte pel meu compte em pot servir per afiançar aquests coneixements.

6 Glossari

Per tota aquella persona no familiaritzada amb els escacs es adient explicar tots aquells mots que han sortir en aquest document que poden ser de difícil comprensió. En els següents lliuraments també es crearà un glossari com aquest en que sortiran les paraules que s'utilitzin en aquests lliuraments.

AEAC: Asociación Española de Ajedrez por Correspondencia.

Apertura: Primers moviments en una partida d'escacs. Es el moment on les peces comencen a desenvolupar-se. N'hi ha moltes apertures, però aquestes es poden resumir en una espècie d'arbre d'apertures que l'usuari tindrà a la seva disposició per tal de poder consultar-lo.

Escac per correspondència: Variant dels escacs en que dos usuaris no cal que estiguin al mateix temps sobre un mateix tauler. A l'escac per correspondència original el normal era que passés molt de temps, la carta havia d'arribar a destí, entre un moviment i un altre. Degut a Internet això s'ha simplificat i s'ha fet molt més ràpid, cosa que permet poder jugar partides no presencials amb controls de temps de l'ordre d'un dia per moviment, per exemple, cosa que fa la partida molt viva i permet a usuaris amb poc temps jugar moltes partides a l'hora. Així com també permet a altres usuaris jugar poques partides, però analitzar cada moviment durant un temps no gens envejable.

Control de temps: Temps que un jugador té per fer un moviment, o varis moviments, o tota la partida. A l'escac per correspondència a Internet, tres exemples poden ser: 3 dies / moviment, 30 dies / tota la partida, 1 dia / moviment + 7 dies de temps addicional. Transcorregut el temps del seu adversari, un jugador pot reclamar la victòria de la partida.

Vacances: Període durant el qual un usuari preveu que no podrà moure. Els diferents sistemes d'escacs presents a la xarxa tracten les vacances de forma diferent. Uns afegeixen dies al rellotges dels jugadors, altres simplement impedeixen que es pugui reclamar una partida contra aquests. En el nostre sistema quan un usuari entri un temps de vacances els seus rellotges seran automàticament aturats durant el temps de vacances que l'usuari hagi escollit, d'aquesta manera penso que les vacances son més visuals. Durant les vacances, un usuari, si vol, podrà moure, sense que això signifiqui que surti de les vacances. Lògicament n'hi haurà un número de dies de vacances limitat els quals s'esgotaran, s'hagin o no fets servir, a final de cada any.

Moviment condicional Aquesta expressió no està relacionada amb els escacs pròpiament dits si no amb el sistema. Quant un usuari preveu que el seu rival farà un moviment concret, pot programar el sistema perquè aquest respongui automàticament a aquest moviment. De manera que la partida s'agilitza.

PGN: Portable Game Notation. Tipus de fitxer que representa tots els moviments així com la informació relacionada amb una partida: torneig, nom dels jugadors, lloc on es juga, data... Aquest tipus de fitxer es pot llegir amb la majoria de programes d'escacs, tant programes de joc, com bases de dades.

FEN: Forsyth-Edwards Notation. Es una notació que permet representar textualment el tauler d'escacs, de manera que sigui possible reprendre una partida a partir de la posició apuntada. Cal veure la diferència entre PGN i FEN. Amb FEN no es té informació dels moviments que van dur fins aquella posició. Es una simple descripció del tauler per tal de continuar jugant des de aquella posició. Amb PGN es té tota la informació de la partida. La majoria de programes d'escacs també permeten la lectura de fitxers FEN, com pegar-los directament, com si de un processador de text es tractés, representant al tauler automàticament aquella posició.

Torneig: Competició en que un jugador o un equip competeix amb altres per ser campió.

Equip: Conjunt de jugadors que s'uneixen per competir amb altres equips.

Capità: Líder d'un equip, es qui escull l'alineació d'un equip per tal de competir a un torneig, així com qui l'hi apunta.

Elo: Sistema inventat als anys seixanta pel nord-americà Arpad Elo, per tal de mesurar la força relativa dels jugadors d'escacs.

FIDE: Fédération Internationale des Échecs

USCF: United States Chess Federation

7 Bibliografia

7.1 Llibres

- **Ashmore, Derek C.** *The J2EE Architects Handbook*. DVD Press. 2004
- **Bauer, Christian, King, Gavin.** *Java persistence with Hibernate*. Manning. 2007
- **García de Jalón, Javier, Rodríguez, José Ignacio y otros.** *Aprenda Java como si estuviera en primero*. Escuela superior de Ingenieros Industriales de San Sebastián. Universidad de Navarra. 2001
- **Hightower, Rick.** *Jakarta Struts Live*. Source Beat. 2004
- **Patchine, Alexandre.** *1000 Java Tips*. 2005
- **Peak, Patrick, Heudecker Nick.** *Hibernate Quickly*. Manning. 2006
- **Robinson, Mike, Finkelstein, Ellen.** *Jakarta Struts for Dummies*. WilleyPublishing. 2004

7.2 Pàgines web

- <http://java.sun.com>, plana de java, on es pot accedir als javadocs i als patrons, a més de poder descarregar el JDK.
- <http://www.hibernate.org>, plana d'hibernate
- <http://today.java.net>, plana utilitzada per trobar alguns exemples de programació per coses concretes
- <http://www.desarrolloweb.com>, igual que l'anterior
- <http://dev.mysql.com>, plana del MySQL
- <http://www.whirlycott.com>, plana utilitzada en alguns moments de dubte
- <http://www.gifworks.com>, plana que té un sistema on-line que permet manipular imatges, utilitzada per tal de modificar els gifs que formen part de l'aplicació.
- <http://www.markfennell.com>, plana des d'on em vaig descarregar les banderes dels països.
- <http://struts.apache.org>, plana d'struts.
- <http://www.fide.com/>, plana de federació internacional d'escacs
- <http://www.matepostal.com>, plana de la federació espanyola d'escacs per correspondència.

- <http://ictk.sourceforge.net/>, plana de l'Internet Chess Toolkit, lliberia opensource que s'ha fet servir per desenvolupar la part del projecte relacionada amb les regles dels escacs

8 Annex A: Sistema de puntuació

8.1 Sistema de puntuació durant les 30 primeres partides

Un jugador començarà sense puntuació. Durant les seves 30 primeres partides s'aplicarà el següent sistema de puntuació:

$$R_n = R_c + D_p * F$$

on:

R_n, es la nova puntuació del jugador

R_c, es la mitjana aritmètica de la puntuació dels jugadors que han estat rivals, si algun d'ells es un rival sense puntuació se li suposarà una puntuació de 1400.

D_p, s'obté de la següent taula, on **P** es el percentatge de victòries aconseguides pel jugador, (unes taules comptarien com mitja victòria), i els nombres en vermell son considerats nombres negatius.

F, també s'obté fent servir **P**, segons la següent fórmula:

$$F = -2 * P^2 + 2 * P + 0,5$$

P	D _p	P	D _p	P	D _p	P	D _p	P	D _p	P	D _p
1.00	677	0.83	273	0.66	117	0.49	7	0.32	133	0.15	296
0.99	677	0.82	262	0.65	110	0.48	14	0.31	141	0.14	309
0.98	589	0.81	251	0.64	102	0.47	21	0.30	149	0.13	322
0.97	538	0.80	240	0.63	95	0.46	29	0.29	158	0.12	336
0.96	501	0.79	230	0.62	87	0.45	36	0.28	166	0.11	351
0.95	470	0.78	220	0.61	80	0.44	43	0.27	175	0.10	366
0.94	444	0.77	211	0.60	72	0.43	50	0.26	184	0.09	383
0.93	422	0.76	202	0.59	65	0.42	57	0.25	193	0.08	401
0.92	401	0.75	193	0.58	57	0.41	65	0.24	202	0.07	422
0.91	383	0.74	184	0.57	50	0.40	72	0.23	211	0.06	444
0.90	366	0.73	175	0.56	43	0.39	80	0.22	220	0.05	470
0.89	351	0.72	166	0.55	36	0.38	87	0.21	230	0.04	501
0.88	336	0.71	158	0.54	29	0.37	95	0.20	240	0.03	538
0.87	322	0.70	149	0.53	21	0.36	102	0.19	251	0.02	589
0.86	309	0.69	141	0.52	14	0.35	110	0.18	262	0.01	677
0.85	296	0.68	133	0.51	7	0.34	117	0.17	273	0.00	677
0.84	284	0.67	125	0.50	0	0.33	125	0.16	284		

Taula 3: Taula de puntuació

Ho entendrem millor amb un exemple. Un jugador es dona d'alta al nostre sistema i juga contra cinc jugadors amb les següents puntuacions: 1650, un sense puntuació, 1900, 1200 i

2100, el jugador ha aconseguit dos victòries, dos taules i una derrota (dona igual contra qui).

Rc, seria la mitja de les puntuacions dels jugadors: $(1600+1400+1900+1250+2100)/5=1650$

P, seria el percentatge de victòries:

1 punt per cada victòria: 2 victòries, per tant, 2 punts +

0,5 punts per cada empat: 2 taules, per tant, 1 punt +

0 per cada derrota: 0 punts,

fent un total de 3 punts de 5 possibles, per tant una **P** de 0,6 (60%)

Dp, per tant, utilitzant la taula seria: 72

i **F** seria $-2*(0,6)^2+2*0,6+0,5 = 0,98$

per tant la puntuació del jugador seria de $1650 + 72*0,98 = 1721$.

Cada cop que un jugador novell (30 o menys partides) acabi una partida aquest sistema de puntuació es posaria en marxa i es tornaria a recalculer afegint-hi el nou rival.

La manera com es calculen els punts dels usuaris novells es la manera utilitzada a la [AEAC](#) tot i que a partir de la partida 31, aquest organisme fa servir una manera diferent de calcular l'Elo, basant-se en taules, en períodes i en mitjanes de puntuacions obtingudes en aquest períodes, mentre en el nostre cas ens basarem en formules, com es veurà al següent apartat. A més que el càlcul es farà al finalitzar cada partida, no en finalitzar un període de temps. Sigui com sigui, tot i que son diferents formes de fer-ho, l'esperit que hi ha darrera es el mateix.

8.2 Sistema de puntuació a partir de la partida 31

A partir de la 31 partida s'aplicaria el sistema Elo, el qual es basa en les següents formules:

1. Resultat (0 a 1) que s'espera que el jugador A faci davant d'una partida contra el jugador B

(**Ra**: Puntuació d'A, **Rb**: Puntuació del rival, de B)

$$Ea = \frac{1}{1 + 10^{(Rb - Ra) / 400}}$$

Com a exemple si el jugador A te 2500 punts i el B, 2000, s'espera que per A el resultat sigui de 0,95 i el de B de 0,05. Òbviament això es impossible perquè el resultat d'una partida per un jugador o bé es 0 (perd) o bé es 0,5 (empata) o bé es 1 (guanya). De totes maneres

ara mateix estem parlant de probabilitats. O sigui que, podríem dir per fer-ho simple, el jugador A te una probabilitat de guanyar del 95%, i el B del 5%.

2. Amb aquestes probabilitats calculades entra en joc una altra fórmula, la que calcula la puntuació final:

$$RFa = Ra + K(Sa - Ea)$$

On **RFa** es la puntuació final d'A, **Ra**, la inicial, **Sa** es el resultat que A ha obtingut realment a la partida, i **K** es calcula així:

$$\begin{aligned} Ra < 2100 &\Rightarrow 32 \\ Ra > 2400 &\Rightarrow 16 \\ Ra \geq 2100 \\ Ra \leq 2400 &\left. \vphantom{\begin{array}{l} Ra \geq 2100 \\ Ra \leq 2400 \end{array}} \right\} \Rightarrow 24 \end{aligned}$$

La K serveix per estabilitzar les puntuacions més elevades i per premiar als jugadors amb puntuacions més baixes si guanyen a un jugador de puntuació més elevada. Existeixen varies maneres de calcular la K, la FIDE utilitza una altra diferent. Aquesta es utilitzada per la USCF, i l'he considerada adequada per què en té tres nivells, cosa que permet una divisió més acurada entre els diferents nivells de jugadors. Mentre que la de la FIDE per jugadors amb més de 30 partides, sols té dos nivells. No s'ha utilitzat la K del sistema de l'AEAC perquè m'ha semblat massa complicat, degut a que implicava dos variables pel seu càlcul.

Continuem, però, amb el nostre exemple, per tal de fixar les idees. Anem a imaginar que A perd, la **K** seria de 16, perquè A te 2500 punts, la **Sa** seria 0 perquè ha perdut per tant la seva nova puntuació seria:

$$RFa = 2500 + 16 * (0 - 0,95) = 2485$$

Per contra la **K** de B seria de 32, té 2000 punts, **Sb** (Sa pel jugador B) seria 1, perquè ha guanyat, per tant la seva nova puntuació seria:

$$RFb = 2000 + 32 * (1 - 0,05) = 2030$$

B ha pujat 30, A ha baixat 15, degut a les diferents K's. Cal adonar-se que si la K arriba a ser la mateixa, B hauria guanyat el que A hauria perdut, perquè $Sa+Sb=1$ i $Ea+Eb$ també igual a 1.

8.3 Consideracions sobre les puntuacions dins el sistema

Normalment tant la FIDE, com d'altres organismes encarregats de organitzar els diversos torneigs d'escacs fan el càlcul de punts per jugadors no novells a partir de diverses partides, (d'un torneig, d'un període de temps, p.e.). De manera que es treballa amb les mitjanes de punts que el jugador ha aconseguit durant aquest període. Així que els càlculs no es calculen al finalitzar cada partida si no cada període (Una cosa semblant a la manera com nosaltres tractem les 30 primeres partides). En el nostre sistema, al igual que a la majoria de sistemes d'escac per servidor, això no serà pas així, a partir de la partida 31, els punts es calcularan en finalitzar cada partida, tenint en compte sols els punts del rival d'aquesta partida. D'aquesta manera la fluctuació de punts serà més habitual, i els jugadors no hauran d'esperar a haver finalitzat un torneig per que es calculi la seva puntuació, cosa que farà que tot sigui més fluid, i que els jugadors hi estiguin més enganxats.

Un altre punt a considerar, el qual neix directament del fet de que cada jugador podrà estar jugant moltes partides al mateix temps, cosa que no passa a l'escac normal, es quina serà la puntuació inicial per fer els càlculs. Anem a entendre millor aquest problema amb un exemple: Un jugador A comença dos partides. Totes dues, lògicament, les comença amb els mateixos punts, posem-hi 1400, acaba primer una d'elles, contra B, imaginem que guanya 100 punts, ja tindrà 1500, tres moviments després acaba l'altra partida, contra C, quina puntuació caldrà fer servir per calcular-hi els nous punts? Aquella amb que va començar la partida contra C? 1400 o aquella amb la que ha acabat la partida contra C? 1500. Cap de les dos solucions sembla justa. La de la puntuació última no sembla ser del tot just perquè C s'ha enfrontat durant tota la partida a un jugador de 1400, sols al final d'aquesta A ha esdevingut 1500. A més això pot dur a tripijocs per part d'alguns jugadors llestos que vegin que tenen una posició guanyadora davant d'un jugador que té una altra posició guanyadora en una altre tauler contra un altre jugador. Posem-hi el mateix cas, imaginem que C tingui una posició de mat imminent sobre A, però C veu que A està apunt de guanyar a B, i decideix endarrerir el final de la partida fins que A hagi guanyat i tingui 1500, llavors dona mat, i guanya més punts perquè ha guanyat a un jugador de 1500 i no de 1400. Si utilitzéssim la puntuació inicial tampoc seria just perquè una partida dura molt i es possible fins i tot que hi hagi 300 o 400 punts de diferencia entre l'inici d'una partida i el seu final, sobre tot si el rival es un rival novell. Després de donar-hi moltes voltes, no he trobat cap manera justa de fer-ho. Es podria pensar en fer el promig de punts del jugador durant la partida, però això també pot dur a tripijocs per part d'alguns jugadors i fins i tot hi hauria alguns que farien anar mes lenta la partida per tal de que el seu promig variés segons els hi convingués, a més que faria que el càlcul de punts no fos simple, i en la simplicitat es troba la màxima transparència i comprensió. Així que s'ha decidit utilitzar la puntuació final, perquè es la que es sol utilitzar en els diversos llocs d'escac per correspondència. Es cert que això comporta riscos de tripijocs, però els jugadors que faran això normalment seran minoria, i per tant es potser la manera més neta, clara i comprensible de fer-ho.

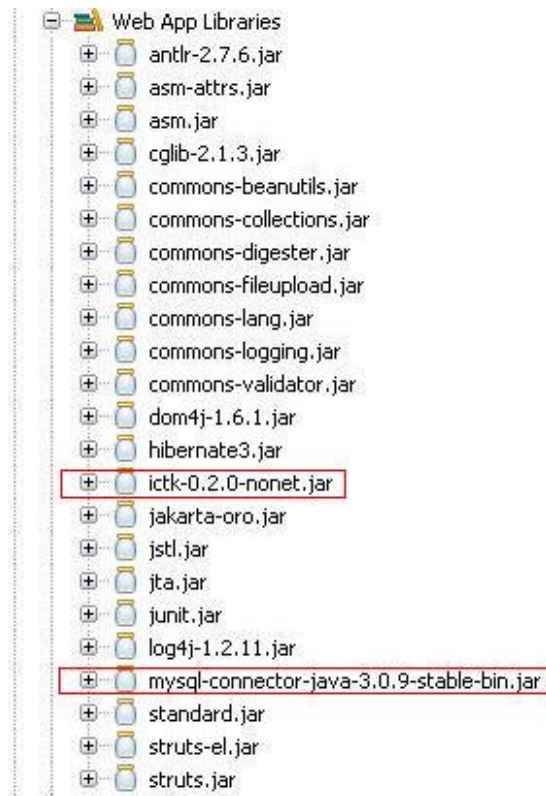
Cal veure també que als jugadors novells la puntuació els hi pot ballar molt i pot estar molt inflada o devaluada sobre tot a les primeres partides. Per tant, a més a més d'aquestes consideracions, cal dir que un jugador no novell tindrà els seus punts protegits i mai perdrà punts davant d'un jugador novell. Pot guanyar-ne, però mai perdre'n. D'aquesta manera també s'incentiva a jugadors amb una puntuació estable a que juguin amb jugadors nous al sistema, sense posar en perill la seva puntuació en enfrontar-se contra un jugador que pugui tenir una puntuació devaluada i més força real de la que diuen els seus punts.

Una altra consideració es troba en el fet de que sols es consideraran partides aptes per canviar puntuació aquelles que durin 10 o més moviments (5 per cada banda), per tal d'evitar que jugadors pactin entre ells fer partides ràpides per tal d'incrementar / decrementar artificialment la seva puntuació.

9 Annex B: Deploy del projecte

Per tal de fer funcionar l'aplicació es necessari desplegar-la en un servidor com ara Tomcat, i també crear la base de dades.

1. S'inclouen un fitxer sql, un anomenat dataBase.sql aquests fitxer s'han de introduir a la base de dades MySql mitjançant la comanda:
`mysql -uusuari -ppassword < dataBase.sql`
2. Això crearà una base de dades amb partides i jugadors per tal de que el sistema sigui provat correctament. Els jugadors creats s'anomenen admin1 fins a admin9 i tots tenen el password root
3. Un cop fet això caldrà modificar el fitxer hibernate.cfg.xml per tal de canviar l'usuari i contrasenya de MySql segons com es tingui configurat
4. No s'ha pogut, per falta de temps, entendre el funcionament d'ant i del build.xml. Per tant l'explicació de l'obtenció del war es farà a través d'Eclipse tal i com s'explica a continuació:
 - a. El primer es obrir Eclipse, la meva versió es Eclipse Europa 3.3.1.1, en anglés per tant l'explicació la faré a partir de la meva versió.
 - b. Crear un nou projecte buit, de tipus estàndard anomenant-lo ChessUoc, File – New – Project – ChessUoc – Finish.
 - c. Després anar a File – Import – Archive File – escollir la carpeta on tenim ChessUoc.zip, quan ens surti l'estructura de la carpeta, escollir l'Arrel, de manera que quedi tot seleccionat, i fer Finish, ens preguntarà si volem sobreescriure les dades del projecte i li diem que si a tot
 - d. Ens copiarà els arxius al nou projecte, ens aquests moments ens sortiran moltíssims errors, perquè el projecte no te les llibreries
 - e. Ara cal importar les llibreries ja que sols s'hi inclouen dos, mysqlconnector, que fa de pont entre Hibernate i la base de dades MySQL i la ictk, de l'internet chess toolkit. Per afegir les altres fem el següent:
 - i. Primer anem a Project – Properties – JavaBuildPath, li donem a Add external jar i naveguem fins a la instal·lació de TomCat 6 on escollim servlet-api.jar. Tot i que aquest pas potser es innecesari, perquè la llibreria ja es trobarà quan fem el deploy, però val la pena fer-lo, perquè així ens assegurarem que no hi ha més errors dels 4 que ha de haver (veure punt f)
 - ii. Les llibreries que utilitzo son les següents:



Imatge 23: Llibreries utilitzades

- iii. Totes pertanyen o bé a l'.struts 1.1 o bé a l'hibernate-3.2.4.sp1 (a part de les encuadrades en vermell) que s'hi inclouen.
 - iv. Ara el que cal fer es obrir les carpetes on es tinguin aquests dos frameworks i arrastar aquests arxius a Web/Web-Inf/Lib
 - v. Un cop fet això ens anem un altre cop a Project – Properties – Java-BuildPath, li donem a Add jar (no external) naveguem per la nostra aplicació fins a la carpeta Lib on afegim totes les llibreries.
 - vi. Un cop fet això ens adonarem que ens surten warnings indicant que aquests fitxers no seran exportats, aquí sols he aconseguit arreglar-ho d'una manera, anant warning per warning, amb el botó dret i donant-li a quick fix.
- f. Un cop feta aquesta tarea àrdua ens adonarem que encara queden 4 errors del fitxer table.jsp son errors normals, producte de que Eclipse pensa que em falten tags <td> però no es així en ser la pàgina dinàmica
 - g. Ens anem a File – Export – Web – WAR file escollim el Web Module ChesUoc, la destinació i fem click a Finish, això ens crearà el WAR

5. Ara queda fer el deploy al servidor, copiant aquest war a la carpeta webapps de Tomcat 6, i teclejar <http://localhost:8080/ChessUoc> a un navegador. Si surt la plana inicial, tot estarà correcte.
6. En cas de que no es pugui compilar, pel que sigui, s'inclourà una direcció d'algun lloc tipus rapidshare per tal de poder allotjar el war i que pugui ser baixat fàcilment davant la gaire bé segura impossibilitat de pujar-lo al campus virtual.

Treball de fi de carrera J2EE: Club d'escacs postal

Objectius

- Desenvolupar una aplicació que permeti a les persones que els hi agrada els escacs disputar les seves partides a través de la web
- Fer que els jugadors tinguin coneixement dels altres jugadors informant-los amb estadístiques dels punts forts i febles dels seus rivals
- Permetre crear fòrums on els jugadors puguin compartir la seva experiència amb els escacs i fer totes les preguntes que vulguin a altres jugadors, així com contestar les qüestions formulades per altres companys.
- Incentivar la competitivitat amb un sistema de punts que informi al jugador de quina es la seva força relativa respecte als altres jugadors, així com otorgant medalles en guanyar campionats.
- Millorar les aplicacions presents a la xarxa, agafant el més bó de cadascuna.

• Objectius

• Funcionalitats

• Actors

• Disseny de la BD

• Arquitectura

• Patrons de disseny

• MVC I

• MVC II

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• Hibernate

• Estructura fonts

• Estructura WAR

• Resultats

• Conclusions

• Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Funcionalitats

- Requeriments per un total de 74 cassos d'ús, dividits entre aquests tipus:
 - Identificació d'usuari (1)
 - Partides (1)
 - Torneigs (2)
 - Llibre d'apertures (2)
 - Missatges (3)
 - Administració (3)
 - Cerca de posicions (3)
 - Fòrums (4)
 - Moviments condicionals (4)
 - Equips (5)
- Casos d'ús dividits en cinc categories segons prioritat, indicades entre parèntesi

• Objectius

• Funcionalitats

• Actors

• Disseny de la BD

• Arquitectura

• Patrons de disseny

• MVC I

• MVC II

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• Hibernate

• Estructura fonts

• Estructura WAR

• Resultats

• Conclusions

• Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Actors

- **Usuari no registrat**, aquell que no s'ha donat d'alta al sistema
- **Usuari novell**, ha jugat 30 o menys partides amb puntuació
 - No pot crear torneigs
 - No pot incriure's a a cap equip
 - Partides que jugui contra no novells no susposaràn pèrdua de punts pel rival.
- **Usuari experimentat**, ha jugat entre 31 i 100 partides.
 - Pot incriure's a equips
 - No pot crear torneigs
 - No pot crear equips ni ser escollit capità
- **Usuari expert**, ha jugat 101 o més partides
 - No té cap mena de limitació
- **Capità**
 - Un usuari expert, tasques pròpies dels equips
- **Administrador**
 - Esborrat automàtic
 - Serà un usuari expert

• Objectius

• Funcionalitats

• **Actors**

• Disseny de la BD

• Arquitectura

• Patrons de disseny

• MVC I

• MVC II

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• Hibernate

• Estructura fonts

• Estructura WAR

• Resultats

• Conclusions

• Gràcies

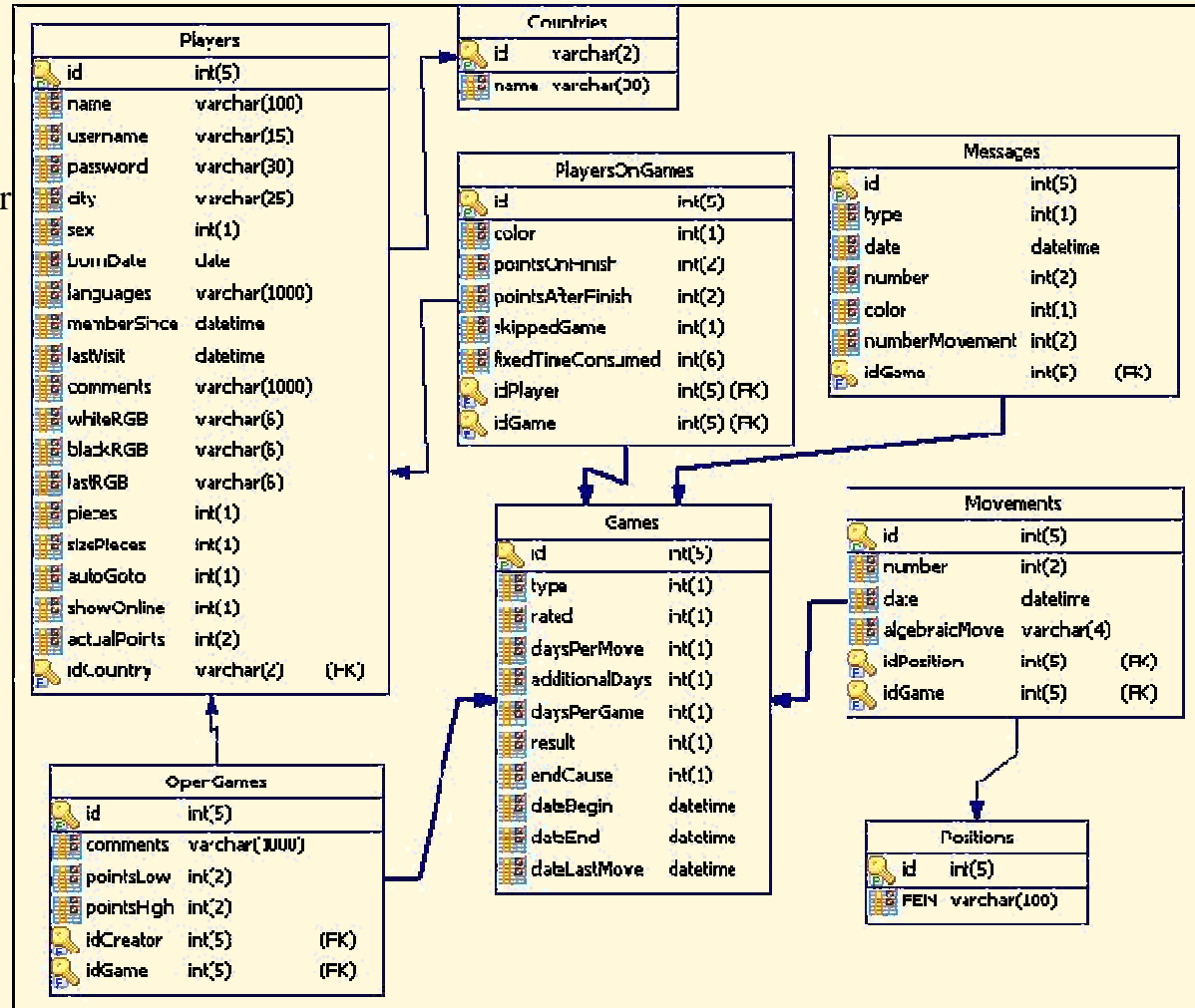
Treball de fi de carrera J2EE: Club d'escacs postal

Disseny de la Base de dades

•MySQL

•Dissenyada amb
Microlap DB Designer
for MySQL

•Crea scripts de
creació e inserció



- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- MVC I
- MVC II
- Composite view
- DAO I
- DAO II
- Implementació i patró Composite
- Struts
- Hibernate
- Estructura fonts
- Estructura WAR
- Resultats
- Conclusions
- Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Arquitectura

- Sistema es desenvoluparà sota la plataforma J2EE
- Client accedirà mitjançant l'explorador
- Contenedor de servlets Tomcat 6.0, per familiaritat
- El RAD utilitzat serà Eclipse
- Arquitectura dividida en capes, tal i com promou J2EE
 - **Client**, formada pel navegador
 - **Web**, patró MVC, formada per Servlets + JSP's
 - **Lògica de negoci**
 - Sun recomana EJB
 - Complicació excessiva per una web d'escacs
 - S'utilitzaran POJO (plain old java objects)
 - **Persistència**, patró DAO.

• Objectius

• Funcionalitats

• Actors

• Disseny de la BD

• **Arquitectura**

• Patrons de disseny

• MVC I

• MVC II

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• Hibernate

• Estructura fonts

• Estructura WAR

• Resultats

• Conclusions

• Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Patrons de disseny

- Patró de disseny: Solució de disseny acceptada com a correcta per la solució d'un determinat problema.
- Utilitzarem tres:
 - **MVC**: Model-View-Controller
 - **DAO**: Data Acces Object
 - **Composite View**

• Objectius

• Funcionalitats

• Actors

• Disseny de la BD

• Arquitectura

• **Patrons de disseny**

• MVC I

• MVC II

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• Hibernate

• Estructura fonts

• Estructura WAR

• Resultats

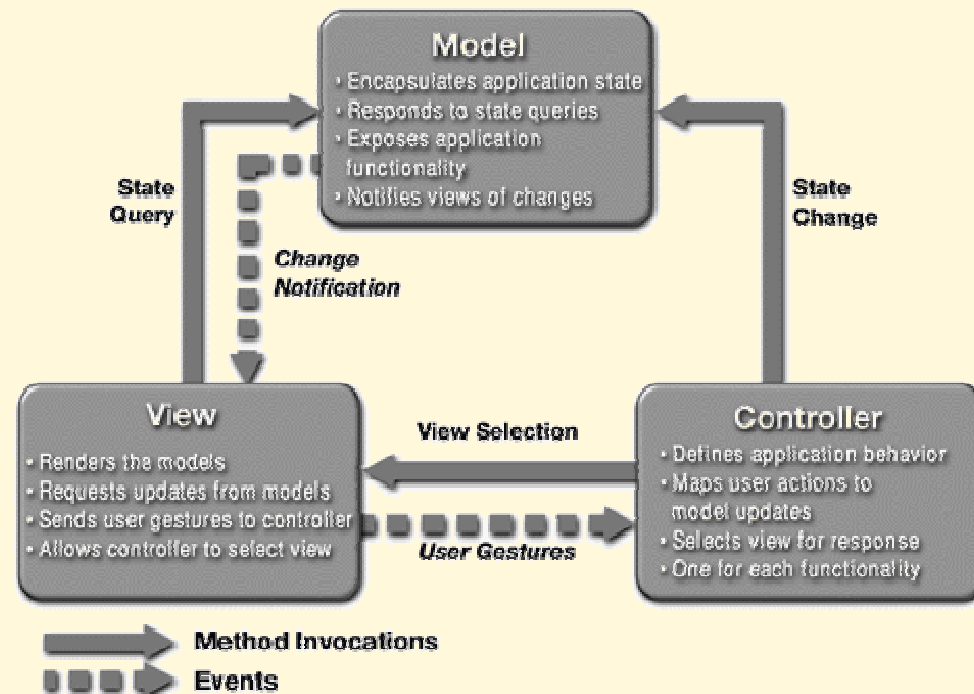
• Conclusions

• Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Model View Controller I

- Patró de disseny
- Separar les dades de l'aplicació de la interfície d'usuari



- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- **MVC I**
- MVC II
- Composite view
- DAO I
- DAO II
- Implementació i patró Composite
- Struts
- Hibernate
- Estructura fonts
- Estructura WAR
- Resultats
- Conclusions
- Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Model View Controller II

- **Model**
 - Dades pròpiament dites de l'aplicació
 - Funcionalitat d'aquesta
- **Controlador**
 - Respón a les accions dels usuaris
 - Interactua amb el model
 - Determina la vista a utilitzar
- **Vista**
 - Presentar informació físicament als usuaris
- **Funcionament**
 - L'usuari mitjançant la interfície gràfica genera petició
 - El controlador rep la petició i decideix com interactuar amb la lògica de negoci per portar a bon terme la petició
 - El controlador passa el control a la vista adequada
 - La vista pot accedir al model, no al revés
 - La vista mostra la informació a l'usuari

• Objectius

• Funcionalitats

• Actors

• Disseny de la BD

• Arquitectura

• Patrons de disseny

• MVC I

• **MVC II**

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• Hibernate

• Estructura fonts

• Estructura WAR

• Resultats

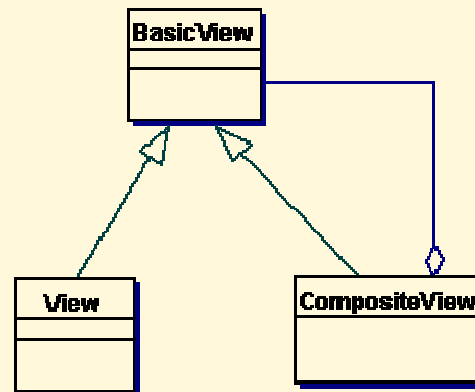
• Conclusions

• Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Composite view

- Patró de disseny
- Tractar les pàgines com un conjunt de subpàgines independents entre elles
- En lloc de veure la pàgina com una unitat, aquesta està formada per subconjunts reutilitzables
- Desenvolupament molt més simple i fàcil de mantenir



Treball de fi de carrera J2EE: Club d'escacs postal

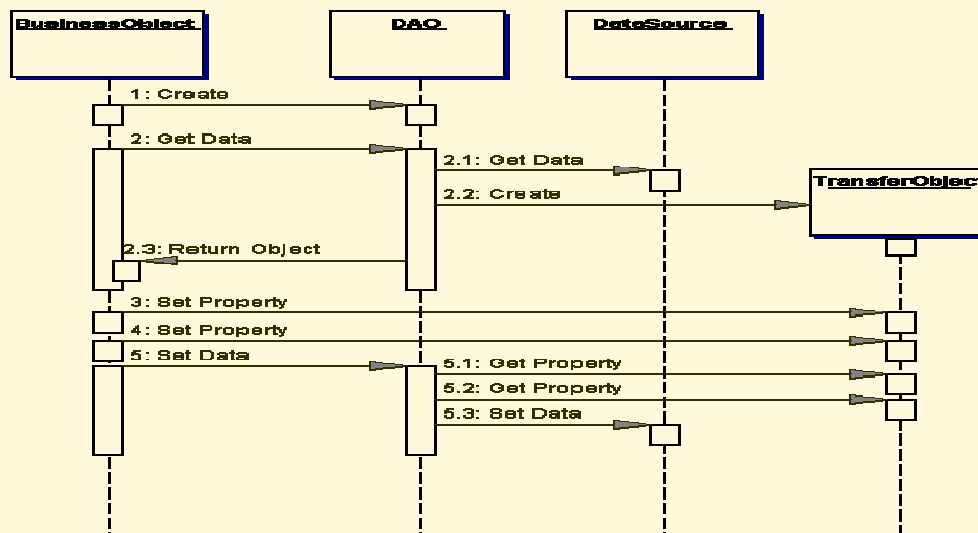
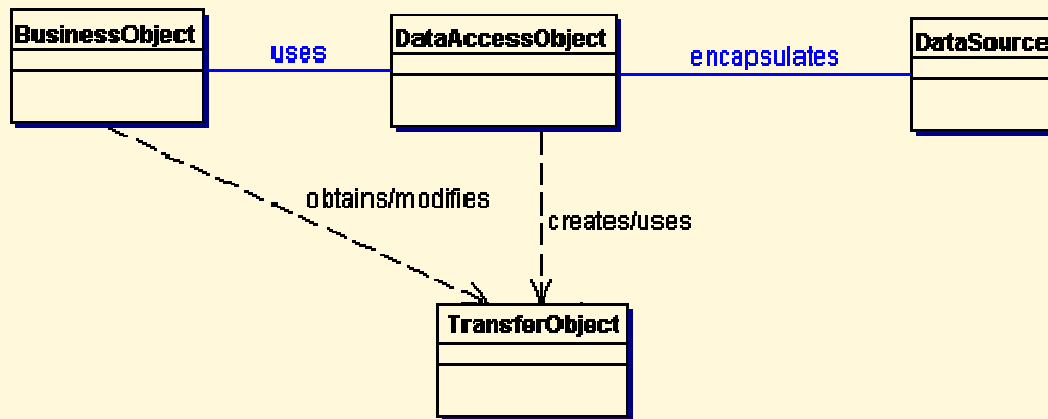
Data access object I

- Patró de disseny
- Serveix per encapsular i abstraure les dades pròpiament dites
- Format per quatre participants
 - **Business Object**
 - Encarregat de contenir la lògica de negoci de l'aplicació
 - L'objecte que necessita tenir l'accés a les dades
 - **Transfer Object**
 - Les dades pròpiament dites
 - **Data Access Object (DAO)**
 - Encapsula l'accés a les dades
 - Permet al Business object tenir un accés a les dades transparent
 - **DataSource**
 - Dades com a tals en el seu contingut físic
 - Fitxer, base de dades, etc...

- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- MVC I
- MVC II
- Composite view
- **DAO I**
- DAO II
- Implementació i patró Composite
- Struts
- Hibernate
- Estructura fonts
- Estructura WAR
- Resultats
- Conclusions
- Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Data access object II



- [Objectius](#)
- [Funcionalitats](#)
- [Actors](#)
- [Disseny de la BD](#)
- [Arquitectura](#)
- [Patrons de disseny](#)
- [MVC I](#)
- [MVC II](#)
- [Composite view](#)
- [DAO I](#)
- **DAO II**
- [Implementació i patró Composite](#)
- [Struts](#)
- [Hibernate](#)
- [Estructura fonts](#)
- [Estructura WAR](#)
- [Resultats](#)
- [Conclusions](#)
- [Gràcies](#)

Treball de fi de carrera J2EE: Club d'escacs postal

- [Objectius](#)
- [Funcionalitats](#)
- [Actors](#)
- [Disseny de la BD](#)
- [Arquitectura](#)
- [Patrons de disseny](#)
- [MVC I](#)
- [MVC II](#)
- [Composite view](#)
- [DAO I](#)
- [DAO II](#)
- **Implementació i patró Composite**
- [Struts](#)
- [Hibernate](#)
- [Estructura fonts](#)
- [Estructura WAR](#)
- [Resultats](#)
- [Conclusions](#)
- [Gràcies](#)

Implementació

- **Tomcat 6.0**, com a contenidor de Servlets
- **Eclipse Europa**, com a RAD
- **Log 4.J 1.2.11**, per fer el log
- S'ha utilitzat **Struts 1.1** per l'MVC
- **Hibernate 3.2.4.sp1** pel patró DAO
- **MySQL 5.0**, com a SGBD
- **MicroOlap DataBase Designer for MySQL**, per dissenyar la BD.
- L'objectiu es l'obtenció d'un fitxer WAR que pugui ser desplegat en un servidor com a exemple Tomcat
- **Internet Chess Toolkit**, llibreria open source d'escacs

Patró composite a la implementació

- S'implementa mitjançant l'estructura `<jsp:include>`
- Tota plana es un conjunt de pàgines que formen la plana
- Gaire bé totes comparteixen majoria de parts
 - Menu, peu, capçalera...
- Si volguèssim canviar el menu
 - Sols caldria canviar la pàgina jsp que conté el menu
 - Totes les altres quedarian modificades en importar el menu

Treball de fi de carrera J2EE: Club d'escacs postal

Jakarta Struts

- Eina per tal d'implementar el patró MVC
- Components:
 - **ActionServlet**
 - Processa la petició de l'usuari
 - Decideix la classe Action que l'utilitzarà
 - **Action**
 - Processa la sol·licitut interactuant amb el mode
 - Obté uns resultats
 - **ActionServlet**
 - Avisat que els resultats ja hi son
 - Decideix la plana JSP que mostrarà les dades
 - **La plana JSP**
 - Accedeix a les dades i les mostra
- Fitxer **struts-config.xml**, es com es configura tota la funcionalitat de l'ActionServlet

- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- MVC I
- MVC II
- Composite view
- DAO I
- DAO II
- Implementació i patró Composite
- **Struts**
- Hibernate
- Estructura fonts
- Estructura WAR
- Resultats
- Conclusions
- Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Hibernate

- Framework encarregat d'encapsular l'accés a les dades
- Cal fer un arxiu .hbm.xml per cada classe relacionada amb una taula
 - Per exemple player.hbm.xml
 - Fitxer conté el mapeig dels diversos caps de la base de dades a membres de la classe
 - Classe ha de tenir els get / set corresponents segons l'atribut
- Es necessari un fitxer de configuració hibernate.cfg.xml
 - L'hem d'indicar quins arxius .hbm.xml fem servir
 - Fa la connexió a la Base de dades amb l'usuari i contrasenya
- Funcionament
 - Es crea una sessió
 - S'obre una transacció
 - Es realitza l'operació (lectura o escriptura)
 - Si es escriptura cal cridar al mètode commit per fer canvis permanents.
 - Si hi ha problemes, cal fer rollback al capturar l'excepció
 - Es tanca la sessió

• Objectius

• Funcionalitats

• Actors

• Disseny de la BD

• Arquitectura

• Patrons de disseny

• MVC I

• MVC II

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• **Hibernate**

• Estructura fonts

• Estructura WAR

• Resultats

• Conclusions

• Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Estructura de les fonts

- Set paquets + arxiu de recursos:
 - **chess.business**
 - Classes de la lògica de negoci
 - No tenen atributs
 - Accedeixen a la capa de dades i retornen la informació a les classes control.ladores
 - **chess.business.exceptions**
 - Excepcions de chess.business
 - **chess.business.utils**
 - Classes utilitzades per accedir al framework Hibernate
 - **chess.controller**
 - Classes actions de l'struts
 - DispatchAction, permeten tots els mètodes en un fitxer
 - No accedeixen a la capa DAO
 - **chess.controller.utils**
 - Classes d'ajuda per les chess.controller
 - **chess.data**
 - Model de dades
 - Els objectes DTO del patró DAO
 - **chess.utils**
 - Objectes auxiliars
 - **resources**
 - application.properties, tot el text de l'aplicació

- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- MVC I
- MVC II
- Composite view
- DAO I
- DAO II
- Implementació i patró Composite
- Struts
- Hibernate
- **Estructura fonts**
- Estructura WAR
- Resultats
- Conclusions
- Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Estructura WAR

- **images**
 - **images/icons**
 - Icones que s'utilitzen a les taules
 - Per representar funcions
 - **images/icons/flags**
 - Banderes dels diferents països
 - **images/pieces**
 - Diferents tipus de peces i tamanys
- **js**, diversos javascript, entre ells el del menu
- **pages**, les planes jsp
 - Amb una estructura de carpetes per tal d'allotjar les diferents subplanes que formen les planes ordenadament
- **WEB-INF**, dos arxius importants:
 - **struts-config.xml**, del qual ja hem parlat
 - **validation.xml**
 - Serveix per fer la validació de les forms
 - Forms dinàmiques (DynaActionForm)
 - Carpeta Lib, amb lliberies de l'aplicació
 - Classes
 - Fitxers compilats de les fonts
 - Arxius de configuració d'Hibernate
 - Arxiu de configuració del sistema de log: log4j

- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- MVC I
- MVC II
- Composite view
- DAO I
- DAO II
- Implementació i patró Composite
- Struts
- Hibernate
- Estructura fonts
- **Estructura WAR**
- Resultats
- Conclusions
- Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Resultats

- No s'ha pogut realment complir amb la planificació
- S'havia pensat el desenvolupament de tots els casos d'us de nivell 1 i 2 i sols s'ha arribat als de nivell 1
- Cosa que de retruc fa que no es diposin de tots els perfils especificats durant l'anàlisi
- Tinc la intenció de portar el projecte a fase d'exploració
 - Tot i que caldrà canviar la versió de l'estructura, perquè aquesta es antiga
 - I també llegir més sobre Hibernate perquè segons tinc entès no es un sistema fiable en un entorn productiu
- Cal millorar apartats
 - El cas dels accents i els caràcters especials
 - Tant web com BD tenien UTF-8
 - Però no funciona adequadament
 - Potser degut a problema amb struts (versió vella)
 - Cal implementar la traducció que ja està feta però no es funcional

• Objectius

• Funcionalitats

• Actors

• Disseny de la BD

• Arquitectura

• Patrons de disseny

• MVC I

• MVC II

• Composite view

• DAO I

• DAO II

• Implementació i patró Composite

• Struts

• Hibernate

• Estructura fonts

• Estructura WAR

• **Resultats**

• Conclusions

• Gràcies

Treball de fi de carrera J2EE: Club d'escacs postal

Conclusions

- Tot i no haver aconseguit les fites marcades al principi del semestre valoro de forma positiva el projecte
- Tant Struts com Hibernate son eines molt potents que permeten un ràpid desenvolupament d'una aplicació web de mitjana complexitat com es aquesta
- Continuo amb ganes de continuar l'aprenentatge de aquestes dues tecnologies
- Desitjo continuar amb el projecte pel meu compte per assolir millor aquests coneixements

- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- MVC I
- MVC II
- Composite view
- DAO I
- DAO II
- Implementació i patró Composite
- Struts
- Hibernate
- Estructura fonts
- Estructura WAR
- Resultats
- **Conclusions**
- Gràcies

Vols jugar a escacs?

Alumne: José Antonio Seguro Castro

Consultor: Albert Grau Perise

Treball de fi de carrera J2EE: Club d'escacs postal

Moltes gràcies per la vostra atenció



José Antonio Seguro Castro

- Objectius
- Funcionalitats
- Actors
- Disseny de la BD
- Arquitectura
- Patrons de disseny
- MVC I
- MVC II
- Composite view
- DAO I
- DAO II
- Implementació i patró Composite
- Struts
- Hibernate
- Estructura fonts
- Estructura WAR
- Resultats
- Conclusions
- **Gràcies**