

Anàlisi. Disseny i implementació d'una utilitat de Workflow aplicat a una web de gestió d'un centre de Reclamacions i Queixes mitjançant J2EE.

Una solució amb Struts i EJBs.

Jaume Balagué Sans
ETIS

Joan-Vicent Orenge Serisuelo

14 de gener de 2008



Agraïments:

A la família, als amics i a tota la comunitat de la UOC ja que sense el seu suport no hauria arribat mai fins aquí.

Resum del Treball Final Carrera.

El treball final de carrera esta compost de l'anàlisi, disseny i implementació d'una utilitat de Workflow que dona servei a una web per gestionar les reclamacions i queixes d'una empresa. La part del Workflow s'ha pensat per ser totalment configurable a les particularitats d'un determinat aplicatiu i per donar servei a tantes aplicacions com faci falta simultàniament.

L'anàlisi i el disseny s'ha basat en la metodologia UML dins el paradigma de l'Orientació a Objectes juntament amb diversos patrons de disseny com pot ser el MVC que ha permès separar completament la lògica de negoci (Workflow) de la presentació (Web). Per les 2 parts: Workflow i web s'ha fet ús d'una arquitectura distribuïda.

La implementació s'ha realitzat mitjançant el llenguatge Java i utilitzant l'arquitectura J2EE per les facilitats que dona a l'hora de treballar amb aplicatius que necessiten de la xarxa per funcionar i donar servei.

Per facilitar el desenvolupament s'ha aprofitat la potencia del framework Struts per fer el paper de Controller dins el patró MVC que simplifica el desenvolupament de la capa de presentació.

Un dels aspectes mes importants ha estat aconseguir un model de dades que permeti que el Workflow sigui totalment configurable i que un canvi en els fluxos d'una queixa o una reclamació no impliqui haver de modificar res més que la base de dades.

1. Índex

Anàlisi. Disseny i implementació d'una utilitat de Workflow aplicat a una web de gestió d'un centre de Reclamacions i Queixes mitjançant J2EE.	1
Resum del Treball Final Carrera.	3
1. Índex.....	4
2. Introducció.....	5
2.1. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC	5
2.2. Objectius del TFC.	5
2.3. Enfocament i mètode seguit.	5
2.4. Planificació del projecte.	6
2.5. Productes obtinguts.	7
2.6. Descripció de la resta de capítols de la memòria.....	7
3. ESPECIFICACIÓ I ANÀLISI DELS REQUERIMENTS	8
3.1. Introducció.	8
3.2. Descripció del projecte.....	8
3.3. Anàlisi preliminar.....	8
3.4. Subsistemes	9
3.5. Casos d'us	11
3.5.1. Actors	11
3.5.2. Utilitat de Workflow.....	11
3.5.3. Aplicatiu de Reclamacions i Queixes	20
3.6. Entitats derivades de l'anàlisi	24
3.6.1. Diagrama de Persones	25
3.6.2. Diagrama d'Objecte de negoci	26
3.6.3. Diagrama d'Estats.....	26
3.6.4. Diagrama d'Accions	27
3.6.5. Llista d'entitats	28
3.7. Diagrames d'activitat	34
3.7.1. Utilitat de Workflow.....	34
3.7.2. Aplicatiu de Reclamacions i Queixes	41
3.8. Definició dels Workflows	45
3.8.1. Diagrama de flux de les Queixes	45
3.8.2. Diagrama de flux de les Reclamacions.....	46
3.9. Requisits del maquinari i programari.	47
3.9.1. Clients.....	47
3.9.2. Servidor	47
4. Disseny	48
4.1. Introducció.	48
4.2. Arquitectura.....	48
4.2.1. Utilitat de Workflow.....	49
4.2.2. Aplicatiu de Reclamacions i Queixes	57
4.3. Disseny de la persistència.....	63
4.3.1. Persistència de Persones	64
4.3.2. Persistència de l'Objecte de Negoci	64
4.3.3. Persistència d'Estats	65
4.3.4. Persistència d'Accions.....	65
5. Implementació	66
5.1. Requeriments de programari.	66
Eines de disseny.	66
Eines de desenvolupament.....	66
6. Valoració econòmica	67
7. Conclusions	67
8. Glossari.....	68
9. Bibliografia.....	69
10. Annex 1. Manual d'instal·lació	69

2. Introducció

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

El TFC sorgeix de la necessitat de desenvolupar un aplicatiu web basat en el paradigma de l'Orientació a Objectes i crear un sistema d'informació amb una arquitectura distribuïda. Es volia crear un aplicatiu robust, configurable i fàcilment ampliable. L'altra raó per l'elecció d'aquest projecte ha estat les ganes que tenia de reprendre i reciclar-me en la tecnologia Java que havia vist fa bastant temps i de la qual estava totalment desconnectat, llevat de les pràctiques i PACs realitzades a nivell d'assignatura de la UOC.

El punt de partida han estat: els coneixements previs sobre Java i els assolits durant els estudis d'enginyeria tècnica a la UOC, sobretot d'assignatures com Enginyeria de Programari, Bases de Dades, combinant-los i ampliant-los amb l'auto-aprenentatge: investigant i dissecionant aplicacions ja finalitzades, aprenent sobre els patrons de disseny, els frameworks com Struts, etc

L'aportació del TFC ha estat per una banda, aconseguir el repte de crear un projecte començant de zero i passar per totes les fases del seu cicle de vida: recollida de requeriments, anàlisi, disseny i implementació. Per altra banda, l'altra gran aportació ha estat posar-me al dia i reciclar-me en la tecnologia J2EE. Per fer-ho, ha estat necessari molt esforç, aprendre patrons, buscar, investigar i llegir manuals de frameworks per decidir si usar-los o descartar-los.

2.2. Objectius del TFC.

Els objectius de TFC han estat dos: realitzar l'anàlisi, disseny i implementació de la web i la utilitat de Workflow mitjançant la tecnologia J2EE tot posant en pràctica els coneixements adquirits ens assignatures com Enginyeria de programari o bases de dades.

L'altre objectiu ha estat reciclar-me de les tecnologies J2EE com poden ser JSPs, EJBs, JDBC, JNDI o frameworks com Struts2 ja que l'última vegada que vaig treballar a nivell professional en Java corria la versió 1.1.8. En aquest sentit, he après algunes de les noves característiques de l'arquitectura J2EE afegides des de la creació de la versió 2 coneguda com Java 2.

2.3. Enfocament i mètode seguit.

L'enfocament utilitzant durant la creació del TFC ha estat en una primera fase treballar la metodologia UML per l'anàlisi i disseny mitjançant la OO intentant detectar i aplicar els patrons adequats en cada moment. Durant aquest temps, també s'ha avançat en l'aprenentatge i posada al dia dels coneixements de la tecnologia J2EE i del framework Struts.

La segona fase correspon a la part del desenvolupament i la posada en pràctica dels coneixements tècnics adquirits anteriorment.

Les principals dificultats han estat en resoldre petits problemes pràctics que anaven sortint, sobretot relacionats amb Struts, ja que s'ha utilitzat la versió 2 i encara no hi ha molta documentació disponible. Altres problemes han estat decidir si usar o no frameworks, i avaluar la seva corba d'aprenentatge amb relació amb les facilitats i estalvi de temps que proporcionen per tal de prendre una decisió correcta.

2.4. Planificació del projecte.

Sobre la planificació inicial es van anar assolint les fites en el temps previst fins arribar a l'implementació on van començar els desajustaments, deguts sobretot a la poca informació i la manca d'exemples sobre els taglibs d'Struts2. Al final, una vegada fetes les 2 primeres pantalles, la resta es van poder finalitzar molt més ràpid del planificat.

El resum de la planificació final és la següent:

Especificació i anàlisi de requeriments: 24 dies

Disseny: 8 dies

Fase d'implementació: 48 dies

Testing: 14 dies

Memòria i presentació final: 14 dies

	Previsió inicial			Dates finals		
	Dies	Data d'inici	Data fi	Dies	Data d'inici	Data fi
Fase d'anàlisi funcional i disseny:						
Casos d'us dels aplicatius	8	28/09/2007	05/10/2007	8	28/09/2007	05/10/2007
Diagrames d'entitats	7	06/10/2007	12/10/2007	7	06/10/2007	12/10/2007
Diagrames d'activitat	7	13/10/2007	19/10/2007	7	13/10/2007	19/10/2007
Definició de la persistència	3	20/10/2007	22/10/2007	2	20/10/2007	21/10/2007
Disseny	5	23/10/2007	27/10/2007	6	22/10/2007	27/10/2007
Maqueta	2	28/10/2007	29/10/2007	2	28/10/2007	29/10/2007
Fase d'implementació						
Creació dels scripts i de la base de dades	4	30/10/2007	02/11/2007	2	30/10/2007	01/11/2007
Definir el Workflow de les queixes i reclamacions	2	03/11/2007	04/11/2007	3	02/11/2007	04/11/2007
Programació de la utilitat de Workflow						
DAO accés a la base de dades	11	05/11/2007	15/11/2007	11	05/11/2007	15/11/2007
Serveis disponibles	8	16/11/2007	23/11/2007	8	16/11/2007	23/11/2007
EJB de connectivitat amb les aplicacions	7	24/11/2007	30/11/2007	5	24/11/2007	28/11/2007
Web de reclamacions i queixes						
Configuració Struts2	4	01/12/2007	04/12/2007	3	29/12/2007	01/12/2007
Capa de comunicació amb el Workflow	7	05/12/2007	11/12/2007	2	02/12/2007	04/12/2007
Pantalla d'entrada (login)	1	12/12/2007	12/12/2007	4	04/12/2007	07/12/2007
Pantalla principal	5	13/12/2007	17/12/2007	6	08/12/2007	13/12/2007
Resta pantalles	15	18/12/2007	01/01/2008	4	14/12/2007	17/12/2007
Proves	5	02/01/2008	06/01/2008	14	18/12/2007	31/12/2007
Memòria	5	07/01/2008	10/01/2008	10	01/01/2008	10/01/2008
Presentació virtual	5	11/01/2008	14/01/2008	4	11/01/2008	14/01/2008

(*) En verd les dates d'entrega de les PACs i l'entrega final

2.5. Productes obtinguts.

El resultat del TFC ha estat un aplicatiu web i una utilitat de Workflow realitzats amb tecnologia J2EE.

Els elements que els componen son els següents:

- 2 fitxers de distribució .ear amb la web i el Workflow.
- Una llibreria amb els beans comuns en format .jar.
- Codi font dels projectes.
- Documentació en format javadoc.
- Fitxer .xml amb la configuració del datasource per connectar a la base de dades.
- Scripts de creació de la base de dades.
- Scripts de configuració del Workflow.

2.6. Descripció de la resta de capítols de la memòria.

En la resta de capítols de la memòria es veuran les fases d'anàlisi, disseny i implementació per les quals ha passat el desenvolupament d'aquest TFC.

3. ESPECIFICACIÓ I ANÀLISI DELS REQUERIMENTS

3.1. Introducció.

En aquest apartat es mostra el procés de creació d'un anàlisi funcional partint dels requisits del projecte i acabant en els diagrames de UML que serviran de base pel disseny tècnic. Per tant s'ha d'aconseguir saber que s'ha de fer i decidir com s'ha de fer.

A partir de la descripció del projecte que fa el client: Requeriments, sabem el que s'ha de fer i que queda fora del projecte o el que es el mateix: l'Abast del projecte.

També es definiran les diferents parts o subsistemes en que dividirem el projecte i una maqueta de com serà d'interfície d'usuari.

3.2. Descripció del projecte.

L'empresa Joguines AC és una empresa de mida mitjana dedicada a la fabricació i distribució de joguines electròniques. Per tal de donar un millor servei als seus distribuïdors i clients ha decidit realitzar pel Departament d'Atenció al Client una web per tal de poder automatitzar les incidències en forma de Queixes i Reclamacions que tenen els seus clients, empleats, distribuïdors o proveïdors. D'aquesta manera s'assoleix un triple objectiu:

- Donar garanties que totes les Reclamacions i Queixes seran avaluades.
- Agilitzar els processos.
- Fer estudis de qualitat amb les dades obtingudes.

El funcionament és el següent: Una vegada s'ha rebut una Queixa o Reclamació per qualsevol via (fax, mail, carta, etc) al Departament d'Atenció al Client, un empleat la llegeix i identifica al departament que se n'ha de fer càrrec, la mateixa persona la dona d'alta via web, i a partir d'aquí aniran intervenint els diferents empleats fins el moment que es dona una resposta a la persona que ha interposat la queixa o reclamació.

També es demana que futures reorganitzacions a nivell de departaments o de funcionament del Departament d'Atenció al Client, no afecti el funcionament de la web.

Per tant, l'objectiu és una aplicació distribuïda, altament configurable, que accedeixi a una base de dades centralitzada. A més s'ha de permetre que tots els empleats de l'empresa amb autorització hi tinguin accés, però la consulta i les accions sobre les Queixes i Reclamacions donades d'alta estiguin restringides a les persones que els corresponguin. Així es s'aconseguirà la confidencialitat necessària. Per casos especials s'han de poder permetre que certes persones tinguin un rol tipus administrador que en cas de tenir accés a una Queixa o Reclamació puguin realitzar totes les accions disponibles en cada moment, amb aquestes persones es garanteix que en cas que un dels empleats estigui de baixa o vacances, el flux no pari.

3.3. Anàlisi preliminar

En un primer cop d'ull es veu que el cicle de vida d'una Queixa i Reclamació es pot modelitzar en un diagrama de flux, per tant haurà de implementar una màquina d'estats.

D'altra banda la transició d'un estat a un altre, implica afegir empleats al cicle de vida de l'objecte, per tant, més a més de passar d'estat, haurà de buscar persones dintre l'organigrama de l'empresa per anar afegint-los a la Queixa o Reclamació.

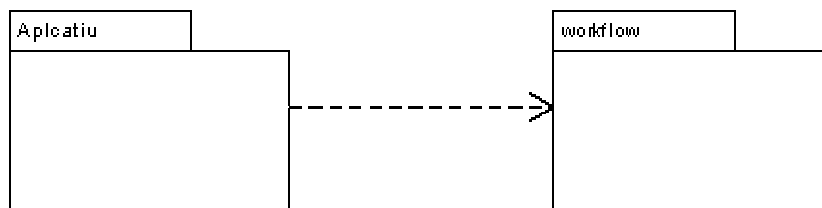
Per tant, la part principal de tota la lògica de negoci es basa en l'execució de les accions que impliquen una transició d'estats.

Així, es poden identificar dues parts ben diferenciades o subsistemes:

- Una web per visualitzar, afegir i modificar dades
- Una maquina d'estats ampliada que anomenarem: Utilitat de Workflow.

3.4. Subsistemes

A cada un dels 2 subsistemes se li ha fet correspondre un paquet tal i com es veu en el diagrama següent:



La utilitat de Workflow ha de proporcionar a l'aplicatiu de reclamacions/queixes tota la lògica de negoci que necessita per treballar. D'altra banda ha tenir com a virtut la de ser suficientment flexible i genèric per tal que permeti interactuar amb una gran diversitat d'objectes (els anomenaré Objectes de Negoci) i a més a més, que en cas de que els requeriments d'un determinat projecte no estiguin dins l'abast de la utilitat es pugui ampliar fàcilment mitjançant tècniques tan senzilles com l'herència de classes, etc.

Una de les característiques que faran especial el Workflow és que a més d'una maquina d'estats (que permet una sèrie d'accions sobre un objecte en funció de l'estat es que es troba en un determinat moment), és que filtrarà l'accés a les accions en funció de si la persona que hi vol accedir, té permisos d'execució o no sobre l'objecte i dels rols que desenvolupa dins del cicle de vida de l'objecte.

Dins dels conceptes del projecte cal fer la definició de "PARTICIPANT" i que es correspon a una persona que ha executat alguna acció sobre l'objecte de negoci amb el que s'està treballant o que se l'ha designat com a destinatari de l'objecte en algun moment del cicle de vida.

Les principals característiques de la utilitat de Workflow que es vol construir són les següents:

- Maquina d'estats.
- Filtratge automàtic de les accions disponibles en funció del rol de l'usuari i de l'estat actual de l'objecte.
- Actualització automàtica de les dades (si l'acció ho requereix).
- Càlcul automàtic de destinataris (Requerirà algun tipus de jerarquia entre els usuaris).
- Registre de tots els moviments realitzats durant el cicle de vida.

Les accions estaran disponibles en funció de:

- Estat amb que es troba l'objecte.
- Rols amb permisos d'execució.
- Participants implicats en accions anteriors.

L'execució d'una acció implica:

- Passar d'un estat inicial a un de final (poden ser els mateixos).
- Guardar les dades a la base de dades.
- Crear un moviment.
- Calcular i afegir els participants destinataris.

Altres serveis proporcionats pel Workflow:

- Recuperar un objecte de negoci de la base de dades.
- Fer cerques d'objectes de negoci dins de la base de dades a partir d'una sèrie de condicions definides per l'usuari. Això inclouria filtres de cerca, paginacions, etc.

L'aplicatiu de reclamacions/queixes serà una eina molt útil per a un departament d'Atenció al Client de l'empresa. Ha de permetre de fer un seguiment i una tramitació ràpida i eficaç de qualsevol queixa o reclamació que un client o treballador consideri oportuna.

A nivell visual ha de tenir les següents pantalles:

- **Pantalla d'identificació d'usuari:**

L'usuari s'haurà d'identificar i mitjançant una contrasenya, es valida que és qui diu ser.

- **Pantalla principal:**

Ha de mostrar la llista de tots els objectes de negoci de tipus reclamació/queixa en que l'usuari ha participat i que encara estiguin actius. La llista ha de disposar d'opcions d'ordenació com podrien ser per data d'alta, identificador de la reclamació, per estat, data de l'últim moviment, nom del participant que ha fet l'alta i per suposat d'una paginació.

Per cada registre de la llista s'ha de mostrar una icona de consulta i una altra per poder modificar l'objecte.

També es important que disposi d'un filtre per tal de facilitar la cerca de reclamacions; les opcions de filtratge han de ser les mateixes que per la ordenació.

En cas que l'usuari tingui permisos per executar altes, una icona o link cap a la pantalla d'altes.

- **Pantalla d'alta:**

Ha de mostrar els camps necessaris per poder crear un objecte queixa/reclamació. En aquest cas serien:

Nom de la persona que fa la reclamació

E-mail.

Telèfon.

Text de la queixa/reclamació.

Com a camps auto-completats per la pantalla hi hauria la data d'alta.

- **Pantalla de modificacions i consultes:**

Es mostrarien tots els camps de l'alta amb les dades actuals, un bloc amb tots els moviments realitzats sobre l'objecte.

En el cas de les modificacions un nou camp de text lliure per tal que es fessin comptar les al·legacions, resolucions, etc que l'usuari cregui oportú. En cap cas es deixa modificar el text/es introduïts amb anterioritat. A la part inferior, es mostraria un botó per cada una de les accions que te disponibles en aquell moment. Com que un usuari pot tenir diferents rols, s'agruparan de poder diferenciar les accions en funció del rol amb que les vol executar.

En cas que no tingui permisos per poder fer cap modificació es mostra la pantalla en mode consulta i a la part inferior on hi haurien d'haver els botons amb les accions que te disponibles, un petit text informant sobre la situació i a sota el botó de tornar a la pàgina principal.

- **Pantalla de resultat de l'acció:**

Una vegada executada una alta o una modificació es necessari informar a l'usuari del resultat de l'acció, per això s'ha de crear una pantalla senzilla on es mostra un missatge tipus: "L'acció X s'ha realitzat correctament" o "S'ha produït el següent error:".

En tots dos casos i a la part inferior, han de sortir els botons: "Acceptar" si tot ha anat correctament i que torna a la pantalla principal. O si hi han hagut errors "Tornar" i "Cancel·lar, que navegaran a la pantalla anterior i a la finestra principal respectivament.

A nivell de funcionament intern, l'aplicatiu web només té la responsabilitat de la capa visual i la de control (seguint l'arquitectura de les 3 capes MVC). Per qualsevol cosa que impliqui accedir a la base de dades, o executar una acció ha de fer-ho a través del Workflow.

Resumint es pot dir que la seva feina es empaquetar les dades de la pantalla, decidir que ha de fer amb elles el Workflow, enviar-les-hi, desempaquetar les noves dades rebudes com a resposta i mostrar-les a l'usuari.

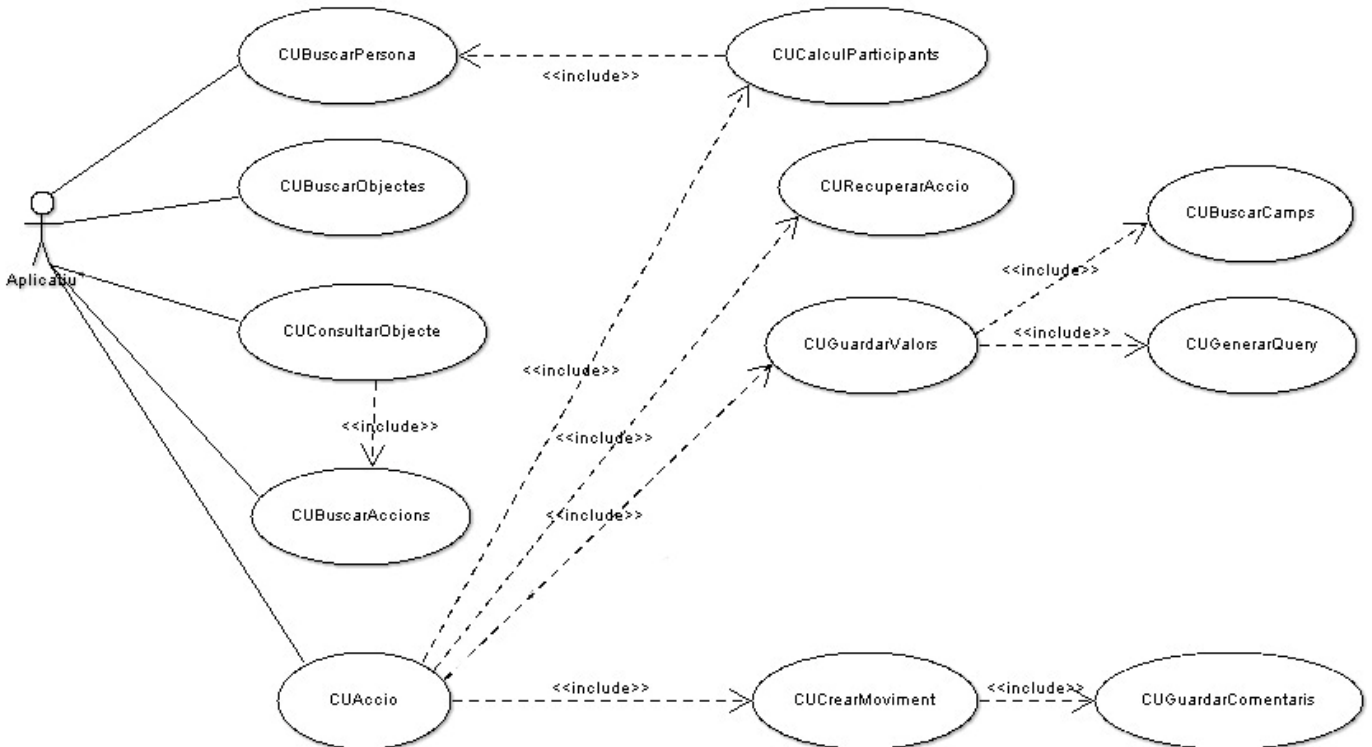
3.5. Casos d'us

3.5.1. Actors

S'han identificat 2 actors, l'aplicatiu de Reclamacions i Queixes te l'actor Usuari, es correspon a un empleat de l'empresa, que disposa d'accés a la xarxa. Per al Workflow s'ha identificat l'actor Aplicatiu que seria cada un dels aplicatius que el crida .

3.5.2. Utilitat de Workflow

Els casos d'us pel subsistema de Workflow és el següent:



3.5.2.1. CUBuscarObjectes

Descripció:

Mostrar tots els Objectes de Negoci actius d'un/s determinat/s tipus als que l'usuari pot accedir. Cal tenir en compte que per una correcta presentació de les dades: s'acota el nombre màxim de resultats retornats en cada crida, s'ordenen i filtren en funció dels paràmetres de la crida.

Permetre que certs rols que s'anomenaran genèrics tinguin accés a tots els objectes independentment de si son participants.

Actors:

Aplicatiu

Precondicions:

Es reben les següents dades:

- Persona que executa la cerca.
- Llista de tipus a cercar.
- Llista de camps que es volen recuperar.
- Llista de dades en format camp/valor amb els que es vol filtrar la cerca. (Si es buit significa que no hi ha filtre).
- Per la paginació:
 - Nombre total de resultats que es volen mostrar a la vegada: n.
 - Pagina(per defecte 1). Si el seu valor es per exemple 3 i n val 10, es retornen el registres del 20 al 30.
 - Camp sobre el que es vol fer l'ordenació i sentit de la mateixa (ascendent o descendent).Si està buit s'ordena per data de creació.

Postcondicions

Es retorna:

- La llista ordenada de n objectes resultants.
- Nombre total de resultats. (Necessari per la paginació)

Flux normal:

1. Recuperar els rols de la persona.
2. Si no te cap rol genèric y per cada un dels tipus de la llista, generar una "query" dinàmica que tingui en compte els següents factors:
 - Que la Persona que executa la cerca. sigui un participant de l'objecte. S'ha de permetre .
 - Agregar a la SELECT els camps a recuperar i la part corresponent de la paginació.
 - Agregar a la part de WHERE els valors del filtre (si n'hi han) i la part corresponent de la paginació.
 - Agregar a la part del ORDER BY el criteri d'ordenació; si no existeix ordenar per data de creació de l'objecte de negoci.
3. Si te més d'un tipus de Objecte de Negoci, unir les "queries" i afegir-hi la part del ORDER BY al criteri d'ordenació; si no existeix ordenar per data de creació de l'objecte de negoci.
4. Executar la sentència i retornar n resultats en funció de la paginació.

Flux alternatiu:

- 2.a) Si te un rol de tipus genèric activat no tenir en compte la part en que es necessita que sigui Participant de l'objecte.
- 3.a) Si només hi ha un tipus, no es necessari aquest pas.

3.5.2.2. CUAccio

Descripció:

Executa una acció amb totes les subtasques que implica.

Actors:

Aplicatiu

Casos d'ús relacionats:

CUCalculParticipants, CUREcuperarAcció, CUGuardarValors,
CUBloquejarObejcte, CUCrearMoviment.

Precondicions:

Es reben com a paràmetres:

- Codi de l' ACCIO a executar.
- Dades de OBJECTE DE NEGOCI amb el que treballem
- Persona que planteja la Queixa o Reclamacio (Només en l'alta)
- Usuari que l'executa.

Postcondicions

S'ha executat una Acció amb totes les subtasques corresponents o s'ha retornat un missatge d'error.

Si durant l'execució s'ha produït un error s'han de desfer tots els canvis de la base de dades i retornar una descripció de l'error i "fals", si tot ha anat be ha de retornar un "cert".

Flux normal:

Recupera les dades de configuració de l'Acció de la base de dades segons el cas d'ús CUREcuperarAccio.

A partir de la configuració ha de fer el següent:

- Si ha de persistir la informació de l'Objecte de negoci, cridar al cas d'ús CUGuardarValors..
- Generar un moviment mitjançant l'execució del cas d'ús CUCrearMoviment, Cal tenir en compte que sempre es generarà moviment, però només serà visible si la configuració de l'acció ho diu explícitament.
- Buscar els Rols i calcular les Persones a les que afegir al cicle de vida del Objecte de negoci fent una crida al cas d'ús CUCalculParticipants.
- E cas que tot l'anterior hagi anat correctament és modifica l'estat de l'Objecte de negoci pel que correspongui segons la configuració.
- Borrar el registre de bloqueig.

Flux alternatiu:

En cas de ser una alta i després d'haver creat el moviment, afegir la persona que ha interposat la queixa o reclamació com a participant.

3.5.2.3. CUREcuperarAccio

Descripció:

S'inicia quan rep del cas d'ús CUAccio un identificador d'acció, un tipus d'Objecte de Negoci i recupera de la base de dades tota la informació de configuració de la mateixa.

Actors:

Precondicions:

Existeix dins la base de dades l'Acció enviada com a paràmetre del cas d'us CUAcció per al tipus d'objecte amb que s'està treballant.

Postcondicions:

Es disposa de tota la configuració referent a una Acció.

Flux normal:

Es comprova si hi ha un registre a l'entitat Bloqueig, si no n'hi ha cap, se li afegeix un registre amb l'hora del sistema. Si hi ha un registre es comprova si entre l'hora actual i l'hora del camp difereixen un cert temps, si es així es considera que ja ha passat el bloqueig, sinó es retorna error.

Recupera la configuració d'una Acció de la base de dades, concretament interessa:

- Estat final de l'Objecte després de l'execució.
- Valors de l'Objecte de negoci que ha de guardar a la base de dades.
- Si genera moviment.
- Participants a afegir.

Flux alternatiu:

Si no es troba cap configuració amb el codi d'Acció i tipus desitjat es retorna un error.

3.5.2.4. CUGuardarValors

Descripció:

A partir dels atributs d'un Objecte de Negoci i la llista de camps a guardar rebuts de la configuració d'una Acció, es genera una query dinàmica per emmagatzemar tots els valors de la llista dins la base de dades.

El cas d'us es vàlid tant per crear un nou Objecte de Negoci com per modificar-lo. En funció de si l'Objecte te informat o no l'atribut que l'identifica inequívocament, es genera un UPDATE o INSERT respectivament.

Per tal de poder guardar la resta de valors no genèrics, propis de cada aplicatiu, s'ha de crear una taula de tipus metadàtica que emmagatzemarà els valors identificats pel parell: identificador de l'Objecte de Negoci/Nom del camp.

Actors:

Casos d'ús relacionats:

CUBuscarCamps, CUGenerarQuery

Precondicions:

Es rep del cas d'us CUAcció l'identificador de l'Acció i un Objecte de Negoci amb dades per persistir.

Si un camp es considera obligatori, ha de venir informat.

Postcondicions:

S'han gravat les dades a la base de dades i en cas de ser una alta, es genera un identificador únic per l'Objecte de Negoci.

Flux normal:

1. Recuperar la llista de camps que s'ha de persistir dins la base de dades segons la configuració de l'Acció, per realitzar aquesta tasca la deleguem al cas d'ús CUBuscarCamps.
2. Si la llista de camps, fa referència a més d'una taula, la partim en subllistes perquè cada vegada només volem generar i executar una query.
3. Per cada una de les subllistes (que es correspon amb els camps d'una única taula) cridar al cas d'ús CUGeneraQuery.
 - Si la taula a la que fa referència la subllista es la de Objecte de Negoci, i aquest no te informat el valor que l'identifica, s'inicialitza amb el valor retornat.

Flux alternatiu:

1.a Si la llista de camps es buida, sortim del cas d'ús.

Si la crida al cas d'ús GUGeneraQuery retorna error, es retorna un error.

2.a Si la subllista es correspon a la taula metadàtica de la que s'ha parlat anteriorment, llavors es considera cada element de la subllista com una subllista en si mateixa a la que se li afegeix l'identificador de l'Objecte de Negoci.

Per exemple si tenim la subllista de la taula metadàtica amb els elements: (numero catàleg, any fabricació, centre venda i codi venedor); es creen 4 subllistes amb els elements: (numero catàleg, identificador de l'Objecte de Negoci), (any fabricació, identificador de l'Objecte de Negoci), (centre venda, identificador de l'Objecte de Negoci) i (codi venedor, identificador de l'Objecte de Negoci). Una vegada fet aquest pas es borra la subllista original.

3.5.2.5. CUCrearMoviment

Descripció:

Genera un registre a la taula de Moviments per un determinat Objecte de Negoci amb l'Acció que s'ha executat i la persona que ho ha fet.

Els moviments han de venir numerats de manera ascendent ja que pot ser necessari recuperar l'històric d'un Objecte de Negoci.

Actors:**Casos d'ús relacionats:**

CUGuardarComentaris

Precondicions:

Es rep del cas d'ús CUAcció l'identificador de l'Objecte de Negoci, la configuració de l'Acció i el identificador de la persona que l'ha executat

Postcondicions

S'ha creat un registre dins l'entitat Moviment i es retorna el seu numero.

Flux normal:

Si no hi ha cap altre moviment a la base de dades relacionat amb l'Objecte de Negoci actual es genera un nou registres i per ser el primer es numera com a 0.

Si hi han altres moviments relacionats amb l'Objecte de Negoci, es mira el nombre de resultats, i es crea el nou moviment numerat com a nombre resultats+1.

Flux alternatiu:

3.5.2.6. CUCalculParticipants

Descripció:

Calcula els destinataris d'un Objecte de Negoci a partir de la configuració d'una acció.

Actors:

Precondicions:

Es rep del cas d'us CUAcció la configuració de l'acció, l'identificador de l'objecte de negoci, la persona que ha executat l'acció i un departament de l'empresa. En el cas de l'alta i opcionalment, les dades de la Persona que ha interposat la Queixa o Reclamació.

Postcondicions

S'ha afegit com a màxim un participant una de les persones que desenvolupa un dels rols que segons la configuració de l'Acció s'han d'afegir com a participants.

Fluxe normal:

1. Afegir com a participant la persona que ha executat l'Acció.
2. Si la Persona que ha interposat la Queixa està informada:
 - 2.1. Si està a la base de dades i és empleada se l'afegir com a participant.
 - 2.2. Si no està a la base de dades significa que és externa a l'empresa i se la dona d'alta, finalment se l'afegir com a participant.
3. Es recupera la llista de rols que s'ha d'afegir com a participants.
4. De manera recursiva es van recuperant les persones que són caps de qui ha executat l'Acció. Per cada una d'elles:
 - 4.1. Es recupera els seus rols
 - 4.2. Si un dels seus rols està a la llista del pas 1, s'afegir com a participant i es treu el rol de la llista
 - 4.3. Es busca el seu cap

Fluxe alternatiu:

2.a) si no ens arriba la persona que ha executat l'Acció, es busca dins la llista de rols el que té l'identificador més petit ja que indica el rol jeràrquicament més baix. Es busca dins del departament una persona amb aquest rol i a partir d'aquí s'inicia la cerca jeràrquica.

Si no hi ha ningú amb el rol més baix, es fa el mateix amb el segon més baix i així successivament.

2.3.a) Si s'arriba a una persona que no té cap, s'atura la cerca recursiva.

3.5.2.7. CUBuscarCamps

Descripció:

Recupera la llista de camps que s'han de guardar al executar una Acció. Com que la llista pot fer referència a diverses taules, ha de venir agrupada per taules i aquestes ordenades en funció de la seqüència que es desitja en segons la configuració de l'Acció (Si es una alta, primer s'ha de fer l'INSERT de la taula que guardar l'Objecte de Negoci i recuperar el seu identificador, que el de les taules auxiliars que podria tenir).

A més a més de retornar quins són els camps, també ha de retornar el com s'han de guardar.

Actors:

CUGuardarValors

Precondicions:

Es rep del cas d'us CUGuardarValors l'Acció que s'està executant i el tipus d'Objecte de Negoci amb el que es treballa.

Postcondicions:

Retorna una llista de camps o un null si no hi ha cap camp a guardar.

Flux normal:

Accedir a la taula que informa dels camps a persistir dins la base de dades al executar l'acció i per cada una d'ells, recuperar la configuració de com es guarda (si es obligatori, clau primària, tipus de camp)

Flux alternatiu:

Si un camp definit com obligatori no ve informat es genera un error.

3.5.2.8. CUGenerarQuery

Descripció:

A partir dels atributs d'un Objecte de Negoci i la llista de camps a guardar generada pel cas d'us CUBuscarCamps i tractada previament per CUGuardarValors, es genera una query dinàmica per emmagatzemar tots els valors de la llista dins la base de dades i s'executa.

El cas d'us es vàlid tant per crear un nou Objecte de Negoci com per modificar-lo. En funció de si l'Objecte te informat o no l'atribut que l'identifica inequívocament, es genera un UPDATE o INSERT respectivament.

Actors:

CUGuardarValors

Precondicions:

Es rep una llista no buida amb la configuració de camps a persistir a la base de dades i un Objecte de Negoci amb dades.

Si un camp es considera obligatori, ha de venir informat.

Postcondicions

S'han gravat les dades a la base de dades i en cas de ser una alta, es genera un identificador únic per l'Objecte de Negoci.

Flux normal:

Per cada element de la llista de camps a guardar :

- Si l'Objecte de Negoci rebut, te informat el camp que l'identifica inequívocament, es genera un INSERT, sinó un UPDATE.
- Recórrer la llista de camps a persistir a la base de dades i per cada camp:
 - Si es genera un INSERT:
 - Afegir-lo a la llista de camps de la query.
 - Recuperar l'atribut corresponent de l'Objecte de negoci i afegir-lo a la part dels valors de la query.
 - Generar un identificador únic per l'Objecte de negoci i inicialitzar-lo dins la classe.
 - Si es genera un UPATE:
 - Si el valor no es clau primària:
 - Afegir-lo a la llista de camps de la query.
 - Recuperar l'atribut corresponent de l'Objecte de negoci i afegir-lo a la part dels valors de la query.
 - Si es clau primària, afegir-lo a la part del WHERE.

Flux alternatiu:

Si un camp definit com obligatori o es clau primària(en cas de ser un UPDATE) no ve informat es genera un error.

3.5.2.9. CUGuardarComentarios

Descripció:

S'associen a un Moviment una llista de Comentarios que s'ha d'associar a l'Objecte de Negoci, els Comentarios en cap moment es modifiquen sinó que sempre es van afegint i numerant seqüencialment.

El cas d'us va associat a la creació d'un moviment i no al cas d'us responsable de guardar valors perquè es necessari saber en quin moment del cicle de vida s'ha creat i qui ha fet el comentari i això només ho podem saber si s'associa a un Moviment.

Els Comentarios d'un Objecte de Negoci aniran numerats de 0 a n.

Actors:

CUMoviment

Precondicions:

Es rep del cas d'us CUAcció el camp identificador d'un Objecte de Negoci informat., una llista no buida de comentaris encapsulada dins els seus atributs i un número de Moviment.

Postcondicions:

S'han gravat les dades a la base de dades i en cas de ser una alta, es genera un identificador únic per l'Objecte de Negoci.

Flux normal:

El primer pas es recuperar un número seqüencial de l'últim comentari que s'ha afegit a l'Objecte de Negoci amb el que s'està treballant. Si no hi havia comentaris anteriors es comença pel zero.

Per cada un dels Comentarios que venen encapsulats dins la llista, guardar-lo i incrementar el numerador pel pròxim comentari de la llista.

Flux alternatiu:

Si un camp definit com obligatori no ve informat es genera un error.

3.5.2.10. CUBuscarAccions

Descripció:

Recupera totes les Accions disponibles per un determinat Objecte de Negoci en funció de l'estat actual i el rol de la persona connectada a l'aplicatiu.

En el cas de voler saber si l'usuari té disponible l'Acció d'Alta, ens trobem amb la incongruència de que no tenim Objecte de Negoci i per tant no tenim un estat sobre el que buscar Accions. Aquesta situació es resol de la següent manera: Es considera una estat "inicial" amb un codi constant, per exemple 0 i que per convenció no serà usat per cap altre Estat. Per tant l'Acció "Alta" serà una acció que surt sempre d'un Estat "inicial" amb codi 0 cap a un altre Estat de codi >0.

Actors:

Aplicatiu

Precondicions:

La persona connectada a l'aplicatiu té rols i l'Objecte de Negoci té informat l'estat i el tipus.

En el cas de l'Alta es un Objecte de Negoci buit amb l'atribut estat amb valor 0 i el tipus amb el valor corresponent.

Postcondicions:

Llista amb les accions disponibles o nul si no n'hi ha cap.

Flux normal:

1. Es busquen els Rols que desenvolupa en l'Aplicatiu l'usuari connectat.
2. Es recuperen totes les Accions disponibles que tenen com a estat de sortida el del Objecte de Negoci enviat com a paràmetre i del mateix tipus.
3. Es comprova si te permisos d'execució per algun dels rols de l'usuari, si es així, s'afegeix l'Acció a la llista de retorn.

Flux alternatiu:

Si algun dels rols te activat l'atribut de genèric se li retornen totes les Accions encapsulades dins la llista i no es realitza el pas 3.

3.5.2.11. CUConsultarObjecte

Descripció:

Recupera tots les valors d'un Objecte de Negoci determinat

Actors:

Aplicatiu

Casos d'ús relacionats:

CUBuscarAccions.

Precondicions:

Es rep un identificador d'Objecte de Negoci i ha d'existir a la base de dades.

Postcondicions:

Es retorna un Objecte de Negoci no buit amb com a mínim, un Moviment i un Participant

Flux normal:

Es rep un identificador d'un Objecte de Negoci amb el seu tipus corresponent i s'accedeix a la base de dades per recuperar:

- Dades genèriques de l'Objecte que serviran per crear l'Objecte de Negoci que es retorna..
- Dades específiques en funció del tipus que es van afegint als seus corresponents atributs.
- La llista de Moviments en ordenada pel seu ordinal de major a menor amb el participant que ha executat l'Acció que l'ha creat (codi i dades de la persona). Només es recuperen els que tenen l'atribut generaMoviment activat dins de la configuració de l'Acció.
- Llista de Participants associats a l'Objecte de negoci ordenats alfabèticament.

Flux alternatiu:

3.5.2.12. CUBuscarPersona

Descripció:

A partir d'un codi de persona o de un usuari/contrasenya, recupera de la base de dades la informació relacionada amb una persona juntament amb tots els rols que pot desenvolupar per aquell tipus d'Objecte de Negoci.

Actors:

Precondicions:

El codi de persona o el parell usuari/contrasenya no es buit.

Postcondicions:

Es retorna un objecte de tipus Persona amb com a mínim un rol.

Flux normal:

Accedeix a la taula que emmagatzema les dades personals i crea un objecte de tipus Persona amb informació.

Va a la taula que relaciona les persones, rols i tipus d'aplicatiu, per recuperar tots els rols que pot desenvolupar, ho encapsula en una llista i ho afegeix a l'objecte de tipus persona. Les dades que es necessiten son: codi, nom del rol i si es genèric o no.

Flux alternatiu:

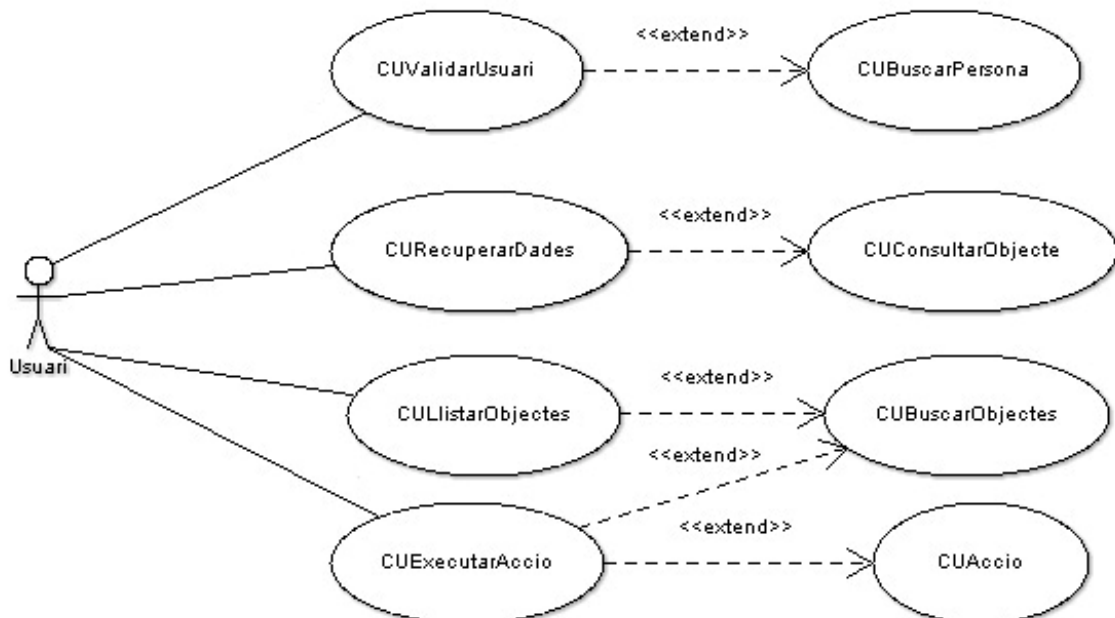
Si al Persona no te cap rol o no usuari/contrasenya no son correctes, es retorna un error.

3.5.3. Aplicatiu de Reclamacions i Queixes

L'aplicatiu d'aquest anàlisi en concret treballa amb 2 tipus d'Objectes de Negoci diferents: Reclamació i Queixa, per tant per cada un tindrem una configuració diferent: un diagrama d'estat propi, potser diferents rols, etc.

Degut a la flexibilitat que proporciona el Workflow fa que aquesta part de l'anàlisi pugui ser totalment reutilitzable per un altre aplicatiu que també el faci anar. Segurament l'únic canvi que s'hauria de fer seria a nivell d'interfície d'usuari ja que cada tipus d'Objecte de Negoci té els seus propis atributs i les pantalles serien diferents.

El diagrama de casos d'us de l'aplicatiu i la seva relació amb els del Workflow és el següent:



3.5.3.1. CUValidarUsuari

Descripció:

S'encarrega d'identificar i validar l'usuari que accedeix a l'aplicatiu.

Actors:

Usuari

Precondicions:

L'usuari existeix en la base de dades.

Postcondicions:

L'aplicatiu disposa de tota la informació necessària de l'usuari.

Flux normal:

L'aplicatiu mostra una pantalla on l'usuari ha d'introduir el seu codi de persona i un contrasenya secreta. Amb aquestes 2 dades, l'aplicatiu les envia al Workflow, aquest li retorna les dades de l'usuari mitjançant el cas d'us CUBuscarPersona i les guarda en memòria. Finalment enllaça amb el cas d'us CULlistarObjectes per tal de mostrar la pantalla principal.

Flux alternatiu:

Si l'usuari introdueix les dades incorrectament el Workflow retorna error i demana de nou introduir les dades.

3.5.3.2. CULlistarObjectes

Descripció:

Recupera la llista d'Objectes de Negoci als que te accés l'usuari, en funció del tipus d'Objecte de Negoci amb els que treballa l'aplicatiu i el seu historial, tant sigui per consulta com per realitzar accions.

Actors:

Usuari

Precondicions:

L'usuari s'ha validat al sistema.

Postcondicions:

Es retornen n Objectes de Negoci ordenats adequadament i el nombre total de resultats que s'ha trobat a la base de dades.

Flux normal:

Crida al cas d'us CUBuscarObjectes del Workflow enviant-li:

- Les dades de la persona que s'ha validat,.
- La llista de camps que es mostren a la pantalla principal.
- Un filtre buit la primera vegada i en successives crides si l'usuari ho desitja amb les dades que ha introduït.
- Per la paginació:
 - Nombre parcial n de registres a mostrar a la pantalla
 - Pagina en la que estem (la primera vegada 0).
 - Camp sobre el que es vol fer l'ordenació i sentit de la mateixa (ascendent o descendent). La primera vegada per identificador de l'Objecte de Negoci en sentit descendent.

i ens retorna una llista amb els n Objectes de negoci que es mostraran en forma de taula juntament amb el nombre total de resultats(A partir d'aquest nombre es construeix la paginació).

Si l'usuari no troba l'Objecte de Negoci desitjat pot activar els filtres, canviar l'ordenació o canviar la pagina mitjançant la paginació, en tots aquest casos es torna a executar el pas anterior de cridar al CUBuscarObjectes amb els valor modificats per l'usuari.

Paralel-lament i només la primera vegada que entra, després d'executar el CUValidarUsuari, per cada un dels tipus d'Objecte de Negoci amb que treballa l'aplicatiu crida al Workflow mitjançant el CUBuscarAccions per saber si l'usuari te permisos per executar l'acció d'alta. Només que disposi de permisos per un sol tipus ja se li mostra un link per accedir a aquesta acció.

Flux alternatiu:

Si l'usuari té algun rol genèric se li permet accedir a Objectes de Negoci ja finalitzats, de manera que tot i que en la primera cerca només es mostren els "actius", mitjançant els filtres se li dona la opció de seleccionar l'estat finalitzat, cosa que a la resta d'usuaris només se'ls hi mostres la resta d'estats.

3.5.3.3. CURecuperarDades

Descripció:

Recupera de la base de dades tota la informació d'un Objecte de Negoci i el mostra en una pantalla, en funció de si l'usuari vol operar amb ell o només consultar-lo, es mostra en format lectura-escritura o només lectura respectivament.

Actors:

Usuari

Precondicions:

Es disposa d'un identificador d'Objecte de Negoci vàlid.

Postcondicions:

Es visualitzen totes les dades de l'Objecte de Negoci

Flux normal:

L'usuari ha seleccionat la icona de consulta o la de operar amb un Objecte de Negoci en concret.

L'Aplicatiu recull l'identificador únic d'Objecte de Negoci i crida al cas d'us CUConsultarObjecte per recuperar-ne tots els seus camps.

Si l'usuari el vol consultar es mostra una pantalla amb les dades de l'Objecte de Negoci en mode lectura, si pel contrari ho ha fet per realitzar alguna Acció, es mostraran en mode lectura-escritura amb la restricció que els camps de tipus comentari anteriors no es poden modificar.

En el cas de voler operar , a la part inferior de la pantalla es mostraran totes les accions disponibles per aquell Objecte, cosa que implica cridar al cas d'us del Workflow CUBuscarAccions.

Flux alternatiu:

Si cas d'us CUBuscarAccions no retorna cap Acció disponible es mostra la pantalla en mode lectura.

3.5.3.4. CUExecutarAccio

Descripció:

L'usuari ha accedit a una pantalla d'alta o modificacions, ha introduït una sèrie de dades, ha premut el botó d'executar una determinada acció (en el cas de l'alta només té disponible un botó d'alta i en el cas de les modificacions totes les accions disponibles per l'Objecte de Negoci segons l'estat i el rol de l'usuari) i es crida al cas d'us CUAccio del Workflow per tal que l'executi segons la configuració de la mateixa.

Actors:

Usuari

Precondicions:

L'usuari ha seleccionat una Accio de la pantalla d'altres o modificacions.

Postcondicions:

S'ha creat/modificat un Objecte de Negoci a la base de dades.

Flux normal:

L'usuari esta en la pantalla d'altres o modificacions on introdueix tota la informació necessària.

Si es una alta prem el botó d'alta i si es una modificació prem el botó que es correspon amb l'acció que desitja executar. L'aplicatiu recull les dades introduïdes a la pantalla i, juntament amb el codi de l'acció i l'usuari, crida al cas d'us CUAccio per tal que sigui el Workflow qui a partir de la configuració de l'Acció, l'executi.

Si l'execució ha estat correcta es mostra una pantalla informat del resultat i després de prémer un boto amb l'opció de continuar s'enllaça amb el cas d'us CULlistarObjectes per tornar a la pantalla principal.

Flux alternatiu:

Si l'aplicatiu treballa amb més d'un tipus d'Objecte de Negoci i pel que fa la pantalla, es pot elegir entre dos tipus d'interfaç:

- Si les dades a introduir son gairebé idèntiques es pot optar per compartir la pantalla i que hi hagi un camp per seleccionar la tipologia. Aquesta opció requereix omplir el camp amb els tipus disponibles, i que es comprovi tipologia a tipologia mitjançant el cas d'us CUBuscarAccions si te disponible l'Acció d'alta i en cas de ser negatiu no mostrar aquella tipologia en concret.
- Si les dades son totalment diferents, al seleccionar l'acció alta de la pantalla principal, accedir també al CUBuscarAccions per cada un dels tipus per si te disponible l'Acció d'alta, llavors mostrar una pantalla intermèdia per que l'usuari seleccioni quin tipus d'Objecte de Negoci vol donar d'alta i saltar a la pantalla corresponent.

Si el Workflow retorna error, l'aplicatiu no es mou de la pantalla de alta-modificacio i mostra a la part superior el missatge d'error retornat.

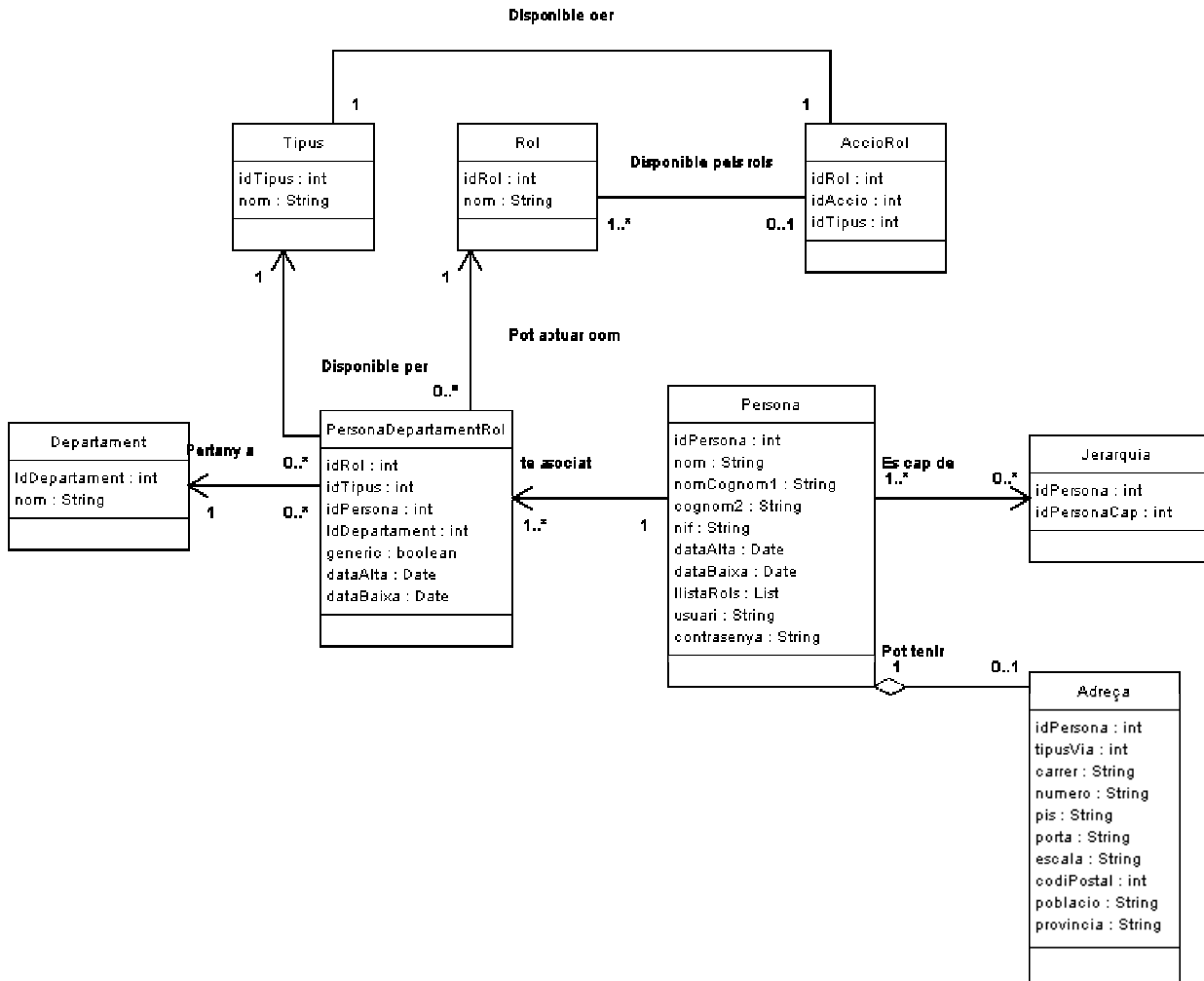
mostra a la part superior el missatge d'error retornat.

Les classes es poden dividir en 4 grans grups:

- Relacionades amb persones.
- Amb els estats
- Objecte de negoci
- Execució d'accions

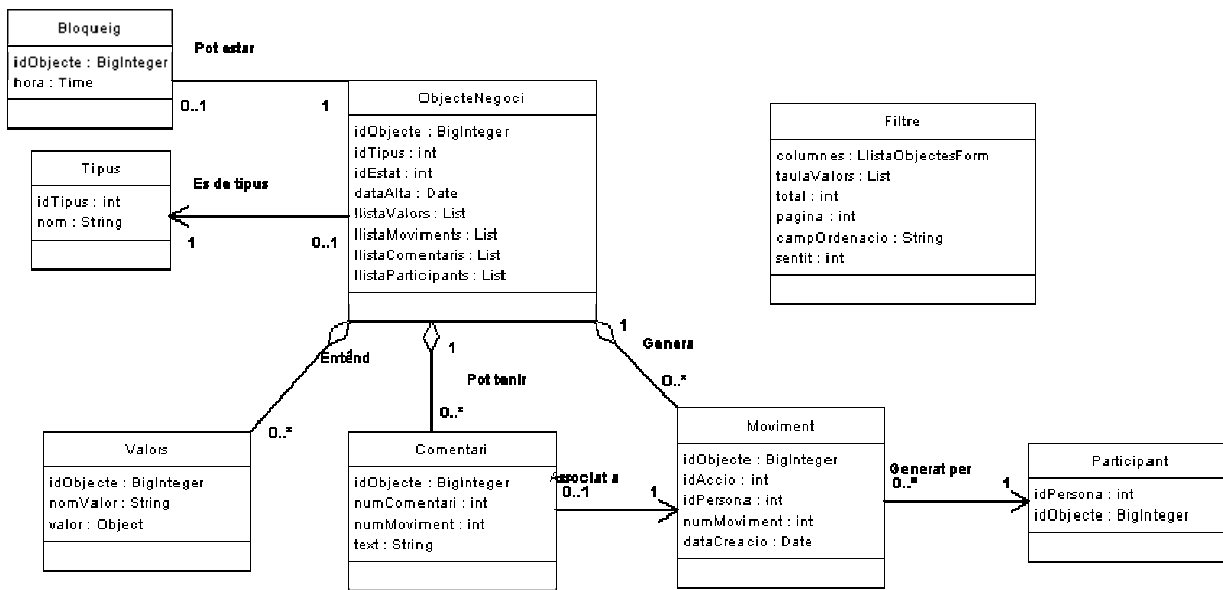
Per cada un d'aquest grans grups he realitzat un subdiagrama del genèric per tal de poder veure en més detall les seves inter-relacions.

3.6.1. Diagrama de Persones



En el diagrama es pot observar com una Persona a més a més de dades personals, pot tenir una adreça postal, jeràrquicament pot tenir un cap i desenvolupa un o varis rols dins un departament en funció de la tipologia d'un Objecte. Per altra banda, es veu a relació entre rols i accions, aquestes estaran associades a n rols, alguns tindran permisos d'execució i altres se'ls afegeix com a participants en el moment de l'execució.

3.6.2. Diagrama d'Objecte de negoci



En el diagrama es veu els diferents objectes que componen un Objecte de Negoci genèric. Es pot observar que pertany a un determinat tipus, que es troba en un estat concret, pot tenir n moviments realitzats per diferents participants, comentaris ordenats i associats al moviment (és la única manera de saber qui els ha incorporat). Finalment la classe Valor que incorpora els m atributs en format camp/valor propis d'una determinada tipologia d'un Objecte de Negoci.

3.6.3. Diagrama d'Estats

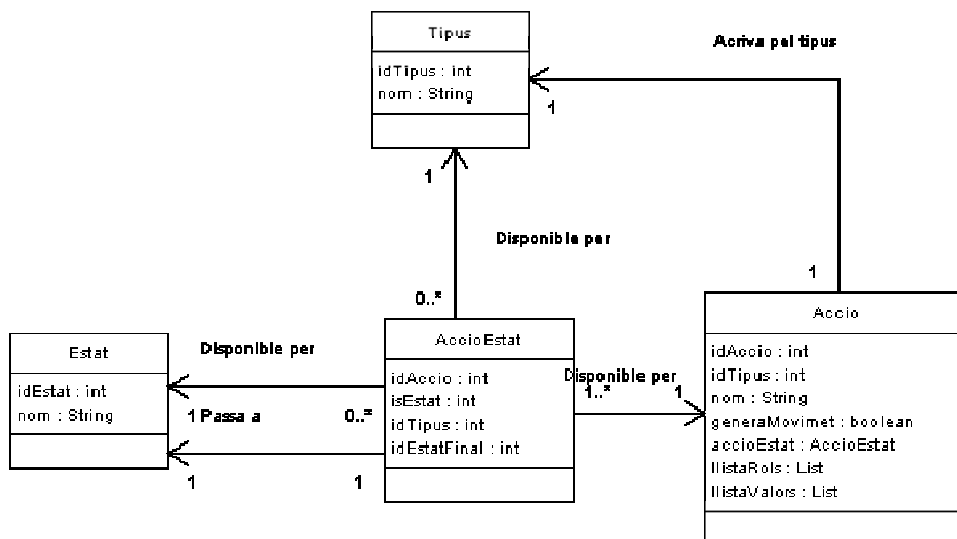
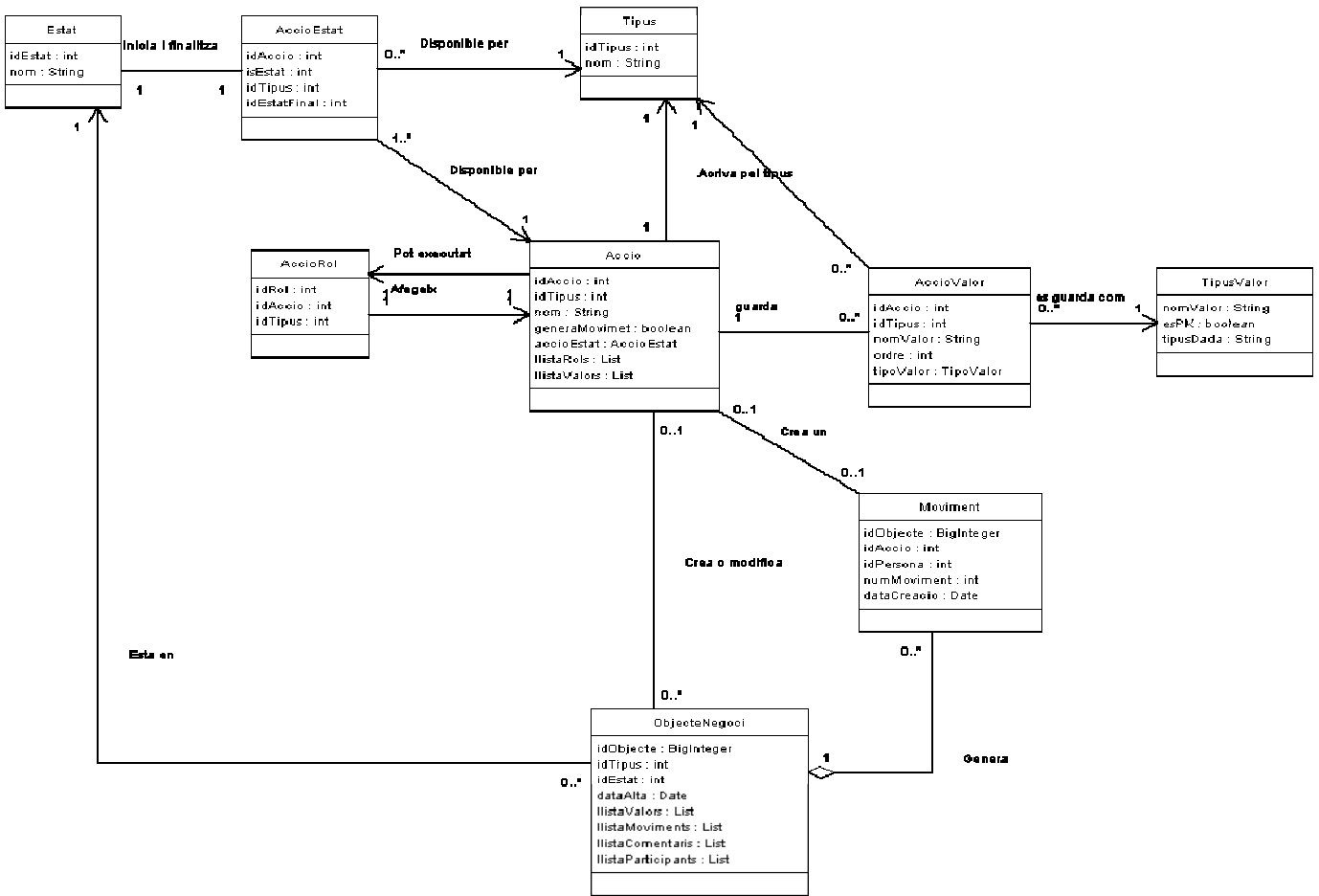


Diagrama on es mostra les relacions entre estats, tipologies i accions. Es veu com els estats poden ser compartits entre diferents tipologies, les accions sí que estan lligades a una tipologia en concret i mitjançant la entitat AccióEstat es relacionen totes 3. D'aquesta manera es pot saber quines accions hi ha disponibles quan l'Objecte de Negoci estat en un estat concret.

3.6.4. Diagrama d'Accions



En el diagrama s'observa per una banda que les accions estan disponibles en funció de l'estat de l'Objecte de Negoci i mitjançant l'entitat AccioRol es pot saber quins rols la poden executar i quins rols seran afegits com a participants. A la banda dreta es veu si l'execució genera un moviment visible o no, i quins camps es guarden a l'entitat valors en funció d'AccioValor i TipusValor, aquesta ultima guarda la configuració de cada un dels camps, com ara si son numèrics, de text, etc.

3.6.5. Llista d'entitats

A continuació es mostra un llistat amb la descripció de totes les entitats i els seus atributs.

3.6.5.1. Persona

Llista de persones que tenen relació amb algun dels aplicatius que usen el Workflow, inclou empleats i en el cas de l'aplicatiu de queixes i reclamacions, persones alienes a l'empresa que en el seu moment han interposat una queixa o reclamació.

També indica de forma sintètica el perfil.

En cas de tractar-se de persones jurídiques només s'informaria el nom.

Atributs

- IdPersona: Token que identifica inequívocament una persona.
- nom: Nom de la persona o nom de l'empresa si es una persona jurídica.
- cognom1: Primer cognom de la persona. Buit si es jurídica.
- cognom2: Segon cognom de la persona. Buit si es jurídica.
- nif: Numero de NIF o CIF si es persona jurídica.
- dataAlta: Data d'incorporació efectiva a la base de dades i a partir de la qual es pot usar pels diferents aplicatius de l'organització.
- dataBaixa: Data en que deixa de estar activa pels aplicatius de l'organització.

3.6.5.2. Empleat

Dades pròpies i sensibles dels empleats de l'empresa com són el seu codi d'empleat i l'usuari d'accés a la xarxa..

Atributs

- IdPersona: Token que identifica inequívocament a una persona que es empleat.
- codiEmpleat: Numero únic amb que l'empresa identifica un empleat.
- usuari: En cas d ser un empleat, l'identificador d'usuari dins la xarxa.
- Contrasenya: Paraula secreta amb que validem que l'usuari es qui diu ser.

3.6.5.3. Adreça

Adreça postal d'una Persona, pot ser necessari posar-se en contacte amb una Persona per tal d'informar-lo de com ha anat la seva Queixa o per demanar-li més informació. Les dades de l'adreça són especialment importants si la Queixa o Reclamació l'ha realitzat una persona aliena a l'empresa i no disposem de cap altra via de comunicació per tal de poder-la informar de la seva resolució.

Atributs

- IdPersona: Token que identifica inequívocament a una persona que es empleat.
- tipusVia: Es correspon al codi que identifica si es un carrer, avinguda, etc.
- carrer: Nom del carrer
- numero: Numero de l'edifici.
- Pis: Pis del domicili.
- Porta: Porta dins la planta.
- escala: Escala de l'edifici
- codiPostal: Les 5 xifres que identifiquen una població.
- població: Població de la persona.
- provincia: Província on es troba la població-

3.6.5.4. Participant

Persona que participa en alguna moment dins del cicle de vida d'un Objecte de Negoci. Pot ser que hagi executat una acció, que sigui una persona externa a l'empresa que hagi formulat una queixa o reclamació, o que al executar una acció se l'hagi afegit com a destinatari per tal que intervingui o sigui coneixedor del procés de vida de l'Objecte de Negoci.

Atributs

- IdPersona: Token que identifica inequívocament una persona.
- IdObjecte: Token que identifica inequívocament un Objecte de Negoci

3.6.5.5. Jerarquia

Tots els empleats de l'empresa tenen algun responsable, amb aquesta entitat s'identifica qui es el seu cap o responsable.

Atributs

- IdPersona: Token que identifica inequívocament una persona.
- IdPersonaCap: Token que identifica inequívocament a la persona que dins l'estructura organitzativa esta jeràrquicament per damunt.

3.6.5.6. Departament

Entitat que emmagatzema la llista de departaments de l'empresa.

Atributs

- IdDepartament: Token que identifica inequívocament el departament.
- nom: Nom descriptiu del departament.

3.6.5.7. Rol

Identifica cadascun dels possibles perfils que poden ser necessaris durant el cicle de vida d'un Objecte de Negoci.

Per tal de tenir una jerarquitxació es important que els rols amb menys importància tinguin identificadors baixos i els de major grau més gran. D'aquesta manera es pot saber si un rol esta per damunt d'un altre.

Atributs

- IdRol: Token que identifica inequívocament un rol.
- nom: Nom descriptiu del rol.

3.6.5.8. PersonaDepartamentRol

Relaciona el departament en que treballa una persona y quin rols hi desenvolupa per un deteminat tipus d'Objecte de Negoci.

Atributs

- IdRol: Token que identifica inequívocament un rol.
- IdTipus: Token que identifica inequívocament un tipus.
- IdPersona: Token que identifica inequívocament una persona.
- genèric: Indica que la persona amb aquest rol pot actuar com a usuari genèric per Objectes de Negoci d'aquest tipus. Entre altres avantatges te disponibles totes les accions disponibles en un determinat estat.
- dataAlta: Data en la que persona pot actuar amb un determinat rol.
- dataBaixa: Data en que deixa de poder treballar el rol.

3.6.5.9. Tipus

Llista amb tots els possibles tipus d'Objectes de Negoci amb que pot treballar la utilitat de Workflow

Atributs

- IdTipus: Token que identifica inequívocament un tipus.
- nomTipus: Nom descriptiu del tipus.

3.6.5.10. ObjecteNegoci

Objecte genèric amb que treballa el Workflow.

Conte tota la informació necessària per poder executar les diferents operacions de negoci.

Atributs

- IdObjecte: Token que identifica inequívocament un Objecte de Negoci
- IdTipus: Token que identifica inequívocament a quin tipus pertany.
- IdEstat: Token que identifica inequívocament en quin estat es troba.
- IdDepartament: Token que identifica inequívocament el departament al que pertany inicialment l'Objecte de Negoci.
- dataAlta: Data en la que el van crear.

3.6.5.11. Valors

Entitat metadàtica per poder emmagatzemar els atributs propis i no genèrics d'un Objecte de Negoci d'un tipus concret.

La llista de valors varia tant en funció del tipus d'Objecte de Negoci com de l'estat en que esta, ja que en funció de la seva evolució es poden anar incorporant elements a la llista.

Atributs

- IdObjecte: Token que identifica inequívocament l' Objecte de Negoci al que va associat.
- NomValor: Identificador del camp al que correspon el valor que emmagatzema.
- valor: Valor concret que emmagatzema.

3.6.5.12. Comentari

Anotacions que un participant ha afegit a un Objecte de Negoci.

Un comentari no pot ser modificat ni eliminat.

Atributs

- IdObjecte: Token que identifica inequívocament l' Objecte de Negoci al que va associat.
- NumComentari: Ordinal del comentari per tal de poder ser llegits de manera ordenada. Per cada Objecte de Negoci es comença pel 1.
- numMoviment: Ordinal per poder identificar l'ordre en que s'han generat els diferents moviments realitzats sobre l'Objecte de Negoci.
- tipusComentari: Identifica el tipus de comentari que s'ha introduït. Els possibles tipus son les següents:
 - 1. Text lliure.
 - 2. Resolució.
 - 3. Replica.
- text: Text del comentari que ha afegit un participant.

3.6.5.13. Moviment

Entitat que conte totes les accions realitzades sobre un Objecte de Negoci. Identifica tant qui la realitzat, com quan l'ha executat i, mitjançant la seva relació amb l'entitat Comentari, si n'hi ha aportat algun.

Atributs

- IdObjecte: Token que identifica inequívocament l' Objecte de Negoci al que va associat.
- IdAccio: Identificador de l'acció realitzada.
- IdPersona: Token que identifica inequívocament la persona que ha executat l'acció.
- NumMoviment: Ordinal per poder identificar l'ordre en que s'han generat els diferents moviments realitzats sobre l'Objecte de Negoci.
- TipusComentari: Identifica el tipus de comentari que s'ha afegit. Les tipologies seran:
 - 1.- Comentari text lliure
 - 2.- Resolució
 - 3.- Replica
- dataCreació: Data del sistema del moment en que es va crear. En camp moment es pot modificar o eliminar el seu valor.

3.6.5.14. Bloqueig

Executar una acció es un proses relativament llarg, costos i que implica accés a múltiples taules a la base de dades, per això, és molt important que ningú pugui executar una altra Acció sobre l'Objecte de Negoci al mateix temps.

Al començar l'execució(excepte a l'alta) i en el moment de buscar la configuració de l'Acció, es comprova si hi ha un registre a l'entitat Bloqueig, si no n'hi ha cap, se li afegeix un registre amb l'hora del sistema i es borra al finalitzar-la. Si hi ha un registre es comprova si entre l'hora actual i l'hora del camp difereixen un cert temps, si es així es considera que ja ha passat el bloqueig, sinó es retorna error.

Atributs

- IdObjecte: Token que identifica inequívocament l' Objecte de Negoci al que va associat.
- Hora: Dia i hora dels sistema en que s'ha bloquejat un Objecte de Negoci

3.6.5.15. Estat

Identifica cadascun dels possibles estats que poden succeir-se durant el cicle de vida d'un Objecte de Negoci.

Atributs

- IdEstat: Token que identifica inequívocament un estat.
- nom: Nom descriptiu de l'estat.

3.6.5.16. EstatTipus

Marca els diferents estats habilitats per una determinada tipologia d'Objecte de Negoci.

Atributs

- IdEstat: Token que identifica inequívocament un estat.
- IdTipus: Token que identifica un tipus per al quan esta disponible el rol.

3.6.5.17. Acció

Identifica les diferents accions que es poden executar sobre els diferents tipus d'Objectes de Negoci.

Atributs

- IdAcció: Token que identifica inequívocament una acció.
- IdTipus: Token que identifica un tipus per al qual esta disponible.
- nom: Nom descriptiu de l'acció.
- generaMoviment: Indica si el moviment que genera l'acció per defecte es visible o no. Per algunes transicions ens interessa que el moviment sigui visible o públic pels participants i altres vegades no es necessari.

3.6.5.18. AccioEstat

Marca els diferents que poden donar origen a una Acció i l'estat final en el que queda l'Objecte després de l'execució.

Atributs

- IdEstat: Token que identifica inequívocament un estat.
- IdAcció: Token que identifica inequívocament una acció.
- IdTipus: Token que identifica un tipus.

3.6.5.19. AccioRol

Marca els diferents rols que poden intervenir en una acció i de quina manera: uns rols tindran permís per poden executar-la i altres seran afegits com a participants com a conseqüència de la seva execució.

Atributs

- IdRol: Token que identifica inequívocament un rol.
- IdAcció: Token que identifica inequívocament una acció.
- IdTipus: Token que identifica un tipus per al qual esta disponible.
- execució: El rol te permís per executar l'acció.
- participant: Al executar-se se li afegeix un participant amb el rol en qüestió; es busca dins la jerarquia de persones la primera que pot actuar amb aquest rol.

3.6.5.20. AccioValor

Indica quins camps s'han de persistir dins la base de dades al executar una acció. També indica si el camp es obligatori i en quin ordre.

Atributs

- IdAcció: Token que identifica inequívocament una acció.
- IdTipus: Token que identifica un tipus per al qual esta disponible.
- Taula: Taula on s'emmagatzema el camp.
- valor: Valor concret que emmagatzema.
- order: Ordinal per identificar la seqüència correcta en que s'ha de guardar.
- obligatori: Atribut per saber si el camp ha de venir informat o no. Si no ho esta s'ha de generar un error.

3.6.5.21. TipusValor

Entitat per poder identificar el format de les dades d'un Objecte de Negoci. Es necessària en el moment de generar les querys dinàmiques ja que s'ha de diferenciar entre valors numerics, text, data, etc.

Atributs

- NomValor: Identificador del camp al que correspon el valor que s'ha de persistir.
- Taula: Identificador de la taula on s'ha de persistir.
- EsPK: Indica si el camp es clau primària de la seva taula.
- tipusDada: Tipus basic de dada, els seus valors podran ser Text, enter, booleà, import, data.

3.6.5.22. Filtre

Entitat per poder agrupar les opcions de filtratge i paginació que l'usuari desitja alhora de fer una cerca entre els Objectes de negoci que te disponibles

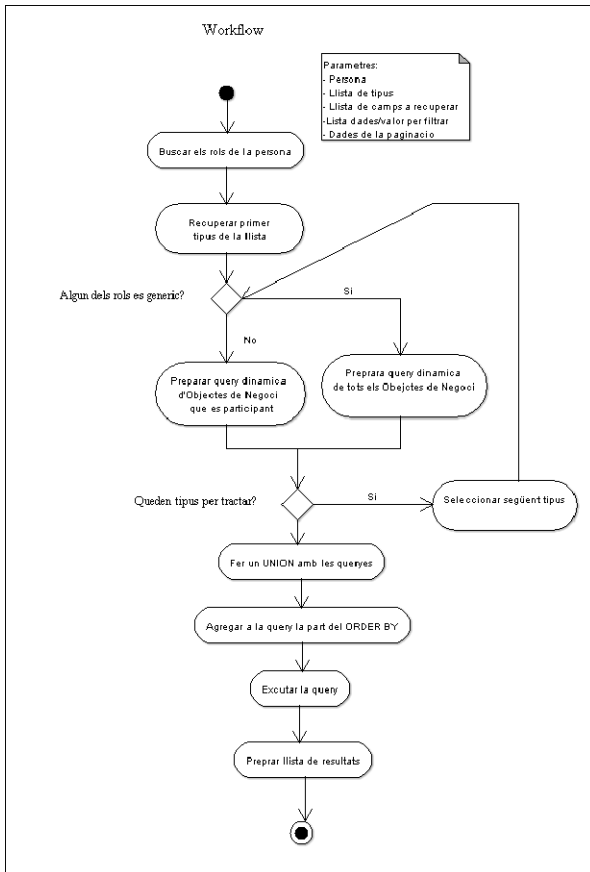
Atributs

- **columnes:** Llista de camps que es volen recuperar i que es corresponen a les columnes de la taula de resultats. També inclou els camps que no seran visibles com ara identificadors que tot i no mostrar-se a l'usuari son necessaris.
- **taulaValors:** Taula de tipus hash que emmagatzema els camps/valors de les opcions de filtre que ha introduït l'usuari.
- **total:** Nombre total de resultats que veuran a la pagina. Si per exemple, la cerca en retorna 200, potser a la pantalla nomes en voldrem mostrar 10, per tant els 190 restants no cal enviar-los.
- **pagina:** Pagina de resultats a la que es vol accedir. Si per exemple mostrem 10 resultats per cerca, estem a la pagina 4 i fem un 'següent', el valor pagina val 5, a la cerca mostrarem els resultats del 40 al 50.
- **campOrdenació:** Columna sobre la que farem l'ordenació, ha de venir informada dintre del camp columnes.
- **sentit:** Pot ser ascendent o descendent

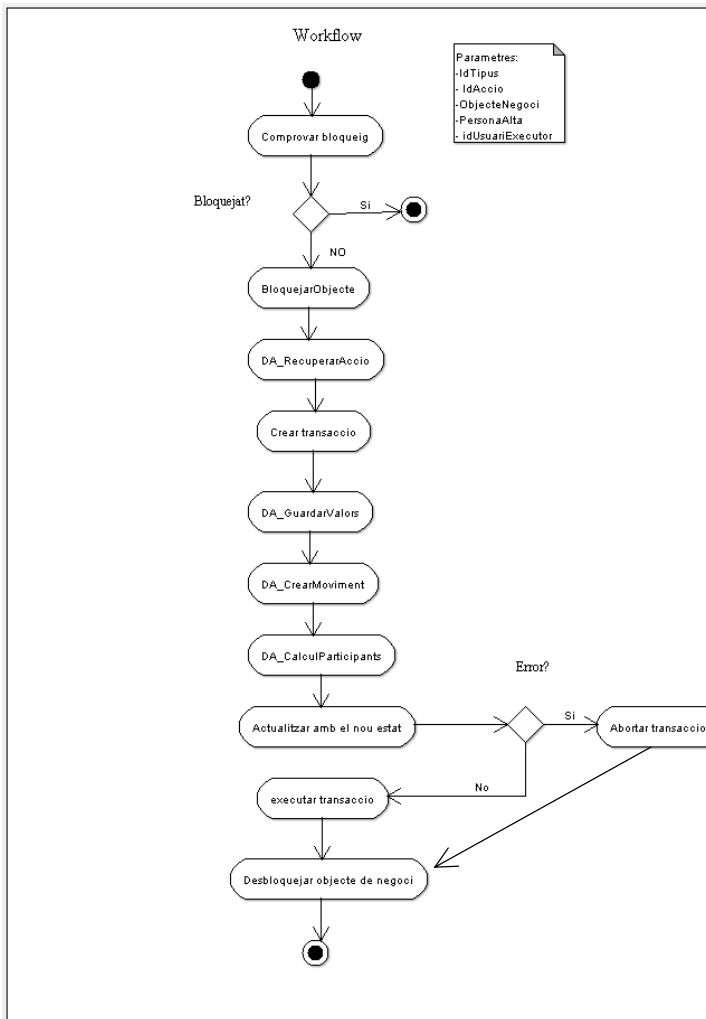
3.7. Diagrames d'activitat

3.7.1. Utilitat de Workflow

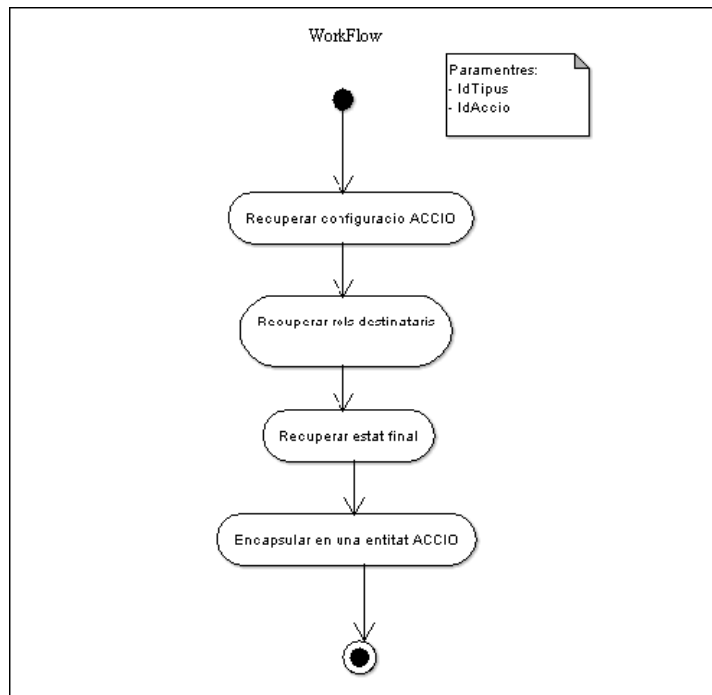
3.7.1.1. CUBuscarObjectes



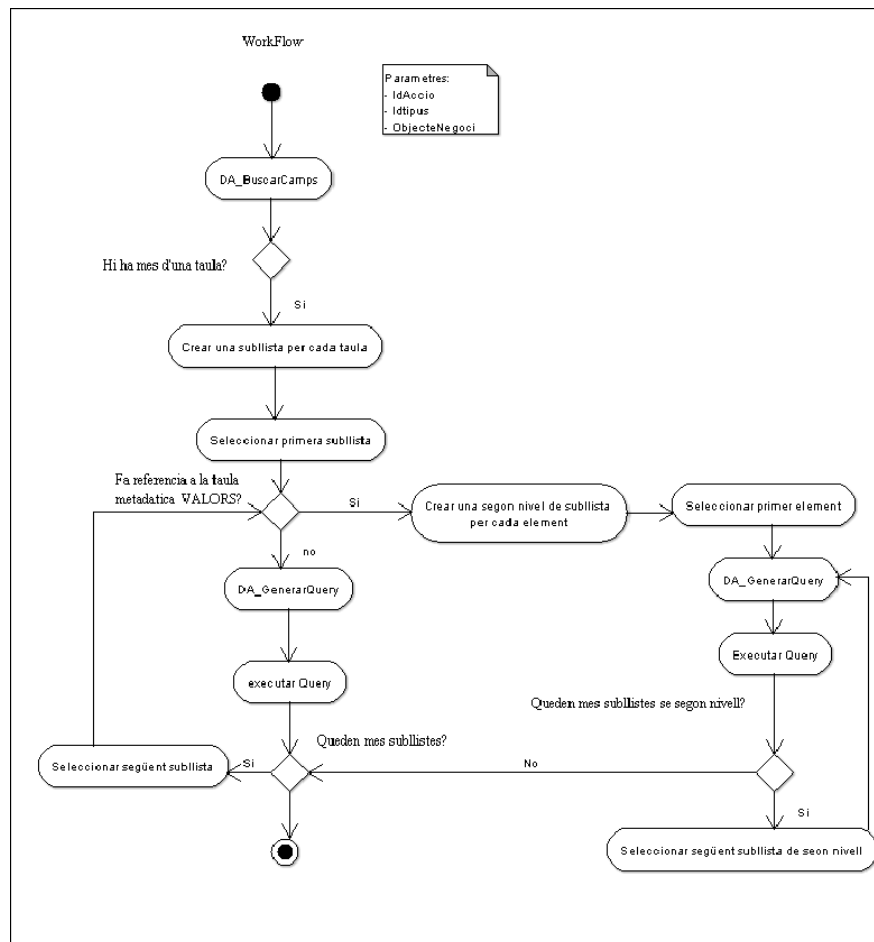
3.7.1.2. CUAccio



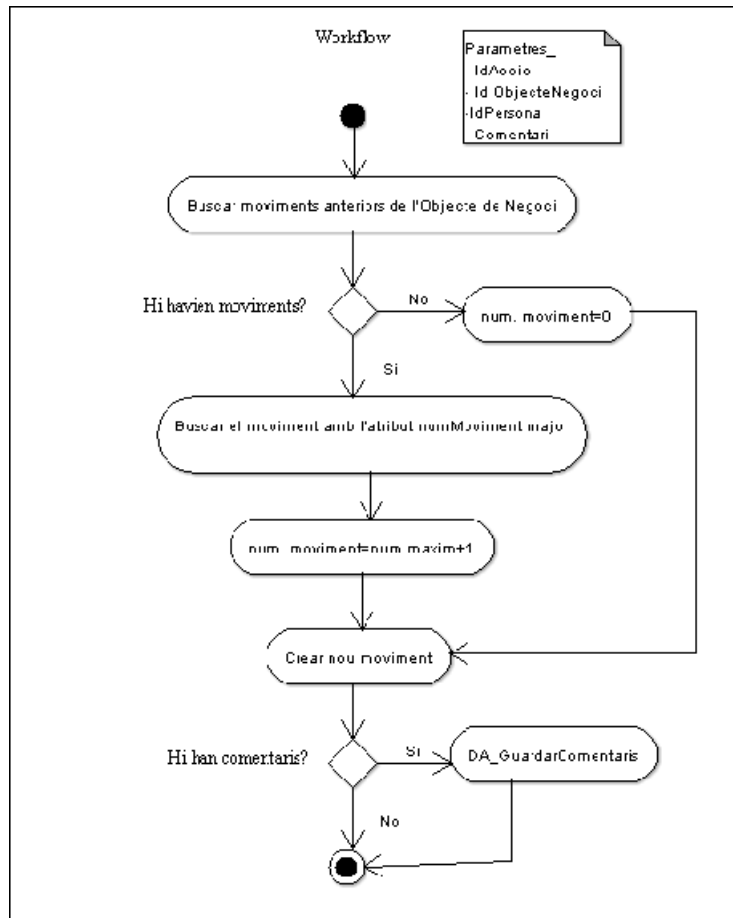
3.7.1.3. CURecuperarAccio



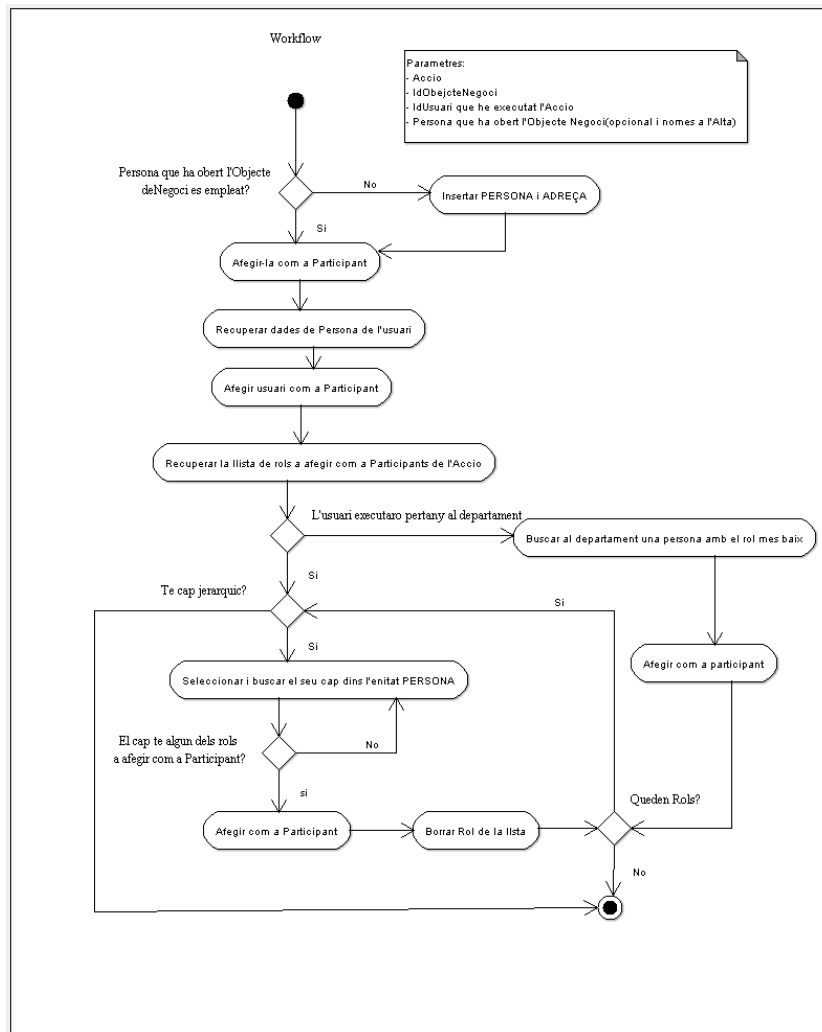
3.7.1.4. CUGuardarValors



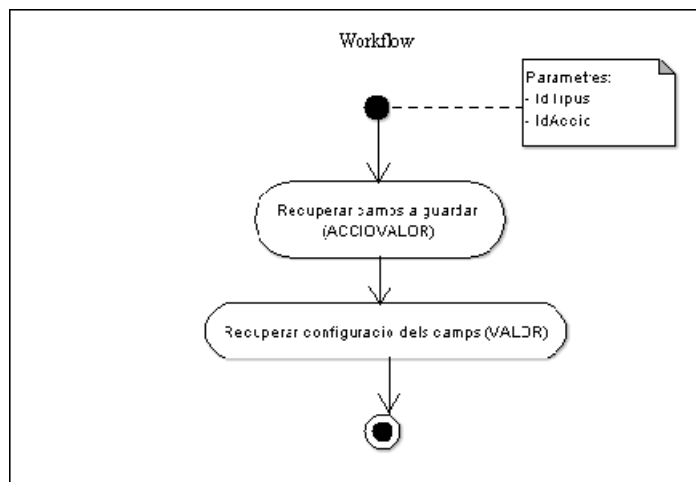
3.7.1.5. CUCrearMoviment



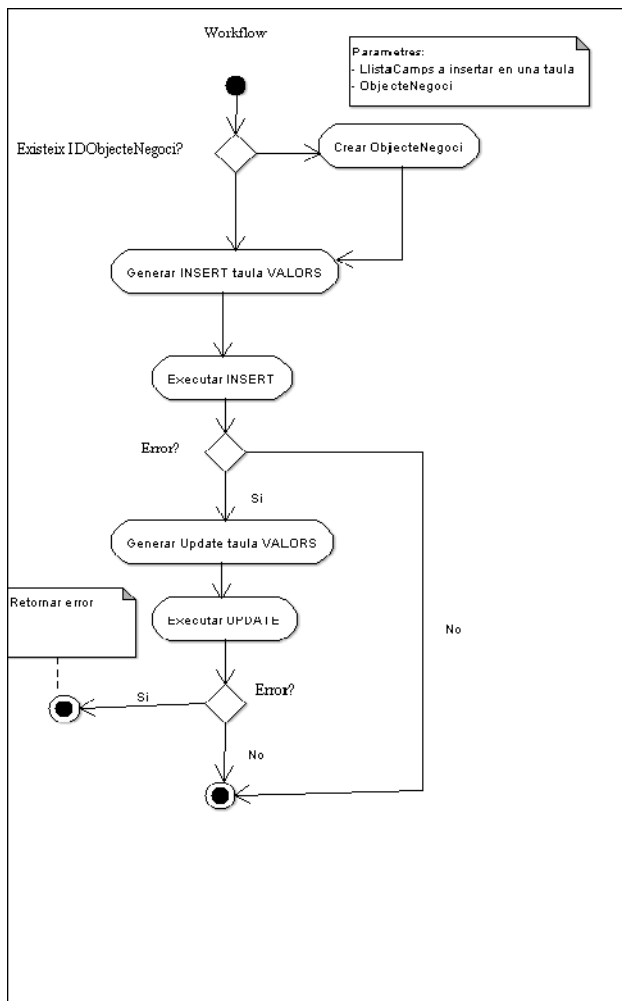
3.7.1.6. CUCalculParticipants



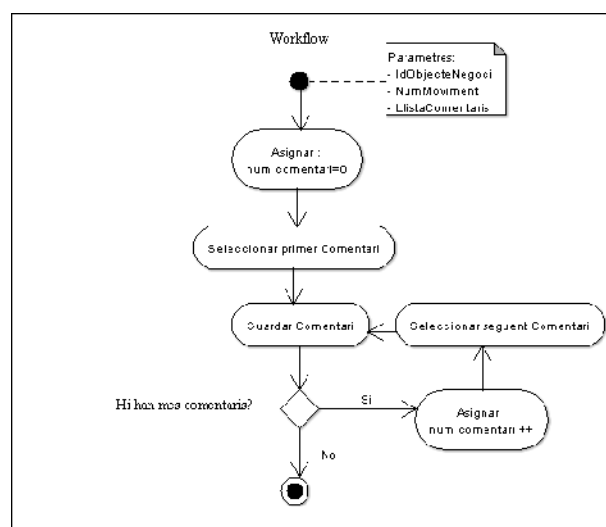
3.7.1.7. CUBuscarCamps



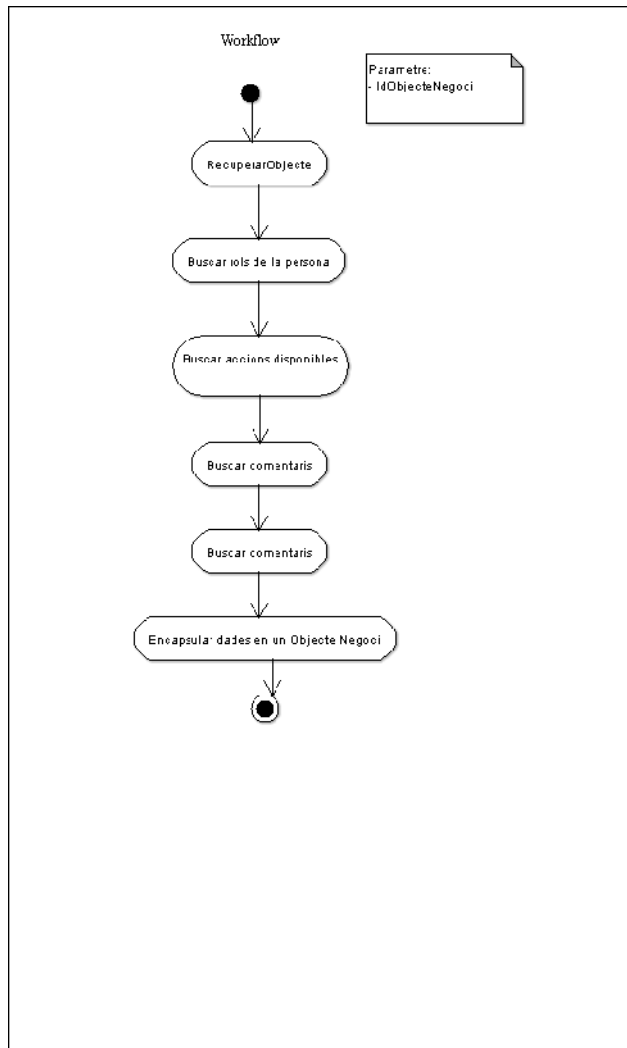
3.7.1.8. CUGnenerarQuery



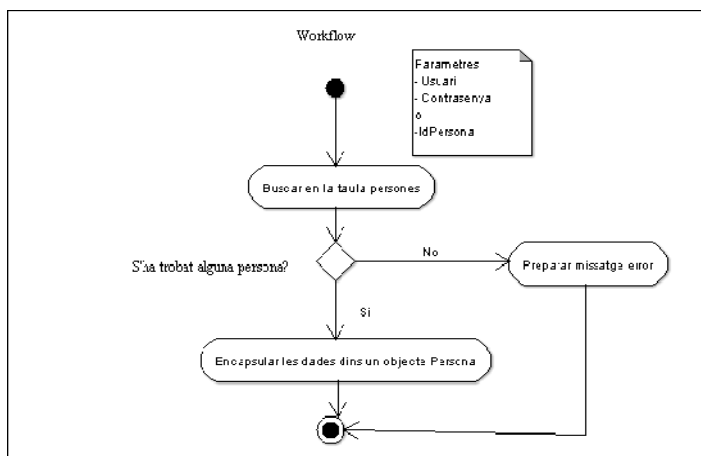
3.7.1.9. CUGuardarComentarios



3.7.1.10. CUConsultarObjecte

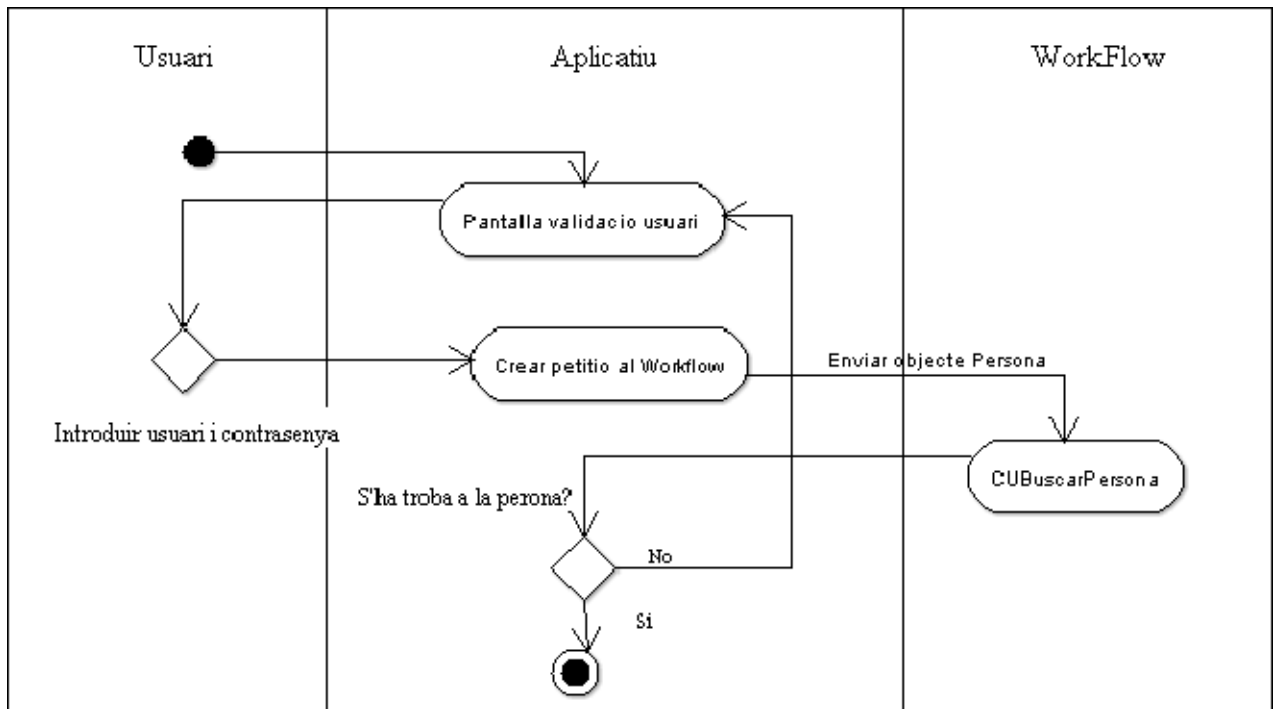


3.7.1.11. CUBuscarPersona

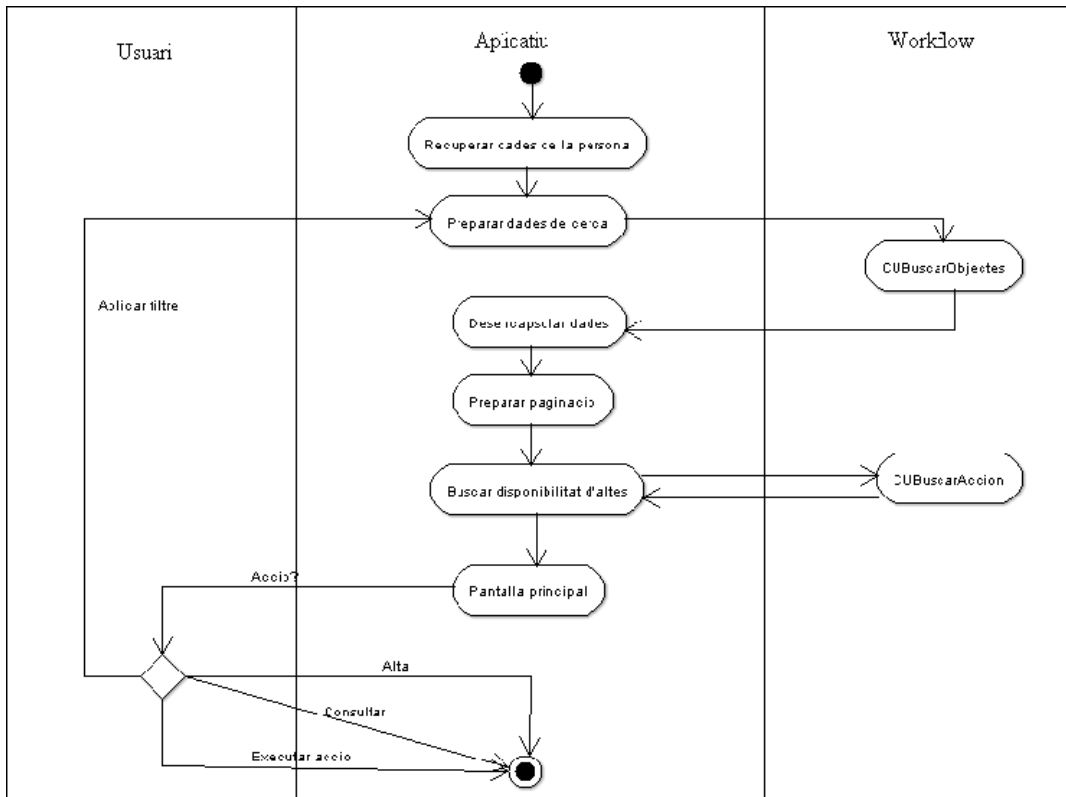


3.7.2. Aplicatiu de Reclamacions i Queixes

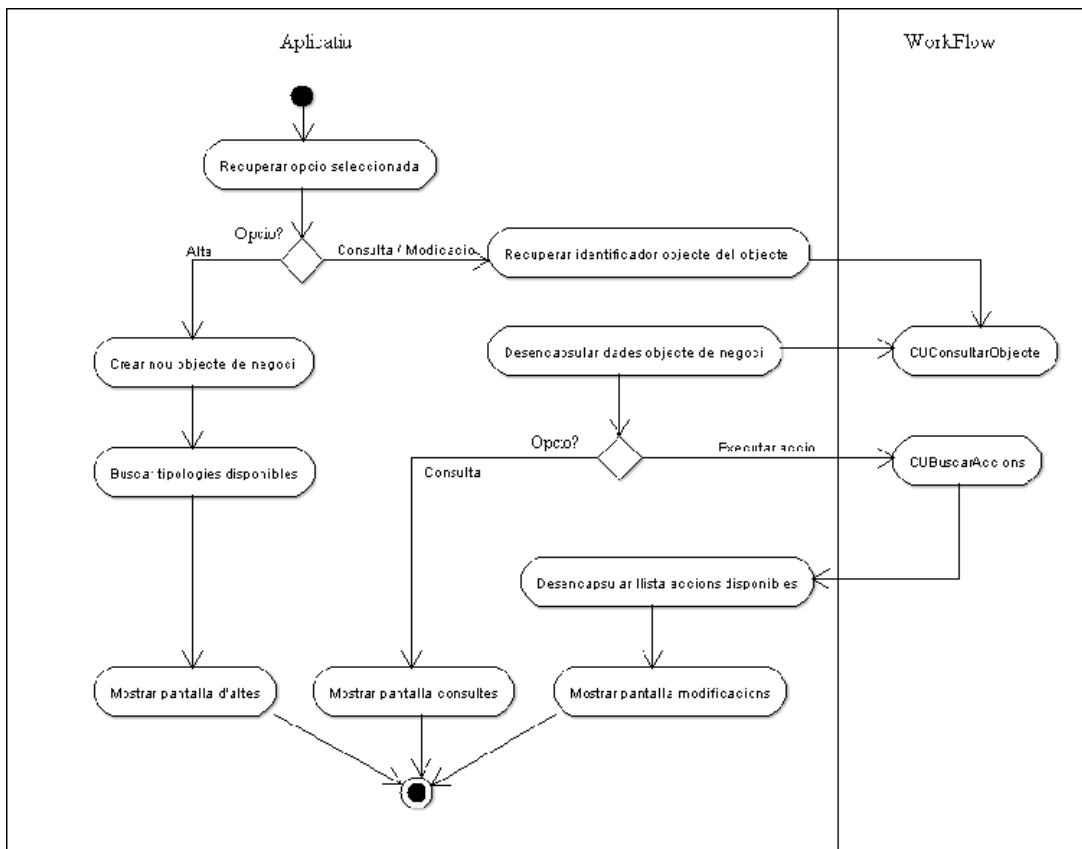
3.7.2.1. CUValidarUsuari



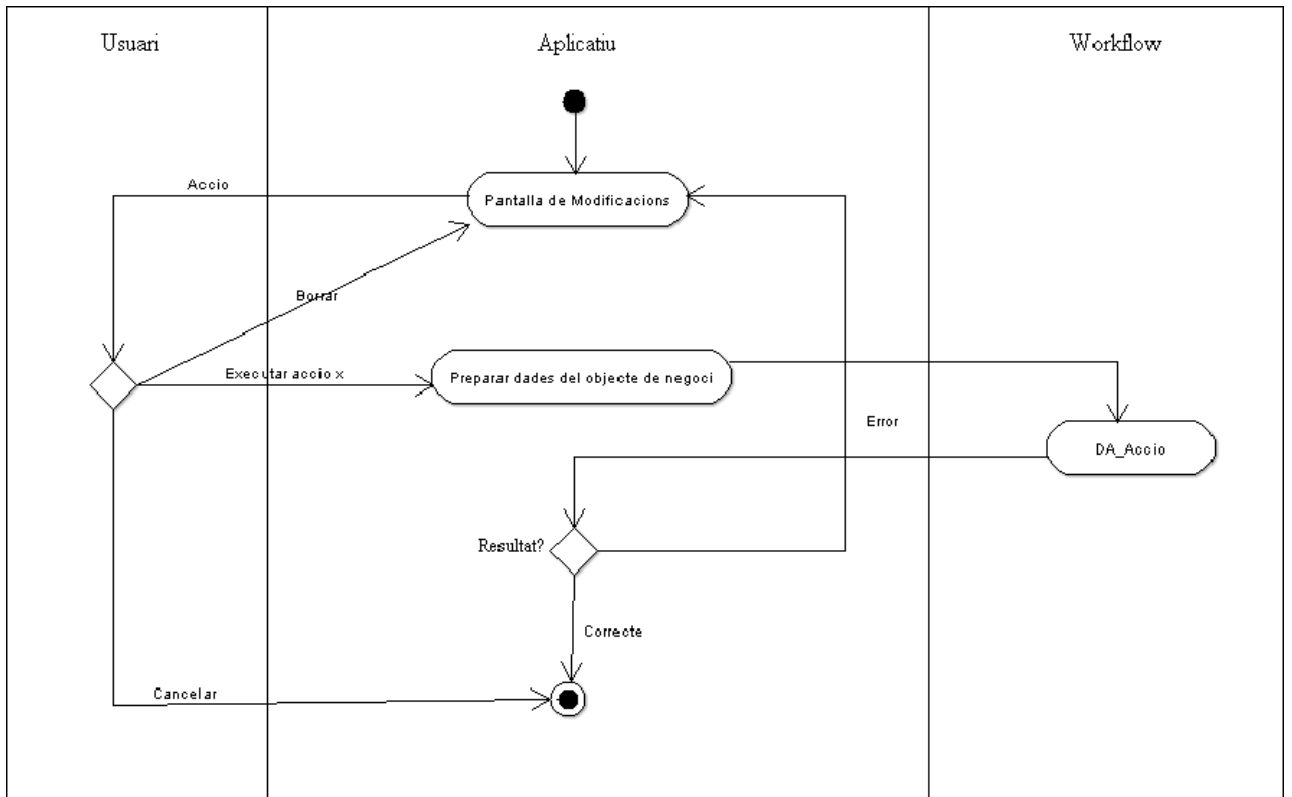
3.7.2.2. CULListarObjectes



3.7.2.3. CURecuperarDades



3.7.2.4. CUExecutarAccio



3.8. Definició dels Workflows

3.8.1. Diagrama de flux de les Queixes



Les Queixes poden provenir de dues vies: per la via interna o el que és el mateix, un empleat te una queixa sobre algun tema concret i per la via externa, quan arriba una queixa a l'empresa per qualsevol via de comunicació (mail, carta, fax. Etc.). Provingui d'on provingui, es fa arribar al departament d'Atenció al client" per tal de canalitzar totes les altes.

100 Pendent Acceptació: Una vegada un dels tramitadors del departament d'Atenció al client" ha fet l'alta, la queixa s'ha enviat a una persona del departament implicat en la Queixa. Des d'aquest departament es poder executar 3 accions:

- Desestimar: Es considera que no té fonament.
- Pujar de nivell: Si la persona no es veu competent sobre tema, fa participant de la Queixa al seu superior per tal que sigui ell qui es faci càrrec de la mateixa. L'acció es pot anar repetint fins arribar al màxim nivell jeràrquic dins l'empresa
- Acceptar: La persona es veu competent sobre la Queixa i comença els tràmits que requereixi la resolució de la queixa. Aquests tràmits no tenen perquè tenir res a veure amb al Web de Queixes i Reclamacions, pot ser des d'avisar a personal per unes dietes d'un empleat mal liquidades o perquè un empleat aparca a la plaça de pàrquing d'una altra persona.

150 No Aceptada: Es un estat final. La queixa es considera improcedent pels motius que siguin i no es pren cap mesura.

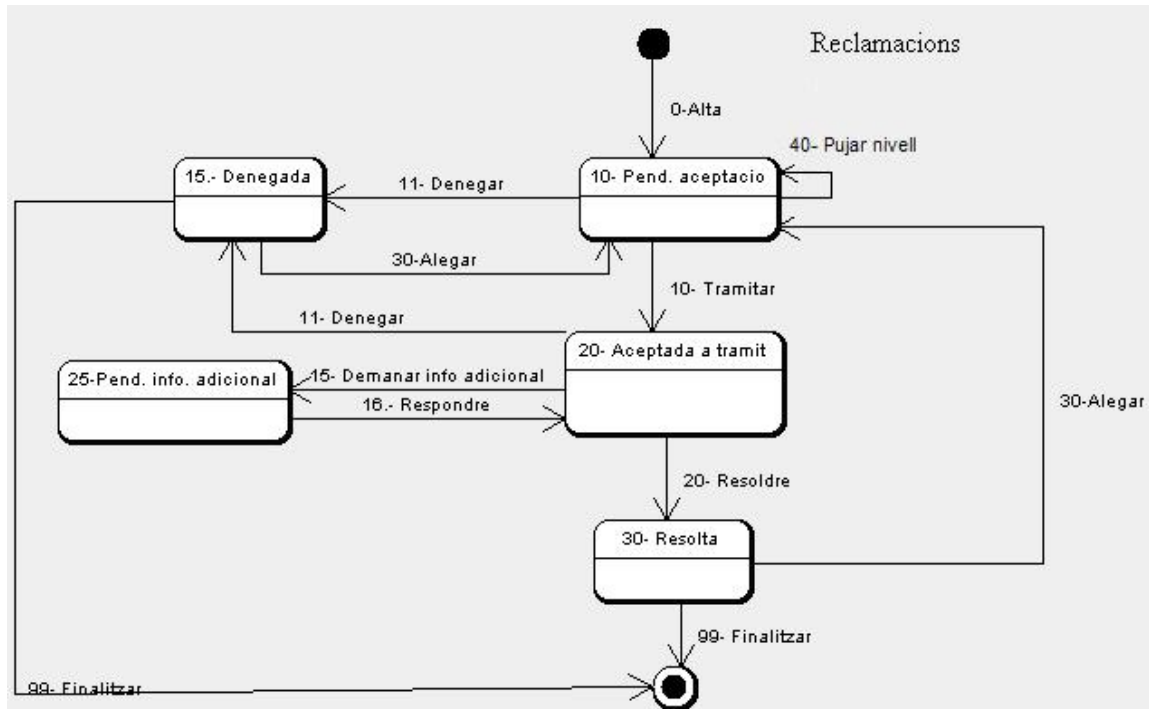
100 Aceptada: L'usuari ha començat els tràmits i les gestions internes per tal de solucionar el motiu de la queixa. Una vegada es s'ha solucionat el tema pot executar 2 accions:

- Desestimar: Després de rebre les respostes dels implicats amb la Queixa es considera que no hi havia fonament i es desestima.
- Finalitzar: Si s'ha arreglat satisfactòriament es passa a estat Finalitzada.

999 Finalitzada: Es un estat final. La Queixa ha finalitzat els seu procés de vida satisfactòriament.

En tots els casos (llevat de l'alta) en l'execució d'una acció només es demana que s'afegeixi un comentari que expliqui el perquè de la decisió que s'ha pres o que s'ha fet en cada moment.

3.8.2. Diagrama de flux de les Reclamacions



Com en el cas de les Queixes, una reclamació pot venir per via interna o externa i, com en el cas anterior, les altes son canalitzades pel departament d'Atenció al client".

Els estats pels que passa una Reclamació son els següents:

10 Pendent Acceptació: Una vegada un dels tramitadors del departament d'Atenció al client ha fet l'alta, la Reclamació s'ha enviat a una persona del departament implicat.. Des d'aquest departament es poder executar 3 accions:

- **Denegar:** Es considera que no té fonament.
- **Pujar de nivell:** Si la persona no es veu competent sobre tema, fa participant de la Queixa al seu superior per tal que sigui ell qui es faci càrrec de la mateixa. L'acció es pot anar repetint fins arribar al màxim nivell jeràrquic dins l'empresa.
- **Acceptar:** La persona es veu competent sobre la Queixa i comença els tràmits que requereixi la resolució de la queixa. Aquests tràmits no tenen perquè tenir res a veure amb al Web de Queixes i Reclamacions, pot ser des d'avisar a personal per unes dietes d'un empleat mal liquidades o perquè un empleat aparca a la plaça de pàrquing d'una altra persona.

15 Denegada: En el cas de la Reclamació no és un estat final. La Reclamació es considera improcedent pels motius que siguin i no es pren cap mesura, o pel contrari la persona que ha fet l'alta (i dins d'un plaç estipulat) pot Al·legar amb el que torna a l'estat 10, pendent de que s'accepti o es torni a Denegar. En cas que passin els dies, es pot Finalitzar amb el que sí que passarà a un estat final.

20 Acceptada a tràmit: L'usuari ha començat els tràmits i les gestions internes per tal de solucionar el motiu de la Reclamació.

Si fan referència a altres departaments la pot deixar executada l'acció Demandar informació addicional i es queda en un estat d'espera.

Finalment una vegada disposa de tota la informació pot decidir entre Resoldre per tal de finalitzar o Denegar la reclamació si creu que no es procedent.

25 Pendent d'informació addicional: L'usuari no disposa de tota la informació per prendre una decisió en aquell moment, pot ser que hagi de buscar unes factures, parlar amb un company que està de vacances, etc. Mentre no disposa de tota la informació que necessita, la deixa en aquest estat temporalment. Una vegada té tot el que necessita, executa l'acció Respondre, per tal de que pugui prendre una decisió.

30 Resolta: S'ha pres una decisió sobre la Reclamació i s'ha informat a l'usuari sobre la mateixa. En cas de no rebre cap resposta i una vegada han passat uns dies, es Finalitza la Reclamació. En cas que la persona que ha interposat la Reclamació no estigui d'acord amb la resolució, pot Al·legar i tornaria a començar el procés.

999 Finalitzada: Es un estat final. La Reclamació ha finalitzat els seu procés de vida satisfactòriament.

3.9. Requisits del maquinari i programari.

3.9.1. Clients

Per accedir al aplicatiu els usuaris només es necessita un ordinador on hi hagi instal·lat un navegador web.

3.9.2. Servidor

Al haver seleccionat la tecnologia J2EE com a llenguatge amb el que desenvolupa l'aplicatiu s'ha escollit com a servidor d'aplicacions JBoss que a més de tenir un gran prestigi entre la comunitat Java, aporta l'avantatge de ser de lliure distribució.

Com a SBGD s'ha escollit MySQL que suporta JDBC.

4. Disseny

4.1. Introducció.

En aquest apartat s'explica el com "s'ha de fer" tot el que s'ha explicat en l'apartat anterior.

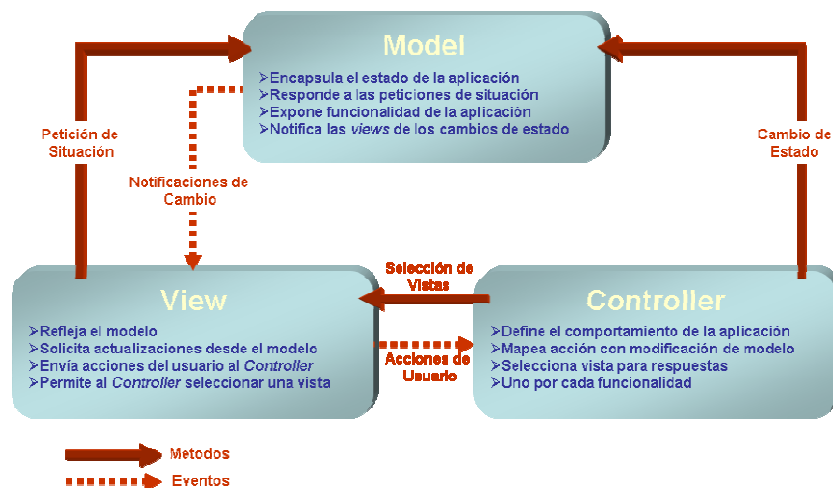
Es veurà com es relacionen les diferents classes, quins mètodes han d'implementar, com s'ha d'estructurar la base de dades on emmagatzemar les Queixes i Reclamacions, a més a més de la configuració del Workflow.

4.2. Arquitectura

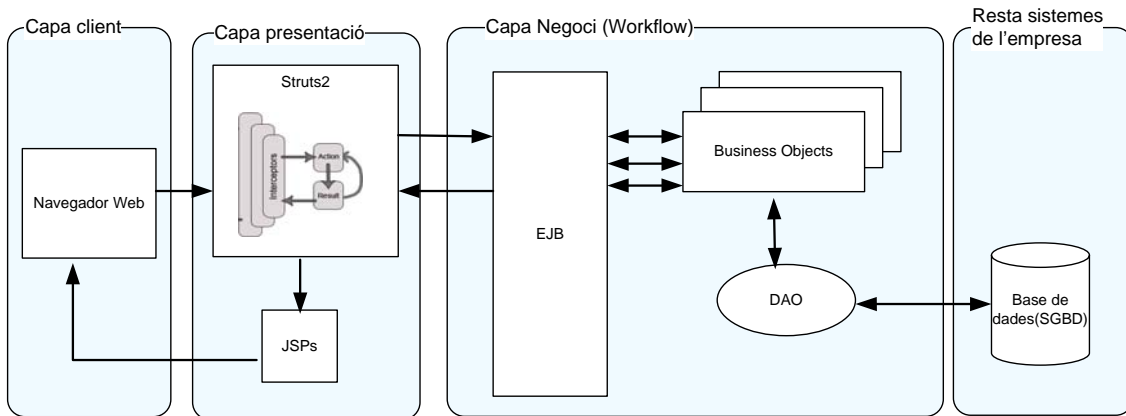
El conjunt del Workflow i el projecte web de reclamacions i queixes s'implementaran mitjançant la programació Orientada a Objectes i dins la plataforma J2EE, S'ha optat per ella, sobretot per la facilitat que dona a l'hora de comunicar diferents sistemes i en el cas que ens ocupa es fonamental; ja que ens vital que la comunicació entre l'aplicatiu i el Workflow sigui molt fluida i fins a cert punt senzilla.

Tot el projecte en conjunt ha de seguir el patró de MVC (Model Vista Controlador) usat per gairebé la totalitat d'aplicacions basades en web dinàmiques i J2EE, que permet dividir la gestió en tres tipus de components especialitzats.

- El Model és la part que representa les dades del programa. Conté les dades y controla les seves transformacions
- El controlador (Controller) interpreta els events generats per la capa de presentació (Vista) i mapejar-los vers diferents comandes que seran enviats al model.
- La Vista es correspon amb l'interfície amb que interactua l'usuari, en el nostre cas JSP que generen HTML dinàmicament.



El diagrama de les diferents capes en que esta dividit el projecte es el següent:



Tal i com es pot observar, el navegador web envia peticions al servidor on el framework Struts2 recull les dades i crida via JNDI al EJB que, a la seva vegada en funció del mètode executat, crida a un "Business Object" i aquest realitza la tota la lògica de negoci. Aquest "Business Object" decideix si ha d'anar a la base de dades mitjançant un DAO o no. Una vegada realitzades les operacions corresponents, es retornen les dades al EJB i després a la classe de tipus Action de Struts i es crida a la JSP que una vegada compilada es mostra en el navegador web.

4.2.1. Utilitat de Workflow

Dins el patró MVC el Workflow realitza el paper del Model ja que es "qui fa la feina" de qualsevol aplicatiu que s'hi connecta.

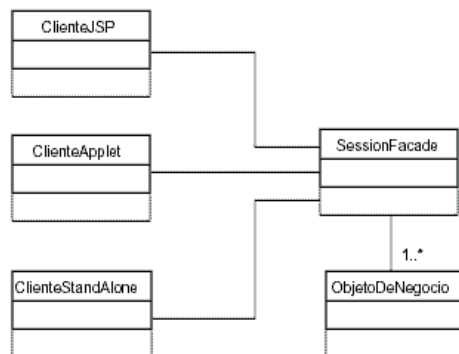
L'estratègia d'implementació d'aquesta part, esta basada en els patrons Facade, i DAO, connectats mitjançant un objecte de negoci (Business Object)

Les peticions dels aplicatius es faran sobre un EJB de sessió sense estat que compleix el paper de façana (Facade) i un DAO a través del qual s'accedeix a la base de dades.

Les principals propietats de la façana son:

- S'introdueix una capa entre els aplicatius i el Workflow
- L'interfície d'accés al Workflow es uniforme per tots els aplicatius.
- Amb l'EJB es redueixen el nombre d'invocacions.

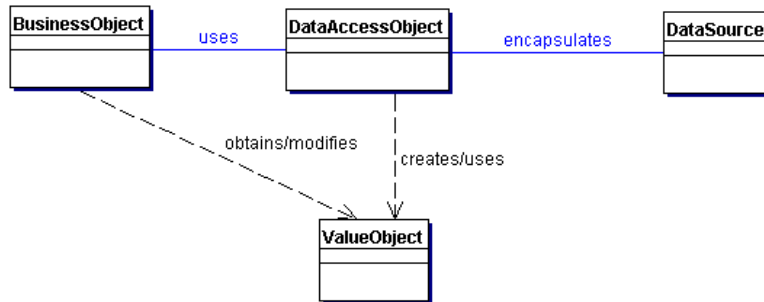
L'esquema genèric del patró Façana (Facade) és el següent:



Pel que respecta al patró DAO les seves propietats son:

- El DAO oculta els detalls de l'implementació de la base de dades i encara que aquesta canviï, els mètodes als que s'accedeix son els mateixos.

- Es pot modificar el sistema d'emmagatzematge sense tenir que canviar la lògica de negoci.



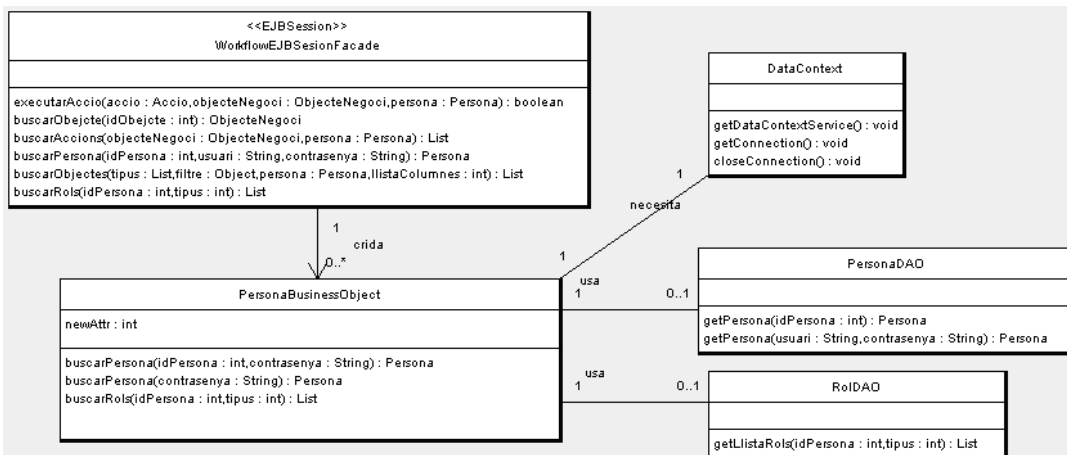
Responsabilitats de cada part:

- L'EJB en funció del tipus de petició, crida un Business Object o un altre.
- Business Object es qui conté la lògica de negoci i qui va cridant als diferents mètodes dels DAO per recuperar les dades i encapsular-les per retornar-les a l'EJB .
- Classes DAO accedeixen mitjançant sentències SQL al SGDB
- ValueObject es correspon a les entitats sobre les que es mapejen les dades de la base de dades.

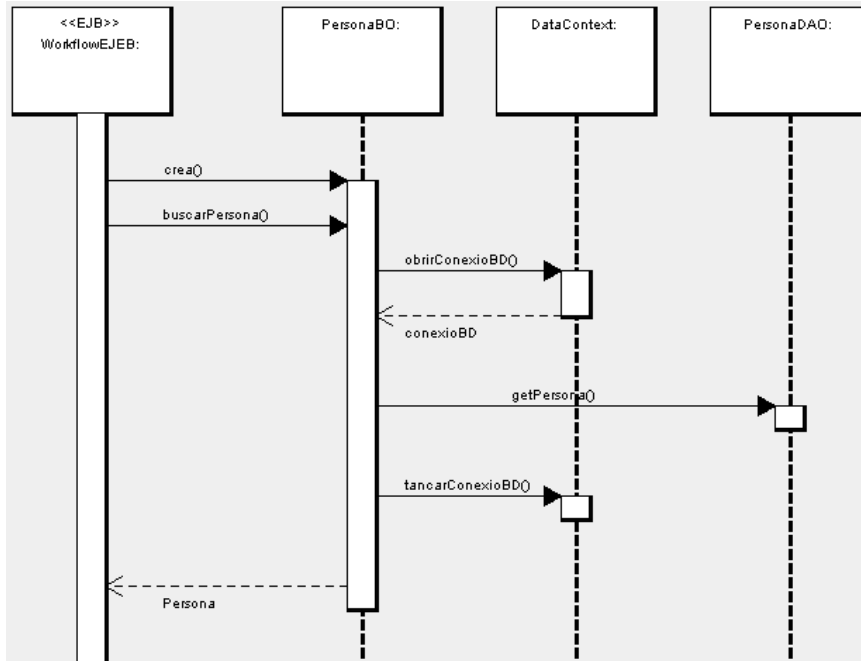
A continuació es mostren les relacions entre les classes

4.2.1.1. Buscar persona

1.1.1.1. Diagrama de classes

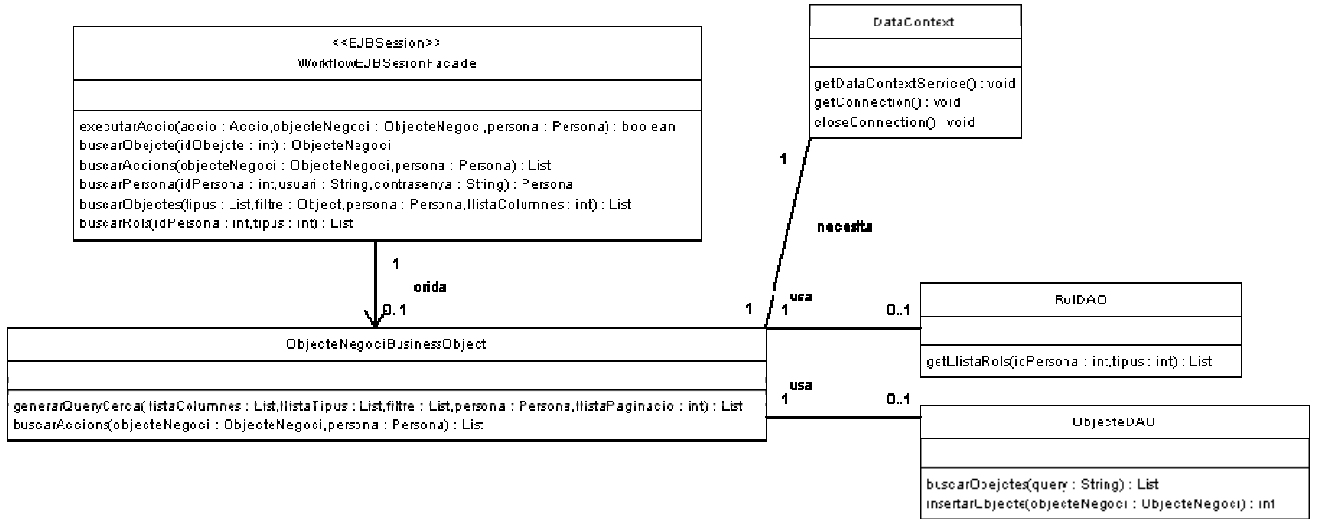


1.1.1.2. Diagrama de seqüència

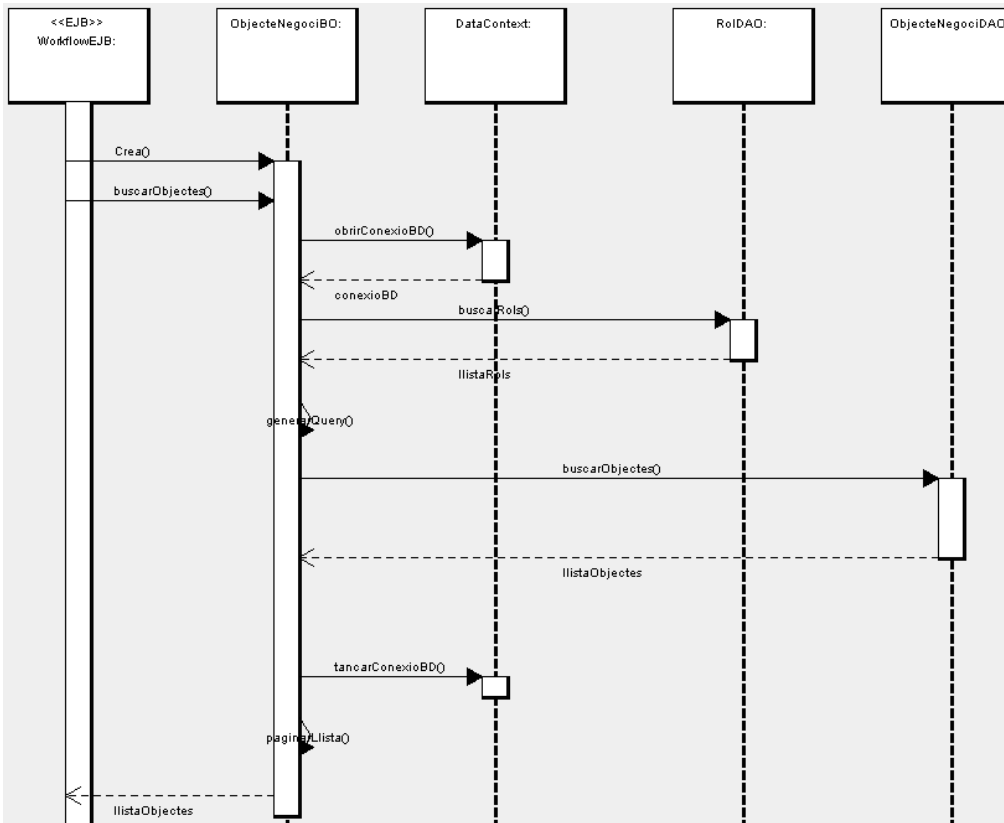


4.2.1.2. Buscar Objectes de Negoci

1.1.1.3. Diagrama de classes

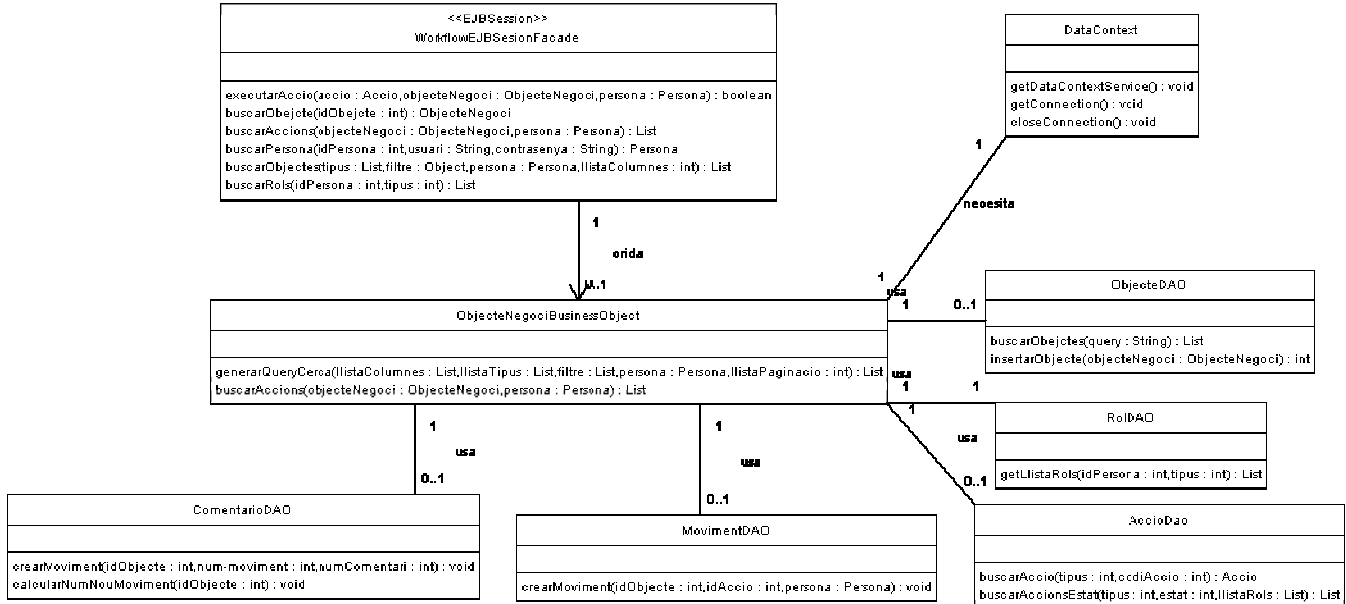


1.1.1.4. Diagrama de seqüència

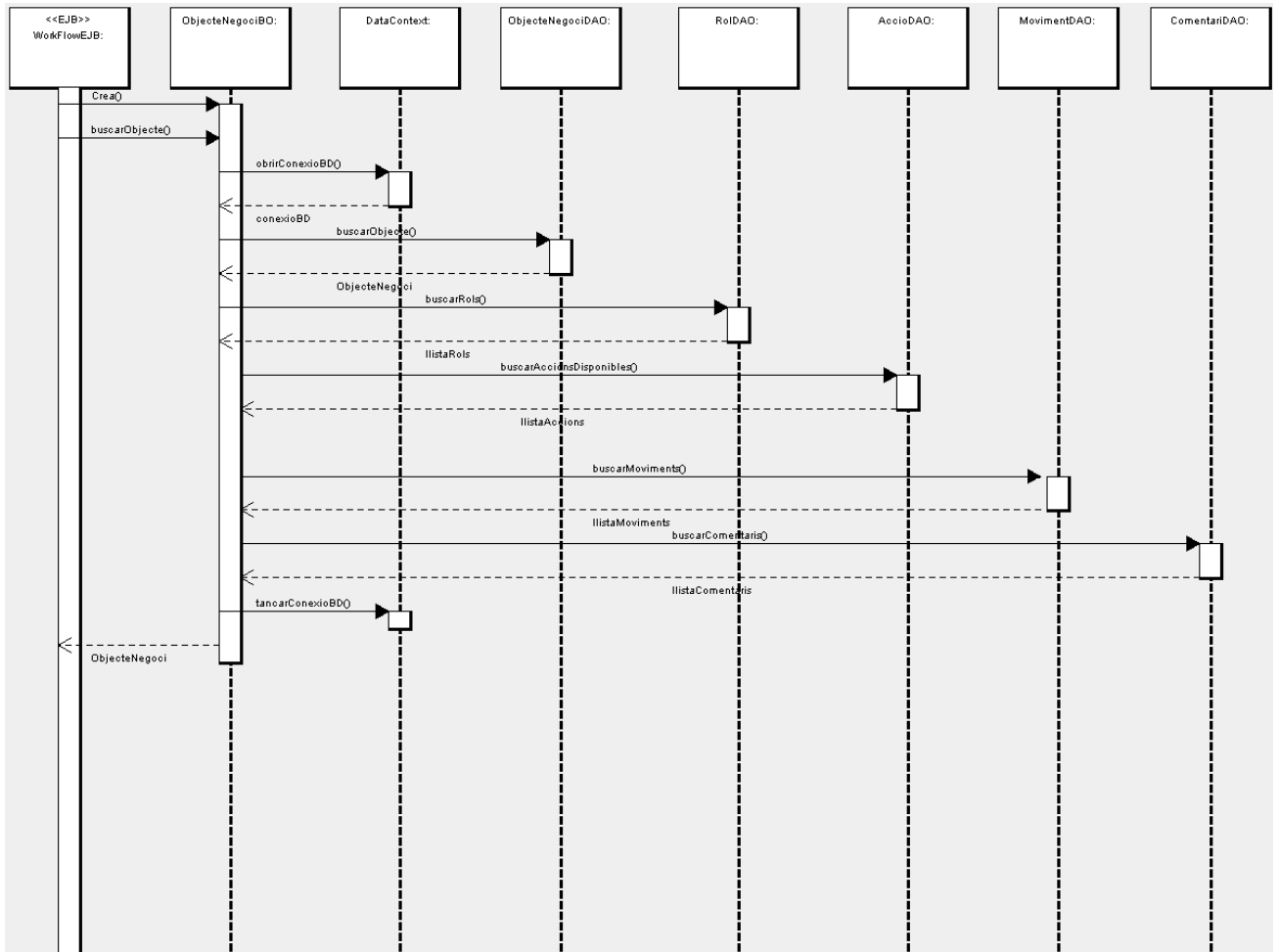


4.2.1.3. Consultar Objecte de Negoci

1.1.1.5. Diagrama de classes

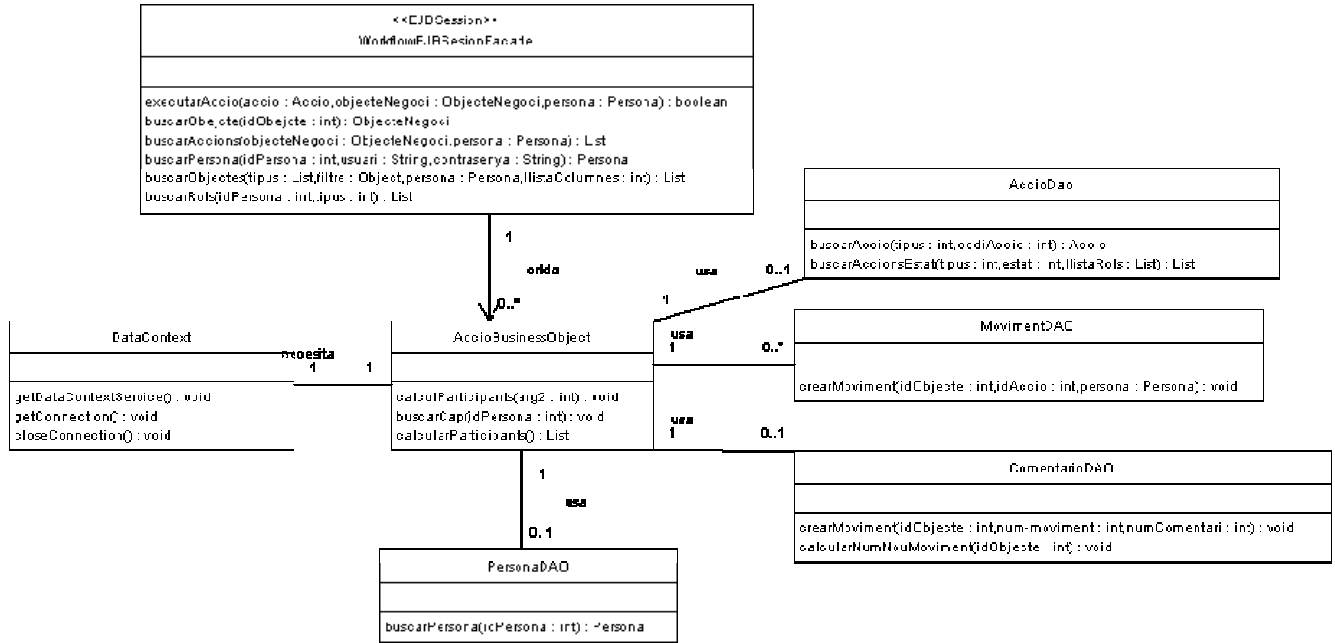


1.1.1.6. Diagrama de seqüència

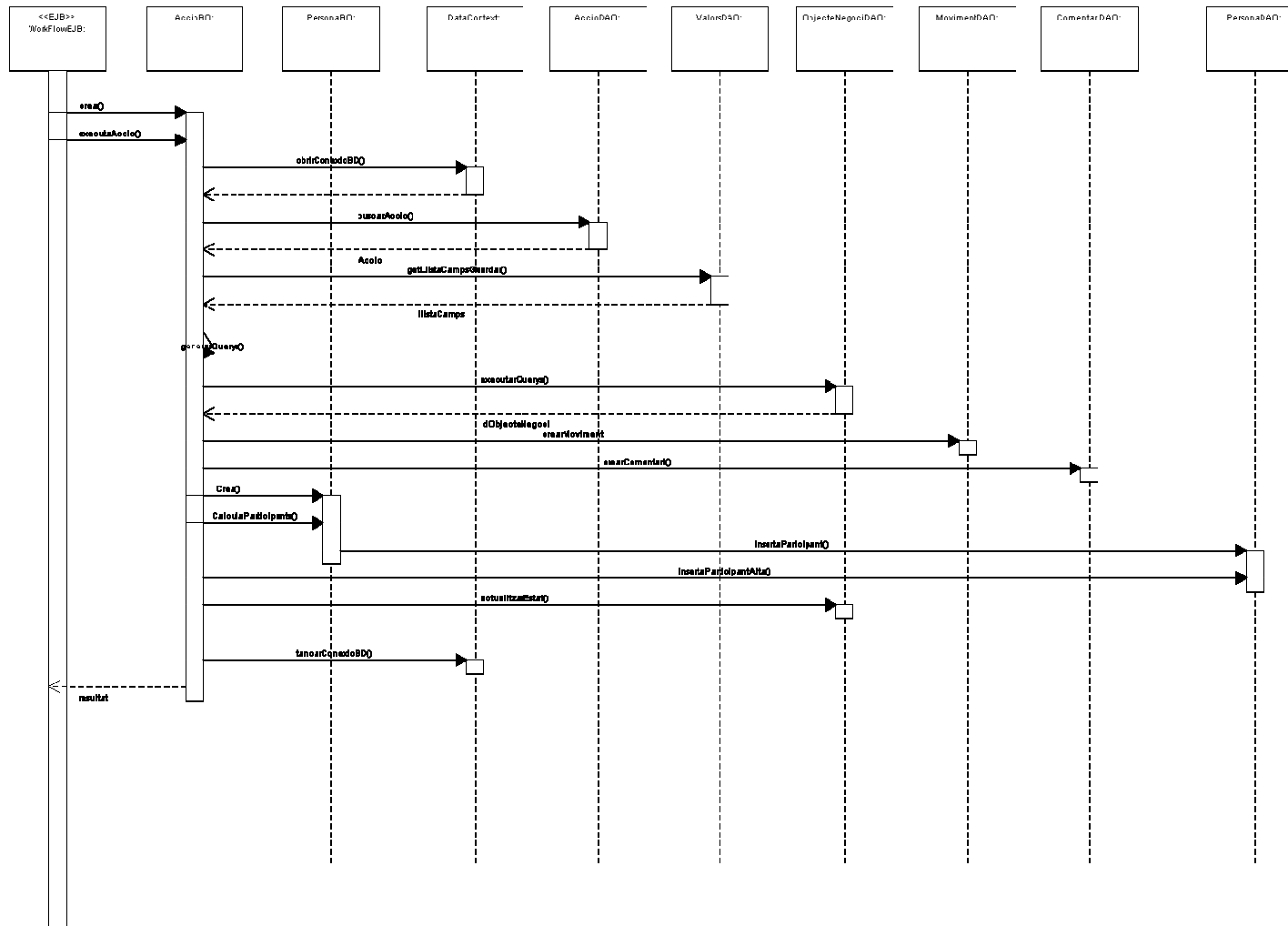


4.2.1.4. Executar una Acció

1.1.1.7. Diagrama de classes



1.1.1.8. Diagrama de seqüència



4.2.2. Aplicatiu de Reclamacions i Queixes

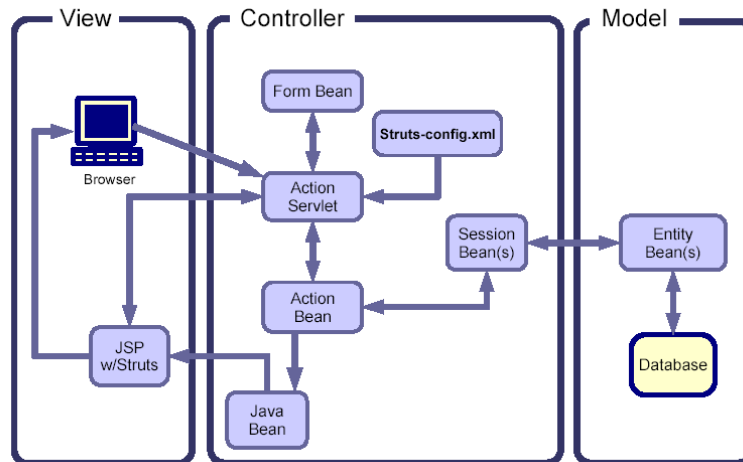
La part corresponent a l'aplicatiu de Reclamacions i Queixes es desenvolupa de la següent manera; per la part visual s'utilitza la tecnologia JSP i per la part del controlador s'aprofita de les virtuts del framework Struts.

Com a convenció la validació de les dades introduïdes a la pantalla es realitza mitjançant JavaScript, concretament, abans de fer el submit() es crida a una funció anomenada validar(). Les validacions seran a nivell de camps obligatori i valors numèrics.

1.1.2. Struts

Es una capa de control flexible que utilitza tecnologies estàndards com Servlets, JavaBeans, XML, Jakarta, etc i que implementa una variació de MVC.

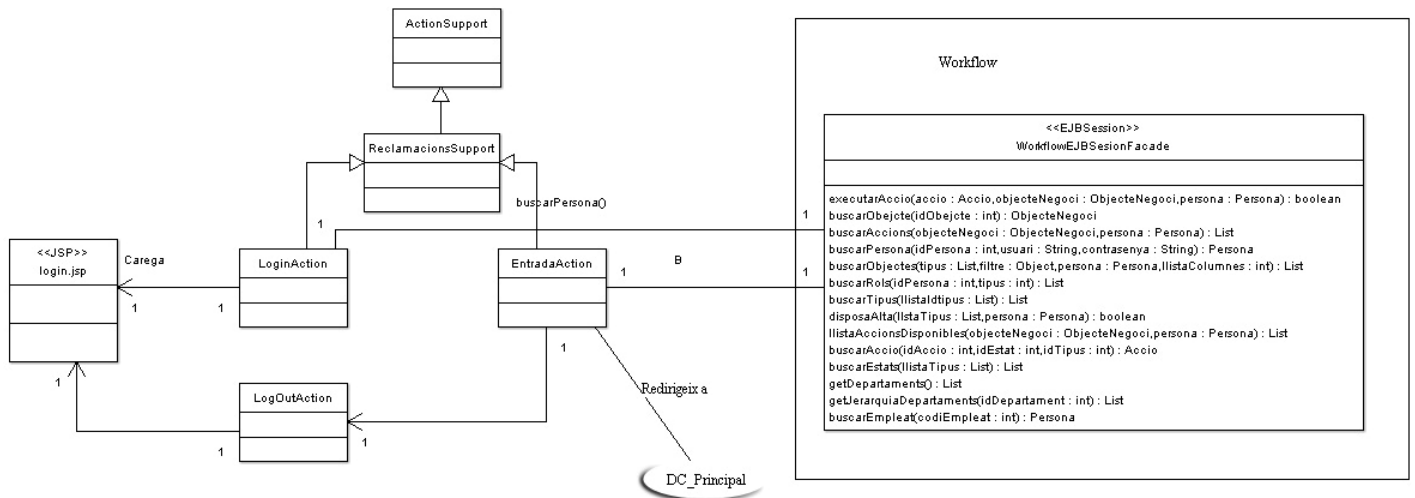
Struts proporciona el seu propi component controlador (Controller) i s'integra amb altres tecnologies per treballar amb el Model i la Vista.



Els components que s'han d'implementar per cada funcioanlitat de l'aplicatiu son:

- Els Action que hereden de ActionSupport (en la versió 1 s'anomenaven ActionForms), i que han d'implementar:
 - El mètode execute().
 - Els diferents get is set per cada un dels inputs d'un formulari d'una pàgina JSP.
- Funcionalment realitza les següents accions:
 - Obtenir els valors del request o session i introduir-los en el seu corresponent atribut.
 - Cridar als objectes de negoci del Model, en aquest el EJB de session que actua com a Facade.
 - Analitzar els resultats i retornar a un altre ActionSupport o a una JSP.
- Fitxer de configuració dels mapejos dels Action anomenat struts-config.xml. Aquest fitxer es llegit pel servlet que inclou internament Struts que es qui rep les sol·licituds de d'usuari, crida al ActionBean corresponent i segons el que retorni crida una JSP.

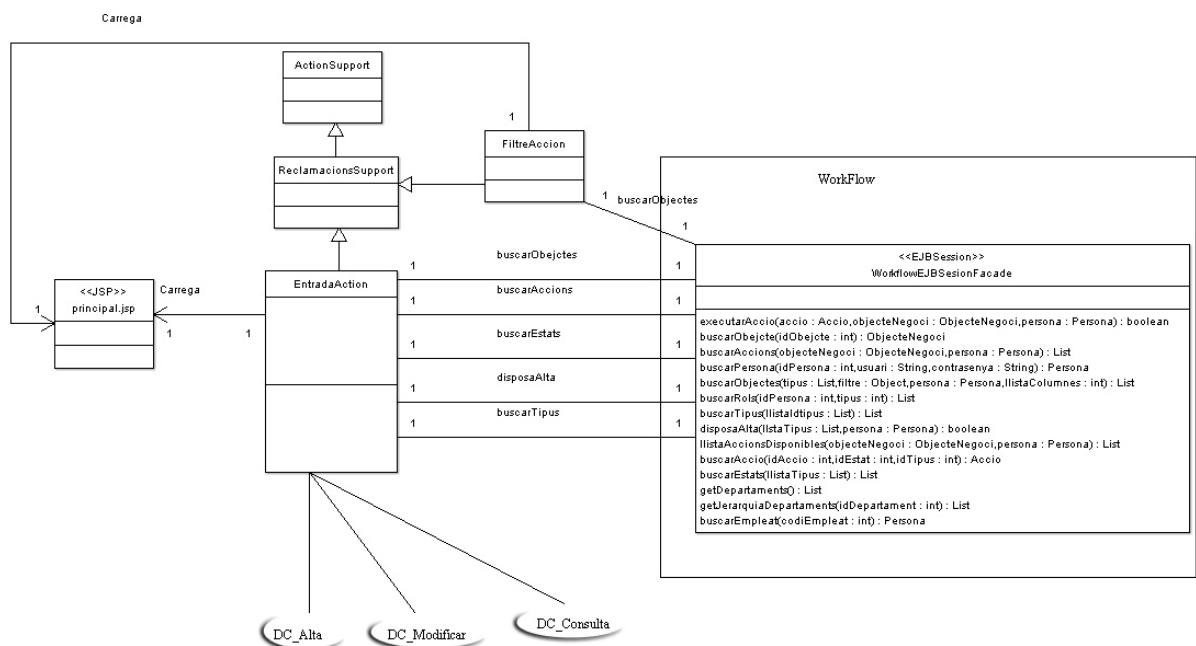
1.1.3. Entrada a l'aplicatiu



El funcionament serà el següent:

1. L'ActionSupport anomenat LoginAction, detecta que no hi ha usuari/password i redirigeix el flux vers la pantalla login.jsp.
2. Des de la pantalla de validació i una vegada l'usuari ha introduït l'usuari i la contrasenya crida al LoginAction de nou
3. Es recuperen els 2 valors introduïts, es crida al EJB del Workflow perquè busqui una persona amb aquest usuari/contrasenya.
 - 3.1. Si existeix: es redirigeix el flux cap al ActionSupport EntradaAction per tal de mostrar la pantalla principal.
 - 3.2. Si no, retorna a login.jsp amb un missatge d'error.

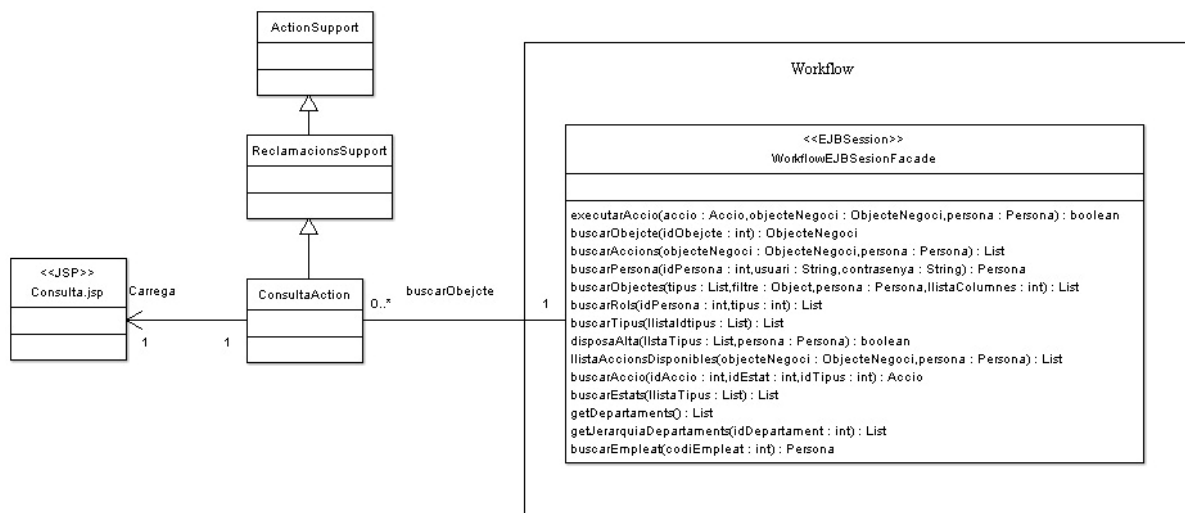
1.1.4. Pantalla principal



En aquest cas el flux és el següent:

1. La primera vegada que s'invoca a l'ActionSupport anomenat EntradaAction:
 - 1.1. Recupera genera un objecte de tipus Filtre amb les dades de les que es disposa. i crida al EJB del Workflow perquè Objectes de Negoci que compleixin el filtre.
 - 1.2. Encapsula aquests Objectes de Negoci dins una llista
 - 1.3. Torna a cridar a l'EJB per comprovar si l'usuari que s'ha connectat te disponible l'acció de Alta per tal de mostrar el link d'alta.
 - 1.4. Redirigeix el flux cap a la JSP: pricipal.jsp.
2. Si des de la JSP es produeix un event:
 - 2.1. Si es de paginació (Pagina següent, anterior, primera, ultima o pagina x) en l'ActionSupport: FiltreAction, crea un objecte de tipus filtre i crida al EJB com en el cas 1.1 i es segueix amb els 1.2, 1.3 i 1.4
 - 2.2. Si es "aplicar filtres" torna a executar el pas 2.1.
 - 2.3. Alta crida a l'ActionSupport: AltaAction
 - 2.4. Consultar a l'ActionSupport: CosultaAction
 - 2.5. Modificar a l'ActionSupport; ModificacioAction

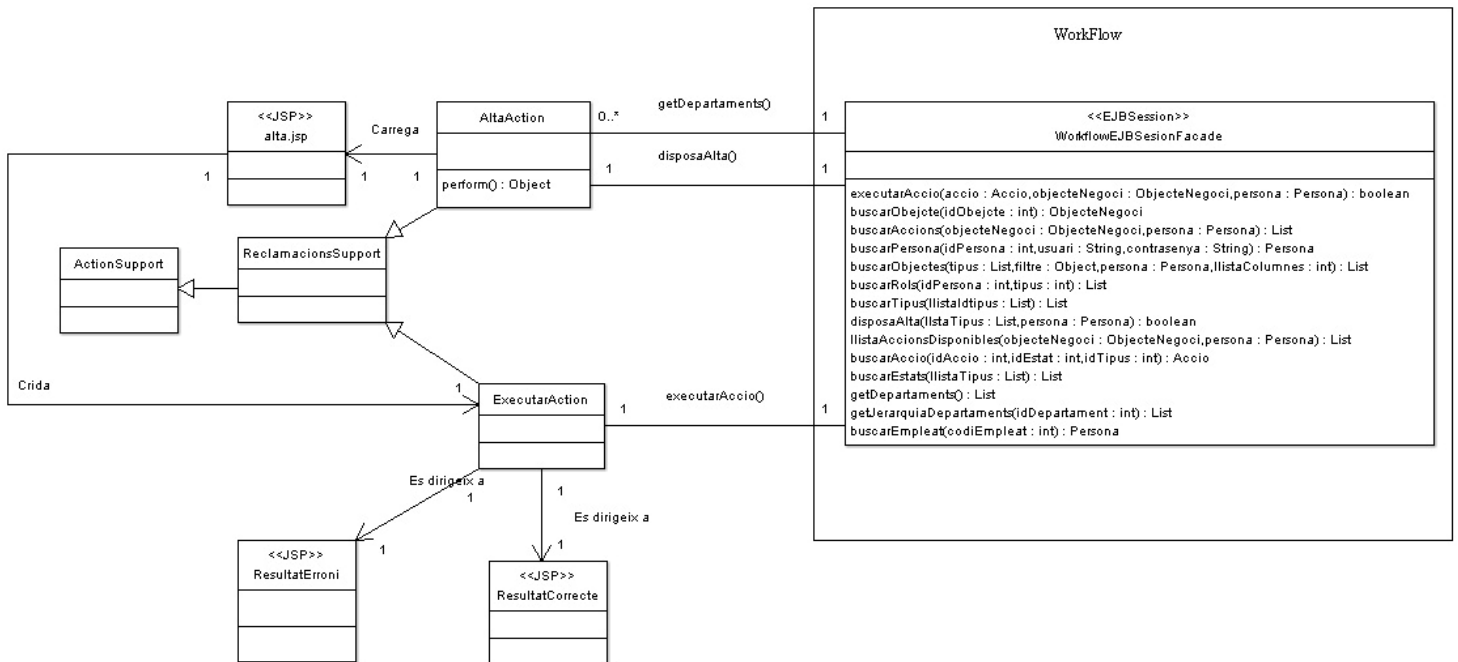
1.1.5. Consultar Objecte de Negoci



El funcionament serà el següent:

1. L'ActionSupport anomenat ConsultarAction realitza les següents accions:
 - 1.1. Recuperar de la pantalla l'identificador de l'Objecte de Negoci a consultar.
 - 1.2. Mitjançant el servei de l'EJB recupera un objecte de tipus: ObjecteNegoci
 - 1.3. Encapsula les seves dades i les encapsula dins d'un l'ObjecteNegoci.
 - 1.4. Salta a la pantalla consulta.jsp.

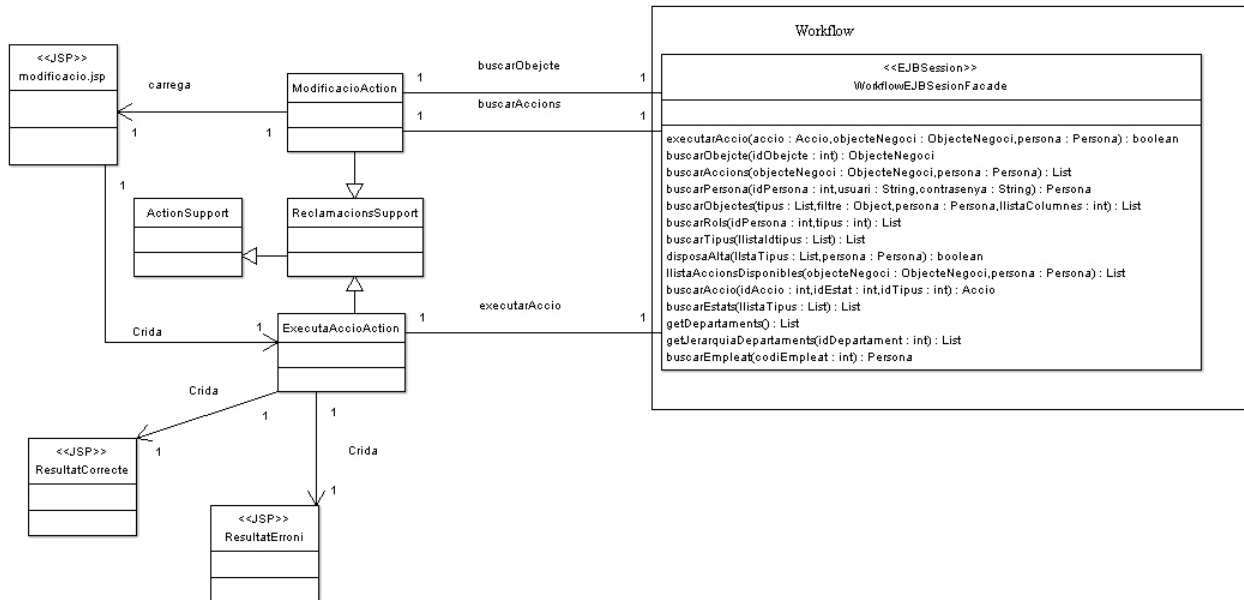
1.1.6. Alta d'un Objecte de Negoci



El funcionament serà el següent:

1. L'ActionSupport anomenat AltaAction carrega la pantalla alta.jsp
2. Quan l'usuari ha introduït les dades i selecciona l'acció "Alta", es crida a l' ActionSupport ExecutaAction.
3. ExecutaAction realitza les següents accions:
 - 3.1. Recupera i encapsula les seves dades en un objecte de tipus: ObjecteNegoci
 - 3.2. Crida al EJB del Workflow perquè executi l'acció d'alta.
 - 3.3. Segons el resultat de l'execució:
 - 3.3.1. Si es correcte carrega la pantalla ResultatCorrecte.jsp
 - 3.3.2. Si es errorni: ResultatErrorni.jsp
 - 3.3.3. Des d'aquestes pàgines s'enllaça amb la pantalla principal.

1.1.7. Modificació d'un Objecte de Negoci

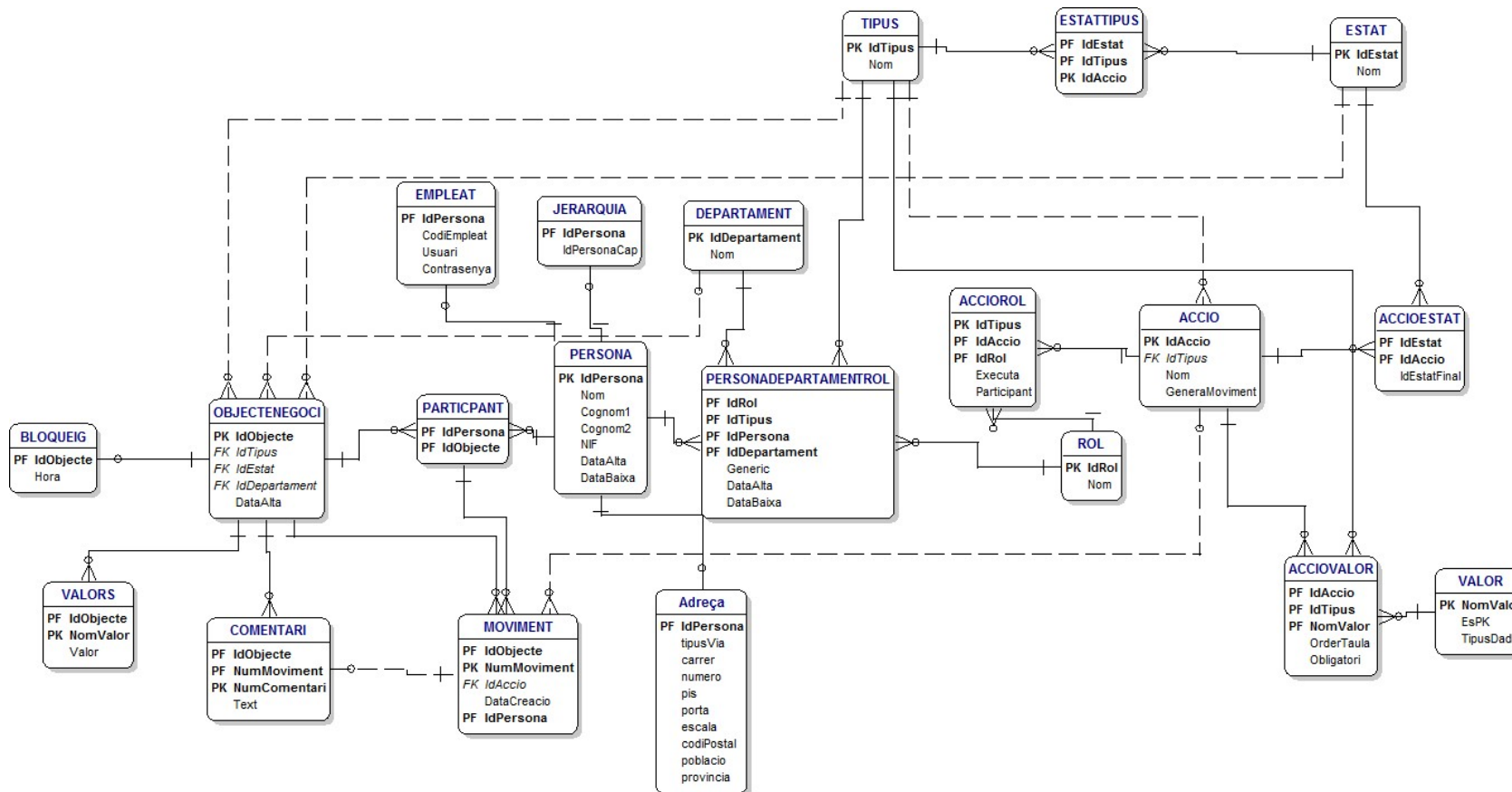


El funcionament serà el següent:

1. L'ActionSupport anomenat ModificacioAction realitza les següents accions:
 - 1.1. Recuperar de la pagina principal l'identificador de l'Objecte de Negoci a consultar.
 - 1.2. Mitjançant el servei de l'EJB recupera un objecte de tipus: ObjecteNegoci
 - 1.3. Encapsula les seves dades i les retorna a ModificacioAction.
 - 1.4. També a través del servei buscarAccions, recupera la llista d'accions disponibles
 - 1.5. Finalment salta a la pantalla modificacio.jsp.
2. Quan l'usuari ha introduït les dades i selecciona l'acció corresponent, es crida a l'ActionBean ExecutaAccioAction.i realitza les següents accions:
 - 2.1. Recupera i encapsula les seves dades en un objecte de tipus: ObjecteNegoci
 - 2.2. Crida al EJB del Workflow perquè executi l'acció corresponent.
 - 2.3. Segons el resultat de l'execució:
 - 3.3.4. Si es correcte carrega la pantalla ResultatCorrecte.jsp
 - 3.3.5. Si es erroni: ResultatErroni.jsp
 (Des d'aquestes pàgines s'enllaça amb la pantalla principal)

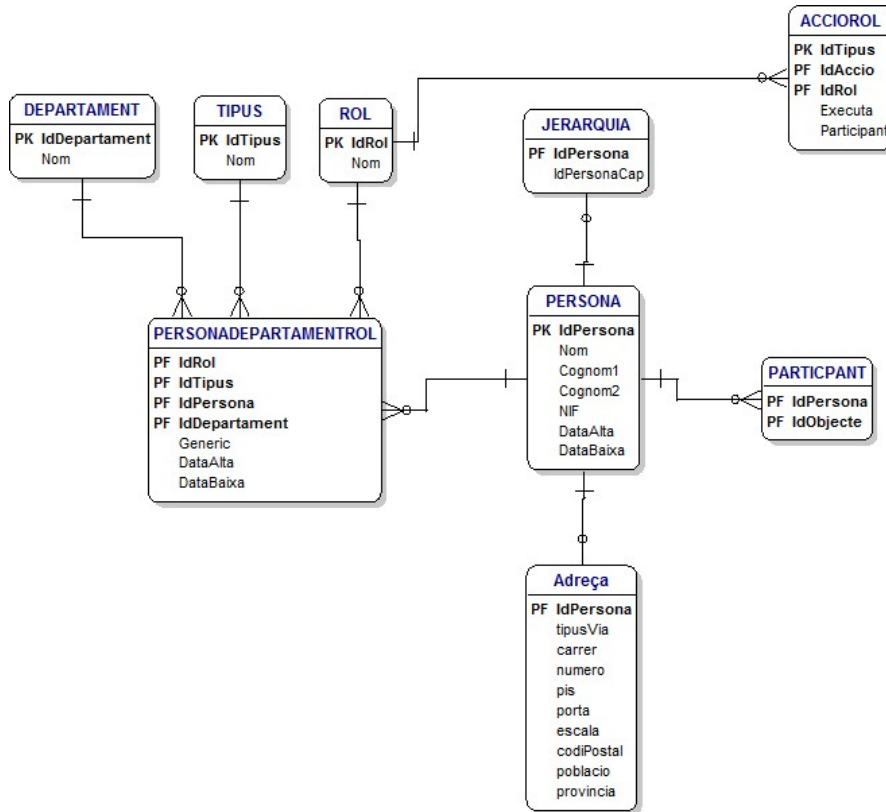
4.3. Disseny de la persistència

Per la persistència s'ha optat per dissenyar una base de dades relacional on es mapeja cada una de les entitats de l'anàlisi en una taula tal i com es mostra en el següent diagrama.

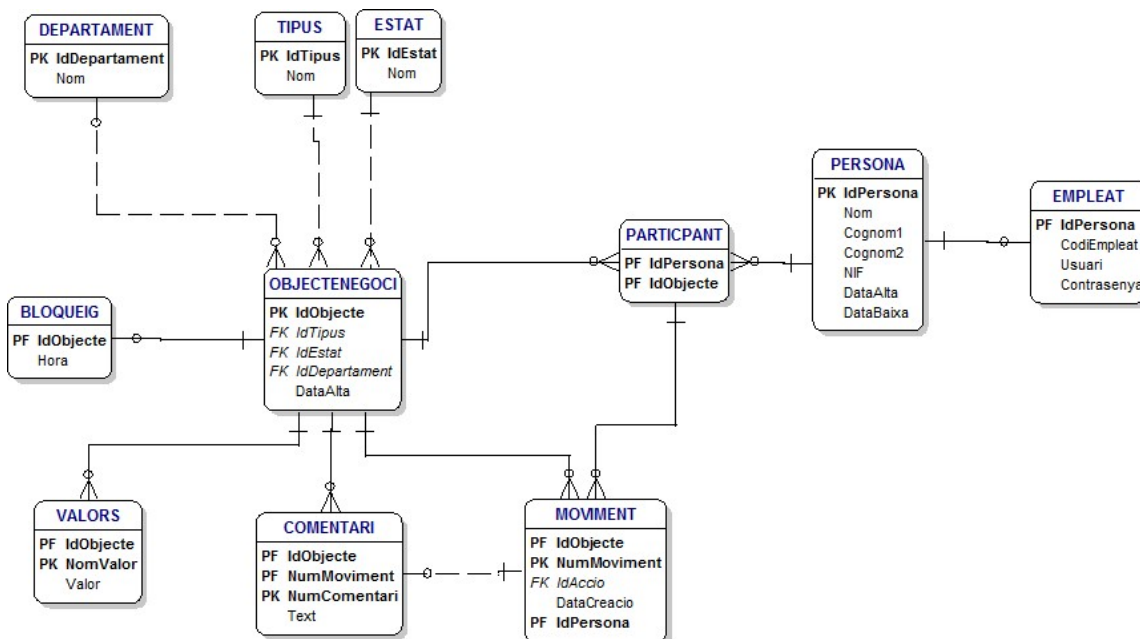


Tal i com s'ha fet amb els diagrames d'entitats es mostra cada un dels subdiagrames següents:

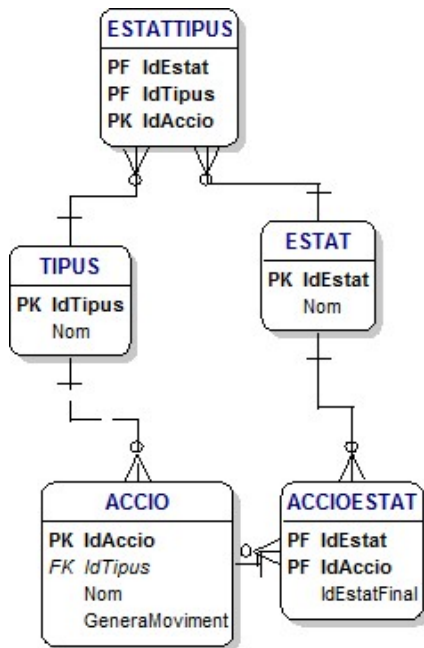
4.3.1. Persistència de Persones



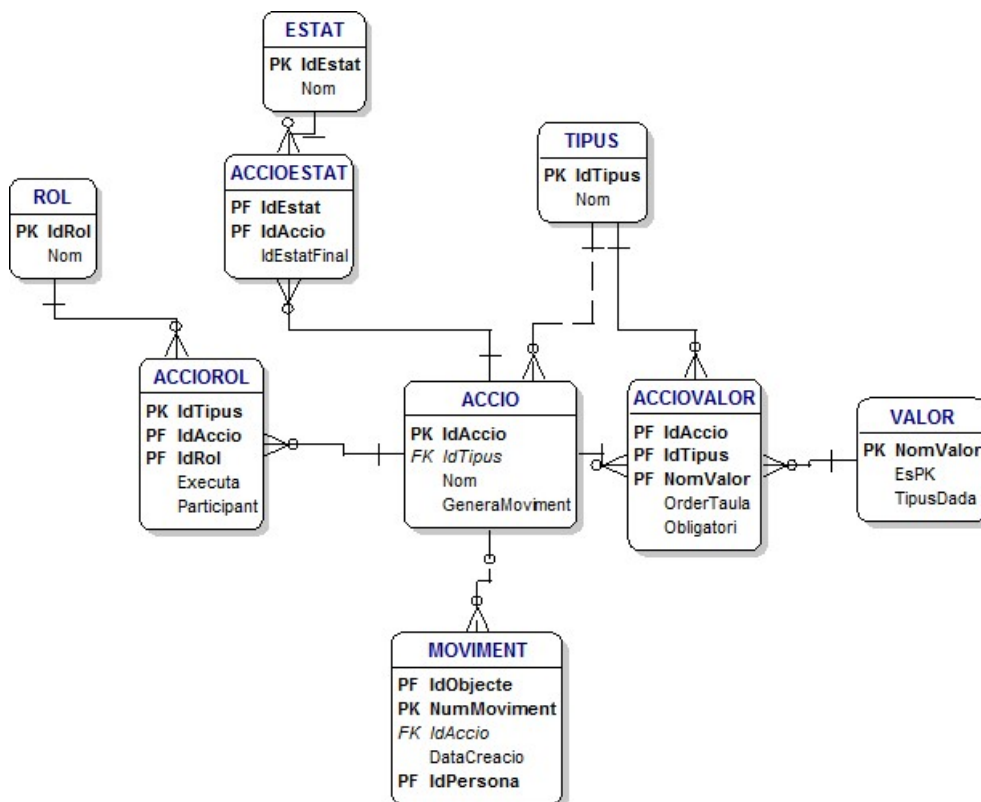
4.3.2. Persistència de l'Objecte de Negoci



4.3.3. Persistència d'Estats



4.3.4. Persistència d'Accions



5. Implementació

5.1. Requeriments de programari.

El requeriments de programari per implementar Per implementar el sistema utilitzarem el següent programari i configuracions:

- Qualsevol navegador web que suporti les especificacions d'HTML 4.01 i JavaScript.
- Plataforma J2SE 1.6.0 (SDK).
- Servidor d'aplicacions JBoss 4.2.2 que incorpora suport per EJB v2 i v3.0. (Inclou servidor web Tomcat)
- Framework Apache Struts 2, v2.0.9.
- SGBD: Servidor de base de dades MySQL v5.0.45 (Distribuït sota llicència GNU GPL Llicència Pública General)
- Connector/J v5.0.8 que es el connector JDBC oficial per MySQL.

Eines de disseny.

Per l'anàlisi s'han utilitzat:

- ArgoUML v.024. Eina CASE pel modelat de sistemes basat en diagrames UML (programari lliure i independent de la plataforma).
- DeZign for Databases V4 . Eina pel disseny i modelant de la base de dades, llicència comercial i s'ha usat un llicència de proves per 30 dies.

Eines de desenvolupament.

El desenvolupament s'ha realitzat amb les següents eines :

- Utilitzem com a IDE, Eclipse 3.3.0. (programari lliure i open source).
- Plugin JBoss JBossIDE-1.6.0-GA. (Permet deployar i gestionar el servidor JBoss des d'Eclipse)

6. Valoració econòmica

Dins dels objectius d'aquest TFC no hi ha l'avaluació econòmica del projecte, ni de temps dedicat ni en costos de programari, ni maquinari.

Tant el programari utilitzat com les llibreries o frameworks utilitzats, son gratuïts excepte el DeZign for Databases (l'he utilitzat perquè ja el coneixia i m'ha permès estalviar unes hores en aprenentatge però es pot substituir per alguna altra eina equivalent. Sobre les particularitats de cada un dels programes, es pot consultar la seva pagina web corresponent.

A més a més, en cas no usar-se un sistema operatiu no pertanyent a Microsoft, tots tenen la versió Windows i Unix disponible.

7. Conclusions

La tria de realitzar en aquest TFC d'una aplicació basada en entorns distribuïts ha vingut sobretot motivada per la fascinació que em provoca Internet i les seves grans possibilitats, tal i com s'ha vist amb empreses com Google, projectes altruistes com Wikipedia per citar-ne només dos.. A més a més de ser una de les àrees amb mes futur professional.

El fet de basar-ho en la tecnologia J2EE s'ha degut a varies raons. La primera fa referència a que em permetia posar en pràctica la majoria de coneixements adquirits durant els estudis a la UOC (Orientació Objectes, disseny, enginyeria de programari, etc.). El segon és el fet de ser una de les tecnologies mes acceptades dins la comunitat d'Internet i del mon empresarial.

A nivell de coneixements tot i no començar de zero perquè havia treballat uns mesos quan Java anava per la versió 1.1.8, val a dir que m'ha sorprès gratament la gran evolució del llenguatge i per tot el que vist el seu creixement encara no ha tocat sostre, ni molt menys.

Professionalment, m'ha permès realitzar totes les etapes d'un desenvolupament, començant per la recollida de requeriments fins al desenvolupament final, passant per l'anàlisi i el disseny. Una de les coses que més m'ha impactat, ha estat la quantitat de frameworks disponibles per ajudar en el desenvolupament: Velocity, Cayenne, Ant, Maven, etc. Entre tots ells he decidit usar Struts, concretament la versió 2, investigant sobre la primera versió vaig trobar que tenia un gran prestigi i com que des d'Octubre del 2007 van alliberar la versió 2, vaig decidir adoptar-la.

8. Glossari

Acció	Transició entre dos estats.
Codi usuari	Codi alfanumèric que identifica inequívocament un empleat.
Contrasenya	Cadena alfanumèrica secreta que juntament amb el codi d'usuari permet identificar un usuari.
Codi empleat	Numero únic i seqüencial que identifica un empleat.
Empleat	Treballador de l'empresa Joguines AC.
Moviment	Registre de l'execució d'una acció sobre un objecte de negoci.
Objecte de Negoci	Objecte genèric amb que treballa el Workflow. Els seus atributs indispensables perquè funcioni el Workflow estan dins l'entitat ObjecteNegoci i els atributs concrets dels diferents tipus, es guardaran en format de llista de clau/valor dins l'entitat valors.
Participant	Persona que ha intervingut en el cicle de vida d'un Objecte de Negoci, tant pot ser perquè ha executat una acció, perquè ha estat incorporat com a destinatari o perquè ha estat la persona origen de la queixa o reclamació.
Queixa	S'entenen com a queixes les manifestacions dels interessats referents al funcionament dels serveis proporcionat, tardances, desatencions, etc., en relació a interessos i drets legalment reconeguts, i derivats de contractes, normatives de transparència , de protecció dels clients, d'incompliment de bones practiques.
Reclamació	En una relació es un queixa en que es posa de manifest la pretensió d'obtenir la restitució d'un interès o dret en relació a interessos o drets legalment reconeguts, derivats de contractes, normatives de transparència , de protecció dels clients, d'incompliment de bones practiques
Rol Genèric	Rol que disposa de permisos d'accés a tots els objectes independentment de si és participant en algun d'ells.
Workflow	Utilitat que automatitza la seqüència d'accions necessàries per l'execució d'un procés.

9. Bibliografia

ARMOUR F, GRANVILLE, M. *Advanced Use Case Modeling*. Addison Wesley

BOOCH, G; RUMBAUGHT,J:JACOBSON,I. *El lenguaje unificado de modelado*. Ed. Adison Wesley

CONALLEN, JIM. *Building Web Applications with UML*. Addison Wesley

D'SOUZA, D, CAMERON WILLS A. *Objects, Components, and Frameworks with UML*. Addison Wesley

CRAIG LARMAN. *UML y Patrones*. Ed.Prentice Hall.

CRAIG LARMAN. *Applying UML and Patterns*. Prentice Hall

DEEPAK ALUR, JOHN CRUPI, DAN MAKES. *Core J2EE Patterns, best practices and design strategies*. Ed. Sun Microsystems.

MONSON-HAEFEL, RICHARD. *Entreprise JavaBeans*. O'Reilly

JOHNSON, ROB. *J2EE Design and development*. Wrox Press.

STRUTS 2 TRAINING, [en línia]. <<http://www.roseindia.net/struts/struts2/>> [consultat novembre - desembre 2007]

STRUTS 2 - Step by step guide for the programmers , . [en línia]. <<http://techienet.org/book-page/sarath/struts2-step-step-guide-programmers> > [consultat octubre - novembre 2007]

SUN MICROSYSTEMS, . [en línia]. *J2EE Blueprints*
<<http://java.sun.com/blueprints/corej2eepatterns/Patterns/>> [consultat octubre - novembre 2007]

10. Annex 1. Manual d'instal·lació

Aquest document s'acompanya d'un manual d'instal·lació dels aplicatius en un document adjunt.