

Projecte Fi de Carrera. *COMPTAXM*

**Anàlisi, disseny i implementació d'una aplicació distribuïda
Client / Servidor basada en tecnologia Java. J2EE**

Xavier Martí Moret

Enginyeria en Informàtica de gestió

Albert Grau Perisé

14 de gener de 2013

Agraïments

Fa tres anys i mig em vaig embarcar en aquesta aventura. Tot primer amb uns objectius a llarg termini, esperant a veure els resultats obtinguts i anant pas a pas. En tot aquests temps haig de dir que em prou feines m'he adonat de l'aconseguit fins avui.

A punt d'acabar aquesta etapa de la meva vida, puc assegurar que he gaudit moltíssim perquè he assolit els objectius proposats i sobretot he après, **he adquirit coneixements**. Així un agraïment sincer a tots els consultors/es, estudiants/es que m'he trobat en aquest viatge. Puc dir que en tots i cadascun d'ells sempre he trobat recolzament i coneixement.

També un agraïment a totes les persones que han fet possible que hagi pogut invertir tant de temps en aquest llarg projecte.

I finalment un sincer agraïment, a l'Albert, el consultor del TFC. He seguit els seus consells fil per randa, i crec que ho he aconseguit, tot i les meves pors al iniciar aquest treball.

“Gaudefix mentre implementes la solució, un problema que es veu negre a la nit, deixa'l, l'endemà trobaràs la solució”

Un gran consell.

Resum

El present document pretén ser la memòria del meu treball final de carrera de l'enginyeria Tècnica de Gestió informàtica, dins l'àrea de J2EE.

Es tracta d'un aplicatiu client/servidor, per donar solució a diferents centres d'una empresa per tal de poder registrar i centralitzar les diferents dades comptables de cada centre mitjançant un servidor d'aplicacions Web.

L'objectiu principal es integrar les dades i poder donar una visió comptable global de l'empresa.

El projecte s'ha dut a terme, en tres fases, al llarg del segon semestre d'any 2012.

- Una primera fase on es detecta el problema a solucionar a grans trets i es realitza un pla d'operacions general a seguir.
- Una segona fase on s'analitza en més detall el desenvolupament del projecte, es detecta les possibles solucions, i s'estudia les eines tecnològiques candidates a ser emprades per aconseguir el propòsit, a la segona part d'aquesta fase es realitza el disseny del projecte, tot definint els actors principals, els sistemes, els casos d'ús implicats, es dissenya el diagrama de classes, la persistència de dades, el prototip de la interfície gràfica i es comença la corba d'aprenentatge sobre les eines tecnològiques a emprar.
- Finalment durant la tercera fase es dur a terme la implementació del disseny i la posterior posta punt de l'aplicatiu tot fent les proves de posada en servei i detecció d'anomalies.

La implementació s'ha dut a terme mitjançant el llenguatge Java, usant l'arquitectura J2EE, amb la aplicació del patró MVC basat sobre STRUTS2.

He fet ús del Framework Tiles, on amb la aplicació de les seves plantilles m'ha proporcionat un estalvi en quant a la duplicat de codi.

Finalment, també he fet ús del Framework Hibernate per tal de poder separar l'accés a les dades de la lògica del negoci i la presentació.

A la pràctica puc dir que he implementat quasi la totalitat de l'especificat en el disseny, amb tot en algun cas, he hagut de redissenyar alguna especificació no del tot correcte o no contemplada en l'etapa de disseny.

Índex

Agraïments	2
Resum	3
Il·lustracions	6
1.Introducció	7
1.1 Objectius.....	7
1.2 Planificació.....	8
1.3 Resultat obtingut.....	10
1.4. Descripció del contingut tècnic d'aquesta memòria.....	10
2.Anàlisi i especificació dels requeriments del cas	11
2.1 Introducció.....	11
2.2 Diagrama de paquets	11
2.3 Actors.....	12
2.4 Casos d'us i diagrames de seqüència. Per cada Subsistema	12
2.5 Lògica de funcionament	29
2.6 Funcionalitats futures	31
2.7 Seguretat	31
3.Disseny	33
3.1 Arquitectura del sistema i programari emprat.	33
3.2 Diagrama de classes	36
3.3 Diagrama d'estats	39
3.4 Disseny de la persistència.....	40
3.5 Disseny de la interfície gràfica.....	41
4.Implementació	44

4.1 Desviacions de la implementació respecte al disseny	45
5.Conclusions.....	52
Annex 1. Patró MVC via STRUTS2.....	53
Annex 2. Sobre el control de la sessió.....	55
Altres usos de col·lecció Map Session	57
Annex 3. Implementació de la internacionalització.....	60
Recursos Emprats.....	62
Glossari de termes	63
Bibliografia consultada	66

II-Il·lustracions

Figura 1. Diagrama de Gantt	9
Figura 2. Diagrama de Paquets.....	12
Figura 3. Cas d'us subsistema de Connexió.....	13
Figura 4. Diagrama de Seqüència. Login Usuari.....	17
Figura 5. Diagrama de seqüència. Afegir Usuari	18
Figura 6. Cas d'us Subsistema de manteniment.....	19
Figura 7. Diagrama de Seqüències. Alta Compte	23
Figura 8. Diagrama de seqüència. Baixa Compte.	23
Figura 9. Cas d'ús Subsistema de rebuts i informes.	25
Figura 10. Diagrama de Seqüència. Entrar nova Factura	28
Figura 11. Diagrama de Seqüència. Veure Informe Global.	29
Figura 19. Diagrama de Classes.....	37
Figura 23. Diagrama d'estats. PeriodeComptable	39
Figura 24. Diagrama de la persistència.....	40
Figura 25. Accés seqüencial i funcional	43
Figura 26. Estructura espai de treball, Eclipse	45
Figura 27. Estructura Espai de treball WorkBench MySql	45
Figura 28. Afegir Nou Proveïdor	46
Figura 29. Prototip Interfície Gràfica.	47
Figura 30. Gestió Centres.....	47
Figura 31. Afegir i/o modificar Centres.....	48
Figura 32. Missatge de confirmació	48
Figura 33. Gestió nous Períodes Comptables.	49
Figura 34. Gestió nous Períodes Comptables. Filtre	49
Figura 35. Servei d'accions predeterminades.....	50
Figura 36. Funcionalitat Multi Idioma.....	51
Figura 38. FilterDispatcher. Struts2. Editorial RAMA	53
Figura 39. Codi Interceptor	55
Figura 40. Codi Interceptors. Struts.xml.....	56
Figura 41. Implementació SessionAware	57
Figura 42. Session.put.....	58
Figura 43. Session.get	58
Figura 44. Interceptor i18n	60
Figura 45. Fitxers de configuració idiomes.	60
Figura 46. Com canviar el RequestLocale.	61

1. Introducció

La principal raó per escollir aquesta àrea, J2EE, desenvolupament d'aplicacions distribuïdes en un entorn Web, és en primer lloc, consolidar els coneixements adquirits durant tota l'enginyeria, sobre l'especificació, disseny i codificació sobre Java d'un projecte i en segon terme poder conèixer i aprendre noves maneres d'implementar i codificar el disseny.

Això m'ha permès adquirir coneixement en eines de desenvolupament sobre l'arquitectura Java i en concret Java Enterprise Edition, la qual considero que avui en dia és una arquitectura molt valorada en el camp de desenvolupament de programari distribuït.

Amb tot, em considero un simple iniciat en aquesta tecnologia, i crec que em queda molt camp per recórrer però m'ha permès realitzar un molt bon punt de partida.

Aclariment de l'autor. En la present memòria es parla sovint de l'objecte 'Factura' i l'entitat 'Comanda'. En aquest context, la memòria, els dos objectes fan referència al mateix objecte. En alguns casos s'usa el nom 'comanda' i d'altres el nom 'factura'. Aquest fet respon a que l'aplicació està codificada d'una manera (l'objecte comanda), però la presentació, o vista, fa referència a 'Factura'. Vaig creure convenient anomenar factura (un rebut amb valor real) per no confondre als usuaris, pel que fa referència a la vista.

1.1 Objectius

En l'actualitat la meva tasca professional és desenvolupa en una empresa que té diferents centres d'explotació repartits en una àrea, comarca, país, en concret.

Una de les tasques encomanades, entre d'altres, és la realització de petits programes informàtics, realitzats en Visual Basic Application, sobre fulls de càlcul, per tal de automatitzar certs processos d'introducció i enviament de dades comptables dels diferents centres de l'empresa i poder-los reunir i centralitzar per obtenir una visió comptable mensual i conjunta de l'organització.

L'objectiu professional, d'aquest TFC, és obtenir un prototip per poder donar solució a la tasca d'integrar les dades comptables, de manera fiable, ràpida i sobretot evitar els errors de duplicitat de dades. I pugui ser un punt de partida per poder desenvolupar un projecte de més envergadura.

L'objectiu acadèmic i principal, és adquirir coneixements sobre la programació multicapa i distribuïda, aplicant les tècniques que ofereix la arquitectura J2EE tot utilitzant els Frameworks Struts, Tiles i hibernate. Eines fins a dia d'avui, completament noves per mi.

1.2 Planificació

La planificació del TFC s'ha dut a terme dins el pla establert de l'assignatura amb la execució i consolidació de les diferents fites/pacs.

En general he seguit el pla establert des de l'inici. Amb tot he hagut de dedicar molts més esforços a la corba d'aprenentatge de les diferents eines de desenvolupament de programari utilitzades (Struts2, Tiles i Hibernate), de les quals no tenia cap coneixement previ.

La meua por a no poder aconseguir els objectius esperats ha fet que des de l'inici del TFC he anat dedicant temps i esforços a la tasca d'aprenentatge, i és l'única tasca que es desvia de la planificació inicial, on he dedicat moltes més hores de les planificades.

Dates Clau

Data	Fita	Descripció
21-09-12 a 03-10-12	PAC1. Pla de treball	Elaboració del pla de treball. Descripció, objectius i planificació. Inici de la corba d'aprenentatge.
04-11-12 a 08-12-12	PAC2. Especificació i disseny	Estudi de les especificacions, realització del disseny del projecte i seguir la corba d'aprenentatge.
09-12-12 a 17-12-12	PAC3. Implementació.	Implementació, codificació i testing del projecte
18-12-12 a 14-01-13	PAC4. Memòria	Elaboració de la memòria i la presentació del projecte

Memòria Projecte. COMPTAXM Xavier Martí Moret

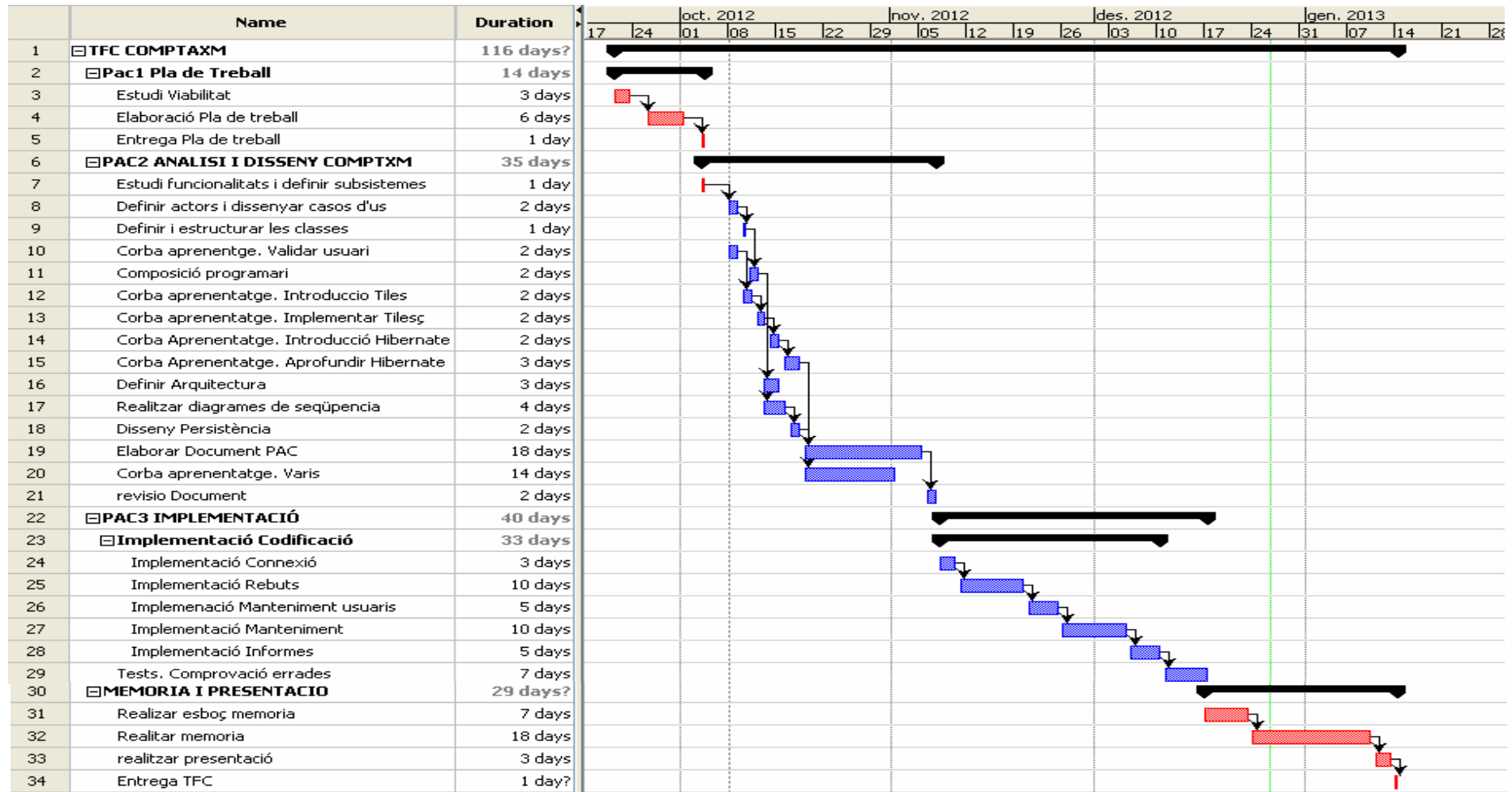


Figura 1. Diagrama de Gannt

1.3 Resultat obtingut

Tal com he comentat a la introducció és un aplicatiu que pretén donar solució a una central de dades comptables.

El producte obtingut es pot contemplar com una primera versió, un prototip per mostrar al possible client, on es podrà mostrar les diferents funcionalitats del programari:

- Configuració del sistema. Manteniment d'usuaris, centres, comptes, períodes Comptables i proveïdors.
- Entrada de dades al sistema. Dades previstes (previsions) i dades reals (factures).
- Dues sortides de dades, com a mostra. Informes (per període Comptable i per centre d'exploració).

La finalitat és donar una solució al client i aquest pugui veure una oportunitat i poder desenvolupar una aplicació completa d'acord amb les seves especificacions, sobretot en quant a la presentació i tractament de les dades comptables.

Com a resultat obtingut:

- Fitxer de desplegament. ComptaXM.war amb totes les llibreries (*.jar) pel correcte funcionament de l'aplicatiu.
- Codi font de tot l'aplicatiu. (fitxers *.java, *.xml i *.jsp)
- Script de creació i manteniment (Dades) de la base de dades.
- Memòria i presentació del Projecte.

1.4. Descripció del contingut tècnic d'aquesta memòria

A partir d'aquí es detallen les diferents etapes seguides durant el desenvolupament del projecte.

- **L'especificació i l'anàlisi del cas.** On es fa esment del diagrama de paquets, els casos de cada subsistema i la seva especificació. Així com la lògica de funcionament de l'aplicatiu, qüestions sobre la seguretat i les expectatives de funcionalitats futures
- **El disseny.** Conté l'arquitectura de l'aplicació, el disseny del diagrama de classes i la seva jerarquia, el disseny de la persistència, diagrama d'estats i el disseny de la interfície gràfica.
- **La implementació.** Es fa esment de la tecnologia i programari emprat per realitzar el projecte i les desviacions de la implementació vers el disseny original.
- **Conclusions.** Finalment la conclusió, on l'autor expressa les seves sensacions en el transcurs de la realització del projecte.

2. Anàlisi i especificació dels requeriments del cas

Aquest apartat conté una descripció, l'anàlisi, especificació i disseny del TFC 'ComptaXM'.

En quant l'anàlisi i especificació s'explica com estarà estructurat el programari, la lògica de funcionament i l'especificació dels casos d'ús detallada per cada subsistema de l'aplicatiu incorporant els diagrames de seqüència més representatius de cada subsistema. També es fa esment de les futures ampliacions i/o noves funcionalitats.

2.1 Introducció.

Per a aconseguir l'èxit en el desenvolupament del projecte, el sistema està estructurat en 3 subsistemes clarament diferenciats sobre les diferents funcionalitats requerides per la implementació. Tals com:

- **Subsistema de connexió i gestió d'usuaris.** Gestiona l'accés al sistema amb les funcionalitats de validació i entrada al sistema. A part es durà a terme totes les funcionalitats de gestió d'usuaris (alta, modificació i baixa).
- **Subsistema de manteniment.** Gestiona la configuració dels diferents components de l'aplicatiu. Dit d'altra manera, aquest subsistema serà l'encarregat de gestionar les dades que faran de suport i/o s'alimentarà el subsistema de rebuts. Tals com la gestió de comptes, períodes comptables, centres i proveïdors. En tots els casos es pot afegir, modificar, donar de baixa i visualitzar les dades dels diferents objectes.
- **Subsistema de rebuts i informes.** Gestiona les dades comptables, pròpiament dites. És a dir, la funcionalitat d'entrada (rebuda de les dades) i la seva posterior presentació sota diferents visions. La primera funcionalitat serà la introducció de rebuts (tant previsions, com factures), on s'haurà d'alimentar dels diferents components (proveïdors, centres, comptes i períodes comptables) per a dur-ho a terme. La segona funcionalitat serà la presentació d'aquests rebuts de forma global, segons comptes, centres i/o períodes comptables.

2.2 Diagrama de paquets

El sistema està compost de dos actors que interactuaran amb els tres subsistemes. Aquesta divisió s'ha fet tenint en compte les funcionalitats i les relacions dels actors amb els subsistemes.

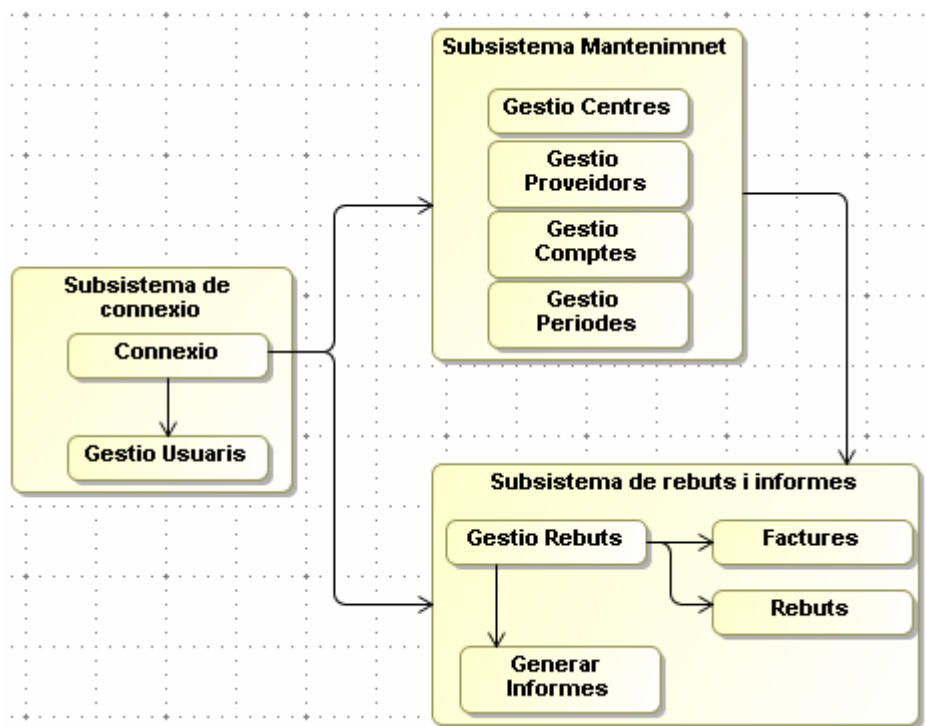


Figura 2. Diagrama de Paquets

2.3 Actors

Hi ha dos tipus d'actors relacionats amb els diferents subsistemes:

- **Administrador.** Actor amb tots els privilegis. Podrà realitzar les tasques del capPlanta, gestionar i configurar el sistema. Pot crear nous usuaris, comptes d'exploració, centres, períodes Comptables, proveïdors, així com modificar-los i donar-los de baixa.
- **CapPlanta.** Actor que representa el paper d'usuari Client. Principalment interactuarà amb el subsistema de rebuts. Cal fer ressenya que el sistema presentarà una interfície privada per cada actor CapPlanta, és a dir, cadascun d'ells estarà vinculat a un o varis centres però un centre només podrà tenir un capPlanta associat. D'aquesta manera un capPlanta només accedirà a la gestió dels centres associats a ell.

Els papers principals assignats son: validar-se al sistema, entrar rebuts, visualitzar informes sobre dades comptables associades als centres que té assignats i gestionar proveïdors dels seus centres associats.

2.4 Casos d'us i diagrames de seqüència. Per cada Subsistema

Subsistema de connexió i gestió d'usuaris

Aquest subsistema és l'encarregat de totes les funcionalitats respecte a l'usuari.

Funcionalitats:

- **Validar.** Comprovar que un usuari estigui donat d'alta i actiu.
- **Donar accés i visibilitat.** Una vegada l'usuari hagi estat validat s'encarregarà de la presentació visual i dades personalitzada segons el tipus d'usuari i centres vinculats.
- **Gestionar els usuaris.** Alta, modificació i baixa dels usuaris, així com poder llistar i seleccionar-los.

Casos d'us del subsistema de connexió i gestió d'usuaris

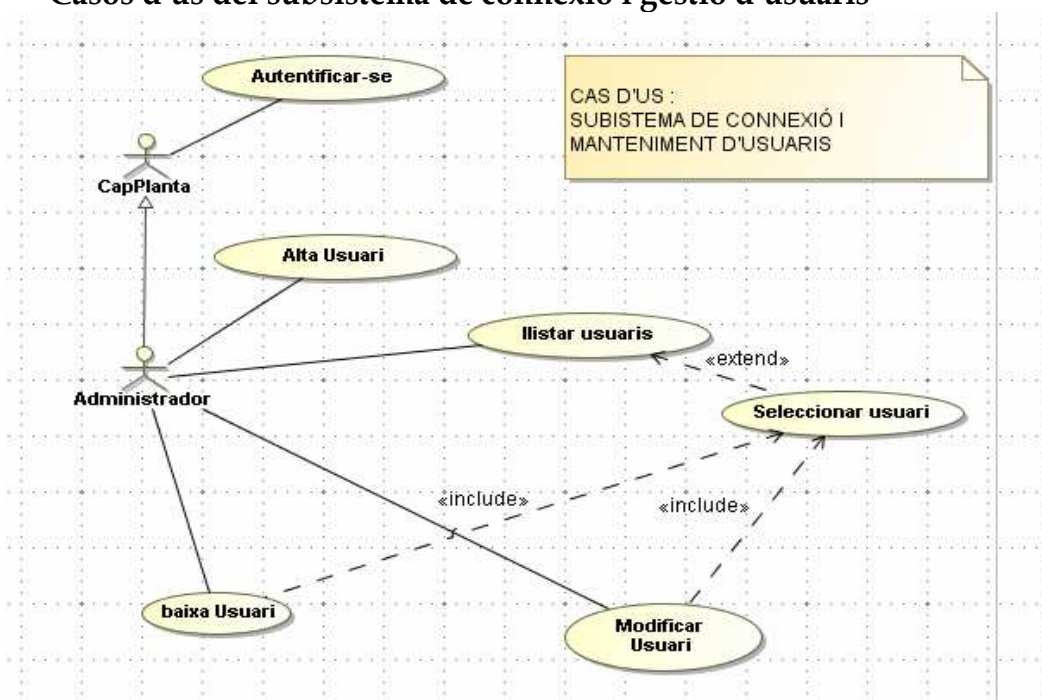


Figura 3. Cas d'us subsistema de Connexió

CAS D'US 1: Validar Usuari	
Descripció:	Valida un usuari el qual ha proporcionat el seu nom d'usuari i clau de pas.
Actors:	Administrador i capPlanta
Paper usuari:	El fan servir tots els usuaris abans d'accedir al sistema
Casos d'us relacionats:	Donar Accés al sistema
PreCondicció:	L'usuari té accés a un navegador Web
PostCondicció:	En cas que la validació sigui certa el sistema donarà accés a l'usuari, en cas contrari, avisarà a l'usuari que no té accés.

Subsistema que hi intervenen:	Subsistema de connexió i gestió d'usuaris.
Comentaris:	Per accedir el sistema TOTS els usuaris s'hauran de validar. Per qüestions de seguretat, el nom d'usuari i la seva clau de pas els proporcionarà una persona responsable que li haurà sigut encomanada aquesta tasca, la qual queda fora de l'abast d'aquest aplicatiu.
CAS D'US 2: Donar Accés al sistema	
Descripció:	El sistema dona un accés personalitzat en funció de l'usuari validat
Actors:	Administrador i capPlanta
Paper usuari:	-
Casos d'us relacionats:	Validar l'usuari
PreCondicció:	L'usuari ha estat validat correctament
PostCondicció:	Presentació de la interfície adequada segons el tipus d'usuari
Subsistema que hi intervenen:	Subsistema de connexió i gestió d'usuari
Comentaris:	Es presentarà la interfície gràfica i les dades corresponents, segons els tipus d'usuari i centres els quals estigui associats.
CAS D'US 3: Llistar usuaris	
Descripció:	El sistema presenta una llista de tots els usuaris donats d'alta al sistema, tant si estan actius o no
Actors:	Administrador.
Paper usuari:	L'usuari escull l'opció gestió usuaris
Casos d'us relacionats:	Donar accés al sistema
PreCondicció:	L'actor és un administrador
PostCondicció:	Mostrar un llistat d'usuaris
Subsistema que hi intervenen:	Subsistema de connexió i gestió d'usuari
Comentaris:	El sistema presentarà una llista de tots els usuaris existents al sistema.
CAS D'US 4: Seleccionar un usuari	
Descripció:	Un usuari administrador selecciona un usuari de la llista d'usuaris per poder visualitzar-ne el detall.

Actors:	Administrador.
Paper usuari:	Un administrador selecciona un usuari.
Casos d'us relacionats:	Llistar usuaris.
PreCondicció:	Hi ha d'haver almenys un usuari donat d'alta al sistema.
PostCondicció	Mostrar el detall de l'usuari seleccionat
Subsistema que hi intervenen:	Subsistema de connexió i gestió d'usuari
Comentaris:	El sistema presentarà un formulari amb tots els camps que puguin ser modificats respecte a l'usuari.

CAS D'US 5: Alta usuari

Descripció:	Permet crear i donar d'alta un usuari.
Actors:	Administrador.
Paper usuari:	El fa servir un usuari administrador per donar accés al sistema a nous usuaris.
Casos d'us relacionats:	Donar accés al sistema
PreCondicció:	L'usuari connectat és un administrador i l'usuari a crear no existeix.
PostCondicció:	L'usuari es creat
Subsistema que hi intervenen:	Subsistema de connexió i gestió d'usuari
Comentaris:	Un administrador podrà donar d'alta a altres administradors o a usuaris capPlanta. Serà l'encarregat de proporcionar les dades al sistema. Prèviament aquestes dades les haurà aconseguit fora de l'abast d'aquest aplicatiu.

CAS D'US 6: Modificar usuari

Descripció:	Permet modificar les dades editables d'un usuari existent en el sistema i prèviament seleccionat
Actors:	Administrador.
Paper usuari:	El fa servir un usuari administrador per modificar les dades d'un usuari.
Casos d'us relacionats:	Seleccionar usuari.
PreCondicció:	L'actor és un administrador i ha seleccionat un usuari.
PostCondicció:	Les dades de l'usuari capPlanta han estat modificades.

Subsistema que hi intervenen:	Subsistema de connexió i gestió d'usuari.
Comentaris:	Un administrador podrà modificar les dades de qualsevol usuari.
CAS D'US 7: Baixa usuari	
Descripció:	Permet donar de baixa un usuari.
Actors:	Administrador.
Paper usuari:	El fa servir un usuari administrador per donar de baixa un usuari.
Casos d'us relacionats:	Seleccionar usuari.
PreCondicció:	L'actor és un administrador i l'usuari existeix en el sistema.
PostCondicció:	L'usuari no pot tornar accedir al sistema.
Subsistema que hi intervenen:	Subsistema de connexió i gestió d'usuari
Comentaris:	L'usuari donat de baixa no podrà accedir al sistema tot i que no serà eliminat físicament. Per qüestions de seguretat i coherència un mateix usuari no es podrà donar de baixa.

Diagrama de seqüència (login usuari)

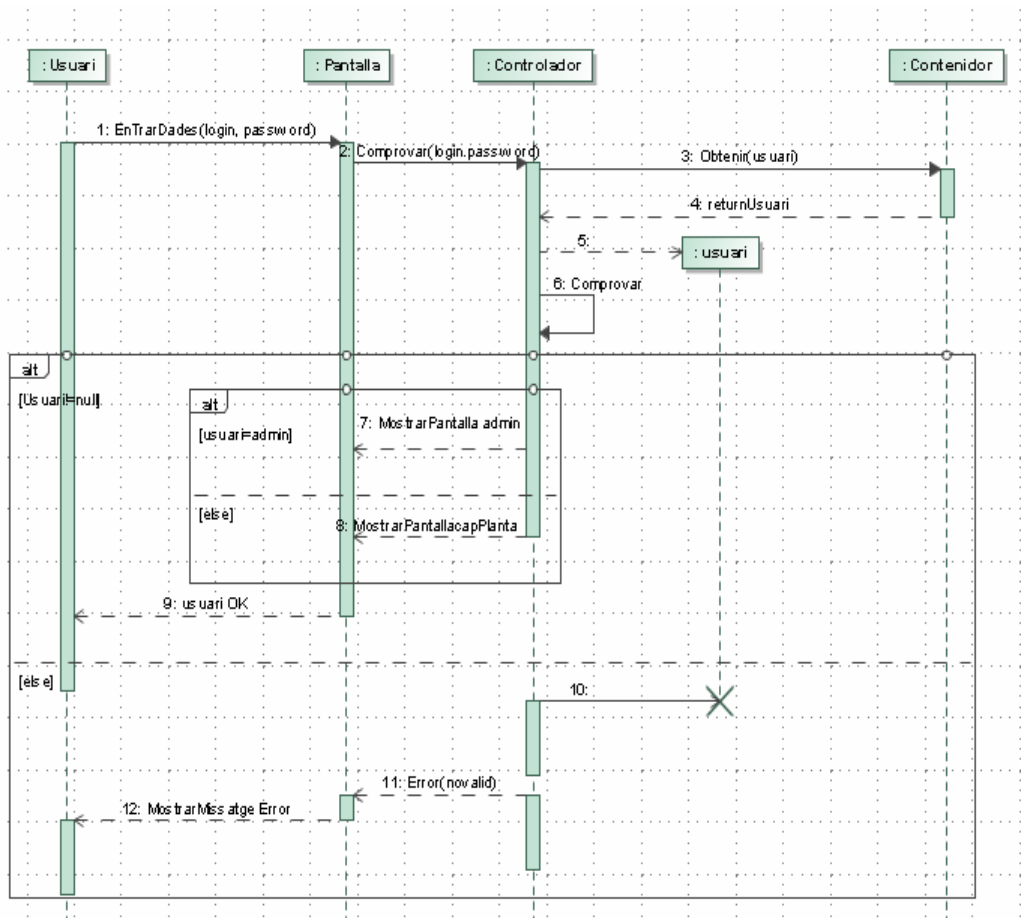


Figura 4. Diagrama de Seqüència. Login Usuari

Diagrama de seqüència (AfegirUsuari)

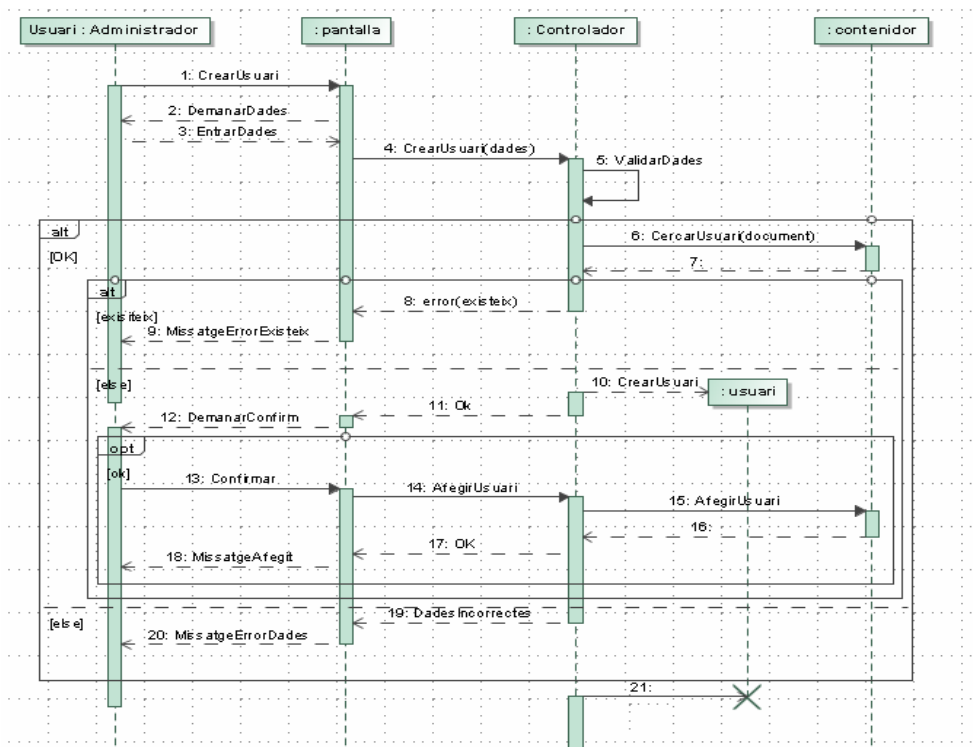


Figura 5. Diagrama de seqüència. Afegir Usuari

Subsistema de manteniment

Es desenvolupen totes les funcionalitats de manteniment general de l'aplicació i només hi participarà l'actor Administrador amb l'excepció de la gestió de proveïdors on l'actor capPlanta podrà gestionar els proveïdors dels seus centres associats.

Funcionalitats:

- **Gestió Comptes.** Un administrador podrà llistar, filtrar per codi i/o descripció, seleccionar, donar d'alta, modificar el codi i la descripció i donar de baixa un compte.
- **Gestió Períodes comptables.** Un administrador podrà llistar, seleccionar, donar d'alta, modificar, donar de baixa i filtrar els períodes comptables per any i/o per mes i per centre.
- **Gestió Centres.** Un administrador podrà llistar, seleccionar, donar d'alta, modificar i donar de baixa un centre.
- **Gestió Proveïdors.** Un CapPlanta podrà llistar, seleccionar, donar d'alta, modificar i donar de baixa un proveïdor del seu centre associat. Cal entendre que els proveïdors gestionats pels CapPlanta només existeixen si existeix un centre. En el cas d'usuari Administrador podrà realitzar les mateixes accions, en aquest cas de tots els centres existents en el sistema.

Casos d'us subsistema de manteniment

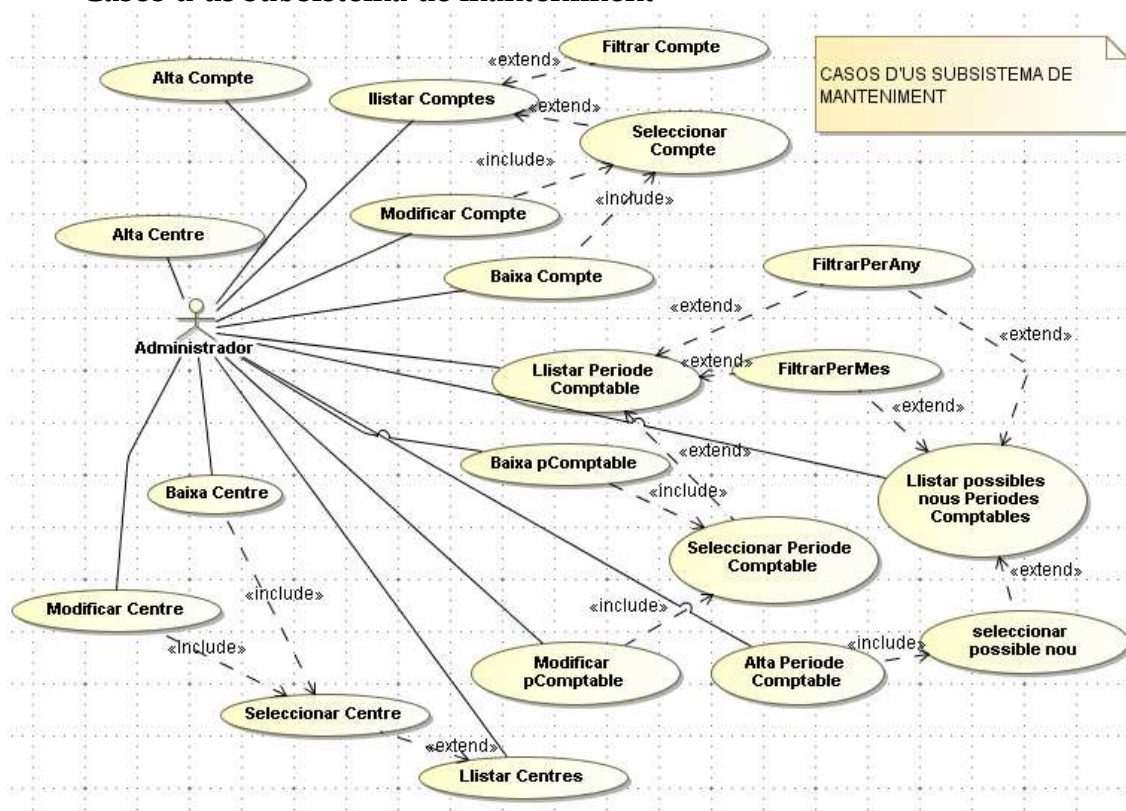


Figura 6. Cas d'us Subsistema de manteniment

CAS D'US 1: Llistar Comptes.	
Descripció:	El sistema presenta una llista de tots els comptes d'explotació existents en el sistema.
Actors:	Administrador i capPlanta
Paper usuari:	El fa servir un usuari per veure tots els comptes.
Casos d'us relacionats:	-
PreCondicció:	Almenys existeix un compte en el sistema.
PostCondicció:	Es mostren tots els comptes existents.
Subsistemes que hi intervenen:	Subsistema de manteniment i subsistema de rebuts.
Comentaris:	Aquest cas d'us intervé en diferents subsistemes. Tant el pot fer servir un usuari capPlanta per poder veure els comptes existents per poder seleccionar un compte i vincular-lo a un rebut com un usuari administrador per poder seleccionar i gestionar-lo.
CAS D'US 2: Seleccionar un compte	

Descripció:	Permet a un usuari seleccionar un compte
Actors:	Administrador i capPlanta
Paper usuari:	El fa servir un usuari per escollir un compte
Casos d'ús relacionats:	Llistar comptes
PreCondicció:	Almenys existeix un compte en el sistema.
PostCondicció:	El compte resta seleccionat
Subsistemes que hi intervenen:	Subsistema de manteniment i subsistema de rebuts.
Comentaris:	Aquest cas d'ús intervé en diferents subsistemes. Tant el pot fer servir un usuari capPlanta per poder seleccionar un compte i vincular-lo a un rebut com un usuari administrador per gestionar-lo.

CAS D'US 3: Filtrar Comptes

Descripció:	Permet a un usuari administrador obtenir una llista personalitzada de comptes. Els podrà filtrar pel seu codi i/o descripció.
Actors:	Administrador
Paper usuari:	El fa servir un usuari per llistar els comptes.
Casos d'ús relacionats:	Seleccionar un compte, llistar comptes
PreCondicció:	Almenys existeix un compte en el sistema.
PostCondicció:	Es mostren tots els comptes existents d'acord amb el filtre proposat
Subsistemes que hi intervenen:	Subsistema de manteniment.
Comentaris:	El sistema cerca els comptes a partir del filtre aplicat. Cercarà tots els comptes que comencin per el codi i/o descripció proposada en el filtre.

CAS D'US 4: Alta compte.

Descripció:	Permet a un usuari administrador donar d'alta un compte en el sistema.
Actors:	Administrador.
Paper usuari:	El fa servir un usuari Administrador per donar d'alta un compte en el sistema.
Casos d'ús relacionats:	-
PreCondicció:	L'actor és administrador i el compte no existeix en el sistema.
PostCondicció:	El nou compte existeix en el sistema

Subsistemes que hi intervenen:	Subsistema de manteniment.
Comentaris:	Per donar d'alta el compte caldrà proporcionar un codi comptable i una descripció.
CAS D'US 5: Modificar un compte	
Descripció:	Permet a un usuari Administrador modificar el codi i la descripció d'un compte.
Actors:	Administrador.
Paper usuari:	El fa servir un usuari administrador per gestionar un compte
Casos d'us relacionats:	Seleccionar un compte.
PreCondicció:	L'actor es administrador i el compte existeix en el sistema.
PostCondicció:	El compte és modificat.
Subsistemes que hi intervenen:	Subsistema de manteniment.
Comentaris:	Es permet modificar el codi i la descripció del compte.
CAS D'US 6: Baixa compte	
Descripció:	Permet a un usuari Administrador donar de baixa un compte.
Actors:	Administrador.
Paper usuari:	El fa servir un usuari per donar de baixa un compte del sistema.
Casos d'us relacionats:	Seleccionar un compte.
PreCondicció:	L'actor es administrador.
PostCondicció:	El compte deixa d'estar actiu
Subsistemes que hi intervenen:	Subsistema de manteniment.
Comentaris:	S'efectua una baixa lògica del sistema.

Nota: no descriu els altres casos d'us del subsistema de manteniment ja que tots segueixen el mateix patró, tant en la gestió de centres i proveïdors. A continuació faig esment dels casos d'ús implicats en la creació de nous períodes comptables, on el seu funcionament difereix un xic.

CAS D'US 7: Cercar nous períodes Comptables	
Descripció:	Permet a l'usuari administrador cercar els possibles nous períodes comptables

Actors:	Administrador.
Paper usuari:	El fa servir un usuari per esbrinar quins períodes poden ser creats.
Casos d'us relacionats:	Llistar períodes Comptables
PreCondicció:	L'actor es administrador.
PostCondicció:	Una llista de possibles nous comptes
Subsistemes que hi intervenen:	Subsistema de manteniment.
Comentaris:	És una acció prèvia abans de poder crear, nous períodes comptables. Primer cal esbrinar quins períodes comptables, poden ser creats a partir d'una data (mes i any). Cal recordar que un període comptable està compost per un any, un mes i un centre. D'aquesta manera l'actor indica un mes i un any i el sistema cerca tots els possibles períodes comptables que encara no han estat creats pel mes i any indicat.

CAS D'US 8: Crear un període Comptable

Descripció:	Permet a un usuari Administrador crear un nou període comptable
Actors:	Administrador.
Paper usuari:	El fa servir un usuari per crear un període Comptable.
Casos d'us relacionats:	Cercar nous períodes comptables i seleccionar un possible nou període comptable
PreCondicció:	L'actor es administrador i el període comptable no existeix en el sistema
PostCondicció:	El període comptable existeix en el sistema
Subsistemes que hi intervenen:	Subsistema de manteniment.
Comentaris:	Per poder crear nous períodes comptables, primer cal esbrinar quins poden ser creats. Una vegada el sistema ha esbrinat i presentat quins son els candidats a ser creats (a partir d'un mes i un any), l'actor ha de seleccionar-los per poder-los crear.

Diagrama de seqüència (Alta Compte)

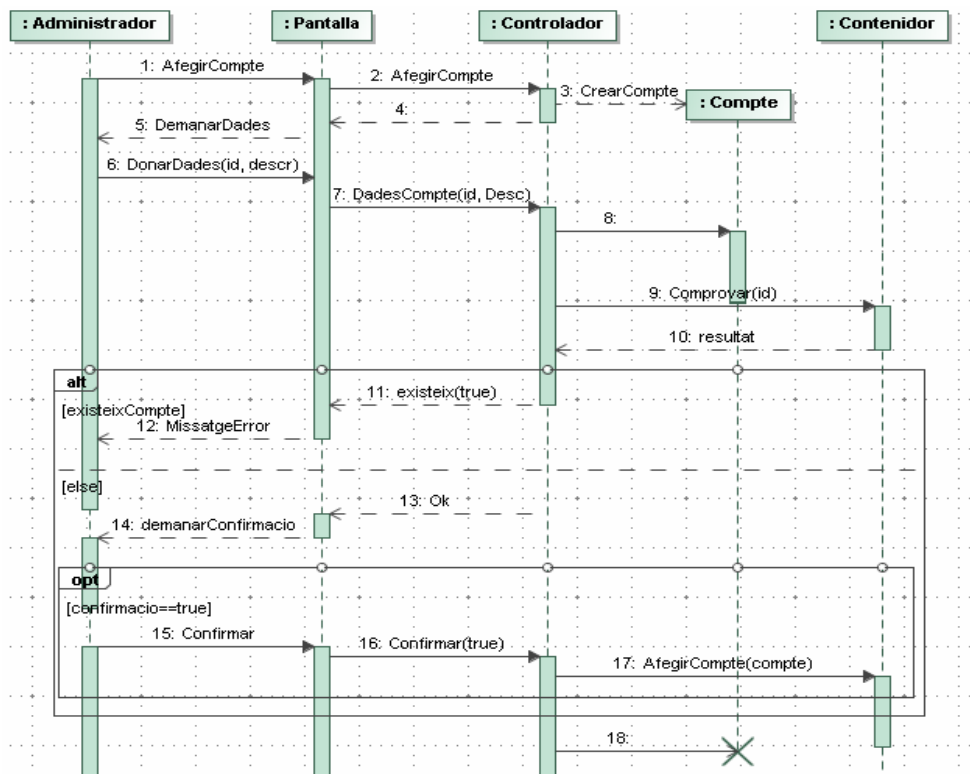


Figura 7. Diagrama de Seqüències. Alta Compte

Diagrama de seqüència (baixaCompte)

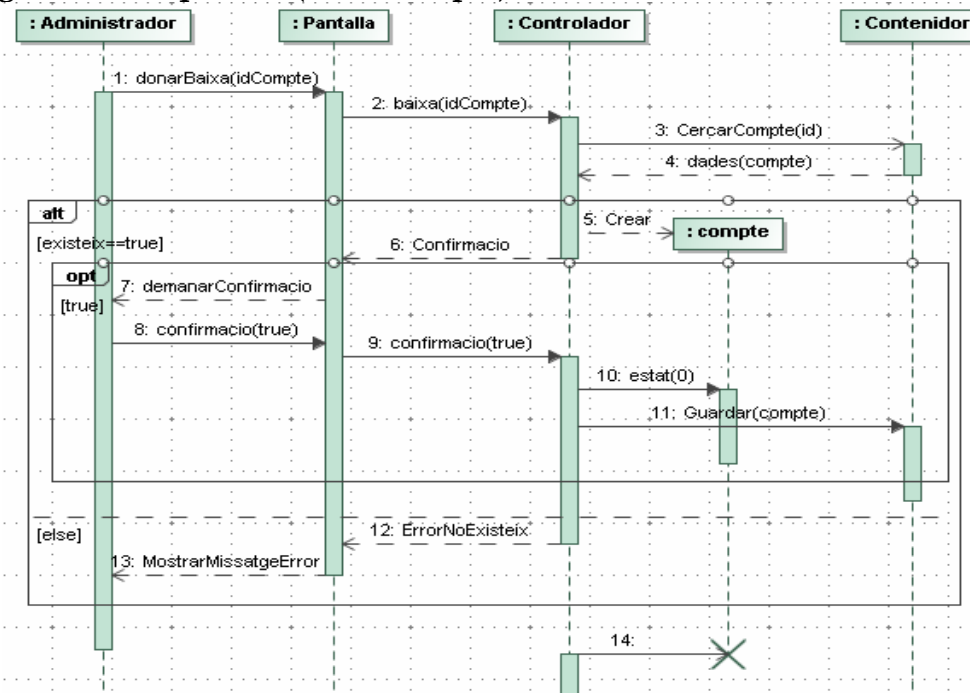


Figura 8. Diagrama de seqüència. Baixa Compte.

Subsistema de rebuts i informes.

En aquest subsistema es desenvoluparan totes les funcionalitats de la gestió del rebuts i la presentació d'informes:

Funcionalitats:

- **Gestió de rebuts.** Qualsevol usuari podrà donar d'alta, modificar i donar de baixa una previsió i/o una factura ([veure lògica de funcionament](#)). Si l'actor és un Administrador podrà donar d'alta qualsevol tipus de rebut per a qualsevol centre, en cas contrari, només al centre associat a l'usuari capPlanta. Els rebuts¹ només es podran gestionar si el període comptable, el centre i el compte estan actius.
- **Gestió d'informes:** Es crearan dos tipus genèrics d'informes:
 - **Informe global per període Comptable.** Informe on es presentaran els períodes Comptables² agrupats per mes i any. L'actor cap de Planta podrà visualitzar els períodes comptables dels centres que té associats
 - **Informe per Centre i any.** El podran visualitzar tots els usuaris, de manera restringida els capPlanta, on només podrà visualitzar els dels seus centres associats. L'informe presentarà les dades comptables d'un centre d'un any escollit.

Casos d'ús subsistema de rebuts i informes

¹ Un rebut està compost d'un període comptable, un centre, un compte i un valor. Si el rebut és una factura aquest tindrà dia i proveïdor.

² Cal recordar que un període comptable està representat per un mes, un any i un centre en concret

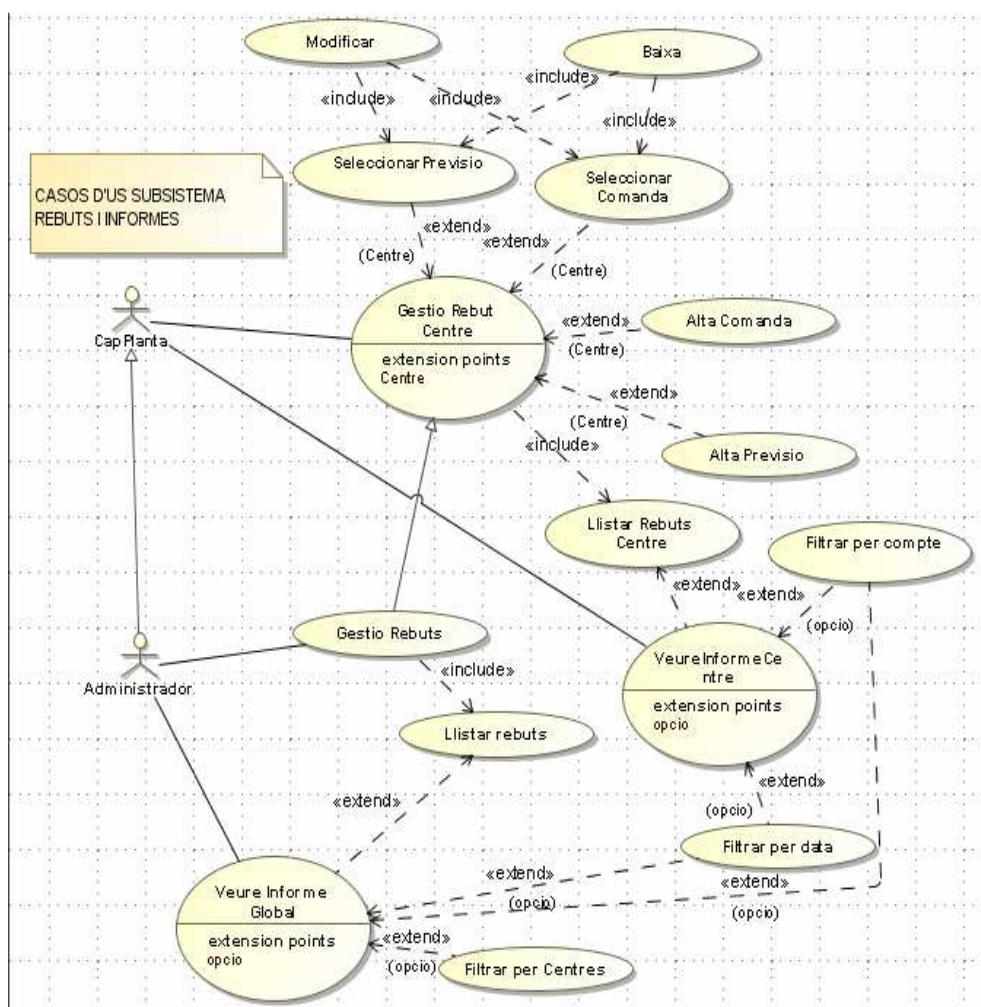


Figura 9. Cas d'ús Subsistema de rebuts i informes.

CAS D'US 1: Alta Previsió	
Descripció:	Permet a un usuari introduir al sistema un rebut del tipus previsió.
Actors:	Administrador i capPlanta
Paper usuari:	El fa servir l'usuari per introduir previsions al sistema
Casos d'us relacionats:	Selecció d'un centre, selecció d'un compte, selecció d'un període comptable.
PreCondicció:	Hi ha almenys un centre, un període comptable i un compte donats d'alta i actius en el sistema.
PostCondicció:	La previsió existeix en el sistema
Subsistemes que hi intervenen:	Subsistema de rebuts.
Comentaris:	Si l'usuari és administrador podrà escollir qualsevol centre que estigui actiu, en el cas de

	l'actor capPlanta només els centres associats actius.
CAS D'US 2: Modificar Previsió	
Descripció:	Permet a l'usuari modificar una previsió
Actors:	Administrador i capPlanta.
Paper usuari:	El fa servir l'usuari per modificar previsions.
Casos d'us relacionats:	Selecció de Previsió.
PreCondicció:	La previsió existeix i associada a un període comptable actiu.
PostCondicció:	La previsió es modificada
Subsistemes que hi intervenen:	Subsistema de rebuts.
Comentaris:	D'una previsió es podrà modificar la descripció i l'import.
CAS D'US 3: Baixa Previsió	
Descripció:	Permet a l'usuari eliminar una previsió (baixa física).
Actors:	Administrador i capPlanta.
Paper usuari:	El fa servir un usuari per eliminar una previsió
Casos d'us relacionats:	Selecció de previsió.
PreCondicció:	La previsió ha d'existir i ha d'estar associada a un període comptable en actiu.
PostCondicció:	La previsió deixa d'existir.
Subsistemes que hi intervenen:	Subsistema de rebuts.
Comentaris:	L'actor Administrador podrà eliminar previsions de tots els centres i l'actor capPlanta dels centres associats.
CAS D'US 4: Selecció de previsió	
Descripció:	Permet a un usuari visualitzar el detall d'una previsió
Actors:	Administrador i capPlanta
Paper usuari:	El fa servir un usuari per veure el detall d'una previsió
Casos d'us relacionats:	Modificar previsió.
PreCondicció:	La previsió ha d'existir
PostCondicció:	Veure el detall de la previsió.

Subsistemes que hi intervenen:	Subsistema de rebuts.
Comentaris:	Cal remarcar que l'acció de seleccionar ja portarà implícita l'acció de visualitzar.

Nota: no comento els casos d'us relacionats amb la gestió de Factures perquè tenen un comportament similar a la gestió de previsions amb la diferència que les factures tenen dia i proveïdor.

CAS D'US 5: Seleccionar Informe	
Descripció:	Permet a l'usuari escollir un tipus d'informe.
Actors:	Administrador, cap de planta
Paper usuari:	El fa servir l'usuari per seleccionar el tipus d'informe a visualitzar
Casos d'us relacionats:	Filtrar Informe.
PreCondicció:	-
PostCondicció:	Es presenta la interfície per crear l'informe
Subsistemes que hi intervenen:	Subsistema de rebuts i informes
Comentaris:	L'acció de seleccionar presenta la interfície gràfica per poder crear l'informe.

CAS D'US 6: Filtrar informe global	
Descripció:	Permet a l'usuari personalitzar un informe.
Actors:	Administrador i cap de planta
Paper usuari:	El fa servir l'usuari per fabricar un informe
Casos d'us relacionats:	Seleccionar informe
PreCondicció:	Hi ha almenys un rebut donat d'alta en el sistema
PostCondicció:	-
Subsistemes que hi intervenen:	Subsistema de rebuts.
Comentaris:	L'informe es podrà filtrar per diferents conceptes. (veure lògica de funcionament).

CAS D'US 7: Visualitzar informe	
Descripció:	Permet a un usuari visualitzar un informe
Actors:	Administrador i capPlanta.
Paper usuari:	-
Casos d'us relacionats:	Seleccionar un informe, filtrar un informe.
PreCondicció:	Hi ha almenys un rebut donat d'alta en el sistema

PostCondió:	Es visualitzen les dades de l'informe
Subsistemes que hi intervenen:	Subsistema de rebuts.
Comentaris:	-

Diagrama de seqüència (entrar una nova Factura)

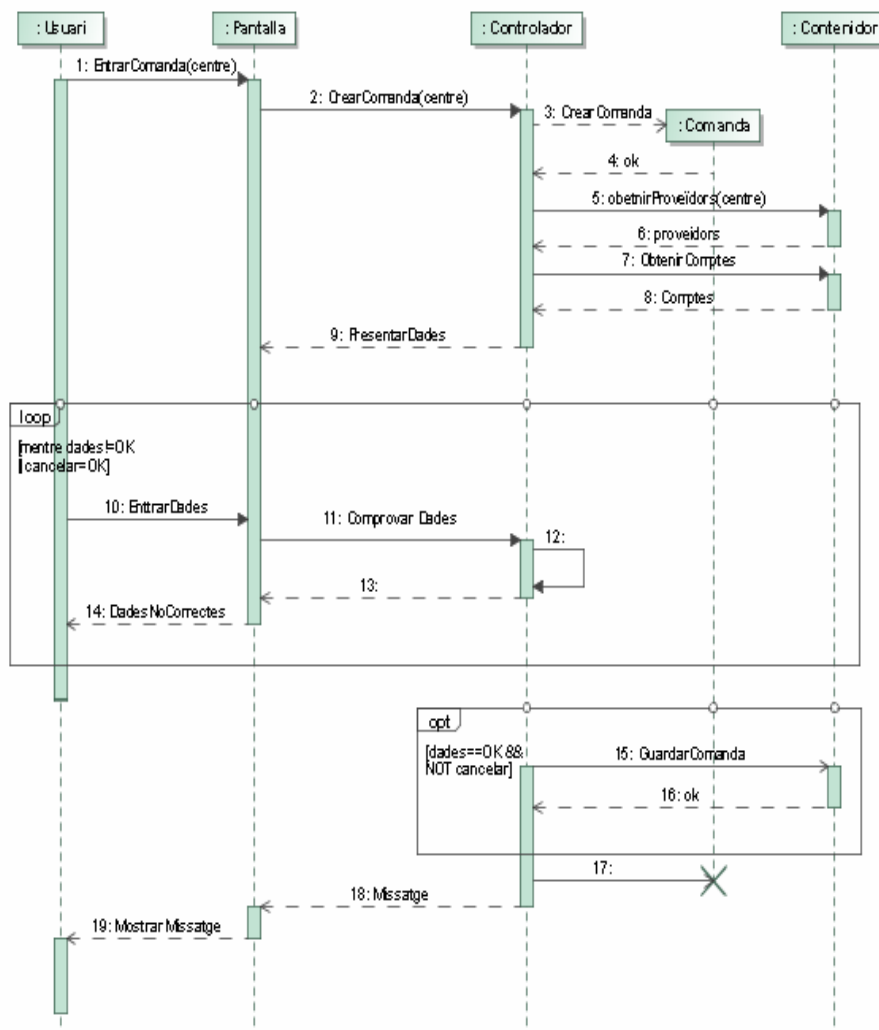


Figura 10. Diagrama de Seqüència. Entrar nova Factura

Diagrama de seqüència (Veure Informe Global)

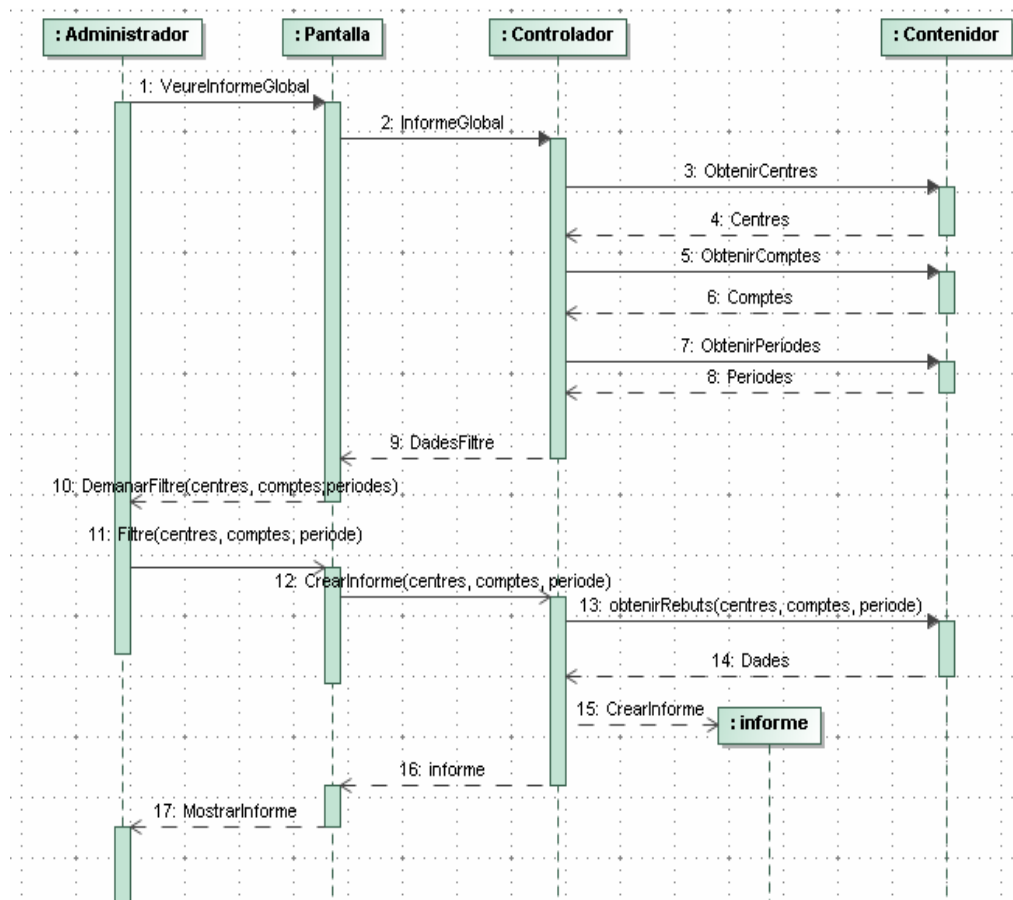


Figura 11. Diagrama de Seqüència. Veure Informe Global.

2.5 Lògica de funcionament

Tal com s’ha pogut entreveure en l’especificació dels casos d’us i la descripció del sistema. Es tracta d’un aplicatiu que dona servei a un col·lectiu d’usuaris, els quals tenen dos nivells d’interacció amb el sistema.

Un actor administrador que gaudeix de tots els privilegis possibles i un actor capPlanta que té restringides les seves accions i a més a més dintre el mateix nivell es tractarà de manera personalitzada. És a dir a un usuari capPlanta només tindrà al seu abast les interaccions amb el seu/s centre/s associats a ell.

Faig esment de les accions de baixa (usuaris, centres, proveïdors, períodes comptables i comptes) del sistema perquè necessiten un funcionament peculiar pel bon funcionament de l’aplicatiu. Totes les baixes seran lògiques és a dir tots els objectes tindran un estat (booleà) quan aquest sigui fals l’objecte causarà baixa del sistema però seguirà existint.

Poso un exemple per fer-ho més entenedor:

Tenim un centre anomenat 'centre1', durant un cert temps amb aquest centre si realitzen un seguit d'operacions típiques, té un usuari i rebuts associats a ell. En el transcurs del temps el centre1 pot deixar de formar part de l'empresa, per diferents motius (fi del període d'explotació, tancament, etc), llavors caldrà donar-lo de baixa del sistema perquè no té sentit que continuï estant actiu, pel simple fet que no si registrarà cap moviment, però si la baixa fos física llavors perdriem part de la informació global del conjunt. Així el centre1 el necessitem per tal de poder realitzar informes globals de l'empresa ja que forma part de l'història de l'empresa.

També puntualitzo la baixa, o millor dit estat **TANCAT** d'un període comptable perquè serà una acció que s'haurà de fer sovint. Més ben dit, tots els períodes comptables en un moment o altre passaran a l'estat de baixa o tancat. Llavors cap usuari podrà associar un rebut al període en concret però podrà visualitzar els moviments realitzats en aquest.

Tots els objectes es podran donar d'alta, modificar i efectuar una baixa lògica, amb tot cal fer una excepció. L'únic que es podrà efectuar una baixa física seran els rebuts mentre estiguin associats a un període comptable que estigui actiu. En aquest cas considero que no té sentit que romangui en el sistema, per tal de facilitar la tasca de la creació d'informes.

D'altra banda l'estat dels objectes podrà ser reversible, per exemple un centre pot estar de baixa un temps i llavors és pot tornar a incorporar al sistema, com queda explicat en l'apartat d'especificacions aquesta acció només la pot realitzar un actor Administrador.

En l'apartat de funcionalitats futures proposo un sistema de neteja per tal d'eliminar els objectes que no prestin cap funcionalitat al sistema. Per exemple un centre que no tingui cap usuari ni rebut associat.

Faig esment a l'operació d'alta dels rebuts. Aquests només es podran donar d'alta si la seva data d'expedició està dintre del mes i l'any, d'un període comptable que estigui donat d'alta i estigui actiu.

Sobre els períodes comptables. Ha d'estar associat a un centre determinat. D'aquesta manera es podrà validar els períodes comptables per centre de manera individualitzada. Dit d'altra manera, un període comptable no pot existir si no té un centre associat. [Veure diagrama de classes](#). Un administrador ha de poder mantenir en actiu els rebuts d'un centre determinat en un període de temps (mensual) i ha de poder mantenir tancat o de baixa en aquest mateix període un altre centre.

Sobre la visibilitat dels proveïdors. Podran estar associats a un centre o podran ser globals.

Els proveïdors globals seran creats per administradors i seran visibles per tots els centres.

En canvi un proveïdor associat a un centre només serà visible per aquell centre i aquest el podrà editar³. [Veure diagrama de classes.](#)

I finalment el tractament de la visualització d'informes. En una primera versió, la que m'ocupa, es presentarà dos tipus d'informes genèrics. On l'usuari podrà filtrar les dades sobre diferents conceptes:

1. Informe per centres:
 - a. Filtrar un centre, l'actor Administrador de tots els centres existents en el sistema, l'actor capPlanta dels que té associats.
 - b. Filtrar un rang de períodes comptables.
 - c. Filtrar un rang de comptes.
2. Informe global per Període Comptable:
 - a. Filtrar un període comptable
 - b. Filtrar un rang de centres.
 - c. Filtrar un rang de comptes.

2.6 Funcionalitats futures

S'estudia implementar un sistema de recollida de brossa. És a dir, si existeix un objecte que no presta cap servei al sistema aquest podrà ser eliminat. Es pot donar el cas que un administrador doni d'alta comptes, centres i/o usuaris per equivocació. En l'actualitat l'aplicació tant sols contempla la possibilitat de donar de baixa i no permet l'eliminació física, per tal de mantenir la integritat del sistema.

S'estudia implementar nous informes, a petició del client. De la mateixa manera, insertar gràfics. De moment queda a les expectatives del client sobre com millorar la rendibilitat de l'aplicatiu.

S'estudia implementar nous formats per presentar i exportar les dades, per exemple en fulls de càlcul i/o fitxers PDF. Així com, aplicar un sistema de consultes dinàmiques, on cada usuari pugui personalitzar les seves pròpies consultes.

2.7 Seguretat

L'aplicació ComptaXM està concebuda per oferir un accés privat i personalitzat als usuaris. De manera que cada usuari capPlanta només tindrà accés a les seves dades.

Per acomplir amb tal propòsit es dotarà al sistema d'un mecanisme de control sobre la sessió amb la implementació de la interfície *SessionAware*.

³ Cal entendre qui edita o gestiona els proveïdors son els usuaris. En aquest cas l'usuari associat al centre.

Totes les accions implementen aquesta interfície, on es guardaran dades de la sessió, una de les quals, l'usuari actual. Llavors mitjançant un interceptor personalitzat, 'IdentificacióInterceptor', es controlarà que sempre hi hagi un usuari acreditat. [Veure Annex2](#)

Quan un usuari accedeixi a l'aplicació aquest quedarà registrat com una nova sessió i en tot moment podrà tancar la sessió accedint a la pestanya 'SORTIR', d'aquesta manera les dades visualitzades en aquella sessió concreta no seran accessibles ni visibles per cap usuari que no s'acrediti adequadament.

Només tindran accés a les dades de tots els centres els usuaris administradors.

Els usuaris capPlanta només podran accedir als centres que tenen assignats i en cap cas podran accedir ni visualitzar les dades de qualsevol altra centre.

Si un usuari excedeix els temps màxim de sessió, la propera acció que realitzi, sigui quina sigui, el sistema retornarà a la pàgina d'inici per tal que s'hagi de tornar a identificar per accedir al sistema.

3. Disseny

En l'etapa del disseny he plasmat en un aspecte més formal⁴, totes les indicacions de l'anàlisi i l'especificació, tals com:

- El diagrama de classes amb totes les seves relacions.
- L'estructura de les entitats relacionals.
- El prototip de la interfície gràfica.
- Concretar les eines a utilitzar per implementar la solució escollida.

També, dintre aquesta etapa, una de les tasques més importants ha estat la realització de la corba d'aprenentatge sobre les eines de desenvolupament escollides.

Finalment una vegada assolides totes les tasques i ja enfocant la pròxima etapa, la implementació, he preparat l'entorn de programació per poder-la començar i afrontar amb èxit.

3.1 Arquitectura del sistema i programari emprat.

L'aplicació **ComptaXM** està desenvolupada en una arquitectura basada en **J2EE (Java Platform, Enterprise Edition)**.

És una plataforma de programació per a desenvolupar i executar programari d'aplicacions en llenguatge de programació JAVA.

Permet utilitzar arquitectures de N capes distribuïdes (multicapa) i s'ajuda àmpliament en components de programari modulars executant-se sobre un servidor d'aplicacions.



Figura 12. JavaEE

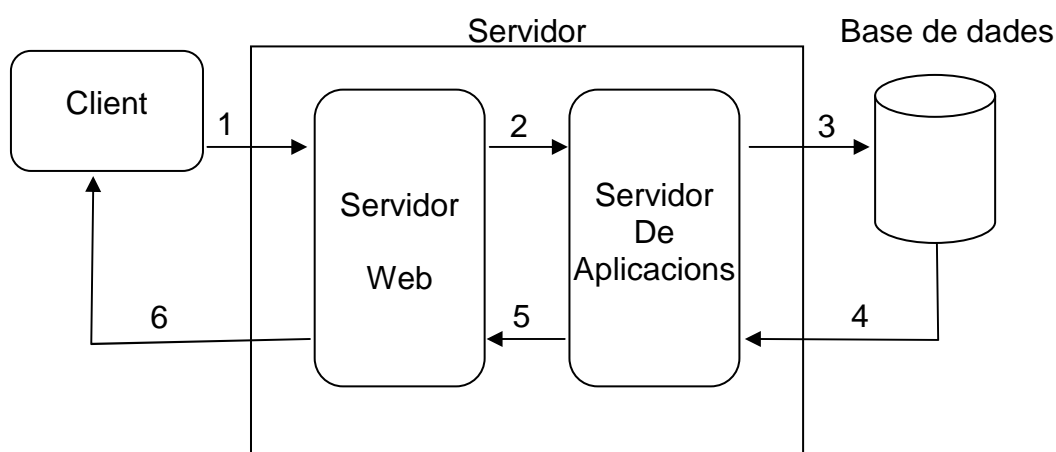


Figura 13. Arquitectura J2EE

⁴ Aciareixo la incorporació dels diagrames de seqüència en l'apartat d'anàlisi i especificació, d'aquesta memòria, per tal de tenir-ho més endreçat i poder-los agrupar per subsistema, però en realitat els diagrames de seqüència formen part de l'etapa de disseny.

- **Capa Client.** Qui interactua amb el sistema. Representada per la interfície gràfica. Encarregat d'introduir dades i rebre respostes.

És la capa superficial. En aquest cas per a resoldre la capa client s'utilitza l'**arquitectura de pàgines JSP**. És una tecnologia Java que permet generar contingut dinàmic WEB, en forma de documents HTML, XML, etc. En aquest cas podem parlar d'un client lleuger perquè permet usar una aplicació des d'un ordinador client sense que s'hagi d'instal·lar cap programari auxiliar. Amb un simple navegador WEB és suficient.

D'altra banda per tal de simplificar el disseny de la interfície gràfica faig ús de la tecnologia/plantilles **Tiles**. Permet utilitzar fragments de pàgines individuals que agrupades en temps de compilació presenten una interfície global.

Aquests fragments poden formar part de diferents interfícies amb la conseqüent estalvi en la repetició de codi.

Aquesta tecnologia és molt aconsellable en projectes on hi ha diferents elements gràfics que es van repetint en diferents interfícies, tals com un menú general, una capçalera unitària i altres. Tal com es en aquest cas.

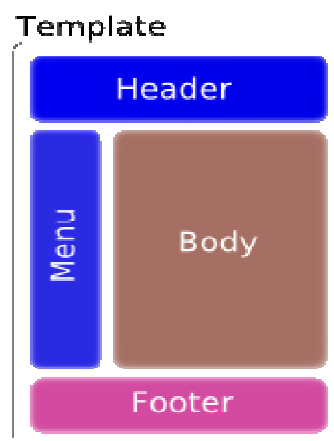


Figura 14. Estructura Tiles

A mode d'exemple:

Amb aquesta estructura podrem mantenir el mateix estil de presentació entre les diferents vistes de l'aplicació. És una manera de simplificar les tasques d'interacció de l'usuari amb el sistema. En tot moment sabrà que el menú és a l'esquerra, les eines de connexió i desconnexió sempre les podrà localitzar a la capçalera, la informació general del sistema per exemple pot està ubicat al peu de pantalla i la informació específica de cada visualització sempre restarà al cos de la pàgina.

- **Capa servidora.** Fa d'interpret. Per una part s'encarrega de rebre les peticions del client, les processa adequadament, si cal s'encarrega de recuperar o endreçar dades i finalment dona respostes al client. Dintre la capa servidora podem diferenciar dos tipus de servidors:

1. **Servidor WEB:** La seva missió principal es servir fitxers i és l'interlocutor amb el client. Per una banda el client s'adreça al servidor



Figura 15. Anagrama TomCat

WEB mitjançant el protocol HTTP, n'extreu el nom del recurs sol·licitat, el busca amb l'ajuda del servidor d'aplicacions, el torna a encapsular i l'envia el client mitjançant una petició HTTP. És un recurs estàtic.

Per a resoldre aquesta part s'utilitza **el servidor WEB TOMCAT 6.0**. És un servidor Web amb suport de Servlets i JSPs. No és un servidor d'aplicacions. Inclou el compilador Jasper per compilar JSPs convertint-les a Servlets.

TOMCAT ha estat desenvolupat en JAVA i per tant funciona sobre qualsevol sistema operatiu que disposi d'una màquina virtual, raó per la qual m'he decantat per aquesta solució ja que permet desenvolupar un programari multiplataforma.

2. **Servidor d'aplicacions.** En aquest cas es comporta de manera diferent, és la part dinàmica del servidor, a part de rebre la petició del client, la interpreta, la processa, realitza les operacions pertinents i retorna el recurs apropiat per la petició rebuda. Per dir-ho de manera plana és qui disposa del codi de l'aplicació, l'executa i el resultat l'envia al client. Una de les eines importants per a resoldre tal escomesa serà **el Framework Struts2** ja que ens permetrà l'aplicació del **patró MVC**, veure [annex1](#).



Figura 16. Anagrama Struts2

Per gestionar la persistència de les dades es realitza mitjançant la tecnologia **Hibernate**. És una eina de mapeig objecte-relacional (ORM) per la plataforma Java que facilita el mapeig dels atributs entre una base de dades relacional tradicional i el model d'objectes d'una aplicació, mitjançant fitxers declaratius XML o anotacions en els beans de les entitats que permeten establir relacions. Amb l'ús d'Hibernate es busca solucionar el problema de la diferència entre dos models de dades coexistents en una aplicació: l'usat en la memòria de la computadora (orientació a objectes) i l'usat en les bases de dades relacionals.



Figura 17. Anagrama Hibernate

Aquesta tecnologia ens permet aplicar **el patró DAO**. Els objectes d'accés a dades és un patró de disseny Core J2EE i considerats una bona pràctica. L'avantatge d'utilitzar objectes d'accés a dades es que

qualsevol objecte de negoci no té que conèixer directament el destí final de la informació que manipula.

- **Capa d'integració o sistema de persistència de les dades.** Qui s'encarrega d'emmagatzemar la informació del client, per tal que aquest en pugui disposar quan ho desitgi. En la figura 13, representada per la base de dades. El servidor d'aplicacions interactua amb el sistema de persistència per tal de satisfer les peticions del client. Per tal de portar a terme aquesta aplicació utilitzaré **el SGBD de MySql**. És un sistema gestor de base de dades relacional i de programari lliure, raó principal per la qual em decanto per aquesta solució.



Figura 18. Anagrama MySql

3.2 Diagrama de classes

A continuació presento el diagrama de classes original. Cal dir que en transcurs de l'etapa d'implementació em van sorgir dos problemes no contemplats en l'etapa del disseny

1. **El tractament donat al proveïdor.** Tal com he esmentat abans, un proveïdor pertany a un o varis centres i aquest ha de poder ser modificat per un usuari CapPlanta. Davant la problemàtica de que un proveïdor en concret podia ser modificat per diferents usuaris, els quals en fan ús, vaig creure convenient de modificar la relació entre Proveïdor i Centre i resoldre-la 1:1. Amb tot un usuari podrà crear un proveïdor per varis Centres. Per resoldre-ho l'aplicatiu, automàticament crea tants proveïdors com centres ha escollit l'usuari, [veure implementació](#). D'aquesta manera, posteriorment es podran modificar individualment, els proveïdors per cada centre.

D'altra banda i pel mateix motiu, he hagut de desvincular la classe Proveïdor de la superClasse Persona, per poder tenir varis proveïdors amb el mateix document identificador.

2. **Càlcul de resultats globals.** A l'hora de presentar les dades agrupades. És a dir l'import total de previsions o factures agrupades per comptes d'explotació, he cregut convenient crear una nova classe associativa entre Període Comptable i Compte d'explotació per tal de tenir endreçades les dades i no haver de fer tot el càlcul, cada vegada que fos sol·licitada una dada d'aquest tipus. Amb aquesta implementació crec que millora bastant l'eficiència a l'hora de crear informes i la presentació dels resultats d'un període comptable. En el disseny original no ho havia contemplat.

Aquest canvi també repercuteix al disseny de la persistència de dades, explicat més endavant.

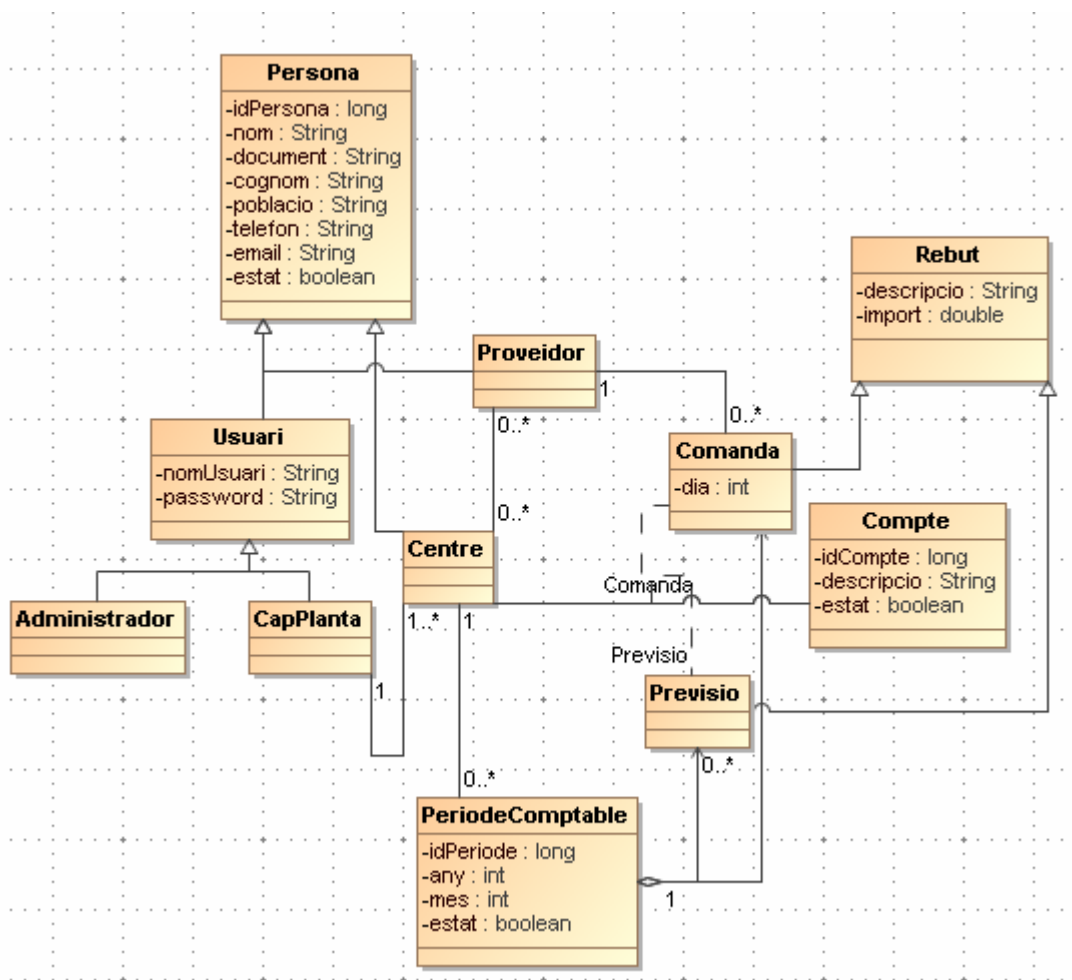


Figura 19. Diagrama de Classes original

Modificacions realitzades al disseny original

Un proveïdor pertany a un centre i un centre pot tenir varis proveïdors.

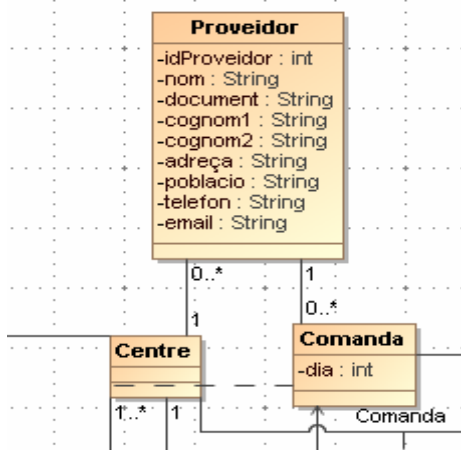


Figura 20. Diagrama de Classes. Modificació Proveïdor

S'incorpora la classe associativa 'PeriodeComptableCompte', on agrupa el totalPrevisions i totalFactures per compte d'exploració. El sentit d'aquesta relació és poder tenir precalculats els totals per compte, cada vegada que hi ha una alta, modificació o eliminació de qualsevol rebut s'actualitza automàticament el totalPrevisions i totalFactura.

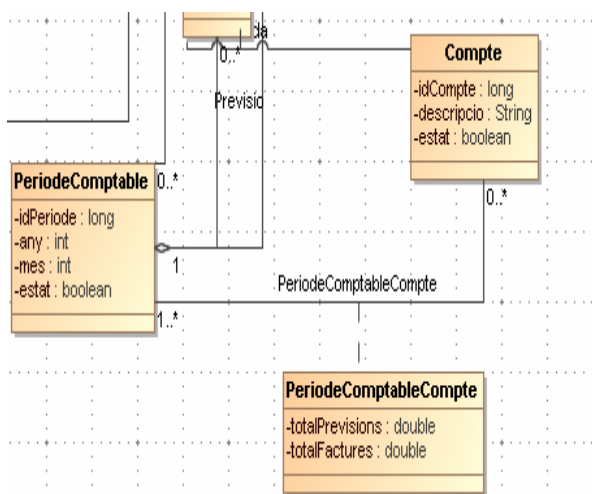
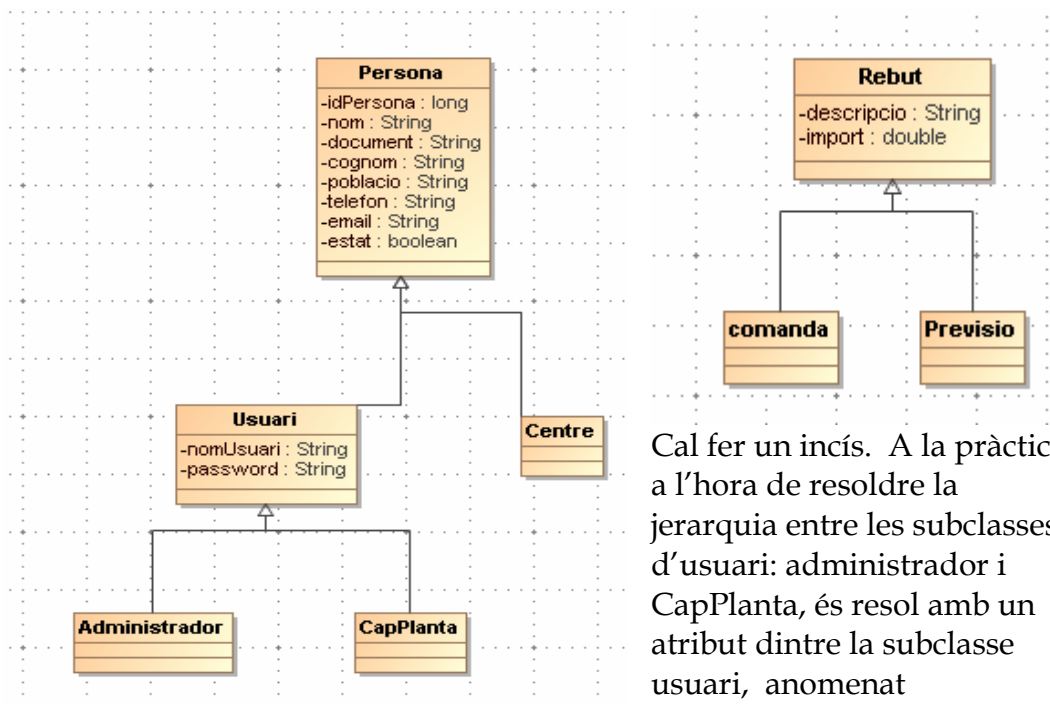


Figura 21. Diagrama de Classes. PeriodeComptableCompte

Jerarquies



Cal fer un incís. A la pràctica a l'hora de resoldre la jerarquia entre les subclasses d'usuari: administrador i CapPlanta, és resol amb un atribut dintre la subclasse usuari, anomenat 'tipusUsuari'.

3.3 Diagrama d'estats

En aquest apartat destaco els estats d'un període Comptable. Cal dir al respecte que és l'objecte el qual té un tracte més diferenciat envers els altres objectes.

Tots els períodes comptables al llarg de la seva vida hauran de passar per tres estats (nou, actiu o tancat). Amb tot aclareixo que l'estat **Nou** és un Pseudoestat o estat de transició, el qual només existeix en el moment abans de ser creats. [Veure apartat implementació.](#)

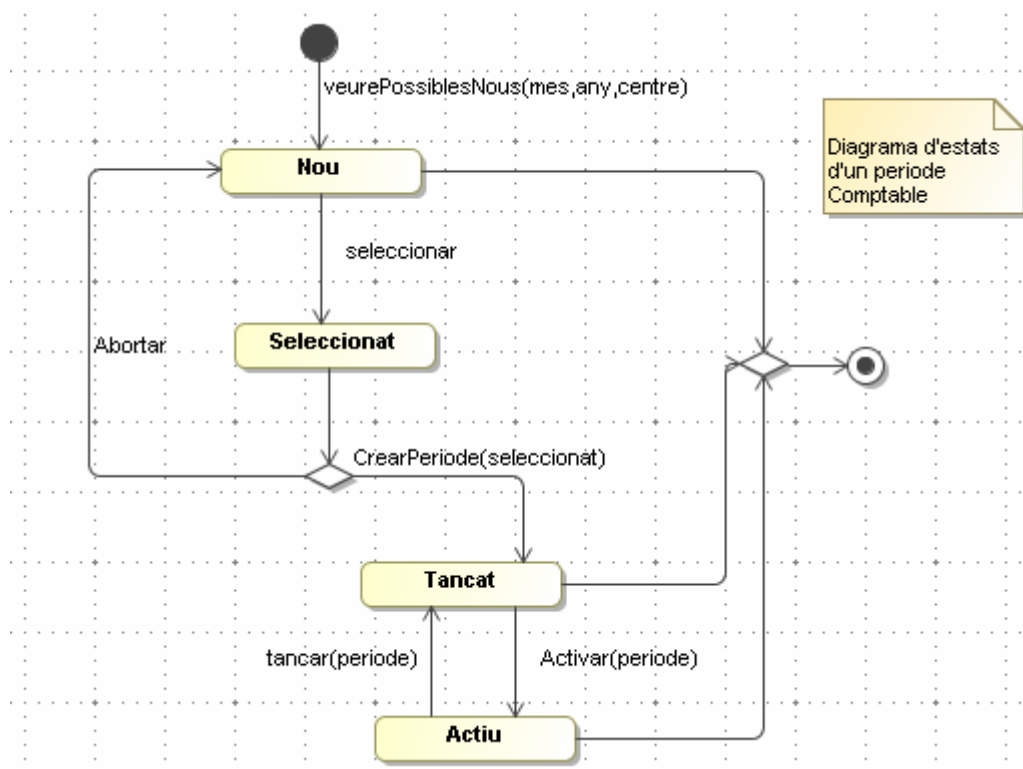


Figura 22. Diagrama d'estats. PeriodeComptable

Els altres objectes de l'aplicació tots poden tenir l'estat actiu o inactiu/baixa lògica.

3.4 Disseny de la persistència

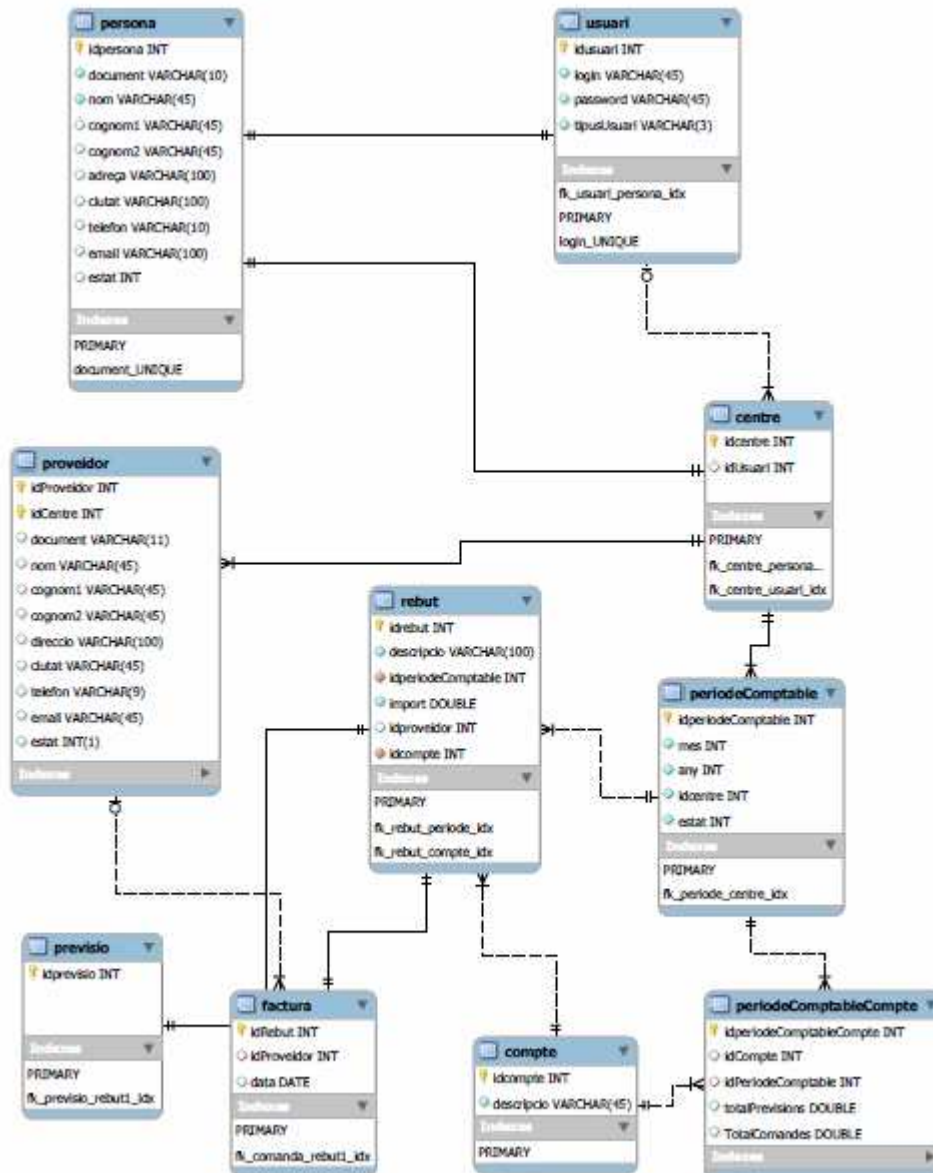


Figura 23. Diagrama de la persistència.

Per solventar la jerarquia de classes en la seva implementació a entitats relacionals, ho faig de la següent manera.

- Hi ha una entitat relacional que representa la SuperClasse on hi ha definits tots els atributs comuns de totes les SubClasses.
- Per a cada subclasse implemento una entitat Relacional nova amb els seus atributs particulars més un atribut que fa referència a la clau primària de la SuperClasse.

Sobre l'entitat PeríodeComptableCompte.

S'alimenta automàticament mitjançant l'aplicació de disparadors a l'inserció, modificació i eliminació de l'entitat **'rebut'**. El seu funcionament és el següent:

- Quan s'afegeix una nou rebut:
 - 1) Cerca si existeix el compte i el període Comptable del rebut, a l'entitat *'periodeComptableCompte'*
 - 2) Si existeix actualitza el *totalPrevisio* o *totalComanda*, en funció de si el rebut és un previsió o una comanda⁵. Per fer-ho suma el valor nou.
 - 3) Si no existeix insereix un nou registre i actualitza el *totalPrevisio* o *totalComanda*.
- Quan es modifica un rebut:
 - 1) Cerca el compte i el període comptable del rebut i actualitza el *totalPrevisio* o *totalComanda*, en funció de si el rebut és una previsió o una comanda. Per fer-ho, primer resta el valor antic i després suma el valor nou.
- Quan s'elimina un rebut:
 - 1) Cerca el compte i el període Comptable del rebut i actualitza el *totalPrevisio* o *totalComanda*, en funció de si el rebut és una previsió o una comanda. Per fer-ho, resta el valor antic.

3.5 Disseny de la interfície gràfica

Tot usuari per accedir a l'aplicació prèviament s'haurà d'identificar, així la primera vista de l'usuari és la pàgina d'identificació i punt d'entrada al sistema.

El projecte ComptaXM està pensat per a funcionar o enfocat a una seqüència d'accés predeterminada. Cal explicar que les operacions que es realitzaran en més assiduïtat serà l'entrada de rebuts. Així la vista seguirà aquesta seqüència, gestió de rebuts. De la següent manera:

- 1) Una vegada l'usuari hagi estat validat accedirà a una pàgina índex, on es presentarà en una taula, els períodes comptables actius en aquell moment i que estan associats⁶ a l'usuari connectat.

En el cas que l'usuari sigui administrador, llavors es presentaran tots els períodes comptables actius. L'usuari tindrà opció de seleccionar qualsevol dels períodes comptables.

⁵ L'entitat Comanda fa referència a una factura del sistema.

⁶ Un Període Comptable està associat a un centre i un centre està associat a un usuari.

- 2) Quan s'hagi escollit el Període Comptable es visualitzarà els comptes de resultats del Període Comptable. On es podrà dur terme dues operacions
 - a. Entrar directament un rebut, una factura o bé una previsió. On caldrà escollir un compte, un dia i un proveïdor (en el cas d'una factura), per poder donar d'alta posteriorment el rebut. A partir d'aquell moment el compte escollit ja estarà donat d'alta pel període Comptable seleccionat.
 - b. Seleccionar un dels Comptes de resultats actiu on es visualitzaran tots els resultats d'aquest. En el cas que un compte no existeixi pel període comptable seleccionat, caldrà entrar directament el rebut (opció a).
- 3) Des de la vista de rebuts es podrà entrar directament rebuts al compte seleccionat del període comptable prèviament. Quan s'afegeixi un rebut, sempre es podrà escollir si es una previsió o una comanda.

En tot moment l'usuari podrà avortar aquesta visualització predeterminada escollint qualsevol altra opció del menú disposat a la part esquerra de la vista.

L'aplicatiu, costat administrador, és capaç de:

- **Gestionar usuaris.** Llistar, filtrar seleccionar, donar d'alta, modificar i donar de baixa.
- **Gestionar proveïdors.** Llistar, filtrar seleccionar, donar d'alta, modificar i donar de baixa.
- **Gestionar centres.** Llistar, seleccionar, donar d'alta, modificar i donar de baixa.
- **Gestionar Comptes d'explotació.** Llistar, filtrar, seleccionar, donar d'alta, modificar i donar de baixa.
- **Gestionar períodes comptables.** Llistar, filtrar, seleccionar, donar d'alta, modificar i donar de baixa.
- **Gestionar rebuts.** Donar d'alta, modificar i eliminar-los
- **Presentació d'informes.** Per centre, després d'escollir un centre, diferents comptes d'explotació i un any. Per període comptable, després de seleccionar un Període Comptable (mes i any), diferents comptes d'explotació i diferents centres.

L'aplicatiu, costat usuari cap Planta, és capaç de:

- **Gestionar proveïdors.** Seleccionar, filtrar, donar d'alta, modificar i donar de baixa proveïdors dels centres en que l'usuari està associat.
- **Gestionar rebuts.** Donar d'alta Previsions i comandes, modificar-les i eliminar-les, dels centres en que l'usuari està associat

- **Presentació d'informes.** Per centre, després d'escollir un centre, diferents comptes d'explotació i un any. Per període comptable, després de seleccionar un Període Comptable (mes i any), diferents comptes d'explotació i diferents centres. Només dels centres en que l'usuari està associat.

Descripció Accés Seqüencial i funcional

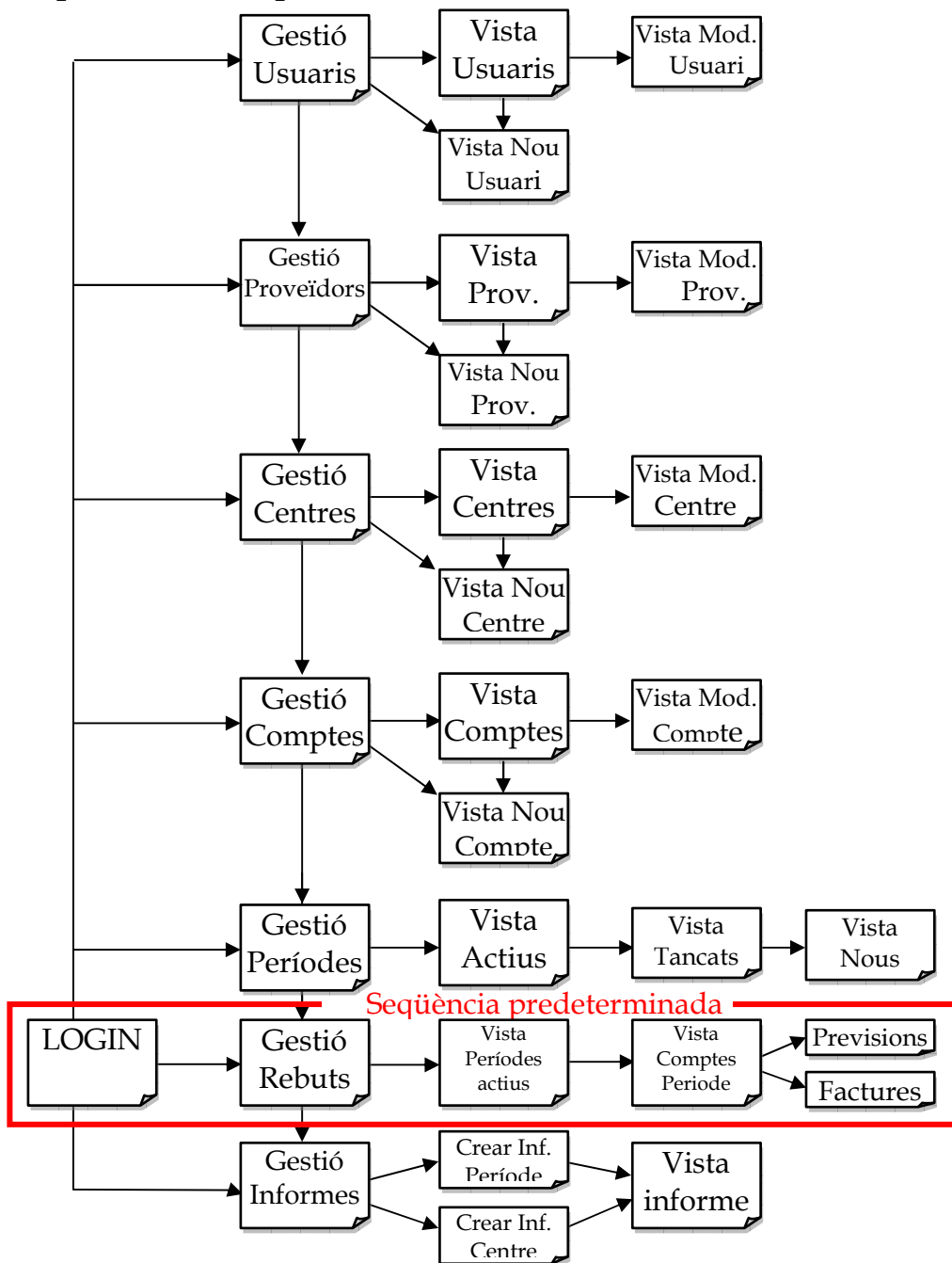


Figura 24. Accés seqüencial i funcional

4. Implementació

Per implementar el sistema al costat servidor utilitzo el següent programari:

- Màquina virtual Java. **JRE7**
- Servidor d'aplicacions i contenidor Web. **Apache Tomcat v6.0.**
- Frameworks utilitzats. **Struts2, Tiles i Hibernate.**
- Servidor Base de dades. **MySql Server 5.5**

Per implementar el sistema des del costat client, no requereix cap programari específic. Amb un navegador Web que es pugui executar la màquina virtual Java és suficient.

Tot el programari utilitzat és de codi obert, això permet una fàcil i barata implementació.

Per desenvolupar la implementació he utilitzat.

- Entorn de treball. **IDE Eclipse EE for Web developers. Versió Indigo Service Release 2.**



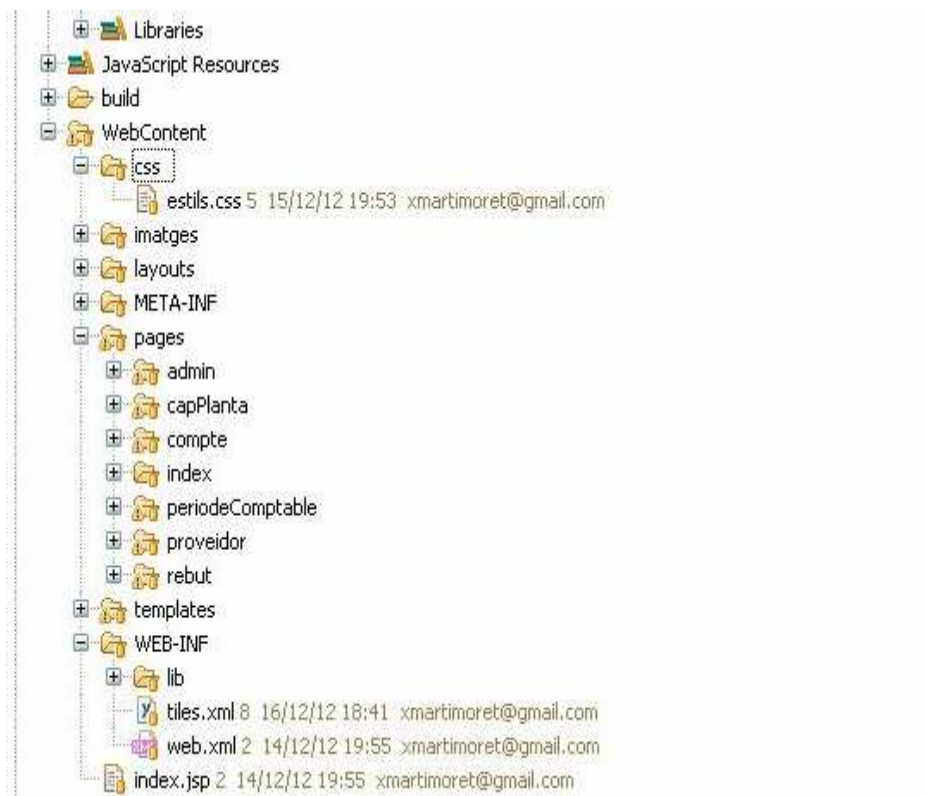


Figura 25. Estructura espai de treball, Eclipse

- Per gestionar la base de dades.

My Sql WorkBench

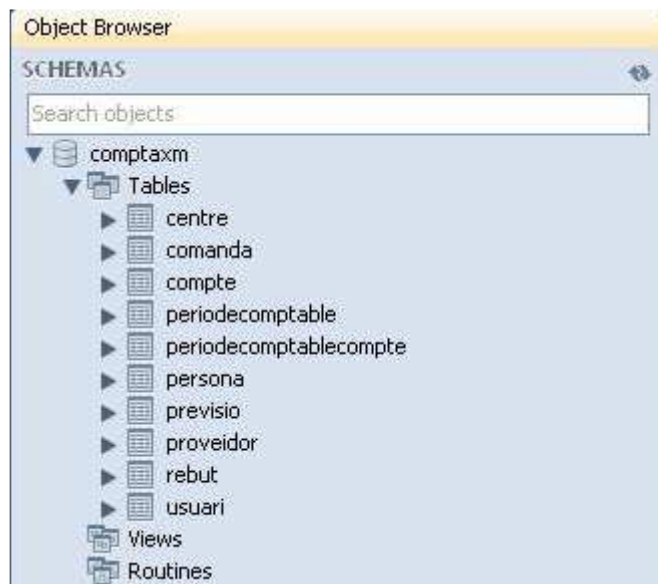


Figura 26. Estructura Espai de treball WorkBench MySql

4.1 Desviacions de la implementació respecte al disseny

A grans trets he implementat la majoria de funcionalitats especificades en el disseny original. Amb tot, el projecte ha sofert algunes petites variacions, sobretot en el disseny de la interfície gràfica.

Algunes, corregint funcionalitats que no havien estat prou madurades, d'altres afegint funcionalitats que he considerant prou importants i que durant l'etapa del disseny no havia contemplat.

Funcionalitats corregides.

- **El tractament donat a la classe proveïdor.** Tal com he reflectit en l'apartat de disseny. Per donar cabuda a proveïdors globals que poden pertànyer a més d'un centre i a la vegada puguin ser modificats de manera personalitzada per cada centre, he hagut de modificar el comportament de la classe Proveïdor i fer una petita 'trampa', a l'hora de crear proveïdors.

[Gestió de Proveïdors >](#)

Afegir Nou Proveïdor

(*) Document:

(*) Nom:

Cognom1:

Cognom2:

Adreça:

Ciutat:

Telefon:

Email:

(*) Centres:

Edar Serinyà
edar terri
LIXIVIATS
Secat montomes
Secat termic
UTE MONTSIA

Sha afegirà un nou Proveïdor per cada Centre Escollit.

Figura 27. Afegir Nou Proveïdor

Per crear un nou proveïdor cal escollir el centre al qual està associat. El sistema dona l'opció de escollir més d'un centre, llavors es crearan tants proveïdors com centres escollits.

Per poder suportar aquesta funcionalitat, tal com esmento en l'apartat de disseny, ha calgut realitzar canvis en l'estructura de les classes.

- La necessitat de guardar **dades pre-calculades** a fi de millorar l'eficiència a l'hora d'operar amb imports agrupats per comptes d'exploració. Tal com esmento a l'apartat del disseny
- **Canvis en la presentació de la interfície gràfica.** Durant el disseny vaig contemplar que determinats objectes es presentarien tant en la vista (llista) com en la modificació i la inserció, en una mateixa interfície gràfica. Tal com podem veure en la figura següent:

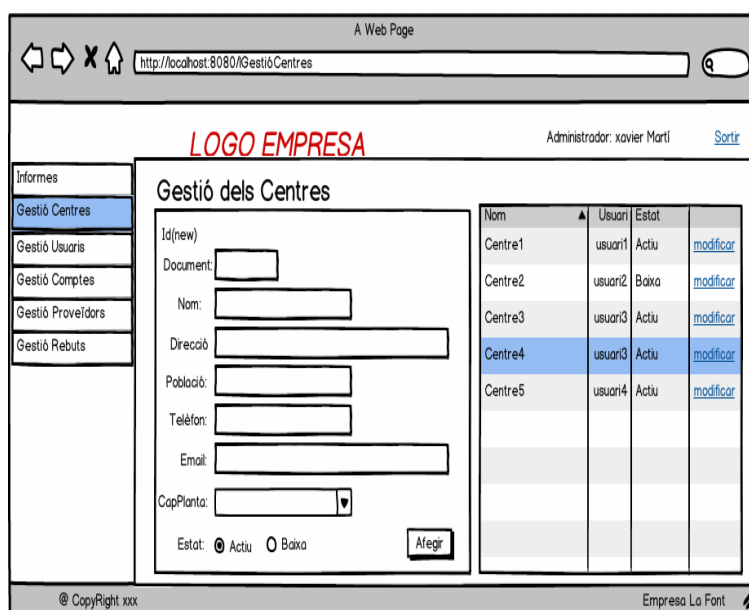


Figura 28. Prototip Interfície Gràfica.

Solució implementada per la gestió de centres.



Figura 29. Gestió Centres

Solució implementada per afegir i/o modificar Centres

Figura 30. Afegir i/o modificar Centres

Després de valorar les diferents opcions, em vaig decantar per separar les operacions de llistat, inserció i modificació per tal de guanyar claredat i no sobrecarregar la presentació visual d'una interfície.

Tots els objectes segueixen el mateix patró una vista pel llistat i una vista per l'inserció i/o modificació.

- **Sobre l'estat d'actiu o baixa d'un objecte** també canvia sobre el disseny proposat. En aquest apartat, cal dir que per facilitar la seva implementació, s'ha desvinculat de la modificació de l'objecte i es realitza en una acció separada. Cada objecte de la llista⁷ té l'opció de canvi d'estat. Al clicar sobre l'opció, si un objecte està actiu es podrà donar de baixa i a la inversa, el sistema presenta un diàleg de confirmació per tal de validar l'acció.

Figura 31. Missatge de confirmació

Totes les accions de activar, eliminar (rebuts) i donar de baixa segueixen el mateix patró.

Sobre Gestió períodes Comptables

Cal dir que tots els objectes de manteniment segueixen el mateix patró a excepció dels períodes comptables, on tenen un altra tractament. Aquests cal

⁷ A cada fila de la taula on es presenta la llista d'objectes hi haurà una opció per canviar el seu estat.

presentar-los en tres estats diferenciats (actius, tancats i nous), els quals es presenten en tres interfícies diferents.

Com crear nous Períodes?

1. Seleccionar l'opció Crear nous períodes.

Figura 32. Gestió nous Períodes Comptables.

2. Escollir el mes i l'any dels nous períodes

Mes	Any	Centre	Crear
Febrer	2013	Edar Serinyà	<input checked="" type="checkbox"/>
Febrer	2013	edar terri	<input checked="" type="checkbox"/>
Febrer	2013	LIXIVIATS	<input checked="" type="checkbox"/>
Febrer	2013	Secat montornes	<input checked="" type="checkbox"/>
Febrer	2013	Secat termic	<input checked="" type="checkbox"/>
Febrer	2013	UTE MONTSIA	<input checked="" type="checkbox"/>

Figura 33. Gestió nous Períodes Comptables. Filtre

El sistema ens presenta els candidats a ser creats amb l'opció de poder-los crear o no. Si un període que correspon al mes i l'any seleccionat, ja ha estat creat anteriorment, el sistema descarta automàticament aquest període i no el presenta.

Al clicar sobre el botó 'Crear els Períodes seleccionats', aquests son creats i passen a l'estat de 'Tancat', llavors l'usuari en el moment que ho desitgi els podrà activar.

Funcionalitats afegides.

- **Un control sobre la sessió.** He fet ús d'un interceptor personalitzat , 'identificadorInterceptor', i he implementat la interfície 'SessionAware', per tal de poder saber en tot moment quin usuari està connectat, i poder guardar dades clau sobre la sessió. D'aquesta manera, cada vegada que s'executa una acció, es pot saber l'estat de la sessió i poder obtenir informació de l'estat d'algunes accions realitzades prèviament. Per comprendre el seu funcionament [veure l'annex2](#).
- **Un servei d'accions predeterminades⁸** que l'usuari podrà escollir al accedir a una vista determinada.

Figura 34. Servei d'accions predeterminades.

- **Funcionalitat multi idioma (català, castellà i anglès).** L'aplicatiu sempre es presentarà en l'idioma predeterminat del navegador Web de l'usuari, mentre aquest sigui l'idioma català, castellà o anglès, sinó sempre es presentarà amb català.

⁸ Aquest conjunt de adreces permet a l'usuari tornar enrera. Haig d'aclarir que la implementació d'aquesta funcionalitat s'ha fet de manera manual i personalitzada a cada pàgina JSP, allà on he cregut convenient, per tal de millorar la navegabilitat de l'aplicatiu. En cap cas segueix un model a partir d'un patró de disseny.

En tot moment l'usuari sempre podrà canviar l'idioma de l'aplicatiu, escollint una de les opcions, disposades a la part superior de la finestra.

The screenshot shows the 'Gestió de Rebuts' application interface. At the top, there is a yellow header bar with the title 'Gestió de Rebuts' and a language selection menu containing 'CAT', 'ESP', and 'EN'. The 'EN' option is circled in red. To the right of the language menu, it says 'Administrador: xavier marti'. Below the header, there is a sidebar with navigation links: 'Gestió Usuaris', 'Gestió Proveïdors', 'Gestió Centres', 'Gestió Comptes', 'Gestió Períodes Comptables', 'Gestió Rebuts' (highlighted in orange), and 'Veure Informes'. The main content area is titled 'Períodes Comptables Actius' and contains a table with columns for 'Centre', 'Ciutat', 'Període', and a 'Seleccionar' link. The table lists various accounting periods for different centers and cities.

Centre	Ciutat	Període	
4020-edar terri	Banyoles	Gener-2012	Seleccionar
4041-Secat montornes	Montornes	Gener-2012	Seleccionar
4040-Secat termic	Banyoles	Gener-2012	Seleccionar
4020-edar terri	Banyoles	Febrer-2012	Seleccionar
4041-Secat montornes	Montornes	Febrer-2012	Seleccionar
4040-Secat termic	Banyoles	Febrer-2012	Seleccionar
4020-edar terri	Banyoles	Març-2012	Seleccionar
4041-Secat montornes	Montornes	Març-2012	Seleccionar
4040-Secat termic	Banyoles	Març-2012	Seleccionar

At the bottom of the interface, there is a yellow footer bar with the text: 'Treball Final de Carrera. AreaJ2EE', 'Alumne: Xavier Martí Moret', and 'ComptaXM 2012'.

Figura 35. Funcionalitat Multi Idioma

Per veure la seva implementació [veure annex3](#).

5. Conclusions

El motiu d'escollir aquesta àrea, J2EE, per realitzar el treball final de carrera va ser poder realitzar un projecte des de l'inici fins el final, des de l'estudi, l'anàlisi, el disseny, la implementació i la documentació d'un cas o problema. Des del primer dia ho he considerat un repte i espero haver-lo aconseguit.

Des d'un punt de vista acadèmic m'ha servit per aprofundir sobre els coneixements adquirits al llarg de tota la carrera, concretament sobre la programació Orientada a objectes, enginyeria del programari i tècniques de desenvolupament de programari orientat a objectes.

A nivell personal, m'ha suposat adquirir una bona quantitat de nous coneixements sobre programació distribuïda i més concretament sobre en un entorn web. Cal dir que no tenia cap experiència sobre les eines utilitzades. I he pogut comprovar de primera mà, els avantatges de aplicar el patró MVC, l'ús de les plantilles Tiles i sobretot el suport d'Hibernate per poder fer ús de la persistència d'objectes.

Cal dir, en un principi no ho tenia gens clar, segurament qüestions de caràcter. Aquest fet m'ha suposat tenir una visió conservadora sobre el disseny global del projecte i ara vist el resultat em queda un xic de regust per no haver estat més optimista i valent.

Finalment dir que estat una experiència molt positiva. Des de l'inici, on es plantegen tots els dubtes, escollir una opció vàlida i donar-hi solució. Tan sols haver arribat fins aquí ja em dono per satisfet. Ara tan sols espero que pugui ser prou valorada.

Gràcies per haver-hi donat la importància que es mereix.

Annex 1. Patró MVC via STRUTS2

Struts 2 és un framework pel desenvolupament d'aplicacions Web que funciona sota la plataforma de JEE. De caràcter lliure on la principal característica rau en la estructuració dels projectes, utilitzant el model MVC. (Model, vista controlador).

- El **model** fa referència a les dades que maneja l'aplicació i les regles del negoci que operen sobre aquestes i que es tradueixen en Struts2 en **les accions**.

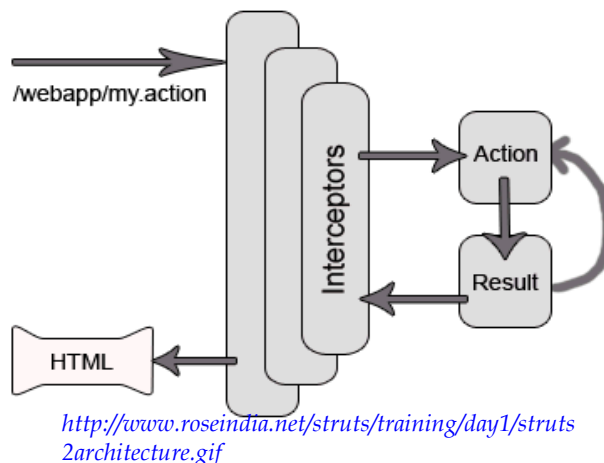


Figura 36. Struts2. Patró MVC

- La **vista** s'encarrega de generar la interfície gràfica on l'usuari interacciona amb l'aplicació . Equival als **resultats** en Struts2.
- El **Controlador** comunica la vista i el model responen a events generats per l'usuari en la vista, invocant canvis en el model i retornant a la vista la informació del model necessari que pugui generar la resposta adequada per l'usuari. El controlador s'implementa en Struts2 mitjançant el **FilterDispatcher**.

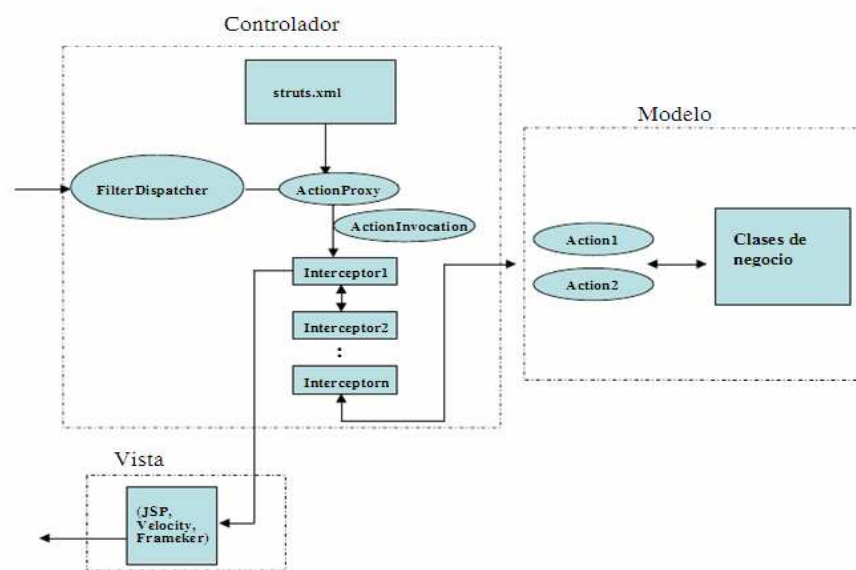


Figura 37. FilterDispatcher. Struts2. Editorial RAMA

FilterDispatcher representa el punt d'entrada a l'aplicació, dirigint-se a ell, totes les peticions que arriben del client. Analitza la petició rebuda i determina amb l'ajuda d'altres objectes auxiliars i de la informació emmagatzemada en el fitxer de configuració **Struts.xml**, el tipus d'acció a executar.

FilterDispatcher forma part de l'API de Struts2, concretament, s'inclou dins del paquet org.apache.Struts2.dispatcher i com es dedueix del seu propi nom s'implementa mitjançant un filtre, pel que ha de ser registrat dintre el fitxer de configuració **web.xml**.

Struts 2 es basa en el model de disseny de tipus MVC II. Permet desenvolupar més ràpid, més flexible i resolent varis problemes de disseny:

- Un sistema evolucionat de **gestió de l'enrutament o la navegació**.
- **Un sistema de validació** de formularis i d'entrades, fàcil d'aplicar.
- Un potent sistema de **complements o d'extensions** (pels gràfics, fonts de dades,..).
- **Gestió de la internacionalització** pel desenvolupament de llocs multi llenguatge.
- Compatibilitat amb la tecnologia AJAX
- Eina de depuració inclosa
- Una potent eina d'etiquetes.

A part ofereix la possibilitat d'utilitzar un mínim de regles de disseny que no són obligatòries.

- No utilitzar codi 100% Java en les pàgines JSP. Tota la lògica de control resideix en les classes d'accions (Servlets).
- Utilitzar biblioteques d'etiquetes per accedir als objectes i navegar per les col·leccions.
- Escriure un mínim de codi repetitiu i utilitzar les que ofereix el framework

En resum, Struts2 està pensat per a facilitar les tasques pròpies del cicle de vida del programari. Afegint la construcció, desenvolupament i manteniment de la solució. És fàcilment extensible degut a que tots els elements estan basats en interfícies i classes bàsiques.

Annex 2. Sobre el control de la sessió

Quan un usuari accedeix a l'aplicatiu, des del moment en que s'acredita i es validat pel sistema, es crea una nova sessió. A partir d'aquí queda registrat com a usuari actiu i per cada acció que executa, el sistema comprovarà l'usuari actiu.

- Si es tracta d'un usuari administrador presentarà les opcions d'administrador a la següent acció.
- Si es tracta d'un usuari capPlanta presentarà les opcions d'usuari capPlanta, a la següent acció.
- Si no hi ha cap usuari actiu, el sistema retornarà a la pantalla inicial per tal de que es pugui tornar acreditat si així ho desitja. Aquest estat pot passar per dos motius.
 - L'usuari actiu fins el moment, es **desconnecta del sistema** voluntàriament, clicant sobre l'opció 'SORTIR'.
 - **La sessió de l'usuari a expirat**, és a dir el temps de durada a la sessió (configurada en el servidor web) ha sobrepassat el límit.

Per implementar el control de la sessió s'ha realitzat de la següent manera.

- 1) **Fent un ús d'un interceptor personalitzat**, que s'activa just després de la validació de l'usuari. Aquest interceptor es crida per cada acció executada.

```

/**
 * interceptor: Redirigeix a l'acció corresponen.
 * Si l'usuari està activat, rederigeix a l'acció correcte
 * En cas contrari retorna l'usuari a la pantalla login
 */
public String intercept(ActionInvocation invocation) throws Exception
{
    Map<String,Object> session=invocation.getInvocationContext().getSession();
    Usuari usuari=(Usuari) session.get("usuari");
    if (usuari==null)
    {
        return Action.LOGIN;
    }
    else
    {
        Action action =(Action) invocation.getAction();
        if (action instanceof UsuariHabilitat) {
            ((UsuariHabilitat) action).setUsuari(usuari);
        }
        return invocation.invoke();
    }
}

```

Figura 38. Codi Interceptor

Funcionament:

- 1.1. Recupera l'usuari actual de la sessió.

1.2. Comprova si l'usuari es nul

- Si es nul retorna el resultat 'Login' que retornarà a la pantalla d'inici, acció per defecte.
- Sinó segueix amb l'acció actual.

2) Per implementar aquest interceptor, cal afegir-lo al fitxer de configuració '**Struts.xml**' de la següent manera:

```

<interceptors>
<interceptor name="IdentificacioInterceptor" class="uoc.tfc.xmartim
  <param name="identificadorPredeterminado">xmartim</param>
  <param name="contrasenaPredeterminada">xmartim</param>
</interceptor>

  <interceptor-stack name="secureStack">
    <interceptor-ref name="createSession"/>
    <interceptor-ref name="defaultStack"/>
    <interceptor-ref name="IdentificacioInterceptor"/>
  </interceptor-stack>
</interceptors>
<default-interceptor-ref name="secureStack" />

```

Figura 39. Codi Interceptors. Struts.xml

2.1. Es declara l'interceptor personalitzat, en aquest cas 'identificadorInterceptor' indicant l'action (classe.java) a la qual fa referència.

2.2. Es declara un grup d'interceptors amb:

- **CreateSession i defaultStack.** Aquest dos interceptors predeterminats de Struts2 serveixen per crear una sessió en cas que aquesta no existeixi, així també crea l'estructura Map <String,object> session, on es podran guardar parells de dades <Clau, valor> on es podran guardar dades estructurades sobre la sessió.
- **IdentificacioInterceptor.** Interceptor personalitzat per controlar l'usuari actiu.

3) Es declara aquest grup d'interceptors per defecte, en aquest cas amb el nom '**secureStack**', d'aquesta manera qualsevol acció que forma part del fitxer '**struts.xml**' cridarà aquests interceptors.

4) Una de les tasques d'aquest interceptor és enregistrar l'usuari en una variable de sessió mitjançant la interfície sessionAware. Aquesta s'implementa a tots els actions (*.java).


```

public class ProveidorAction extends ActionSupport implements SessionAware, UsuariHabilitat {

    private List<String> errorsComptaXM= new ArrayList<String>();
    private List<Proveidor> proveidors = new ArrayList<Proveidor>();
    private ProveidorModel proveidorModel;
    private CentreModel centreModel;
    private UsuariModel usuariModel;
    private Usuari usuari;
    private List<Centre>centresUsuari= new ArrayList<Centre>();
    private List<Integer>nousProveidorsPerCentre= new ArrayList<Integer>();
    private Proveidor proveidorActual;
    private int idProveidorActual;
    private int idCentreActual;
    private Map<String, Object> session;

```

Figura 40. Implementació SessionAware

Implementa SessionAware, tal com descriu abans i també la interfície UsuariHabilitat, on concretament ens servirà per crear i guardar l'usuari actiu. Amb tot i com és normal, a cada acció ens caldrà implementar els atributs de les dues interfícies per poder-hi accedir.

Altres usos de col·lecció Map Session

He aprofitat aquesta col·lecció, Map<clau,valor> session per guardar-hi diferents dades.

- **Per implementar els missatges d'error i validació de formularis**, de les diferents interfícies gràfiques. Per fer-ho he implementat un parell clau='errorsComptaXM', valor=String()' per tal de guardar els diferents missatges d'errors de validació d'introducció de dades dels diferents formularis que consta l'aplicatiu.

Ho he fet d'aquesta manera i no fent ús de l'interceptor 'validator' incorporat a Struts2 perquè en quasi tots els casos necessitava tenir constància de quin objecte estava modificant o inserint i amb l'ús de l'interceptor 'validator' no me'n sortia, ja que perdia informació⁹.

Funcionament:

- 1) En les accions de validar dades d'un formulari, guarda els errors de validació.

⁹ En aquest aspecte reconec que no he aprofundit prou en la aplicació de l'interceptor Validator incorporat dintre Struts2, perjudicant la claredat del codi.

En aquest apartat, he considerat més apropiat fer-ho d'aquesta manera, per guanyar eficiència, i no haver de fer una transacció a la base de dades, per saber-ho.

Annex 3. Implementació de la internacionalització.

L'interceptor que s'encarrega de la internacionalització en Struts2 es el `i18n`. Per poder-lo utilitzar cal carregar-lo en el fitxer de configuració '`struts.xml`'

```
<struts>
  <constant name="struts.enable.DynamicMethodInvocation" value="false" />
  <constant name="struts.devMode" value="false" />
  <constant name="struts.custom.i18n.resources" value="lenguatge"></constant>
  <package name="default" namespace="/" extends="struts-default">
    <result-types>
      <result-type name="tiles"
        class="org.apache.struts2.views.tiles.TilesResult" />
    </result-types>
  </package>
</struts>
```

Figura 43. Interceptor `i18n`

On.

- **Name.** Fa referència a la llibreria `.jar` que conté l'interceptor.
- **Value.** El fitxer per defecte que contindrà les definicions de l'idioma.

Hi haurà un fitxer per cada idioma definit en el sistema. Aquest fitxer de propietats conté parells de valors `<clau, valor>` per cada definició.



Figura 44. Fitxers de configuració idiomes.

Per fer referència a un idioma en concret, cal afegir al nom del fitxer, la localització de l'idioma i del país separats per `'_'`.

Si l'idioma escollit és el català, el nom del fitxer serà `llenguatge_ca_ES`

- **Llenguatge**, es el nom definit en la propietat `'value'` en el fitxer `'struts.xml'`
- **ca**, localització de l'idioma català.
- **ES**, localització del país.

Per utilitzar un idioma tant sols hem de canviar el valor de la variable `'RequestLocale'`, des de ComptaXM es fa de la següent manera:

```

<td align="center" width="4%"><s:url id="localeCA" action="ObtenirPeriodesComptablesAdm" >
  <s:param name="request_locale">ca ES</s:param>
</s:url>
<s:a href="{localeCA}">CAT</s:a></td>
<td align="center" width="4%"><s:url id="localeCA" action="ObtenirPeriodesComptablesAdm" >
  <s:param name="request_locale">es ES</s:param>
</s:url>
<s:a href="{localeCA}">ESP</s:a></td>
<td align="center" width="4%"><s:url id="localeCA" action="ObtenirPeriodesComptablesAdm" >
  <s:param name="request_locale">en GB</s:param>
</s:url>
<s:a href="{localeCA}">EN</s:a></td>
<td width="20%" align="center">

```

Figura 45. Com canviar el RequestLocale.

Finalment per poder accedir al parell de valors que contenen els fitxers de propietats de cada idioma.

- Des d'un fitxer *_action.java. Fent ús de la sentència,


```
(getText("proveidor.error.nom"))
```

 on "proveidor.error.nom" és la clau.
- Des d'un fitxer *.jsp.

```
<s:property value="{getText('label.menu.usuari')}" />
```

Recursos Emprats

Eines d'edició:

MagicDraw UML personal Edition: Per a realitzar els diferents diagrames de disseny del projecte.

MySql Workbench 5.2 CE: Per a realitzar el disseny de la persistència del projecte.

Microsoft Word: Per elaborar el diferents documents.

Microsoft Excel: Per elaborar el glossari.

OpenProject: Per elaborar el diagrama de GANNT.

Balsàmic Web: Per elaborar el disseny del prototip de la interfície gràfica

Paint: Per fer retocs de les diferents figures emprades en la documentació.

Power Point: per a elaborar la presentació del TFC.

Eines d'implementació:

Eclipse INDIGO EE: Per a realitzar proves d'aprenentatge sobre la tecnologia a emprar per portar a terme el TFC i codificar i implementar el projecte.

MySQL 5.5: Per implementar el disseny de la persistència

Apache Tomcat 6.0: Per implementar la part servidora de continguts web.

JRE7: Màquina virtual java.

Glossari de termes

Actor	Conjunt de papers d'una entitat exterior en relació en el sistema de programari considerat.
Administrador	Usuari que té tots els privilegis.
Anàlisi	Etapa on s'especifica els requeriments del programari que es vol desenvolupar, consultant les especificacions del client.
capPlanta	Usuari que té restringits els privilegis.
Centre	Entitat productora o gestora d'una instal·lació. Encarregada de generar rebuts.
Comanda	Tipus de rebut emesos pels centres. Representen els costos i produccions dels centres.
Compte	Número del pla comptable. Està compost d'un codi i una descripció. Ens permet agrupar rebuts segons la seva procedència.
Controlador	Respon a events, usualment accions dels usuaris, i invoca peticions al model i, probablement a la vista.
DDL	<i>(llenguatge de definició de dades)</i> . Llenguatge proporcionat pel sistema gestor d'una base de dades que permet als usuaris de la mateixa portar a terme les tasques de definició de les estructures que emmagatzemaran les dades, així com els procediments i funcions que permetran consultar-los.
Diagrama de Casos d'us	Diagrama de comportament que mostra les interaccions dels actors amb els diferents objectes d'un sistema de programari.
Diagrama de Gantt:	Eina gràfica que té per objectiu mostrar el temps de dedicació previst per diferents tasques o activitats al llarg d'un temps total determinat.
Diagrama de paquets	En el llenguatge UML Mostra com un sistema està dividit en agrupacions lògiques mostren les dependències entre aquests. Subministren una descomposició de la jerarquia lògica d'un sistema de

	programari.
Diagrama de seqüències	Diagrama que mostra la interacció d'un conjunt d'objectes en una aplicació a través del temps i es modela per cada cas d'us.
Diagrama de classes	Diagrama estàtic que descriu l'estructura d'un sistema mostrant les seves classes i les relacions entre elles.
Disseny	Etapa on es realitza la representació gràfica de les especificacions obtingudes en l'etapa d'anàlisi.
FrameWork	(Espai de treball). Estructura conceptual i tecnològica de suport definit normalment amb artefactes o mòduls de programari amb base a la qual un altre projecte de programari pot ser més fàcilment organitzat i desenvolupat.
Hibernate	Eina per la plataforma Java que facilita el mapeig entre els atributs d'una base de dades relacional i el model d'objectes d'una aplicació mitjançant fitxers declaratius XML que permeten establir aquestes relacions
J2EE	Java Enterprise Edition. Plataforma de programació per desenvolupar i executar programari distribuït, en llenguatge Java
JSP	(Java Server Pages) És una tecnologia que permet als desenvolupadors de programari generar contingut dinàmic Web, en forma de documents html, xlm i altres.
Model	Representació específica de la informació en la qual opera un sistema.
Open Source	Terme amb el que es coneix el programari distribuït i desenvolupat lliurement.
Patró MVC	Patró o model d'abstracció de programari que separa les dades d'una aplicació, les interfícies gràfiques i la lògica de negoci en tres components diferents.
Període comptable	Espai de temps mensual que serveix per agrupar els rebuts i els centres. Un període comptable estarà compost per un mes, un any, un centre i un estat.

Persistència	Propietat de les dades que permet que romanguin més enllà de l'execució d'un procés o aplicatiu que les ha creades.
Planificació	És un procés continuo que incideix sobre un mateix objecte d'intervenció des de diferents nivells d'anàlisi.
Previsió	Tipus de rebut emesos pels centres. Representen les previsions sobre costos i produccions dels centres.
Rebut	Unitat bàsica comptable. Ens permet relacionar un import, un proveïdor, un centre, un compte, un període Comptable.
Sessió	És el cinquè nivell de model de capes OSI. Respon a peticions del servei de capa de presentació i obté els serveis de la capa de transport. .
Struts 2	Frame Work de desenvolupament Web. Facilita les tasques de programació i configuració d'entorns desenvolupats sobre DinamicWeb.
Subsistema	Part d'un sistema. En aplicacions grans es solen dividir els components en subsistemes per afinitats, classes en comú, relacions entre objectes etc. Ajuda a una major comprensió i cohesió del programari.
Tiles	Sistema de plantilles que permet un disseny global d'una aplicació mitjançant la combinació de pàgines JSP. Està pensat per estalviar repetir trossos de pàgines que son similars entre diferents vistes d'una aplicació.
Vista	Presentació visual del model adequada per interectuar. Interfàs gràfica de l'usuari.

Bibliografia consultada

Sobre desenvolupament de projectes:

- Tècniques de desenvolupament de programari. UOC (material de l'assignatura). FATOS XHAFA.
- Enginyeria del programari. UOC (material de l'assignatura) Benet Campderrich Falgueras, Recerca informàtica, SL.

Sobre Tomcat:

- Web: <http://tomcat.apache.org/>

Sobre J2EE:

- Llibre: Desarrollo de aplicaciones WEB con JEE6 de Thierry GROUSSARD
- Web: <http://docs.oracle.com/javaee/1.4/api/>

Sobre FrameWork STRUTS2

- Web: <http://struts.apache.org/2.x/>
- Web: <http://viralpatel.net/blogs/category/j2ee/struts-2/>
- Web: <http://www.javatutoriales.com/2011/06/struts-2-parte-1-configuracion.html>
- Web: <http://www.roseindia.net/struts/struts2/>

Sobre JSP

- Web: <http://www.jsptut.com>

Sobre tecnologia TILES

- Web: <http://tiles.apache.org/framework/tutorial/basic/concepts.html>

Sobre Hibernate

- Web: <http://viralpatel.net/blogs/category/j2ee/hibernate/>

- Web: <http://www.roseindia.net/hibernate/examples/hibernate-api.html>
- Web: <http://docs.jboss.org/hibernate/orm/3.5/api/>

Seguiment aprenentatge:

- Llibre: STRUTS2 El framework de desarrollo de aplicaciones JAVA EE. de Jérôme LAFOSSE

Informació Conceptes:

- Web: <http://es.wikipedia.org>.