

**Títol: Ondema (Online Deliverable Manager)**

**Volum : 1/1**

**Alumne: Josep Lagunas i Calpe**

**Director: Ángel Priegue Mayán**

**Ponent: Ruth Raventós Pagès**

**Departament: Llenguatges i Sistemes d'Informació**

**Gener de 2011**



---

## DADES DEL PROJECTE

**Títol del Projecte:** *Ondema (Online Deliverable Manager)*

**Nom de l'estudiant:** *Josep Lagunas i Calpe*  
**Titulació:** *Enginyeria Tècnica en Informàtica de Gestió*

**Crèdits:** *22.5*  
**Modalitat:** *B*  
**Empresa:** *CIMNE*

**Director:** *Ángel Priegue Mayán*  
**Ponent:** *Ruth Raventós Pagès*

**Departament:** *Llenguatges i Sistemes d'Informació*

---

## MEMBRES DEL TRIBUNAL

**President:** *Enric Mayol Sarroca*

**Vocal:** *Ramon Nonell Torrent*

**Secretària:** *Ruth Raventós Pagès*

---

## QUALIFICACIÓ

**Qualificació numèrica:**  
**Qualificació descriptiva:**

**Data:**

---



# Índex

Resum .....	1
<b>Capítol 1 .....</b>	<b>3</b>
Introducció .....	3
Estructura d'aquest document .....	3
Descripció del PFC.....	5
Descripció del projecte empresarial .....	6
Context de l'empresa .....	6
Context del client.....	6
Descripció del producte .....	7
Objectius personals i professionals.....	9
Agraïments.....	9
<b>Capítol 2 .....</b>	<b>10</b>
Planificació inicial .....	10
Fases del projecte.....	10
Contextualització.....	10
Anàlisi i captura de requeriments.....	10
Especificació del sistema.....	10
Disseny del sistema.....	11
Implementació del sistema .....	11
Estimacions temporals inicials .....	12
<b>Capítol 3 .....</b>	<b>13</b>
Estat de l'art.....	13
<b>Capítol 4 .....</b>	<b>14</b>
Anàlisi de requeriments.....	14
Requeriments no funcionals .....	14
Requeriments funcionals .....	16
<b>Capítol 5 .....</b>	<b>18</b>
Especificació del sistema .....	18
El llenguatge UML.....	19
Diagrames d'estructura.....	20
Diagrames de comportament.....	21
Diagrames d'interacció .....	22
Definició de les classes del model .....	27
Restriccions d'integritat textuals del model de conceptual.....	37
Rols d'usuari .....	41
Casos d'ús .....	41
Diagrama complet .....	42
Diagrama client .....	43
Diagrama Administrador de projecte.....	44
Diagrama Líder o Coordinador de tasca .....	45
Diagrama Membre d'equip .....	46
Diagrama Administrador de sistema .....	47
Diagrama Sistema.....	48
Diagrama Validador de tasca.....	48
<b>Capítol 6 .....</b>	<b>115</b>
Disseny del sistema .....	115
Normalització del model conceptual.....	115

Diagrames de seqüència .....	116
Procés de normalització de model conceptual .....	117
Aplicació de patrons de disseny .....	119
Diagrames de seqüència .....	121
<b>Capítol 7 .....</b>	<b>134</b>
Implementació del sistema .....	134
Llenguatges de programació al costat servidor .....	134
Visual Basic.Net .....	134
ASP.Net .....	136
Alternatives a VisualBasic.Net .....	137
Implementació de la persistència del sistema .....	138
Persistència de dades amb NHibernate .....	138
Sistema Gestor de Base de Dades (SQL Server 2005) .....	138
Implementació de la base de dades .....	138
Llenguatges de programació al costat client .....	145
HTML .....	145
Javascript .....	145
Format per l'intercanvi de dades .....	146
JSON .....	146
Implementació de controls utilitzant jQuery i HTML .....	147
Components i widgets implementats a la interfície d'usuari .....	148
Control de selecció d'usuaris .....	148
Control autocompletar noms d'usuari .....	149
Finestra de conversa a través del xat .....	149
Widget per a la visualització de diagrames de Gantt .....	150
Servidor de xat .....	152
<b>Capítol 8 .....</b>	<b>154</b>
Conclusions .....	154
Planificació final .....	155
Diagrama de Gantt .....	156
Desviacions patides durant l'execució .....	157
Costos associats al projecte .....	158
Futures línies de treball .....	159
Valoració personal .....	160



## **Resum**

Aquest document és la memòria descriptiva del Projecte Final de Carrera de títol Ondema (Online Deliverable Manager) desenvolupat per Josep Lagunas a l'empresa CIMNE com a projecte empresarial per a l'empresa client AppPlus.

La finalitat d'aquest document és la de descriure de manera integral tot el cicle de desenvolupament, desglossant cadascuna de les parts i analitzant-les amb deteniment amb especial atenció a la seva vessant tècnica.

L'objectiu d'aquest projecte ha estat el desenvolupament d'un sistema de monitorització i seguiment d'un projecte d'enginyeria a nivell integral. A més a més, és pretén dotar d'eines als equips de treball que en formen part per poder compartir coneixement i agilitzar els temps de resposta i correcció davant diferents tipus de desviaments que es poden produir durant l'execució d'un projecte.

Les primeres reunions en les que va néixer la idea de desenvolupar l'eina que finalment han donat lloc a aquest projecte van realitzar-se a finals de l'any 2008-inicis de l'any 2009. Al moment de finalitzar aquesta memòria la seva implementació és dona pràcticament per conclosa quedant pendents certes correccions.





---

# Capítol 1

---

## Introducció

El principal objectiu d'aquest primer capítol és el de contextualitzar el treball ha desenvolupar. S'explicarà la problemàtica que ha donat origen a aquest projecte així com una breu explicació dels canvis que durant l'execució del projecte aquest ha anat patint.

## Estructura d'aquest document

Aquest document s'ha estructurat en 8 capítols:

**Capítol 1**– Introducció: Capítol descriptiu del projecte on s'aprofundeix en l'objectiu i la necessitat que aquest projecte vol cobrir. Així com explicació del mateix document.

**Capítol 2**– Planificació: Enumeració i explicació de les fases que han constituït la realització del projecte així com la identificació de les principals tasques que han donat lloc a cadascuna d'aquestes etapes.

**Capítol 3**– Estat de l'art en el món de la gestió empresarial i necessitat d'omplir mancances trobades en els gestors analitzats.

**Capítol 4**– Requeriments a cobrir: en aquest capítol es fa una descripció del procés que es va fer servir per la captura dels requeriments del sistema (tant funcionals com no funcionals).

**Capítol 5**– Especificació del Sistema: un cop feta la captura dels requeriments que cal cobrir és durà a terme la especificació completa del sistema. En aquest capítol s'emprarà el llenguatge UML i OCL per a definir tot allò que tingui a veure amb especificació (casos d'us, diagrames d'objectes, etc. ).

**Capítol 6**– Disseny del Sistema: amb la finalització de l'especificació del capítol anterior el següent pas serà el disseny. Aquesta part serà coberta en aquest capítol. Aquí s'explica el procés de normalització en el pas cap al disseny de classes i s'analitzarà i estudiaran el comportament que ha de tenir el sistema a construir.

**Capítol 7** – Per últim, un cop hem parlat de la especificació i del disseny del sistema en aquest capítol s'explica la implementació del mateix. Analitzarem quines són les tecnologies que s'han fet servir en cadascuna de les parts del sistema, des de la interfície gràfica passant per el domini del sistema fins a la base de dades que dona suport a la persistència.

**Capítol 8** – En aquest últim capítol on ja hem desglossat tot el sistema parlarem de les desviacions respecte a la planificació inicial, dels costos vinculats al desenvolupament del projecte i de quins han estat els objectius assolits i la valoració personal.

## Descripció del PFC

Ondema és un projecte desenvolupat íntegrament per l'alumne Josep Lagunas Calpe estudiant d'enginyeria tècnica informàtica de Gestió. Es tracta d'un projecte empresarial desenvolupat per a l'empresa CIMNE.

El principal objectiu és aconseguir un sistema de gestió de projectes enfocat al món de l'enginyeria. En aquest tipus de projectes, a diferència d'altres, cadascuna de les tasques persegueix com a objectiu final l'obtenció d'una peça o prototipus que forma part d'un resultat global (el projecte complet).

Aquestes tasques en les que finalment cal obtenir un "element resultant" són assignades a grups d'experts (un usuari tipus del nostre sistema) que, coordinats per un dels seus membres i de manera jeràrquica, han d'obtenir els resultats dins d'un termini temporals que són inicialment fixats.

A més a més, per a cadascuna de les tasques assignades i resultats obtinguts s'hi assigna un "validador" ( un altre tipus d'usuari del sistema ). La tasca d'aquest és la d'acceptar o rebutjar l'"element" presentat com a resultat de la tasca de manera que si la seva valoració resulta satisfactòria l'execució del projecte segueix el seu curs natural mentre que, de no ser-ho, aquest fet implica un nova iteració en el procés d'obtenció de resultats.

El sistema ha de ser capaç de servir com a eina per a cobrir aquestes necessitats així com d'altres que seran explicades més endavant entre les que destaquen per exemple la gestió documental i el foment de la col·laboració entre els usuaris involucrats en un mateix projecte i/o tasques.

## **Descripció del projecte empresarial**

Inicialment el projecte va ser desenvolupat en exclusiva com a projecte per a una empresa client de CIMNE (Applus) però posteriorment la direcció de CIMNE va optar per a fer un desenvolupament més generalista de manera que el sistema pogués ser venut com a producte comercial.

## **Context de l'empresa**

CIMNE és un centre de recerca dedicat als mètodes numèrics i la simulació basada en els elements finits però des de ja fa uns anys i, de manera gradual, ha anat potenciant un departament de tecnologies de la informació que, a dia d'avui, ha vist ampliada la seva tasca inicial d'oferir suports als diferents projectes en que CIMNE participava (la majoria relacionats amb els mètodes numèrics) a l'execució de projectes propis (dins de l'àmbit de les TIC) alguns dels quals han esdevingut productes comercials que s'han convertit en una important font d'ingressos per a l'empresa.

## **Context del client**

Tal i com anteriorment s'ha esmentat, inicialment aquest projecte va néixer com a petició d'un important client de CIMNE, l'empresa de certificació Applus.

CIMNE treballa des de fa temps en col·laboració amb Applus, en particular amb la seva divisió d'aeronàutica. Applus, a la seva vegada, treballa amb importants clients (amb qui CIMNE també col·labora) com en aquest cas Airbus certificant les peces que formen part dels seus avions (des de grans parts com per exemple un ala fins a petits cargols). La funció d'Applus és la de certificar que totes i cadascuna de les parts d'un avió compleixi els requeriments especificats pel fabricant entre d'altres funcions.

## Descripció del producte

Ondema s'ha concebut com un sistema de planificació, suport a l'execució i seguiment d'un projecte d'enginyeria. A més oferta una sèrie de funcionalitats que brinden suport al treball en comunitat, la gestió documental i a la traçabilitat del projecte.

Donat que un dels requeriments era que el sistema havia de ser fàcilment accessible es va decidir implementar una interfície d'accés web de manera que únicament fos necessari un navegador per tal de poder començar a treballar-hi. De tota manera, no és pot descartar anar oferint amb el temps nous tipus d'interfícies com són l'accés mitjançant una aplicació d'escriptori o via mòbil des de qualsevol de les actuals plataformes que hi ha al mercat.

Des de la seva vessant de gestor de projectes el sistema permet a l'usuari la creació d'un projecte des de zero. L'usuari pot definir tantes unitats de treball com s'estimi necessari (workpackages) i dins d'aquestes tantes tasques i subtasques com la planificació del projecte requereixi. Per a cadascuna de les tasques, l'usuari encarregat de la creació del projecte assignarà recursos (persones) encarregades de treballar en equip per assolir l'objectiu i, de la mateixa manera, n'indicarà quin d'aquest exercirà el rol de coordinador de la tasca. Aquest coordinador tindrà la capacitat de gestionar el seu equip com ell consideri necessari i serà l'encarregat d'interaccionar amb els coordinadors d'altres tasques i/o workpackage al qual pertany. En el cas de coordinació de workpackage, la interacció serà tant amb els coordinadors de les tasques que formen el wp que ell coordina com amb el cap de tot el projecte.

Per a dur a terme aquestes interaccions i, cobrir les funcionalitats requerides per permetre i facilitar la comunicació entre els membres de la plataforma, el sistema ofereix 2 grans canals de comunicació que permeten als usuaris estar connectats entre ells, el servei de correu intern i el servei de missatgeria instantània.

A més a més, el sistema permet que l'usuari configuri un sistema de notificacions que l'informaran d'aquelles activitats a les que decideixi subscriure's, per exemple, un usuari validador d'una tasca, podria rebre la notificació de manera automàtica per part del sistema de que el document o tasca que ha de ser validat ja es troba disponible per ser avaluat.

En resum, el principal objectiu del sistema és facilitar i agilitzar els processos que acompanyen l'execució de projectes d'aquests tipus i dotar als equips de treball de les eines necessàries en cada cas per dur a terme les seves tasques des d'un entorn globalment accessible via navegador web però tractant d'obtenir l'agilitat pròpia de les aplicacions d'escriptori.

## Objectius personals i professionals

Aquest projecte és l'últim pas per concloure els estudis que m'han de portar a obtenir el títol d'Enginyer tècnic en Informàtica. És un repte motivador, que comporta haver d'assolir una sèrie d'objectius. Aquests són diversos, però els intentaré resumir breument.

Adquirir nous coneixements. Aquest projecte ha de servir per obrir-me pas en un conjunt de noves tecnologies fins ara no conegudes amb massa detall (o fins i tot gens en algun cas). Per tant caldrà tenir en compte que existirà una corba d'aprenentatge durant el procés que podrà ser valorada a la fi de l'execució del projecte.

Adquirir experiència. Aprendre dels errors comesos, i dels solucionats, per a no cometre'ls en el futur.

Treball dins un context empresarial. Aquest projecte s'ha d'acabar amb la sensació de ser un projecte de qualitat. L'objectiu final de les empreses és guanyar diners, i si aquest projecte no proporciona la qualitat suficient, ni satisfà els objectius proposats, haurà suposat no ser un projecte empresarial vàlid.

## Agraïments

Durant el temps que ha durat aquest projecte he rebut el suport de molta gent i no voldria deixar d'agrair a aquestes persones la seva paciència i comprensió.

Així voldria agrair tot el suport incondicional de tota la meva família i, molt especialment, als meus pares Carlos i Ana Maria, i germans Xavier i Carles i a la meva parella l'Ana per haver estat sempre al peu del canó.

De la mateixa manera voldria agrair a tots el meus companys de CIMNE-TIC els bons moments passats durant tot el temps que he estat amb ells així com a la meva directora de projecte Ruth Raventós pel seu ajut i comprensió en moments difícils per mi.

En definitiva moltes gràcies a tots, sense vosaltres probablement aquest projecte mai hauria acabat de sortir endavant.



---

## **Capítol 2**

---

### **Planificació inicial**

Primer de tot cal fer una aproximació a la manera com s'estructurà la feina que cal dur a terme per posteriorment tractar de quantificar els temps que seran necessaris per dur a terme cada tasca identificada.

### **Fases del projecte**

#### ***Contextualització***

Com a primer pas cal tenir una idea global de que es el que es vol per part de client. Per aquest motiu caldrà fixar un seguit de reunions per contextualitzar la nostra feina. Aquestes reunions ens serviran per entendre de manera general que es el que el client està demandant i quines són les necessitats que hem de ser capaços de cobrir.

#### ***Anàlisi i captura de requeriments***

Seguidament, un cop s'ha fixat un marc de treball, cal aprofundir i concretar aquestes idees generals que ens han aportat. Per dur a terme aquesta tasca, es fixaran un seguit de reunions amb diferents tipus d'usuaris que tindrà el sistema que cal desenvolupar per tal de ser capaços de fer una captura de requeriments. Aquest hauran de ser tant a nivell funcional (que és el que cal que el programa sigui capaç de dur a terme) com no-funcionals (per exemple si l'accés haurà de ser a través del navegador, temes de seguretat, etc).

#### ***Especificació del sistema***

Un cop feta la captura de requeriments es procedirà a fer l'anàlisi del sistema per obtenir una especificació complerta. Caldrà generar tota la documentació necessària tal com diagrama del model conceptual, redacció dels casos d'ús, etc.

## ***Disseny del sistema***

La següent fase consistirà a dur a terme el disseny complet de l'aplicació a partir de l'especificació obtinguda i anar refinant-la. En aquesta fase haurem de normalitzar el model conceptual per obtenir el diagrama de classes que apareixeran al nostre sistema. A més, generarem els diagrames de seqüència de les principals operacions a implementar de manera.

El disseny de la base de dades serà una altra de les tasques a desenvolupar en aquesta fase. Identificarem de quina manera cal fer persistents les dades i obtindrem l'esquema de la base de dades que sustentará l'aplicació.

## ***Implementació del sistema***

Per últim, un cop completades les etapes anteriors, implementarem el sistema dissenyat mitjançant les eines que hagin estat seleccionades a tal efecte. En aquesta fase cal tenir present que és probable que degut al fet de no conèixer totes les tecnologies que es faran servir haurà de existir un període de formació que ens permeti dominar la tecnologia per poder-la aplicar de manera correcta i eficient a l'hora d'implementar.

Un cop implementat el sistema serà necessari un període de validació del mateix per tal de detectar possibles errors i corregir-los de cara a l'entrega del sistema.

## Estimacions temporals inicials

Així doncs, podríem resumir en la següent taula la feina a fer i assignar una estimació a cadascuna de les tasques identificades.

Captura de requeriments	1 mes	Memòria i documentació
Reunions setmanals amb el client	1 mes	
Especificació del sistema	1 mes	
Identificació i definició del model conceptual	1 mes	
Definició dels casos d'ús	3 setmanes	
Disseny del sistema	2 mesos	
Normalització i ajustaments del model conceptual	1 setmana	
Disseny de la base de dades	2 setmanes	
Definició de les vistes	1 mes	
Diagrames de seqüència de les principal operacions	1 mes	
Implementació	2 mesos	
I+D tecnologies	3 setmanes	
Implementació del sistema	2 mesos	
<b>Estimació Total de temps</b>	<b>Aprox. 6 mesos</b>	

Taula 1

Un cop assignem les nostres estimacions sobre cadascuna de les fases per les que el desenvolupament del nostre projecte ha de passar estimem que necessitarem un 6 mesos per poder tenir una primera versió.

---

## Capítol 3

---

### Estat de l'art

En el món de la gestió empresarial el software més estès és Microsoft Project. És tracta d'un sistema per a la gestió integral de projectes en totes les seves vessants.

Microsoft Project representa una aposta segura i molt fiable per aquelles empreses disposades a pagar per tenir un producte com aquest. És un sistema desenvolupat per l'empresa Microsoft destinat a permetre la gestió de tot tipus de projectes independentment de la envergadura que tinguin.

Per altra banda i, més enllà de les possibilitats d'una gran empresa com Microsoft, existeixen un gran nombre de productes freeware que permeten dur a terme la gestió d'un projecte de manera similar però amb força més limitacions. Algunes d'aquestes propostes (la gran majoria) es presenten com aplicacions d'escriptori i d'altres com a aplicació web, disponible online sense la necessitat d'instal·lar cap software.

En aquest últim grup és on ens centrarem. La majoria de sistemes estan enfocats a la gestió integral d'un projecte i no tant al seguiment de les entregues. Aquí és on el sistema a desenvolupar se separa de la resta de sistemes. La principal singularitat d'aquest sistema ha de ser la de permetre fer un seguiment exhaustiu del cicle de vida d'un producte final que s'ha d'obtenir com a resultat de la realització d'una determinada tasca.

Per fer aquest seguiment és necessari incorporar-hi eines pròpies dels sistemes online d'avui dia (com el xat o el correu). D'aquesta manera es pretén potenciar l'existència de fluxos d'informació entre els actors implicats que permetin una col·laboració i un feedback constants.

Així doncs, el sistema a desenvolupar s'ha de centrar més en la consecució final de documents entregables resultants de les diferents fases o etapes d'un projecte que no pas en l'assoliment dels terminis a acomplir (tot i que sense perdre de vista els aspectes relacionats amb aquest últim punt).

Això, juntament amb la faceta col·laboracional de la que es pretén dotar al sistema hauran de ser els seus trets diferencials respecte d'altres productes existents al mercat.

---

## Capítol 4

---

### Anàlisi de requeriments

La captura de requeriments funcionals ha estat duta a terme mitjançant reunions entre el desenvolupador del sistema i un grup de membres de l'empresa client. Durant el transcurs d'aquestes reunions s'ha arribat a un conjunt de requeriments a partir dels quals, posteriorment, han estat enumerats i descrits els casos d'ús.

Del total de requeriments extret d'aquestes reunions s'han separat ens requeriments funcional i no funcionals i que a continuació s'enumeren i descriuen.

#### **Requeriments no funcionals**

- *Accés web:*

L'accés al sistema haurà de fer-se a través d'un navegador web estàndard sense necessitat d'instal·lar cap software a excepció de possible plugins necessaris per al correcte funcionament del propi navegador.
- *Disponibilitat del sistema:*

El sistema desenvolupat ha d'assegurar la disponibilitat i l'accés al mateix i a la informació continguda en ell. Al tractar-se d'un sistema basat en l'accés web aquest requeriment és bàsic.
- *Certificació del sistema:*

El sistema conté informació sensible de manera que les comunicacions han de ser encriptades mitjançant qualsevol dels sistemes disponibles al mercat.
- *Documentació:*

Es fa entrega al client d'un manual d'usuari del sistema per tal de facilitar-ne l'ús i minimitzar la corba d'aprenentatge dels usuaris.
- *Eficiència:*

Tot i tractar-se d'un sistema web, es pretén dotar a l'usuari d'un sistema de resposta ràpida i fiable. Per això, s'ha d'assegurar una experiència d'usuari el més propera a una aplicació d'escriptori tal i com fan la majoria de sistemes actuals.

- *Extensibilitat:*

El sistema ha d'estar completament obert a noves extensions que siguin requerides en un futur sense que aquest fet comprometi els desenvolupaments duts a terme fins aquell moment.
- *Compliment de la llei de LOPD:*

Donat que el sistema ha d'incorporar un potent sistema de comunicació entre els membres de la seva comunitat d'usuaris es important assegurar el compliment de la legislació vigent seguint acuradament la llei de LOPD i informant a tots els usuaris de possibles situacions que puguin comprometre la intimitat de les dades a les que hi tinguin accés.
- *Escalabilitat:*

El disseny del sistema ha d'assegurar la fiabilitat del mateix. Ha d'estar preparat per a progressivament adequar-se a un volum d'usuaris que es preveu creixent.
- *Seguretat:*

Donada la naturalitat dels documents que contindrà el sistema es requereix que l'accés a la informació del sistema ha d'estar completament limitat a aquells usuaris que hi tinguin obert l'accés.
- *Estabilitat:*

El sistema ha d'estar preparat a possibles errors i ha de donar informació a l'usuari sobre l'error produït i possibles alternatives per a solucionar el problema.
- *Fomentar la col·laboració:*

El sistema ha de facilitar la col·laboració entre els seus usuaris de manera que es hi hagi un flux constant de la informació i canvis que vagin patint els projectes en els que un usuari hi participa. Així doncs caldrà desenvolupar les eines necessàries per a permetre que un usuari de la plataforma es comuniqui amb els altres.

## Requeriments funcionals

- *Creació de projectes:*

El sistema ha de permetre a un usuari crear una estructura de projecte definida per un conjunt d'unitats que seran anomenades workpackages. Dins de cadascuna d'aquestes unitats s'hi podran definir tantes tasques com calgui i que, a l'hora, podran descompondre's en tantes subtasques com s'estimi necessari.

Cada workpackage, tasca i subtasca (així com un projecte) hauran de tenir en tot moment una data inicial i una data final les qual sempre hauran d'estar dins del rang de dates vàlid definit per la data inicial i la data final de la unitat de la jeràrquicament en són filles exceptuant les dates definides per a un projecte que no estan subjectes a cap restricció.

- *Definició de reunions:*

El sistema ha de permetre definir reunions per a una tasca i/o una subtasca. A l'hora de fixar una reunió caldrà definir-hi un nom o motiu de reunió i una petita descripció, el lloc on serà duta a terme i la data de la reunió (que haurà d'estar dins del rang vàlid de dates que quedi definit per la tasca o subtasca a la que pertany. A més caldrà indicar-hi qui seran les persones assistents per tal que la convocatòria els hi pugui ser comunicada.

- *Repositori documental:*

El sistema ha de permetre gestionar tots els documents d'un usuari de manera senzilla i clara. Tot document del sistema ha d'estar vinculat per una banda al projecte al que pertany, per altra pot o no estar-ho a qualsevol de les unitats que formen part d'un projecte (workpackage, tasca, subtasca o reunió) i, si cal, a una jerarquia de documents a la que pertany.

Un document finalment s'ha de poder compartir amb altres usuaris per tal que aquests hi tinguin accés a ell des de la seva vista del repositori documental o per a cada document carregat per ells al sistema o compartit cap a ells per altres usuaris hi podran veure-hi tota la informació relacionada i descarregar el seu contingut.

- *Servei de missatges:*

El sistema ha de permetre l'intercanvi intern de missatges entre usuaris. Cada usuari disposa d'un accés a la seva bústia de correu intern on rep els missatges dels altres usuaris i on pot contestar i enviar nous missatges. Cada missatge que rep l'usuari dins la plataforma pot ser reenviat en forma de notificació a un compte de correu extern que l'usuari tingui informat a les seves opcions de configuració si hagi ho configura.
- *Servei de missatgeria instantània:*

A part del sistema de missatges convencionals el sistema ha de disposar d'un sistema de missatgeria instantània propi on els usuaris puguin xatejar amb altres usuaris en temps real que estiguin connectats a la plataforma.
- *Sistema d'avaluació de documents:*

El sistema ha de permetre l'avaluació de documents resultants de les tasques que formen part d'una unitat de workpackage per part dels usuaris que hagin estat designats com a usuaris validadors en aquella tasca. Aquest procés haurà de ser iteratiu fins a assolir el document resultant de la tasca l'estat d'acceptat.
- *Visualització del projecte:*

El sistema ha de mostrar de manera senzilla, ràpida i clara com es compona un projecte, qui en són les persones que en formen part i de quina manera i quin és l'estat actual en el que el projecte i cadascuna de les parts que el defineixen es troba.
- *Traçabilitat:*

El sistema ha de permetre a un usuari amb els rols necessaris per fer-ho poder comprovar els canvis que afecten a una determinada unitat dins d'un projecte o al propi projecte. Per exemple si una tasca pateix un canvi en una data de les que la defineixen el sistema ha d'actualitzar aquest canvi però sense perdre informació històrica de manera que sigui possible visionar l'evolució de tot canvi que es produeixi en l'estructura d'un projecte en el temps així com la manera com aquests es propaguen a terceres unitats que es puguin veure afectades.



---

## Capítol 5

---

### Especificació del sistema

Un cop duta a terme la captura dels requeriments del sistema el següent pas natural és el de transformar els requeriments funcionals en casos d'ús. Els casos d'ús són bàsicament requisits funcionals que indicaran quines són les coses que farà el sistema.

Per dir-ho d'un altre manera, els casos d'ús són, en principi, documents de text, no diagrames. Però donat que farem servir el llenguatge notacional conegut com a UML en aquest sí que els casos d'ús van acompanyats d'un diagrama on queden il·lustrats els noms dels casos d'ús i els actors que hi participen així com les seves relacions.

Però abans d'introduir-nos en profunditat en l'especificació del sistema veurem una mica més en profunditat en que consisteix el llenguatge UML que és el que farem servir per descriure el sistema que cal construir.

## **El llenguatge UML**

UML o (Unified Model Language) és un llenguatge notacional estandarditzat. Consisteix en un llenguatge gràfic per a construir, documentar, visualitzar i especificar un sistema de software.

L'any 2005 el llenguatge va passar a ser un estàndard aprovat per la ISO com ISO/IEC 19501:2005 Information technology – Open Distributed Processing – Unified Modeling Language (UML) en la seva versió 1.4.2

UML ofereix les eines i la potència necessàries per a poder crear un model complet del sistema a construir. Gràcies a aquestes eines que enumerarem més endavant podem modelar els processos de negoci, les funcions i els esquemes de la base de dades i fins i tot expressions pròpies dels llenguatges de programació.

Per a poder dur a terme totes aquestes tasques l'UML està proveït de diferents tipus de diagrames. A la versió 2.0 del llenguatge es disposa de 13 tipus diferents de diagrames els quals es poden agrupar en 3 categories diferents segons la naturalesa de la seva funcionalitat final:

- **Diagrames d'estructura**

Els diagrames d'estructura pretenen definir detalls propis de la implementació. Amb ells es descriu com es desplega un sistema o com els seus components es modularitzen buscant obtenir un sistema que encapsuli la implementació i sigui reemplaçable i desplegable fàcilment.

- **Diagrames de comportament**

La idea bàsica dels diagrames de comportament és la d'aconseguir definir què és el que ha de fer el sistema però sense arribar a explicar com ho ha de fer.

- **Diagrames d'interacció**

Els diagrames d'interacció busquen plasmar de manera clara com els objectes interaccionen els uns amb els altres a través de missatges. En aquests tipus de diagrames es expressa la manera com el sistema "fa" el que "ha de fer". És en aquests punts del disseny d'un sistema on poden

aplicar-se patrons que en facilitin la seva construcció. Els patrons són guies i principis estructurats que ajuden a resoldre problemes comuns de manera eficaç i provada.

## Diagrames d'estructura

- *Diagrama d'objectes:*

Defineix el model de domini d'objectes en el món real del sistema que s'està estudiant. En ell s'il·lustren els conceptes o objectes del món real que en formen part. S'identifiquen els objectes que apareixen en el domini, els atributs de cadascun d'ells i la manera com es relacionen entre ells.

- *Diagrama de classes:*

Defineix una vista estàtica del model de classes. Tot i la seva semblança amb el model de domini que és representat pel diagrama d'objectes, en aquest diagrama el que s'està mostrant ja no són conceptes del món real sinó classes software. Cadascuna de les classes software que s'hi identifiquen van acompanyades del que es coneix com atributs de classe i de les operacions que definiran els seu comportament i que s'identifiquen en els diagrames d'interacció del sistema, del que se'n parlarà més endavant.

- *Diagrama de components:*

Els diagrames de components serveixen per a modelar un vista estàtica del sistema. Mostra la organització i les dependències entre un conjunt d'elements que en formen part. Aquest tipus de diagrames serveix per a mostrar la organització i les dependències entre el conjunt d'elements. Un dels usos principals és que pot servir per veure quins components poden ser compartits entre sistemes o entre diferents parts d'un sistema.

- *Diagrama d'estructura composta (introduït a la versió 2.0):*

Un diagrama d'estructura composta és un diagrama que mostra l'estructura interna d'una classe, incloent-hi els seus punts d'interacció amb altres parts del sistema. Això mostra la configuració i la relació de les parts que juntes duen a terme el comportament. Els elements de classe han estat descrits en gran detall pels diagrames de classe. Aquests diagrames descriuen en gran detall la manera en que les classes poden ser mostrades com a elements exposant interfícies i contenint ports i parts.

- *Diagrama de desplegament:*

Els diagrames de components són utilitzats amb una finalitat ben clara, la de modelar els components hardware del sistema i la manera com es relacionen entre ells.

- *Diagrama de paquets:*

Els diagrames de paquets són utilitzats per reflectir l'organització dels diferents elements del sistema. Els diagrames de paquets dels elements de classe s'acostumen a fer servir per proveir una visualització dels espais de noms del sistema. Els usos més comuns per aquest tipus de diagrames són per a la organització dels diagrames dels casos d'ús i dels diagrames de classe tot i que cal fer notar que la seva utilització no està limitada a aquest elements UML.

## **Diagrames de comportament**

- *Diagrama d'activitats:*

El diagrama d'activitats és una variant especial dels diagrames d'estat amb la diferència de que incorpora i modela una seqüència d'accions i condicions que són preses dins d'un procés. En altres paraules la finalitat d'un diagrama d'estats és la de definir un procés de flux de treball i modelar les operacions proporcionades per un objecte per tal d'assolir el flux definit

- *Diagrama de casos d'ús:*

Cal remarcar que tot i que el llenguatge UML aporta un tipus de diagrama específic per als casos d'ús aquest ha de ser entès com una manera d'il·lustrar els noms dels casos d'ús i dels actors i les relacions existents entre ells. Els diagrames d'ús i les relacions que s'estableixen entre ells han de ser entesos com a secundaris a l'hora de treballar els casos d'ús en sí. Els casos d'ús han de ser un document de text on quedin perfectament recollits totes les interaccions possibles entre el sistema i els actors que en ell hi participen. Així doncs, cal entendre aquests tipus de diagrames com una manera d'obtenir una visió ràpida del parell actor-objectiu.

- *Diagrama d'estats:*

El diagrama d'estats tracta de representar aquells esdeveniments i estats que tenen una especial rellevància d'un objecte i el comportament d'un objecte com a reacció a un esdeveniment. En el diagrama d'estats les transicions son representades amb fletxes que estan etiquetades amb els esdeveniments que provoquen aquesta transició. Un diagrama d'estat serveix per a mostrar el cicle de vida d'un objecte. No és necessari reflectir tots els possibles esdeveniments, així doncs, es important remarcar que el nivell de detall d'un diagrama d'estats es fixa de manera totalment arbitrària.

## Diagrames d'interacció

- *Diagrama de seqüència:*

Els diagrames de seqüència es troben serveixen per a il·lustrar el disseny de la interacció entre objectes software per a completar un objectiu. En altres paraules, és una manera de veure gràficament com es componen les operacions definides dins d'una classe software. Tot i això, i encara que no estan definits a nivell de "sistema" en el llenguatge UML, els diagrames de seqüència també són empleats per a per mostrar un escenari específic d'un cas d'ús. Així, un cas d'ús pot ser representat com un conjunt d'esdeveniments entre un actor i el sistema que és vist com un tot, com una mena de caixa negra. Aquest esdeveniments que se succeeixen de manera seqüencial poden estar parametrizats.

- *Diagrama de comunicació:*

Aquests tipus de diagrames neixen d'una combinació entre els diagrames de classes, els diagrames de seqüència i els diagrames de casos d'ús. És l'encarregat de modelar les interaccions entre objectes i ho fa valent-se del missatges de seqüència. És per aquest últim motiu que la informació aportada entre aquests dos tipus de diagrames (comunicació i seqüència) és força similar i pot resultar relativament senzill transformar-ne un en un altre.

Per tal de mantenir l'ordre entre els missatges aquests són etiquetats cronològicament. D'aquesta manera el primer missatge serà l'etiquetatge amb el número 1 i així successivament.

- *Diagrama de temps (introduït a la versió 2.0):*

Aquest tipus de diagrames representen de manera especial la interacció entre els objectes del sistema. Ho fan centrant-se en el temps dels missatges enviats. La seva funcionalitat es la de detallar restriccions de manera detallada sobre el temps, o per tal de mostrar els canvis amb línies de vida respecte al temps. Són molt útils i empleats en sistemes en temps real.
- *Diagrama de vista de la interacció (introduït a la versió 2.0):*

Aquest tipus de diagrames mostren un vista de les interaccions del sistema. Tot i això estan força diferenciats del altres diagrames de comportament com el diagrama de seqüència i el de comunicacions. Alguns dels components empleats en aquest tipus de diagrames també els podem trobar en els diagrames d'activitats. En aquest tipus de diagrames podem trobar la definició d'un workflow on a dins dels nodes d'estat s'hi defineixin diagrames de seqüència que si bé poden estar definits directament en el node, també poden ser referències cap a altres diagrames de seqüència que estiguin modelats a un altre lloc.

Tot i oferir aquests tretze artefactes, no és necessari desenvolupar tots ells a l'hora de desenvolupar un sistema d'informació.

En el nostre cas en particular, desenvoluparem tan sols 4 d'aquests artefactes.

Primer procedirem a realitzar el diagrama d'objectes on quedarà representat el model de domini del sistema. En aquest diagrama hi quedaran reflectits tots els objectes del món real que participaran i la manera com estan relacionats entre ells.

Els diagrames d'objectes tenen però certes restriccions, sovint no és possible representar al 100% un model extret del món real de manera únicament gràfica. Per aquest motiu per representar aquest tipus de restriccions farem ús d'un altre llenguatge conegut com OCL (Object Constraint Language). Aquest llenguatge permet precisar de manera textual i, valent-se d'expressions pseudo-matemàtiques, les restriccions que gràficament no són representables eliminant així possibles errors d'interpretació, ambigüitats i regles invariants que el sistema a modelar requereix que siguin complertes.

Posteriorment, un cop identificats els requeriments capturats durant les entrevistes amb els diferents tipus d'usuaris que faran ús del sistema, redactarem els casos d'ús per tal de generar el diagrama resultant on pas a pas seran descrits de manera seqüencial els esdeveniments que componen cadascun d'ells acompanyats d'un diagrama de seqüència del sistema.

Val a dir, que els passos o etapes descrites anteriorment, acostumen a repetir-se iterativament per tal de refinar a cada volta l'especificació del sistema que s'acabarà construir.

Un cop obtinguts els diagrames corresponents a cada pas el resultat final serà una especificació que ens demarcarà amb precisió el context del que el sistema ha de fer, la manera com haurà de comportar-se. No serà fins a la següent etapa en que ens centrarem a com "solucionar" el plantejat a l'especificació, és a dir, a dissenyar les solucions que han de fer el que el sistema es comporti tal i com és desitjat.

Així doncs, un cop enumerats els passos que es duran a terme en l'etapa d'especificació, tot seguit veiem la seva aplicació al nostre cas d'estudi.

Com a primer pas, procedim a especificar el sistema. S'han identificat un elevat nombre d'objectes que apareixeran en el model de domini i s'han establert les relacions que entre ells existeixen fins a obtenir un diagrama d'objectes acurat tal i com s'aprecia a la figura 1. Tanmateix s'han identificat un conjunt de restriccions d'integritat textuals que han estat expressades en llenguatge OCL per acabar de completar el diagrama del model de domini.

Cal recordar que es tracta d'un procés iteratiu de manera que el diagrama presentat és el resultat final d'un model inicial el qual que s'ha anat refinant i corregint a cada iteració realitzada.





## Definició de les classes del model

### Rols d'usuari

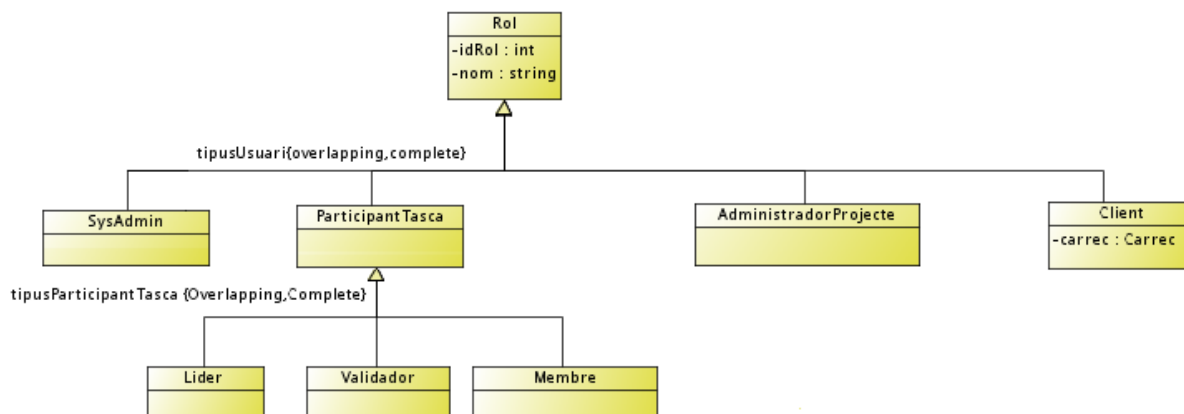


figura 2

### Descripció:

La jerarquia de classes nascuda a partir de la classe Rol engloba tots els possibles tipus d'usuaris que poden donar-se al sistema. Tot usuari té assignat un d'aquests rols poden així executar una sèrie de funcionalitats. Només en el cas del rol d'administrador de sistema (SysAdmin) aquest pot ser compartit amb un altre rol per un mateix usuari per a tasques concretes de manteniment del sistema.

### Atributs:

*Classe Rol:*

**idRol** : identifica un rol de manera única al sistema.

**nom**: El nom assignat a aquell rol.

*Classe Client (inclou els atributs de Rol):*

**carrec**: El càrrec del client dins la seva empresa.

## Persones i Usuaris

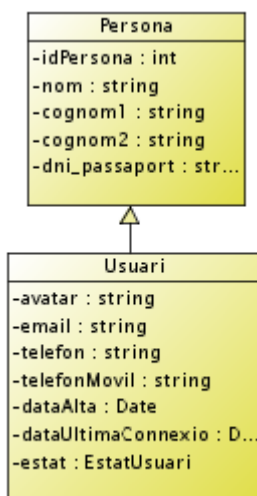


figura 3

**Descripció:**

La jerarquia de classes nascuda a partir de la classe **Persona** engloba tota la informació relacionada amb dades personals d'interès pel sistema. Per una banda es registra informació relacionada amb persones que no necessàriament han de ser usuaris, però que poden participar en un moment donat en algun dels projectes i que, per tant, cal tenir controlats. Per altra banda, tenim la informació que queda registrada d'aquelles persones que interactuen amb el sistema, els usuaris. La classe **Usuari** conté tota la informació necessària per a poder contactar amb un usuari i per a conèixer el seu estat.

**Atributs:**

*Classe Persona:*

**IdPersona:** clau externa que identifica de manera inequívoca una persona dins del sistema.

**nom:** el nom de la persona.

**cognom1:** El primer cognom de la persona.

**cognom2:** El segon cognom de la persona.

**DNI\_passaport:** el DNI o passaport de la persona.

*Classe Usuari (inclou els atributs de Persona):*

**avatar:** Imatge que representa a la persona al sistema.

**email:** Adreça de correu electrònic de l'usuari.

**telefon:** Número de telèfon fix on localitzar a l'usuari.

**telefonMovel:** Número de telèfon mòbil on localitzar l'usuari.

**dataAlta:** Data en que l'usuari va ser donat d'alta al sistema.

**dataUltimaConnexio:** Data en que l'usuari es va connectar al sistema per darrer cop.

**estat:** Informa sobre si actualment l'usuari està actiu al sistema o bé no hi té accés habilitat.

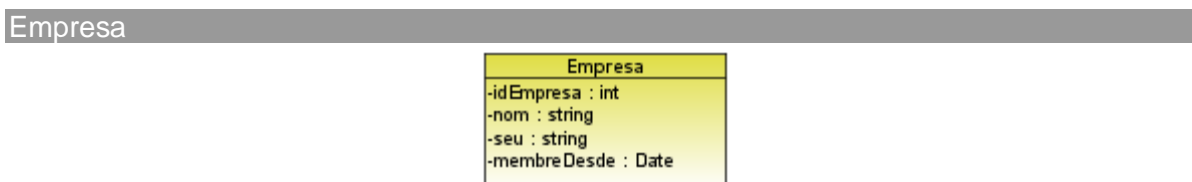


figura 4

### **Descripció:**

La classe Empresa representa a l'empresa client, l'empresa per a la qual s'està duent a terme el projecte que s'està gestionant, una empresa pot tenir molts projectes contractats però no necessàriament ha de tenir la mateixa persona de contacte per a monitoritzar tots els projectes.

### **Atributs:**

*Classe Empresa:*

**idEmpresa:** clau externa que identifica de manera inequívoca una empresa dins del sistema.

**nom:** El nom comercial de l'empresa.

**seu:** Localització de la seu central de l'empresa.

**membreDesde:** Data en que per primer cop es va dur a terme un projecte amb l'empresa com a client.

Projecte

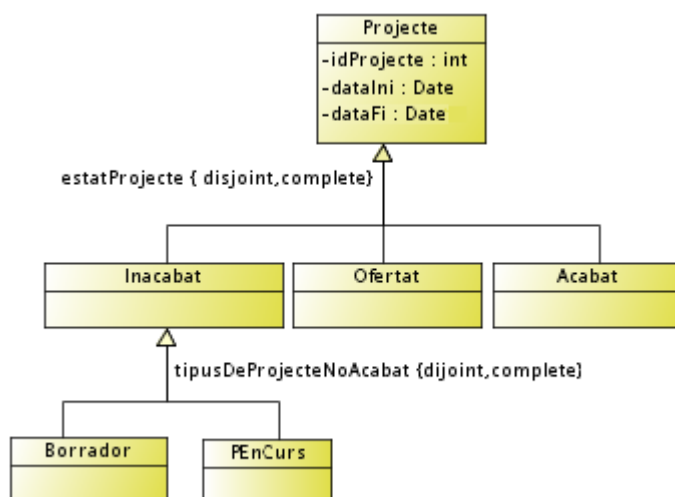


figura 5

**Descripció:**

La jerarquia de classes nascuda a partir de la classe Projecte engloba tota la informació bàsica relacionada amb un projecte creat al sistema. La resta de classes de la jerarquia representen els possibles estats en que un projecte pot trobar-se durant el seu cicle de vida per els quals passa seguint el següent diagrama d'estats:

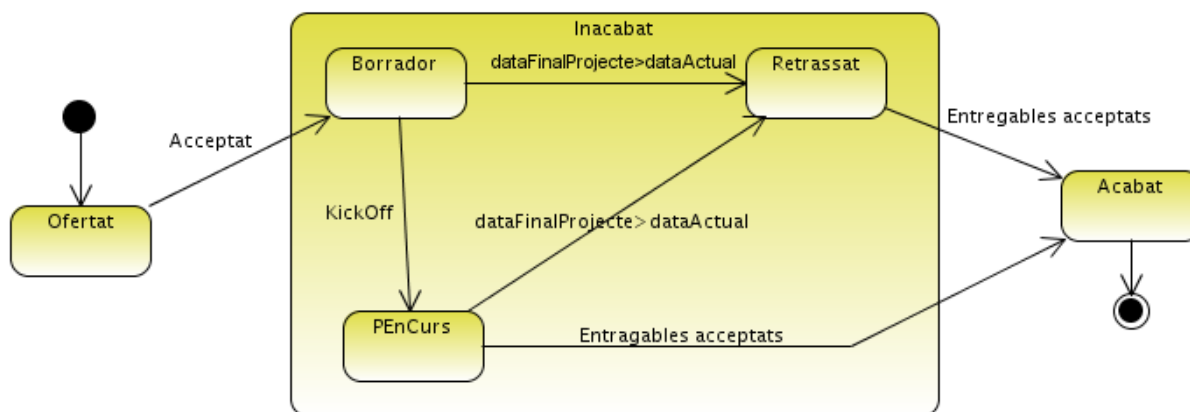


figura 6

Com es pot apreciar en el diagrama superior, l'estat inicial d'un projecte és el d'Ofertat. Un cop s'ha acceptat la proposta de projecte per part de l'empresa client, el projecte passa a un estat de Esborrany en que durant un període de temps es succeeixen les reunions entre les dues empreses. Després de que es dugui a terme la reunió de KickOff el projecte passa a ser actiu i el seu estat s'actualitza a PEnCurs. L'estat de

projecte Endarrerit només s’adquireix quan la data Final del projecte supera a la data actual del sistema, moment en que automàticament el sistema actualitza l’estat del projecte a Endarrerit. Per últim, quan tots els entregables que constitueixen un projecte han estat avaluats i acceptats, el projecte passa a Acabat i s’uneix al repositori de projectes finalitzats per poder-lo consultar.

**Atributs:**

*Classe Projecte:*

- idProjecte:** clau externa que identifica de manera inequívoca un projecte dins del sistema.
- dataIni:** data en la que oficialment s’inicia el projecte.
- dataFi:** data en la que oficialment s’ha d’haver acabat el projecte.

Document

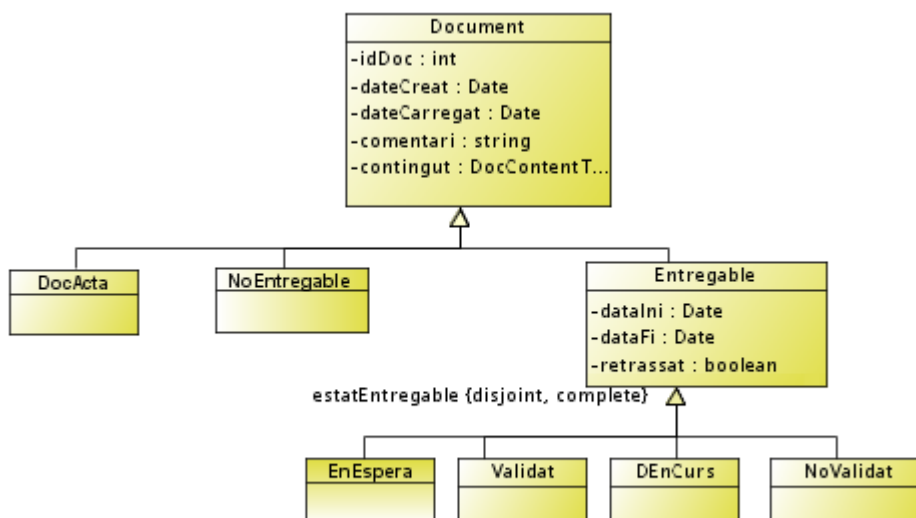


figura 7

**Descripció:**

La jerarquia d’objectes nascuda a partir de Document engloba tota la informació bàsica relacionada amb els documents que es generen durant la vida d’un projecte. Cadascun dels elements de la jerarquia representa un dels possibles tipus de document. Així doncs, Document aporta tota la informació descriptiva del fitxer (creació, càrrega al sistema, comentaris i tipus del contingut). La resta d’objectes serveixen per a definir si es

tracta d'un Acta de reunió , o bé si es tracte d'un document entregable o no. En el cas dels documents entregables, s'inclou nova informació, definida posteriorment.

A continuació es presenta el diagrama d'estats que representa el cicle de vida d'un document entregable:

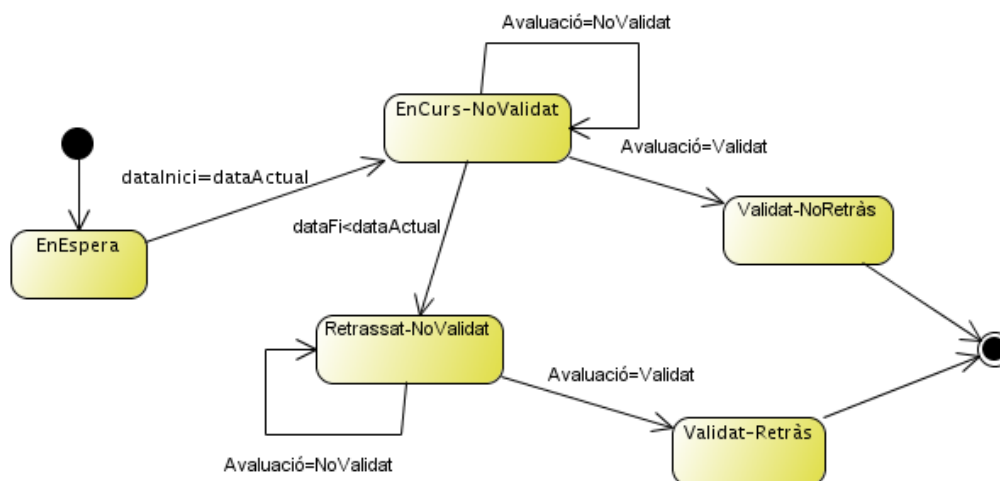


figura 8

Com es pot apreciar en el diagrama superior, l'estat inicial d'un entregable és EnEspera (quan encara no ha arribat la seva data d'inici). Un cop arriba la data d'inici que te assignada, l'estat del entregable passa a ser EnCurs però encara no té cap avaluació assignada. En el moment en que la data del sistema supera la data final que tenia l'entregable, si aquesta encara no té una avaluació vàlida, el sistema el marca automàticament com a endarrerit i ja no podrà canviar a un altre estat que no sigui el de avaluació superada (estat final) però amb la marca de Endarrerit per haver-ho fet fora de temps. Altrament, si durant l'interval entre la data inicial i la final l'entregable completa les seves validacions passa a obtenir l'estat de Validat en el seu període.

#### Atributs:

##### *Classe Document:*

**idDoc:** clau externa que identifica de manera inequívoca un document dins del sistema.

**dateCreat:** data de creació del document.

**dateCarregat:** data de càrrega del document al sistema.

**comentari:** comentari adjuntat al document per l'usuari que el carrega.

**contingut:** tipus de contingut del document.

*Classe Entregable(inclou els atributs de Document):*

**dataIni:** data planificada per a l'inici de l'entregable.

**dataFi:** data planificada per a obtenir la validació de l'entregable.

**endarrerit:** atribut que indica si l'entregable es troba dins del termini o bé està endarrerit.

## Workpackage

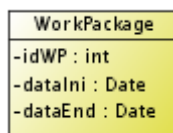


figura 9

### Descripció:

La classe WorkPackage representa una unitat de treball dins del projecte. La seva funció és la d'englobar una sèrie d'entregables associats d'alguna manera entre ells. Un WorkPackage, de la mateixa manera que els documents de tipus entregable o els projectes mateix, té associada una data d'inici i una data de fi. Aquestes, les pot assignar directament el sistema a partir de les dates inicials i finals dels diferents documents entregables que el component o bé permetre que sigui un usuari amb prous privilegis qui els fixi.

### Atributs:

*Classe WorkPackage:*

**idWP:** clau externa que identifica de manera inequívoca un WorkPackage dins del sistema.

**dataIni:** data inicial associada a un Workpackage. Normalment la data inicial més baixa de totes les dates inicials dels entregables que componen un WorkPackage.

**dataFi:** data final associada a un WorkPackage. Normalment la data inicial més baixa de totes les dates inicials dels entregables que componen un WorkPackage.



## Reunió

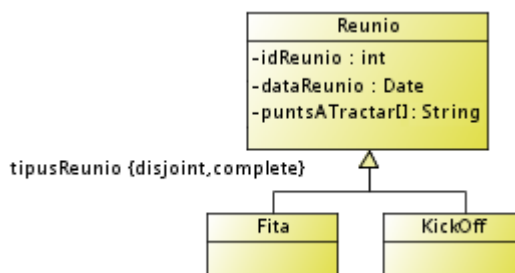


figura 10

### Descripció:

La jerarquia de classes que neix a partir de Reunió engloba informació relacionada amb les trobades dutes a terme durant la durada d'un projecte i el tipus de reunió. A més d'emmagatzemar informació bàsica com data, etc, a través de les seves associacions, s'obté informació sobre els assistents. Cal destacar que tota reunió té associat un document d'acta que és accessible a través de la documentació d'un projecte on es troba tota la informació detallada sobre els temes tractats.

### Atributs:

*Classe Reunió:*

**idReunio:** clau externa que identifica de manera inequívoca una Reunió dins del sistema.

**dataReunio:** data en la qual té lloc la reunió representada per la classe.

**puntsATractor:** llista de temes que són debatuts en el transcurs de la reunió.

## Missatge

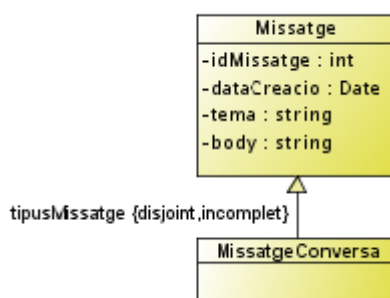


figura 11

**Descripció:**

El sistema, en la seva vessant de plataforma col·laborativa, ha de permetre i facilitar el flux d'informació entre els seus usuaris. Per aquest motiu es fa necessària la incorporació de missatges. Aquests missatges poden ser en temps real, mitjançant l'eina de xat incorporada al sistema o bé diferits en forma de notificació i/o correu electrònic segons l'usuari tingui configurades les seves notificacions.

La resta d'informació associada a un missatge, com l'emissor o el receptor, és accessible a través de les seves associacions.

**Atributs:**

*Classe Missatge:*

**idMissatge:** clau externa que identifica de manera inequívoca un missatge dins del sistema.

**dataCreacio:** la data en que el missatge s'ha creat.

**tema:** El tema sobre el que tracta el contingut del missatge, si en té.

**body:** el contingut mateix del missatge.

## Conversa



figura 12

### Descripció:

La classe Conversa engloba informació sobre una conversa entre una sèrie d'usuaris. La informació bàsica associada a una conversa és la data i hora d'inici i la seva data i hora de fi. A més a través de les seves associacions s'obté quins missatges en formen part i quins usuaris han intercanviat aquest missatges.

### Atributs:

*Classe Conversa:*

**idConversa:** clau externa que identifica de manera inequívoca una conversa dins del sistema.

**Inici:** Instant de temps inicial d'una conversa. Moment en que el primer missatge associat a ella és crea.

**Fi:** instant de temps final d'una conversa. Moment en que l'últim usuari que en pren part, l'abandona.

## Restriccions d'integritat textuais del model de conceptual

1.- Claus externes:

**context Rol inv**

**Rol.allInstances->isUnique(idRol)**

**context Empresa inv**

**Empresa.allInstances->isUnique(idEmpresa)**

**context Document inv**

**Document.allInstances->isUnique(idDoc)**

**context Missatge inv**

**Missatge.allInstances->isUnique(idMissatge)**

**context Conversa inv**

**Conversa.allInstances->isUnique(idConv)**

**context Projecte inv**

**Projecte .allInstances->isUnique(idProjecte)**

**context Plantilla inv**

**Plantilla.allInstances->isUnique(idPlantilla)**

**context WorkPackage inv**

**WorkPackage.allInstances->isUnique(idWP)**

**context Reunio inv**

**Reunio.allInstances->isUnique(idReunioI)**

2.- Un usuari només pot rebre i enviar missatges Conversa a d'altres usuaris associats a la mateixa conversa.

**context Conversa inv**

**self.usuaris.misReb.oclAsType(MissatgeConversa)->forAll(m|m.conv=self)**

**and**

**self.usuaris.misEnv.oclAsType(MissatgeConversa)->forAll(m|m.conv=self).**

3.- Un document no pot generar-se a ell mateix.

**context Document inv**

**self.generat->not exists(idDoc=self.idDoc)**

4.- La data de creació d'un document no pot ser posterior a la data de càrrega al sistema.

**context Document inv**

**self.dataCreated<=self.dataUploaded**

5.- La data de finalització d'un document entregable ha de ser posterior a la seva data de d'inici.

**context Entregable inv**

**self.dataIni<=self.dataFi**

6.- Si la data de finalització d'un entregable és més gran o igual a la data actual del sistema i l'estat de l'entregable és diferent a Validat llavors l'estat de l'entregable ha de ser Endarrerit.

**context Entregable inv**

**self.dataFi>=system.date implies self.retrasat=true**

7.- La data corresponent a la darrera connexió d'un usuari al sistema ha de ser més gran o igual a la data de la seva alta al sistema.

**context Usuari inv**

**self.dataUltimaConnexio>=self.dataAlta**

8.- La data inicial d'un projecte ha de ser anterior o igual a la data inicial de qualsevol dels WP que el formen.

**context Projecte inv**

**Projecte.allInstances->(p|p.wp->forAll(wp|wp.dataIni>=wp.proj.dataIni))**

9.- La data final d'un projecte ha de ser posterior o igual a la data final de qualsevol dels WP que el formen.

**context Projecte inv**

**Projecte.allInstances->(p|p.wp->forAll(wp|wp.dataFi<=wp.proj.dataFi))**

10.- La data inicial d'un entregable ha de ser posterior o igual a la data final de qualsevol dels documents entregables que el precedeixen.

**context Entregable inv**

**self.previ->not exists(dataFi>=self.dataIni)**

11.- La data de càrrega d'un document d'acta de reunió ha de ser posterior o igual a la data en que es celebra la reunió.

**context DocActa inv**

**self.dataCarregat>=self.reunio.dataReunio**

12.- La data d'una reunió ha de ser posterior o igual a la data inicial del projecte al que pertany.

**context Reunio inv**

**self.dataReunio>=proj.dataIni**

13.- La data final d'una reunió ha de ser anterior o igual a la data final del projecte al que pertany.

**context Reunio inv**

**self.dataReunio<=proj.dataFi**

14.- No pot haver dues persones al sistema amb mateixos nom, cognom1, cognom2 i DNI o passaport.

**context Persona inv**

**Persona.allInstances ->isUnique(nom) and Persona.allInstances ->isUnique  
(cognom1) and Persona.allInstances ->isUnique(cognom2) and  
Persona.allInstances ->isUnique(DNI\_passport)**

15.- Un projecte no pot tenir estat "EnCurs" si encara no s'ha celebrat un Reunió de tipus Kick-Off.

**context PenCurs inv**

**self.reunio.oclAsType(KickOff)->size()=1**

16.- L'estat d'un entregable no pot ser "EnCurs" si l'estat del projecte al qual pertany el wp al que està associat té el seu estat com "Esborrany".

**context Esborrany inv**

**self.docs.oclAsType(DEnCurs)->isEmpty()**

## Rols d'usuari

Un cop definit el model de domini, es procedeix a la redacció dels casos d'ús que el sistema haurà de ser capaç de realitzar. Donada la naturalesa del sistema s'han identificat un total de 6 rols d'usuaris o actors diferents els quals són:

- Administrador del sistema o SysAdmin
- Administrador de Projecte
- Coordinador de WorkPackage/Tasca/Subtasca
- Validador de Tasca/Subtasca
- Membre equip
- Client

Per a cada tipus d'actor s'ha estudiat quines són les accions que ha de poder completar dins del sistema i s'han desenvolupat els casos d'ús que corresponents.

## Casos d'ús

Un cop identificat els diferents actors que interactuaran amb el sistema s'inicia la redacció dels casos d'ús del sistema.

A la figura 2 s'hi pot apreciar una vista del conjunt total de casos d'ús identificats relacionats amb els actors o actor que pot realitzar-los. Més endavant veurem en detall cada cas d'ús i de manera detallada el seu contingut on s'indicarà quin és el flux d'execució normal així com les possibles excepcions que pugui haver-hi.

Primerament veiem els diagrames per cada actor amb els casos d'ús que pot executar.



# Diagrama complert

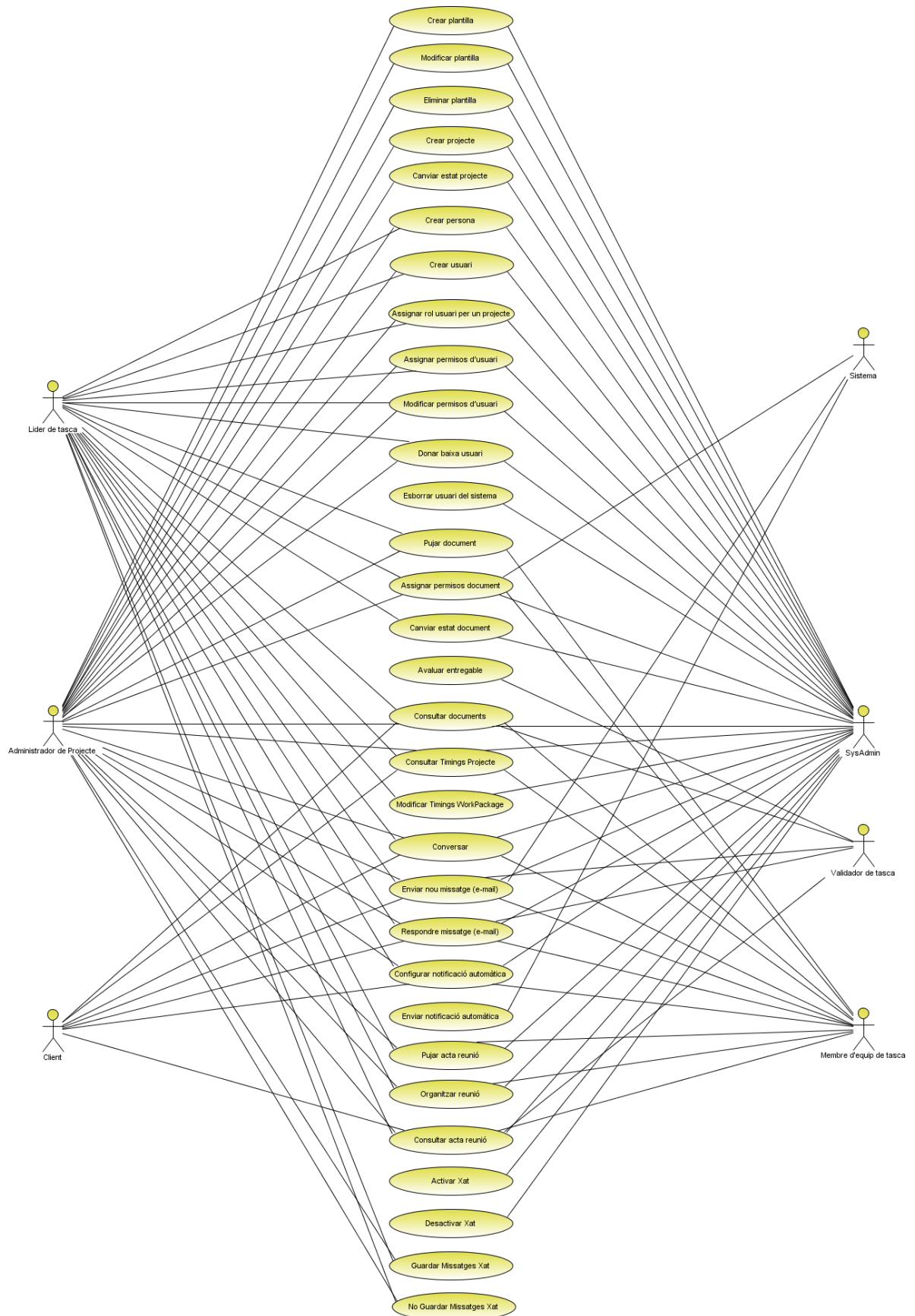


figura 13

## Diagrama client

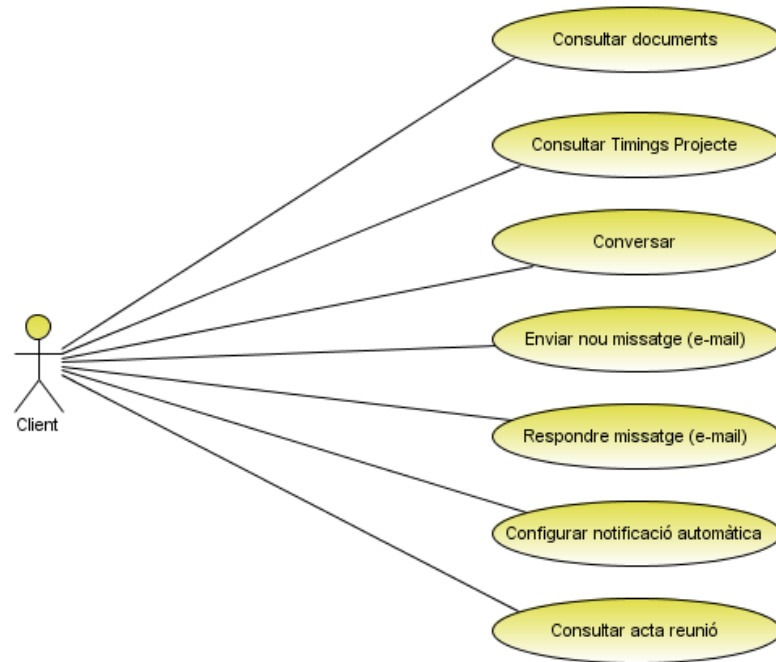


figura 14

### Diagrama Administrador de projecte

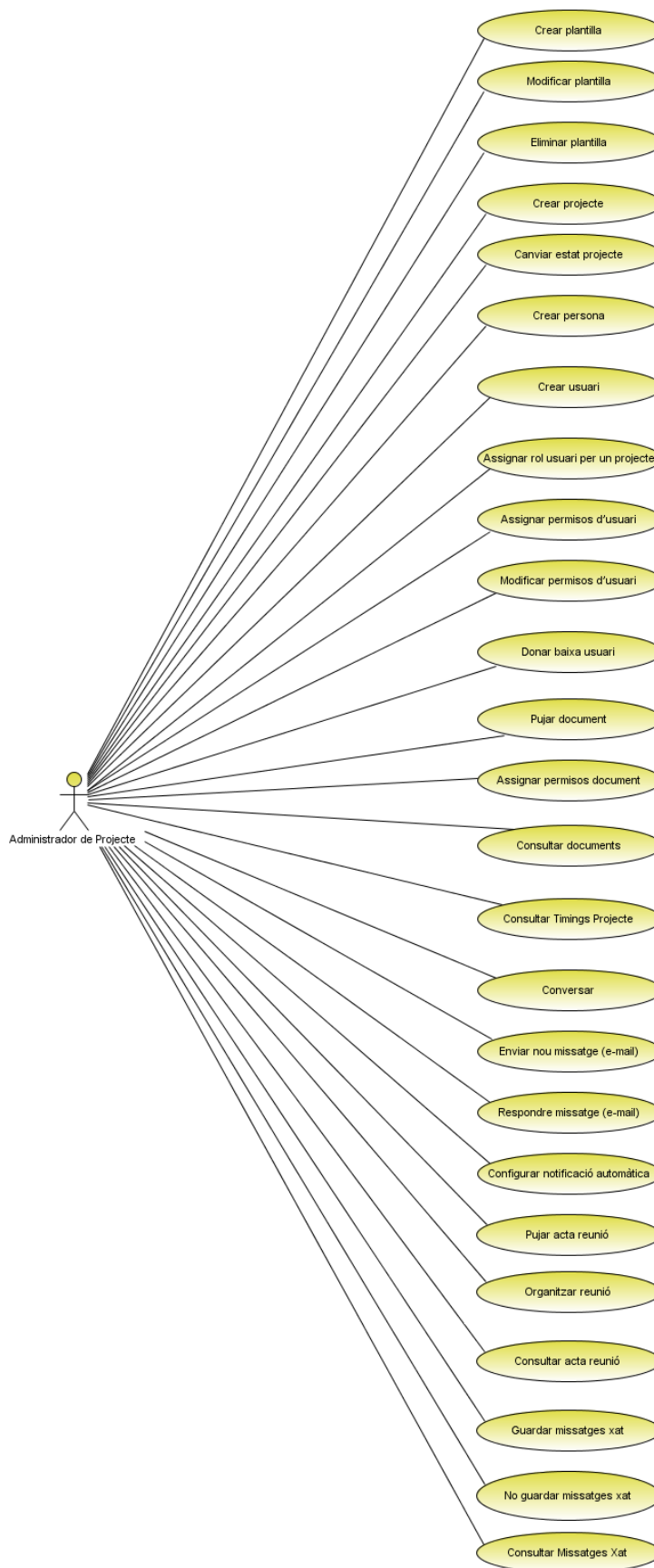


figura 15

### Diagrama Líder o Coordinador de tasca



figura 16

### Diagrama Membre d'equip



figura 17

### Diagrama Administrador de sistema

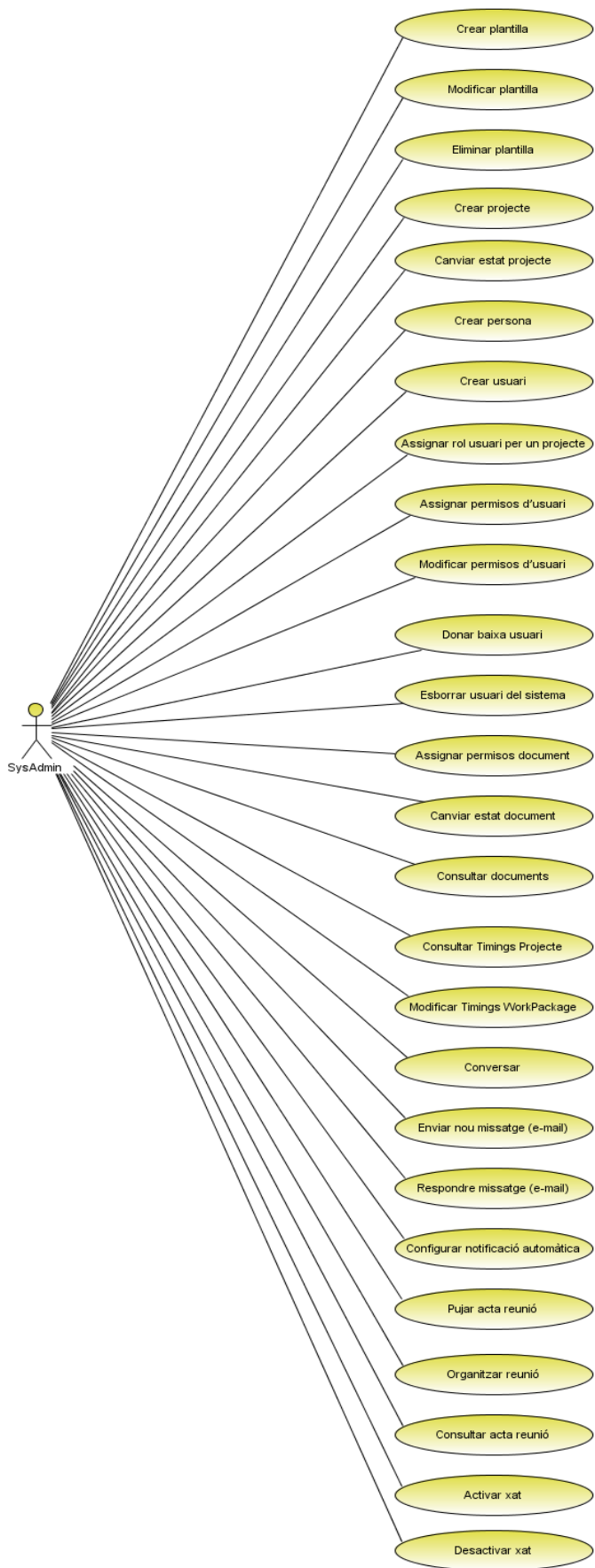


figura 18

### Diagrama Sistema

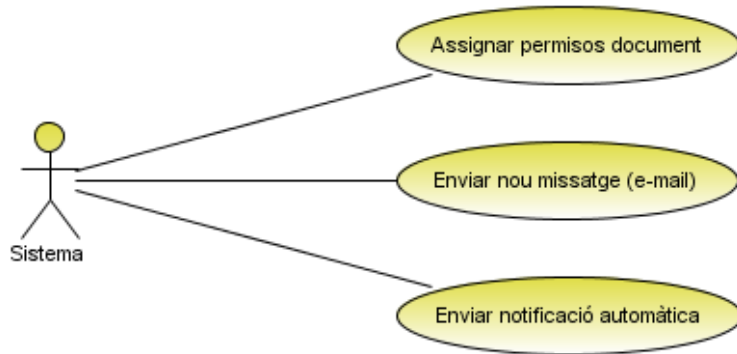


figura 19

### Diagrama Validador de tasca

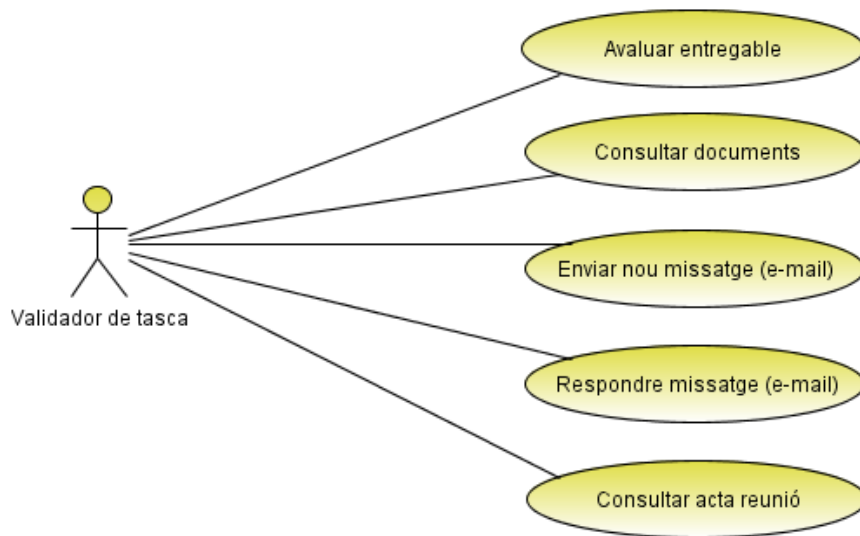


figura 20

**Cas d'ús: Crear plantilla**

**Objectiu:** Crear una plantilla a partir de la qual es puguin generar projectes de la mateixa estructura.

**Disparador:** Es vol crear una plantilla de projecte per a poder-la fer servir més endavant per a projectes similars.

**Actor principal:** Director de projectes (DDP)

**Personal involucrat:** Director de projectes. La resta d'usuaris amb privilegis suficients per a poder fer servir les PLTLL públiques creades pel DDP.

**Precondició:** El DDP s'ha d'identificar i autenticar al sistema.

**Postcondició:** La nova PLTLL s'ha generat i ha quedat emmagatzemada en el repositori de plantilles.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'usuari la informació per crear una nova plantilla: número de workpackage, número de entregables per workpackage, usuaris i els seus rols, número de reunions.
- 2.- L'usuari completa com a mínim la informació obligatòria (número de workpackage, número d'entregables, la plantilla és pública).
- 3.- El sistema enregistra la nova plantilla
- 4.- Fi



**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

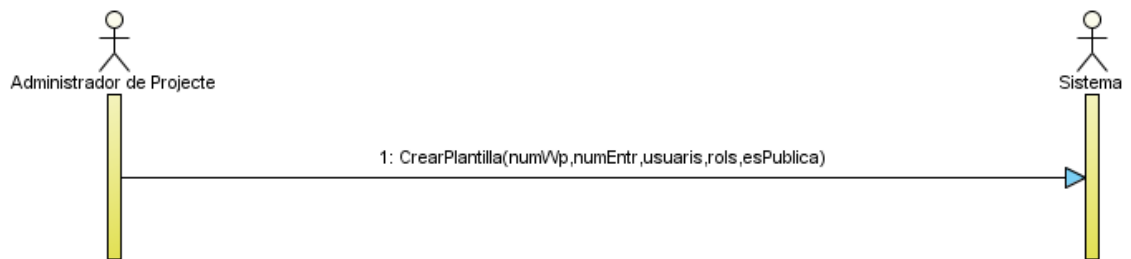
**Diagrama de seqüència operació de sistema:**

figura 21

**Cas d'ús: Modificar plantilla**

**Objectiu:** Modificar el contingut d'una plantilla creada prèviament.

**Disparador:** Un DDP vol modificar una plantilla del repositori de plantilles.

**Actor principal:** Director de projectes (DDP)

**Personal involucrat:** Director de projectes. Vol poder modificar una plantilla (PLTLL) que ja es troba al repositori. La resta de DDP que facin servir la plantilla que serà modificada en cas de ser aquesta pública.

**Precondició:** La plantilla que es vol modificar ha d'haver estat creada pel mateix DDP que vol modificar-la.

**Postcondició:** La plantilla ja existent ha estat modificada i afegida al repositori. La versió anterior a la modificada ja no és accessible.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'usuari la informació que identifica la plantilla a modificar.
- 2.- L'usuari introdueix al sistema les noves dades de la plantilla a modificar (número de workpackage, número d'entregables per workpackage, si la plantilla és pública).
- 3.- El sistema enregistra la modificació de la plantilla.
- 4.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

2a.- El DDP tria una PLTLL pública que no ha estat originalment creada per ell:

El sistema informa a l'usuari de que no pot modificar un PLTLL pública que no hagi creat ell

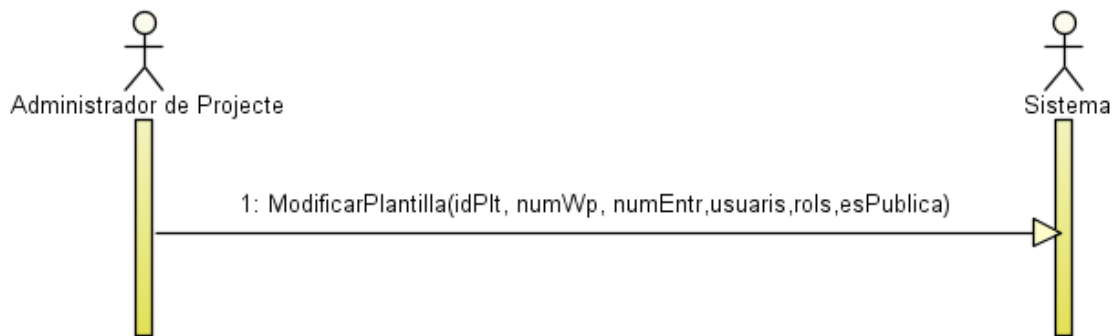
**Diagrama de seqüència operació de sistema:**

figura 22

**Cas d'ús: Eliminar plantilla**

**Objectiu:** Eliminar una plantilla del conjunt disponible de plantilles de projecte.

**Disparador:** El DDP vol eliminar una plantilla del repositori.

**Actor principal:** Director de projectes (DDP)

**Personal involucrat:** Director de projectes. Vol poder eliminar una plantilla (PLTLL) que ja es troba al repositori.

**Precondició:** La plantilla que es vol eliminar ha d'haver estat creada pel mateix DDP que vol modificar-la.

**Postcondició:** La plantilla eliminada deixa d'estar disponible desapareix del repositori de plantilles.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'usuari la informació que identifica a la plantilla a eliminar.
- 2.- El sistema elimina la plantilla del sistema.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

- 1a.- El DDP tria una PLTLL pública que no ha estat originalment creada per ell:  
El sistema informa a l'usuari de que no pot eliminar una PLTLL pública que no hagi creat ell i li permet triar un altre.

3a.- El DDP vol eliminar una PLTLL que té visibilitat pública:

El sistema mostra un avís al DDP recordant que la PLTLL pot ser útil per altres usuaris amb privilegis per a fer-la servir i li demana confirmació per eliminar-la definitivament.

**Diagrama de seqüència operació de sistema:**

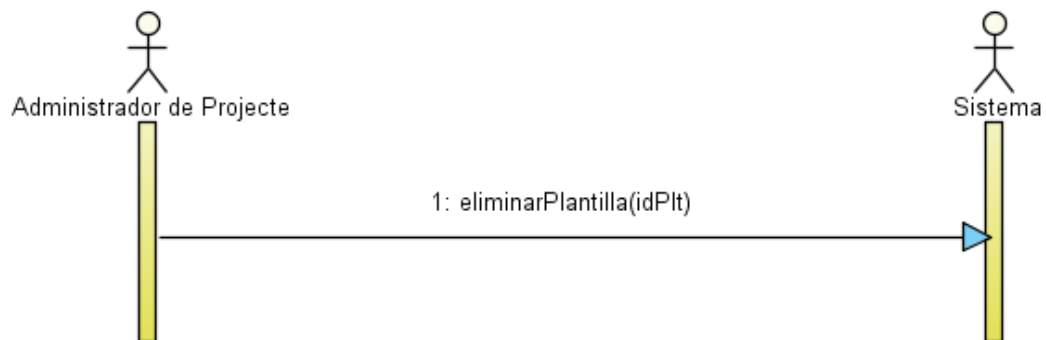


figura 23

**Cas d'ús: Crear projecte**

**Objectiu:** Crear l'estructura de un projecte amb totes les seves dades inicialitzades.

**Disparador:** El DDP vol crear un nou projecte al sistema.

**Actor principal:** Director de projectes (DDP)

**Personal involucrat:** DDP, els líders d'una tasca d'un projecte, els membres de l'equip de treball d'un DLV d'un projecte, els validadors d'un DLV.

**Precondició:** El DDP s'ha d'identificar i autenticar al sistema.

**Postcondició:** S'ha creat un nou esborrany de projecte (i possiblement una PLTLL) al sistema que té tots els seus WP amb tots els seus DLV definits. A més tots els DLV tenen una data d'inici i final que pot estar o no definida.

**Escenari principal d'èxit:**

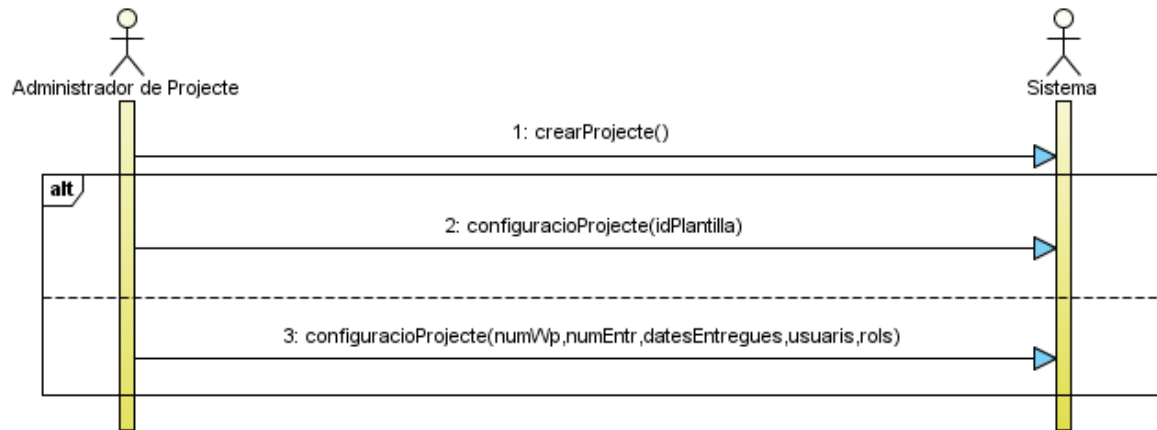
- 1.- El sistema demana al DDP les dades per donar d'alta un nou projecte.
- 2.- El DDP introdueix les dades necessàries per a crear un nou projecte fent servir una plantilla o bé des de zero.
- 3.- El sistema enregistra un nou projecte amb les dades introduïdes pel DDP.
- 4.- Fi

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

**Diagrama de seqüència operació de sistema:****figura 24**

**Cas d'ús: Administrar projecte**

**Objectiu:** Modificar l'estructura d'un projecte, gestionar usuaris i permisos, reunions i veure'n l'estat.

**Disparador:** El DDP vol canviar o visualitzar qualsevol informació relacionada amb un projecte.

**Actor principal:** Director de Projectes(DDP)

**Personal involucrat:** DDP

**Precondició:** El DDP ho és del projecte que es pretén administrar

**Postcondició:** S'han enregistrat canvis realitzats pel DDP en el projecte i s'ha enviat un mail d'avís amb el canvi a les persones implicades en ell.

**Escenari principal d'èxit:**

- 1.- El sistema demana al DDP les dades que identifiquen a un projecte.
- 2.- El DDP, si vol, modifica les dades que configuren el projecte (usuaris, rols, permisos de documents, reunions)introdueix
- 3.- El sistema enregistra els canvis al projecte.
- 4.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

En qualsevol moment el DDP pot decidir fer un canvi de context cap a un altre projecte dels que ell administra:

- 1.- El sistema emmagatzema els canvis al projecte actual (si s'escau)



2.- El sistema mostra a la zona de treball del DDP el nou projecte seleccionat tornant al punt 1

**Diagrama de seqüència operació de sistema:**

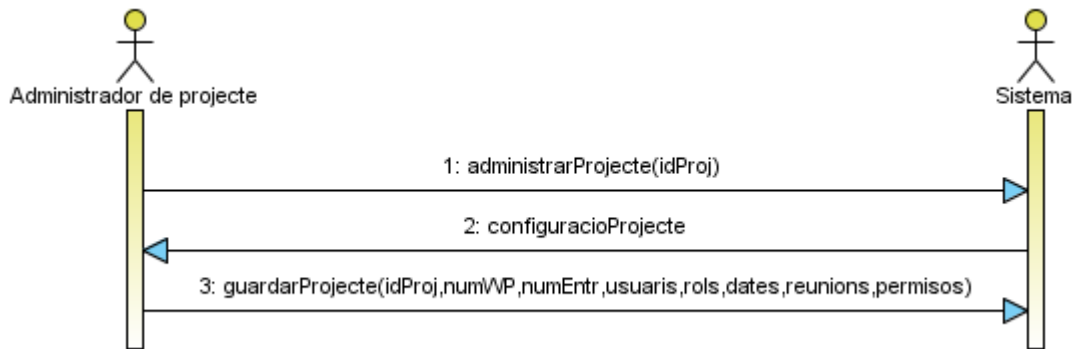


figura 25

**Cas d'ús: Canviar estat projecte**

**Objectiu:** Canviar l'estat d'un projecte a un altre dels seus estats possibles.

**Disparador:** Un DDP vol actualitzar l'estat d'un projecte, per exemple després de produir-se la reunió de kick-off.

**Actor principal:** Director de Projectes(DDP)

**Personal involucrat:** DDP

**Precondició:** Existeix el projecte del qual volem canviar-ne l'estat i l'actor és DDP del projecte

**Postcondició:** S'han enregistrat canvis realitzats pel DDP en el projecte i s'ha enviat un mail d'avís amb el canvi a les persones implicades en ell.

**Escenari principal d'èxit:**

- 1.- El DDP accedeix a modificar les dades d'un projecte.
- 2.- El DDP consulta l'estat actual del projecte i el modifica a un dels estats possibles.
- 3.- El sistema enregistra el canvi i confirma l'acció.
- 4.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

2a.- El DDP pretén canviar a un nou estat que no permetrà el retorn a l'actual:

El sistema demana una confirmació per part del DDP abans de registrar el nou estat.

2b.- El DDP tracta d'instaurar un estat que no és accessible des de l'actual:

El sistema informa al DDP de que l'acció que tracte de dur a terme no està permesa.

**Diagrama de seqüència operació de sistema:**

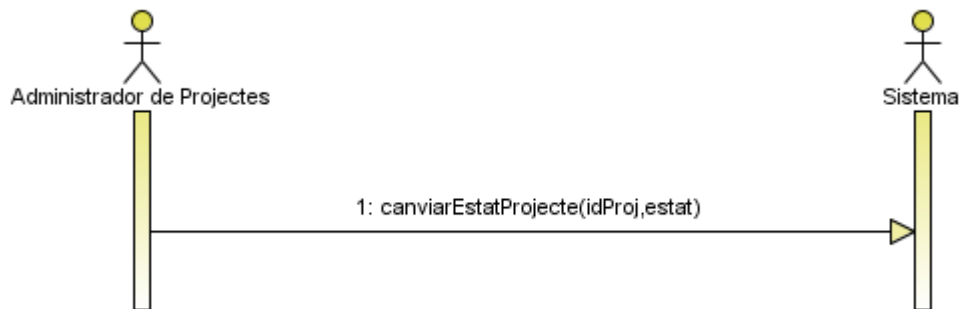


figura 26

**Cas d'ús: Crear persona**

**Objectiu:** Incloure al sistema les dades que identifiquen una persona que té a veure amb un o més dels projectes del sistema però que no és usuari del mateix.

**Disparador:** Cal incloure al sistema dades d'una persona que no és usuari però està relacionada amb algun dels projectes.

**Actor principal:** SA,DDP,TL

**Personal involucrat:** DDP

**Precondició:** No existeix la persona a registrar al sistema amb les mateixes dades que identifiquen a la que es pretén donar d'alta

**Postcondició:** S'ha registrat una nova persona al sistema amb les dades introduïdes

**Escenari principal d'èxit:**

- 1.- El sistema demana les dades necessàries per donar d'alta una nova persona al sistema
- 2.- L'actor introdueix les dades que identifiquen la persona
- 3.- El sistema enregistra la nova persona amb les dades introduïdes
- 4.- El sistema pregunta a l'actor si vol donar d'alta la persona com usuari
- 5.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

2a.- L'actor introdueix les dades d'una persona que ja consta al sistema:

El sistema informa a l'usuari de que ja existeix una persona identificada amb les mateixes dades.

4a.- L'actor confirma la petició de donar d'alta un nou usuari amb les dades de la persona creada:

El sistema completa executa el cas d'ús Crear Usuari completant les dades ja introduïdes per l'actor que identifiquen la persona.

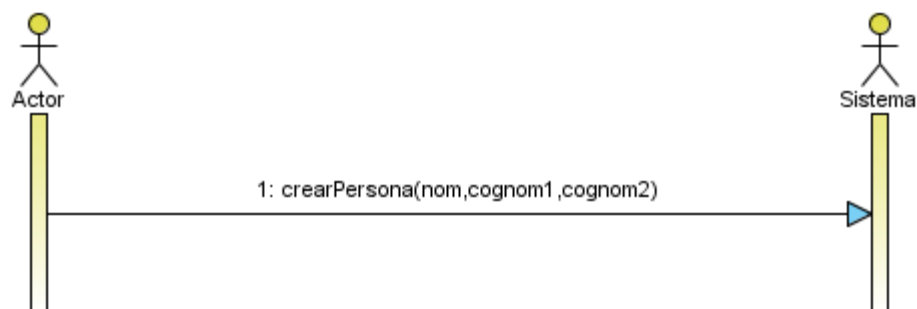
**Diagrama de seqüència operació de sistema:**

figura 27

**Cas d'ús: Crear usuari**

**Objectiu:** Donar d'alta un nou usuari, a partir de les dades d'una persona ja inclosa al sistema, o des de zero assignant-la a un o més dels projectes del sistema i assignant-li un rol específic per a cadascun d'ells.

**Disparador:** Es vol donar d'alta al sistema un usuari que participa en un o més projectes del sistema i que ha de poder interactuar.

**Actor principal:** SA,DDP,TL

**Personal involucrat:** usuari

**Precondició:** No existeix l'usuari a registrar al sistema amb les mateixes dades que identifiquen al que es pretén donar d'alta

**Postcondició:** S'ha registrat un nou usuari al sistema amb les dades introduïdes

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'actor les dades que identifiquen una persona al sistema (nom, cognoms, projectes als que participa, rol dins dels projectes).
- 2.- El sistema mostra el resultat de la cerca de persones que coincideixen amb les dades introduïdes.
- 3.- L'actor selecciona una persona de la llista, si és el cas, i en completa les dades necessàries per donar-la d'alta com a usuari.
- 4.-El sistema enregistra el nou usuari amb una contrasenya temporal i li envia un correu al nou usuari.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

1a.- Inclou: crearPersona()

2a.- Si el sistema té una persona identificada amb les mateixes dades que el nou usuari l'esborra del sistema com a persona.

**Diagrama de seqüència operació de sistema:**

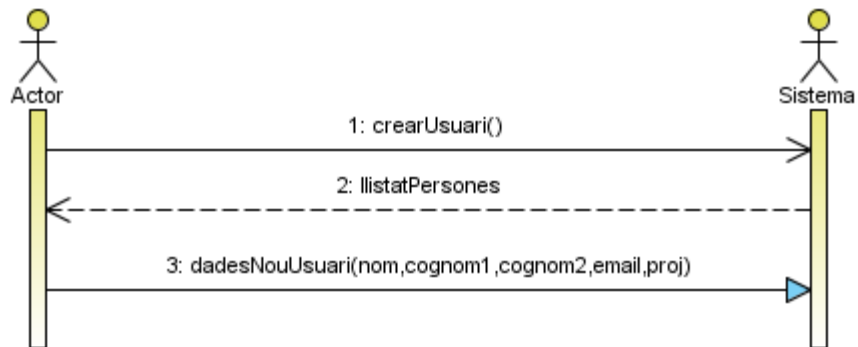


figura 28

**Cas d'ús: Assignar rol d'usuari per un projecte**

**Objectiu:** Assignar a una persona un dels rols disponibles dins d'un projecte.

**Disparador:** Un usuari, amb privilegis suficients, vol donar una sèrie de funcionalitats dins del sistema a un usuari per a un projecte determinat.

**Actor principal:** SA,DDP,TL

**Personal involucrat:** SA,DDP,TL, usuari a assignar.

**Precondició:** Existeix el projecte al qual ha d'estar associada la persona amb el rol

**Postcondició:** Si la persona no estava al sistema, s'ha creat un usuari nou amb les seves dades i s'ha associat amb el projecte amb un rol determinat per l'actor.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'usuari les dades que identifiquen a un usuari del sistema.
- 2.- El sistema mostra una llista de projectes i rols
- 3.- El sistema demana el projecte i el rol per a l'usuari.
- 3.- El sistema enregistra el rol especificat per a l'usuari dins del projecte especificat.
- 4.- Fi

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

1a.- Usuari seleccionat no apareix al sistema: inclou Crear Usuari

1b.- Usuari seleccionat apareix com a persona però no com a usuari: inclou Crear Usuari



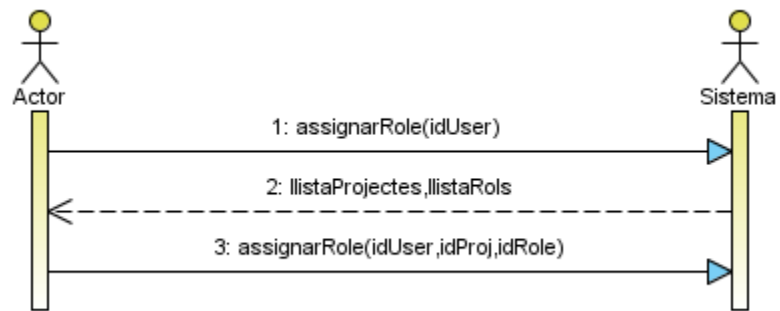
**Diagrama de seqüència operació de sistema:**

figura 29

**Cas d'ús: Assignar permisos d'usuari**

**Objectiu:** Assignar permisos a un usuari ja associat a un projecte.

**Disparador:** Cas d'ús Assignar rol d'usuari per un projecte.

**Actor principal:** SA,DDP,TL

**Personal involucrat:** SA, DDP, TTL, usuari

**Precondició:** Existeix al sistema l'usuari al qual es vol atorgar permisos.

**Postcondició:** S'han modificat els permisos de l'usuari.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'actor les dades que identifiquen a un usuari.
- 2.- El sistema mostra la llista de permisos.
- 3.- L'actor principal introdueix els nous permisos per a l'usuari.
- 4.- Fi

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

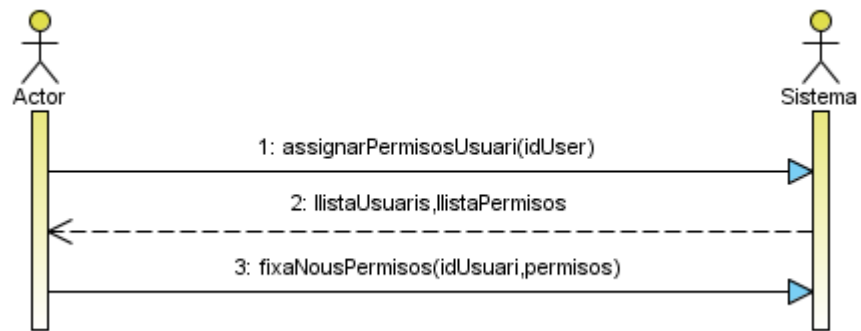
**Diagrama de seqüència operació de sistema:**

figura 30

**Cas d'ús: Modificar permisos d'usuari**

**Objectiu:** Modificar permisos de visibilitat d'informació d'un usuari d'un projecte independentment del seu rol.

**Disparador:** Un usuari, amb privilegis suficients, vol modificar els permisos d'un usuari amb un rol inferior al seu per a un projecte determinat.

**Actor principal:** SA,DDP,TL

**Personal involucrat:** SA,DDP,TL, usuari a modificar.

**Precondició:** Existeix l'usuari i treballa a un projecte gestionat per el DDP o una tasca liderada pel TL.

**Postcondició:** S'han modificat els permisos de l'usuari relacionats al projecte gestionat per DDP o una tasca liderada pel TL.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'actor les dades que identifiquen a un usuari.
- 2.- El sistema mostra els la llista de projectes on participa l'usuari i els seus rols.
- 3.- L'actor principal modifica els permisos de l'usuari per a aquell projecte
- 4.- El sistema enregistra els canvis i n'informa als usuaris si cal.
- 5.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

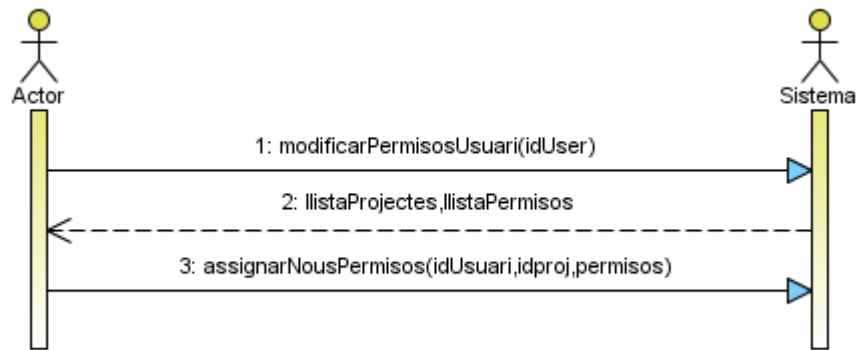
**Diagrama de seqüència operació de sistema:**

figura 31

**Cas d'ús: Donar baixa usuari**

**Objectiu:** Desactivar un usuari que per un motiu qualsevol abandona un projecte i, per tant, deixa de tenir-ne accés.

**Disparador:** Un usuari vol desactivar un altre usuari, amb rol inferior al seu al sistema i dependent d'ell, dins d'un projecte.

**Actor principal:** SA,DDP,TL

**Personal involucrat:** SA, DDP, TL, usuari a eliminar.

**Precondició:** Existeix l'usuari al sistema com usuari en estat d'alta.

**Postcondició:** L'estat de l'usuari al sistema ha passat a ser baixa, ha desactivat el seu accés al sistema i ha notificat del canvi als TL de les tasques de tots els projectes on participava l'usuari.

**Escenari principal d'èxit:**

- 1.- L'actor principal introdueix la llista de dades que identifiquen el usuari/s que es volen donar de baixa
- 2.- El sistema enregistra el nou estat de l'usuari/s.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

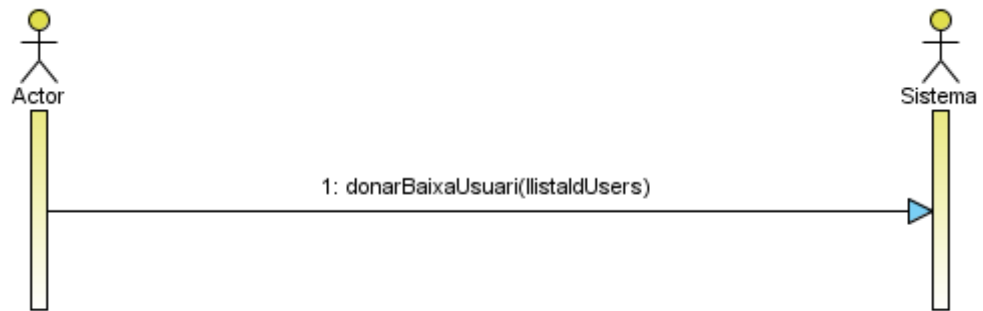
**Diagrama de seqüència operació de sistema:**

figura 32

**Cas d'ús: Esborrar usuari**

**Objectiu:** Eliminar completament de la base de dades a un usuari i tota la seva informació relacionada.

**Disparador:** Un administrador de sistema vol eliminar un usuari del sistema i totes les dades relacionades amb ell.

**Actor principal:** SA

**Personal involucrat:** SA, usuari a eliminar.

**Precondició:** Existeix l'usuari al sistema.

**Postcondició:** L'usuari ja no existeix al sistema.

**Escenari principal d'èxit:**

- 1.- L'actor principal introdueix la llista de dades que identifiquen el usuari/s que es volen esborrar del sistema.
- 2.- El sistema esborra dels sistemes els usuaris.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

2a.- Si l'usuari a esborrar està en estat d'alta, el sistema informa al SA de que no es pot eliminar un usuari que està participant en projectes actualment.



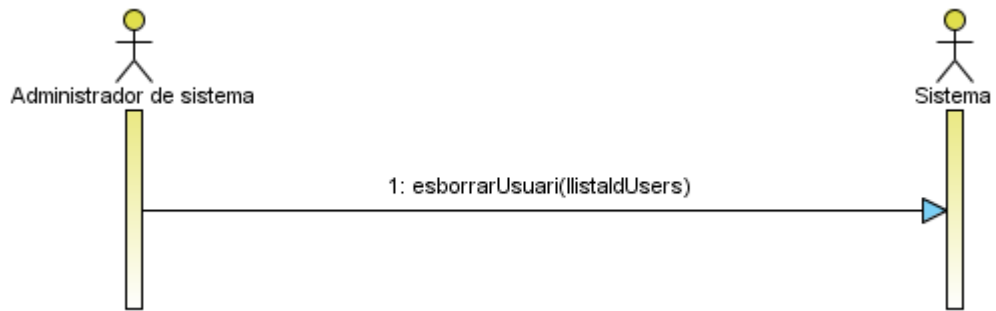
**Diagrama de seqüència operació de sistema:**

figura 33

**Cas d'ús: Pujar document**

**Objectiu:** Afegir un document relacionat amb un projecte al sistema de manera que pugui ser compartit pels usuaris que formen part del mateix projecte segons els permisos d'accés establerts.

**Disparador:** Un usuari vol incloure al sistema un nou document o pujar una nova versió d'un existent.

**Actor principal:** usuari

**Personal involucrat:** usuari que puja el document, SISTEMA

**Precondició:** l'usuari ha de formar part de l'equip de treball de la tasca amb la que està relacionat el document que es vol pujar o bé se DDP del projecte.

**Postcondició:** El document passa a formar part de la documentació relacionada amb la tasca de la que forma part de l'equip l'usuari que l'ha pujat amb els permisos seleccionats per aquest i s'ha enviat notificació als usuaris interessats.

**Escenari principal d'èxit:**

1.- El sistema demana a l'usuari les dades que identifiquen el document: (projecte i tasca als que està vinculat, nom del document, tipus de contingut, idDoc original si és una nova versió i permisos.

2.- El sistema carrega el nou document i l'associa als documents del projecte i la tasca.

3.- Fi

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

1a.- Inclou: Assignar permisos document

1a.- Si qui puja el document és un TM el SISTEMA assigna els permisos configurats per defecte.

**Diagrama de seqüència operació de sistema:**

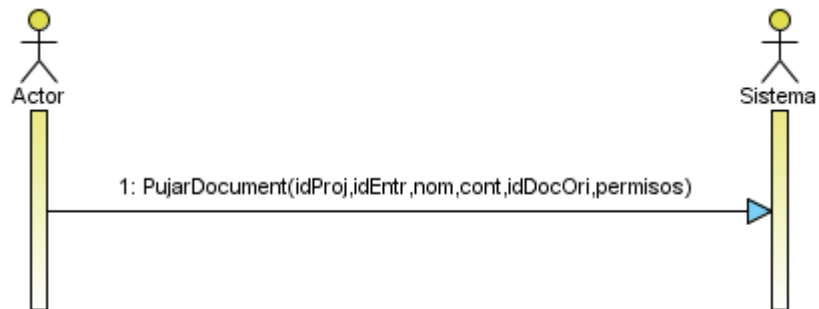


figura 34

**Cas d'ús: Assignar permisos document**

**Objectiu:** Assignar permisos d'accés a un document del sistema. De manera que tan sols sigui accessible per aquelles persones que hi tenen accés.

**Disparador:** cas d'ús Pujar document

**Actor principal:** usuari

**Personal involucrat:** usuari que puja el document.

**Precondició:** El document al qual es volen assignar permisos existeix al sistema

**Postcondició:** S'han afegit permisos al document.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'actor les dades que identifiquen un document.
  
- 1.- El sistema mostra a l'actor principal un llistat amb els permisos que pot fixar per al document segons el seu rol i els usuaris que hi tenen accés.
  
- 2.- L'usuari selecciona els permisos que vol per al document segons rols d'altres usuaris o bé amb particularitats per usuaris individuals.
  
- 3.- El sistema registra el permisos i els associa al document.
  
- 4.- Fi

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

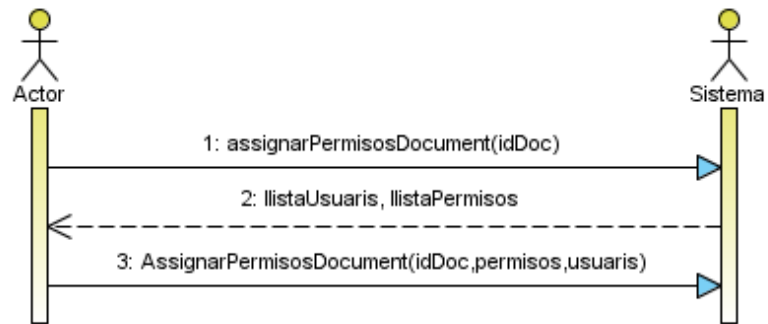
**Diagrama de seqüència operació de sistema:**

figura 35

**Cas d'ús: Canviar estat document**

**Objectiu:** Designar un document com a entregable de manera que passi a ser “validable”

**Disparador:** Un usuari validador vol canviar l'estat d'un document del sistema relacionat amb un workpackage dins d'un projecte.

**Actor principal:** TL

**Personal involucrat:** DDP, TL, VL

**Precondició:** El document del qual es vol canviar l'estat existeix al sistema i no és un entregable

**Postcondició:** El document ha canviat el seu estat a entregable i el sistema ha notificat al DDP i als TV d'aquella tasca que hi ha un nou entregable pendent de Avaluació.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'actor les dades que identifiquen a un document i el nou estat que s'ha d'assignar al document.
- 2.- El sistema registra el nou estat del document i notifica al Validadors i al DDP si cal de que el document pot ser avaluat.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

1a.- Si la tasca ja té un document marcat com a entregable, el sistema notifica al TV que ja existeix un document marcat com entregable i pendent de validació.

4a.- Si la data límit marcada per entregar el document al TV ha estat sobrepassada el sistema canvia l'estat a "entregable endarrerit" i ho afegeix a les notificacions per al DDP i el TV (i el client si s'ha inclòs com a receptor de notificació).

**Diagrama de seqüència operació de sistema:**

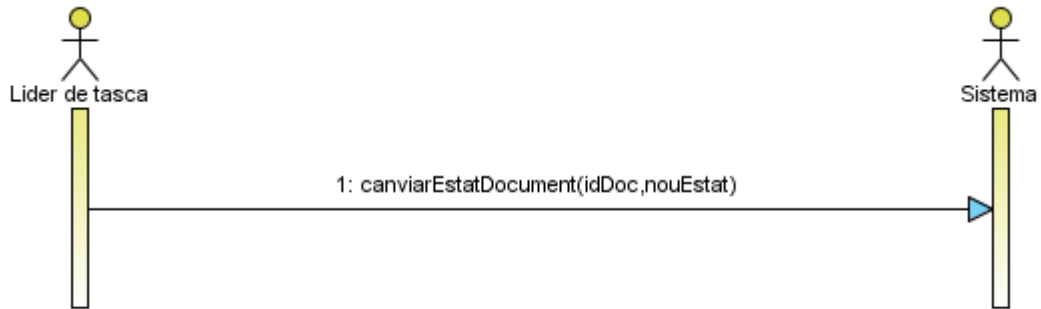


figura 36

**Cas d'ús: Avaluar entregable**

**Objectiu:** Obtenir l'avaluació d'un document "entregable" per part del/s TV d'una tasca

**Disparador:** El TV té nous entregables per avaluar.

**Actor principal:** TV, SISTEMA

**Personal involucrat:** DDP, TL, VL

**Precondició:** El TV no ha emès la seva valoració per a l'entregable que se li demana.

**Postcondició:** El TV ha enregistrat la valoració del TV per a l'entregable

**Escenari principal d'èxit:**

- 1.- El sistema demana al TV les dades que identifiquen al document a validar i la seva valoració i ofereix la possibilitat d'incloure un text explicatiu per al TL i un altre pel DDP.
- 2.- El sistema enregistra la valoració fet pels validadors i les notifica al TL i al DDP si cal.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

3a.- Si el TV era l'últim en avaluar l'entregable dels associats a aquell i la valoració total de l'entregable és "Vàlida" l'entregable passa a estat "Vàlid" i el SISTEMA ja no acceptarà més documents "entregables" per aquella tasca, altrament passa a estat "No vàlid" i el sistema passa a acceptar un nou entregable.



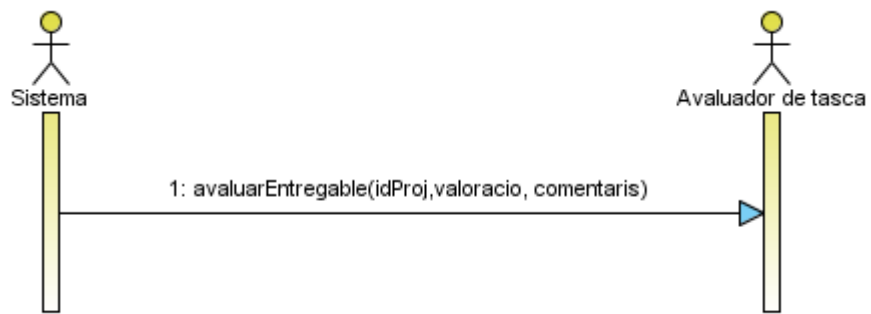
**Diagrama de seqüència operació de sistema:**

figura 37

**Cas d'ús: Consultar document**

**Objectiu:** Consultar el contingut de qualsevol dels documents per al que un usuari té permisos.

**Disparador:** un dels actors principals vol consultar un document.

**Actor principal:** CLIENT, DDP, TL , TM.

**Personal involucrat:** Actor principal i SISTEMA

**Precondició:** L'actor principal té permisos sobre el document

**Postcondició:** S'ha mostrat la informació associada i el contingut del document a l'actor principal.

**Escenari principal d'èxit:**

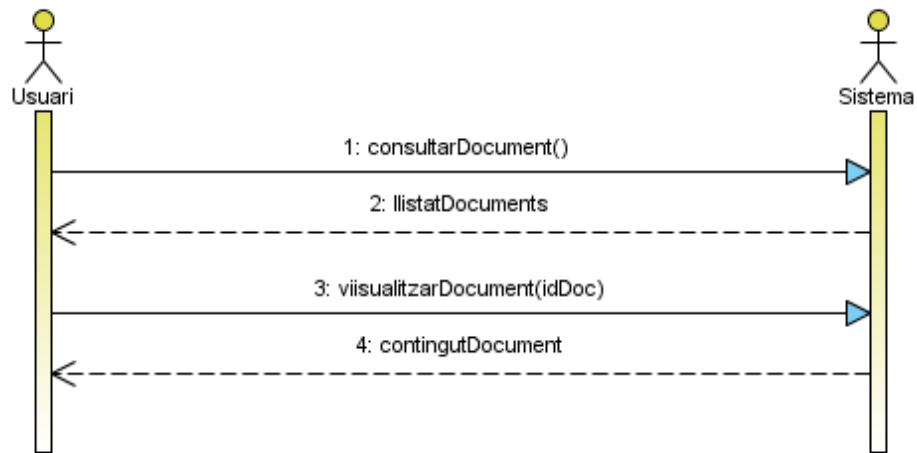
- 1.- l'Actor principal selecciona un dels documents d'entre una llista ofertada pel sistema amb el documents als quals hi té accés.
- 2.- El sistema mostra la informació a la que l'actor tingui accés per al document i li ofereix la possibilitat d'obrir-lo si té permisos habilitats.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

**Diagrama de seqüència operació de sistema:****figura 38**

**Cas d'ús: Consultar timings projecte**

**Objectiu:** Consultar els períodes de temps dedicats a cadascuna de les tasques d'un projecte i els punts de reunió.

**Disparador:** un dels actors principals vol consultar les dates d'entrega.

**Actor principal:** CLIENT, DDP, TL , TM.

**Personal involucrat:** Actor principal i SISTEMA

**Precondició:** --

**Postcondició:** S'ha mostrat la informació dels timings del projecte.

**Escenari principal d'èxit:**

- 1.- l'Actor principal selecciona un dels projectes en que hi participa.
- 2.- El sistema mostra el diagrama de Gantt associat a aquell projecte.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

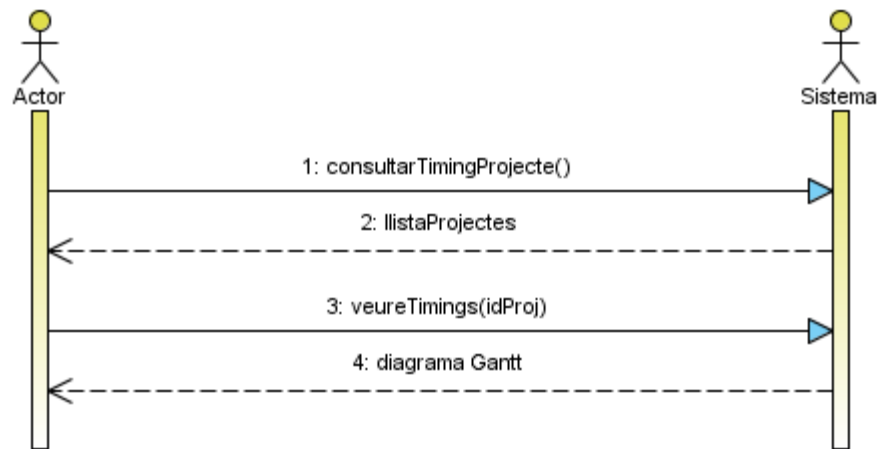
**Diagrama de seqüència operació de sistema:**

figura 39

**Cas d'ús: Modificar timings entregable**

**Objectiu:** Modificar els períodes de temps dedicats a l'entrega i validació d'un entregable associat a una tasca del projecte.

**Disparador:** Un dels actors principals vol modificar una o més dates associades a un dels entregables en que hi participa.

**Actor principal:** TL

**Personal involucrat:** Tots

**Precondició:** la nova data inicial no pot ser anterior a la data final d'alguna de les tasques precedents a la que es vol modificar.

**Postcondició:** S'han modificat les dates associades a l'entregable de la tasca modificada.

**Escenari principal d'èxit:**

- 1.- l'actor principal selecciona una tasca d'entre una llista presentada pel sistema amb les que ell lidera.
- 2.- El sistema mostra la informació temporal per aquella tasca
- 3.- El TL modifica els valors
- 4.- El sistema demana confirmació i pregunta a l'usuari si vol notificar també al client la replanificació
- 5.- Si el TL confirma el pas 3, el sistema registra els canvia, replanifica les dates de les tasques que es puguin veure afectades pel canvi i notifica els canvia als usuaris interessats.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

5a.- Si la data inicial s'ha modificat de manera que la tasca s'hauria d'iniciar abans de la conclusió planificada d'una o més de les seves entregues predecessors, el SISTEMA mostra un missatge informatiu de que la data inicial no es vàlida.

**Diagrama de seqüència operació de sistema:**

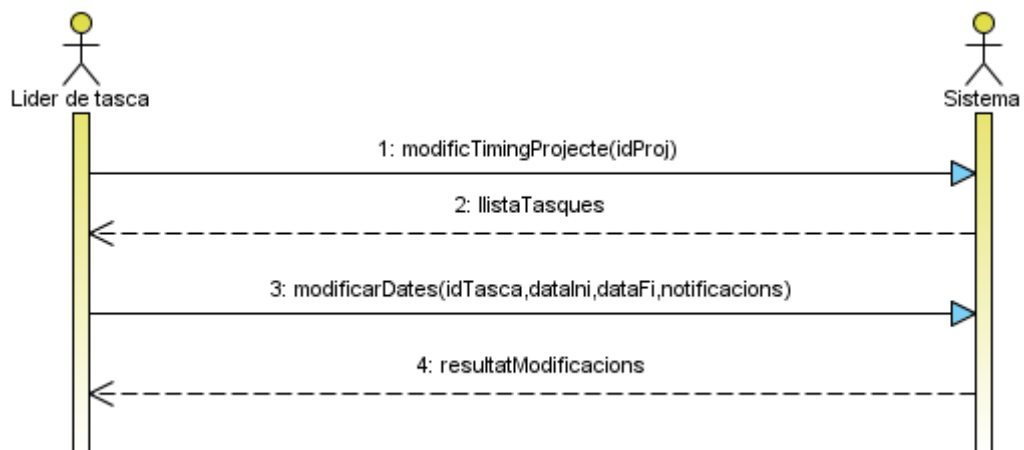


figura 40

**Cas d'ús: Conversar**

**Objectiu:** Comunicar-se amb un altre usuari del sistema en temps real.

**Disparador:** un dels actors principals vol xatejar amb un usuari o grup d'usuaris.

**Actor principal:** Usuari

**Personal involucrat:** Usuaris.

**Precondició:** Els usuaris participants estan connectats al sistema.

**Postcondició:** S'ha enviat un missatge d'un usuari cap a un o altre usuaris connectats.

**Escenari principal d'èxit:**

- 1.- L'actor principal selecciona un usuari dels que apareixen a la llista d'usuaris connectats.
- 2.- El sistema mostra una conversa on apareixen els missatges intercanviats pels usuaris.
- 3.- El sistema enregistra els continguts de les converses.
- 4.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

En qualsevol moment un dels participants a la conversa decideix afegir un nou participant:

El sistema ho notifica a cadascun dels participants a la conversa mitjançant un missatge comú.

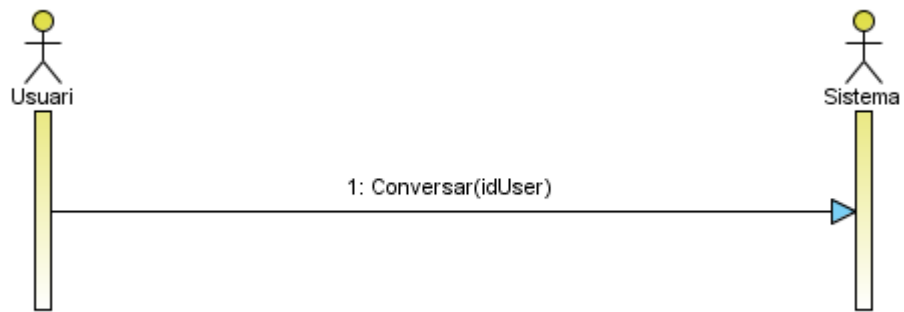
2a.- inclou: Enviar missatge conversa

2b.- inclou: Rebre missatge conversa

2c.- Si un dels participants abandona el sistema o la conversa, el sistema ho notifica a la resta dels participants i n'actualitza la llista d'usuaris connectats.



**Diagrama de seqüència operació de sistema:**



**figura 41**

**Cas d'ús: Enviar missatge conversa**

**Objectiu:** Comunicar-se amb un altre usuari del sistema en temps real.

**Disparador:** un dels actors principals vol enviar un missatge usuari o grup d'usuaris.

**Actor principal:** Usuari

**Personal involucrat:** Usuaris.

**Precondició:** Els usuaris participants estan connectats al sistema.

**Postcondició:** S'ha enviat un missatge d'un usuari cap a un o altre usuaris connectats.

**Escenari principal d'èxit:**

- 1.- El sistema demana els usuaris que han de rebre el missatge i el contingut del missatge.
- 2.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

**Diagrama de seqüència operació de sistema:**

figura 42

**Cas d'ús: Rebre missatge conversa**

**Objectiu:** Comunicar-se amb un altre usuari del sistema en temps real.

**Disparador:** El sistema fa arribar un missatge d'un usuari a un altre.

**Actor principal:** Usuari

**Personal involucrat:** Usuaris.

**Precondició:** Els usuaris participants estan connectats al sistema.

**Postcondició:** S'ha enviat un missatge d'un usuari cap a un o altre usuaris connectats.

**Escenari principal d'èxit:**

1.- El sistema notifica a l'usuari que l'usuari emissor li envia un missatge mostrant-li el contingut.

2.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

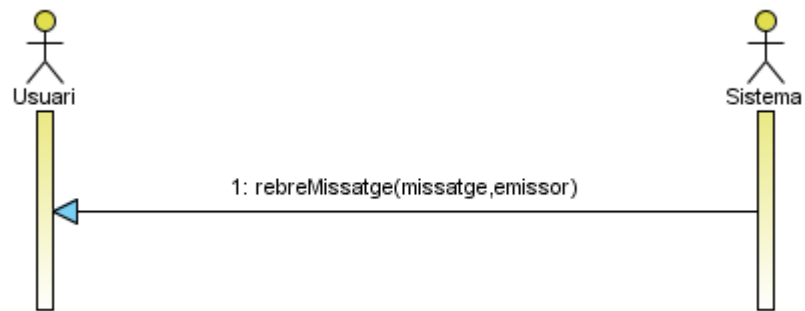
**Diagrama de seqüència operació de sistema:**

figura 43

**Cas d'ús: Enviar nou missatge**

**Objectiu:** Comunicar-se amb un altre usuari o usuaris de manera diferida.

**Disparador:** un dels actors principals vol intercanviar un missatge amb un usuari o grup d'usuaris o respondre un altre missatge.

**Actor principal:** Usuari

**Personal involucrat:** Usuaris.

**Precondició:** l'usuari al que s'envia el missatge ha d'estar al sistema.

**Postcondició:** S'ha avisat a la zona de notificacions dels receptors del nou missatge per part de l'actor principal i a més se'ls ha enviat un correu amb el contingut i un link directe a la zona de notificacions.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'usuari les dades que identifiquen al receptor/s i el contingut del missatge a enviar.
- 2.- El sistema enregistra el missatge i fa aparèixer el missatge a la zona de missatges dels receptors i fa les notificacions oportunes.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

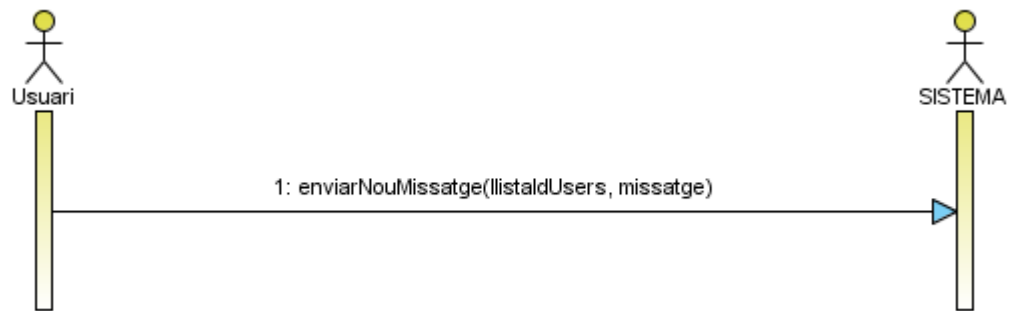
**Diagrama de seqüència operació de sistema:**

figura 44

**Cas d'ús: Respondre missatge**

**Objectiu:** Respondre el missatge enviat per altre usuari.

**Disparador:** un dels actors principals vol intercanviar un missatge amb un usuari o grup d'usuaris o respondre un altre missatge.

**Actor principal:** Usuari

**Personal involucrat:** Usuaris.

**Precondició:** s'ha rebut un missatge al que es respon.

**Postcondició:** S'ha avisat als receptors del nou missatge per part de l'actor principal.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'usuari les dades que identifiquen al missatge original, al receptor/s i el contingut del missatge a enviar.
- 2.- El sistema enregistra el missatge i fa aparèixer el missatge a la zona de missatges dels receptors i fa les notificacions oportunes.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.



**Diagrama de seqüència operació de sistema:**

figura 45

**Cas d'ús: Configurar notificació automàtica per defecte**

**Objectiu:** Fixar la configuració de enviament de notificacions automàtiques per defecte per a un usuari determinat segons el seu rol.

**Disparador:** un dels actors principals vol rebre notificacions dels missatges intercanviats entre els usuaris inferiors a ell dins la jerarquia.

**Actor principal:** SA,DDP, TL, TM, TV

**Personal involucrat:** Usuaris.

**Precondició:** --

**Postcondició:** L'usuari ha configurat quins tipus de notificacions vol rebre de manera automàtica per part del sistema i per quines vol rebre un e-mail a la seva compte de correu.

**Escenari principal d'èxit:**

- 1.- L'actor principal selecciona de quins usuaris dels que pot accedir vol rebre notificacions automàtiques de les seves activitats i per quines vol rebre un correu electrònic.
- 2.- El sistema notifica als usuaris que seran monitoritzats per l'actor principal aquest fet \*.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

2a.- Si un TL vol monitoritzar les activitats d'un altre TL:

El SISTEMA mostrarà a la zona de notificacions del TL que ha de ser monitoritzat una petició per part del TL que vol rebre notificacions, de manera que si aquesta és acceptada, a partir de llavors les seves activitats seran notificades.

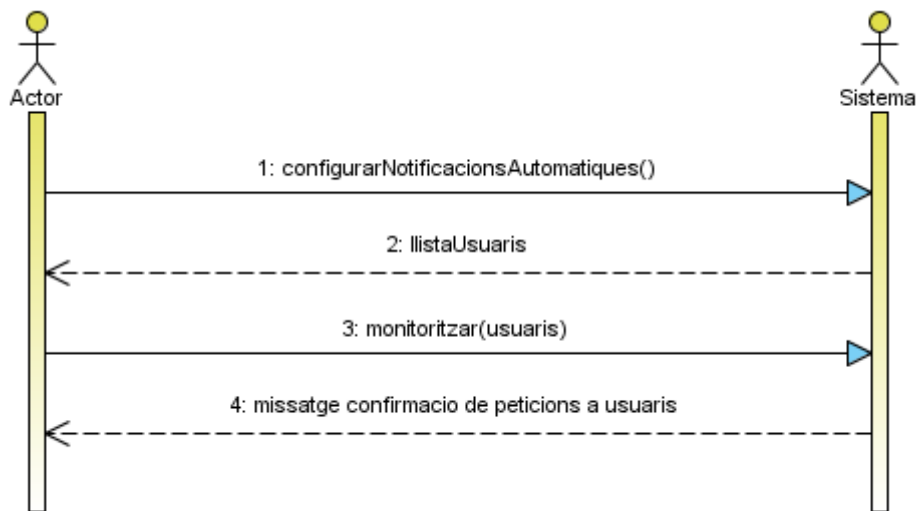
**Diagrama de seqüència operació de sistema:**

figura 46

**Cas d'ús: Enviar notificació automàtica**

**Objectiu:** Notificar als usuaris subscrits una notificació amb informació sobre quelcom observat.

**Disparador:** S'ha produït un canvi al sistema que ha de ser notificat automàticament a un usuari o usuaris.

**Actor principal:** Usuari

**Personal involucrat:** SISTEMA, Usuaris.

**Precondició:** S'ha produït un canvi al subjecte observat

**Postcondició:** S'ha escrit un missatge notificant del canvi a la zona de notificacions del subscriptor i s'han dut a terme les accions programades per aquest en aquest cas.

**Escenari principal d'èxit:**

- 1.- L'element observat pateix un canvi en el seu estat.
- 2.- El SISTEMA notifica a cada subscriptor el canvi i dur a terme les accions configurades en cada cas.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

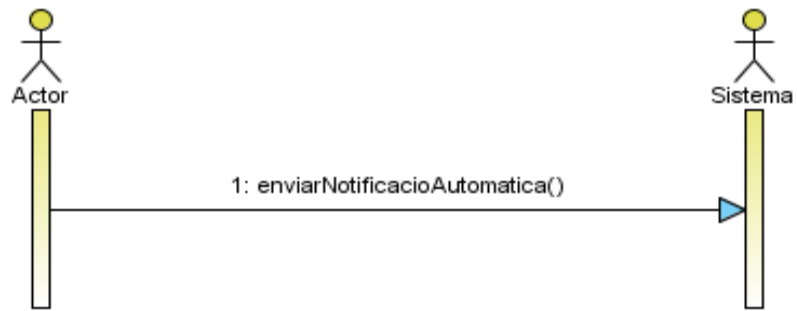
**Diagrama de seqüència operació de sistema:**

figura 47

**Cas d'ús: Organitzar reunió**

**Objectiu:** Organitzar una reunió entre usuaris de la plataforma.

**Disparador:** El DDP o TL vol organitzar una reunió.

**Actor principal:** Usuari

**Personal involucrat:** Usuaris.

**Precondició:** --

**Postcondició:** S'ha avisat a la zona de notificacions dels receptors d'una petició de reunió informant del dia, lloc, hora i temes a tractar.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'actor principal les dades necessàries per a convocar una nova reunió.
- 2.- El sistema demana als usuaris confirmació de participació o bé que proposin un altre dia i/o hora i afegeixin temes a tractar.
- 3.- El pas 2 es repeteix fins obtenir totes les confirmacions dels participants.
- 4.- El SISTEMA notifica als participants el dia, hora, lloc i temes de la reunió així com els assistents i anotacions suplementàries i ho notificarà als usuaris subscrits.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

Un dels participants no podrà assistir a la reunió cap dia:

L'actor principal ha de modificar les persones participants i treure al participant que no hi pot assistir.

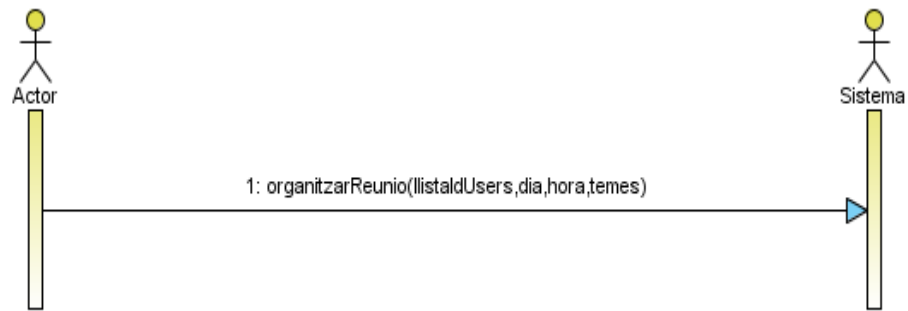
**Diagrama de seqüència operació de sistema:**

figura 48

**Cas d'ús: Pujar acta reunió**

**Objectiu:** Compartir amb la resta d'usuaris membres del projecte al qual pertany la reunió celebrada, els temes tractats i acords

**Disparador:** S'ha produït una reunió de la qual n'ha sortit un document en forma d'acta.

**Actor principal:** usuari

**Personal involucrat:** usuari que puja el document, SISTEMA

**Precondició:** La reunió ja s'ha d'haver realitzat i l'usuari que puja el document ha d'haver participat a la reunió.

**Postcondició:** El document passa a formar part de la documentació relacionada amb la tasca de la que forma part de l'equip l'usuari que l'ha pujat amb els permisos seleccionats per aquest i s'ha enviat notificació als usuaris interessats.

**Escenari principal d'èxit:**

- 1.- El sistema demana a l'usuari les dades que identifiquen l'acta (reunió a la que està vinculada i permisos).
- 2.- El sistema carrega l'acta l'associa als documents del projecte i la tasca.
- 3.- Fi

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

1a.- Si qui puja el document és un TM el SISTEMA assigna els permisos configurats per defecte.

1b.- inclou: Assignar permisos document.



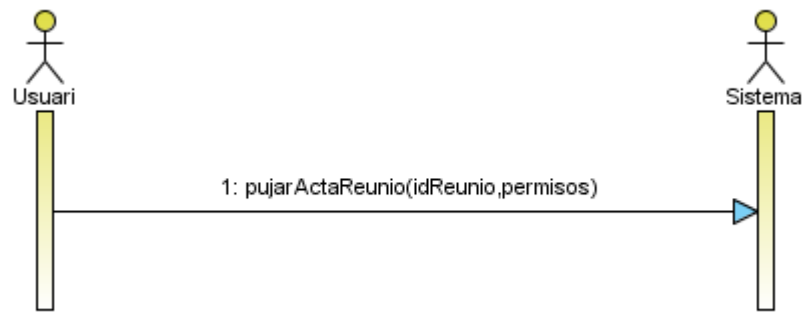
**Diagrama de seqüència operació de sistema:**

figura 49

**Cas d'ús: Consultar acta reunió**

**Objectiu:** Consultar el contingut de l'acta d'una reunió.

**Disparador:** un dels actors principals vol consultar un document.

**Actor principal:** CLIENT, DDP, TL , TM.

**Personal involucrat:** Actor principal i SISTEMA

**Precondició:** L'actor principal té permisos sobre el document

**Postcondició:** S'ha mostrat la informació associada i el contingut del document a l'actor principal.

**Escenari principal d'èxit:**

- 1.- l'Actor principal selecciona un dels documents d'acta d'entre les actes de les reunions a les que ha assistit (excepte si es un DDP qui té accés a totes les actes).
- 2.- El sistema mostra la informació a la que l'actor tingui accés sobre la reunió i li ofereix la possibilitat d'obrir-lo si té permisos habilitats.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

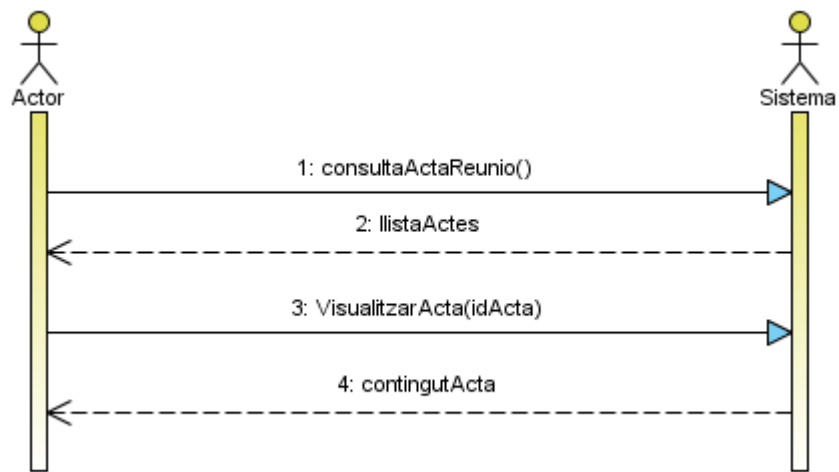
**Diagrama de seqüència operació de sistema:**

figura 50

**Cas d'ús: Activar xat**

**Objectiu:** Activar el xat de l'aplicació.

**Disparador:** un administrador del sistema vol activar el xat.

**Actor principal:** SysAdmin.

**Personal involucrat:** SysAdmin.

**Precondició:** --

**Postcondició:** S'ha activat el xat per a tots els usuaris del sistema.

**Escenari principal d'èxit:**

1.- L'administrador del sistema habilita l'opció de xat.

2.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

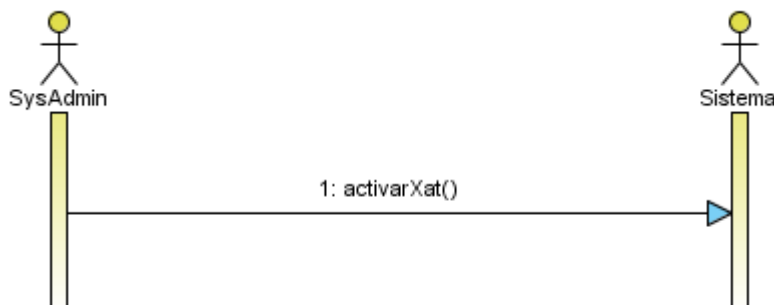
**Diagrama de seqüència operació de sistema:**

figura 51

**Cas d'ús: Desactivar xat**

**Objectiu:** Desactivar el xat de l'aplicació.

**Disparador:** un administrador del sistema vol desactivar el xat.

**Actor principal:** SysAdmin.

**Personal involucrat:** Tots els usuaris.

**Precondició:** --

**Postcondició:** S'ha desactivat el xat per a tots els usuaris del sistema.

**Escenari principal d'èxit:**

1.- L'administrador del sistema deshabilita l'opció de xat.

2.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

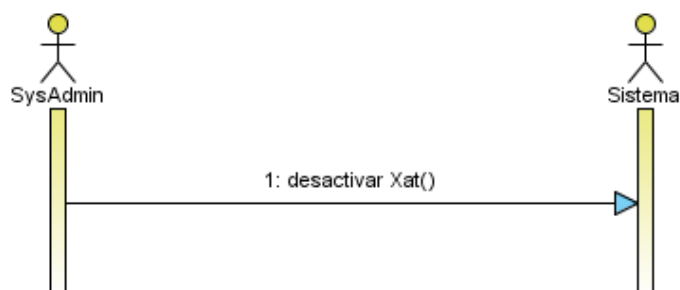
**Diagrama de seqüència operació de sistema**

figura 52

**Cas d'ús: Guardar missatges de xat**

**Objectiu:** Enregistrar els missatges de xat d'un o més usuaris.

**Disparador:** Un administrador de projecte o un líder de tasca vol monitoritzar els missatges d'un o més usuaris.

**Actor principal:** DDP, TL.

**Personal involucrat:** Tots els usuaris.

**Precondició:** El xat ha d'estar activat.

**Postcondició:** Es registraran tots els missatges d'entrada i sortida d'aquell usuari.

**Escenari principal d'èxit:**

1.- L'actor principal indica els usuaris dels que cal enregistrar els missatges.

2.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.

En qualsevol moment s'assigna un rol a un dels usuaris monitoritzats que impedeix, per jerarquia, que pugui seguir sent monitoritzat:

El sistema informa al DDP de que l'usuari deixa de ser "monitoritzable" i deixen d'enregistrar-se els seus missatges.

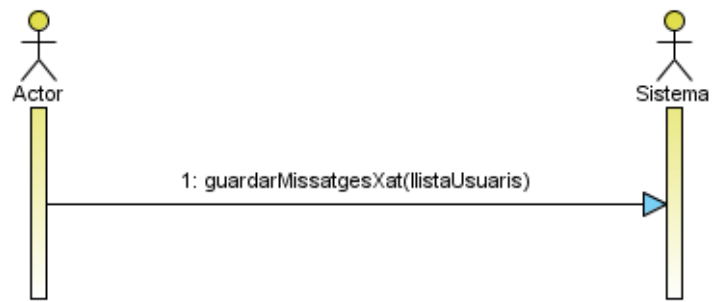
**Diagrama de seqüència operació de sistema**

figura 53

**Cas d'ús: Consultar missatges de xat**

**Objectiu:** Consultar els missatges enregistrats d'un o més usuaris.

**Disparador:** Un administrador de projecte o un líder de tasca vol consultar els missatges guardats d'un o més usuaris.

**Actor principal:** DDP, TL.

**Personal involucrat:** Tots els usuaris.

**Precondició:** --.

**Postcondició:** Es mostra el conjunt de missatges de l'usuari especificat entre les dates especificades.

**Escenari principal d'èxit:**

- 1.- L'actor principal indica l'usuari i l'interval de dates del que el sistema ha d'oferir-li els missatges.
- 2.- El sistema li retorna els missatges.
- 3.- Fi.

**Extensions:**

En qualsevol moment es produeix una fallada del sistema:

\* Si l'error és aliè al sistema:

El DDP ha de reiniciar la operativa, si fa menys d'un temps màxim d'inactivitat permès pel sistema, continua amb la mateixa sessió si no cal que torni a introduir les seves dades per entrar al sistema.



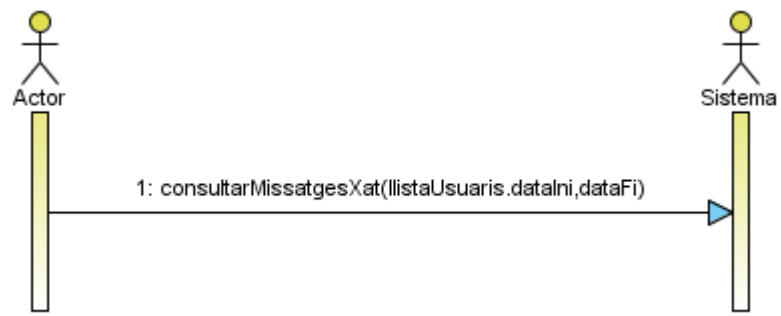
**Diagrama de seqüència operació de sistema**

figura 54

---

## Capítol 6

---

### Disseny del sistema

Un cop completada la fase d'especificació del sistema, el següent pas natural ens condueix a la fase de disseny. En aquesta fase ens centrarem bàsicament en definir dos dels artefactes o diagrames que amb anterioritat hem vist.

#### ***Normalització del model conceptual***

Per una banda i partint del model de domini obtingut, realitzarem els passos necessaris per obtenir el diagrama de classes del sistema. Això és, la transformació del model conceptual cap al model de software del sistema. En aquest punt, haurem de realitzar una sèrie de transformacions sobre el diagrama del model de domini per tal d'adequar-lo al marc propi d'un sistema d'informació mitjançant un procés conegut amb el nom de normalització.

La normalització d'un model conceptual s'obté aplicant una sèrie de passos, com per exemple, la substitució de tota relació n-ària on  $n > 2$  per relacions binàries fent-hi aparèixer les classes necessàries per assolir aquest objectiu. Tanmateix és necessari continuar assegurant el compliment de totes les restriccions definides al model conceptual per tal d'assegurar la consistència del model de classes.

Un cop normalitzat el model conceptual i transformats els objectes del món real en entitats software obtindrem el diagrama de classes. En ell hi apareixen els atributs de cada classe software així com mètodes de comportament d'aquestes classes.

## **Diagrames de seqüència**

A més a més, introduïrem els diagrames de seqüència que són una representació gràfica de la manera com els objectes interactuen els uns amb els altres i que permeten l'assoliment dels objectius definits als casos d'ús.

En aquest fase del desenvolupament entra en joc l'aplicació del que s'anomenen patrons de disseny. Els patrons de disseny poden ser vistos com un repositori de solucions genèriques que poden ser aplicades per donar solució a determinades problemàtiques força comuns i que ja han estat llargament resoltes de manera eficient. L'ús de patrons de disseny, tot i no ser de pràctica obligada, sovint aporta agilitat i robustesa a les solucions trobades per resoldre problemes de disseny i, a més a més, afavoreix aspectes com la mantenibilitat i la seguretat ajudant així, en molts casos, a l'assoliment de requisits no funcionals del sistema que s'està desenvolupant.

Centrant-nos de nou en el cas que ens ocupa, i com a pas previ a mostrar el diagrama complet normalitzat obtingut, farem un recorregut per totes les classes identificant-ne cadascuna d'elles i donant una petita descripció del que representa així com un llistat amb els seus atributs.

De la mateix manera, en els casos escaients, l'explicació s'ha acompanyat amb un diagrama de seqüència on s'explica els diferent estats pels que pot passar l'objecte representat i quins són els esdeveniments que provoquen el pas dels uns als altres

## Procés de normalització de model conceptual

Com ja s'ha dit, el procés de normalització transforma el diagrama del model de domini de manera que desapareixen totes les relacions no binàries i dona lloc a l'aparició de noves classes que serveixen per solucionar aquests procés de transformació fent possible posteriorment la implementació del model en un llenguatge OO i una BD.

Un cop aplicat el procés de normalització sobre el model conceptual inicial, la següent figura mostra el diagrama de classes normalitzat.

Diagrama de classes normalitzat

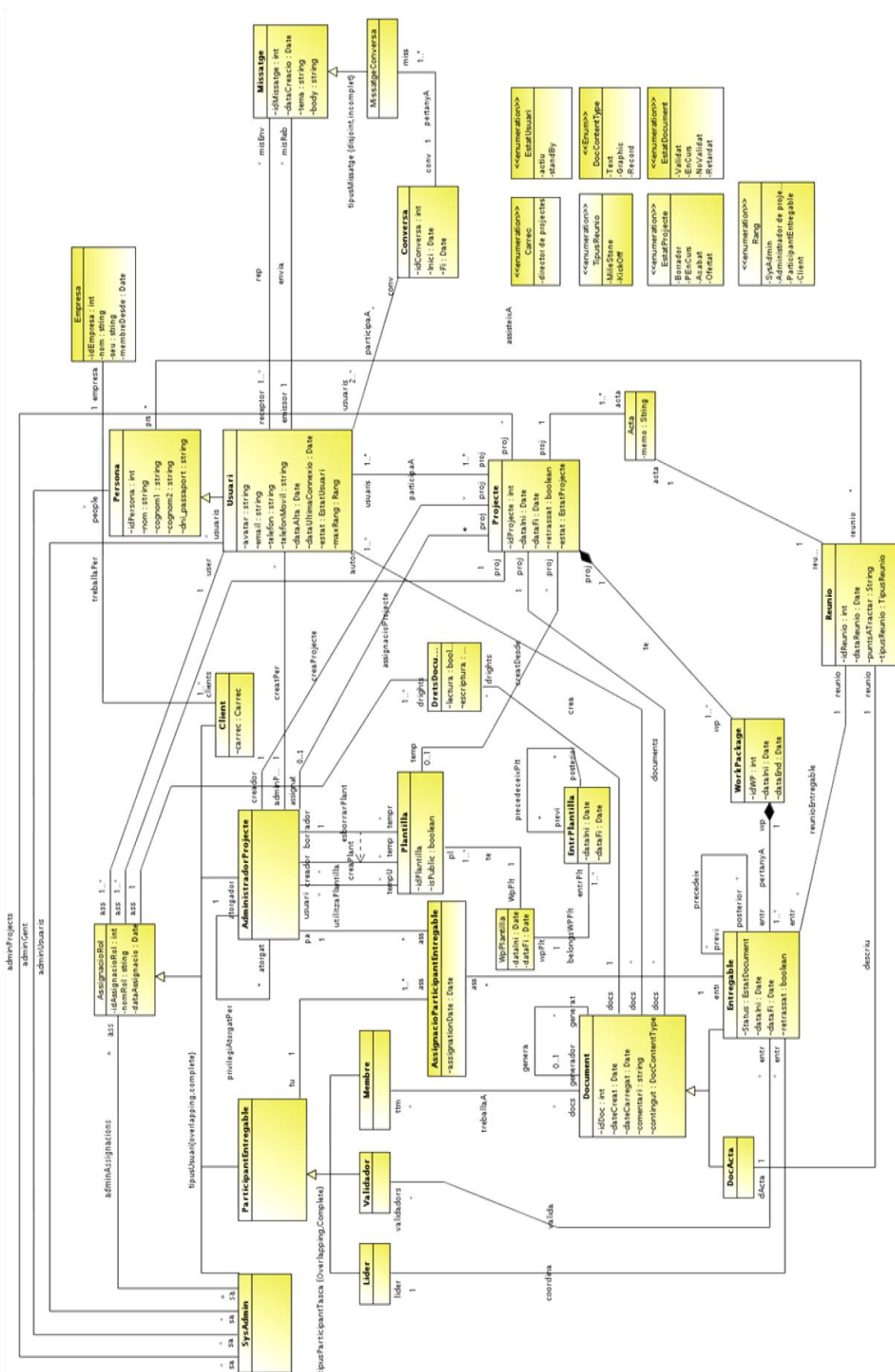


figura 55

## Aplicació de patrons de disseny

Un cop obtingut el diagrama de classes normalitzat definirem el comportament del sistema dissenyant els diagrames de seqüència que ens mostraran la interacció entre els elements que apareixen al model.

En aquesta fase aplicarem alguns dels patrons de disseny més estesos com per exemple el patró expert.

El patró expert busca bàsicament delegar responsabilitats en els objectes a l'hora de realitzar una acció però sovint aquestes responsabilitats estan disperses entre diferents objectes del model de manera que, de vegades, no sigui del tot desitjable l'aplicació del patró degut a l'acoblament que introdueix entre les classes fent-les dependents unes d'altres.

En el disseny d'un sistema software és molt desitjables mantenir com més baix millor el nivell d'acoblament entre classes de cara a facilitar aspectes com la mantenibilitat o canviabilitat, d'aquí que l'aplicació d'aquests patrons sigui quelcom que ha d'estar regit pel sentit comú per tal de no abusar-ne en excés.

Un altre patró del que també se'n farà ús és el patró creador. Aquest patró ens ajuda a l'hora de decidir quins objectes han de tenir la responsabilitat de crear altres objectes. La seva aplicació és relativament senzilla donat que està basada en l'assignació de la responsabilitat en cas de donar-se un seguit de casos clarament definits per el propi patró:

- Si l'objecte agrega objectes de la classe a crear
- Si l'objecte ja conté objectes de la classe a crear
- Si l'objecte enregistra instàncies de la classe a crear
- Si l'objecte fa un ús intensiu d'objectes del tipus de la classe a crear
- Si l'objecte és un expert respecte de la classe a crear

Si alguna de les condicions anteriors es compleix llavors podem afirmar que l'objecte és un ferm candidat a convertir-se en creador. Però de nou, hi ha ocasions en que aplicar aquest patró es contraproductiu i es requereix la creació d'una classe dedicada només a aprovisionar objectes del tipus desitjat, aquest patró és conegut amb el nom de Fàbrica.

Per últim, aplicarem un patró específic per a l'execució dels casos d'ús que el nostre sistema necessita. Per fer-ho aplicarem el patró Controlador.

El patró Controlador consisteix bàsicament en encapsular les operacions que executen un cas d'ús en particular. La tasca principal d'un controlador no és la de que estigui carregat amb responsabilitats, es tracta més aviat de que sigui l'encarregat de coordinar i controlar el flux d'execució d'un cas d'ús deixant la responsabilitat a la interacció entre les classes del model cadascuna amb la seva responsabilitats.

En el nostre cas, per evitar acoblament i assegurar modularitat, s'ha optat per la creació del 3 controladors els quals es comuniquen només entre ells de manera que assolim un baix acoblament entre els elements del domini amb altres "agents" del sistema com la capa de dades o les vistes encarregades de la interacció de l'usuari amb el sistema. Així, s'obté un disseny repartit en 3 capes on la capa superior engloba tots els aspectes relacionats amb la interfície d'usuari, la capa intermèdia encapsula tots els objectes del domini de la nostra aplicació i la lògica de negoci i, per últim, la capa més inferior que es l'encarregada de donar suport a les dades i a la persistència.

En la següent figura es mostra el disseny complet de l'aplicació agrupat en capes o s'hi pot apreciar la manera com es comuniquen entre elles mitjançant els controladors.

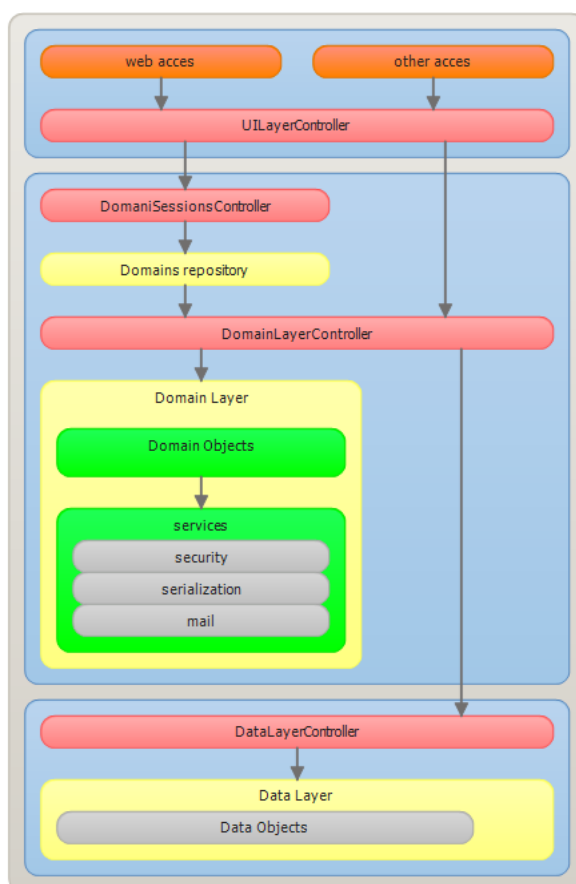


figura 56

## Diagrames de seqüència

Un cop normalitzat el sistema a continuació es mostra un seguit d'operacions disponibles en el controlador de domini on s'hi pot apreciar la interacció entre diferents objectes del model. Degut a l'elevat nombre d'operacions resultants, només és presenta un reduït nombre a mode d'exemple:

Operació: **CrearProjecte** (CrearProjecte(in dtoProj:DTOProjectDataObject):Integer)

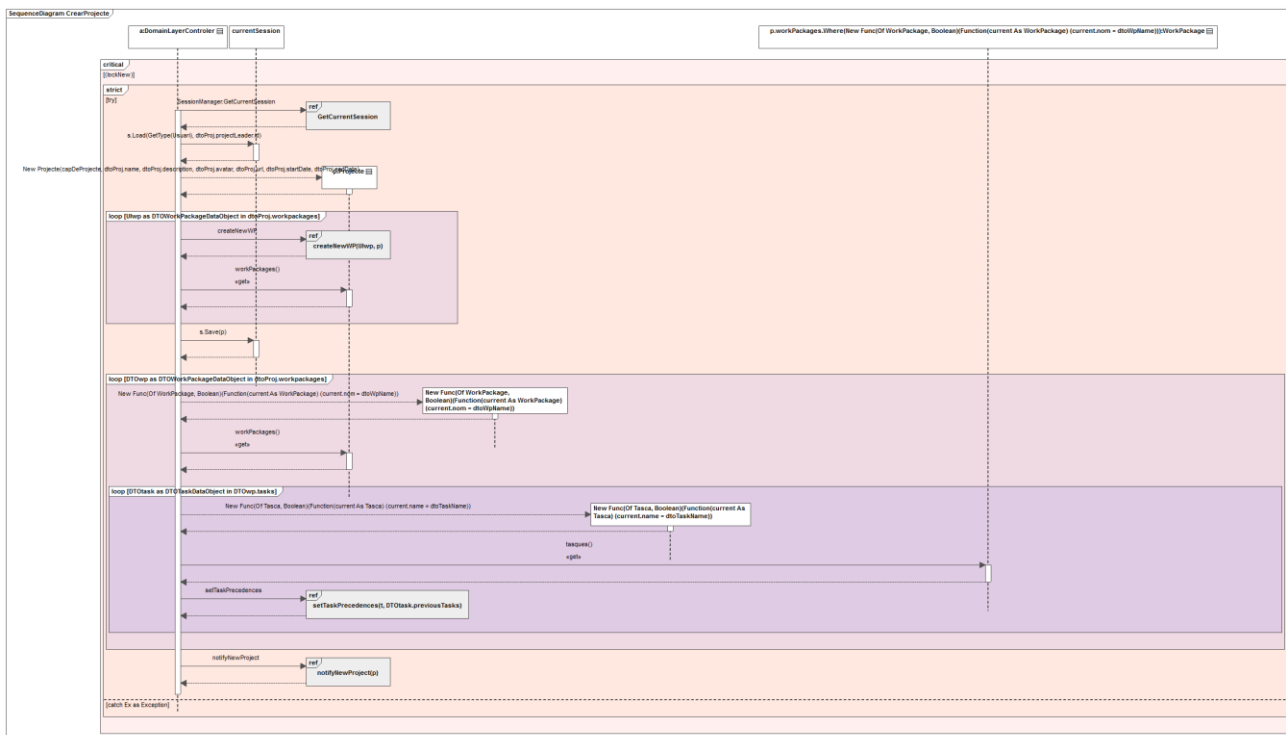


Figura 57



Operació: **createNewWP** (createNewWP(in Ulwp:DTOWorkPackageDataObject, in p:Projecte):WorkPackage)

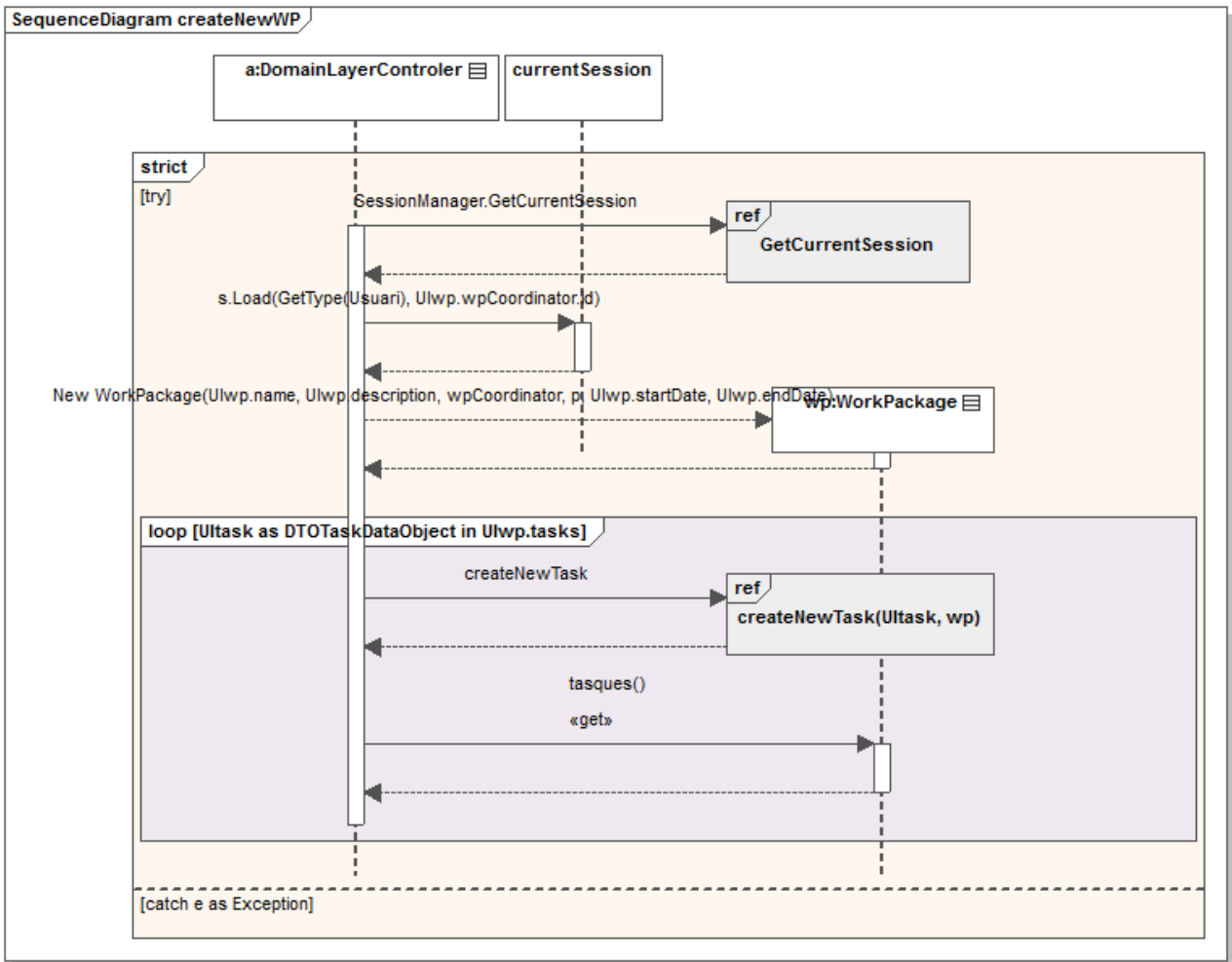


Figura 58

Operació: **createNewTask** (createNewTask(in Ultask:DTOTaskDataObject, in wp:WorkPackage):Tasca)

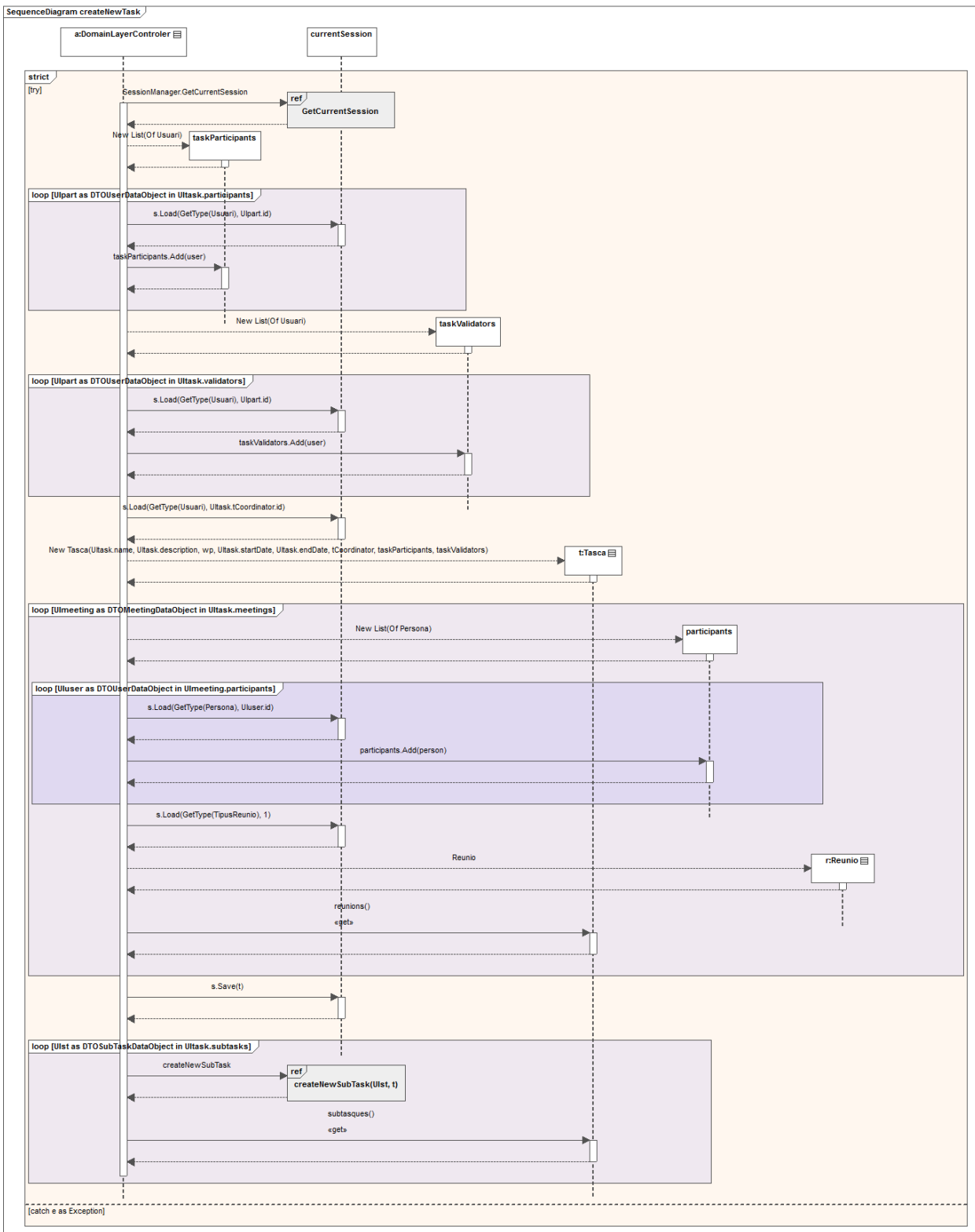


Figura 59

Operació: **createNewSubTask** (createNewSubTask(in Ulst:DTOSubTaskDataObject, in t:Tasca):Subtasca)

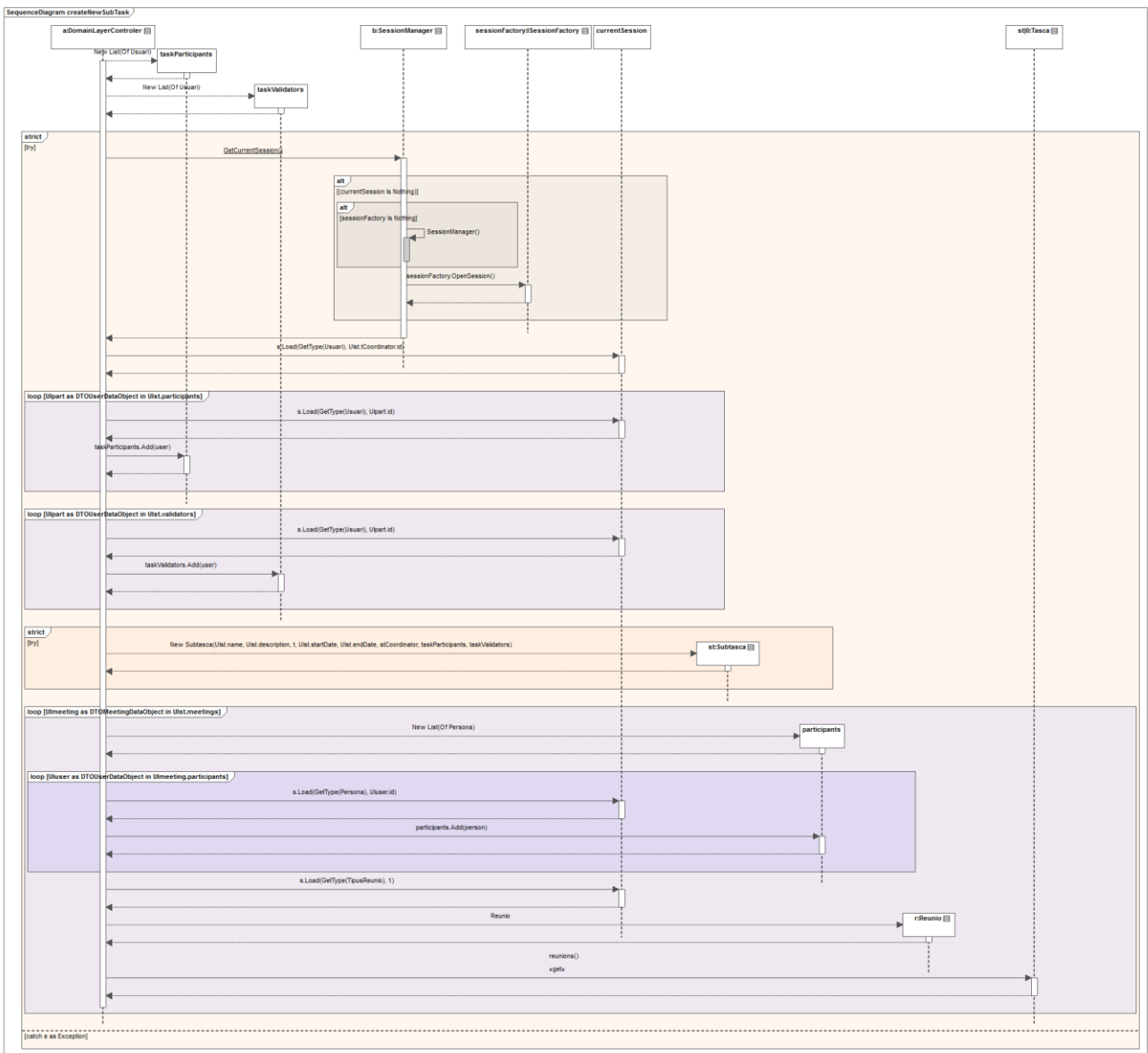


Figura 60

Operació: **getAllProjectesUsuari** (getAllProjectesUsuari():Dictionary<T1->DTOProjectDataObject,T2->List<T1->DTORoleAssignmentDataObject>>)

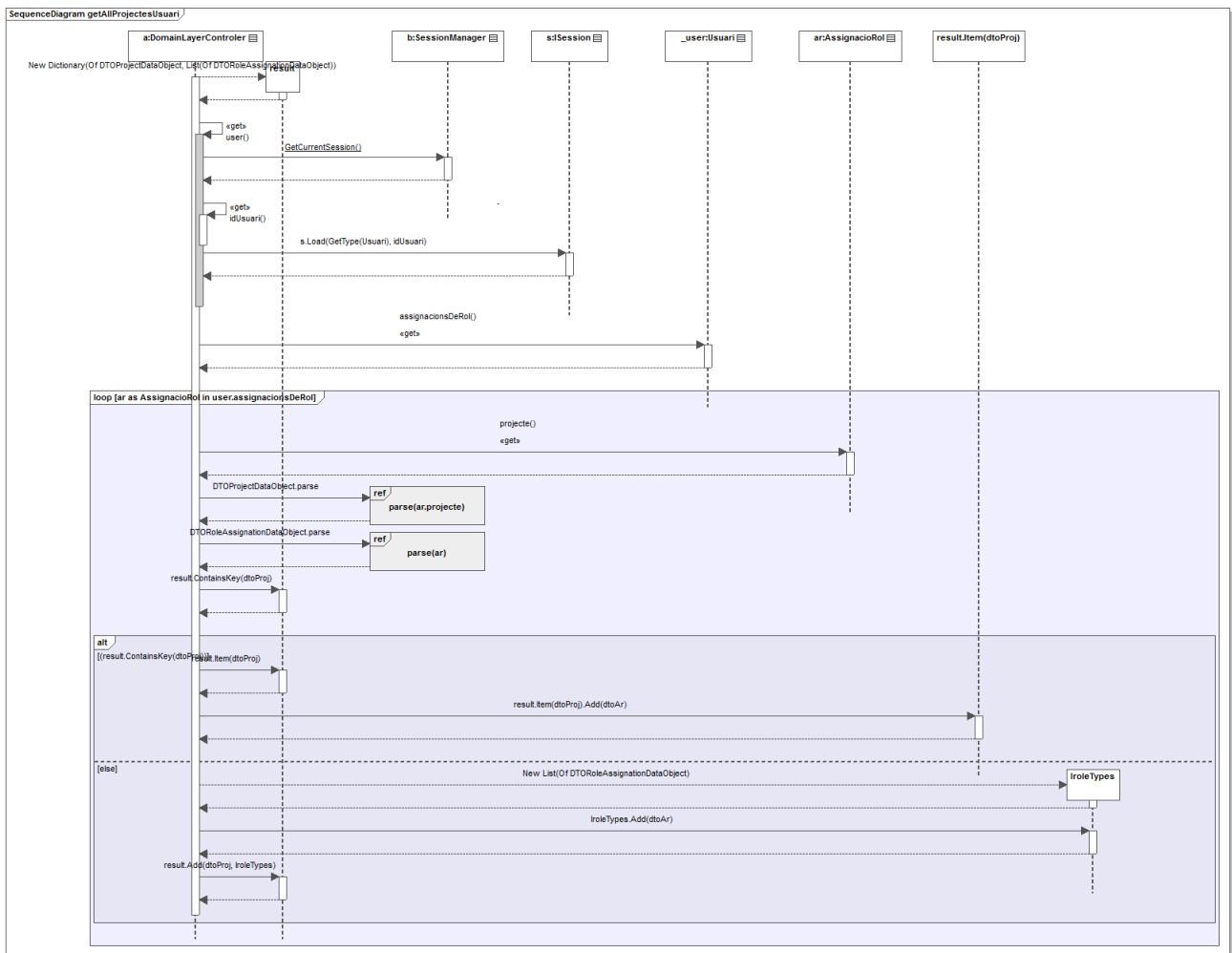


Figura 61

Operació: **checkEmailsUnique** (checkEmailsUnique(in email:String):Boolean)

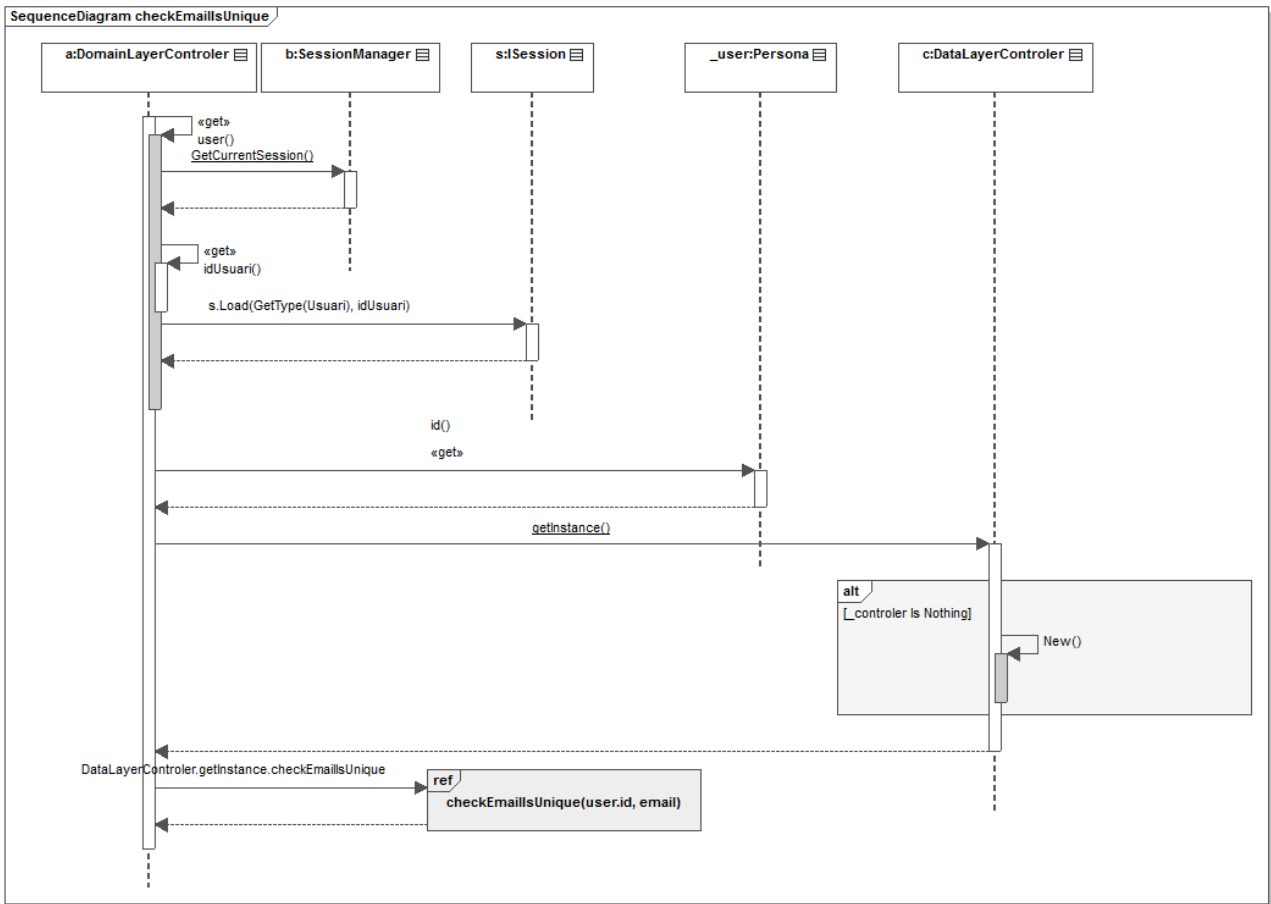


Figura 62

Operació: **deleteMail** (deleteMail(in idMissatge:Integer):Boolean)

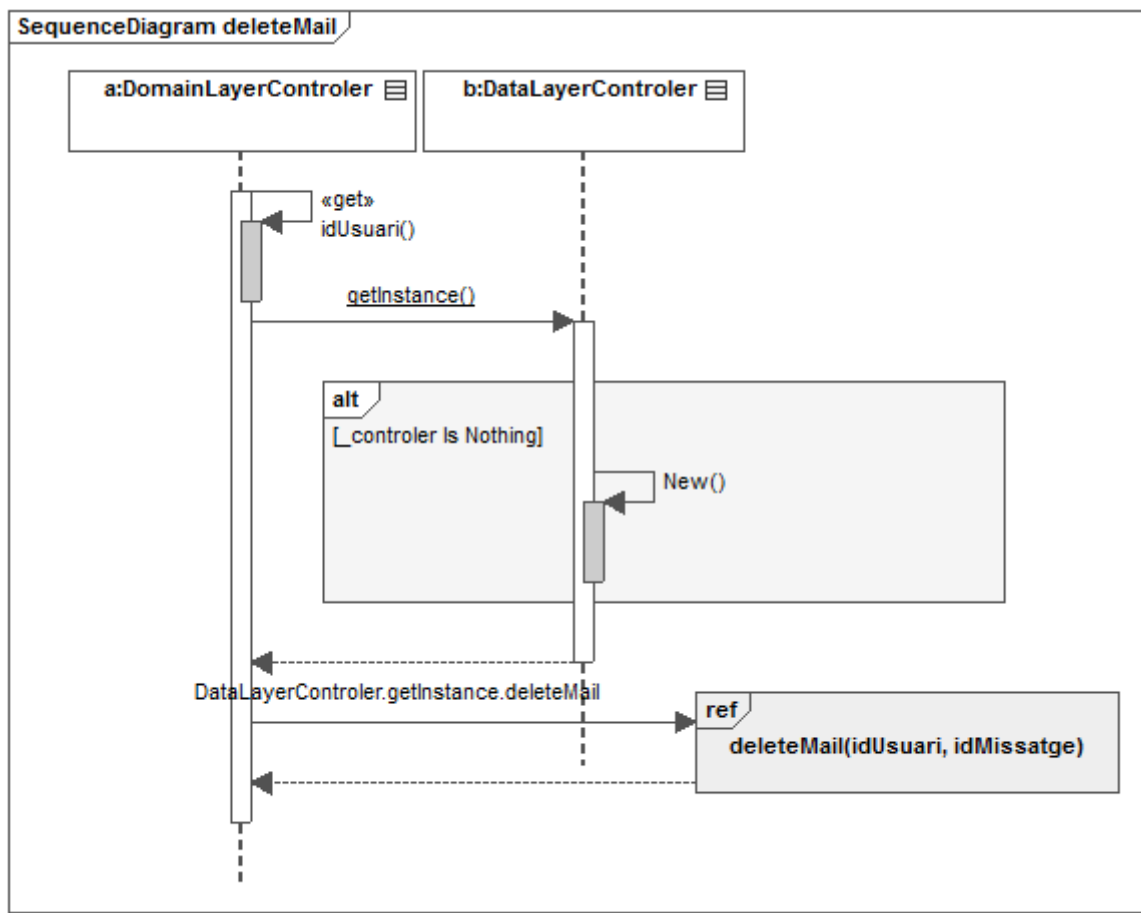


Figura 63

Operació: **deleteSendMail** (deleteSendMail(in idMail:Integer):Boolean)

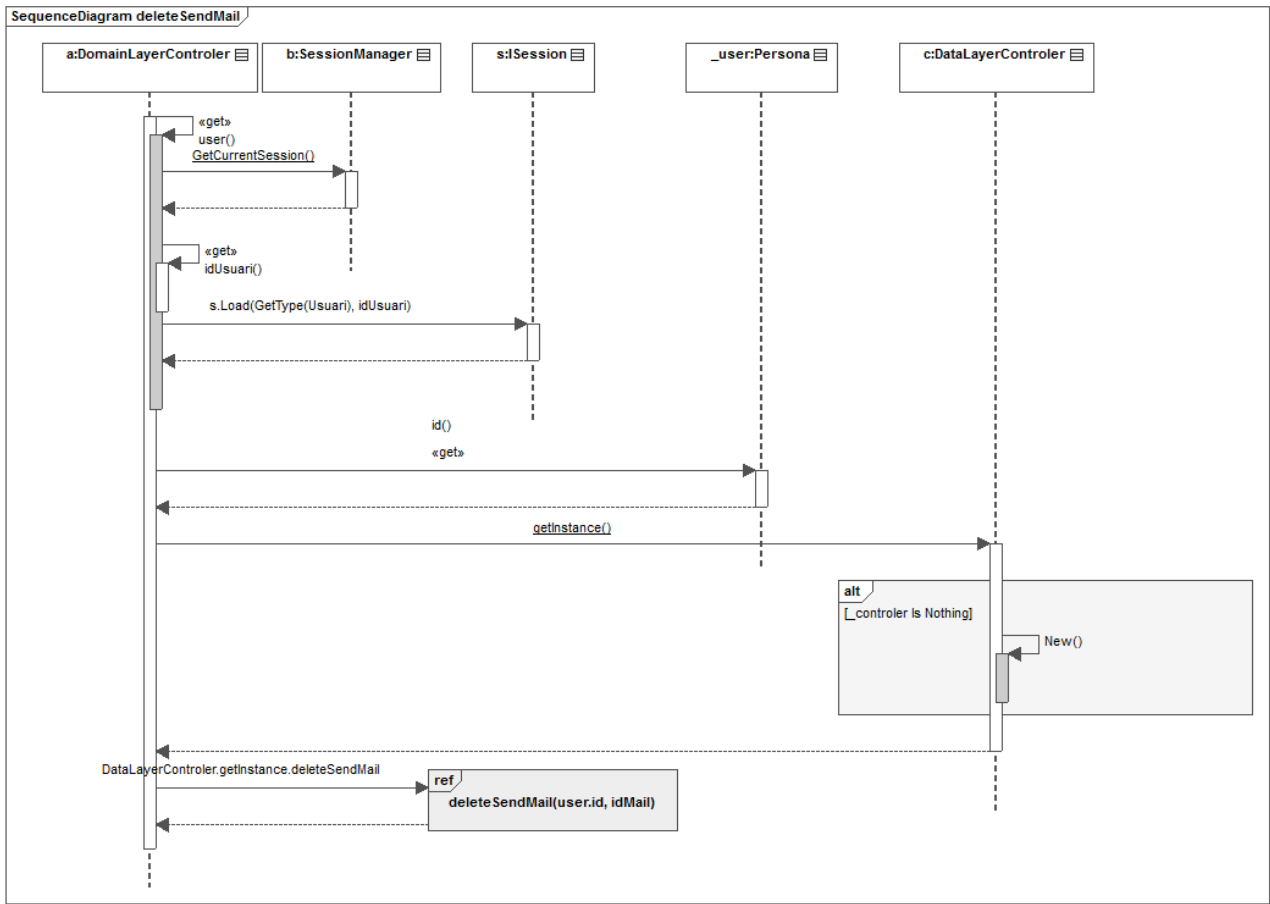


Figura 64

Operació: **deleteThreadMail** (deleteThreadMail(in idMail:Integer):Boolean)

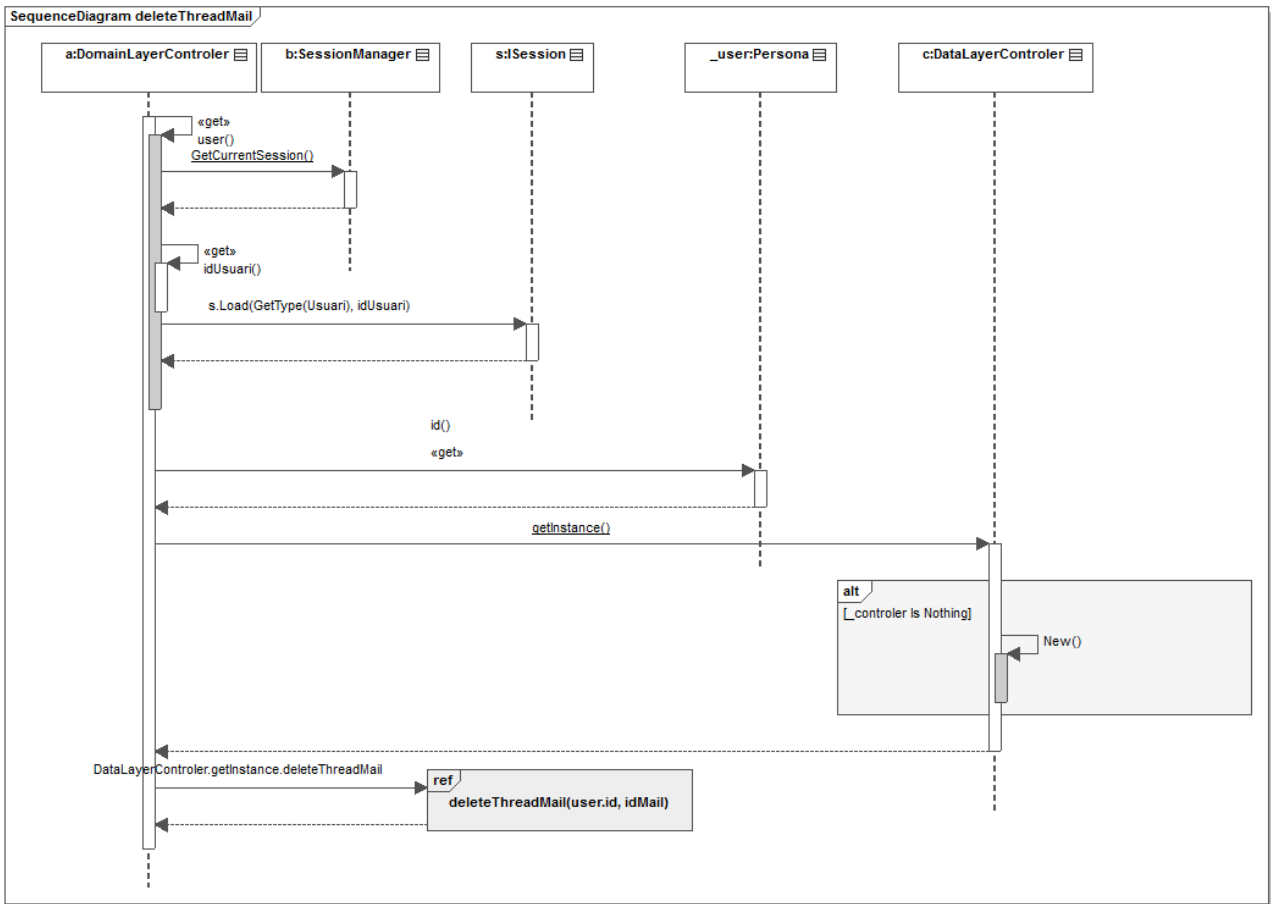


Figura 65



Operació: **getAllMessages** (getAllMessages():List<T1->DTOMessageDataObject>)

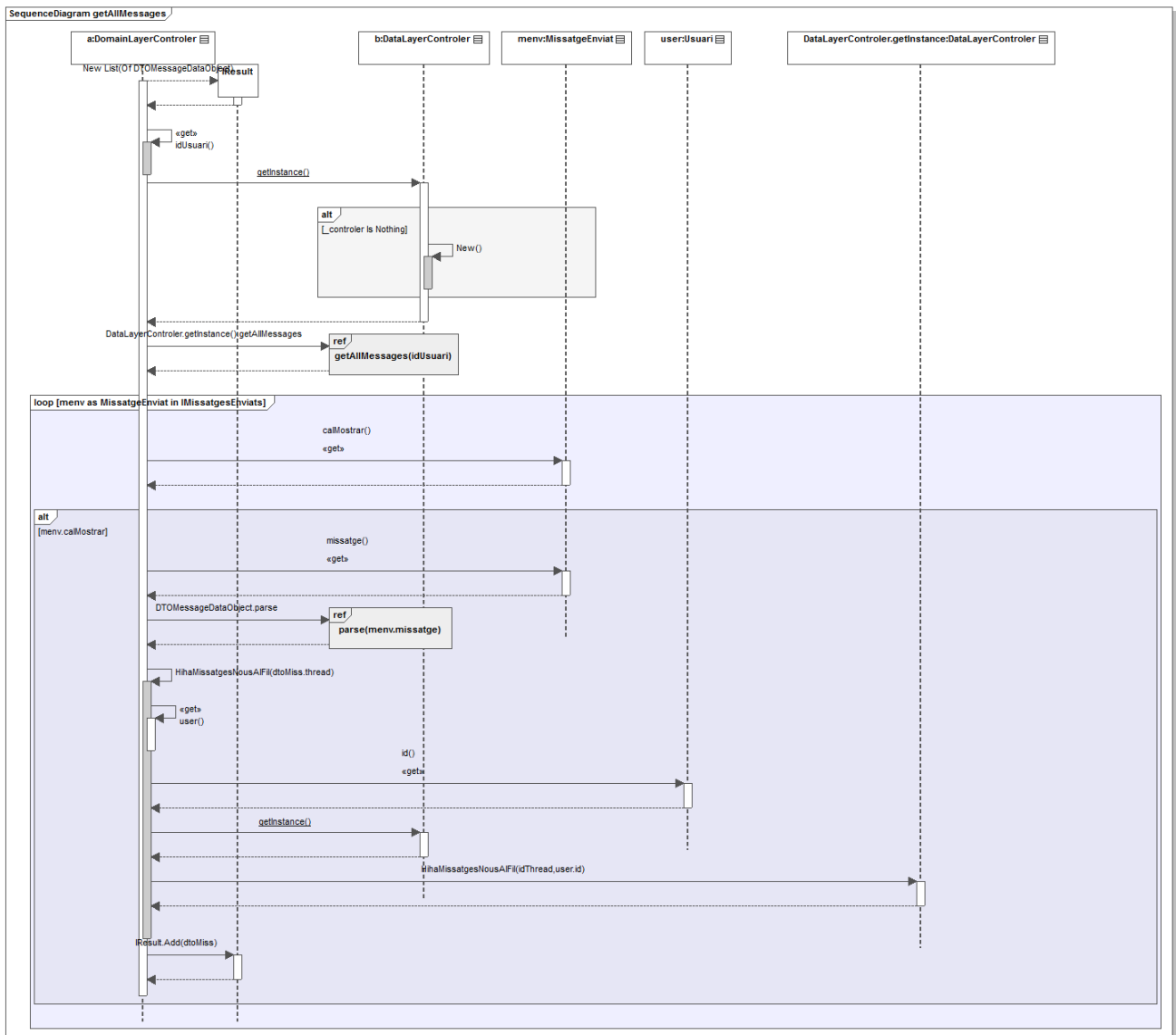


Figura 66

Operació: **getAllSendMessages** (getAllSendMessages():List<T1->DTOMessageDataObject>)

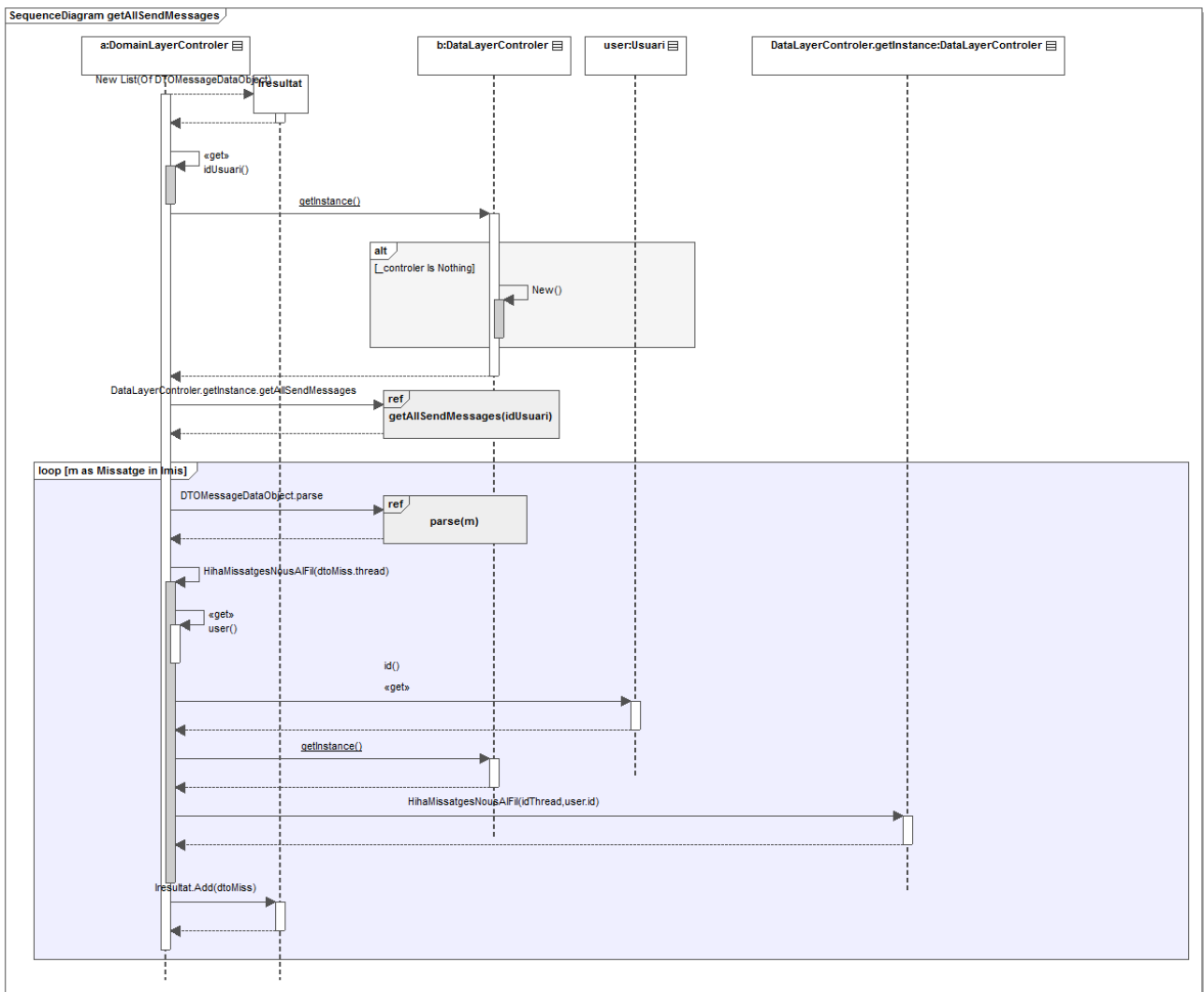


Figura 67



Operació: **getDocument** (getDocument(in idDoc:Integer, in includeContent:Boolean):DTODocumentDataObject)

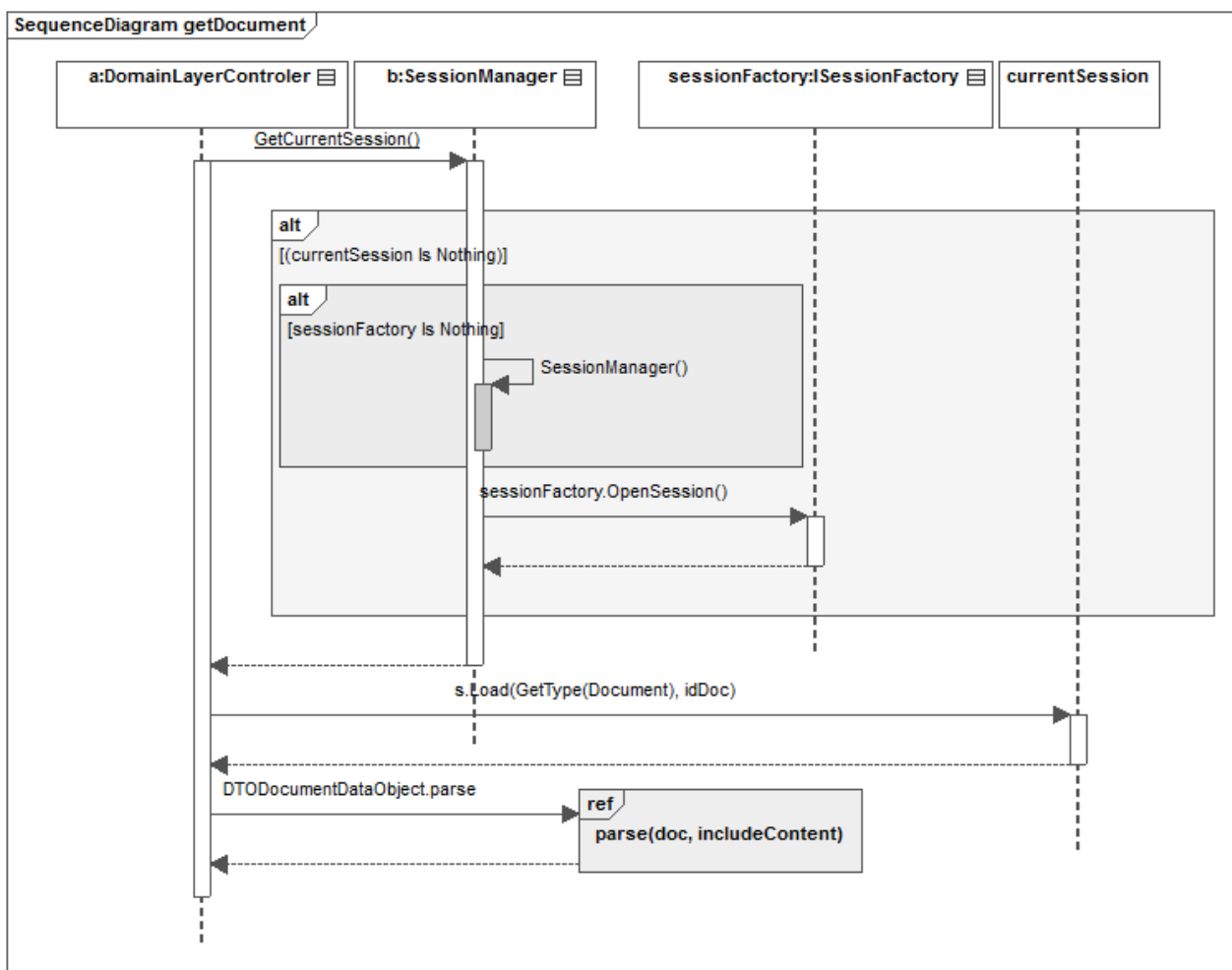


Figura 69

---

## Capítol 7

---

### Implementació del sistema

Un cop s'han completat totes les fases d'especificació i disseny del sistema el següent pas natural consisteix en la seva implementació. Aquest serà l'aspecte que es tractarà en aquest capítol.

Els principals punts a desglossar seran la tecnologia triada a l'hora de dur a terme la implementació a tots els nivells i el motiu pel qual ha estat triada. Tanmateix es donarà una visió d'aspectes com la interfície d'usuari desenvolupada i quins han estat els detalls de la seva implementació per finalment entrar a valorar el sistema gestor de base de dades encarregat de la persistència de les dades del sistema i com ha estat implementada aquesta.

### Llenguatges de programació al costat servidor

El llenguatge de programació que s'ha utilitzat per a desenvolupar el gruix del sistema al costat del servidor és Visual Basic 9. Visual Basic és un llenguatge orientat a objectes desenvolupat per l'empresa Microsoft el qual es troba integrat dins d'un IDE (o entorn de desenvolupament) conegut amb el nom de Visual Studio. La versió utilitzada en la implementació ha estat la versió 2010.

#### **Visual Basic.Net**

Es tracta d'un llenguatge de programació que necessita un entorn de execució conegut amb el nom de .Net Framework. Es tracta d'una màquina virtual que ofereix tota una sèrie de funcions que abstraen la necessitat d'atacar directament les crides a funcions del propi sistema operatiu ( de la mateixa manera com fa la màquina virtual de JAVA ).

La programació en aquest llenguatge no permet inicialment l'accés multiplataforma, tot i que, existeixen eines al mercat que són una implementació per altres sistemes operatius que sí que permeten una execució sobre altres plataformes com per exemple el projecte Mono.

### Com són executades les aplicacions?

Per respondre aquesta pregunta primer explicarem que és el Common Language Runtime (CLR). El CLR podria ser definit com el nucli del Framework .Net, ja que es tracta de l'entorn d'execució en el qual les aplicacions desenvolupades són carregades un cop s'executen.

El compilador de VB.net compila el codi font obtenint com a resultat un codi intermedi anomenat MSIL (Microsoft Intermediate Language). Aquest codi obtingut però, no és codi màquina directament executable la qual cosa exigeix un segon pas de compilació en de la que s'encarrega un segon compilador conegut amb el nom de JIT (Just-in-Time).

Un cop duta a terme la compilació del codi MSIL generat pel compilador de VB.net pel JIT, el codi obtingut és codi màquina real que ja pot ser executat sobre la plataforma o sistema operatiu que tingui la màquina.

Les avantatges obtingudes són clares. Independentment de l'eina de desenvolupament triada, només es necessari que el compilador utilitzat generi un codi MSIL vàlid i disposar d'un compilador JIT adaptat a la plataforma sobre la que es vol executar el codi per tal de que aquest pugui ser executat.

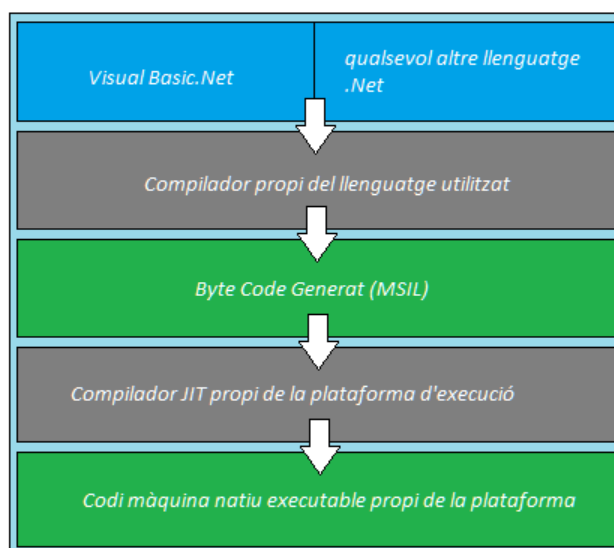


Figura 70

A la figura superior podem observar el procés de compilació d'una aplicació escrita en vb.net i executada sobre el .Net Framework.

Però per ser encara més específics cal que parlem del framework específic utilitzat per a desenvolupar aplicacions web de Microsoft i que és on realment s'emmarca el desenvolupament d'aquesta aplicació, el conegut com a ASP.Net.

### ASP.Net

ASP.Net és un framework per al desenvolupament de pàgines web dinàmiques que permet desenvolupar de manera senzilla webs. Permet la combinació de codi HTML per a la creació de la vista web (anomenada dins d'aquesta tecnologia com a formulari web o *webform*) on cadascuna d'aquestes vistes és, generalment, generada dinàmicament al servidor segons la petició de rebuda per la interacció del sistema amb l'usuari.

El formularis web consten cadascun principalment de dos fitxers que van lligats entre ells. Per una part es té la vista on s'hi pot descriure els aspectes estàtics de la pàgina i que està definit mitjançant llenguatge HTML i/o qualsevol dels llenguatges d'scripting (principalment javascript). Per altre, existeix un fitxer que conté el codi de servidor associat a aquella pàgina en particular i que conté part de la lògica a executar en el servidor quan s'interactua amb la pàgina.

En el cas del nostre sistema. Aquestes pàgines es troben encabides a la capa d'usuari. El codi de servidor que s'executa per a cadascuna d'aquestes pàgines serveix per enllaçar amb el controlador de capa que és qui fa la connexió amb el controlador de la capa homònima on es troben els objectes de domini de l'aplicació i la seva lògica de negoci.

Així doncs, si ens basem en la figura 56 i expandim la capa superior del diagrama observem com aquest fet afecta a la capa d'usuari.

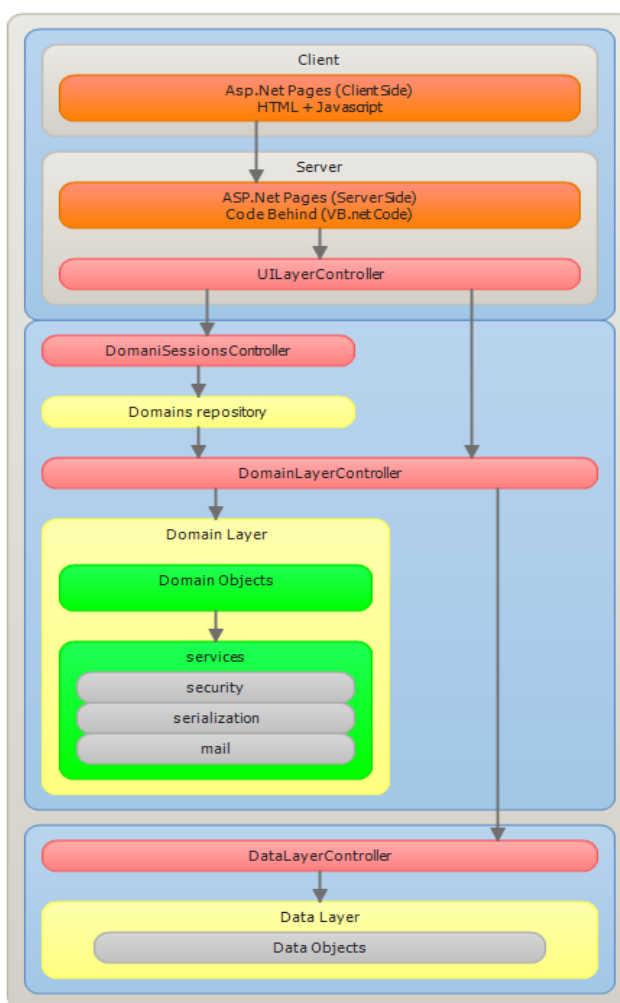


figura 71

L'accés web a l'aplicació consta d'un conjunt de pàgines asp que es troben al costat client i que es comuniquen amb la seva part corresponent al servidor. Un cop ja la petició llençada des de la part client és rebuda al servidor llavors es produeixen les interaccions necessàries cap al controlador de la capa que és l'encarregat de gestionar-les i comunicar-se amb la següent capa descarregant tota responsabilitat relacionada amb la lògica de negoci dels elements de la capa d'usuari.

Els avantatges de fer ús del framework d'ASP.net són principalment el fet del suport a nivell d'infraestructura que aporta aquest Out Of The box com per exemple:

- control d'estat de sessió.
- control d'estat de sessió a nivell d'aplicació.
- ús de tota una llibreria de controls encapsulats (taules, etc).
- controladors de captura de peticions HTTP.
- extens ús la qual cosa permet trobar molta informació amb molta facilitat.

### ***Alternatives a VisualBasic.Net***

Existeixen al mercat molts altres llenguatges de "servidor" que podrien haver estat triats per a implementar el sistema. Per exemple jsp (JAVA) o php entre d'altres. Tots ells aporten característiques similars que els fan igualment vàlids a l'hora de seleccionar-los per a implementar el sistema.

Tant PHP com JSP són tecnologies àmpliament esteses i que aconsegueixen els requisits demandats a l'hora d'especificar l'aplicació (de fet ambdós són d'ús gratuït cosa que els fa encara més atractius) però finalment es va optar per fer servir tecnologia .Net per dues grans raons. La primera pel fet de tractar-se d'una tecnologia amb la qual estava familiaritzat a l'hora de treballar i la segona i més important per ser una imposició definida per part del client des de bon inici del projecte.



## Implementació de la persistència del sistema

### **Persistència de dades amb NHibernate**

Per simplificar les operacions de persistència de dades s'ha optat per fer servir un ORM (Object-Relational Mapping). NHibernate és una versió portada al framework .Net des de JAVA on és una solució llargament consolidada, motiu pel qual ha estat triat com a solució a aplicar en el nostre cas.

El fet d'utilitzar un ORM com NHibernate ens permet abstraure'ns a de les dades i centrar-nos a l'hora de desenvolupar tan sols en el nostre domini d'objectes.

### **Sistema Gestor de Base de Dades (SQL Server 2005)**

Tot i això, és necessari implementar una base de dades que contingui totes les entitats del sistema que cal que siguin persistents. Per fer-ho hem triat com a sistema gestor de bases de dades el producte de Microsoft per a tal efecte. La raó d'haver triat aquest i no altre es el fet de que és una eina amb la qual estic familiaritzat de manera que no existeix una corba d'aprenentatge que pugui endarrerir el projecte en aquest punt a més de ser un requeriment per part del client de fer servir tecnologies d'aquesta empresa allà on sigui possible. El producte en qüestió es Microsoft SQL server 2005.

### **Implementació de la base de dades**

A partir d'aquest software ha estat generat l'esquema de la base de dades. S'han generat un total de 57 taules les qual es descriuran a continuació:

NOM TAULA	CAMPS	TIPUS	PK
<b>AdministradorProjecte</b>	idAdminProj	<i>int</i>	<b>X</b>
	idAtorgador	<i>int</i>	
<b>AssignacioParticipantTasca</b>	idAssignacioParticipantTasca	<i>int</i>	<b>X</b>
	idTascaAssignada	<i>int</i>	
	idDataAssignacio	<i>int</i>	
	idParticipantTasca	<i>int</i>	
<b>AssignacioRol</b>	idAssignacioRol	<i>int</i>	<b>X</b>
	idNomRol	<i>int</i>	
	idUsuari	<i>int</i>	
	idProjecte	<i>int</i>	
	idDataAssignacio	<i>int</i>	
<b>AssistentReunio</b>	idAssistentReunio	<i>int</i>	<b>X</b>

	idPersona	int	
	idReunio	int	
<b>CarrecClient</b>	idCarrec	int	<b>X</b>
	nomCarrec	varchar	
<b>Client</b>	idClient	int	<b>X</b>
	idEmpresa	int	
<b>ClientsAEmpreses</b>	idClientAEmpresa	int	<b>X</b>
	idCarrec	int	
	idClient	int	
	idEmpresa	int	
	dataAlta	nchar	
<b>Conversa</b>	idConversa	int	<b>X</b>
	inici	datetime	
	fi	datetime	
<b>CreadorDocument</b>	idCreadorDocument	int	<b>X</b>
	idUsuariCreador	int	
	idDocument	int	
<b>DocActa</b>	idDocActa	int	<b>X</b>
	idReunio	int	
<b>Document</b>	idDoc	int	<b>X</b>
	nom	varchar	
	dataCreat	datetime	
	dataCarregat	datetime	
	comentari	nvarchar	
	contingut	int	
	idProj	int	
	idDocGenerador	int	
	idWP	int	
	idFile	int	
	idUsuariCarregador	int	
<b>DretsDocuments</b>	idDretsDocument	int	<b>X</b>
	idDocument	int	
	idAssignacioRol	int	
	Lectura	bit	
	Esriptura	bit	
<b>Empresa</b>	idEmpresa	int	<b>X</b>
	nom	varchar	
	seu	varchar	
	membreDesde	datetime	
<b>Entregable</b>	idEntregable	int	<b>X</b>
	retrassat	bit	

	EstatEntregable	<i>int</i>	
	idLider	<i>int</i>	
	idTasca	<i>int</i>	
<b>EntrPlantilla</b>	idEntrPlantilla	<i>int</i>	<b>X</b>
	dataIni	<i>datetime</i>	
	dataFi	<i>datetime</i>	
	idWpPlt	<i>int</i>	
<b>EstatsEntregable</b>	idEstatEntregable	<i>int</i>	<b>X</b>
	nom	<i>varchar</i>	
<b>EstatsProjecte</b>	idEstatProjecte	<i>int</i>	<b>X</b>
	nom	<i>varchar</i>	
<b>EstatsUsuari</b>	idEstatUsuari	<i>int</i>	<b>X</b>
	Estat	<i>varchar</i>	
<b>FinalProjecte</b>	idFinalProjecte	<i>int</i>	<b>X</b>
	idProjecte	<i>int</i>	
	idData	<i>int</i>	
	idDataCanvi	<i>int</i>	
<b>FinalTasca</b>	idFinalTasca	<i>int</i>	<b>X</b>
	idTasca	<i>int</i>	
	idData	<i>int</i>	
	idDataCanvi	<i>int</i>	
<b>FinalWP</b>	idFinalWP	<i>int</i>	<b>X</b>
	idWp	<i>int</i>	
	idData	<i>int</i>	
	idDataCanvi	<i>int</i>	
<b>Fitxer</b>	idFile	<i>int</i>	<b>X</b>
	idDoc	<i>int</i>	
	fileName	<i>varchar</i>	
	fileSize	<i>bigint</i>	
	contentType	<i>varchar</i>	
	SHA256	<i>varbinary</i>	
	fileContent	<i>varbinary</i>	
<b>IniciProjecte</b>	idIniciProjecte	<i>int</i>	<b>X</b>
	idProjecte	<i>int</i>	
	idData	<i>int</i>	
	idDataCanvi	<i>int</i>	
<b>IniciTasca</b>	idIniciTasca	<i>int</i>	<b>X</b>
	idTasca	<i>int</i>	
	idData	<i>int</i>	
	idDataCanvi	<i>int</i>	
<b>IniciWP</b>	idIniciWP	<i>int</i>	<b>X</b>

	idWP	<i>int</i>	
	idData	<i>int</i>	
	idDataCanvi	<i>int</i>	
<b>Missatge</b>	idMissatge	<i>int</i>	<b>X</b>
	dataCreacio	<i>datetime</i>	
	tema	<i>varchar</i>	
	bodyText	<i>varchar</i>	
	bodyHTML	<i>varchar</i>	
	idConversa	<i>int</i>	
	esMail	<i>bit</i>	
	idMissatgeAnterior	<i>int</i>	
	idEmissor	<i>int</i>	
	EmissorHaBorrat	<i>bit</i>	
	thread	<i>int</i>	
	<b>MissatgeConversa</b>	idMissatgeConv	<i>int</i>
idConversa		<i>int</i>	
<b>MissatgesEnviats</b>	idMissatgeEnviat	<i>int</i>	<b>X</b>
	idUsuariDestinatari	<i>int</i>	
	idMissatge	<i>int</i>	
	llegit	<i>int</i>	
	calMostrar	<i>bit</i>	
	dataEnviat	<i>datetime</i>	
<b>NomRols</b>	idRol	<i>int</i>	<b>X</b>
	Rol	<i>varchar</i>	
<b>ParticipacionsAConverses</b>	idParticipacionsAConverses	<i>int</i>	<b>X</b>
	idUsuari	<i>int</i>	
	idConversa	<i>int</i>	
<b>ParticipantTasca</b>	idParticipantTasca	<i>int</i>	<b>X</b>
<b>Persona</b>	idPersona	<i>int</i>	<b>X</b>
	nom	<i>varchar</i>	
	cognom1	<i>varchar</i>	
	cognom2	<i>varchar</i>	
	dni_passaport	<i>varchar</i>	
<b>Plantilla</b>	idPlantilla	<i>int</i>	<b>X</b>
	nom	<i>varchar</i>	
	publica	<i>bit</i>	
	operativa	<i>bit</i>	
	idCreador	<i>int</i>	
<b>PrecedenciesEntregablesPlantilla</b>	idPrecedenciaEntregablePlantilla	<i>int</i>	<b>X</b>
	idEntregablePreviPlt	<i>int</i>	
	idEntregablePostPlt	<i>int</i>	

<b>PrecedenciesTasques</b>	idPrecTasca	<i>int</i>	<b>X</b>
	idTascaPrevia	<i>int</i>	
	idTascaPost	<i>int</i>	
<b>Projecte</b>	idProjecte	<i>int</i>	<b>X</b>
	nom	<i>varchar</i>	
	descripcio	<i>text</i>	
	logo	<i>varchar</i>	
	website	<i>varchar</i>	
	retrassat	<i>bit</i>	
	idEstat	<i>int</i>	
	idAdminProjCreador	<i>int</i>	
	idAdminProjAssignat	<i>int</i>	
	idPlantilla	<i>int</i>	
<b>Reunio</b>	idReunio	<i>int</i>	<b>X</b>
	nom	<i>varchar</i>	
	descripcio	<i>varchar</i>	
	idDate	<i>int</i>	
	tipusReunio	<i>int</i>	
	idTasca	<i>int</i>	
	idProj	<i>int</i>	
	localitzacio	<i>varchar</i>	
<b>Subtasca</b>	idSubtasca	<i>int</i>	<b>X</b>
	idTasca	<i>int</i>	
<b>SysAdmin</b>	idSysAdmin	<i>int</i>	<b>X</b>
<b>SysAdminAdministracioAssignacions</b>	idSysAdminAssignacio	<i>int</i>	<b>X</b>
	idSysAdmin	<i>int</i>	
	idAssignacioRol	<i>int</i>	
	dataModificacions	<i>int</i>	
<b>SysAdminAdministracioPersones</b>	idSysAdminPersona	<i>int</i>	<b>X</b>
	idSysAdmin	<i>int</i>	
	idPersona	<i>int</i>	
	dataModificacions	<i>datetime</i>	
<b>SysAdminAdministracioProjectes</b>	idSysAdminProjecte	<i>int</i>	<b>X</b>
	idSysAdmin	<i>int</i>	
	idProj	<i>int</i>	
	dataModificacions	<i>datetime</i>	
<b>SysAdminAdministracioUsuaris</b>	idSysAdminUsuari	<i>int</i>	<b>X</b>
	idSysAdmin	<i>int</i>	
	idUsuari	<i>int</i>	
	dataModificacions	<i>datetime</i>	

<b>SystemDate</b>	idSystemData	<i>int</i>	
	Valor	<i>datetime</i>	
<b>Tasca</b>	idTasca	<i>int</i>	<b>X</b>
	idTascaPrevia	<i>int</i>	
	idWp	<i>int</i>	
	descripcio	<i>varchar</i>	
	name	<i>varchar</i>	
	idTaskCoordinator	<i>int</i>	
<b>TascaPrevia</b>	idPreviousTask	<i>int</i>	<b>X</b>
	idTasca	<i>int</i>	
	idTascaPrevia	<i>int</i>	
<b>TaskCoordinator</b>	idCoordinadorTasca	<i>int</i>	<b>X</b>
<b>TaskValidator</b>	idValidadorTasca	<i>int</i>	<b>X</b>
<b>TasquesCoordinades</b>	idTascaCoordinada	<i>int</i>	<b>X</b>
	idCoordinador	<i>int</i>	
	idTasca	<i>int</i>	
<b>TemesReunio</b>	idTema	<i>int</i>	<b>X</b>
	Tema	<i>varchar</i>	
	idReunio	<i>int</i>	
<b>TipusContingutDocument</b>	idTipus	<i>int</i>	<b>X</b>
	Tipus	<i>varchar</i>	
<b>TipusReunio</b>	idTipusReunio	<i>int</i>	<b>X</b>
	Tipus	<i>varchar</i>	
<b>Usuari</b>	idUsuari	<i>int</i>	<b>X</b>
	userName	<i>varchar</i>	
	password	<i>varchar</i>	
	avatar	<i>varchar</i>	
	email	<i>varchar</i>	
	telefon	<i>varchar</i>	
	telefonMovil	<i>varchar</i>	
	dataAlta	<i>datetime</i>	
	dataUltimaConnexio	<i>datetime</i>	
	creatPer	<i>int</i>	
	idEstat	<i>int</i>	
<b>UtilitzacioPlantilla</b>	idUtilitzacioPlantilla	<i>int</i>	<b>X</b>
	idPlantilla	<i>int</i>	
	idAdminProj	<i>int</i>	
	dataUs	<i>datetime</i>	
<b>ValidadorsTasques</b>	idValidacioTasca	<i>int</i>	<b>X</b>
	idValidador	<i>int</i>	
	idTasca	<i>int</i>	

<b>ValoracionsTasques</b>	idValoracioTasca	<i>int</i>	<b>X</b>
	nota	<i>int</i>	
	comentari	<i>varchar</i>	
	idDataValoracio	<i>int</i>	
	idValidadorTasca	<i>int</i>	
<b>WorkPackage</b>	idWP	<i>int</i>	<b>X</b>
	nom	<i>varchar</i>	
	descripcio	<i>text</i>	
	idProj	<i>int</i>	
	idCoordinator	<i>int</i>	
<b>WpPlantilla</b>	idWpPlt	<i>int</i>	<b>X</b>
	dataIni	<i>datetime</i>	
	dataFi	<i>datetime</i>	
	idPlantilla	<i>int</i>	

**Taula 2**

A la taula 2 s'hi mostren totes les taules que formen part del sistema així com el camps que formen part de cadascuna.

Com ja s'ha dit, l'ús del framework ASP.net aporta la possibilitat de fer servir un ampli nombre de controls, ja encapsulats, que ofereixen molta potència des de bon inici (la qual cosa pot resultar molt atractiva a priori). Però el funcionament de la majoria d'aquests controls està molt limitada permetent fer tant sols allò pel que han estat dissenyats sense donar massa marge de maniobra a l'hora de fer-ne ús.

A més, la gran majoria no exploten les tècniques de programació asíncrona tant àmpliament estesa avui dia a les grans aplicacions web disponibles al mercat com GMAIL o FACEBOOK.

Per aquest motiu es va optar per cercar una via alternativa a l'hora d'implementar la capa d'usuari al costat client de l'aplicació. Els resultats obtinguts han estat molt satisfactoris però no exempts d'un treball extra al ser necessari programar des de zero un gran nombre de controls vistes.

## Llenguatges de programació al costat client

Els llenguatges utilitzats per a desenvolupar tota la interfície d'usuari al costat client han estat HTML i javascript.

### **HTML**

HTML (**HyperText Markup Language**) és un llenguatge estàtic basat en etiquetes que és utilitzat per descriure l'estructura i el contingut en format de text. Mitjançant fulls d'estil hom pot definir l'aparença d'una pàgina web. L'HTML és el llenguatge que un navegador web interpreta i, a partir del qual, renderitza les pàgines web.

### **Javascript**

Javascript per la seva banda és un llenguatge d'scripting interpretat que, generalment, s'utilitza per dotar les pàgines web de dinamisme sense necessitat d'interactuar amb el servidor disminueixen així la quantitat de tràfic que genera l'ús de l'aplicació.

Es tracta d'un llenguatge de sintaxi molt similar a JAVA. Les seves característiques principals són:

- implementat per tots els navegadors moderns
- permet orientació a objectes
- dèbilment tipat

Però sens dubte la principal característica de javascript és la capacitat que ens dona de poder comunicar-nos de manera asíncrona amb el servidor sense necessitat de recarregar tot el contingut d'una pàgina. Aquesta tècnica de comunicació és coneguda amb el nom d'AJAX.

**AJAX** (Asynchronous Javascript and XML) és una tècnica de comunicació entre client-servidor en un context web on la comunicació amb el servidor es duta a terme en un segon pla sense interferir en la vista. D'aquesta manera aconseguim augmentar les pàgines d'interactivitat, velocitat i aconseguir una experiència d'usuari més propera a la d'una aplicació típica d'escriptori. Alguns exemples d'aquest comportament els trobem a aplicacions com **Gmail** o **Facebook** on es busca minimitzar el nombre de recàrregues de pàgina tant com sigui possible.



La última tecnologia la que hem fet ús és **Silverlight**. Es tracta d'un model de programació per desenvolupar el que es coneix com RIA (Rich Internet Applications) desenvolupat per Microsoft.

La idea principal és la de desenvolupar un petit programa que un cop desplegat s'executa en el client tenint com a contenidor la pròpia pàgina on es hostatjat. Internament està programat en VB.net i ens ofereix un subconjunt del framework .net que es troba disponible gràcies a la instal·lació d'un plugin multinavegador i multiplataforma.

Una alternativa a la utilització de Silverlight és Flash, tecnologia desenvolupada per Adobe que ha estat desestimada per evitar introduir un nou llenguatge.

Tot i permetre desenvolupar una interfície gràfica, en el nostre cas no en farem ús ja que tota la interfície es construirà sobre HTML + javascript.

De tot el ventall de possibilitats que ofereix Silverlight, explotarem la possibilitat d'establir un socket des d'un client silverlight cap a un servidor desenvolupat en vb.net. Això permetrà la comunicació bidireccional entre servidor i client permetent fer push de dades de manera asíncrona entre servidor-client sense necessitat d'iniciar la comunicació per part del client. Més endavant en detallarem el seu ús quan expliquem en profunditat la implementació del xat.

## Format per l'intercanvi de dades

L'intercanvi de dades entre client-servidor normalment utilitza XML tot i que val a dir que cada cop és més comú fer servir com a format d'intercanvi JSON (tal i com s'ha implementat en el nostre cas).

**JSON** (Javascript Object Notation) és un format d'intercanvi de dades basat en text pla on les dades són definides entre claus i per parells clau-valor podent definir vectors de valors. És important fer notar que JSON no accepta definicions d'estructures de dades recursives.

Així doncs feta la introducció de les tecnologies que apareixen a la implementació a continuació en desglossarem el seu ús veient-ne alguns exemples sobre la pròpia implementació realitzada.

## Implementació de controls utilitzant jQuery i HTML

Per tal de minimitzar la recàrrega de pàgines a l'aplicació, s'ha optat per desenvolupar una sèrie de controls o widgets reutilitzables al llarg de l'aplicació mitjançant javascript que interactuïn amb el servidor realitzant crides en segon pla.

Per simplificar la implementació amb javascript hem fet ús d'una llibreria àmpliament utilitzada anomenada jQuery que facilita enormement aspectes com la manipulació d'elements HTML de la pàgina així com la comunicació assíncrona amb el servidor.

jQuery és una llibreria que inclou un elevadíssim nombre de funcions que permeten manipular per complert l'estructura d'una pàgina HTML així com afegir-hi tot tipus d'animacions. A més a més incorpora tot tipus de controladors d'esdeveniments que permeten de manera molt senzilla definir comportaments i millorar moltíssim la interacció dels usuaris amb les pàgines web que visiten.

Per utilitzar-ho només és necessari descarregar-se un únic fitxer d' aproximadament 25 kb i afegir-hi una referència des de la pàgina on vulguem fer-ne ús.

De totes les llibreries javascript que hi ha al mercat (Mootools, Scriptaculous, etc) jQuery s'ha convertit en el més estès gràcies a la seva facilitat d'ús. Tant important és la seva presència que Microsoft ha decidit incorporar-ho per defecte als projectes d'aplicació web que són creats des de Visual Studio motiu pel qual nosaltres el fem servir.

## Components i widgets implementats a la interfície d'usuari

### Control de selecció d'usuaris



figura 72

El control de selecció d'usuaris ha estat desenvolupat com un widget el qual pot ser fàcilment instanciat en qualsevol moment i lloc dins de les diferents vistes que componen l'aplicació.

Un cop instanciat l'objecte Javascript corresponent, automàticament es genera una vista com la que ens mostra la figura 72. En ella l'usuari pot clicar al damunt de qualsevol quadre d'usuari per tal de seleccionar-lo o deseleccionar-lo. el widget ofereix un mètode que retorna uns identificadors d'aquells ítems seleccionats per l'usuari.

El funcionament intern d'aquest control o widget és força senzill. Un cop s'inicialitza en el client mitjançant Javascript es produeix un crida en segon pla que recupera la informació necessària dels usuaris a mostrar i les seves imatges de manera que la seva adquisició des del servidor resulta transparent per l'usuari evitant un recàrrega de pàgina innecessària.

## Control autocompletar noms d'usuari

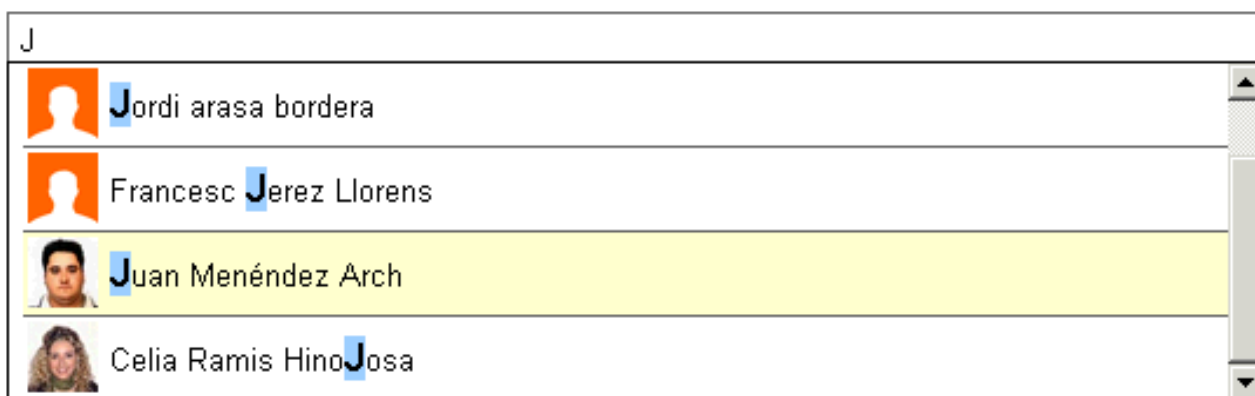


figura 73

El control autocompletar ha estat desenvolupat com un widget el qual pot ser fàcilment instanciat en qualsevol moment i lloc dins de les diferents vistes que componen l'aplicació.

Un cop instanciat l'objecte Javascript corresponent, automàticament es genera una vista com la que ens mostra la figura 73. A cada pulsació de l'usuari, el sistema detecta el caràcter i fa un cerca en segon pla al servidor de la paraula parcial escrita i ofereix un conjunt de resultats entre els qual l'usuari pot navegar amb les tecles de cursor o bé clicar amb el ratolí.

## Finestra de conversa a través del xat

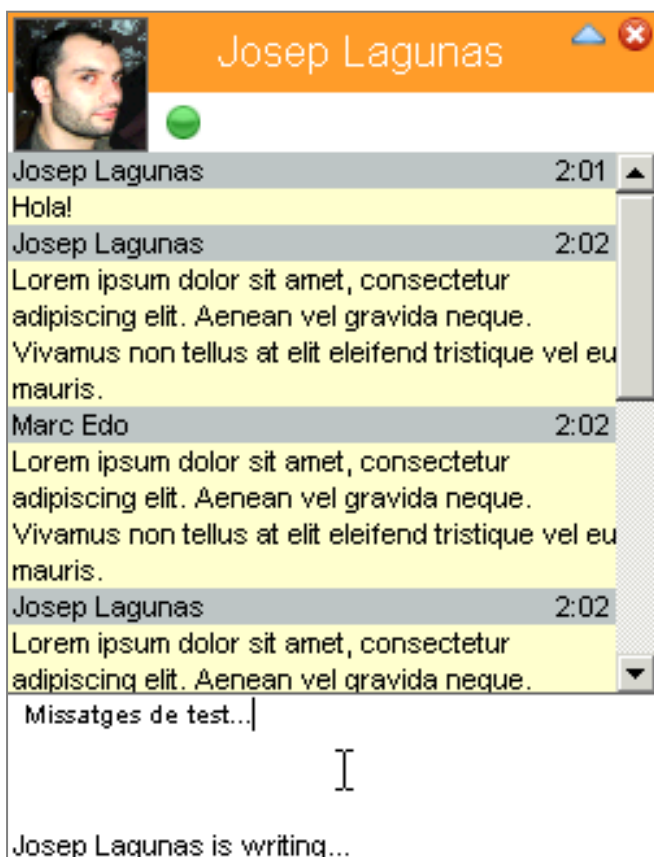


figura 74

La finestra a través de la qual es produeixen les converses entre els usuaris de la plataforma és un tipus de widget que pot flotar sobre tota la superfície del navegador i que pot estar maximitzada (figura 74) o minimitzada (figura 75). En ambdós casos l'usuari té informació sobre l'estat del l'altre així com avís d'escriptura. Per cada conversa que mantingui l'usuari s'obrirà una d'aquestes finestres.



figura 75

## Widget per a la visualització de diagrames de Gantt

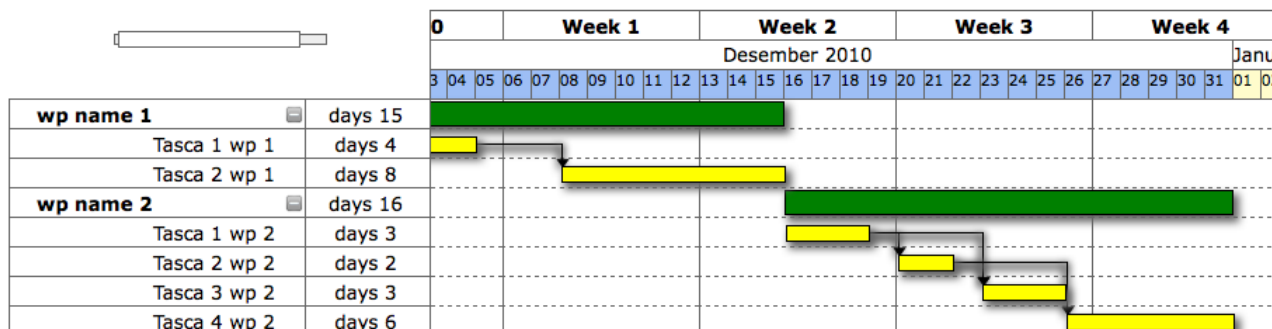


figura 77

S'ha implementat un graficador de diagrames de Gantt que permet visualitzar qualsevol projecte creat a dins del sistema amb aquest.

Aquest visor a estat desenvolupat completament des de zero utilitzant únicament la combinació de elements HTML juntament amb javascript.

El seu funcionament es força simple. S'han definit unes funcions per afegir elements que hi puguin aparèixer com workpackages (representats per una barra de color verd), tasques i subtasques. Les tasques i subtasques tenen definit un color per cadascuna de manera que resulta molt senzill a cop d'ull diferenciar-les.

En cas de ser necessari, la vista permet col·lapsar el gràfic a nivell de workpackage per simplificar la seva visió. Això pot ser dut a terme en clicar a la icona que apareix a la cel·la juntament amb el nom de la tasca.

El següent gràfic resumeix el funcionament dels widgets anteriors basats en comunicació asíncrona via AJAX:

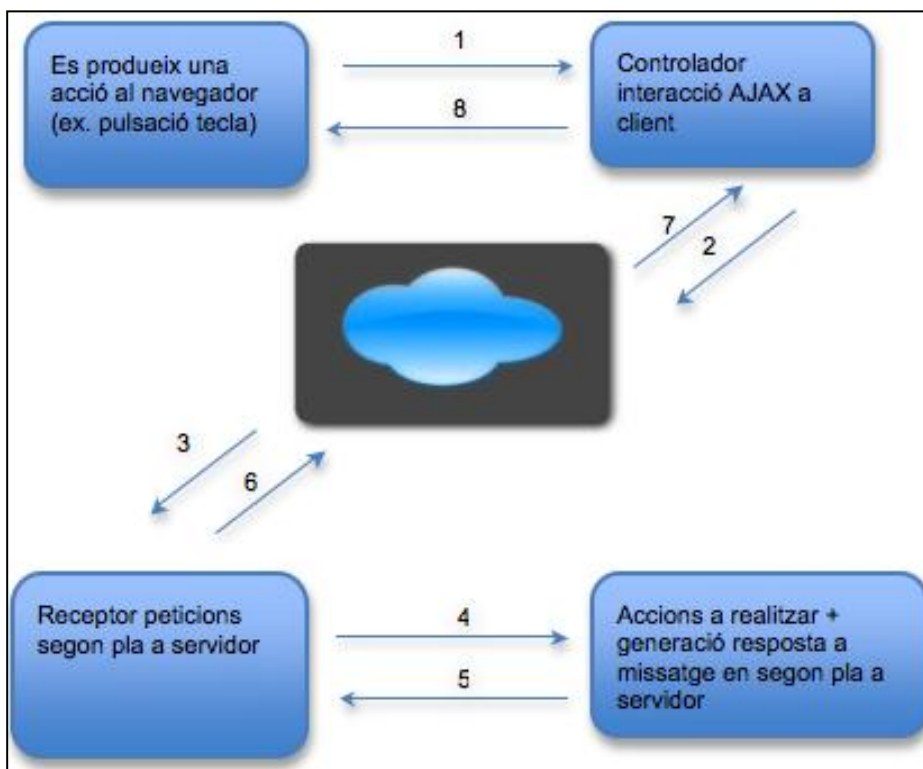


figura 78

1. És produeix un esdeveniment en el navegador que és capturat pel codi de client i que requereix d'informació o d'alguna acció que ha de ser duta a terme al servidor. El controlador d'interacció és notificat i, si s'escau, rep la informació que ha de ser transmesa.
2. El controlador d'interacció AJAX transforma la informació a transmetre al format de transport empleat (JSON)
3. El receptor de peticions del servidor rep la informació transmesa des del client en segon pla i la deserialitza a un format vàlid en cas de ser necessari.
4. Tot seguit, la passa a la capa de lògica de negoci on s'apliquen els càlculs o processos necessaris segons el tipus de petició.
5. El procés de negoci retorna el valor a controlador de la capa d'usuari del costat servidor on es transforma el valor a retornar, en cas de que n'hi hagi, al format de transport (JSON)
6. Les dades en JSON són retornades al controlador d'interacció que ha llençat la petició des del client.
7. El controlador de client rep les dades i actua en conseqüència (per exemple invocant mètodes que actualitzin la vista).
8. L'usuari rep feedback de la petició en segon pla, si cal, per exemple veien noves dades a la pantalla.

## Servidor de xat

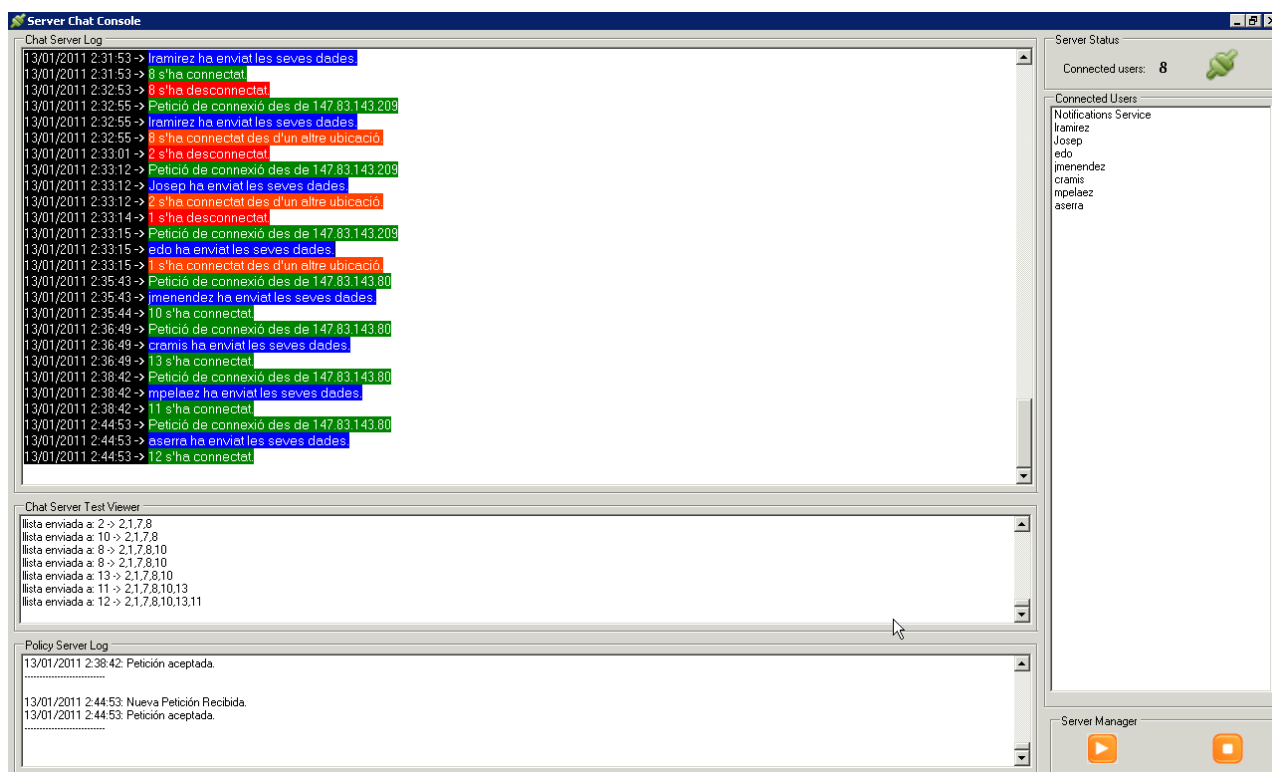


figura 79

A la figura superior podem apreciar la interfície gràfica del servidor de xat que s'ha afegit al Sistema. La implementació del servei de xat s'ha fet tenint en compte que aquest pugui ser fàcilment adaptable com a servei a altres aplicacions.

Aquest servei ha estat desenvolupat completament des de zero incloent-hi el protocol que es fa servir.

Aquest protocol està basat en l'intercanvi d'un conjunt de paquets de dades que viatgen entre els clients i el servidor. Cadascun d'ells té una finalitat diferent i és el servidor l'encarregat de gestionar tota la lògica en funció del tipus de paquet rebut.

Així, tenim una infraestructura que ens permet intercanviar entre els clients unitats d'informació que poden transportar missatges de text, de presència i, fins i tot, dades binàries per transmetre fitxers (tot i que aquest últim tipus no es troba actualment operatiu en aquest sistema).

Aquest servidor ha estat implementat com una aplicació d'escriptori que s'executa en un servidor i disposa d'una sèrie de visors i controls que actualment permeten de manera senzilla les següents tasques d'administració:

- activar o desactivar el servei de xat per a tot el sistema en un sol clic.
- saber la IP de cada petició de connexió rebuda
- saber el número d'usuaris connectats en tot moment
- saber l'entrada i sortida de cada usuari del xat (i per tant del sistema).
- saber el número de reconexions efectuades per cada usuari (i per tant número de sessions que s'han obert per part d'un usuari concret).



---

## Capítol 8

---

### Conclusions

En aquest darrer capítol ens centrarem en extreure valoracions sobre el projecte desenvolupat a diferents nivells.

- Desglossarem les fases per les que ha calgut passar i les planificacions inicials en dies per a cadascuna d'aquestes fases.
- Comentarem les desviacions que ha patit el projecte i podrem calcular els seus costos.
- Analitzarem quins han estat els punts més complexos durant la realització del projecte.
- Estudiarem i proposarem futures línies de desenvolupament pel sistema.
- Finalment es farà una valoració a nivell personal del que ha estat aquest projecte.

## Planificació final

La planificació inicial del projecte en temps efectiu ha estat finalment la que a continuació es detalla:

Captura de requeriments	1 mes	Memòria i documentació
Reunions setmanals amb el client	1 mes	
Especificació del sistema	2 mesos	
Identificació i definició del model conceptual	1 mes	
Definició dels casos d'ús	3 setmanes	
Disseny del sistema	2 mesos	
Normalització i ajustaments del model conceptual	1 setmana	
Disseny de la base de dades	2 setmanes	
Definició de les vistes	1 mes	
Diagrames de seqüència de les principal operacions	1 mes	
Implementació	5 mesos	
I+D tecnologies	2 mesos	
Implementació funcionalitats centrals	3 mesos	
Creació de projectes	3 setmanes	
Gestió d'usuaris	2 setmanes	
Assistent creació WP + Tasques i Subtasques	3 setmanes	
Assistent càrrega massiva fitxers i compartició	1 mes	
Implementació servidor de missatgeria instantània + protocol	3 mesos	
Implementació funcionalitats missatgeria	1 mes	
Implementació client web missatgeria instantània	2 mesos	
Aplicació Silverlight hostatjada	5 setmanes	
Pont javascript-silverlight	1 setmana	
Funcionalitats Javascript interacció gràfica	2 setmanes	
<b>Estimació Total de temps</b>	<b>Aprox. 10 mesos</b>	

Taula 3

## Diagrama de Gantt

si transportem la taula anterior a un diagrama de Gantt obtenim el següent:

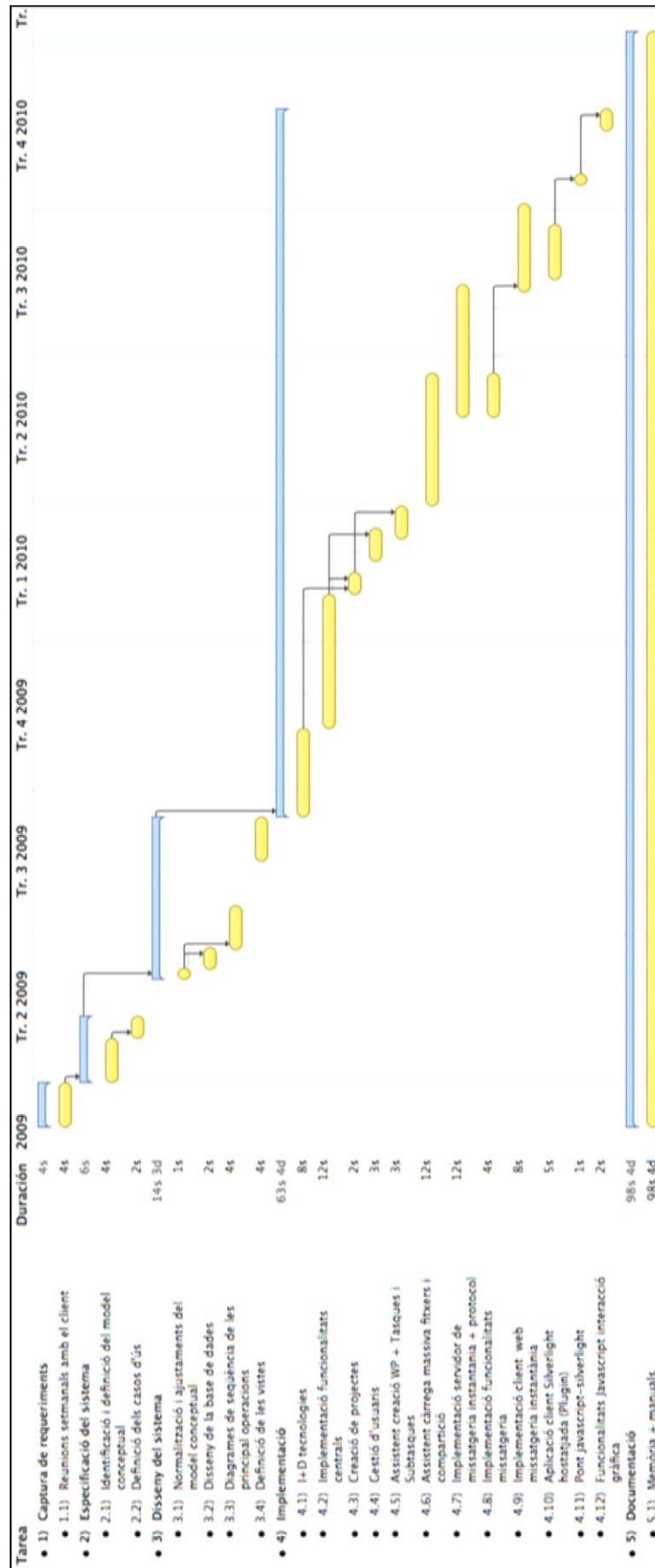


figura 80

Observant el diagrama de Gantt veiem com hi ha tasques que, tot i no ser dependents entre elles, no han estat paral·lelitzades allargant així la durada del projecte. Això és degut al fet de que per la execució del projecte s'ha comptat amb una sola persona com a recurs de treball. Per tant, tot i que a priori la durada del projecte és considerable, no cal perdre de vista aquest detall i fer notar que amb un número més elevat de recursos el seu temps d'execució hauria pogut disminuir considerablement.

## Desviacions patides durant l'execució

Si tenim en compte la planificació inicial representada a la taula 1 i la comparem amb els diagrama de la figura 80 s'observa clarament com la fase que més temps s'ha desviat de les estimacions inicials es la d'implementació.

En un primer moment es va valorar que en tan sols dos mesos es podria controlar les noves tecnologies i acabar la implementació del sistema però, clarament, va ser una estimació massa optimista. La corba d'aprenentatge que va suposar el fet d'aprendre i controlar les noves tecnologies va ser més pronunciada del que s'esperava allargant-se fins als dos mesos irremediablement.

En el diagrama es pot observar com el temps necessitat de formació ha estat d'unes 8 setmanes les quals, finalment, han estat formant part d'un període més llarg.

Per altra banda, la pròpia implementació del sistema ha estat més complexa del previst a priori entre altres motius pel fet de fer conviure fins a 4 tecnologies simultàniament (situació que ha provocat complicacions inesperades).

A aquesta desviació temporal, cal afegir retards provocats per canvis en les peticions que han estat rebudes de part del client i que en repetides ocasions han obligat a iterar sobre especificacions i dissenys a priori ja aprovats i que finalment han hagut de ser modificats.

Per últim és important fer notar que el diagrama representa el context perfecte, on el 100% dels recursos/persones designats per a l'execució d'un projecte poden ser dedicats plenament a aquest fi. En el nostre cas no ha pogut ser així però tot i això s'ha optat per representar tots els blocs seguits en el temps per donar una visió completa i continuada del projecte.

## Costos associats al projecte

Per calcular els costos associats al desenvolupament del projecte s'ha partit d'un nombre d'hores extret de les dades aportades pel diagrama de Gantt i s'han identificat dos rols diferenciats dins del suposat equip de treball.

Per una banda s'han desglossat les hores dedicades a tasques d'anàlisi, això engloba tota tasca que tingui a veure amb aspectes d'especificació i disseny. Per altra s'han sumat els totals d'hores dedicades a tasques de codificació i altres aspectes com, per exemple, disseny d'interfícies gràfiques.

En quan als sous, s'ha optat per fer una cerca de diferents sous recollits en convenis de caire públic i s'ha procedit a realitzar una mitjana per calcular un preu hora obtenint els següents resultats:

Funció desenvolupada	Preu per hora
Analista	40 €
Programador	27.5 €

**Taula 4**

resultats calculats desglossant el total d'hores pels dos rols definits a taula anterior:

Rol dins l'equip de desenvolupament	Número total d'hores calculat
Analista	800
Programador	800

**Taula 5**

Amb els valors anteriorment obtinguts ja podem donar un valor aproximat del cost del projecte un cop finalitzat.

Rol dins l'equip	Número d'hores	Preu/hora	Total per rol
Analista	800	40 €	32000 €
Programador	800	27.5 €	22000 €
TOTAL			54000 €

**Taula 6**

## Futures línies de treball

Una de les primeres coses que podrien ser tingudes en compte com a futura línia de treball seria el fet d'aprofundir en aspectes i millores de la interfície d'usuari. Avui dia existeixen professionals dedicats exclusivament a millorar l'experiència d'usuari i considero que encara queda molta feina per fer en aquest aspecte.

Per altra banda, però seguint en la mateixa línia, crec que seria interessant desenvolupar dues noves interfícies d'usuari. Una per accedir al sistema des de dispositius mòbils (de manera que, un usuari fos capaç d'estar en qualsevol moment actualitzat dels possibles canvis que el sistema pateixi) i l'altre, com a client d'escriptori.

El fet de desenvolupar un client d'escriptori o smartClient permetria dotar de molta potència la interfície d'usuari ja que es deixaria d'estar limitat pel propi navegador. Alguns exemples d'aquests tipus de sistema són per exemple PICASSA, l'aplicació d'escriptori de e-Bay o Spotify.

Abandonant ja aspectes relacionats amb la capa d'usuari, també crec que seria interessant readaptar el client de xat aprofitant les noves característiques que ofereix el llenguatge HTML 5.

HTML 5 incorpora de sèrie la possibilitat d'establir via javascript una connexió a través del que s'anomena websocket cap al servidor. Aquest fet permet que, de manera nativa i sense necessitat de fer ús de plugins com ara mateix es fa al nostre sistema, s'estableixi una comunicació fullduplex entre el navegador del client i el servidor podent fer d'aquesta manera enviament de dades des de servidor cap a client sense la necessitat d'instal·lar cap plugin extern com fins ara.

## Valoració personal

Tot i haver estat un camí llarg i feixuc per arribar fins aquí, sens dubte molt més del que em pensava en el moment en que vaig triar aquest treball, no puc evitar sentir-me content i satisfet pel que la seva culminació significa.

Durant el camí hi han hagut moments bons i moments no tant bons. Sens dubte, el que més cal valorar per a mi és tot el que he après en aquest temps. He tingut ocasió de viure tot el procés que hi ha al darrera de la creació d'una aplicació "real" i, per tant, d'enfrontar-me als problemes que això comporta des del tracte amb clients fins a les nits en blanc tractant de trobar solucions que, semblava, no volien aparèixer.

A nivell més personal, una de les coses per les que més m'ha servit ha estat per acabar de confirmar que, el que realment m'agrada d'aquesta professió, és tot allò relacionat amb les fases d'anàlisi, especificació i disseny d'un sistema d'informació. El fet de pensar com caldrà organitzar la informació, descobrir les relacions que hi han entre els diferents elements que en formen part i la manera com cal fer que interactuïn entre ells.

Tornant al treball de projecte i, focalitzant més en la part tecnològica, he tingut també la oportunitat de treballar amb eines que fins ara mai no havia fet servir i aprofundir-hi en el seu ús fins al punt de que aquest fet m'ha permès canviar de feina i començar a encaminar la meua carrera laboral fent el que realment m'agrada.

També hi hagut moments durs, moments en els que han sorgit problemes que, per moments, semblaven irresolubles però per als que a la fi, amb més o menys temps i esforç, acabava trobant una solució o altra.

Crec que no em confonc si dic que de totes les etapes que he cremat durant la carrera el fer el PFC és de les que més sento que m'han fet madurar per finalment poder dir que m'he sentit com un enginyer.

En definitiva i, per concloure, només puc valorar l'experiència viscuda d'una manera molt positiva i sentir-me realment content per tot el que he après en aquesta etapa.