

eTaller – Gestió d'Escoles Taller

Gonçal VERGE ARNAU
ETIG

Antoni OLLER ARCAS

9 de gener de 2006

1. Dedicatòria i agraïments

Aquest treball està inspirat en la feina que cada dia, durant aquests últims tres anys, s'ha realitzat amb l'ajuda dels meus companys de l'escola taller i de totes les escoles tallers de Catalunya, una feina tant desconeguda com valuosa.

Gràcies a la gent que m'ha envoltat durant aquest últim semestre per suportar un procés de programació sense acabar dels nervis.

Agraïment especial al meu professor Javi de Java al centre de tecnologia de Saragossa, qui només en deu dies em va obrir un infinit oceà de possibilitats, per on després m'ha anat guiant.

2. Resum

Memòria final del treball de final de carrera en l'àrea de J2EE.

Es presenta el projecte **eTaller** – Gestió d'Escoles Taller com a marc escollit per a demostrar el procés **d'aprenentatge i assimilació dels coneixements adquirits** durant la carrera en les diferents assignatures i posar-los en pràctica en la gestió d'un projecte concret.

L'elaboració del producte ha seguit tot el **cicle de vida de software** fent ús dels procediments emprats en assignatures com Enginyeria de Programari, Bases de Dades, Estructura de la Informació, Interacció Humana amb els Ordinadors... i sobretot **Programació Orientada a Objectes** (amb tecnologia Java).

Primerament es va estructurar la feina en un **pla de Treball** presentat en el seu moment i que ha estat l'eina primera de treball. Després s'ha realitzat el procés **d'anàlisi** d'un cas concret, la gestió d'una Escola Taller, on s'ha de controlar les altes i baixes dels usuaris, grups, tasques i sancions, les assignacions d'aquestes i el control de tots aquests factors.

En la fase de **disseny** s'ha fet us d'eines com el **JUDE** a l'hora de treballar amb els diagrames necessaris, així com el **Word, Excel** i altres eines ofimàtiques.

La **implementació**, ha estat l'etapa més dura, i s'ha estès més enllà de la franja temporal contemplada prèviament. De totes maneres ha estat força enriquidor. L'eina principal en aquesta fase ha estat el **netBeans**, que ha fet de nexa d'unió amb altres aplicacions necessàries com el **tomcat** (a través del connector a servidors) o el **MySQL** (a través del mySQL connector).

Per a finalitzar, **l'elaboració de la memòria**, on s'exposarà tot el procés resumit aquí, incloent detalls de disseny, instruccions d'ús i d'instal·lació, problemàtiques...

El projecte fa servir com a interfície la tecnologia **WEB**, això fa que la major part hagi estat programat amb **JSP** (pàgines dinàmiques) fent us **d'HTML** (presentació i pàgines estàtiques) i com no **Java** per a crear totes les classes necessàries (usuari, grup, tasca... així com les capes d'accés **DAO i els servlets**).

3. Estructura de la Memòria

1. Dedicatòria i agraïments.
2. Resum. Paraules clau. Nom de l'àrea de TFC.
3. Estructura de la memòria.
4. Cos de la memòria, dividit en capítols.
 - 4.1. Introducció.
 - 4.1.1. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.
 - 4.1.2. Objectius del TFC.
 - 4.1.3. Enfocament i mètode seguit.
 - 4.1.4. Planificació del projecte.
 - 4.1.5. Productes obtinguts.
 - 4.1.6. Breu descripció dels altres capítols de la memòria.
 - 4.2. Anàlisi i especificació de l'aplicació.
 - 4.2.1. Descripció.
 - 4.2.2. Composició del Programari.
 - 4.2.3. Funcionalitats.
 - 4.2.4. Seguretat.
 - 4.2.5. Funcionalitats proposades per a properes versions
 - 4.2.6. Casos d'ús.
 - 4.3. Disseny.
 - 4.3.1. Diagrama de paquets.
 - 4.3.2. Diagrames UML.
 - 4.3.3. Diagrames de col·laboració i seqüència.
 - 4.3.4. Interfície gràfica d'usuari.
 - 4.3.5. Persistència.
 - 4.3.6. Reutilització.
 - 4.4. Implementació.
 - 4.4.1. Pàgines dinàmiques i estàtiques
 - 4.4.2. Les classes principals
 - 4.4.3. Els gestors
 - 4.4.4. Els servlets
 - 4.4.5. Proves
 - 4.5. Conclusions.
 - 4.6. Manuals.
 - 4.6.1. Manual d'instal·lació.
 - 4.6.2. Manual d'usuari.
5. Bibliografia.
6. Glossari.

4.1 Introducció

Justificació del TFC.

Es clar que un dels motius principals que porta la gent a estudiar a la UOC és la possibilitat de compaginar-ho amb el món laboral.

Els estudis d'enginyeria Tècnica en Informàtica de Gestió proporcionen un formació en diferents àrees que, en cas d'avinentesa, es poden posar en pràctica de forma immediata en el dia a dia del món del treball.

El marc escollit es una Escola Taller, la seva estructura es semblant a la d'una escola normal, per tant la seva gestió es similar, altes, baixes, faltes d'assistència... d'alumnes però, amés s'ha de gestionar les tasques que es realitzen, aquest apartat es similar al que podria dur a terme una brigada municipal.

Es realitzarà una aplicació WEB per a la gestió del centre, de forma que qualsevol membre, ja sigui administrador, professor o treballador pugui accedir a les dades accessibles pel seu perfil. Així tots els usuaris de l'escola tindran informació a l'acte de qualsevol acció relacionada amb la seva persona (tasques a realitzar, sancions imposades, dades personals...).

Per altra banda cal mencionar que les possibilitats que dona Internet fan possible l'accés a la Web des de qualsevol lloc amb connexió a la xarxa, això permet als pares o tutors visualitzar d'una forma ràpida l'evolució dels alumnes.

Context en el qual es desenvolupa: punt de partida i aportació del TFC.

A l'actualitat l'escola treballa amb una base de dades dissenyada amb Access per a controlar els usuaris (altes, baixes...) i tot un mecanisme burocràtic (amb paper) per a gestionar les faltes d'assistència, sancions, tasques a realitzar o realitzades pel grup...

Es pretén automatitzar aquests dos processos per a poder ser gestionats des de la web, de forma directa, ràpida i controlada. Poder accedir de forma immediata a les dades introduïdes per qualsevol dels usuaris de l'escola alliberant així la informació i fent-la accessible a tots (tenint en compte els permisos assignats a cada perfil d'usuari).

Objectius del TFC.

- Demostrar els coneixements adquirits en el procés d'aprenentatge que s'ha dut a terme durant la carrera.
- Realitzar una aplicació útil i de fàcil implantació, amb una corba d'aprenentatge el més suau possible.
- Elaboració de tot el material associat al projecte, per a l'obtenció d'un producte final ben acabat i complet.

Objectius implícits:

- Desenvolupament de servlets i ús de Tomcat(contenedor de Servlets).
- Tecnologia JSP.
- Estudi i ús de patrons de disseny, aplicant-los en l'arquitectura J2EE.
- Comunicació amb bases de dades relacionals (MySQL).
- Ús de capes d'accés a dades (DAO's).
- Ús de Sessions web.

Enfocament i mètode seguit.

En tot moment s'ha intentat portar el projecte cap a la realitat laboral més immediata, ja que des de la seva concepció pertany a un cas concret real.

Això ha permès elaborar una documentació acotada al màxim a les necessitats del dia a dia a l'Escola Taller.

Entrevistes amb la direcció, treballadors i testeig en forma de prototip avalen el procediment.

S'ha seguit el cicle de vida en cascada, utilitzant la metodologia d'orientació a objectes, i aplicant patrons de disseny.

Planificació del projecte.

Calendari del projecte

Id	Nom Tasca	Duració	Inici	Final	Predecessores
1	Pla de Treball	7	19/09/2005	26/09/2005	
2	Descripció	1	19/09/2005	20/09/2005	
3	Calendari	3	20/09/2005	23/09/2005	2
4	Document Pla de Treball	3	23/09/2005	26/09/2005	3
5	Anàlisi	21	26/09/2005	17/10/2005	1
6	Decripció Projecte	7	26/09/2005	03/10/2005	
7	Casos d'ús	7	03/10/2005	10/10/2005	6
8	Diagrama d'entitats	7	10/10/2005	17/10/2005	7
9	Disseny	21	17/10/2005	07/11/2005	5
10	Diagrama de classes	3	17/10/2005	20/10/2005	
11	Diagrama de casos d'ús	3	20/10/2005	23/10/2005	10
12	Diagrama de col·laboració	3	23/10/2005	26/10/2005	11
13	Diagrama d'estats	3	26/10/2005	29/10/2005	12
14	Reutilització i disseny	3	29/10/2005	01/11/2005	13
15	Excepcions	3	01/11/2005	04/11/2005	14
16	Interfícies gràfiques	3	04/11/2005	07/11/2005	15
17	Implementació	32	07/11/2005	09/12/2005	9
18	Estructura	5	07/11/2005	12/11/2005	
19	Gestió continguts	5	12/11/2005	17/11/2005	18
20	Usuaris	5	17/11/2005	22/11/2005	19
21	Gestió Usuaris	7	22/11/2005	29/11/2005	20
22	Projecte	10	29/11/2005	09/12/2005	21
23	Presentació i Memòria	31	09/12/2005	09/01/2006	17
24	Proves	3	09/12/2005	12/12/2005	
25	Assimilació Funcionament	3	12/12/2005	15/12/2005	24
26	Retocs*	3	15/12/2005	18/12/2005	25
27	Redacció de la memòria	15	18/12/2005	02/01/2006	26
28	Elaboració Presentació	7	02/01/2006	09/01/2006	27

Temporalització i documentació

Seguint la temporalització establida per al lliurament de les proves d'avaluació continuada, les diferents dates d'entrega aniran paral·lelament a les fases del projecte. Podem distingir les següents:

Títol	Lliurament
Pla de treball	26/09/2005
Anàlisi	17/10/2005
Disseny	07/11/2005
Implementació	09/12/2005
Memòria + Presentació	09/01/2006

Productes obtinguts. Breu explicació.

- **Memòria.** Presentació del projecte. Inclou el material elaborat durant el procés de construcció del programari (pla de treball, anàlisi, disseny, manuals...), comentaris, conclusions, justificacions i valoració econòmica.
- **Producte.** Aplicació web creada. Es proporciona en format .war, de forma empaquetada. Al manual d'instal·lació es detalla la seva instal·lació i posada en funcionament.
- **Presentació.** Document explicatiu en forma resumida sobre el treball realitzat durant el procés de creació del projecte final de carrera.

Breu descripció dels altres capítols de la memòria.

Aquesta memòria intentarà ser un reflex fidel de tot l'esforç i treball que s'ha emprat per a la realització del projecte, en forma de producte de software.

Seguint la pauta marcada per la temporalització del mateix TFC, es podrà distingir clarament entre els processos de:

- Definició del pla de treball (al capítol anterior)
- Anàlisi.
- Disseny.
- Implementació.

Amés s'inclou a forma de conclusions, les inquietuds, dificultats i vivències que han sorgit durant el procés.

Per a que el producte estigui acabat del tot també es presenten els manuals d'instal·lació i funcionament de l'aplicació Web.

Per acabar, Glossari, bibliografia i annexos.

4.2. Anàlisi i especificació de l'aplicació.

4.2.1. Descripció

El programa eTaller ha de permetre que un equip format per un nombre indeterminat d'usuaris puguin consultar, introduir dades o gestionar el funcionament d'un centre de formació (escola Taller) o realitzar altres tasques des de el seu domicili o una ubicació remota.

Al local de l'escola hi ha un petit equip d'auxiliars administratius, coordinats per un administrador que portaran tota la gestió de l'escola:

- Altes/Baixes/Modificacions de treballadors.
- Altes/Baixes/Modificacions de grups.
- Altes/Baixes de tasques.
- Altes/Baixes de sancions.
- Gestió administrativa.
- Etc.

La comunicació entre els professors i l'escola es directa, però les accions dels professors es realitzaran via Internet:

- Consulta de dades pròpies.
- Consulta de dades del grup.
- Assignació de tasques.
- Assignació de sancions.
- Finalització de tasques.

La comunicació entre els treballadors i l'escola es directa, però les accions dels treballadors es realitzaran via Internet:

- Consulta de dades pròpies.
- Consulta de dades del grup.
- Consulta de tasques a realitzar.
- Consulta de sancions assignades.

La present aplicació es limitarà a tracta el manteniment de treballadors i el manteniment de tasques i sancions sota les premisses que s'aniran detallant en el present document.

4.2.2. Composició del Programari

El programari constarà de dues part:

- Part del Servidor: Instal·lada en l'equip servidor situat a les oficines centrals de l'escola. És on fa el manteniment de treballadors i de tasques.
- Part Client: Simplement des d'un navegador d'Internet i introduint la direcció de l'aplicació en el servidor de l'escola. No fa falta cap instal·lació prèvia, simplement haver estat donat d'alta al servei per algun dels administradors.

4.2.2.1. Instal·lació de eTreball

A l'ordinador central de *eTaller* s'hi hauran d'instal·lar tots el recursos i dades necessàries per permetre la correcta execució de la aplicació, via Internet des de qualsevol ubicació remota.

4.2.2.2. Gestió d'Usuaris

Tenim tres tipus d'usuaris caracteritzats en la present aplicació, cadascun amb una sèrie de característiques ben definides:

- Administrador: Té funcions de treballador administratiu de l'oficina central. Pot fer el manteniment d'usuaris, grups, tasques i sancions.
- Professors: Poden assignar i donar per finalitzades tasques al grup, sancions de forma personal, consultar dades personals i del grup i accedir a l'historial de tasques i sancions del seu grup.
- Treballadors: Poden consultar dades personals i del grup i accedir a l'historial de tasques del seu grup i sancions personals.

4.2.3. Funcionalitats

Tots els usuaris hauran de passar per:

Identificació i validació de contrasenya

En el moment d'iniciar l'aplicació, el sistema demanarà un codi d'usuari (dni) i una contrasenya.

Un cop entrat en l'aplicació l'administrador podrà donar d'alta a la resta d'usuaris. L'administrador donarà a cada usuari una contrasenya per a que pugui accedir al sistema, l'usuari la podrà canviar quan vulgui utilitzant l'opció Canvi Contrasenya.

Cara a tenir un control d'accessos no autoritzats, tant en la gestió d'usuaris com en l'accés al sistema per part dels treballadors de l'empresa, si s'introdueix tres vegades un codi erroni l'usuari es marca com bloquejat, l'administrador l'haurà de desbloquejar entrant a l'opció de manteniment d'usuaris.



The screenshot shows a login interface with a yellow background. At the top, it says 'Escola Taller Hecavònia' with a small logo. Below that, the text 'Benvingut :: Identificació' is centered. There are two input fields: 'DNI:' and 'Contrassenya:'. Below the fields is an 'Acceptar' button.

Tancament de la sessió

En el moment en que l'usuari es troba al menú d'opcions pot donar per finalitzada la seva sessió prement l'opció logout. Immediatament es procedeix al tancament de sessió i direccionament a la pàgina d'identificació.



The screenshot shows a yellow background with the text 'logout' in purple. Below it is a link 'tornar a l'inici'. At the bottom, there is a copyright notice: '© Gonçal Verge gverge@uoc.edu ETIG a la UOC'.

Com Administrador es podran realitzar les funcionalitats:**4.2.3.1 Alta D'usuaris**

Aquesta opció permetrà afegir nous usuaris al sistema, de manera que quedaran autoritzats per accedir a l'aplicació.

L'administrador encarregat d'enregistrar l'alta haurà d'indicar el NIF, nom i cognoms, així com el perfil d'usuari (professor o treballador) i assignar-li una contrasenya. Els únics perfils d'usuari que coneixerà el sistema són els administrador, els professors i els treballadors. Opcionalment també es podrà introduir l'adreça, l'e-mail, telèfon, població, etc. I el grup de treball al que pertany (electricitat, paleta, informàtica...).

Tots el codis d'usuaris han de ser únics, de manera que no se n'acceptarà cap de duplicat. Per tant, si s'indica un codi repetit, el programa rebutjarà l'alta.

Un cop introduïdes totes les dades s'haurà de prémer el botó Acceptar per que el programa les guardi i, en aquest moment, el sistema assignarà automàticament un data d'alta a l'usuari.

4.2.3.2. Baixa D'usuaris

Aquesta opció permetrà donar de baixa als diferents usuaris del sistema. Un usuari en situació de baixa no podrà utilitzar el programa.

Inicialment l'administrador només podrà introduir el codi de l'usuari i, després de prémer el botó Acceptar el programa li mostrarà les seves dades.

Un cop mostrades les dades de l'usuari, l'administratiu després de comprovar que efectivament es l'usuari a esborrar pot escollir l'opció eliminar i esborrar-lo de la base de dades.

4.2.3.3. Modificacions D'usuari

Amb aquesta opció, després d'introduir-ne el codi i prémer el botó Acceptar, qualsevol usuari administrador podrà modificar les dades d'identificació d'un usuari, excepte el seu DNI. A més, podrà assignar una nova contrasenya.

4.2.3.4. Recerca D'usuaris

Es podrà cerca l'usuari que es vol consultar a traves del seu dni.

Així, un cop introduïdes les condicions de cerca i prémer el botó Cercar, el programa mostrarà una nova finestra on es mostraran les dades de l'usuari a cercar.

4.2.3.5. Llistat D'usuaris

Es mostren tots els usuaris de la base de dades sense diferenciació de grups ni perfils.

4.2.3.6. Altres funcionalitats d'administrador

Seguint els mecanismes d'alta, baixa i consulta d'usuaris és pot relacionar clarament aquestes funcionalitats amb les pertinents a grups, tasques i sancions. L'administrador pot realitzar les mateixes accions amb aquestes classes.

Cal mencionar però, que l'alta de tasques o sancions no te relació directa amb l'assignació de les mateixes als usuaris, ja que aquestes accions han de ser realitzades pel professor. L'administrador doncs es limitarà a crear una tasca o una sanció determinada de forma genèrica, el professor serà l'encarregat d'assignar-les al grup o a l'alumne pertinent incloent la data de sanció i la informació necessària.

Com a Professor es podran realitzar les funcionalitats:



4.2.3.7. Assignació de Tasca.

El professor podrà assignar una tasca al seu grup de forma senzilla, només haurà de seleccionar la tasca (prèviament donada d'alta per l'administrador) de la llista de tasques i donar una data d'alta.

Tots els membre del grup tindran la tasca assignada immediatament. Quedarà reflectit al llistat de tasques assignades.

4.2.3.8. Finalització de Tasca.

Un cop la tasca estigui finalitzada el professor podrà reflectir-ho a l'aplicació web accedint a l'opció de finalització de tasca, només caldrà seleccionar-la a la casella (on només apareixen les tasques iniciades amb la seva data d'inici) i afegir la data de finalització.

Tots els membre del grup tindran la tasca finalitzada immediatament. Quedarà reflectit al llistat de tasques finalitzades.

4.2.3.9. Assignació de Sanció.

El professor podrà sancionar un alumne de forma senzilla, només haurà de seleccionar la sanció (prèviament donada d'alta per l'administrador) de la llista de sancions i l'alumne al qual es sanciona, les hores de sanció i un petit comentari.

El membre del grup al qual s'ha sancionat tindrà la sanció assignada immediatament. Quedarà reflectit al llistat de sancions assignades (totals, al professor) i sancions assignades (personals, al treballador).

4.2.3.10. Sancions Assignades.

Llistat de sancions assignades pel professor on consta, el codi de sanció, l'alumne sancionat, les hores assignades i els comentaris pertinents.

Com a Professor i alumne es podran realitzar les funcionalitats:**4.2.3.11. Consulta de les dades personals.**

Un treballador/professor podrà consultar les seves dades personals sempre que vulgui. Com que ja ha estat prèviament validat en la connexió, no cal que es torni a identificar per demanar-les (només pot accedir a les seves).

Se li mostra la mateixa pantalla que poden veure els Auxiliars i l'Administratiu si be no hi pot fer cap modificació. Tindrà un botó addicional per fer el càlcul de la seva remuneració.

4.2.3.12. Consulta dels components del grup.

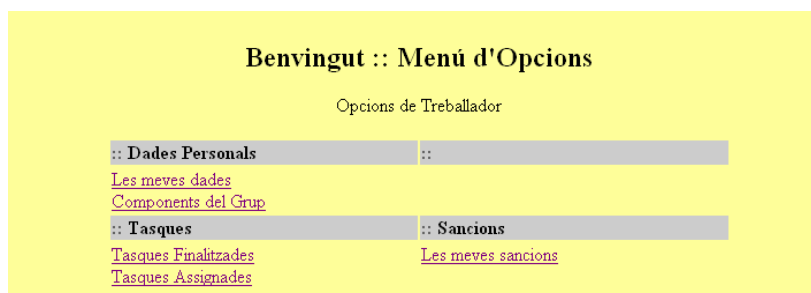
Un treballador/professor podrà consultar els membres del seu grup. Només pitjant en l'opció membres del grup, es mostrarà un llistat amb els professors i treballadors del grup.

4.2.3.13. Consulta dels components del grup.

Un treballador/professor podrà consultar les tasques assignades al seu grup. Només pitjant en l'opció tasques assignades, es mostrarà un llistat amb les tasques assignades al seu grup.

4.2.3.14. Consulta dels components del grup.

Un treballador/professor podrà consultar les tasques finalitzades al seu grup. Només pitjant en l'opció tasques finalitzades, es mostrarà un llistat amb les tasques finalitzades pel seu grup.

Com a alumne es podran realitzar les funcionalitats:

Amés de les funcionalitats que es comparteix amb el professor del grup, s'ha de recordar l'opció que ens mostra les sancions assignades personalment.

4.2.4 Seguretat

El sistema de seguretat establert pel programa eTaller es basa en la identificació i validació del seu codi de usuari i contrasenya, quan s'inicia l'aplicació, l'accés a l'aplicació es fa mitjançant un DNI d'usuari i la seva contrasenya, i es comprova la seva existència i si la contrasenya és correcta.

En primera instància l'administrador la crea al fer l'alta, i l'usuari només la podrà canviar posant-se en contacte amb aquest.

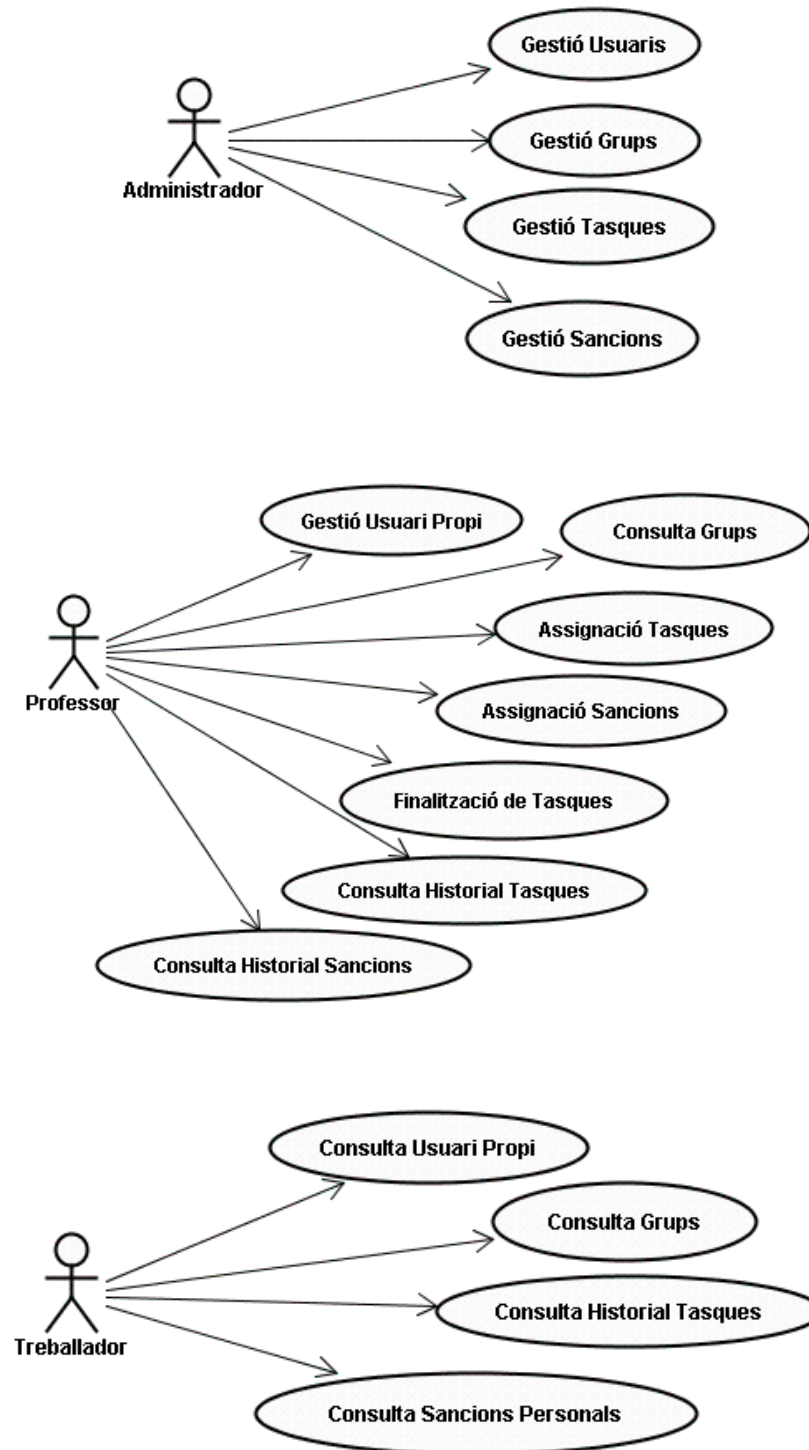
No estan previstes dates de caducitat fixes. L'administrador procedirà de la manera que cregui convenient usuari per usuari.

4.2.5 Funcionalitats proposades per a properes versions

- De cara a optimitzar l'accés al servidor per part dels treballadors i garantir el bon funcionament del sistema , aquells treballadors que estiguin més d'uns 15 minuts aproximadament sense fer cap operació , possibilitat de desconnectar-los del sistema.
- Màxim de tres intents a l'hora d'introduir la contrasenya.
- Llistats per dates d'alta, modificació...
- Justificació de faltes on-line.
- Preinscripció on-line.
- Missatgeria.
- Gestió de remuneracions.
- Tractament d'excepcions.

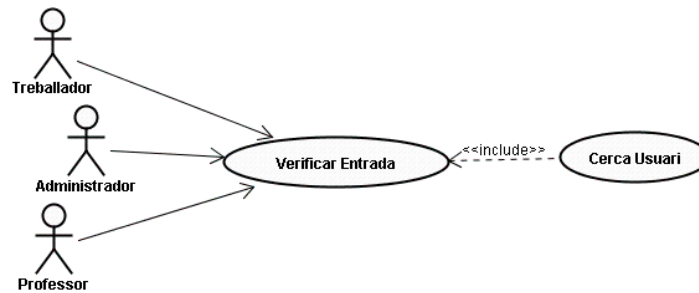
4.2.6. Casos d'us.

Simplificat amb tots els actors.



Alguns exemples principals.

Identificació d'usuari.



Funcionalitat General: Verifica que el DNI i contrasenya entrats correspongui a un usuari (de qualsevol tipus) donat d'alta dins del sistema.

Paper dins el treball de l'usuari: És el primer cas d'ús en que es troba l'usuari.

Actors: Administrador, Professor, Treballador

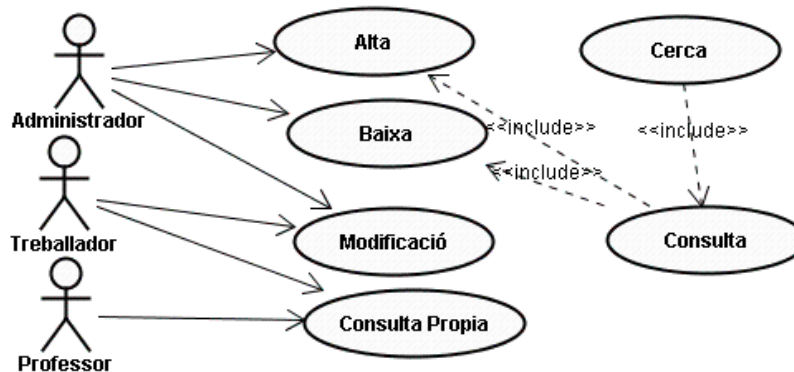
Casos d'ús relacionats: Cerca Usuari

Precondició: L'usuari és vàlid i coneix la seva contrasenya.

Postcondició: El sistema ha validat l'usuari i, si és el cas, li ha permès l'entrada a les opcions corresponents.

Descripció: El sistema demana el DNI de l'usuari i la seva contrasenya, es comprova que existeixi i que la contrasenya sigui correcte. En el cas de que les dades siguin correctes es permet a l'usuari l'entrada al seu menú d'opcions dependent del seu perfil.

Casos d'Ús de Gestió d'Usuaris



Funcionalitat General: Permet Altes / Baixes (Administrador), Modificacions (Administrador i Professor), Consultes (Administrador, Professor i Treballador).

Paper dins el treball de l'usuari: Defineix tot el personal que pot actuar dins de l'aplicació.

Actors: Administrador, Professor, Treballador

Casos d'ús relacionats: Verificar Entrada

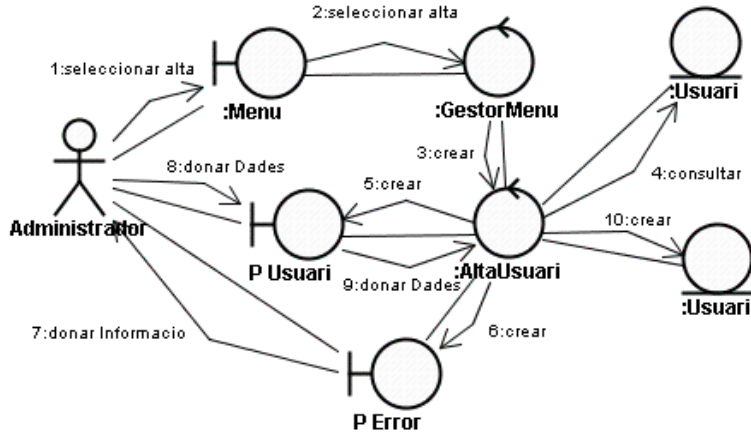
Precondició: L'usuari sigui del tipus que sigui s'ha validat.

Postcondició: Les accions seleccionades s'han realitzat.

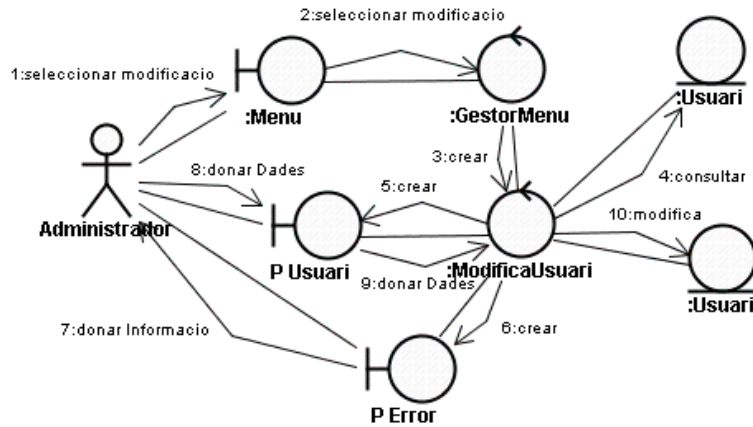
Descripció: A partir de la opció de menú corresponent s'introdueix el DNI de l'usuari (excepte alta). Quan es selecciona modificació o baixa, si existeix l'usuari es mostra tota la informació disponible (permetent modificar-la -no el DNI-, si s'escull l'opció de modificació).

Observacions: En modificacions tots el camps son modificables excepte el DNI.

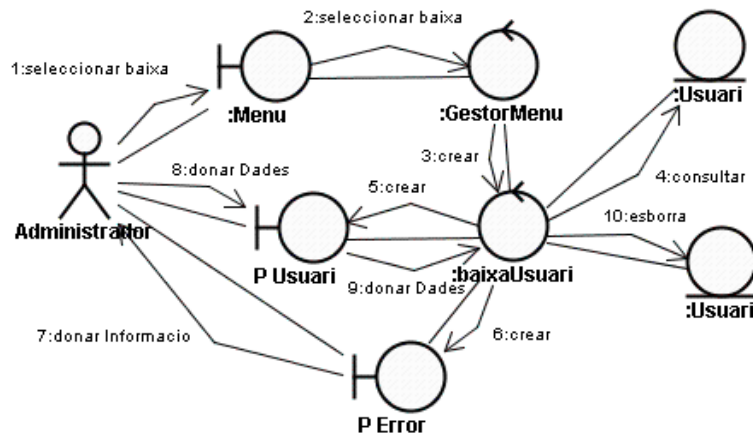
Alta Usuari



Modificació Usuari

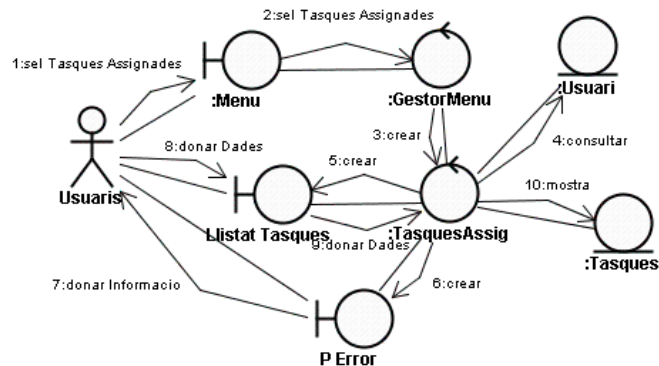


Baixa Usuari



Els casos d'ús de alta/modificació/consulta/baixa de tasques/grups/sancions son molt similars i no els mostrarem.

Cas d'ús Consulta Tasques Assignades



Resum funcionalitats: Es mostra una llista amb totes les tasques que s'han assignat al grup al qual pertany l'usuari que ha iniciat la sessió i que encara no han estat finalitzades.

Paper dins el treball de l'usuari: Obtenir informació sobre les tasques que s'estan realitzant o s'han de realitzar. És un cas d'ús secundari.

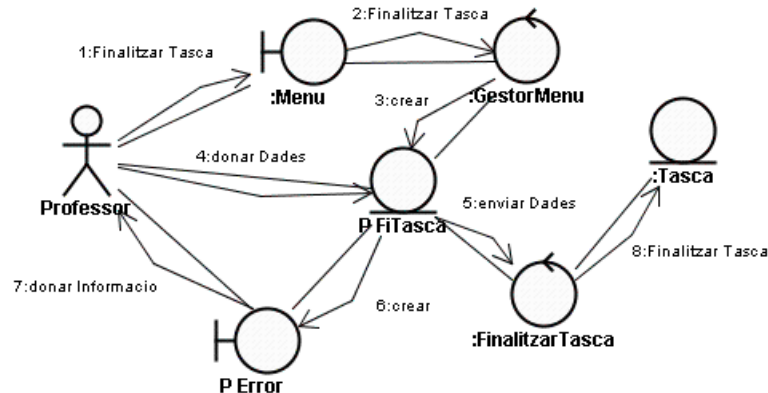
Actors: Professors i Treballadors

Precondició: L'usuari s'ha validat en el sistema indicant el seu DNI i la seva contrasenya i està autoritzat per a realitzar aquesta consulta.

Postcondició: S'ha mostrat una finestra amb les dades sol·licitades o s'ha mostrat un missatge indicant l'eventualitat que no ha permès mostrar-les.

Descripció: L'usuari escollirà en el menú l'opció de tasques assignades, un cop validat, es mostrarà una finestra on es relacionen les tasques que el treballador te assignades però no estan finalitzades.

Cas d'Ús Validar Tasca



Resum funcionalitat: finalitza una tasca.

Paper dins el treball del Professor: cada vegada que ha acabat una tasca i ell la dona per finalitzada.

Actors: Professor

Casos d'ús relacionats: Rebre llistat tasques

Precondició: Ha rebut un llistat i ha seleccionat una tasca pendent de finalitzar.

Postcondició: Ha enviat la informació del nou estat de la tasca al servidor

Descripció: Cada vegada que el grup finalitza una tasca el professor accedeix a l'aplicació, rep el llistat de tasques, selecciona la tasca i la dona per finalitzada.

4.3. Disseny.

El programari ETaller permetrà al personal de l'escola mantenir un seguiment de les diferents tasques que li són encomanades, les seves tasques, sancions i assistència.

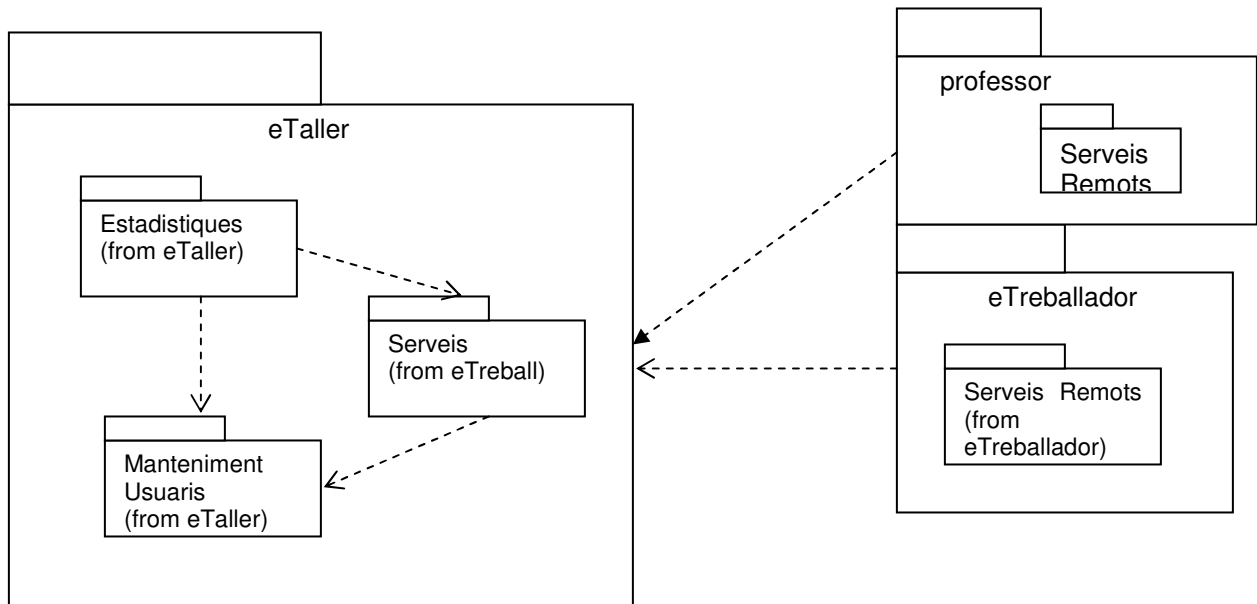
El conjunt de funcionalitats que tindrà s'han distribuït en 4 subsistemes:

- Subsistema de manteniment
- Subsistema de serveis
- Subsistema de seguiment
- Subsistema de connexió i serveis remots

Cal destacar que les funcionalitats també estan determinades pel perfil d'usuari:

- Administrador
- Professor
- Treballador

4.3.1. Diagrama de paquets



4.3.2. Diagrames UML

Diagrames de classes i jerarquies. Diagrama de classes del Subsistema

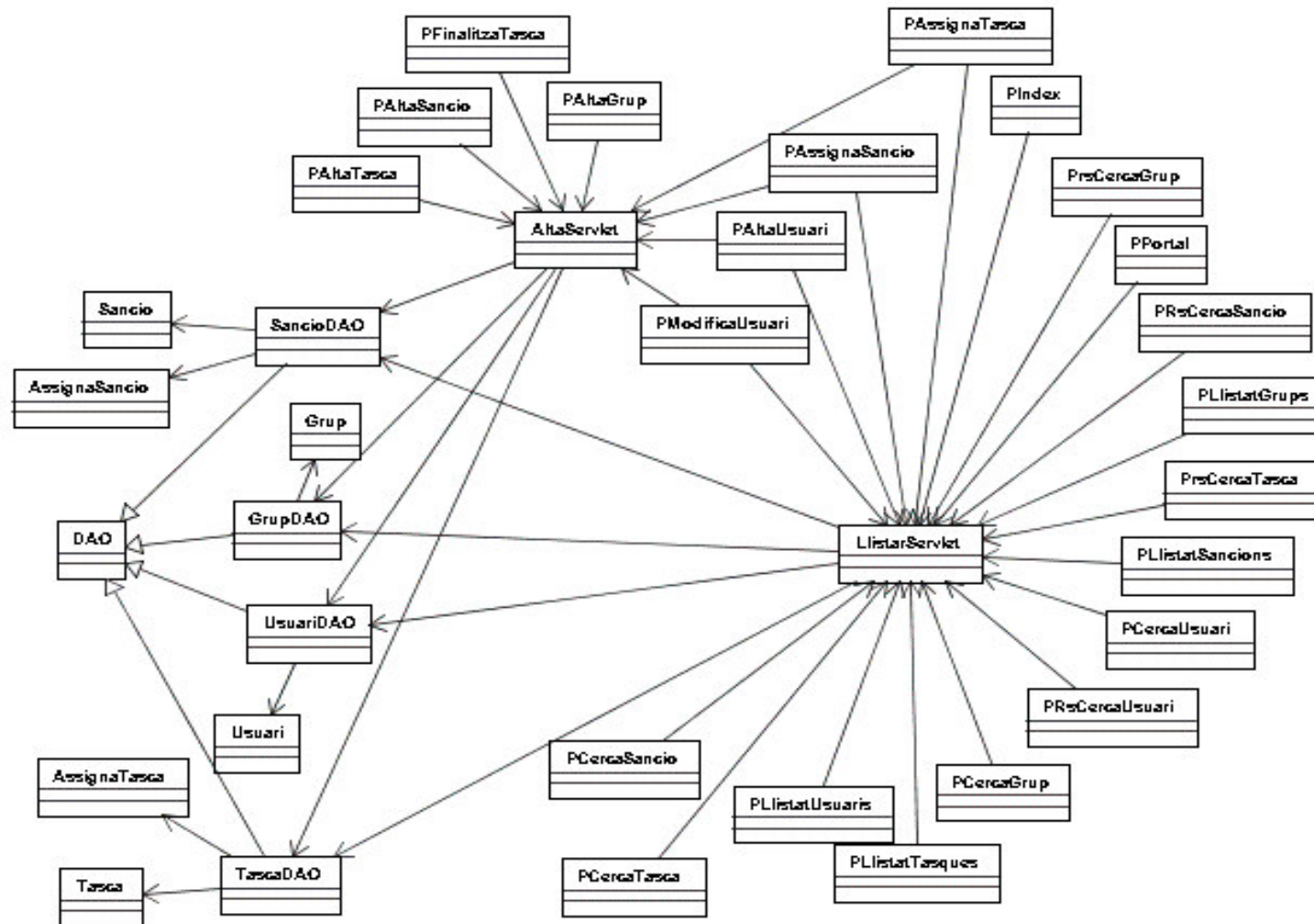


Diagrama de Gestors

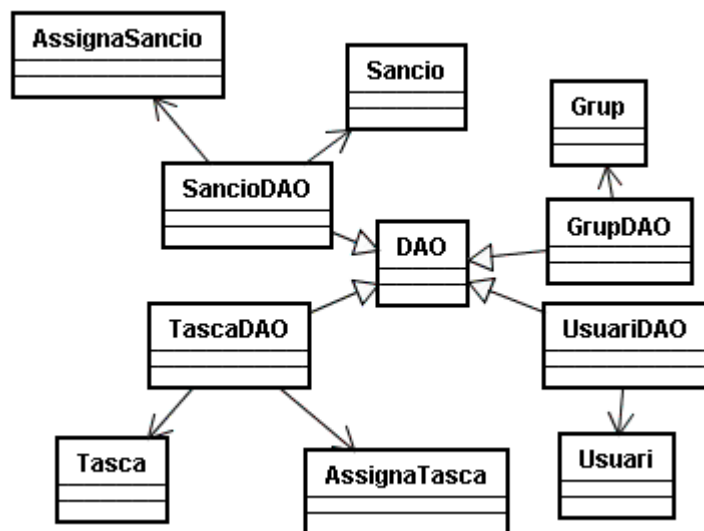
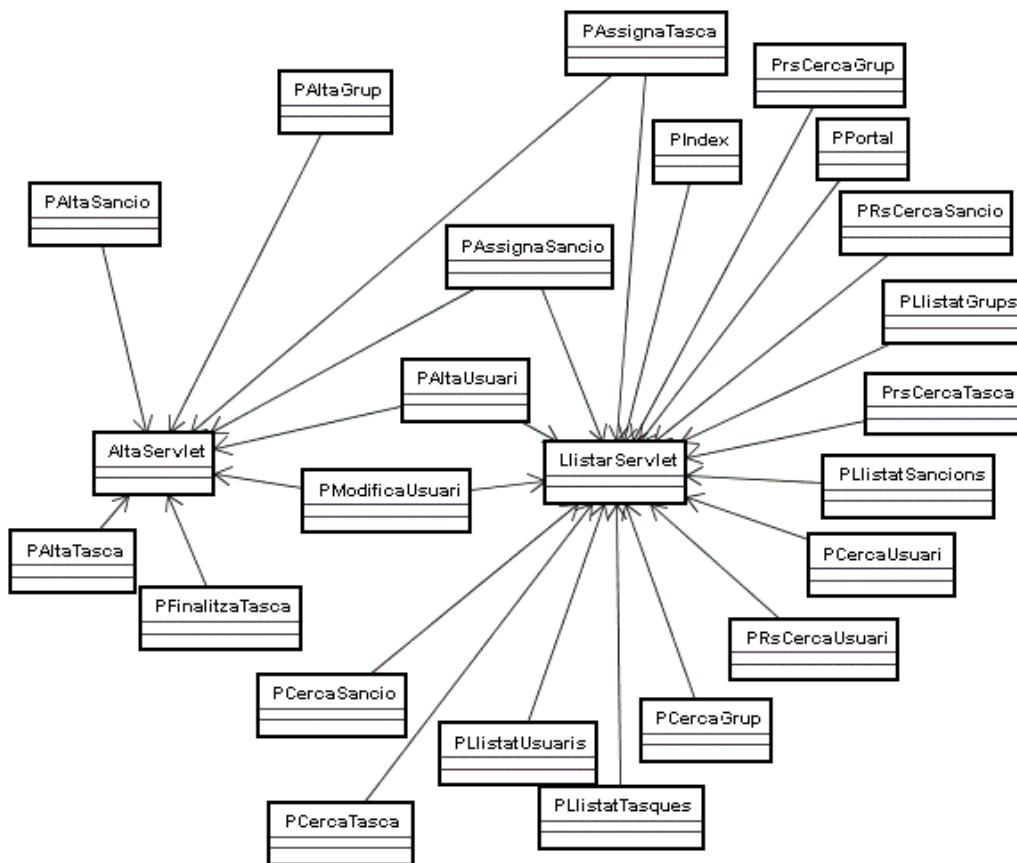


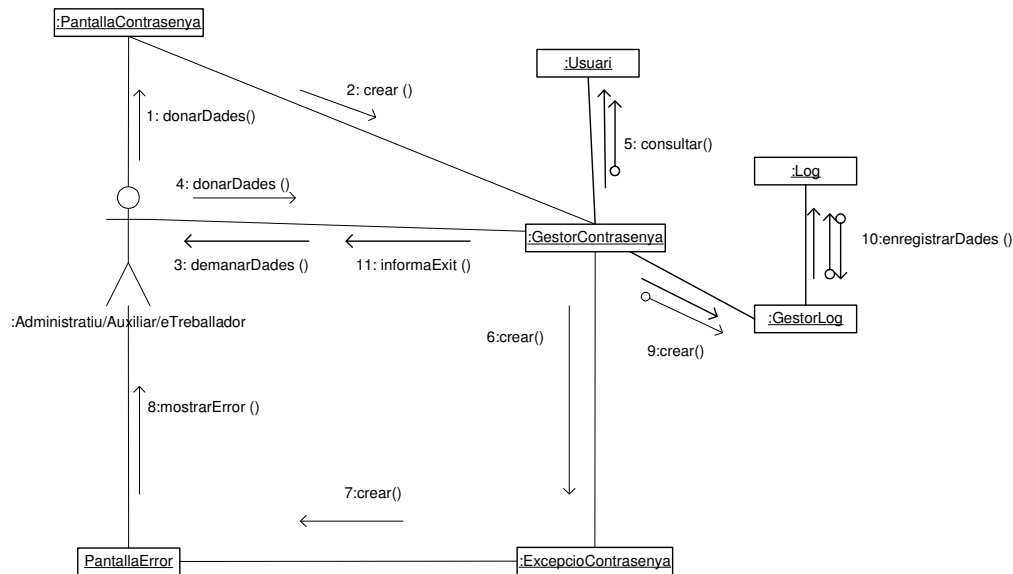
Diagrama de Pantalles



4.3.3. Diagrames de col·laboració i seqüència

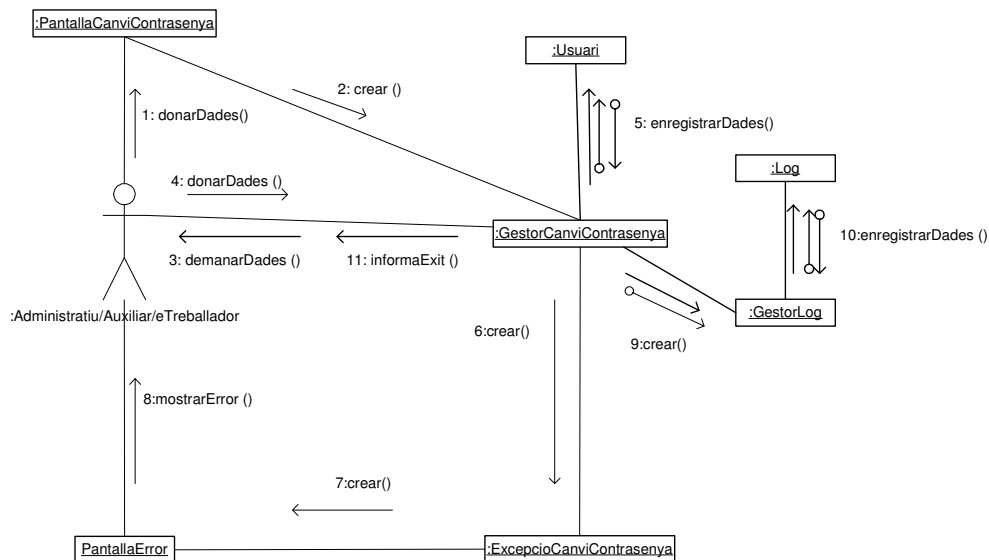
Validar-se al sistema

Tot usuari que vulgui accedir al sistema haurà de validar-se o sigui tenir un codi d'usuari i una contrasenya, si les dades entrades són correctes entrarà sense cap problema, en cas contrari tindrà dos intents més si no ho aconsegueix, llavors l'usuari quedarà bloquejat i tindrà que demanar un nova contrasenya a l'administrador.



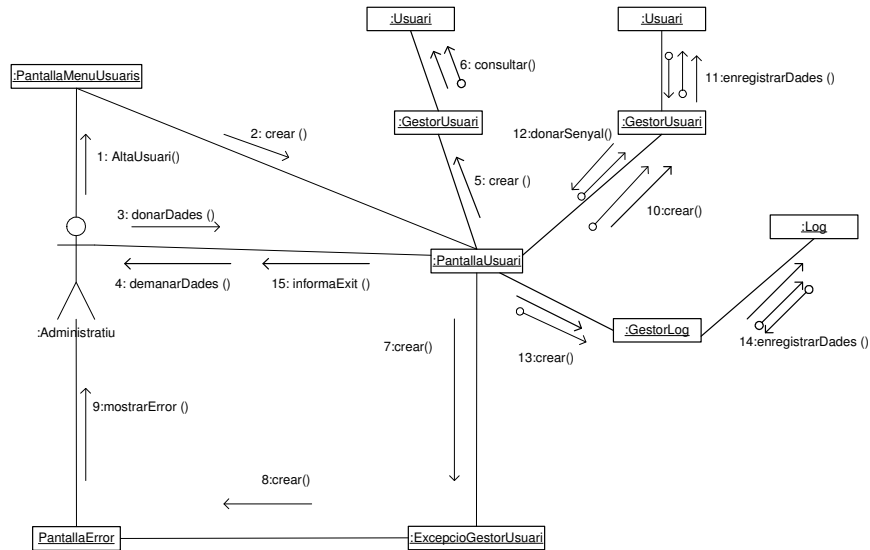
Canviar de Contrasenya(només administrador)

L'administrador podrà canviar la contrasenya dels usuaris des de l'opció, modificació d'usuaris.



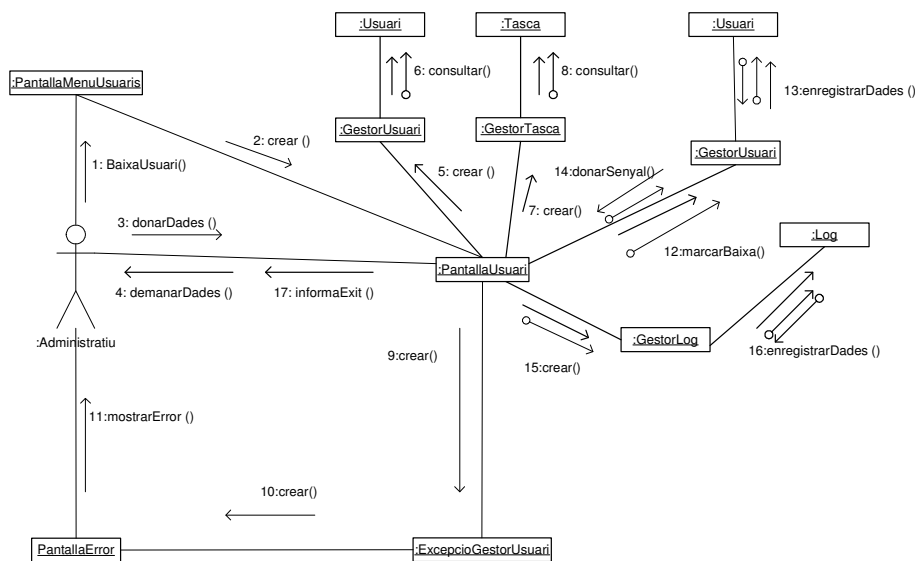
Alta d'un Usuari(només administrador)

Per fer una alta haurem de seleccionar l'opció del Portal, ens mostrarà AltaUsuari, amb tots els camps per omplir, entrarem un DNI d'usuari(no existent) i seguidament totes les dades.



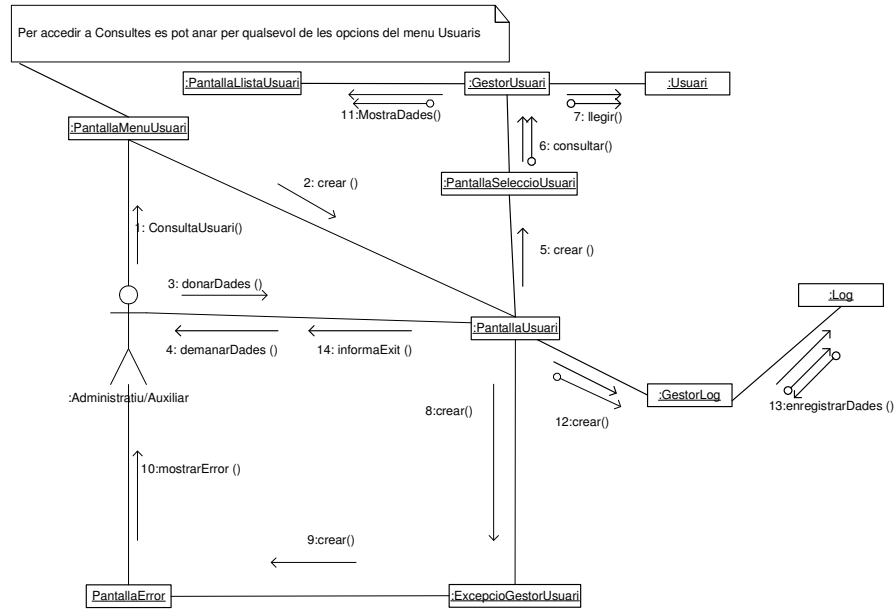
Baixa d'un Usuari(només administrador)

En introduir el codi de l'usuari a donar de baixa, sens mostraran les dades del mateix en cas de que estem segurs de que sigui l'usuari correcte acceptarem i serà eliminat.



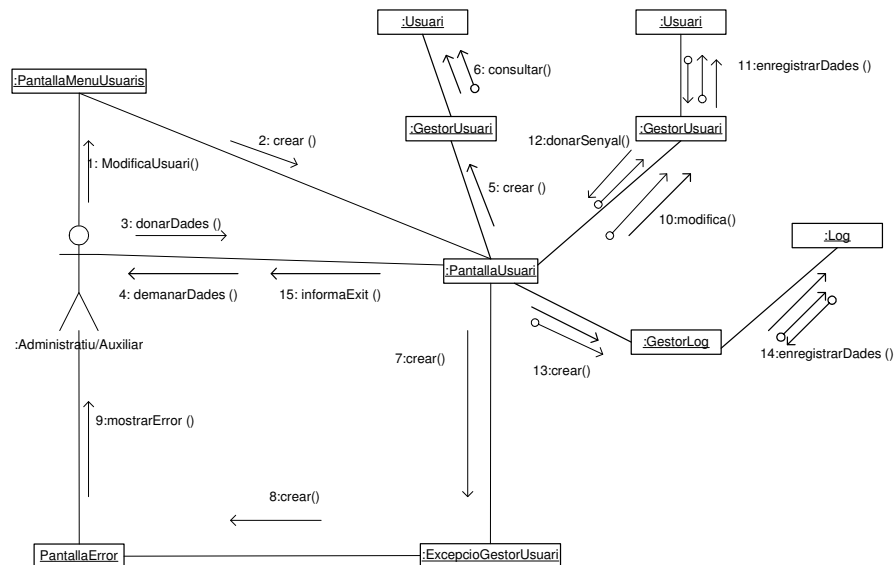
Consulta d'un Usuari o varis

S'ha d'introduir el DNI de l'usuari a cercar. Si la consulta ha estat satisfactòria ens mostrarà una nova finestra rsCercaUsuari, amb les dades de l'usuari cercat. Se'n mostraran les opcions d'eliminar i modificar.

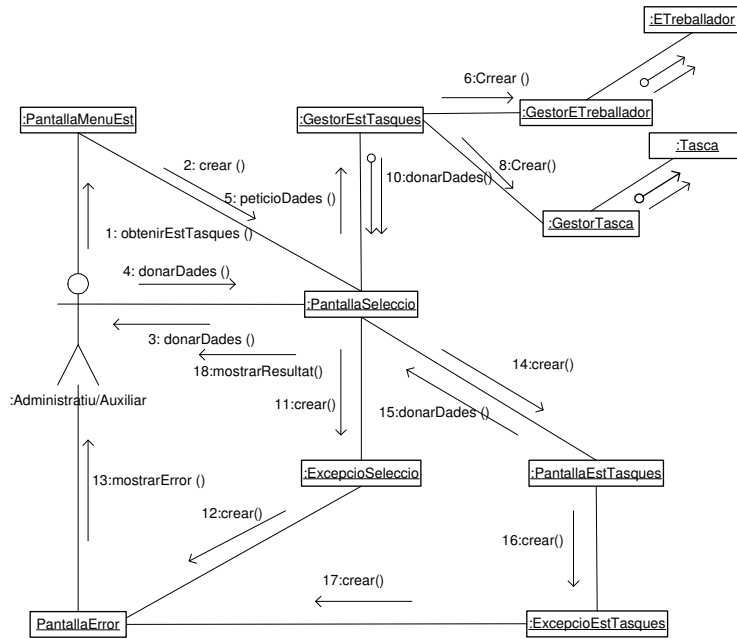


Modificació d'Usuari

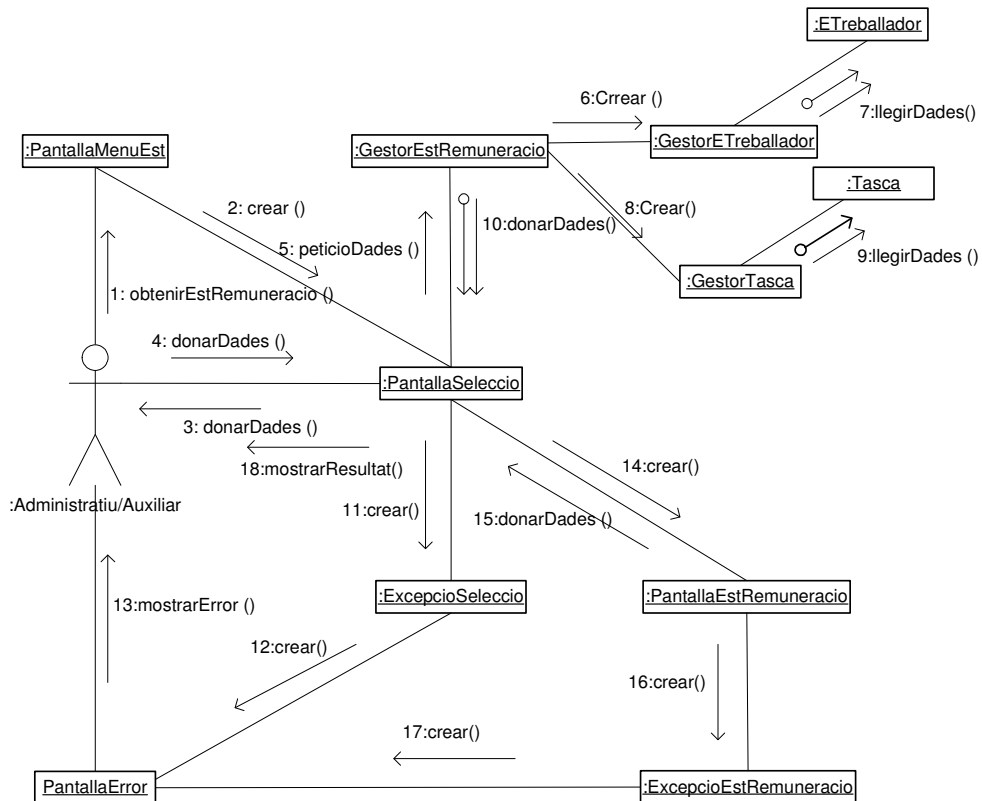
Introduïrem el DNI d'usuari, si existeix ens mostrarà les dades en la rsCercaUsuari i seleccionarem l'opció modificar. A continuació fem les modificacions pertinents i guardarem els canvis realitzats.



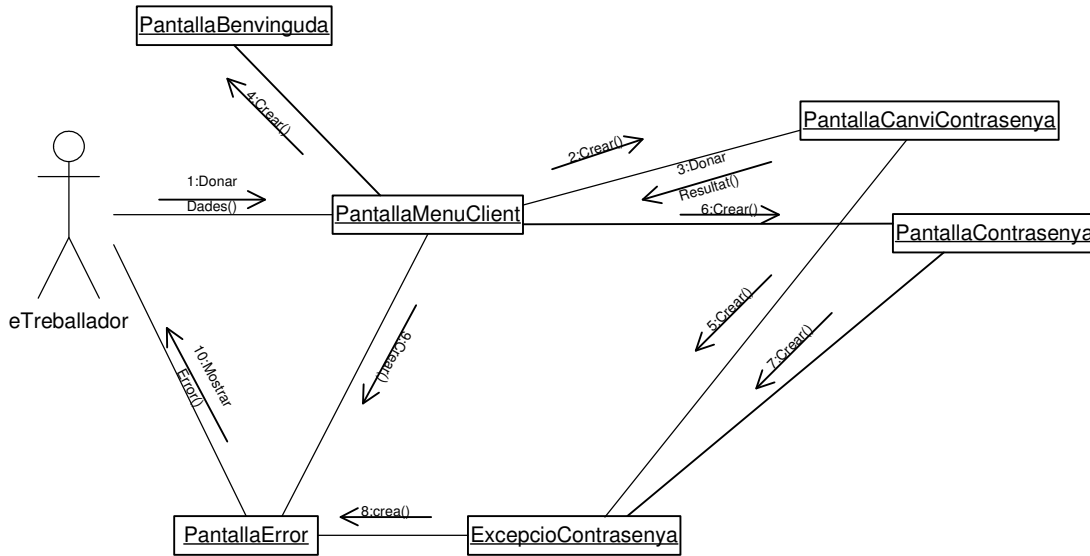
Tasques assignades o finalitzades.



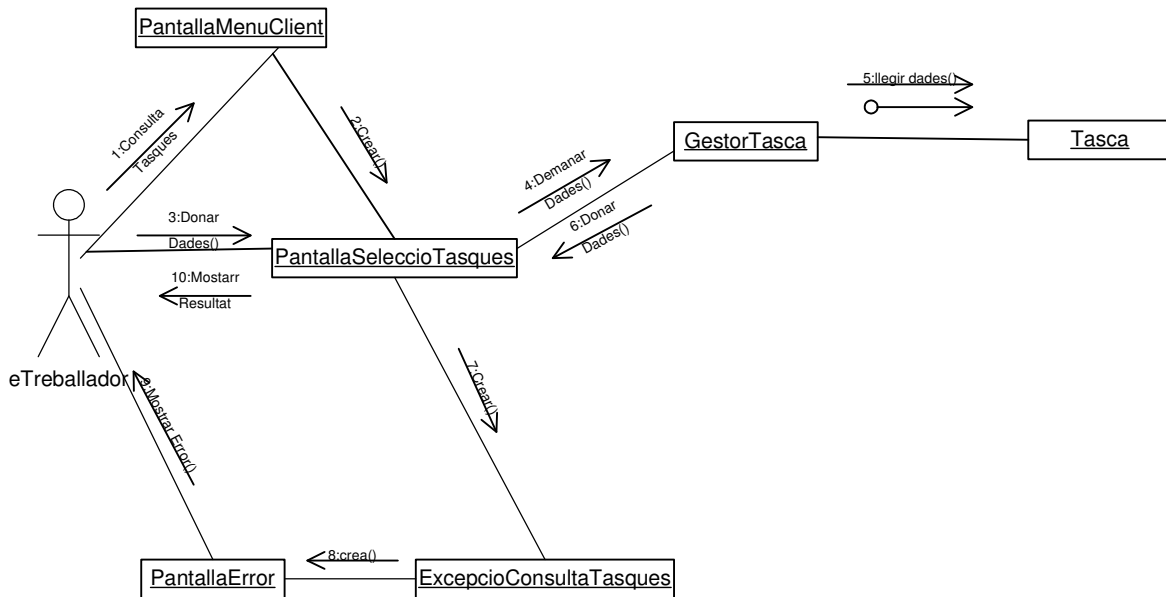
Càlcul remuneració (no implementada).



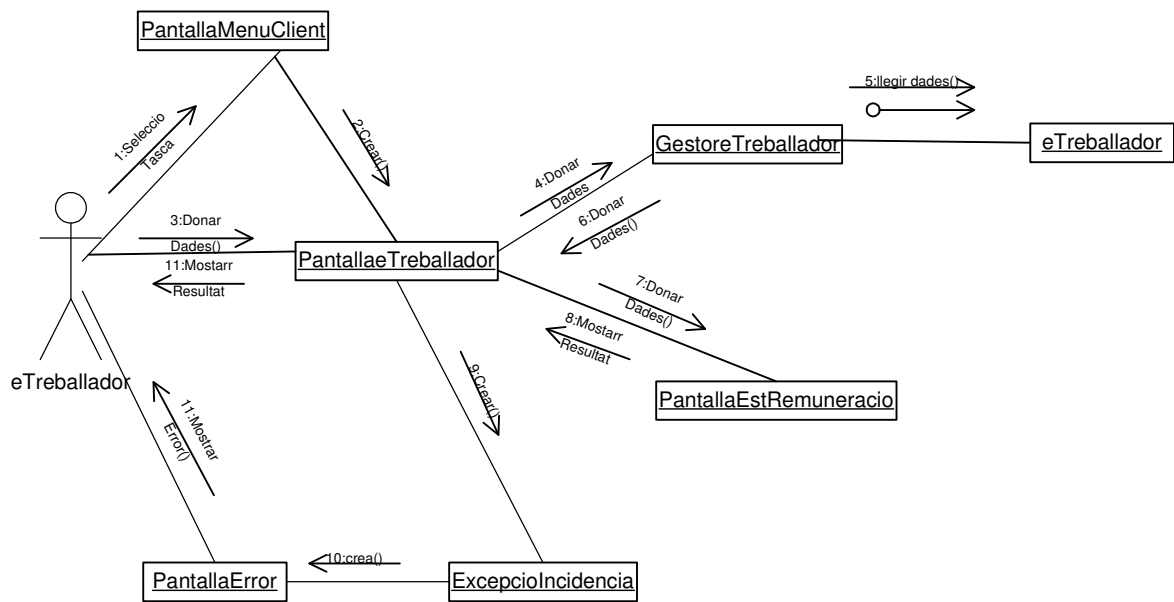
Connexió al Sistema



Consulta de Tasques Assignades



Consulta Dades Personals



4.3.4. Interfície gràfica d'usuari.

Index.jsp

Pantalla d'identificació. S'ha d'introduir el DNI i la contrasenya per a poder accedir a l'aplicació.



Escola Taller Ilercavònia 

Benvingut :: Identificació

DNI:

Contrassenya:

Acceptar

4.3.4.1. Interfícies d'administrador.

Portal1.jsp

Pàgina d'inici per a administradors, on es mostra el menú d'accions.



Escola Taller Ilercavònia 

Benvingut :: Menú d'Opcions

:: Llistats	:: Altes
Llistar Usuaris	Alta Usuari
Llistar Grups	Alta Grup
Llistar Sancions	Alta Sancio
Llistar Tasques	Alta Tasca
:: Recerques	:: Baixes
Cerca Usuari	Baixa Usuari
Cerca Grup	Baixa Grup
Cerca Sancio	Baixa Sancio
Cerca Tasca	Baixa Tasca
:: Modificacions	:: Altres
Modificar Usuari	

logout

[tornar a l'inici](#)

© Gonçal Verge
gverge@uoc.edu
ETIG a la UOC

A partir d'ara continuarem amb les opcions d'usuari, totes les demés funcionalitats segueixen el mateix patró de disseny, i l'aspecte es most similar.

AltaUsuari.jsp

Pàgina que permet omplir el formulari d'alta d'un usuari. La casella de grup conté una llista de grups prèviament introduïts.

LlistatUsuaris.jsp

Pàgina que mostra tots els usuaris de l'escola sense diferenciació de perfils.

::DNI Usuari	::Nom	::Cognoms
0000000G	Administrador	Inicial
0000001G	Profe1	Modul1
0000002G	Profe2	Modul2
0000003G	Alumne1	Modul1
0000004G	Alumne2	Modul2

© Gonçal Verge
gverge@uoc.edu
ETIG a la UOC

CercaUsuari.jsp

Pàgina que permet cercar un usuari introduint el seu DNI a la casella i prement acceptar.

Escola Taller Hecavònia

::Cerca d'Usuaris::

DNI de l'Usuari:

Acceptar

[tornar a l'inici](#)

© Gonçal Verge
gverge@uoc.edu
ETIG a la UOC

rsCercaUsuari.jsp

Pàgina que mostra les dades de l'usuari cercat prèviament, es mostren les opcions d'eliminar i modificar a la part inferior.

Escola Taller Hecavònia

::Fitxa d'Usuari::

DNI:	0000001G
Contrassenya:	PASS1
Nom:	Profe1
Cognoms:	Modul1
Tipus Usuari:	P
Data d'Alta:	2005-03-03
Grup:	I

[eliminar / modificar](#)

[tornar a l'inici](#)

© Gonçal Verge
gverge@uoc.edu
ETIG a la UOC

Aquesta pàgina també s'utilitzarà per a mostrar les dades personals dels professors o dels treballadors en prémer l'opció indicada. L'acció del servlet serà similar en els tres casos, ja que buscarà l'usuari pertinent i mostrarà els resultats en la pàgina rsCercaUsuari.jsp.

modificaUsuari.jsp

Pàgina que mostra les dades de l'usuari cercat prèviament, en aquest cas però, les dades poden ser modificades i introduïdes a la base de dades.

Escola Taller Ilercavònia 

Formularis :: Modificació Usuari

DNI:	<input type="text" value="0000001G"/>
Assignar Contrassenya:	<input type="text" value="PASS1"/>
Nom:	<input type="text" value="Profe1"/>
Cognoms:	<input type="text" value="Modul1"/>
Tipus Usuari:	<input type="text" value="P"/> <small>(A, P o T)</small>
Data d'Alta:	<input type="text" value="2005-03-03"/>
Grup:	<input type="text" value="I"/>

4.3.4.2. Interfícies de Professor.

Portal2.jsp

Pàgina d'inici per a professors, on es mostra el menú d'accions.

Escola Taller Ilercavònia 

Benvingut :: Menú d'Opcions

Opcions de Professor

:: Dades Personals	:: Assignacions
Les meves dades	Assignar Tasca
Components del Grup	Finalitzar Tasca
	Assignar Sancio
:: Historial Tasques	:: Historial Sancions
Tasques Finalitzades	Sancions Assignades
Tasques Assignades	

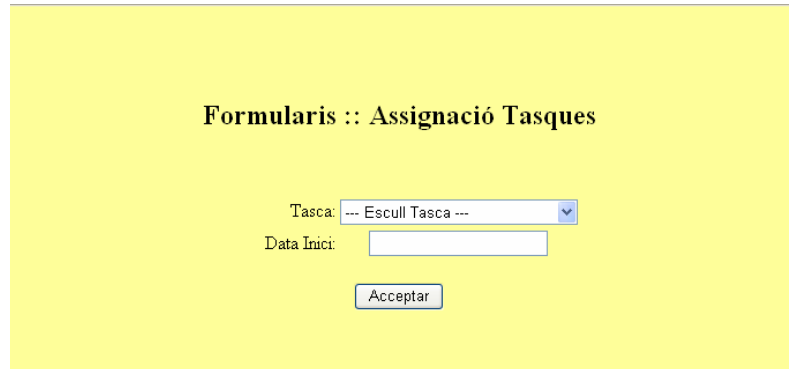
logout

[tornar a l'inici](#)

© Gonçal Verge
gverge@uoc.edu
ETIG a la UOC

AssignaTasca.jsp

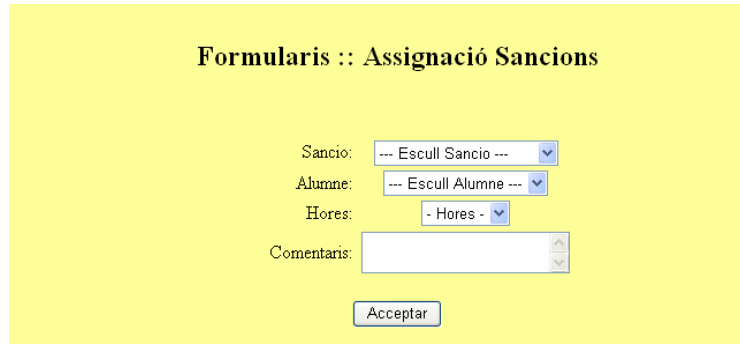
Pàgina on es pot assignar una tasca al grup del qual l'usuari es professor tutor. Només cal seleccionar la tasca(prèviament donada d'alta per l'administrador) i introduir la data d'inici.



The screenshot shows a web form titled "Formularis :: Assignació Tasques" on a yellow background. It contains a dropdown menu for "Tasca" with the text "--- Escull Tasca ---", a text input field for "Data Inici:", and an "Acceptar" button.

AssignaSancio.jsp

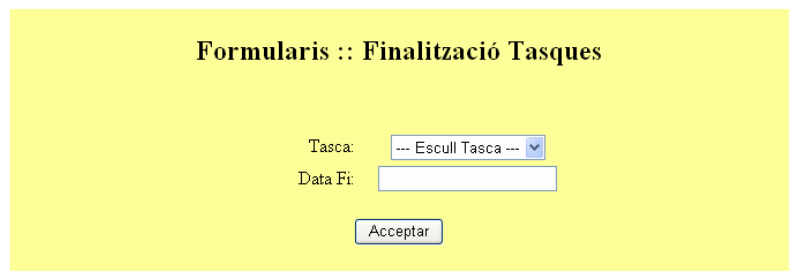
Pàgina on es pot assignar una sanció a un alumne determinat del qual l'usuari es professor tutor. Només cal seleccionar la sanció(prèviament donada d'alta per l'administrador), el treballador sancionat, les hores de sanció i introduir els comentaris pertinents.



The screenshot shows a web form titled "Formularis :: Assignació Sancions" on a yellow background. It contains four dropdown menus: "Sancio:" with "--- Escull Sancio ---", "Alumne:" with "--- Escull Alumne ---", "Hores:" with "- Hores -", and "Comentaris:" with a scrollable area. Below the dropdowns is an "Acceptar" button.

FinalitzaTasca.jsp

Pàgina on es pot donar per finalitzada una tasca del grup del qual l'usuari es professor tutor. Només cal seleccionar la tasca(prèviament assignada al grup) i introduir la data de finalització.



The screenshot shows a web form titled "Formularis :: Finalització Tasques" on a yellow background. It contains a dropdown menu for "Tasca" with the text "--- Escull Tasca ---" and a text input field for "Data Fi:". Below the input fields is an "Acceptar" button.

LlistatSancions.jsp

Pàgina on es mostren les sancions assignades pel professor que ha iniciat sessió a tots els alumnes del seu mòdul. Hi consta el codi de la sanció i els comentaris pertinents (alumne sancionat, tipus de sanció i altres dades).

::Llistat de Sancions	
::CodiSancio	::Descripcio
S01	Falta Assistencia Llieu (Alumne1 Modul1 - 6 hores) - 20/12/2005
S02	Falta Assistencia Greu (Alumne1 Modul1 - 8 hores) - 21/12/2005 - Reincidència

Aquesta pàgina es fa servir per a tots els llistats de sancions, ja sigui des del perfil d'administrador a l'hora de llistar totes les sancions del sistema, com del professor per a comprovar les sancions assignades, com del treballador per a visualitzar les seves sancions personals. El servlet fa accions diferents però envia les dades al mateix destí: LlistatSancions.jsp.

LlistatTasques.jsp

Pàgina on es mostren les tasques assignades pel professor que ha iniciat sessió al grup. Hi consta el codi de la tasca, el títol, la descripció (depenent de l'estat de la tasca hi constaran uns comentaris o uns altres) i la durada que té en principi la tasca.

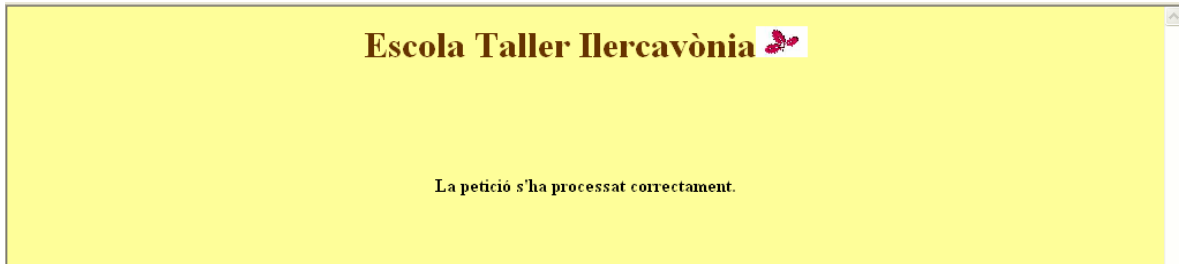
::Llistat de Tasques::			
::CodiTasca	::Titol	::Descripcio	::Durada
IM1	Arquitectura d'Ordinadors	(2005-12-25)	100
IM2	Xarxes	(2005-12-01)	50

Aquesta pàgina es fa servir per a tots els llistats de tasques, ja sigui des del perfil d'administrador a l'hora de llistar totes les tasques del sistema, com del professor per a comprovar les tasques assignades, com del treballador per a visualitzar les seves tasques com a membre de grup. També cal mencionar que es fa servir tant per al llistat de tasques assignades al grup, com per a les finalitzades. El servlet fa accions diferents però envia les dades al mateix destí: LlistatTasques.jsp.

4.3.4.3. Interfícies Generals.

ok.html

Pàgina mostrada com a resposta en qualsevol petició que es processi correctament.



errorValidacio.html

Pàgina que indica un error en la validació de l'usuari.



error.jsp

Pàgina establerta com a excepció a l'hora d'executar el codi de la pàgina. S'utilitza amb la directiva `<%@page errorPage="error.jsp"%>`.



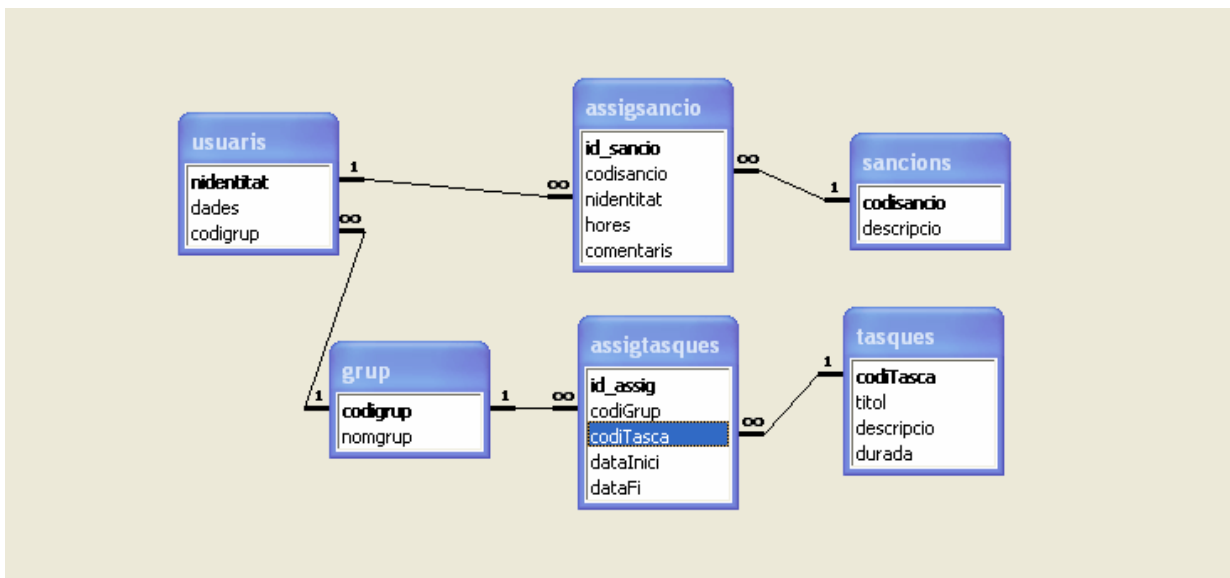
4.3.5. Persistència.

Disseny de la persistència

Per aconseguir la persistència de les dades crearem un conjunt d'objectes que utilitzarem com a instàncies dels elements de les taules a l'hora de emmagatzemar o recuperar les dades de la base de dades i mostrar-les a l'aplicació web.

Totes les accions d'accés i manipulació de dades les realitzarem mitjançant els objectes gestors (DAO) que cada classe te implementats. Això fa que l'aplicació pugui funcionar independentment de la ubicació de les dades i per tant es pugui desvincular de la base de dades original i treballar amb altres.

Diagrama de la base de dades



Gestors de la persistència

Utilitzem una sèrie de gestors de dades per a poder accedir a la BD desvinculant-la però de l'aplicació, així el treball amb les dades es produirà de forma totalment independent.

La classe principal és *DAO* que permet establir una connexió a la base de dades i crear sentències per interactuar-hi.

La resta d'objectes d'accés a dades són subclasses de *DAO* que es cuiden de guardar la informació de cadascuna de les classes del model a la base de dades: *UsuariDAO*, *SancioDAO*, *TascaDAO*, *GrupDAO*.

Així doncs per afegir, esborrar, cercar o qualsevol altra acció relacionada amb la classe *usuaris*, es crearà un objecte *UsuariDAO*, que gestionarà totes aquestes accions i el qual crearà un *DAO* per a connectar amb la BD.

Altres aspectes

La tecnologia escollida per a la confecció de l'aplicació web ha estat J2EE, treballant amb servlets i pàgines JSP, això ens proporciona diferents possibilitats molt interessants de les quals destacarem:

Sessions

Pel que fa a la persistència de dades i degut a la fragilitat d'aquesta en entorns web, les sessions han estat un gran aliat alhora de treballar dades que no canvien durant tota una sessió de treball. Així doncs en iniciar el treball un usuari qualsevol, el servlet, mitjançant els gestors corresponents crea una sessió (que no es destrueix fins que l'usuari acaba el seu treball) on emmagatzema l'objecte Usuari corresponent al propi usuari que ha iniciat la sessió.

D'aquesta manera en totes les accions que implícitament necessiten consultar dades com el grup o el tipus d'usuari, només han de rescatar-la de l'objecte creat a la sessió actual.

Patrons

L'arquitectura de patrons mvc ens permet gestionar de forma separada les dades amb que treballa l'aplicació i la presentació de les mateixes. Per a que aquest model funcioni correctament tot el pes recau en els gestors, dels quals ja em parlat anteriorment. Els gestors treballen amb la informació en la capa de dades i un cop preparades d'acord amb les peticions de l'usuari les envien a les pàgines dinàmiques per a ser representades.

En el cas d'eTreball la petició que fa l'usuari es recollida per un dels dos servlets implementats segons la seva naturalesa (listarServlet i altaServlet) que gestiona la petició i crida el DAO corresponent a l'objecte sobre el qual es vol treballar (usuari, grup, sanció o tasca). Un cop el gestor ha finalitzat el tractament de la informació, retorna la responsabilitat al servlet que es l'encarregat d'enviar la informació corresponent a la pàgina resultant (ja sigui informació corresponent als objectes, errors o simplement un comprovant).

En una aplicació d'aquest tipus podem amés de les opcions rutinàries (altes, baixes, consultes) podem necessitar funcions de preparació de dades (per a omplir desplegable) comprovacions de dades... per aquest motiu s'ha decidit confeccionar dos servlets per separat, un per a la edició de la BD i un altre per a tot tipus de consultes.

4.3.6. Reutilització

La tecnologia JAVA implica en tots els casos una aposta per la reutilització, en aquest aspecte s'ha intentat seguir aquesta política utilitzant en els casos en que ha estat possible funcions, classes... ja implementades i provades.

Utilitzarem la llibreria estàndard de Java per diverses coses:

- Paquets habituals: les classes per al tractament de cadenes de text, per a càlculs numèrics, les classes contenidors d'objectes, ...
- Servlets i JSP, que ens permetran que la nostra aplicació s'integri en un servidor web compatible amb J2EE.
- JDBC, que ens permetrà connectar amb una BD, oblidant les peculiaritats del sistema gestor de bases de dades utilitzat.

Algunes classes , s'han modificat per a l'ocasió essent extretes en el seu dia de la biblioteca de components TDP.

També convé destacar que utilitzarem dues eines que ja existeixen per a la fase d'implementació, i la posterior fase de desplegament de l'aplicació:

- Tomcat: un servidor web que segueix l'especificació J2EE, lliure i gratuït.
- MySQL: Sistema gestor de bases de dades lliure i gratuït.

4.4. Implementació

Detallarem ara la implementació que s'ha fet sobre el disseny anterior.

Tot el material resultant (jsp, html i java) està empaquetat en eTaller.war, que ha generat el netbeans.

El paquet eTaller

En obrir el paquet ens podem trobar amb una estructura similar a una web, amb les pàgines estàtiques i dinàmiques al primer nivell, juntament amb les imatges utilitzades en el disseny de les interfícies.

4.4.1. Pàgines dinàmiques i estàtiques

Cal destacar *index.jsp*, que és la pàgina principal d'entrada a l'aplicació (seguint l'estàndard utilitzat per la majoria de servidors).

També cal mencionar els html *cap* i *peu* que son utilitzats en totes les pàgines jsp com encapçalament i peu de pàgina, mitjançant la directriu `<%@include file="cap.htm"%>` molt útil a l'hora de dissenyar les interfícies, així doncs podem parlar d'una espècie de patrons de disseny, ja que en fer qualsevol canvi a l'arxiu cap.html o peu.html totes les pàgines jsp que fan un include d'aquests son modificades immediatament.

També la directriu de pàgina `<%@page errorPage="error.jsp"%>` ens permet fer un direccionament a la pàgina error.jsp sempre que es produeixi un error en l'execució del codi de la pàgina.

La possibilitat que dona la tecnologia jsp d'introduir retalls de codi java i executar-los des de la mateixa pàgina ha estat una bona manera de tractar les sessions d'usuari i els permisos de les mateixes.

D'aquesta forma, i per no carregar més els servlets, quan un usuari entra a l'aplicació i es vol fer una comprovació de quin tipus d'usuari es per a portar-lo a la seva pàgina d'opcions s'envia tots els usuaris contra portal1.jsp i es la mateixa pàgina la que s'encarrega de gestionar els accessos, fent una comprovació de la informació emmagatzemada en la sessió iniciada de la forma següent:


```
<%  
String tipusUsuari="";  
HttpSession sesionOk=request.getSession();  
System.out.println(sesionOk.getAttribute("tipusUsuari"));  
if(sesionOk.getAttribute("tipusUsuari").equals("T")){%>  
    <jsp:forward page="Portal3.jsp">  
    <jsp:param name="limitat" value="Trellador"/>  
    </jsp:forward>  
<%}else if(sesionOk.getAttribute("tipusUsuari").equals("P")){%>  
    <jsp:forward page="Portal2.jsp">  
    <jsp:param name="limitat" value="Professor"/>  
    </jsp:forward>  
<%}else  
    tipusUsuari=(String)sesionOk.getAttribute("tipusUsuari");%>
```

A l'hora d'escollir una opció s'ha seguit dos tipus de funcionament per una banda tenim l'accés directe a la pàgina sol·licitada:

```
<a href="AltaGrup.jsp">Alta Grup</a>
```

O en el cas de que s'hagi de preparar algun quadre de selecció, com en el cas de l'alta d'usuari (on es crea una llista desplegable amb els grups existents) es crida al servlet per a que prepari la llista i després direccioni l'usuari cap a la pàgina destí:

```
<a href="LlistarServlet?accioAlta=s" name="altaUsuaris">Alta Usuari</a>
```

També podem trobar retalls de codi on es comprova quin tipus d'usuari ha iniciat la sessió per a decidir si mostrar o no alguna opció:

```
<%  
String tipusUsuari="";  
HttpSession sesionOk=request.getSession();  
System.out.println(sesionOk.getAttribute("tipusUsuari"));  
if(sesionOk.getAttribute("tipusUsuari").equals("T")){  
else{%>  
    <a href="AltaServlet?baixaUsuari=${usuari.nidentitat}"  
name="baixaUsuari">eliminar</a> /  
    <a href="LlistarServlet?modificaUsuari=${usuari.nidentitat}"  
name="modificaUsuari">modificar</a>  
    <%}%>
```

I la recuperació d'informació de sessió en cridar el servlet per a demanar una informació determinada:

```
<a href="LlistarServlet?cercaUsuariP=<%=nidentitat%>" name="llistarUsuaris">Les  
meves dades</a>
```

Pel que fa a la presentació de dades en les pàgines dinàmiques podem trobar retalls de codi com el següent,

```
c:forEach items="{requestScope.llistaUsuaris}" var="usuariActual">
  <tr>
    <td>${usuariActual.nidentitat}</td>
    <td>${usuariActual.nom}</td>
    <td>${usuariActual.cognoms}</td>
    <td>&nbsp;</td>
  </tr>
</c:forEach>
```

On es fa un recorregut pels elements de llistaUsuaris, un vector resultant del mètode corresponent del servlet, que ja veurem més endavant.

Casos com el següent per a emplenar un formulari de modificació on es mostren dades en caselles de text,

```
<td><div align="right">Nom:</div></td>
<td><input type="text" name="nom" value=${usuari.nom}></td>
</tr>
<tr>
  <td><div align="right">Cognoms:</div></td>
  <td><input type="text" name="cognoms" value=${usuari.cognoms}></td>
```

agafats des d'un objecte usuari rebut desde el servlet.

O llistes de selecció que s'han d'emplenar amb objectes existents a la BD que es reben des del servlet,

```
select name="grup" size="1">
  <option value="0">
    --- Escull Grup ---
  </option>
  <c:forEach items="{requestScope.llistaGrups}" var="grupActual">
    <option value="{grupActual.codigGrup}">
      ${grupActual.nomGrup}
    </option>
  </c:forEach>
</select>
```

4.4.2. Les classes principals

Quan parlàvem de persistència hem fet referència a la importància de crear una sèrie d'objectes necessaris per a representar les dades en una entitat de la naturalesa apropiada. En aquest cas estem parlant de quatre classes principals per a l'aplicació:

Usuari

Representa l'objecte usuari, que es el principal en l'aplicació, podem trobar-hi atributs que recuperarà de la taula usuaris de la BD tals com:

```
private String nidentitat;  
private String contrassenya;  
private String tipusUsuari;  
private String dataAlta;  
private String nom;  
private String cognoms;  
private String codiGrup;
```

i altres que ens poden servir per a futures millores d'implementació. Els seus constructors i els mètodes d'accés.

Tasca

Representa l'objecte tasca, que fa referència als treballs a realitzar per un grup de treballadors de l'escola, potser que una tasca faci referència a un treball en el que poden participar dos grups diferents i per tant els assignarem mitjançant una altra taula, assignasques. Podem trobar-hi atributs que recuperarà de la taula usuaris de la BD tals com:

```
private String codiTasca;  
private String titol;  
private String descripcio;  
private Integer durada;
```

i altres que ens poden servir per a futures millores d'implementació. Els seus constructors i els mètodes d'accés.

Sancio

Representa l'objecte sanció, que fa referència a les sancions assignades als alumnes, en principi son creades com a generals (exp. falta lleu) però necessiten una taula assignSancio per a ser assignades a un alumne concret i amb un nombre d'hores determinades. Podem trobar-hi atributs que recuperarà de la taula usuaris de la BD tals com:

```
private String codiSancio;  
private String descripcio;
```

i altres que ens poden servir per a futures millores d'implementació. Els seus constructors i els mètodes d'accés.

Grup

Representa l'objecte grup, que fa referència als grups en els que es pot agrupar els usuaris. Podem trobar-hi atributs que recuperarem de la taula usuaris de la BD tals com:

```
private String codiGrup;  
private String nomGrup;
```

i altres que ens poden servir per a futures millores d'implementació. Els seus constructors i els mètodes d'accés.

AssignaSancio

Representa una assignació d'una sanció a un usuari, i correspon directament amb la taula assignaSancio de la BD, per tant a l'hora d'assignar una sanció a un Usuari determinat es crearà un objecte d'aquest tipus que quedarà enregistrat com una línia a la taula corresponent. Podem trobar-hi atributs que recuperarem de la taula usuaris de la BD tals com:

```
private String codiSancio;  
private String nidentitat;  
private Integer hores;  
private String comentaris;
```

i altres que ens poden servir per a futures millores d'implementació. Els seus constructors i els mètodes d'accés.

AssignaTasca

Representa una assignació d'una tasca a un grup, i correspon directament amb la taula assignTasques de la BD, per tant a l'hora d'assignar una tasca a un Grup determinat es crearà un objecte d'aquest tipus que quedarà enregistrat com una línia a la taula corresponent. Podem trobar-hi atributs que recuperarem de la taula usuaris de la BD tals com:

```
private Integer idAssignacio;  
private String codiGrup;  
private String codiTasca;  
private String dataInici;  
private String dataFi;
```

i altres que ens poden servir per a futures millores d'implementació. Els seus constructors i els mètodes d'accés.

4.4.3. Els gestors

Son els encarregats de gestionar les dades de la BD, exercint d'intermediaris entre aquesta i els objectes concrets. Per tant contindran els mètodes d'afegir, esborrar, llistar, modificar... les taules de la base de dades, sempre interactuant amb els objectes principals que seran l'origen o el destí de les dades. Podem trobar un gestor per a cada classe principal, que hereten d'un gestor principal, encarregat de fer la connexió amb la base de dades. Així doncs:

DAO

Es el gestor principal del qual tots hereten per a poder establir la connexió amb la base de dades. Conte només un mètode:

```
public Connection getConnexio(){
    Connection con=null;
    try{
        if(registrat==false){
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            registrat=true;
        }
    }
}
```

Retorna un objecte connection amb la connexió ja establerta.

UsuariDAO

Es el gestor d'objectes del tipus usuari, l'utilitzarem com a exemple ja que és el que més interactua amb la BD i servirà per a fer-nos una idea de com funciona la resta de gestors.

Inclou diferents mètodes d'accés a la BD de dades amb la qual connecta gràcies al mètode definit anteriorment a la seva classe mare DAO. Podem destacar:

getUsuaris()

que agafa tots els usuaris de la taula corresponent a la BD.

```
try{
    con=super.getConnexio();
    stmt=con.createStatement();
    sql="SELECT * FROM usuaris;";
    rs=stmt.executeQuery(sql);//nomes val per al select, si es vol agragar camps...
    executeUpdate
    while(rs.next()==true){
        usr=new Usuari();
        usr.setNidentitat(rs.getString("nidentitat"));
        usr.setNom(rs.getString("nom"));
        usr.setCognoms(rs.getString("cognoms"));
        Usuaris.add(usr);
    }
}
```

i els emmagatzema en un vector d'objectes usuari que després s'enviarà al servlet i aquest al seu temps enviarà a la pàgina corresponent per a ser representats.

getUsuarisGrup(String codiGrup){

que agafa tots els usuaris de la taula corresponent que pertanyen al grup amb el codi corresponent a l'string rebut a codiGrup. Només canvia la sentència sql.

```
sql="SELECT * FROM usuaris WHERE codigrup LIKE '%" + codiGrup + "%";
```

getCercaUsuari(String dni)

fa una cerca acurada d'un usuari concret de la taula amb dni corresponent a l'string rebut a dni.

```
sql="SELECT * FROM usuaris WHERE nidentitat LIKE '%" + dni + "%";
```

el mètode comprova que no hi hagi més d'un usuari amb el mateix codi i en cas de que així fos donaria error.

```
if (rs.next()) {
    // S'ha trobat més d'un usuari amb aquest codi
```

També comprova que com a mínim existeixi un usuari amb aquest dni.

getEliminaUsuari(String dni)

elimina l'usuari de la taula amb dni corresponent al'string rebut a dni.

```
sql="DELETE FROM usuaris WHERE nidentitat LIKE '%" + dni + "%";
```

en aquest cas i al no tractar-se d'un select, l'ordre es

```
rs=pstmt.executeUpdate(sql);
```

afegirUsuari(Usuari usr)

afegeix a la taula corresponent l'usuari rebut com a usr. S'extrauen tots els atributs de usr per a inserir-los en la sentència sql utilitzant un preparedStatement

```
sql="INSERT INTO etaller.Usuaris (nidentitat,CONTRASSENYA," +
"TIPUSUSUARI,DATAALTA,BLOQUEJAT,NOM,COGNOMS,CODIGRUP)"
+ " VALUES (?,?,?,?,?,?,?,?);";
```

```
PreparedStatement pstmt=con.prepareStatement(sql);
pstmt.setString(1,usr.getNidentitat());
pstmt.setString(2,usr.getContrassenya());
pstmt.setString(3,usr.getTipusUsuari());
pstmt.setDate(4,data.valueOf(usr.getDataAlta()));
pstmt.setString(5,usr.getBloquejat());
pstmt.setString(6,usr.getNom());
pstmt.setString(7,usr.getCognoms());
pstmt.setString(8,usr.getCodiGrup());
System.out.println(pstmt.toString());
```

```
// Executem la sentència
rs = pstmt.executeUpdate();
```

modificarUsuari(Usuari usr)

modifica les dades d'un usuari concret a la taula. S'extrau el dni de l'usuari rebut a usr, per a fer-ne referència com a dni de l'usuari a modificar i les altres dades per a preparar la sentència sql. El procediment es molt similar a afegirUsuari,

```
sql="UPDATE usuaris SET "+ "CONTRASSENYA=?, TIPUSUSUARI=?, "
+ "DATAALTA=?, BLOQUEJAT=?, NOM=?, COGNOMS=?, CODIGRUP=? "
+ "WHERE nidentitat='"+usr.getNidentitat()+"''";
```

Cal mencionar que a l'hora de llistar dades una mica més elaborades la sentència augmenta de volum i s'utilitza una consulta a vèries taules per a que el resultat sigui una mica més entenedor.

Com a exemple utilitzarem la sentència utilitzada per a llistar les tasques assignades a un grup determinat que encara no han estat finalitzades.

```
sql="SELECT assigtasques.coditasca, assigtasques.datainici,assigtasques.datafi,
tasques.titol, tasques.descripcio, tasques.durada FROM tasques INNER JOIN
assigtasques ON tasques.coditasca=assigtasques.coditasca WHERE
assigtasques.codigrup LIKE '%" + codigrup + "%' AND assigtasques.datafi IS NULL";
```

4.4.4. Els servlets

S'han creat dos servlets per a fer més intel·ligible la seva lectura, separant-los per la naturalesa dels seus mètodes així doncs podem trobar:

LlistarServlet que hereta de **HttpServlet**

El servlet es crida des de les pàgines de l'aplicació quan s'ha de gestionar les dades de la base de dades per a ser representades en les pàgines dinàmiques, així doncs ha de saber distingir entre els diferents tipus de crides per a poder realitzar unes funcions determinades. Això ho resolem d'una forma un poc ortodoxa però eficient,

```
if(request.getParameter("accioUsuaris")!=null){
    desti="LlistatUsuaris.jsp";
    prepararLlistatUsuaris(request, response);
}else if(request.getParameter("accioGrups")!=null){
    desti="LlistatGrups.jsp";
    prepararLlistatGrups(request, response);
...

```

Es fa un request sobre els paràmetres que s'han declarat a les pàgines i aïllant el que es diferent de null, podem saber que es el que s'està demanant, executar el mètode corresponent i enviar a l'usuari al seu destí.

Els mètodes implementats, poden ser de preparació de llistes on s'ha d'escollir un element i on ha de constar la informació d'una taula corresponent com,

```
protected void prepararLlistatGrups(HttpServletRequest request, HttpServletResponse response){
    GrupDAO grpDAO;
    Vector<Grup> grups;
    grpDAO=new GrupDAO();
    grups=grpDAO.getGrups();
    System.out.println("Grups Agafats");

    request.setAttribute("llistaGrups",grups);
    System.out.println("Grups posats a llistaGrups");
}
```

On es crida la funció `getGrups` de `GrupDAO`, que proporciona un vector amb els grups existents i es col·loca com a paràmetre `llistaGrups`, que serà agafat per la pàgina dinàmica a l'hora de fer un llistat.

Tots els mètodes del servlet segueixen una estructura similar ja que la funció principal que se l'hi ha assignat es la de llistar informació, però cal destacar dos mètodes de màxima importància tb inclosos en aquest servlet.

```
protected String comprovaContrassenya(HttpServletRequest request, HttpServletResponse response){
    UsuariDAO usrDAO;
    Usuari usuari;
    String desti;
    usrDAO=new UsuariDAO();
    //System.out.println(request.getParameter("nidentitat"));
    usuari=usrDAO.getCercaUsuari(request.getParameter("n_identitat"));
    if(usuari==null)
        desti="errorValidacio.html";
    else if(usuari.getContrassenya().equals(request.getParameter("contrassenya"))){
        desti="Portal1.jsp";
        HttpSession sesionOk = request.getSession();
        sesionOk.setAttribute("tipusUsuari",usuari.getTipusUsuari());
        sesionOk.setAttribute("grupUsuari",usuari.getCodiGrup());
        sesionOk.setAttribute("nidentitat",usuari.getNidentitat());
    }
    else
        desti="errorValidacio.html";
    return(desti);
}
```

Comprova la identificació cercant un usuari amb el DNI rebut del formulari d'identificació i comparant-ne la contrasenya. En el cas que tot sigui correcte crea una sessió amb la informació necessària que es mantindrà activa fins que l'usuari faci el tancament.

I per altra banda trobem el mètode,

```
protected void tancaSessio(HttpServletRequest request, HttpServletResponse response){  
  
    HttpSession sessionOk=request.getSession();  
    if(sessionOk !=null)  
        sessionOk.invalidate();  
  
}
```

De tancament de la sessió.

Continuem amb l'altre servlet,

AltaServlet que hereta de **HttpServlet**

Aquest servlet respon a totes les peticions referents a manipulació de dades, ja sigui afegir, esborrar o modificar registres. Podem trobar mètodes del tipus,

```
protected void altaGrup(HttpServletRequest request, HttpServletResponse response){  
    Grup grp;  
    GrupDAO grpDAO;  
    grp=new Grup();  
    grp.setCodiGrup(request.getParameter("codiGrup"));  
    grp.setNomGrup(request.getParameter("nomGrup"));  
  
    grpDAO=new GrupDAO();  
    grpDAO.afegirGrup(grp);  
  
}
```

que crea un objecte grup amb les dades que agafa (mitjançant un request) del formulari corresponent, i l'envia com a paràmetre a l'hora de cridar el mètode afegir grup de la classe grupDAO.

Tots els mètodes de creació i modificació d'elements de la base de dades segueixen unes pautes molt similars a l'anterior mètode.

Pel que fa als mètodes d'esborrat podem trobar,

```
protected void baixaGrup(HttpServletRequest request, HttpServletResponse response){  
    GrupDAO grpDAO;  
    boolean rs;  
  
    grpDAO=new GrupDAO();  
    System.out.println(request.getParameter("baixaGrup"));  
    grpDAO.getEliminaGrup(request.getParameter("baixaGrup"));  
    System.out.println("Grup Esborrat");  
  
}
```

que crida el mètode getEliminaGrup de la classe grupDAO passant com a paràmetre baixaGrup que correspon al codi de grup que l'usuari introdueix com a grup a eliminar. Tots els altres mètode segueixen una política similar.

4.4.5. Proves

El sistema de proves s'ha basat des del principi en l'elaboració d'un prototip que s'ha anat modificant al llarg del període d'implementació.

El fet de testejar l'aplicació en l'entorn real per al qual ha estat dissenyada i pels usuaris finals als quals s'ha utilitzat durant la fase d'anàlisi fa que les proves realitzades acotin bastant el seu funcionament a una metodologia ja establerta. Tot i que el tractament d'errors no ha estat el més adient per la falta de temps en aquest apartat de la implementació, cal destacar que el coneixement de tots els procediments bàsics per part dels usuaris sobre els que s'ha testejat l'aplicació fa que el comportament inicial de la mateixa sigui sòlid i estable.

De totes formes es suposa una corba d'aprenentatge una mica més pronunciada per a usuaris no iniciats en el sector.

4.5. Conclusions.

En el moment d'escollir un tipus d'aplicació determinada per al treball de fi de carrera, vaig optar per J2EE per diferents motius i per la gestió d'escoles taller per un molt particular.

J2EE es un entorn que no coneixia abans de començar la carrera i que m'ha absorbit molt des del principi, el que necessitava ara per a poder-la fer operativa totalment en el meu entorn de treball era simplement aplicar-la a les tecnologies Web, ja que el futur ens porta cap a aquesta tendència. Per tant volia aprendre més, volia conèixer el món del JSP, tot i no haver-lo tocat mai.

Perquè la gestió d'una Escola Taller? Per que treballa en una i conec molt bé les necessitats i mancances que es tenen en un entorn com aquest.

Anàlisi

Ha estat l'etapa més treballada, però a l'hora la menys costosa, ja que la ubicació del projecte correspon a una ubicació laboral real. Això m'ha proporcionat l'oportunitat d'anar analitzant els requeriments de l'aplicació i l'entorn real en el que es pot aplicar.

Disseny

En el meu cas la fase més costosa, tot i que al meu cap les necessitats i l'anàlisi de l'aplicació son ben clars, és difícil plasmar en un paper de forma intel·ligible tot el funcionament de l'aplicació a mode de diagrames, de totes maneres he intentat desenvolupar tots els coneixements adquirits en assignatures com Estructura de la Informació, Enginyeries del Programari, TDP... He utilitzat el programari JUDE per a l'elaboració dels diagrames.

Implementació

La part més emocionant i alhora més complexa.

Partint d'un nivell 0 en JSP però una base elemental de Java el repte era complicat, i no pel projecte en si, sinó per que la implementació s'havia de fer en un mes.

La corba d'aprenentatge ha estat més pronunciada del que esperava i les meves ambicions inicials han posat les coses encara més complicades.

He utilitzat des del principi Dreamweaver per al disseny de les pàgines i després he incorporat les dades dinàmiques amb el Netbeans. Tenia

nocions dels dos programes per vivències passades i la veritat es que s'han mostrat prou competents i estables durant tot el procés.

Durant aquesta fase he intentat mostrar tot el que havia après en assignatures com Fonaments de Programació I i II, Bases de Dades o inclús en altres com Interacció Humana amb els ordinadors.

El resultat crec que ha estat molt satisfactori, assentant les bases d'un projecte per a després anar-lo millorant al llarg del temps, amb nous coneixements, noves implementacions o inclús noves tecnologies.

Es pot dir que he complert diferents punts del programa entrant en el món del JSP, coneixent-lo de forma extensa encara que poc profunda i treballant amb diferents camps com:

- Connexió amb Base de Dades.
- Capes d'accés a dades DAO.
- Servlets.
- Disseny d'interfícies i patrons.
- Sessions.
- Ús de Tomcat i MySQL.
- Tecnologia JSP.

...

Per tant es pot dir que l'aplicació final es mostra prou estable i acotada amb les aspiracions inicials, pel que fa al tractament de dades, mostra de les mateixes disseny de les interfícies...

La espina que s'ha quedat (sempre ha de quedar alguna) ha estat per a mi, el tractament de les excepcions. Si bé un usuari que coneix el món de les escoles taller no tindria massa dificultat potser un usuari menys involucrat pot arribar a sentir-se perdut amb alguns errors en que la informació resultant es poc concreta.

De totes formes es pot dir que amb una mica de pràctica i coneixent l'entorn de treball l'aplicació funciona correctament i es pot treballar de forma estable i fiable.

4.6. Manuals.

4.6.1. Manual d'instal·lació.

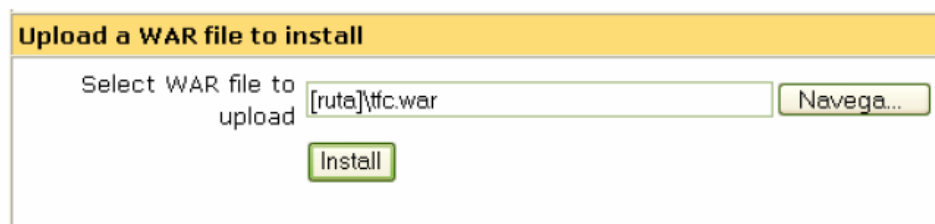
Ubicant-nos al punt de partida per a la instal·lació de l'aplicació, tenint en compta la naturalesa Web de l'aplicació i la seva interacció amb bases de dades per a poder mostrar les dades, necessitem contar amb dues aplicacions bàsiques per al seu correcte funcionament:

El servidor Apache Tomcat, que és un servidor web que pot contenir servlets i es pot trobar a l'adreça <http://jakarta.apache.org/tomcat/> de forma lliure.

El sistema gestor de bases de dades MySQL, que es pot trobar a l'adreça <http://www.mysql.com/> també de forma lliure.

Un cop contem amb les dues aplicacions en correcte funcionament, comencem la instal·lació de la nostra aplicació, els passos son els següents:

- Creació de la base de dades: executem l'script eTaller.sql (també executarem eTaller_Inici.sql per afegir a la Base de dades uns Usuaris inicials i grups per a poder començar a treballar). Podem executar l'script per parts des de MySQL Browser.
- Config.properties, modificar l'arxiu que hi ha a la carpeta on estan ubicades les classes al servidor (eTaller\WEB-INF\classes\com\boikot\etaller), que per defecte son:
 - Usuari: root
 - Password: 123456
- Desplegar l'aplicació Web en el servidor:
 - L'aplicació es proporciona en un paquet eTaller.war, per a desplegar-la al servidor només cal escollir l'opció a Tomcat i cercar-la.



L'aplicació està llesta per a funcionar, tot i que s'hagués volgut implementar un mètode de configuració de la BD propi, de moment i per falta de temps s'ha deixat la possibilitat oberta.

4.6.2. Manual d'Usuari.

Inici del programa

En connectar-se a la pàgina principal de l'aplicació index.jsp, el resultat serà un missatge de benvinguda i la petició d'identificació per a poder iniciar una sessió de treball,



The screenshot shows a web page with a yellow background. At the top, it says "Escola Taller Mercavònia" with a small logo. Below that, the title is "Benvingut :: Identificació". There are two input fields: "DNI:" and "Contrassenya:". Below the fields is a button labeled "Acceptar".

En aquesta finestra s'haurà d'indicar el DNI d'usuari i la contrassenya actual. Es precís que la contrassenya indicada respecti les lletres minúscules i majúscules. En prémer el botó Acceptar el programa validarà el DNI d'usuari i contrassenya proporcionats i, si tot ha estat correcte, mostrarà una finestra similar a la següent:




The screenshot shows a web page with a yellow background. The title is "Benvingut :: Menú d'Opcions". Below the title, it says "Opcions de Professor". There are two columns of options. The left column has "Dades Personals" (with sub-options: "Les meves dades", "Components del Grup") and "Historial Tasques" (with sub-options: "Tasques Finalitzades", "Tasques Assignades"). The right column has "Assignacions" (with sub-options: "Assignar Tasca", "Finalitzar Tasca", "Assignar Sancio") and "Historial Sancions" (with sub-option: "Sancions Assignades"). At the bottom, there is a "logout" button and a link "tornar a l'inici". At the very bottom, it says "© Gonçal Verge gverge@uoc.edu ETIG a la UOC".

En aquesta finestra es poden veure les opcions principals del programa ja habilitades i preparades per treballar i la possibilitat de tancar la sessió.

Les opcions seran diferents per a cada perfil d'usuari, que podrem diferenciar per un missatge inferior a la benvinguda que ens informarà del tipus de menú que se'n mostra.

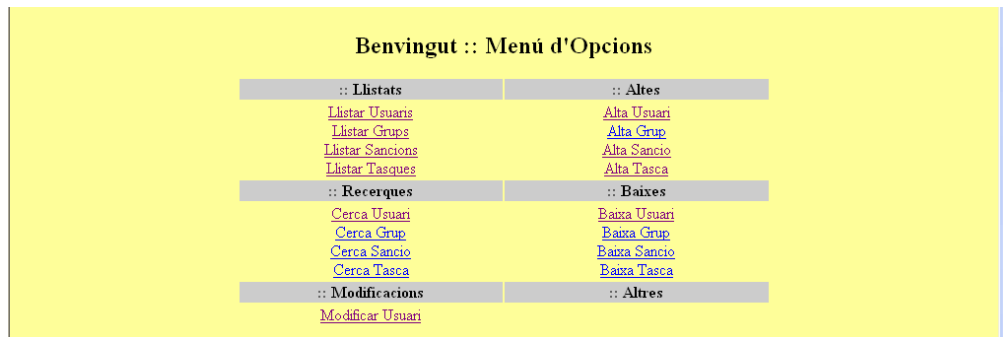
En cas de que la validació no hagi estat satisfactòria ens mostrarà un missatge d'error.



The screenshot shows a yellow error message box. The title is "Error de validació!". The text inside says "La teva contrassenya o nom d'usuari no son vàlids."

Manteniment de l'aplicació

Els administradors tenen la funció de realitzar tot el manteniment de l'aplicació i son els únics usuaris que tenen aquesta possibilitat. El seu menú es el següent:



Alta d'usuaris

Tot usuari que hagi d'utilitzar el programa haurà d'haver estat donat d'alta. Per fer l'alta d'un usuari cal seleccionar l'opció Alta Usuari.

Formularis :: Alta Usuari

DNI:

Assignar Contrassenya:

Nom:

Cognoms:

Tipus Usuari: (A P o T)

Data d'Alta:

Grup:

En seleccionar aquesta opció es mostrarà la finestra on s'hauran d'indicar totes les dades sol·licitades.

- DNI: És el codi exclusiu que identifica l'usuari al sistema. Aquesta dada és obligatòria.
- Contrassenya: És el codi imprescindible per a poder identificar-se a l'hora d'iniciar la sessió. Dada Obligatòria.
- Data d'alta: Data en que s'ha introduït l'usuari. Ha de seguir el format AAAA-MM-DD. No obligatòria.
- Nom: No pot superar els 15 caràcters. Aquesta dada és obligatòria.
- Cognoms: Espai on s'hauran d'indicar el o els cognoms. No pot superar els 25 caràcters. Aquesta dada és obligatòria.
- Tipus Usuari: Caràcter únic que indica el perfil d'usuari. Dada obligatòria. Aquest fet és el que permetrà assignar els privilegis d'accés
- Grup: Camp on es pot escollir el grup al qual s'assigna l'usuari. Dada Obligatòria.

Quan ja s'hagin indicat totes les dades s'haurà de prémer el botó Acceptar per indicar-li al programa que es volen guardar. Si tot a anat bé l'usuari quedarà enregistrat a la base dades i, per tant, ja estarà en disposició d'utilitzar el programa. Es mostra un missatge comprovant.

Cerca d'usuaris

Excepte en el cas d'alta, les altres opcions del manteniment d'usuaris requereix un cerca d'usuari prèvia. És a dir, cal indicar-li al programa quin usuari es vol consultar, modificar o esborrar. En seleccionar qualsevol d'aquestes opcions es mostrarà la finestra de cerca d'usuaris.

::Cerca d'Usuari::

DNI de l'Usuari:

Consulta d'usuari

S'ha d'introduir directament el codi de l'usuari en el camp corresponent i prémer el botó Acceptar.

Si les dades son correctes es mostrarà la informació corresponent a l'usuari cercat, amb les opcions d'eliminar i modificar.

::Fitxa d'Usuari::

DNI:	00000001G
Contrassenya:	PASS1
Nom:	Profel
Cognoms:	Modul1
Tipus Usuari:	P
Data d'Alta:	2005-03-03
Grup:	I
eliminar / modificar	

Modificació d'usuaris

El primer pas consisteix en fer la selecció de l'usuari que es vol modificar seguint el procediment descrit en l'apartat "Cerca d'usuaris". Després només cal prémer l'opció de modificar.

No es podrà modificar el DNI d'usuari ja que aquest es la clau principal.

El procediment i la pantalla resultat es el mateix que en una alta d'usuari.

Baixa d'usuari

Aquesta opció només la tenen disponibles els usuaris Administratius i, per utilitzar-la, l'usuari que es vol donar de baixa haurà d'existir a la base de dades.

El primer pas consisteix en fer la selecció de l'usuari seguint el procediment descrit en l'apartat "Cerca d'usuaris".

Un cop es mostren les dades s'haurà de prémer el botó d'eliminar. Si l'operació s'ha realitzat correctament apareixerà un missatge comprovant.

La petició s'ha processat correctament.

Llistar Usuaris

Aquesta opció, permet llistar tots els usuaris d'alta a la Base de Dades amb les seves dades.

Només cal prémer l'opció corresponent en el menú inicial per a obtenir un resultat similar al següent::

::Llistat d'Usuaris::		
::DNI Usuari	::Nom	::Cognoms
00000000G	Administrador	Inicial
00000001G	Profe1	Modul1
00000002G	Profe2	Modul2
00000003G	Alumne1	Modul1
00000004G	Alumne2	Modul2

Altres funcions de manteniment.

Totes les altres opcions de manteniment funcionen de forma molt similar i no es creu necessari extendre's més en les eves instruccions d'ús.

Opcions de Professor (Exclusivament)

Benvingut :: Menú d'Opcions	
Opcions de Professor	
:: Dades Personals	:: Assignacions
Les meves dades	Assignar Tasca
Components del Grup	Finalitzar Tasca
:: Historial Tasques	:: Historial Sancions
Tasques Finalitzades	Sancions Assignades
Tasques Assignades	

Assignació de Tasques

Ens permet assignar una tasca al grup del qual l'usuari que ha iniciat la sessió es professor, només ens cal seleccionar la tasca de la llista desplegable i introduir la data d'inici. Automàticament tots els membres del grup tindran assignada la tasca corresponent.

Formularis :: Assignació Tasques	
Tasca:	IM1 - Arquitectura d'Ordinadors ▾
Data Inici:	2005-12-01
<input type="button" value="Acceptar"/>	

Finalització de Tasques

Ens permet donar per finalitzada una tasca prèviament assignada al grup del qual l'usuari que ha iniciat la sessió es professor, només ens cal seleccionar la tasca de la llista desplegable i introduir la data de finalització. Automàticament a tots els membres del grup els constarà com a finalitzada la tasca corresponent.

Formularis :: Finalització Tasques

Tasca:

Data Fi:

Assignació de Sancions

Ens permet assignar una sanció a un alumne del qual l'usuari que ha iniciat la sessió es professor, només ens cal seleccionar l'alumne de la llista desplegable, el tipus de sanció corresponent, les hores assignades i els comentaris oportuns. Automàticament el treballador sancionat tindrà assignada la sanció corresponent.

Formularis :: Assignació Sancions

Sancio:

Alumne:

Hores:

Comentaris:

Sancions Assignades

En prémer l'opció corresponent al menú es llistaran les sancions assignades a mode d'historial, de forma automàtica sense necessitat d'introduir cap dada més. Constarà el codi i com a descripció, el tipus de sanció (alumne sancionat - hores de sanció) i - comentaris.

::Llistat de Sancions

::CodiSancio	::Descripcio
S01	Falta Assistència Lleu (Alumne1 Modul1 - 6 hores) - 20/12/2005
S02	Falta Assistència Greu (Alumne1 Modul1 - 8 hores) - 21/12/2005 - Reincidència

Opcions de Professor i Treballador

Les meves dades

En prémer l'opció corresponent al menú apareixeran les dades de l'usuari de forma automàtica, sense necessitat d'introduir cap dada més, el professor podrà modificar-les, a diferència del treballador que només podrà visualitzar-les.

::Fitxa d'Usuari::

DNI:	00000001G
Contrassenya:	PASS1
Nom:	Profe1
Cognoms:	Modul1
Tipus Usuari:	P
Data d'Alta:	2005-03-03
Grup:	I
eliminar / modificar	

Components del grup

En prémer l'opció corresponent al menú es llistaran els membres del grup al qual pertany l'usuari, de forma automàtica sense necessitat d'introduir cap dada més.

::Llistat d'Usuaris::

::DNI Usuari	::Nom	::Cognoms
00000001G	Profe1	Modul1
00000003G	Alumne1	Modul1

Tasques Finalitzades

En prémer l'opció corresponent al menú es llistaran les tasques finalitzades a mode d'historial, de forma automàtica sense necessitat d'introduir cap dada més. Constarà el codi, el títol i la informació d'inici i fi de la tasca així com el comentari i la durada aconsellada.

::Llistat de Tasques::

::CodiTasca	::Titol	::Descripcio	::Durada
IM1	Arquitectura d'Ordinadors	(2005-12-01 fins 2006-01-03)	100

Tasques Assignades

En prémer l'opció corresponent al menú es llistaran les tasques assignades a mode d'historial, de forma automàtica sense necessitat d'introduir cap dada més. Constarà el codi, el títol, la informació d'inici de la tasca així com el comentari i la durada aconsellada.

::Llistat de Tasques::

::CodiTasca	::Titol	::Descripcio	::Durada
IM1	Arquitectura d'Ordinadors	(2005-12-25)	100
IM2	Xarxes	(2005-12-01)	50

Opcions de Treballador(Exclusivament)

Benvingut :: Menú d'Opcions

Opcions de Treballador

:: Dades Personals	::
Les meves dades	
Components del Grup	
:: Tasques	:: Sancions
Tasques Finalitzades	Les meves sancions
Tasques Assignades	

Les meves Sancions

En prémer l'opció corresponent al menú es llistaran les sancions assignades a titol personal a mode d'historial, de forma automàtica sense necessitat d'introduir cap dada més.

Constarà el codi i com a descripció, el tipus de sanció (hores de sanció) i - comentaris.

::Llistat de Sancions

::CodiSancio	::Descripcio
S01	Falta Assistencia Lleu (6 hores) - 20/12/2005
S02	Falta Assistencia Greu (8 hores) - 21/12/2005 - Reincidència

5. Bibliografia.

Escrita

- **JSP**, Anaya Multimedia, Guía práctica
- **JSP Ejemplos prácticos**, Anaya Multimedia Andrew Patzer

Electrònica

- **Tecnologías de servidor con Java: Servlets, JavaBeans i Jsp**, Ángel Esteban
- **JAVA SERVLET PROGRAMMING**, O'REILLY JSP
- **Lawebdelprogramador.com**

6 Glossari

- **Administrador:** Persona que treballa a l'empresa eTaller i s'encarrega de la gestió i manteniment. És l'encarregat de fer les altes i baixes d'usuaris, tasques, grups...
- **assigTasca:** Assignació d'una tasca a un grup determinat, conte una data d'inici, de fi, comentaris i els codis de grup i tasca de l'assignació.
- **assigSancio:** Assignació d'una sancio a un treballador determinat, conte els codi d'usuari i sanció de l'assignació, les hores i un petit comentari.
- **codiTasca:** Identificador de tasca o codi assignat a una tasca.
- **codiSancio:** Identificador de sanció o codi assignat a una sanció.
- **codiGrup:** Identificador de grup o codi assignat a una grup.
- **Contrasenya :** Paraula de pas que necessita qualsevol usuari per identificar-se a l'aplicació. En un primer moment serà l'administrador de l'empresa qui assigni aquesta paraula de pas en el moment de fer l'alta, posteriorment l'usuari demanar el canvi.
- **DNI:** Número que identifica a l'usuari dins eTaller, serà únic.
- **Connexió:** Vincle entre el Servidor Web i el Client Web. S'estableix cada vegada que un usuari accedeix al sistema.
- **eTaller:** Escola Taller capaç de fer feines en concepte de pràctiques, del tipus corresponent a l'especialització dels grups.
- **Professor:** Al igual que l'administrador treballa a l'empresa eTaller i realitza feines de gestió estrictament sobre usuaris.
- **Tasca:** Feina encarregada a l'escola que serà realitzada per un grup.
- **Treballador:** Persona vinculada a l'empresa que realitza les tasques encarregades.
- **Usuari:** Tota persona que utilitza el programa.