

# **Gestió acadèmica en una escola**

**Oscar Jaén Collado**  
ETIG

**Albert Grau Perisé**

20/06/2011

*A la meva família i a la Noe, que han tingut molta paciència amb mi al llarg de tant de temps.*

## Resum Treball Fi de Carrera

El projecte desenvolupat ha estat una aplicació web per gestionar una escola. Mitjançant el portal, es permetrà gestionar les matriculacions, l'expedient acadèmic i notificacions i avisos que es produeixin a l'escola.

Existiran diferents perfils d'ús dintre de l'aplicatiu. El *personal administratiu*, gràcies al desenvolupament d'aquest programari, li permetrà un accés fàcil a tota la informació i agilitzarà el manteniment de les dades. Serà el responsable de mantenir la informació necessària pel correcte funcionament en el procés de la matriculació: manteniment d'anys acadèmics, tipus de qualificacions, tipus de titulacions, cursos, aules, assignatures. A més a més, podrà assignar els *professors* a les assignatures que s'imparteixin en les aules. Per últim, també podrà ubicar a cada *alumne* a les aules de cada curs.

El *professor* podrà publicar les notes dels *alumnes* de cada una de les assignatures que ell imparteixi en el centre escolar de cada període lectiu.

Els *alumnes* podran veure el seu expedient acadèmic així com la relació de qualificacions de cada assignatura cursada. Per cada any acadèmic podran realitzar la matriculació d'una manera totalment *on-line* i sense haver de presentar-se a l'escola i haver d'omplir formularis.

Aquesta aplicació web també ha d'estar orientada als *parens/tutors legals* dels alumnes. Ells podran accedir-hi a les dades personals dels seus fills, així com poder realitzar la matrícula (en el cas que els seus fills encara siguin massa petits).

Un servei afegit en aquest portal serà la possibilitat de notificar avisos a diferents nivells (escola, curs, aula, assignatura). Aquestes notificacions podran ser publicades pel *personal administratiu* o bé pels *professors*. D'aquesta manera tots els usuaris del portal estaran al corrent de notícies, activitats extraescolars, excursions, festes....

Els diferents tipus d'usuari podran utilitzar l'aplicació mitjançant un navegador i per un accés des d'Internet. Quan un usuari entri al sistema, tindrà la possibilitat de canviar la informació de les seves dades bàsiques, així com poder modificar la seva clau d'accés al sistema.

Per desenvolupar aquest projecte s'ha fet servir una de les plataformes més conegudes i reeixides que treballen amb els sistemes de la informació, J2EE. Mitjançant aquest treball final de carrera he conegut i profunditzat en aquesta plataforma oberta i estàndard, així com els patrons de disseny, centrant-me en el MVC, *Model Vista Controlador*, ja que separa les **dades** de l'aplicació, la **interfície** d'usuari i la **lògica de negoci** en tres components diferenciats.

Tecnològicament parlant, s'ha fet servir una base de dades *MySQL* sota un servidor d'aplicacions *JBoss*. Per la persistència de dades he utilitzat *Hibernate*, per l'encapsulament de tots els accessos a la font de les dades el patró *Data Transfer Object (DAO)* i per la capa de presentació el patró de disseny *Struts*. Finalment, per desenvolupar tot el codi he fet servir l'entorn de desenvolupament integrat *Eclipse*.

### Paraules clau

Alumnes, aplicació distribuïda, aplicació web, capa de presentació, Data Transfer Object (DAO), Eclipse, framework, Hibernate, HTML, Internet, iReport, J2EE, Java, JasperReport, JBoss, JSP, Lògica de Negoci, MySQL, MVC (Model Vista Controlador), notificació, ORM, pare/tutor legal, patró de disseny, personal administratiu, portal, professor, servidor d'aplicacions, servlets, Struts, UML.

### Àrea TFC

J2EE

## Índex

Resum Treball Fi de Carrera .....	3
1 Introducció .....	7
1.1 Justificació del TFC i context.....	7
1.2 Objectius .....	7
1.3 Enfocament i mètode seguit .....	8
1.4 Planificació del projecte .....	8
1.5 Productes obtinguts .....	9
1.6 Breu descripció dels altres capítols de la memòria .....	9
2 Anàlisi i Disseny .....	9
2.1 Punt de vista de l'empresa .....	9
2.2 Punt de vista de la informació .....	33
2.3 Punt de vista computacional.....	37
2.4 Punt de vista de l'enginyeria.....	41
2.5 Punt de vista de la tecnologia.....	42
2.6 Implementació del sistema.....	42
3 Disseny de la interfície gràfica d'usuari .....	42
3.1 Autenticació de l'usuari.....	43
3.2 Subsistema de Manteniment.....	44
3.3 Subsistema d'usuari .....	48
3.4 Subsistema de notificacions.....	51
3.5 Subsistema de matriculació.....	53
3.6 Subsistema de gestió de l'expedient .....	54
3.7 Llegenda d'ícones .....	55
4 Valoració econòmica .....	56
5 Conclusions i línies de futur .....	56
5.1 Conclusions .....	56
5.2 Línies de futur.....	57
6 Glossari .....	57
7 Bibliografia.....	59
7.1 Struts .....	59
7.2 Hibernate .....	59
7.3 Jboss .....	60
7.4 JasperReport .....	60
7.5 Altres .....	60
8 Annex: Creació del projecte des d'Eclipse.....	60
8.1 Variables d'entorn .....	60
8.2 Instal·lació del plugin de JBoss .....	60
8.3 Instal·lar JBoss com a Runtime i Servidor .....	61
8.4 Creació del projecte.....	61

8.5	<i>JasperReport</i> .....	63
8.6	<i>Arrencar servidor d'aplicacions JBoss</i> .....	63
8.7	<i>Instal·lació del connector JDBC de MySQL</i> .....	63
8.8	<i>Configuració de JBoss per usar MySQL</i> .....	64
8.9	<i>Crear taules a MySQL</i> .....	64
8.10	<i>Hibernate</i> .....	64
8.11	<i>Generar amb l'Ant</i> .....	64
8.12	<i>Execució des del navegador</i> .....	65
8.13	<i>Accés al sistema</i> .....	65
8.14	<i>Programari utilitzat durant el projecte</i> .....	66

### Índex de les il·lustracions

Il·lustració 1.	Diagrama de Gantt .....	8
Il·lustració 2	Cas d'ús autenticació del sistema.....	14
Il·lustració 3.	Cas d'ús Subsistema de Manteniment .....	22
Il·lustració 4.	Cas d'ús Manteniment d'aules .....	23
Il·lustració 5.	Cas d'ús manteniment usuaris.....	25
Il·lustració 6.	Cas d'ús manteniment de notificacions .....	27
Il·lustració 7.	Cas d'ús manteniment de matriculació .....	29
Il·lustració 8.	Cas d'ús gestió de l'expedient .....	31
Il·lustració 9.	Diagrama d'activitats d'una matriculació .....	32
Il·lustració 10.	Diagrama d'activitats d'assignació automàtica d'alumnes a aules.....	33
Il·lustració 11.	Diagrama de classes .....	34
Il·lustració 12.	Diagrama Entitat Relació.....	35
Il·lustració 13.	Model lògic relacional .....	36
Il·lustració 14.	Punt de vista computacional .....	37
Il·lustració 15.	Arquitectura programa detallada .....	38
Il·lustració 16.	Principal classes del controlador Struts .....	39
Il·lustració 17.	Principal classes del patró DAO.....	40
Il·lustració 18.	Diagrama seqüència autenticació usuari .....	41
Il·lustració 19.	Punt de vista d'enginyeria .....	42
Il·lustració 20.	Diagrama de desplegament de l'aplicació .....	42
Il·lustració 21.	Entrada al sistema .....	43
Il·lustració 22.	Canvi de paraula clau .....	43
Il·lustració 23.	Cerca any acadèmic .....	44
Il·lustració 24.	Manteniment any acadèmic .....	44
Il·lustració 25.	Cerca assignatura .....	44
Il·lustració 26.	Manteniment assignatura.....	45
Il·lustració 27.	Cerca tipus de qualificació .....	45
Il·lustració 28.	Manteniment tipus de qualificació .....	45

II-lustració 29. Cerca tipus de titulació .....	46
II-lustració 30. Manteniment tipus de titulació .....	46
II-lustració 31. Cerca de curs .....	46
II-lustració 32. Manteniment de curs .....	47
II-lustració 33. Cerca d'aula.....	47
II-lustració 34. Manteniment d'aula .....	48
II-lustració 35. Cerca usuari .....	48
II-lustració 36. Manteniment personal administratiu .....	49
II-lustració 37. Manteniment professor .....	49
II-lustració 38. Manteniment alumne .....	50
II-lustració 39. Manteniment pare/tutor legal.....	50
II-lustració 40. Actualització foto .....	51
II-lustració 41. Menú principal.....	51
II-lustració 42. Cerca de notificació.....	52
II-lustració 43. Manteniment de notificació.....	52
II-lustració 44. Cerca de matriculació.....	53
II-lustració 45. Manteniment de matriculació .....	53
II-lustració 46. Cerca expedient acadèmic .....	54
II-lustració 47. Manteniment expedient acadèmic.....	54
II-lustració 48. Llistat expedient acadèmic .....	55
II-lustració 49. Creació instància runtime .....	61
II-lustració 50. Creació projecte Struts .....	62
II-lustració 51. Selecció versió servlet .....	62
II-lustració 52. Visió genèrica de com queda el projecte .....	63
II-lustració 53. Arrancar el servidor JBoss .....	63
II-lustració 54. Creació de taules a MySQL.....	64
II-lustració 55. Generar amb l'Ant .....	65
II-lustració 56. Executar aplicació.....	65

# 1 Introducció

Amb el treball fi de carrera s'ha posat en pràctica tots els coneixements adquirits durant la carrera. Vaig escollir l'àrea de J2EE per poder desenvolupar amb la tecnologia Java, ja que ens aquests últims anys s'ha convertit en un estàndard per al desenvolupament d'aplicacions en sistemes distribuïts.

Mitjançant aquesta projecte s'ha pogut investigar en diferents tecnologies concretes que són utilitzades a les empreses dintre de la plataforma J2EE, i així aprofundir en coneixements en algunes i aprendre d'altres per la meua inexperiència.

## 1.1 Justificació del TFC i context

Gràcies a aquest projecte final de carrera he conegut i aprofundit en J2EE, que és una plataforma oberta i estàndard i que permet desplegar aplicacions empresarials multicapa amb n-nivells, distribuïdes i basades en components. Porto ja un temps treballant amb Java, però amb la plataforma J2SE en un entorn gràfic. Sempre he volgut realitzar una aplicació web ('amb cara i ulls'), i amb aquest projecte he tingut l'oportunitat de fer-ho.

Treballar amb J2EE comporta una sèrie d'avantatges que he pogut veure i comprovar durant el desenvolupament del projecte, com és la facilitat de desenvolupar i oferir serveis als usuaris del sistema de manera àgil i ràpida.

Un altre fi per fer el treball final de carrera ha estat investigar, estudiar, i aprofundir en tecnologies que utilitzen la tecnologia Java. Durant la realització del projecte hi ha hagut un alt percentatge d'aprenentatge previ abans d'aconseguir resultats raonables, centrant-me en el patró de disseny Struts i l'eina de mapeig Hibernate. Això ha permès aplicar el model vista controlador (MVC) en l'arquitectura de programari.

Per realitzar el projecte he posat en pràctica tots els coneixements adquirits al llarg de la carrera, i especialment m'he centrant en assignatures com la *Programació Orientada a Objectes*, *Enginyeria del Programari*, *Enginyeria del Programari de Components* i *de Sistemes Distribuïts*, *Tècniques de Desenvolupament de Programari* i *Gestió d'Organitzacions i Projectes Informàtics*. Haver realitzat aquestes assignatures abans ha ajudat molt pel bon desenvolupament del projecte.

## 1.2 Objectius

L'objectiu d'aquest treball fi de carrera ha estat aprendre a desenvolupar un programari orientat a objectes, utilitzant la plataforma J2EE, a partir dels coneixements obtinguts al llarg de tota la carrera. He posat en marxa el cicle de vida del projecte, dividint el projecte en diferents fases.

Gràcies a l'anàlisi previ o **presa de requeriments** es va definir els grans trets del sistema de programari.

Mitjançant l'**anàlisi** es va definir detalladament les necessitats d'informació que havia de resoldre el programari, sense tenir en compte els mitjans tècnics amb què s'havia de dur a terme el desenvolupament del programari.

Amb la tercera etapa, el **disseny**, es va determinar com el programari havia de fer la seva funció, determinant la seva arquitectura general, l'estructura de dades (base de dades), l'especificació de cada programa i les interfícies amb l'usuari (menús, dissenys de pantalla, llistats...). Mitjançant el disseny de diagrames UML es va poder dissenyar els paquets (subsistemes), les dependències que hi havia entre ells i els diagrames de classes.

La quarta etapa va ser la **programació**, la més laboriosa. S'ha de dir que fet un bon treball amb les etapes anteriors, facilita molt la tasca, ja que no provoca incertesa en el desenvolupament i evita d'aquesta manera els errors i el deute tècnic futur. De totes maneres s'ha de dir que pel desconeixement tècnic en les eines utilitzades en el projecte, aquesta part de projecte va ser la

més costosa, amb molta investigació i proves realitzades abans de poder obtenir un resultat òptim del projecte.

L'etapa de **prova** va consistir a verificar el correcte funcionament del programari des de diversos punts de vista, tenint en compte els diferents perfils o rols d'usuari de l'aplicatiu. Naturalment es van localitzar i corregir els errors que es van detectar.

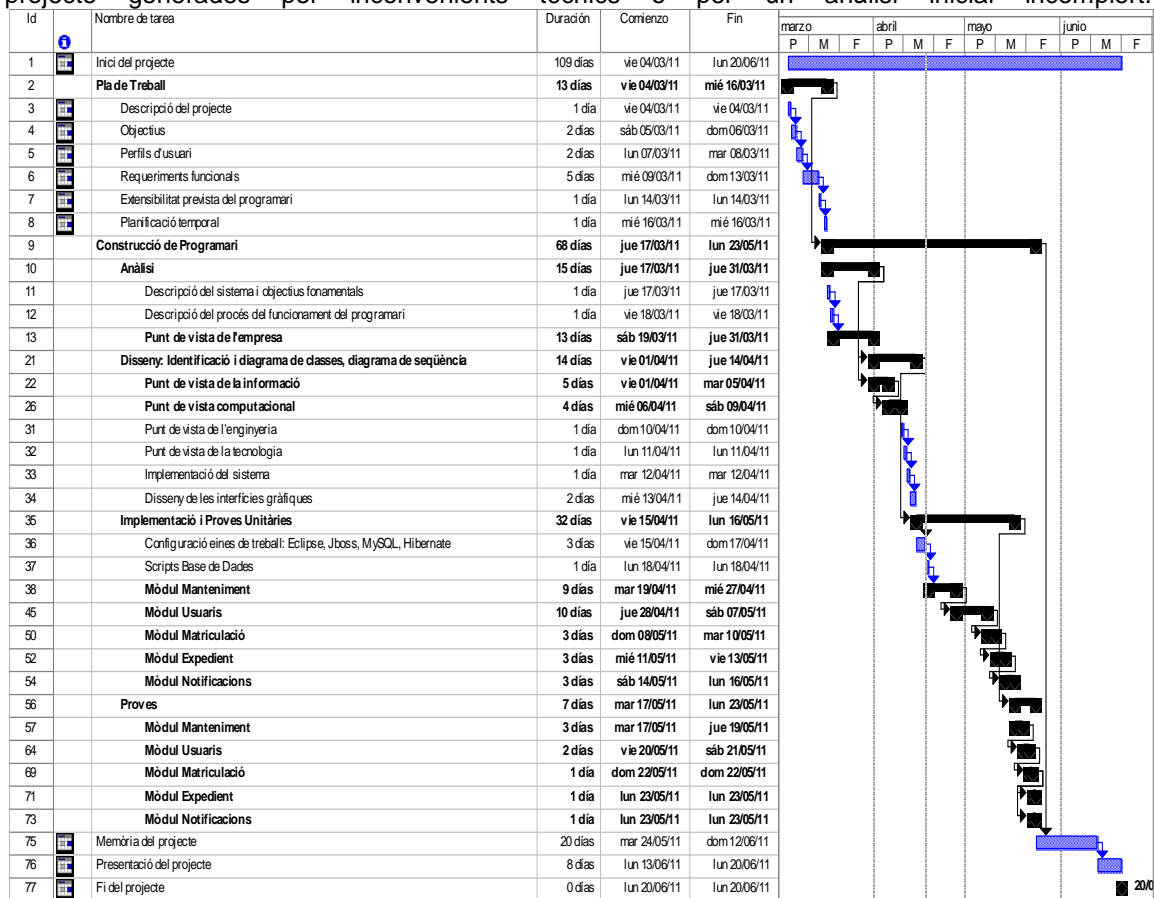
En resum, el desenvolupament del projecte ha permès una implementació complerta d'una aplicació distribuïda, centrant-nos en la comunicació entre les 3 capes que la componen (dades, lògica de negoci i interfície de l'usuari), i profunditzant en l'arquitectura J2EE mitjançant l'anàlisi, disseny i implementació d'una aplicació basada en aquesta arquitectura.

### 1.3 Enfocament i mètode seguit

Pel desenvolupament del projecte he cregut convenient utilitzar el cicle de vida clàssic –o en cascada– per a desenvolupar el nou sistema. La divisió en *PACs* al llarg de l'assignatura va ajudar molt a seguir aquesta metodologia, ja que es va separar el projecte en **pressa de requeriments, anàlisi i disseny, codificació, proves i implantació** (i si fos un cas real aquest projecte necessitaria una última etapa que seria la del manteniment).

### 1.4 Planificació del projecte

Per fer la planificació del projecte vaig fer servir el diagrama de Gantt. Mitjançant aquest diagrama vaig dividir el projecte en les diferents etapes del projecte, desglossant el treball en petites tasques. Això va aportar menys incertesa, provocant que no es produís desviacions temporals al llarg del projecte generades per inconvenients tècnics o per un anàlisi inicial incomplet.



II-lustració 1. Diagrama de Gantt



## 1.5 Productes obtinguts

Al finalitzar el treball final de carrera hem aconseguit aquests productes:

- **GestioAcademica.war**: És el fitxer desplegat del projecte *Gestió Acadèmica*.
- **GestioAcademica.ear**: Conté *GestioAcademica.war* i empaqueta els mòduls de negoci i integració.
- **Codi font del projecte**: Està separat en diferents paquets (*packages*) per diferenciar la *capa de presentació*, *lògica de negoci* i *dades de l'aplicatiu*.
- **Documentació**: Informació de les classes programades generades mitjançant la utilitat Javadoc a partir del codi font.
- **Arxiu de processament per lots** (*script*) per crear les taules de la base de dades MySQL.
- **Memòria**: És aquest document estructurat, on s'explica els aspectes més rellevants del projecte.
- **Presentació**: És el treball realitzat però sintetitzat al llarg del semestre.
- **Readme.txt**: Brevíssima explicació del projecte i explicació el contingut de les carpetes del codi font.
- **HowToInstall.txt**: Explica breument i molt esquemàticament els passos a seguir per fer la instal·lació del projecte en una altra màquina.

## 1.6 Breu descripció dels altres capítols de la memòria.

En el desenvolupament dels següents capítols aprofundirem sobre el desenvolupament de la *Gestió Acadèmica*. En el capítol 2 treballarem amb l'anàlisi i disseny. Utilitzarem el model de referència per al processament obert i distribuït, o RM-ODP, i definirem cinc punts de vista genèrics i bàsics a l'hora de definir un sistema: de l'**empresa**, la **informació**, la **computació**, l'**enginyeria** i la **tecnologia**. En el capítol 3, mostrarem la interfície de l'usuari així com la llegenda de cada icona utilitzada en el programari. En els següents capítols (del 4 al 7), es parla de la valoració econòmica del projecte, així com les conclusions obtingudes en la realització del treball final de carrera i les possibles línies de futur que es podrien aplicar en l'aplicatiu. També es llista tot el glossari de les paraules utilitzades al llarg del document i la bibliografia que s'ha utilitzat de suport. Finalment, en el capítol 8 s'inclou un annex on es parla de la configuració del projecte amb l'eina Eclipse.

## 2 Anàlisi i Disseny

Per descriure l'arquitectura global del sistema distribuït de la gestió acadèmica, ens basarem en la identificació i separació de punts de vista independents. Cada un d'aquests punts de vista se centra en una sèrie d'aspectes concrets, abstractant-se de la resta i simplificant, per tant, l'anàlisi i el disseny.

Es va decidir utilitzar aquest enfocament ja que actualment té un interès creixent per part de la indústria, on la complexitat de les aplicacions requereix dissenys manejables, estructurats i fàcils de mantenir.

### 2.1 Punt de vista de l'empresa

El punt de vista de l'empresa descriu els requisits del sistema i del seu entorn des de la perspectiva del mateix negoci.

#### 2.1.1 Objectius fonamentals i rols

Els objectius els podem enfocar respecte als diferents rols dels usuaris.

### 2.1.1.1 Personal administratiu

El personal administratiu estalviarà moltes de les gestions que antigament es feien en paper. L'objectiu principal del personal administratiu és mantenir el sistema amb la informació correcta perquè el circuit de la gestió de les matriculacions i expedients acadèmics no tinguin cap bloqueig.

### 2.1.1.2 Professors

Els professors veuran els alumnes de les assignatures que ells imparteixen. Des d'aquí ha de poder introduir les qualificacions dels estudiants. També per cada assignatura que exerceixi podrà crear notificacions i avisos.

### 2.1.1.3 Alumnes

L'objectiu d'aquest portal, des del punt de vista dels alumnes, és poder principalment visualitzar el seu expedient acadèmic i fer el tràmit de la matriculació en el següent any acadèmic. Hauria de poder veure els avisos que són a nivell de l'escola, del curs en el que es troba, de l'aula a la que assisteix o bé de l'assignatura que estudia dintre del curs.

### 2.1.1.4 Pares/Tutors Legals

Aquest rol hauria de poder veure l'expedient acadèmic dels relacionats a ells (fills), així com veure els avisos i notificacions que només afecten als seus fills. Durant el període d'inscripció, podrà matricular als seus fills.

## 2.1.2 Identificació de les funcionalitats de cada subsistema

Els projecte el podem dividir en 6 subsistemes, que en el següent punt els anirem desglossant:

- Autenticació de l'usuari
- Subsistema de Manteniment:
  - Manteniment d'any acadèmic
  - Manteniment d'assignatures
  - Manteniment de tipus de qualificacions
  - Manteniment de tipus de titulació
  - Manteniment de cursos
  - Manteniment d'aules
- Subsistema d'Usuaris:
  - Manteniment de personal administratiu
  - Manteniment de professors
  - Manteniment d'alumnes
  - Manteniment de pares
- Subsistema d'avisos i notificacions
- Subsistema matriculació
- Subsistema expedient

## 2.1.3 Descripció de les funcionalitats de cada subsistema

### 2.1.3.1 Autenticació de l'usuari

L'usuari, per poder entrar al sistema (*login*), ha d'estar registrat en base de dades. Per poder entrar al sistema haurà d'introduir un usuari i una contrasenya. Si els valors són correctes, aquest entrarà al portal.

Depenent del tipus d'usuari que sigui tindrà més o menys opcions dintre del portal.

Si l'usuari es desconnecta (*logout*) i desitja entrar de nou a l'aplicació, s'haurà de tornar a identificar.

## 2.1.3.2 Subsistema de Manteniment

### 2.1.3.2.1 *Manteniment d'any acadèmic*

Aquest subsistema permetrà al personal administratiu gestionar el període de vigència de cada any acadèmic. En cada període de matriculació els administratius hauran de crear un nou any acadèmic per poder així matricular-se els alumnes.

### 2.1.3.2.2 *Manteniment d'assignatures*

El personal administratiu podrà gestionar les diferents assignatures que imparteix el centre.

### 2.1.3.2.3 *Manteniment de tipus de qualificacions*

Depenent a cada centre, es podran definir el tipus de qualificacions que s'utilitzarà. Diferents qualificacions podrien ser:

- Matrícula Honor, Excel·lent, Notable, Bé, Suficient, Insuficient, Molt deficient
- A+,A,B,C+,C-,D,E

### 2.1.3.2.4 *Manteniment de tipus de titulació*

Cada escola pot impartir diferents titulacions com són l'educació primària, educació secundària (ESO), batxillerat...

### 2.1.3.2.5 *Manteniment de cursos*

Des d'aquest manteniment es podrà gestionar els diferents cursos que s'imparteixen en el centre per cada tipus de titulació.

### 2.1.3.2.6 *Manteniment d'aules*

Depenent de cada any acadèmic podrà haver més o menys aules associades a cada curs.

## 2.1.3.3 Subsistema d'usuari

Permetrà gestionar el personal actiu en l'Escola. Existeixen cinc perfils: Administrador, Personal Administratiu, Professor, Alumne, Pare/Tutor Legal.

## 2.1.3.4 Subsistema de notificacions

Gestionar qualsevol tipus de notificacions i avisos que l'Escola hagi de fer per mantenir informats del dia a dia als professors, pares i/o alumnes.

Existiran diferents nivells de notificacions:

- **Escola:** L'avís serà adreçat a tot els usuaris de l'escola. Aquest tipus de missatge només el podrà crear el personal administratiu.
- **Curs:** La notificació serà dirigida als professors que imparteixen en el curs, al alumnes que estudien aquell curs (o els pares dels alumnes). Aquest tipus de missatge només el podrà donar d'alta el personal administratiu.
- **Aula:** L'avís serà dirigit als professors que imparteixen en l'aula, al alumnes que estudien en aquella aula (o els pares dels alumnes). Aquest tipus de missatge només el podrà enviar el personal administratiu.

- **Assignatura en un curs:** La notificació serà dirigida als professors que imparteixen aquella assignatura en un curs, als alumnes que estudien aquella assignatura (o els pares dels alumnes). Aquest tipus de missatge el podrà crear el personal administratiu o bé el professor que dona l'assignatura.

### 2.1.3.5 Subsistema de matriculació

Des d'aquest subsistema es permetrà realitzar totes les operacions referents a la matriculació dels alumnes a un curs. S'haurà de controlar que ha d'haver aprovat el curs anterior (no pot haver suspès més de 2 assignatures). Si l'alumne té més de dues assignatures suspeses, s'haurà de tornar a matricular en el mateix curs en el nou any acadèmic.

### 2.1.3.6 Subsistema de gestió de l'expedient

Permet gestionar les qualificacions de cada assignatura de cada alumne. Cada alumne o pare de l'alumne podrà veure la relació de les notes de cada assignatura del curs vigent (o inclús veure també les dels anys anteriors). El professor podrà introduir les notes de cada alumne que pertany a l'aula on ell imparteix les seves assignatures.

Dividirem el curs en tres trimestres i es podran realitzar dos exàmens de recuperació en el cas que s'hagi suspès. Per tant les notes les podríem desglossar de la següent manera:

Assignatura	Qualificacions					
	1er Trimestre	2on Trimestre	3er Trimestre	Repesca 1	Repesca 2	Nota Final
Nom assignatura	Qualificació 1	Qualificació 2	Qualificació 3	Qualificació Repesca 1 (si s'ha suspès l'assignatura)	Qualificació Repesca 2 (si s'ha suspès la repesca 1)	Qualificació final

El professor podrà introduir fins a un màxim de 6 notes al llarg de l'any acadèmic d'una assignatura: 1<sup>er</sup> trimestre, 2<sup>on</sup> trimestre, 3<sup>er</sup> trimestre, Repesca 1, Repesca 2, Nota Final. Si l'alumne no ha suspès res al final de l'any acadèmic ha de tenir 4 qualificacions: 1<sup>er</sup> trimestre, 2<sup>on</sup> trimestre, 3<sup>er</sup> trimestre i Nota Final.

### 2.1.4 Recursos necessaris

El programari estarà distribuït en un servidor. La base de dades estarà en el mateix servidor (depenent del volum que tinguem es podria utilitzar un altre servidor).

Per part dels usuaris, podran accedir-hi mitjançant un ordinador, apuntant a un direcció d'una pàgina web, des d'intranet (en l'Escola) o bé des d'Internet.

### 2.1.5 Glossari de les entitats i actors principals del programari

**Administratiu:** Persona encarregada de gestionar els tràmits que es realitzen en el centre docent. S'encarregarà del manteniment del programari perquè es puguin realitzar les gestions acadèmiques i matriculacions dintre del centre.

**Alumne:** Persona que cursa estudis en el centre.

**Any acadèmic:** Període lectiu on es realitza el curs escolar. Comprèn el curs escolar, les vacances i les festes.

**Aula:** Cadascuna de les sales destinades a fer-hi classes.

**Assignatura:** Lliçons que imparteix un professor i que l'estudiant ha de cursar obligatòriament per obtenir el títol.

**Curs:** El temps assenyalat dins de cada any acadèmic per a assistir a les lliçons.

**Escola:** Lloc, edifici, etc, on es desenvolupa l'activitat escolar i docent. Estarà formada per un conjunt d'alumnes, professors i aules.

**Estudiant:** Veure *alumne*

**Matrícula:** Procés on s'inscriu un alumne en un any acadèmic dintre d'un centre docent.

**Notes:** Veure *qualificacions*.

**Pare, mare:** Les persones que han engendrat i que són el responsables legals dels fills.

**Personal administratiu:** Veure *Administratiu*.

**Professor:** Persona que es dedica a l'ensenyament d'un art, una ciència o una llengua, etc. Serà l'encarregat de la introducció de les qualificacions de cada alumne.

**Responsable Legal:** Veure *tutor*

**Qualificacions:** Nota que s'utilitza per valorar una assignatura.

**Tutor Legal:** Persona que encara que no és el pare de l'alumne, és el responsable legal.

**Usuari:** Persona que farà servir el programari a desenvolupar.

## 2.1.6 Casos d'ús

Des del punt de vista tècnic, el document d'especificació inclou una llista de les funcionalitats que ofereix el programari i una descripció precisa de cada una. Una bona part d'aquestes funcionalitats es poden especificar en termes de precondició i postcondició, mitjançant els casos d'ús.

### 2.1.6.1 Autenticació de l'usuari

L'usuari, per poder entrar al sistema, ha d'estar registrat en base de dades. Per poder entrar al sistema haurà d'introduir un usuari i una contrasenya. Si els valors són correctes, aquest entrarà al portal.

#### 2.1.6.1.1 Entrada al sistema

- **Resum funcionalitat general:** Permetrà accedir un usuari al programari.
- **Paper dins el treball de l'usuari:** És un cas principal ja que ens hem de connectar cada vegada que volem entrar al sistema.
- **Actor:** personal administratiu, administrador, professor, alumne, pare (tutor, responsable, legal)
- **Casos d'ús relacionats:** Sortida del sistema, modificació contrasenya
- **Precondició:** L'usuari ha d'existir i haurà de saber el seu codi d'usuari i contrasenya.
- **Postcondició:** Depenent del seu perfil, l'usuari accedirà a les opcions que tingui accés dintre del sistema.
- **Observacions:** La primera vegada que entri al sistema un usuari no tindrà contrasenya i accedirem obligatòriament a informar la nova contrasenya. La paraula de pas s'encriptarà utilitzant l'algorisme MD5.

#### 2.1.6.1.2 Sortida del sistema

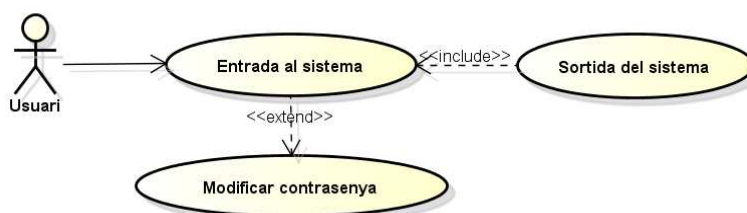
- **Resum funcionalitat general:** Permetrà abandonar la sessió del programari.
- **Paper dins el treball de l'usuari:** És un cas principal ja que ens hem de desconnectar cada vegada que volem sortir del sistema.

- **Actor:** personal administratiu, administrador, professor, alumne, pare (tutor, responsable legal)
- **Casos d'ús relacionats:** Entrada al sistema
- **Precondició:** Hem d'estar dintre del sistema.
- **Postcondició:** -
- **Observacions:** Passats 15 minuts d'inactivitat dintre de la sessió, aquesta quedarà expirada.

### 2.1.6.1.3 Modificació de contrasenya

- **Resum funcionalitat general:** Permetrà canviar la contrasenya de l'usuari.
- **Paper dins el treball de l'usuari:** És un cas principal ja que cada usuari hauria de canviar la seva contrasenya cada cert temps.
- **Actor:** personal administratiu, administrador, professor, alumne, pare (tutor, responsable legal)
- **Casos d'ús relacionats:** Entrada al sistema
- **Precondició:** Hem d'estar dintre del sistema.
- **Postcondició:** La contrasenya estarà canviada en la base de dades.
- **Observacions:** Cada vegada que es crea un usuari en el sistema, la primera vegada que accedeixi s'haurà d'actualitzar la contrasenya.

### 2.1.6.1.4 Diagrama cas d'ús autenticació del sistema



Il·lustració 2 Cas d'ús autenticació del sistema

## 2.1.6.2 Subsistema de Manteniment

### 2.1.6.2.1 Manteniment d'any acadèmic

#### 2.1.6.2.1.1 Alta any acadèmic

- **Resum funcionalitat general:** Permetrà crear un nou any acadèmic associat al centre escolar.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria ha de crear-ne un nou registre cada any.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca any acadèmic
- **Precondició:** L'any acadèmic no existeix en base de dades
- **Postcondició:** L'any acadèmic ha de quedar registrat en la base de dades.
- **Observacions:** Els anys acadèmics no poden estar solapats en el període de vigència.

#### 2.1.6.2.1.2 Modificació any acadèmic

- **Resum funcionalitat general:** Permetrà modificar un any acadèmic ja existent en base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creat l'any acadèmic no s'ha de modificar cap dada.

- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca any acadèmic
- **Precondició:** L'any acadèmic existeix en base de dades
- **Postcondició:** L'any acadèmic ha de quedar registrat en la base de dades amb les modificacions realitzades.
- **Observacions:** Els anys acadèmics no poden estar solapats en el període de vigència.

#### 2.1.6.2.1.3 Eliminació any acadèmic

- **Resum funcionalitat general:** Permetrà eliminar un any acadèmic de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap any acadèmic.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca any acadèmic
- **Precondició:** L'any acadèmic existeix en base de dades
- **Postcondició:** L'any acadèmic no existirà en base de dades.
- **Observacions:** Només es podrà eliminar un any acadèmic si no existeix cap alumne matriculat en ell.

#### 2.1.6.2.1.4 Consulta any acadèmic

- **Resum funcionalitat general:** Permetrà consultar un any acadèmic de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha de consultar mai cap dada de l'any acadèmic.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca any acadèmic
- **Precondició:** L'any acadèmic existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades de l'any acadèmic.
- **Observacions:-**

#### 2.1.6.2.1.5 Cerca any acadèmic

- **Resum funcionalitat general:** Permetrà cercar els anys acadèmics de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat l'any acadèmic es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta any acadèmic
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació d'anys acadèmics que compleixen els criteris de cerca.
- **Observacions:-**

### 2.1.6.2.2 Manteniment d'assignatures

#### 2.1.6.2.2.1 Alta assignatura

- **Resum funcionalitat general:** Permetrà crear una nova assignatura al sistema.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria només en la implantació inicial del sistema s'utilitzarà aquesta opció.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca assignatura
- **Precondició:** L'assignatura no existeix en base de dades
- **Postcondició:** L'assignatura ha de quedar registrada en la base de dades.



- **Observacions:** -.

#### 2.1.6.2.2.2 Modificació assignatura

- **Resum funcionalitat general:** Funcionalitat que permet al personal administratiu fer algun canvi referent a l'assignatura.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creades les assignatures no s'ha de modificar cap dada.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca assignatura
- **Precondició:** L'assignatura existeix en base de dades
- **Postcondició:** L'assignatura ha de quedar registrada en la base de dades amb les modificacions realitzades.
- **Observacions:** Es pot realitzar una baixa lògica mitjançant la modificació ja que modificarem l'estat passant d'alta a baixa. D'aquesta manera no esborrem cap informació però deixem de treballar amb ella.

#### 2.1.6.2.2.3 Eliminació assignatura

- **Resum funcionalitat general:** Permetrà eliminar una assignatura de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap assignatura (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca assignatura
- **Precondició:** L'assignatura existeix en base de dades
- **Postcondició:** L'assignatura no existirà en base de dades.
- **Observacions:** Només es podrà eliminar només si no existeix cap curs que imparteixi aquesta assignatura.

#### 2.1.6.2.2.4 Consulta assignatura

- **Resum funcionalitat general:** El personal administratiu podrà consultar la informació referent a cada assignatura.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha de consultar mai cap dada de l'assignatura.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca assignatura
- **Precondició:** L'assignatura existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades de l'assignatura.
- **Observacions:-**

#### 2.1.6.2.2.5 Cerca assignatures

- **Resum funcionalitat general:** Permetrà cercar les assignatures de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat les assignatures es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta assignatura
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació d'assignatures que compleixen els criteris de cerca.
- **Observacions:-**

### 2.1.6.2.3 Manteniment de tipus de qualificacions



Depenent a cada centre, es podran definir el tipus de qualificacions que s'utilitzarà.

#### 2.1.6.2.3.1 Alta tipus de qualificacions

- **Resum funcionalitat general:** Permet afegir un nou tipus de qualificació al sistema.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria només en la implantació inicial del sistema s'utilitzarà aquesta opció.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de qualificacions
- **Precondició:** El tipus de qualificació no existeix en base de dades
- **Postcondició:** El tipus de qualificació ha de quedar registrat en la base de dades.
- **Observacions:** Cada tipus de qualificació anirà associat a 'Aprovat' o 'No aprovat'.

#### 2.1.6.2.3.2 Modificació tipus de qualificacions

- **Resum funcionalitat general:** Funcionalitat que permet al personal administratiu fer algun canvi referent al tipus de qualificació.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creats els tipus de qualificacions no s'ha de modificar cap dada.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de qualificacions
- **Precondició:** El tipus de qualificació existeix en base de dades
- **Postcondició:** El tipus de qualificació ha de quedar registrat en la base de dades amb les modificacions realitzades.
- **Observacions:** Es pot realitzar una baixa lògica mitjançant la modificació ja que modificarem l'estat passant d'alta a baixa. D'aquesta manera no esborrem cap informació si un dia deixem de treballar amb ella.

#### 2.1.6.2.3.3 Eliminació tipus de qualificacions

- **Resum funcionalitat general:** Permetrà eliminar un tipus de qualificació de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap tipus de qualificació (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de qualificacions
- **Precondició:** El tipus de qualificació existeix en base de dades
- **Postcondició:** El tipus de qualificació no existirà en base de dades.
- **Observacions:** Es podrà eliminar només si cap professor ha qualificat a algun alumne en alguna de les assignatures que imparteix.

#### 2.1.6.2.3.4 Consulta tipus de qualificacions

- **Resum funcionalitat general:** El personal administratiu podrà consultar la informació referent a cada tipus de qualificació.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha de consultar mai cap dada del tipus de qualificació.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de qualificacions
- **Precondició:** El tipus de qualificació existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades del tipus de qualificació
- **Observacions:-**

#### 2.1.6.2.3.5 Cerca tipus de qualificacions

- **Resum funcionalitat general:** Permetrà cercar els tipus de qualificacions de la base de dades mitjançant diferents criteris de selecció.

- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat el tipus de qualificació es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta de tipus de qualificacions
- **Precondició:-**
- **Postcondició:** Es visualitzarà la relació de tipus de qualificacions que compleixen els criteris de cerca.
- **Observacions:-**

#### 2.1.6.2.4 Manteniment de tipus de titulació

Cada escola pot impartir diferents titulacions com són l'educació primària, educació secundària (ESO), batxillerat...

##### 2.1.6.2.4.1 Alta tipus de titulacions

- **Resum funcionalitat general:** Permet afegir un nou tipus de titulació al sistema.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria només en la implantació inicial del sistema s'utilitzarà aquesta opció.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de titulacions
- **Precondició:** El tipus de titulació no existeix en base de dades
- **Postcondició:** El tipus de titulació ha de quedar registrat en la base de dades.
- **Observacions:-.**

##### 2.1.6.2.4.2 Modificació tipus de titulacions

- **Resum funcionalitat general:** Funcionalitat que permet al personal administratiu fer algun canvi referent al tipus de titulació.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creats els tipus de titulacions no s'ha de modificar cap dada.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de titulacions
- **Precondició:** El tipus de titulació existeix en base de dades
- **Postcondició:** El tipus de titulació ha de quedar registrat en la base de dades amb les modificacions realitzades.
- **Observacions:** Es pot realitzar una baixa lògica mitjançant la modificació ja que modificarem l'estat passant d'alta a baixa. D'aquesta manera no esborrem cap informació si un dia deixem de treballar amb ella.

##### 2.1.6.2.4.3 Eliminació tipus de titulacions

- **Resum funcionalitat general:** Permetrà eliminar un tipus de titulació de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap tipus de titulació (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de titulacions
- **Precondició:** El tipus de titulació existeix en base de dades
- **Postcondició:** El tipus de titulació no existirà en base de dades.
- **Observacions:** Es podrà eliminar només si cap alumne s'ha matriculat en algun curs d'aquesta titulació.

##### 2.1.6.2.4.4 Consulta tipus de titulacions

- **Resum funcionalitat general:** El personal administratiu podrà consultar la informació referent a cada tipus de titulació.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha de consultar mai cap dada del tipus de titulació.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca tipus de titulacions
- **Precondició:** El tipus de titulació existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades del tipus de titulació
- **Observacions:-**

#### 2.1.6.2.4.5 Cerca tipus de titulacions

- **Resum funcionalitat general:** Permetrà cercar els tipus de titulacions de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat els tipus de titulacions es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta de tipus de titulacions
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació de tipus de titulacions que compleixen els criteris de cerca.
- **Observacions:-**

#### 2.1.6.2.5 Manteniment de cursos

Des d'aquest manteniment es podrà gestionar els diferents cursos que s'imparteixen en el centre per cada tipus de titulació.

##### 2.1.6.2.5.1 Alta cursos

- **Resum funcionalitat general:** Permet afegir un nou curs al sistema. Associarem cada curs al tipus de titulació. Podrem relacionar cada curs amb n registres d'assignatures.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria només en la implantació inicial del sistema s'utilitzarà aquesta opció.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca de cursos
- **Precondició:** El curs no existeix en base de dades.
- **Postcondició:** El curs ha de quedar registrat en la base de dades.
- **Observacions:** El tipus de titulació associat al curs ja existeix en base de dades.

##### 2.1.6.2.5.2 Modificació cursos

- **Resum funcionalitat general:** Funcionalitat que permet al personal administratiu fer algun canvi referent al curs.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creats els cursos no s'ha de modificar cap dada.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca de cursos
- **Precondició:** El curs existeix en base de dades
- **Postcondició:** El curs ha de quedar registrat en la base de dades amb les modificacions realitzades.
- **Observacions:** Es pot realitzar una baixa lògica mitjançant la modificació ja que modificarem l'estat passant d'alta a baixa. D'aquesta manera no esborrem cap informació si un dia deixem de treballar amb ella.

##### 2.1.6.2.5.3 Eliminació cursos

- **Resum funcionalitat general:** Permetrà eliminar un curs de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap curs (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca de cursos
- **Precondició:** El curs existeix en base de dades
- **Postcondició:** El curs no existirà en base de dades.
- **Observacions:** Es podrà eliminar només si cap alumne s'ha matriculat en aquest curs.

#### 2.1.6.2.5.4 Consulta cursos

- **Resum funcionalitat general:** El personal administratiu podrà consultar la informació referent a cada curs.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha de consultar mai cap dada del curs.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca de cursos
- **Precondició:** El curs existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades del curs
- **Observacions:-**

#### 2.1.6.2.5.5 Cerca cursos

- **Resum funcionalitat general:** Permetrà cercar els cursos de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat els cursos es podrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta de cursos
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació de cursos que compleixen els criteris de cerca.
- **Observacions:-**

#### 2.1.6.2.6 Manteniment d'aules

Depenent de cada any podrà haver més o menys aules associades a cada curs.

##### 2.1.6.2.6.1 Alta aules

- **Resum funcionalitat general:** Permet afegir una nova aula al sistema. Aquesta aula estarà associada a un curs i a un any acadèmic. Al final de cada any acadèmic aquesta aula deixarà d'estar operativa. Es podrà definir un aforament màxim d'alumnes.
- **Paper dins el treball de l'usuari:** És un cas que es farà servir en cada període d'inscripció en cada any acadèmic.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca d'aules
- **Precondició:** L'aula no existeix en base de dades.
- **Postcondició:** L'aula ha de quedar registrat en la base de dades.
- **Observacions:** El curs associat a l'aula ha d'existir en base de dades.

##### 2.1.6.2.6.2 Modificació aules

- **Resum funcionalitat general:** Funcionalitat que permet al personal administratiu fer algun canvi referent a l'aula.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creades les aules no s'ha de modificar cap dada.
- **Actor:** personal administratiu, administrador

- **Casos d'ús relacionats:** Cerca d'aules
- **Precondició:** L'aula existeix en base de dades
- **Postcondició:** L'aula ha de quedar registrada en la base de dades amb les modificacions realitzades.
- **Observacions:** Es pot realitzar una baixa lògica mitjançant la modificació ja que modificarem l'estat passant d'alta a baixa. D'aquesta manera no esborrem cap informació si un dia deixem de treballar amb ella.

#### 2.1.6.2.6.3 Eliminació aules

- **Resum funcionalitat general:** Permetrà eliminar una aula de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap aula (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca d'aules
- **Precondició:** L'aula existeix en base de dades
- **Postcondició:** L'aula no existirà en base de dades.
- **Observacions:** Es podrà eliminar només si cap alumne pertany a aquesta aula.

#### 2.1.6.2.6.4 Consulta aules

- **Resum funcionalitat general:** El personal administratiu podrà consultar la informació referent a cada aula.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha de consultar mai cap dada del aula.
- **Actor:** personal administratiu, administrador, professor
- **Casos d'ús relacionats:** Cerca d'aules
- **Precondició:** L'aula existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades del aula
- **Observacions:** El professor podrà consultar les aules en les que ell impartirà les assignatures.

#### 2.1.6.2.6.5 Cerca aules

- **Resum funcionalitat general:** Permetrà cercar els aules de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat les aules es podrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador, professor
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta d'aules
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació d'aules que compleixen els criteris de cerca.
- **Observacions:** El professor, al realitzar la cerca, tindrà com a resultat les aules on ell impartirà les assignatures.

#### 2.1.6.2.6.6 Gestió automàtica assignació dels alumnes a cada aula

- **Resum funcionalitat general:** El personal administratiu podrà ubicar automàticament els alumnes matriculats a les aules creades.
- **Paper dins el treball de l'usuari:** Aquest cas d'ús es farà servir en cada període de matriculació.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Consulta d'aules, Consulta d'alumnes, Consulta de Matriculació
- **Precondició:** Les aules i els alumnes han d'existir en base de dades. L'alumne ha d'estar matriculat en l'any acadèmic actual.

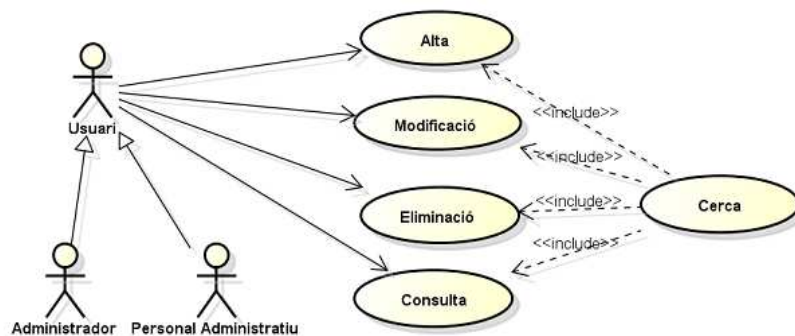
- **Postcondició:** Automàticament si l'alumne no té aula assignada se li associarà una aula on hi hagi places vacants.
- **Observacions:** Aquest procés permet agilitzar la feina del personal administratiu. De totes maneres, el propi administratiu pot manualment assignar un alumne a una aula.

### 2.1.6.2.6.7 Assignació professor a aules i assignatura

- **Resum funcionalitat general:** Definir per cada aula quin professor haurà d'impartir cada assignatura.
- **Paper dins el treball de l'usuari:** Aquest procés s'haurà de realitzar al inici de cada any acadèmic o bé si hi ha alguna baixa d'un professor i ha de ser substituït per un altre.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Consulta d'aules, Modificació d'aules
- **Precondició:** El professor existeix en base de dades i l'assignatura que s'imparteix en una aula no té un professor assignat.
- **Postcondició:** L'assignatura que s'imparteix en una aula tindrà un professor associat.
- **Observacions:** -.

### 2.1.6.2.7 Diagrama casos d'ús subsistema manteniment

Per tot els casos d'ús abans esmentats (a excepció del manteniment de les aules), el diagrama tindrà el següent format:

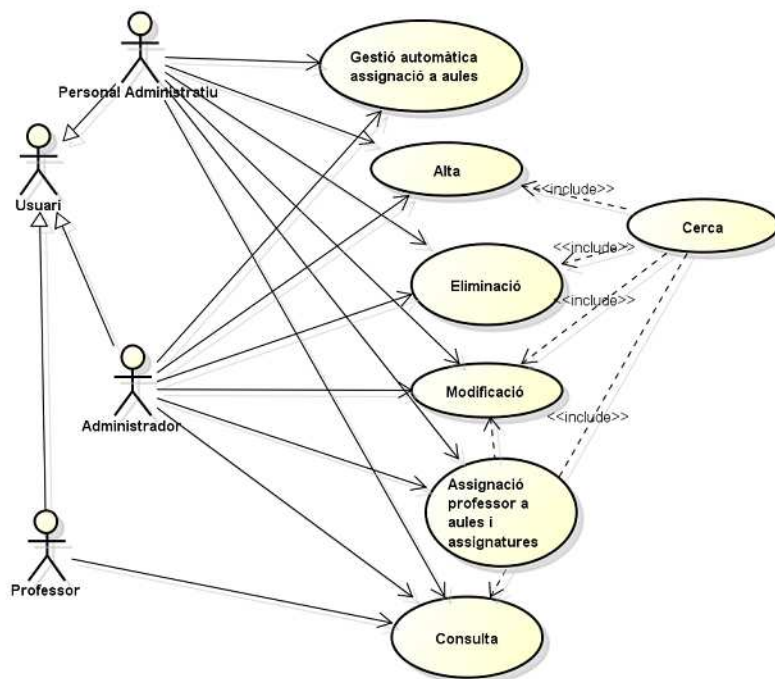


Il·lustració 3. Cas d'ús Subsistema de Manteniment

#### 2.1.6.2.7.1 Diagrama cas d'ús manteniment d'aules

El diagrama pel manteniment d'aules, que és diferent als altres manteniments abans explicats, té el següent format:





Il·lustració 4. Cas d'ús Manteniment d'aules

### 2.1.6.3 Subsistema d'usuari

Permetrà gestionar els usuaris en l'Escola.

#### 2.1.6.3.1 Alta usuari

- **Resum funcionalitat general:** Permetrà crear un nou usuari al sistema. S'haurà d'indicar si és personal administratiu, professor, alumne o pare.
- **Paper dins el treball de l'usuari:** És un cas principal ja que cada vegada que hi hagi una alta s'haurà d'utilitzar.
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca d'usuari
- **Precondició:** L'usuari no existeix en base de dades
- **Postcondició:** L'usuari ha de quedar registrat en la base de dades.
- **Observacions:** Cada vegada que es crea un usuari automàticament generarà el codi d'usuari. Seguirem la nomenclatura "4 primeres lletres del primer cognom" + "\_" + "3 primeres lletres del nom". En el cas que el codi d'usuari ja existeixi anirem substituïm l'última lletra per un número correlatiu. Exemple: Garcia Merino, Manel, el seu usuari serà GARC\_MAN. En el cas que existeixi un altre usuari amb el mateix codi serà reemplaçat per GARC\_MA1 i així successivament. En el cas que el tipus d'usuari sigui personal administratiu, professor o pare el tipus de document serà obligatori i es validarà si el NIF és correcta. En el cas que sigui un alumne haurem d'especificar quin és el curs en el que es matricularà per primera vegada a l'Escola. Si es desitja, per l'alumne, també es podrà informar les dades bancàries (també es crearà una validació si estan informades). Al crear l'estudiant automàticament es generarà la primera matrícula (sense assignar a una aula).

#### 2.1.6.3.2 Modificació usuari

- **Resum funcionalitat general:** Funcionalitat que permet a l'usuari fer algun canvi referent a la fitxa de l'usuari.

- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creades les dades del personal administratiu no s'ha de modificar cap dada.
- **Actor:** personal administratiu, administrador, professor, alumne, pare
- **Casos d'ús relacionats:** Cerca d'usuari.
- **Precondició:** L'usuari existeix en base de dades
- **Postcondició:** L'usuari ha de quedar registrat en la base de dades amb les modificacions realitzades.
- **Observacions:** Es pot realitzar una baixa lògica mitjançant la modificació ja que modificarem l'estat passant d'alta a baixa. D'aquesta manera no esborrem cap informació però deixem de treballar amb ella. L'administrador i el personal administratiu podrà modificar tota la informació referent a tots els usuaris de la base de dades. També podrà reiniciar la contrasenya de qualsevol usuari en el cas d'oblit, per així que al tornar-se a connectar l'usuari pugui actualitzar la seva paraula de pas. El professor i l'alumne només podran modificar les seves dades personals. Els pares podran modificar-se les seves dades personals, però també les del seus fills.

### 2.1.6.3.3 Eliminació usuari

- **Resum funcionalitat general:** Permetrà eliminar un personal administratiu de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap dada de l'usuari (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca d'usuari
- **Precondició:** L'usuari existeix en base de dades
- **Postcondició:** L'usuari no existirà en base de dades.
- **Observacions:** Si el personal administratiu té una notificació associada no es podrà eliminar. Si el professor imparteix alguna assignatura no es podrà eliminar. Si l'alumne està matriculat i amb alguna qualificació no es podrà eliminar. En aquests casos s'haurà de fer una baixa lògica.

### 2.1.6.3.4 Consulta usuari

- **Resum funcionalitat general:** L'usuari podrà consultar la informació de la seva fitxa.
- **Paper dins el treball de l'usuari:** És un cas principal ja que en teoria cada usuari pot consultar la seva fitxa.
- **Actor:** personal administratiu, administrador, professor, alumne, pare.
- **Casos d'ús relacionats:** Cerca d'usuari.
- **Precondició:** L'usuari existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades de l'usuari.
- **Observacions:** L'administrador i el personal administratiu podrà consultar tota la informació referent a tots els usuaris de la base de dades. El professor podrà consultar les seves dades personals però també les dels alumnes que estan en les aules de les assignatures que ell imparteix. L'alumne només podrà consultar les seves dades personals. Els pares podran consultar les seves dades personals, però també les del seus fills.

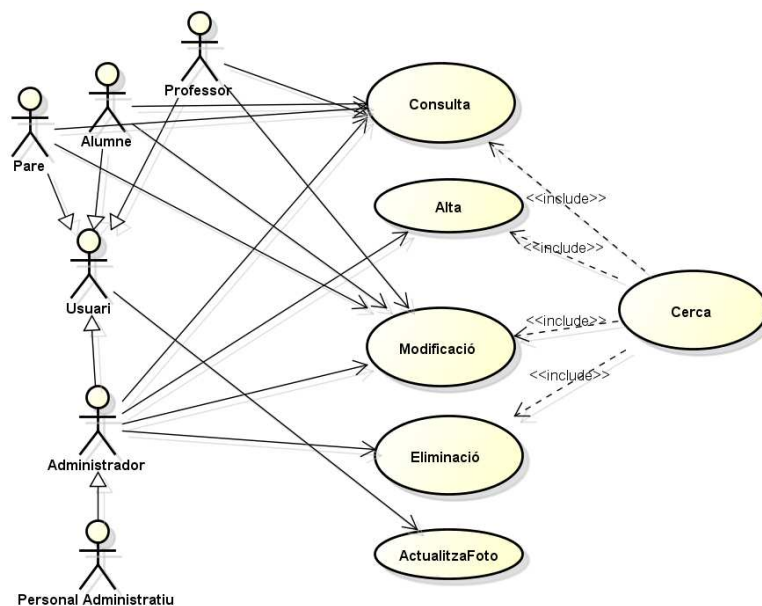
### 2.1.6.3.5 Cerca usuari

- **Resum funcionalitat general:** Permetrà cercar els usuaris de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat l'usuari es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador, professor, alumne, pare
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta d'usuari



- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació d'usuaris que compleixen els criteris de cerca.
- **Observacions:** L'administrador i el personal administratiu podrà cercar tots els usuaris de la base de dades. El professor podrà cercar-ne a ell mateix o bé als alumnes que estan en les aules de les assignatures que ell imparteix. L'alumne només podrà cercar-se a ell mateix. Els pares podrà cercar als seus fills o bé a ell mateix.

### 2.1.6.3.6 Diagrama cas d'ús manteniment d'usuaris



II-lustració 5. Cas d'ús manteniment usuaris

### 2.1.6.4 Subsistema de notifikacions

Permet gestionar qualsevol tipus de notifikacions i avisos que l'Escola hagi de fer per mantenir informats del dia a dia als pares i/o alumnes.

#### 2.1.6.4.1 Alta notifiació

- **Resum funcionalitat general:** Permetrà crear un nou avís al sistema.
- **Paper dins el treball de l'usuari:** És un cas principal ja que es podran fer diferents tipus d'avisos per part de diferents actors.
- **Actor:** personal administratiu, administrador, professor
- **Casos d'ús relacionats:** Cerca de notifikacions
- **Precondició:** La notifiació no existeix en base de dades
- **Postcondició:** La notifiació ha de quedar registrat en la base de dades.
- **Observacions:** Al donar d'alta a una notifiació s'haurà d'informar quin tipus de notifiació és (escola, curs, aula, assignatura). Aquesta notifiació tindrà un període de vigència. El professor només podrà crear notifikacions d'assignatura.

#### 2.1.6.4.2 Modifiació notifiació

- **Resum funcionalitat general:** Funcionalitat que permet fer algun canvi referent a un notifiació.

- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creades les dades del notificació no s'ha de modificar cap dada.
- **Actor:** personal administratiu, administrador, professor.
- **Casos d'ús relacionats:** Cerca de notificacions
- **Precondició:** La notificació existeix en base de dades
- **Postcondició:** La notificació ha de quedar registrada en la base de dades amb les modificacions realitzades.
- **Observacions:** El professor només podrà modificar notificacions creades per ell.

#### 2.1.6.4.3 Eliminació notificació

- **Resum funcionalitat general:** Permetrà eliminar un notificació de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap notificació (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador, professor
- **Casos d'ús relacionats:** Cerca de notificacions
- **Precondició:** La notificació existeix en base de dades
- **Postcondició:** La notificació no existirà en base de dades.
- **Observacions:** El professor només podrà eliminar les creades per ell.

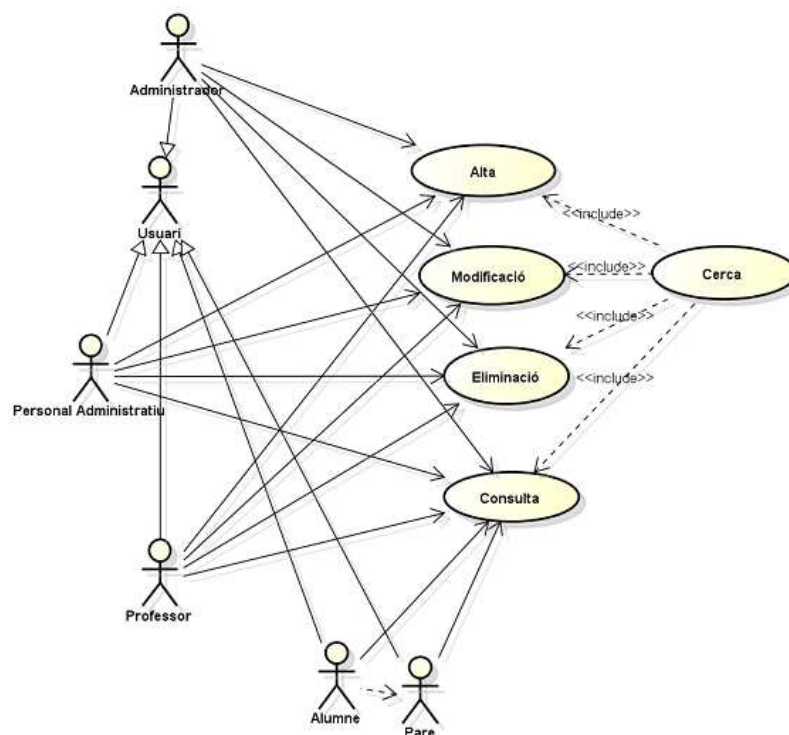
#### 2.1.6.4.4 Consulta notificació

- **Resum funcionalitat general:** Es permetrà visualitzar la informació de les notificacions.
- **Paper dins el treball de l'usuari:** És un cas principal ja que les notificacions podran ser consultades per part de diferents actors.
- **Actor:** personal administratiu, administrador, alumne, pare/tutor, professor
- **Casos d'ús relacionats:** Cerca de notificacions
- **Precondició:** La notificació existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades de la notificació.
- **Observacions:** L'alumne i pare/tutor podran veure els avisos a nivell d'escola, o si pertany al curs/aula/assignatura relacionat amb la notificació. El professor podrà visualitzar les dades de la notificació si és a nivell d'escola o si pertany al curs/aula/assignatura relacionat amb la notificació.

#### 2.1.6.4.5 Cerca notificacions

- **Resum funcionalitat general:** Permetrà cercar els notificacions de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat les notificacions es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador, professor, alumne, pare/tutor
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta de notificacions
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació de notificacions que compleixen els criteris de cerca.
- **Observacions:** : L'alumne i pare/tutor podran veure els avisos a nivell d'escola, o si pertany al curs/aula/assignatura relacionat amb la notificació. El professor podrà visualitzar les dades de la notificació si és a nivell d'escola o si pertany al curs/aula/assignatura relacionat amb la notificació.

#### 2.1.6.4.6 Diagrama cas d'ús mòdul de manteniment de notificacions



Il·lustració 6. Cas d'ús manteniment de notificacions

### 2.1.6.5 Subsistema de matriculació

Des d'aquest subsistema es permetrà realitzar totes les operacions referents a la matriculació dels alumnes a un curs.

#### 2.1.6.5.1 Alta matriculació

- **Resum funcionalitat general:** Permetrà crear un matriculació d'un alumne en un any acadèmic.
- **Paper dins el treball de l'usuari:** És un cas principal ja que cada alumne haurà de matricular-se en cada any acadèmic.
- **Actor:** personal administratiu, administrador, alumne, pare/tutor
- **Casos d'ús relacionats:** Cerca matriculacions, Alta usuari
- **Precondició:** La matriculació no existeix en base de dades
- **Postcondició:** La matriculació ha de quedar registrada en la base de dades.
- **Observacions:** L'alumne només podrà matricular-se a ell mateix. Els pares/tutors podran matricular al seus fills. El personal administratiu podrà matricular a qualsevol estudiant. Al realitzar l'alta d'una matriculació d'un estudiant en un curs s'haurà de validar que el curs anterior ha estat aprovat (com a màxim podrà tenir 2 assignatures suspeses). Si no és el cas, l'alumne haurà de repetir curs. Al fer una alta d'un usuari es crearà automàticament una alta de matrícula.

#### 2.1.6.5.2 Modificació matriculació

- **Resum funcionalitat general:** Funcionalitat que permet fer algun canvi referent a una matriculació.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria un cop creades les dades de matriculació no s'ha de modificar cap dada.

- **Actor:** personal administratiu, administrador.
- **Casos d'ús relacionats:** Cerca matriculacions
- **Precondició:** La matriculació existeix en base de dades
- **Postcondició:** La matriculació ha de quedar registrada en la base de dades amb les modificacions realitzades.
- **Observacions:** Només el personal administratiu podrà realitzar modificacions en la matrícula. La modificació podrà realitzar-se fora del període de matriculació. El personal administratiu podrà associar manualment un usuari a una aula (per no executar la gestió automàtica d'alumnes a les aules).

#### 2.1.6.5.3 *Eliminació matriculació*

- **Resum funcionalitat general:** Permetrà eliminar una matriculació de la base de dades.
- **Paper dins el treball de l'usuari:** És un cas esporàdic ja que en teoria no s'ha d'esborrar mai cap matriculació (es recomana fer una baixa lògica).
- **Actor:** personal administratiu, administrador
- **Casos d'ús relacionats:** Cerca matriculacions
- **Precondició:** La matriculació existeix en base de dades
- **Postcondició:** La matriculació no existirà en base de dades.
- **Observacions:** Només es podrà esborrar una matriculació si l'alumne no ha tingut cap qualificació durant l'any acadèmic. En el cas que l'alumne abandoni els estudis llavors el que s'haurà de fer és una baixa lògica de la matriculació.

#### 2.1.6.5.4 *Consulta matriculació*

- **Resum funcionalitat general:** Es permetrà visualitzar la informació de les matriculacions.
- **Paper dins el treball de l'usuari:** És un cas principal ja que les matriculacions podran ser consultades per part de diferents actors.
- **Actor:** personal administratiu, administrador, alumne, pare/tutor
- **Casos d'ús relacionats:** Cerca matriculacions
- **Precondició:** La matriculació existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades de la matriculació.
- **Observacions:** L'alumne només podrà visualitzar la seva matrícula. Els pares/tutors només podran veure la matrícula dels seus fills. El personal administratiu podrà consultar qualsevol matrícula de la base de dades.

#### 2.1.6.5.5 *Cerca matriculacions*

- **Resum funcionalitat general:** Permetrà cercar els matriculacions de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat les matriculacions es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador, alumne, pare/tutor
- **Casos d'ús relacionats:** Alta, eliminació, modificació, consulta de matriculacions
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació de matriculacions que compleixen els criteris de cerca.
- **Observacions:** L'alumne i pare/tutor no tindrà permís a les opcions de modificació i eliminació.

#### 2.1.6.5.6 *Diagrama cas d'ús mòdul de manteniment de matriculació*



- **Postcondició:** La qualificació ha de quedar registrada en la base de dades amb les modificacions realitzades.
- **Observacions:** El professor ha d'impartir l'assignatura a l'alumne en aquell curs.

#### 2.1.6.6.3 Consulta qualificació

- **Resum funcionalitat general:** Es permetrà visualitzar la informació de les qualificacions.
- **Paper dins el treball de l'usuari:** És un cas principal ja que les qualificacions podran ser consultades per part de diferents actors.
- **Actor:** personal administratiu, administrador, alumne, pare/tutor, professor
- **Casos d'ús relacionats:** Cerca qualificació
- **Precondició:** La matriculació existeix en base de dades
- **Postcondició:** Es visualitza un formulari amb les dades de la matriculació.
- **Observacions:** Cada alumne o pare de l'alumne podrà veure la relació de les notes de cada assignatura del curs vigent (o inclús veure també les dels anys anteriors). El professor podrà visualitzar les notes de cada alumne que pertany a l'aula on ell imparteix les seves assignatures. Podrà veure totes les notes de cada alumne.

#### 2.1.6.6.4 Cerca qualificacions

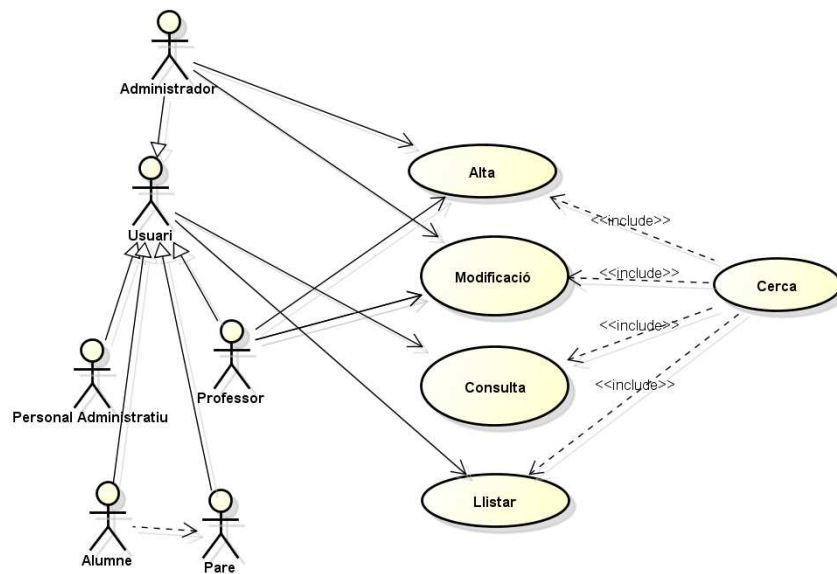
- **Resum funcionalitat general:** Permetrà cercar les qualificacions de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat les qualificacions es permetrà realitzar les opcions de consulta, modificació, eliminació (i alta en el cas que no existeixi).
- **Actor:** personal administratiu, administrador, alumne, tutor/pare, professor
- **Casos d'ús relacionats:** Alta, modificació, consulta, llistat de qualificacions
- **Precondició:** -
- **Postcondició:** Es visualitzarà la relació de qualificacions que compleixen els criteris de cerca.
- **Observacions:** Els alumnes només podran visualitzar les seves qualificacions. Els tutors/pares només podran veure les notes dels seus fills.

#### 2.1.6.6.5 Llistat qualificacions

- **Resum funcionalitat general:** Permetrà llistar les qualificacions de la base de dades mitjançant diferents criteris de selecció.
- **Paper dins el treball de l'usuari:** És un dels casos principals perquè un cop cercat les qualificacions es permetrà realitzar les opcions de llistar.
- **Casos d'ús relacionats:** Consulta de qualificacions
- **Actor:** personal administratiu, administrador, alumne, tutor/pare, professor
- **Precondició:** Haver realitzat una cerca inicial amb criteri de selecció.
- **Postcondició:** Es generarà un llistat que es podrà gravar en format PDF o bé imprimir-ho per impressora.
- **Observacions:** -

#### 2.1.6.6.6 Diagrama cas d'ús mòdul de la gestió de l'expedient





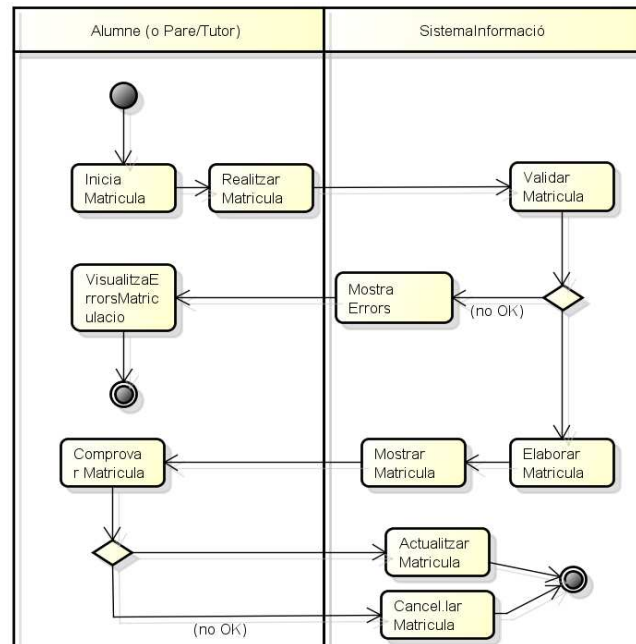
II-lustració 8. Cas d'ús gestió de l'expedient

## 2.1.7 Diagrama d'activitats

En UML, és possible representar d'una manera més detallada cada un dels processos del punt de vista de l'empresa mitjançant una activitat (amb el seu diagrama d'activitats associat), on cada un dels carrers (*swimlanes*) representa un rol involucrat en el procés, mentre que els nodes d'activitat representaran les accions i els nodes objecte als recursos.

### 2.1.7.1 Diagrama d'activitats de matriculació

A continuació s'exposa el diagrama d'activitat en el procés de matriculació.

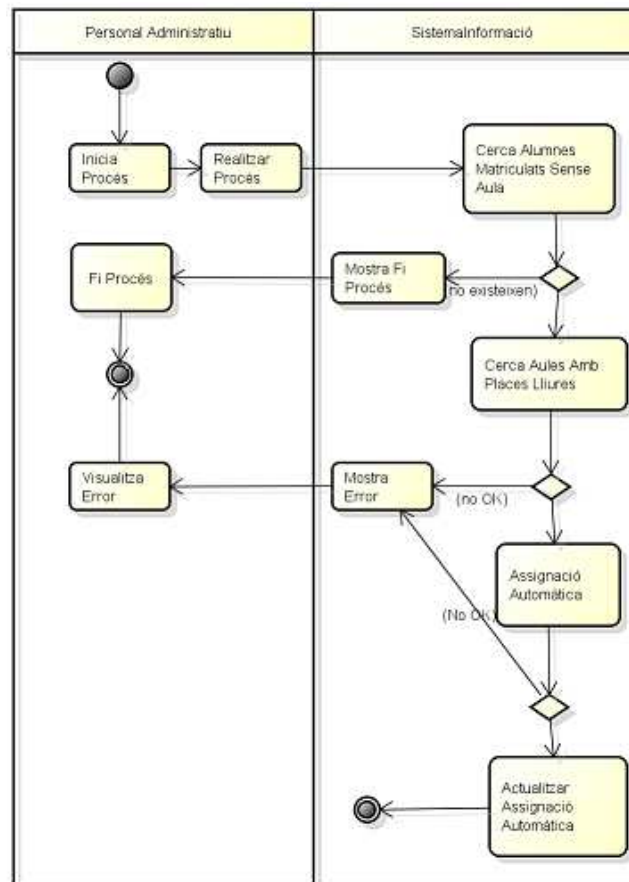


II-Il·lustració 9. Diagrama d'activitats d'una matriculació

### 2.1.7.2 Diagrama d'activitats d'assignació automàtica d'alumnes a aules

Diagrama d'activitats per assignar d'una manera automàtica els alumnes a cada aula després de cada període de matriculació. Aquest procés s'ha d'executar abans del inici de cada any acadèmic.





II-lustració 10. Diagrama d'activitats d'assignació automàtica d'alumnes a aules

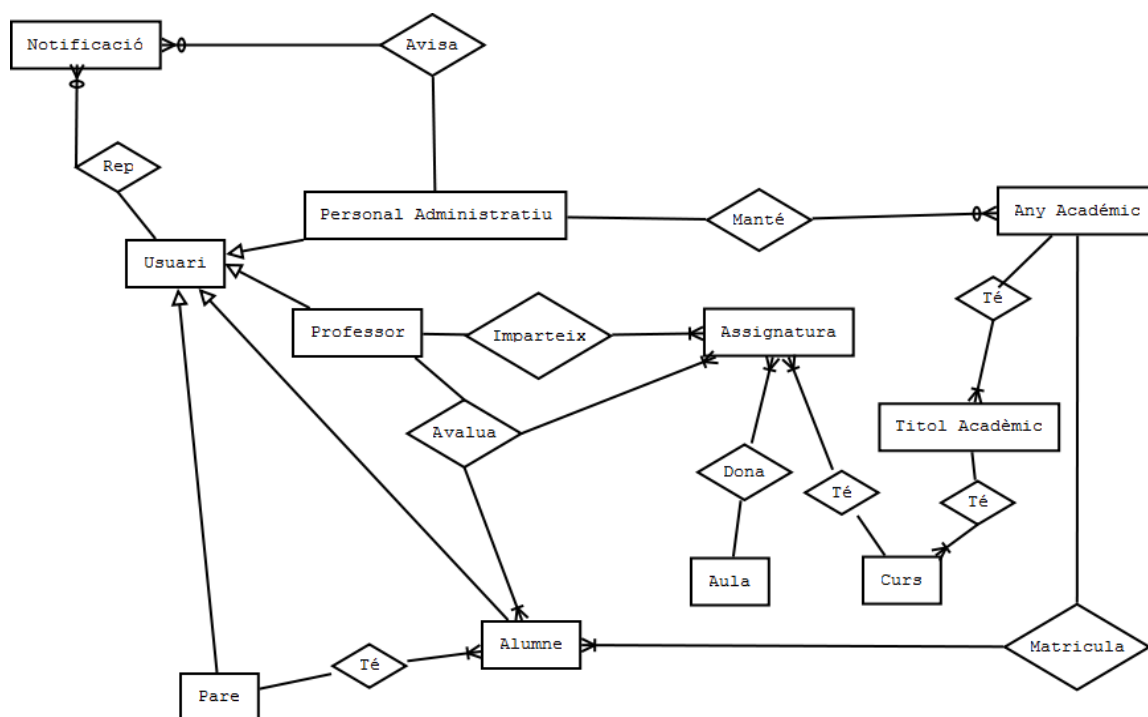
## 2.2 Punt de vista de la informació

Aquest punt de vista ens descriu amb detall la informació que tractarà el sistema, incloent-hi les estructures de les dades que manega el programari i les relacions entre aquestes. La informació aportada en aquesta vista ens permetrà garantir la coherència de les dades del sistema que gestionen les altres vistes ja que ens ofereix una visió global.

### 2.2.1 Diagrama de classes

El diagrama de classes ens mostra un esquema invariant. Aquesta representació ens descriu les estructures de dades que gestiona el sistema. Determina els tipus de dades, les seves relacions i algunes restriccions sobre ells (per exemple, la multiplicitat de les associacions).

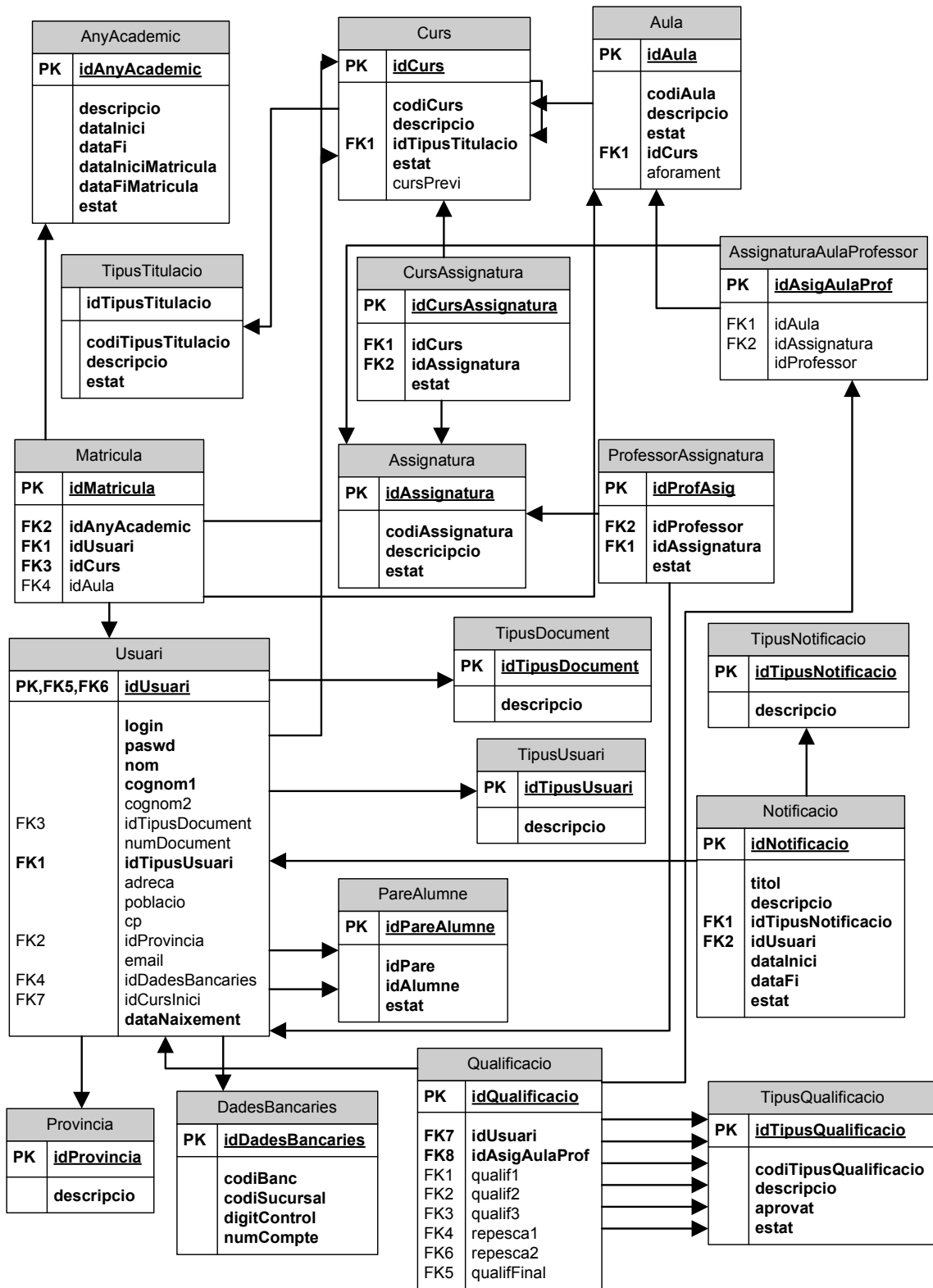




II-lustració 12. Diagrama Entitat Relació

### 2.2.3 Model lògic relacional

A partir del diagrama de classes i del diagrama entitat relació podem traduir l'esquema conceptual per produir un esquema lògic i un conjunt d'esquemes de transaccions que expressen l'estàtica i la dinàmica del sistema de la informació.



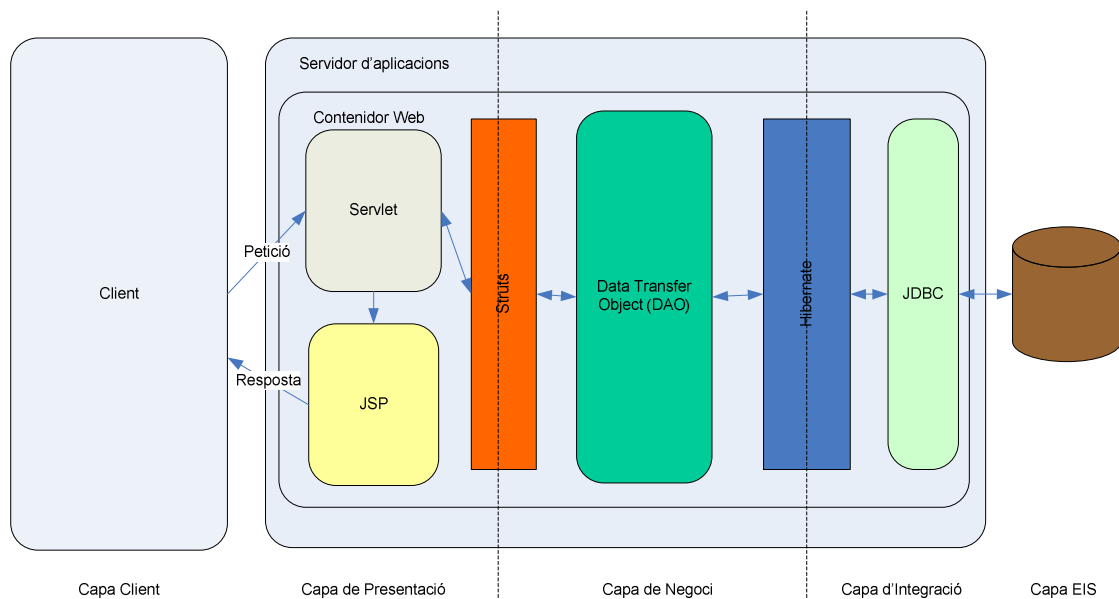
II-lustració 13. Model lògic relacional

## 2.3 Punt de vista computacional

El punt de vista computacional descriu la funcionalitat bàsica del sistema, així com la seva descomposició i organització en objectes que interactuen entre si per proporcionar aquesta funcionalitat.

La comunicació entre la capa de presentació i la capa de negoci es distribuïda i, fins i tot, la comunicació entre diferents components de la capa de negoci pot ser distribuïda.

- **Capa de presentació** amb servlets i JSP seguint un patró de disseny MVC. La capa de presentació es comunica amb la capa de negoci mitjançant el patró de disseny Struts.
- **Encapsulament** tots els accessos a la **font de les dades** mitjançant el patró Data Transfer Object (DAO)
- **Capa d'integració** amb Hibernate.



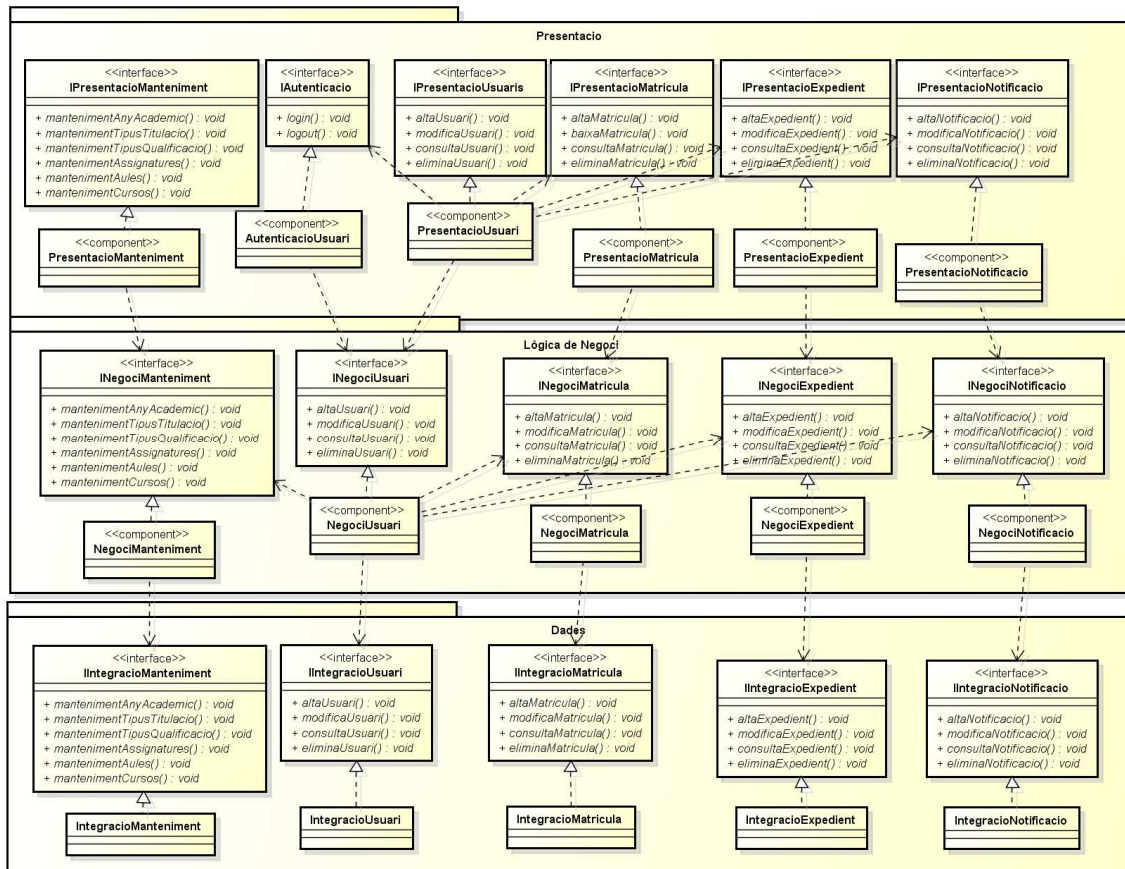
II-Il·lustració 14. Punt de vista computacional

### 2.3.1 Arquitectura programari detallada

Com el desenvolupament del projecte va ser una aplicació web, la millor solució és un estil arquitectònic heterogeni, model client-servidor i en tres capes:

- **Presentació:** És la capa on l'usuari pot interactuar amb el servei.
- **Lògica de Negoci:** Implementa les funcionalitats de l'aplicació
- **Integració:** Aquesta capa interactuarà amb les dades emmagatzemades en la base de dades.

A continuació mostrem l'arquitectura programari de l'aplicació:



II-lustració 15. Arquitectura programa detallada

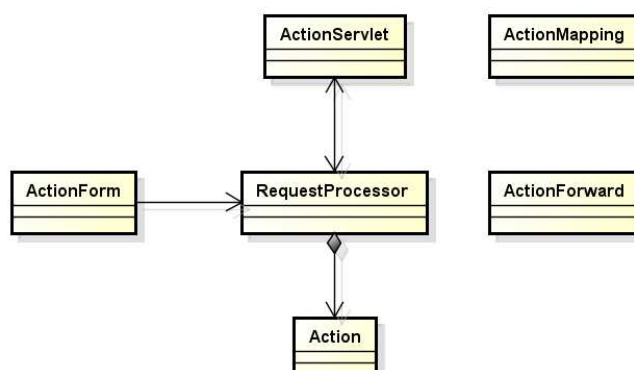
## 2.3.2 Component de presentació

Sol ser molt convenient utilitzar un patró de disseny MVC (model vista controlador) per als components de presentació. Si seguim aquest patró de disseny i suposem que el model estarà implementat en els components de negoci corresponents, podem dividir el component de presentació en el component que farà de controlador i el que farà de vista.

### 2.3.2.1 Patró de disseny Struts

Struts és una eina de suport pel desenvolupament d'aplicacions web sota el patró Model Vista Controlador (Model 2). Struts permet validar els formularis abans de que una acció sigui invocada. Si un formulari falla en la validació pot fer-se que Struts envii a l'usuari de nou a la vista (JSP) des de la que es va realitzar el Post en primer lloc, juntament amb un missatge d'error.

En la següent figurar es mostra les principals classes del controlador Struts:



Il·lustració 16. Principal classes del controlador Struts

La primera classe controladora és una *Servlet Java* anomenada *ActionServlet*. Aquesta classe manega totes les peticions dels usuaris. Utilitzant la informació de la configuració, la classe *ActionServlet* obté la classe apropiada *RequestProcessor* que conté la informació, que es part de la petició, i la posa en *ActionForm*, que és un *bean* on està la informació enviada per l'usuari mitjançant el formulari d'usuari. L'últim pas del controlador és delegar el control al manegador específic del tipus de resposta. Aquest manegador és sempre una subclasse de la classe *Action*.

### 2.3.2.2 Configuració de Struts en el projecte

S'ha de dir que l'eina Eclipse facilita molt la creació d'un projecte amb Struts (tal com es veurà en l'annex del document, capítol 8).

Com tot patró de disseny es minimitza el codi Java a escriure i es maximitza la configuració a través de fitxers XML

#### 2.3.2.2.1 Struts-config.xml

El patró de disseny Struts depèn d'un o més fitxers de configuració per poder crear i carregar l'aplicació. Un dels fitxers de configuració es *struts-config.xml*. En ell es defineixen quin són els formularis, quines accions s'han de realitzar, quines validacions s'han de fer i a quines pàgines hem d'anar depenent del tipus d'acció.

#### 2.3.2.2.2 Web.xml

Aquest fitxer de configuració conté els mapejos de URL per enviar les peticions d'arribada al *ActionServlet*. També es controla el temps de la sessió.

#### 2.3.2.2.3 ApplicationResources.properties

Tots els literals utilitzats en missatges en els JSP's, tant de dades d'aplicació com d'errors, són ubicats en el fitxer *ApplicacionResources.properties*. A més a més Struts proporciona etiquetes personalitzades en els JSP per accedir a aquests recursos. Encara que no s'ha tractat durant el projecte, Struts permet la internacionalització. Per cada idioma es podria utilitzar un fitxer (per exemple *ApplicacionResouces\_es.properties* contindria els missatges en castellà).

#### 2.3.2.2.4 Llibreria d'etiquetes

Per evitar incloure codi Java en els JSP's, el *framework* Struts proporciona un codi de 6 llibreries d'etiquetes que assisteixen en la tasca de la creació de la vista de MVC, com són els fitxers *Bean*, *HTML*, *Logic*, *Nested*, *Template*, i *Tiles Tags*.

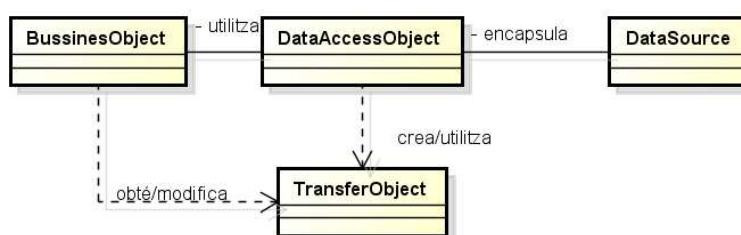


### 2.3.3 Component lògic de negoci

A la capa de lògica de negoci aplicarem el patró de disseny Data Access Object (DAO) que consisteix en encapsular tots els accessos a la font de les dades. DAO oculta completament els detalls d'implementació de la font de dades als seus clients.

Aquest patró consta de:

- **BusinessObject**, que és l'objecte que vol accedir a la font de dades per poder emmagatzemar o consultar dades
- **DataAccessObject** que abstruï al BusinessObject dels detalls de l'accés a la font de dades
- **DataSource** que representa la implementació de la font de dades
- **TransferObject** que és un objecte intermediari entre el BusinessObject i el DataAccessObject



Il·lustració 17. Principal classes del patró DAO

### 2.3.4 Component d'integració

Utilitzarem Hibernate en aquesta capa. És un *framework* de mapeig objecte/relacional per a implementar la persistència a la capa d'integració de les aplicacions J2EE. És un procés que automatitza el procés de creació, lectura, actualització i escriptura d'entitat a les bases de dades i ofereix una eina completa de mapeig objecte/relacional. Implementa d'una manera senzilla el patró de disseny DAO.

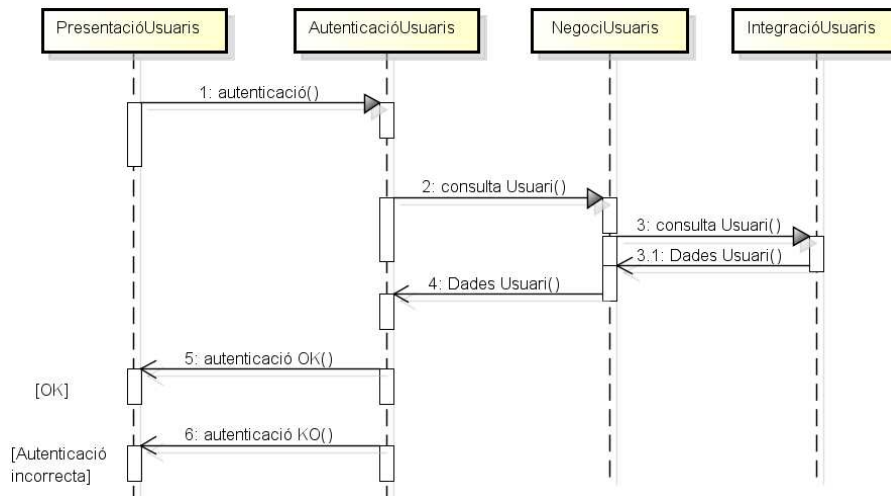
Podem dir que és el *framework* de mapeig de codi obert més madur i més complet que hi ha fins ara. Per realitzar la configuració utilitzarem el fitxer *hibernate.cfg.xml* on indicarem sobre quina base de dades ens connectarem i quins mapejos s'han de fer servir. Per cada taula crearem un fitxer amb extensió *hbm.xml* on es crearà la relació entre l'objecte relacional i l'objecte en Java.

### 2.3.5 Diagrama seqüència autenticació

Des del punt de vista de la computació, la descripció d'aquest comportament s'ha de fer d'una manera independent de la seva distribució, localització física, i de la tecnologia concreta que després s'usarà per a la seva implementació. És per això que en aquest punt de vista s'utilitza el diagrama de seqüència.

El diagrama de seqüència següent mostra les operacions dels components involucrats en el procés de accés al sistema d'un usuari.





II-lustració 18. Diagrama seqüència autenticació usuari

## 2.4 Punt de vista de l'enginyeria

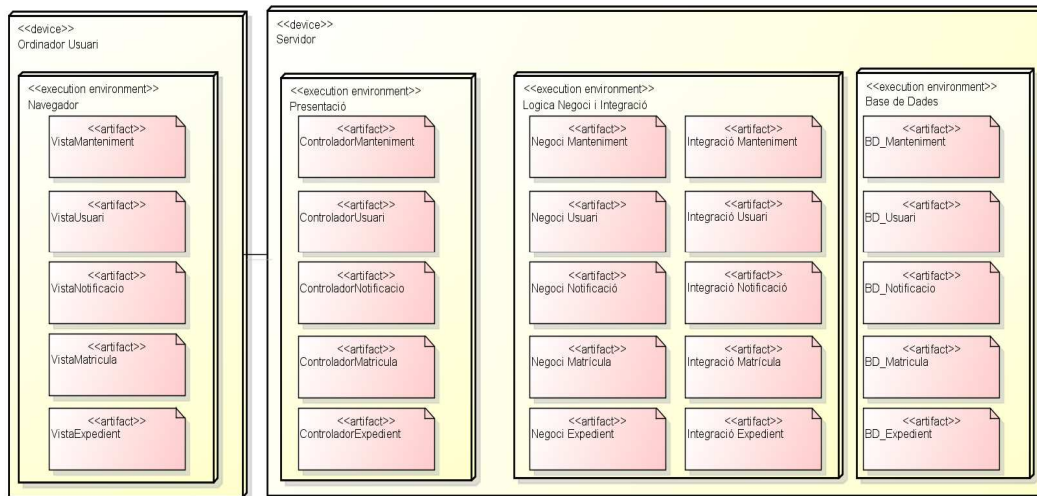
El punt de vista de l'enginyeria és l'encarregat d'especificar els aspectes de distribució, reproducció i localització dels objectes del punt de vista de la computació.

Els components són independents de la distribució en màquines físiques i de tecnologia que utilitzem per a la seva implementació.

Prendrem les següents consideracions:

- Els usuaris utilitzaran ordinadors, que accediran de forma remota a un servidor.
- L'accés des del servidor on resideixen la base de dades ha d'estar preparat perquè l'accés sigui local o remot (els gestor de la base de dades podria executar-se en una màquina diferent al servidor).
- Els components controladors de la capa de presentació s'executarà en el servidor i les vistes de presentació en l'ordinador de l'usuari.
- La capa de presentació s'executarà en el servidor i l'accés entre cada component sempre serà local.
- L'accés als components d'integració pels components de la lògica de negoci ha de ser local.

En la nomenclatura utilitzada <<device>> representa els ordinadors (els dels usuaris finals i el servidor). Els components s'executen en nodes estereotipats <<executionenvironment>>



II-lustració 19. Punt de vista d'enginyeria

## 2.5 Punt de vista de la tecnologia

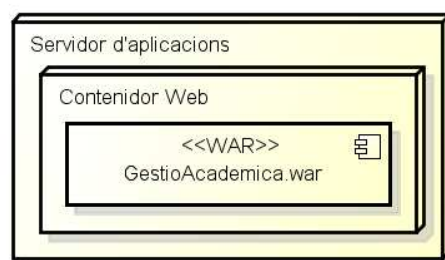
El sistema funcionarà amb les següents característiques tecnològiques:

- L'usuari final podrà utilitzar qualsevol tipus d'ordinador.
- Les comunicacions entre el client i el servidor es farà per internet.
- L'usuari final podrà interactuar amb el programari mitjançant un navegador (qualsevol dels disponibles en el mercat).
- La resolució de pantalla mínima ha de ser 1024x860.

Utilitzarem MySQL com a motor gestor de base de dades relacional, però és senzill canviar a d'altres com a Informix o Oracle (canviant el fitxer *hibernate.config.xml* i instal·lant el nou connector JDBC de la base dades en el servidor d'aplicacions).

## 2.6 Implementació del sistema

L'aplicació s'empaquetarà en un fitxer .ear anomenat *GestioAcademica.ear*, que conté la capa de presentació en una aplicació web denominada *GestioAcademica.war*. Aquest és el diagrama de desplegament de l'aplicació:



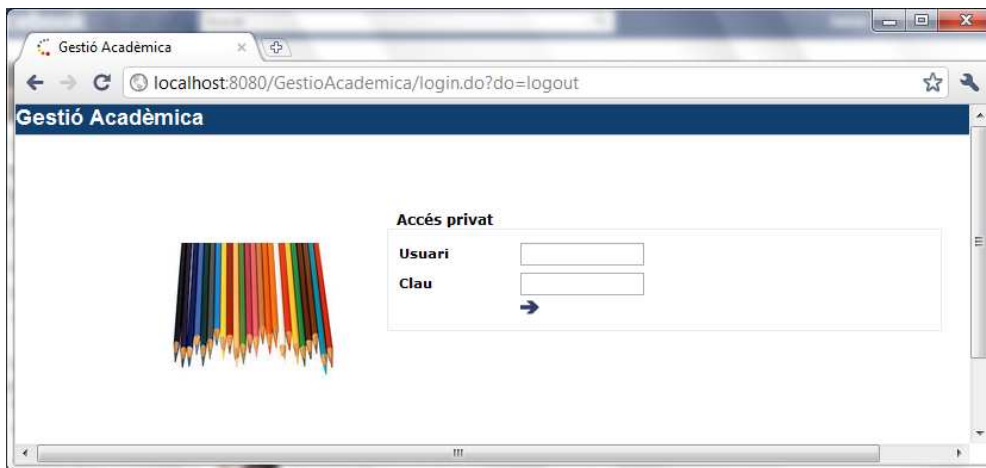
II-lustració 20. Diagrama de desplegament de l'aplicació

## 3 Disseny de la interfície gràfica d'usuari

En aquest punt exposarem els efectes del funcionament del programari visibles per l'usuari. La coherència entre la funcionalitat i la interfície d'usuari hauria d'estar garantida.

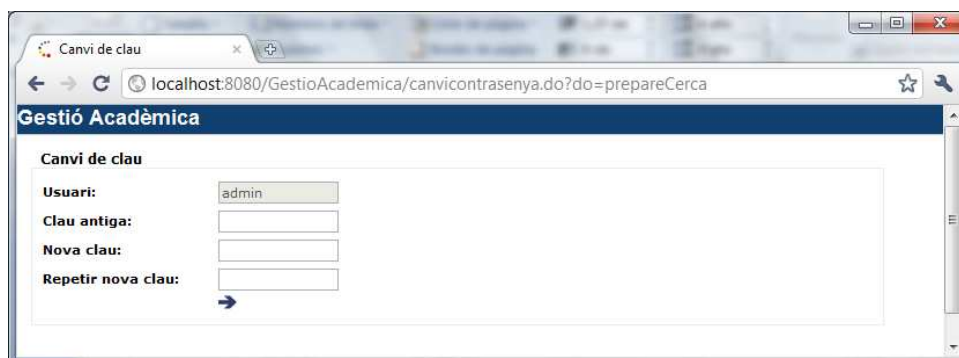
## 3.1 Autenticació de l'usuari

### 3.1.1 Entrada al sistema



Il·lustració 21. Entrada al sistema

### 3.1.2 Modificació de contrasenya



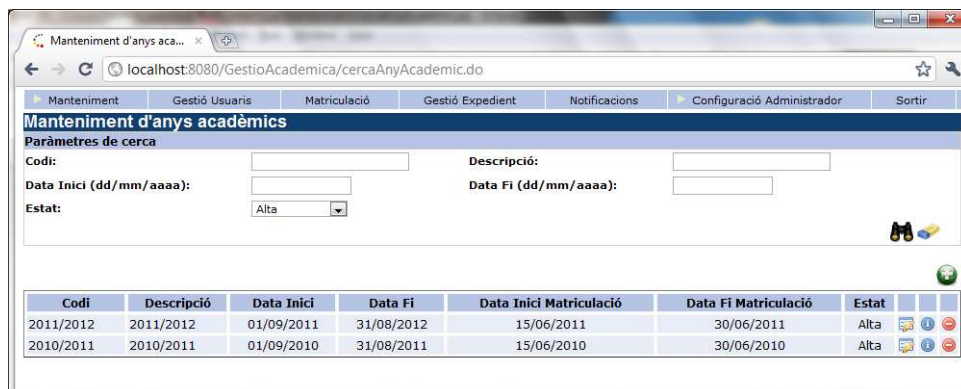
Il·lustració 22. Canvi de paraula clau

Recordem que la contrasenya estarà encryptada mitjançant l'algorisme MD5.

## 3.2 Subsistema de Manteniment

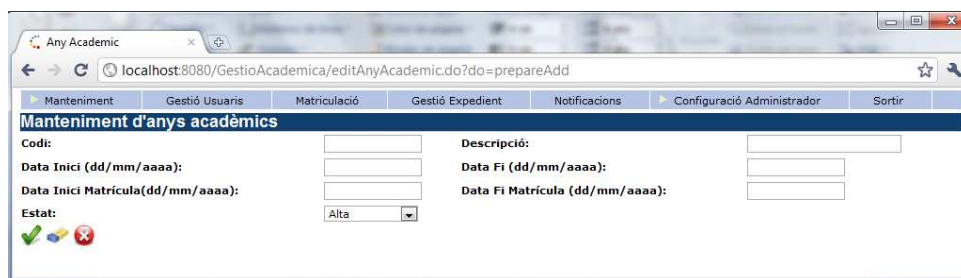
### 3.2.1 Manteniment d'any acadèmic

#### 3.2.1.1 Cerca any acadèmic



Il·lustració 23. Cerca any acadèmic

#### 3.2.1.2 Alta/elimina/modifica any acadèmic

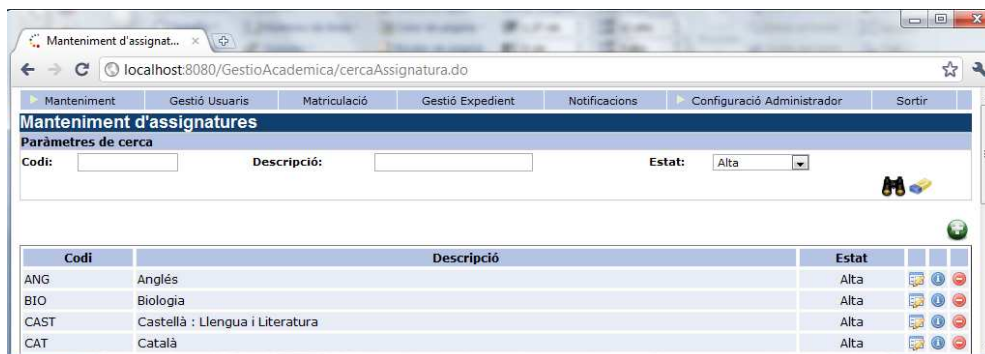


Il·lustració 24. Manteniment any acadèmic

Els anys acadèmics no es podran solapar en el període de vigència.

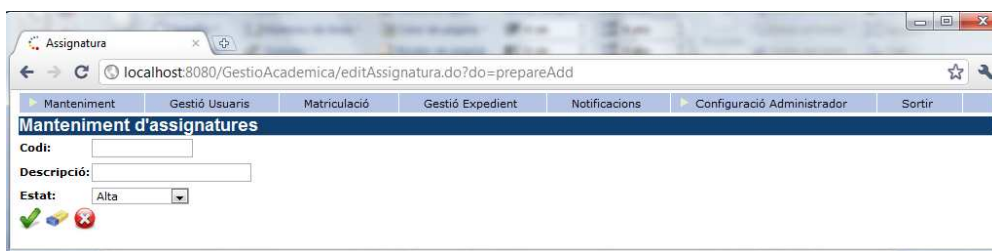
### 3.2.2 Manteniment d'assignatures

#### 3.2.2.1 Cerca assignatures



Il·lustració 25. Cerca assignatura

### 3.2.2.2 Alta/Elimina/Modifica assignatures



II-lustració 26. Manteniment assignatura

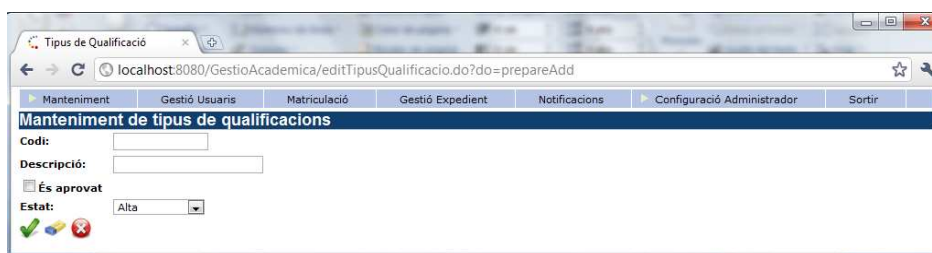
### 3.2.3 Manteniment de tipus de qualificacions

#### 3.2.3.1 Cerca tipus de qualificacions



II-lustració 27. Cerca tipus de qualificació

#### 3.2.3.2 Alta/Elimina/Modifica tipus de qualificacions

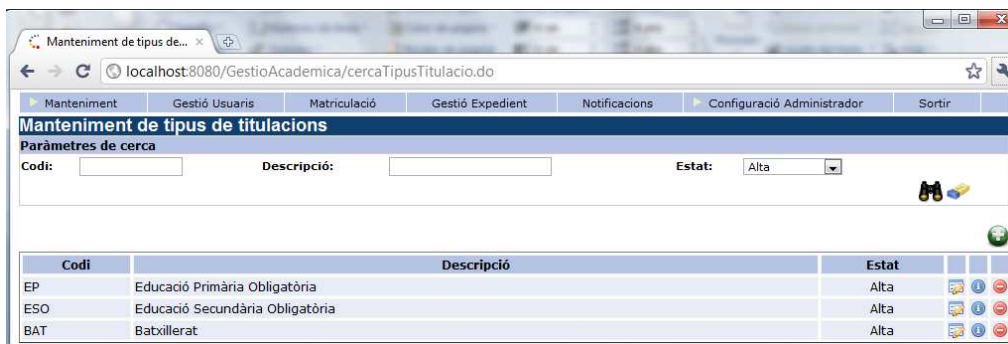


II-lustració 28. Manteniment tipus de qualificació

La marca 'Aprovat' ens indicarà quines qualificacions fan que l'estudiant aprovi l'assignatura.

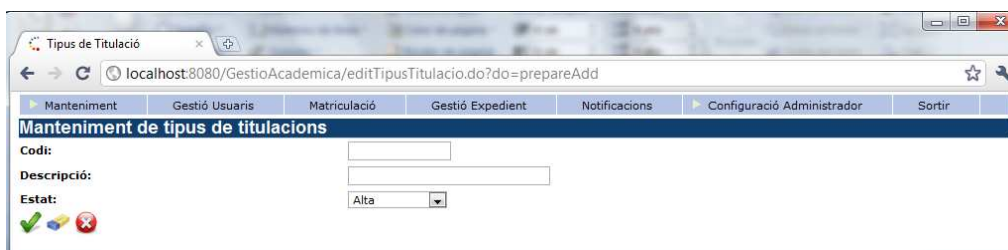
## 3.2.4 Manteniment de tipus de titulació

### 3.2.4.1 Cerca tipus de titulacions



II-lustració 29. Cerca tipus de titulació

### 3.2.4.2 Alta/Elimina/Modifica tipus de titulacions



II-lustració 30. Manteniment tipus de titulació

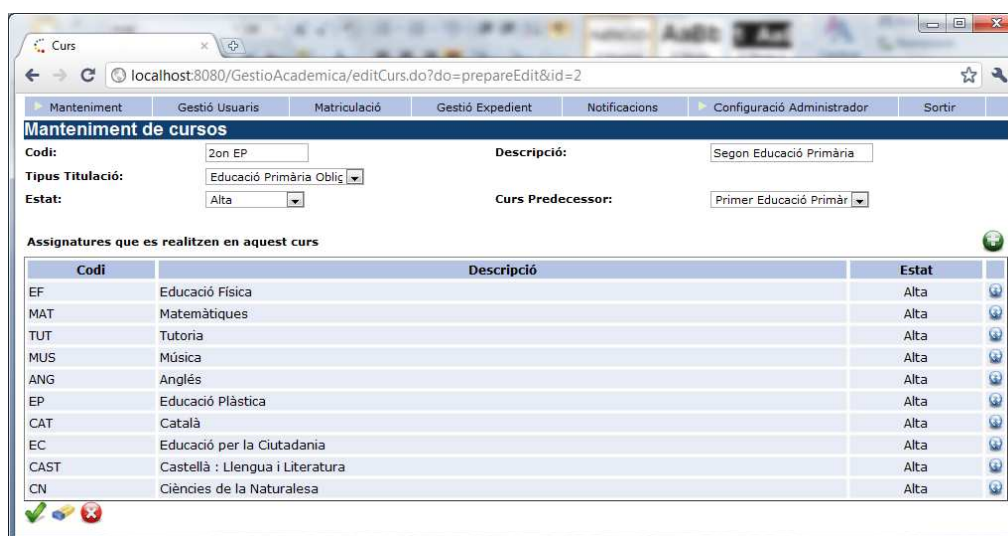
## 3.2.5 Manteniment de cursos

### 3.2.5.1 Cerca cursos



II-lustració 31. Cerca de curs

### 3.2.5.2 Alta/Elimina/Modifica cursos



II-lustració 32. Manteniment de curs

Des d'aquesta pantalla hem de relacionar cada curs amb el tipus de titulació. A més a més, indicarem totes les assignatures que s'imparteixen en el curs. Per cada curs hem de dir quin és el seu curs predecessor.

### 3.2.6 Manteniment d'aules

#### 3.2.6.1 Cerca aules



II-lustració 33. Cerca d'aula



### 3.2.6.2 Alta/Elimina/Modifica aules

Codi	Descripció	Professor
MUS	Música	Acebes Ramos Pablo
MAT	Matemàtiques	Acebes Ramos Pablo
ANG	Anglès	Acebes Ramos Pablo
EP	Educació Plàstica	Acebes Ramos Pablo
CAST	Castellà : Llengua i Literatura	Acebes Ramos Pablo
EF	Educació Física	Acebes Ramos Pablo
CAT	Català	Acebes Ramos Pablo

II-lustració 34. Manteniment d'aula

Cada aula la relacionarem amb un curs. A més a més, haurem d'indicar quins professors imparteixen les diferents assignatures de cada curs.

## 3.3 Subsistema d'usuari

### 3.3.1 Cerca usuari

El disseny de la cerca dels diferents tipus d'usuari serà igual tant per personal administratiu, professors, alumnes i pares.

Codi	Primer Cognom	Segon Cognom	Nom	Num.Document	Tipus Usuari	Estat
4	Garcia	Puertas	Noelia	40999932V	Alumne	Alta
6	Lamine		Mohamed	X1234567G	Alumne	Alta

II-lustració 35. Cerca usuari

El professor només podrà visualitzar els alumnes que estan en les assignatures que ell imparteix. L'alumne només podrà visualitzar la seva fitxa personal. Els pares/tutors legals només podran cercar els seus fills (i la seva pròpia fitxa).



### 3.3.2 Mòdul de manteniment del personal administratiu

#### 3.3.2.1 Alta/Elimina/Modifica personal administratiu

The screenshot shows a web browser window with the URL `localhost:8080/GestioAcademica/editUsuari.do?do=prepareEdit&id=2`. The page title is 'Manteniment d'usuaris'. The form contains the following fields:

- Num.Usuari:** 2
- Login:** Garc\_Lau
- Reiniciar contrassenya:**
- Primer Cognom:** Garcia
- Segon Cognom:** Perez
- Nom:** Laura
- Tipus Document:** DNI / NIF
- Num. Document:** 11111111H
- Tipus Usuari:** Personal Administratiu
- Adreça:** Xiprer 5
- CP:** 8940
- Població:** Cornellà
- Província:** Barcelona
- Correu electrònic:** laura.garcia@gmail.com
- Telèfon:** 933776543
- Estat:** Alta
- Data Naixement (dd/mm/aaaa):** 01/02/1980

II-lustració 36. Manteniment personal administratiu

El codi d'usuari es crearà automàticament al crear un usuari. Aquest codi no es podrà modificar. El personal administratiu podrà reiniciar la contrasenya als usuaris per si s'obliden d'aquesta i poder tornar a entrar en el sistema.

### 3.3.3 Mòdul de manteniment de professors

#### 3.3.3.1 Alta/Elimina/Modifica professor

The screenshot shows a web browser window with the URL `localhost:8080/GestioAcademica/editUsuari.do?do=prepareEdit&id=3`. The page title is 'Manteniment d'usuaris'. The form contains the following fields:

- Num.Usuari:** 3
- Login:** Aceb\_Pab
- Reiniciar contrassenya:**
- Primer Cognom:** Acebes
- Segon Cognom:** Ramos
- Nom:** Pablo
- Tipus Document:** DNI / NIF
- Num. Document:** 52625086M
- Tipus Usuari:** Professor
- Adreça:** Ganduxer 141
- CP:** 8080
- Població:** Barcelona
- Província:** Barcelona
- Correu electrònic:** pablo.acebes@uoc.edu
- Telèfon:** 935553245
- Estat:** Alta
- Data Naixement (dd/mm/aaaa):** 01/03/1977

Below the form, there is a table titled 'Assignatures pot exercir':

Codi	Descripció	Estat
CAST	Castellà : Llengua i Literatura	Alta
DIB	Dibuix Tècnic	Alta
EF	Educació Física	Alta
FI	Filosofia	Alta
TUT	Tutoria	Alta
HIST	Història	Alta
EC	Educació per la Ciutadania	Alta

II-lustració 37. Manteniment professor

Per cada professor s'haurà d'indicar les assignatures que pot impartir. Aquest manteniment només el pot mantindre el personal administratiu.

### 3.3.4 Mòdul de manteniment d'alumnes

#### 3.3.4.1 Alta/Elimina/Modifica alumne

The screenshot shows a web browser window with the URL `localhost:8080/GestioAcademica/editUsuari.do?do=prepareEdit&id=4`. The page title is 'Manteniment d'usuaris'. The form contains the following fields:

- Num.Usuari:** 4
- Primer Cognom:** Garcia
- Tipus Document:** DNI / NIF
- Adreça:** Rambla Prim 33
- Província:** Barcelona
- Estat:** Alta
- Dades bancàries:** (empty fields)
- Login:** Garc\_Noee
- Segon Cognom:** Puertas
- Num. Document:** 40999932V
- CP:** 8020
- Correu electrònic:** noelia@yahoo.es
- Data Naixement (dd/mm/aaaa):** 23/11/2005
- Reiniciar contrassenya:** (checkbox)
- Nom:** Noelia
- Tipus Usuari:** Alumne
- Població:** Barcelona
- Telèfon:** 669345678
- Curs Primera Matricula:** Primer Educació Primàr

II-lustració 38. Manteniment alumne

A l'alumne se li podrà associar les dades bancàries. En un futur es podria treballar amb aquestes dades si es desenvolupés el mòdul de facturació.

### 3.3.5 Mòdul de manteniment de pares

#### 3.3.5.1 Alta/Elimina/Modifica pare/tutor

The screenshot shows a web browser window with the URL `localhost:8080/GestioAcademica/editUsuari.do?do=prepareEdit&id=5`. The page title is 'Manteniment d'usuaris'. The form contains the following fields:

- Num.Usuari:** 5
- Primer Cognom:** Garcia
- Tipus Document:** DNI / NIF
- Adreça:** Camí Llarg 3
- Província:** Barcelona
- Estat:** Alta
- Fill:** (checkbox)
- Login:** Garc\_Mar
- Segon Cognom:** Navas
- Num. Document:** 52625087Y
- CP:** 8050
- Correu electrònic:** mgarcia@hotmail.com
- Data Naixement (dd/mm/aaaa):** 06/05/1975
- Reiniciar contrassenya:** (checkbox)
- Nom:** Maria
- Tipus Usuari:** Pare/Tutor
- Població:** Esplugues
- Telèfon:** 938763456

Below the form, there is a table titled 'Fills':

Codi	Nom	Estat
4	Garcia Puertas Noelia	Alta

II-lustració 39. Manteniment pare/tutor legal

Per cada pare donat d'alta s'haurà de relacionar amb els fills que té en l'Escola. Aquesta relació només la podrà fer el personal administratiu.

### 3.3.6 Actualització de foto



Il·lustració 40. Actualització foto

Cada usuari es podrà actualitzar la seva foto de perfil.

## 3.4 Subsistema de notificaciones

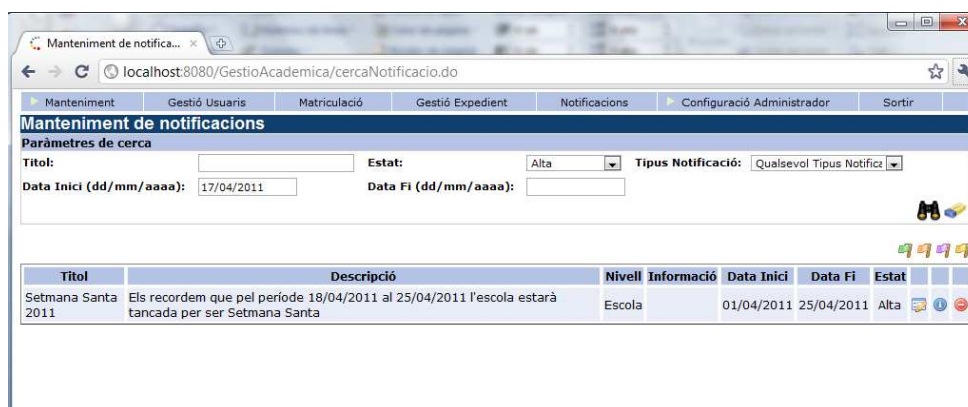
### 3.4.1 Avís notificaciones en menú principal



Il·lustració 41. Menú principal

A l'entrar el sistema ens indicarà si existeixen notificaciones i avisos vigents.

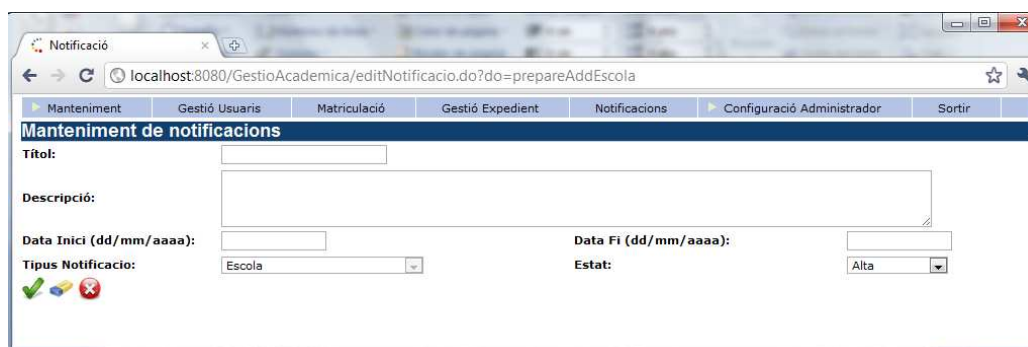
### 3.4.2 Cerca notificació



II-lustració 42. Cerca de notificació

El personal administratiu podrà veure (i filtrar) per nivell Escola/Curs/Assignatura/Aula. Els professors només podran veure els avisos a nivell d'Escola o Assignatura que ell imparteix. Els alumnes i pares/tutors podran cercar els avisos a nivell d'Escola i a nivell de Curs/Assignatura/Aula si està en ell matriculat.

### 3.4.3 Alta/Elimina/Modifica notificació



II-lustració 43. Manteniment de notificació

## 3.5 Subsistema de matriculació

### 3.5.1 Cerca matriculació

Codi	Nom	Num.Document	Estat	Matriculat Any Acadèmic Actual	Estat Matrícula
60	Abad Casquero Lluís	11111156V	Alta	Si	Alta
61	Abad Casquero Manel	11111157H	Alta	Si	Alta
62	Abad Casquero Marc	11111158L	Alta	Si	Alta

II-lustració 44. Cerca de matriculació

Des del cercador de matrícules podrem veure si l'alumne està matriculat en l'any acadèmic vigent. Si no és així es podrà realitzar la matriculació. El sistema ens indica si el període de matriculació és obert o tancat.

### 3.5.2 Alta/Elimina/Modifica matriculació

II-lustració 45. Manteniment de matriculació

Per realitzar una nova matrícula cercarem qualsevol matrícula antiga i des d'aquí polsarem sobre el botó 'Alta Matrícula'. Recordar que si l'alumne no està assignat a una d'aula no se li podrà qualificar. El procés d'assignació d'un alumne a una aula el podrà realitzar directament el personal administratiu des de la modificació de la matrícula o bé executar el procés d'assignació automàtica d'alumnes a les aules.

## 3.6 Subsistema de gestió de l'expedient

### 3.6.1 Cerca qualificació

The screenshot shows the 'Gestió de l'expedient' web application. The search parameters are as follows:

- Codi: 60
- Tipus Document: DNI / NIF
- Any Acadèmic: 2010/2011
- Primer Cognom: (empty)
- Segon Cognom: (empty)
- Num. Document: (empty)
- Estat: Alta
- Aula: Qualsevol Aula
- Assignatura: Qualsevol Assignatura

The results table is as follows:

Codi	Nom	Titulació	Curs	Aula	Assignatura	Estat	Qualif.1	Qualif.2	Qualif.3	Repesca 1	Repesca 2	Qualif.Final
60	Abad Casquero Lluís	Educació Primària Obligatòria	Cinquè Primària	Cinquè A Educació Primària	Anglès	Alta	Excel.lent	Excel.lent	Notable			Excel.lent
60	Abad Casquero Lluís	Educació Primària Obligatòria	Cinquè Primària	Cinquè A Educació Primària	Castellà : Llengua i Literatura	Alta	Bé	Excel.lent	Notable			Notable
60	Abad Casquero Lluís	Educació Primària Obligatòria	Cinquè Primària	Cinquè A Educació Primària	Ciències de la Naturalesa	Alta	Notable	Excel.lent	Notable			Notable
60	Abad Casquero Lluís	Educació Primària Obligatòria	Cinquè Primària	Cinquè A Educació Primària	Ciències MediAmbientals	Alta	Excel.lent	Excel.lent	Notable			Excel.lent

II-lustració 46. Cerca expedient acadèmic

### 3.6.2 Alta/Modifica qualificació

The screenshot shows the 'Qualificació' web application. The maintenance form is as follows:

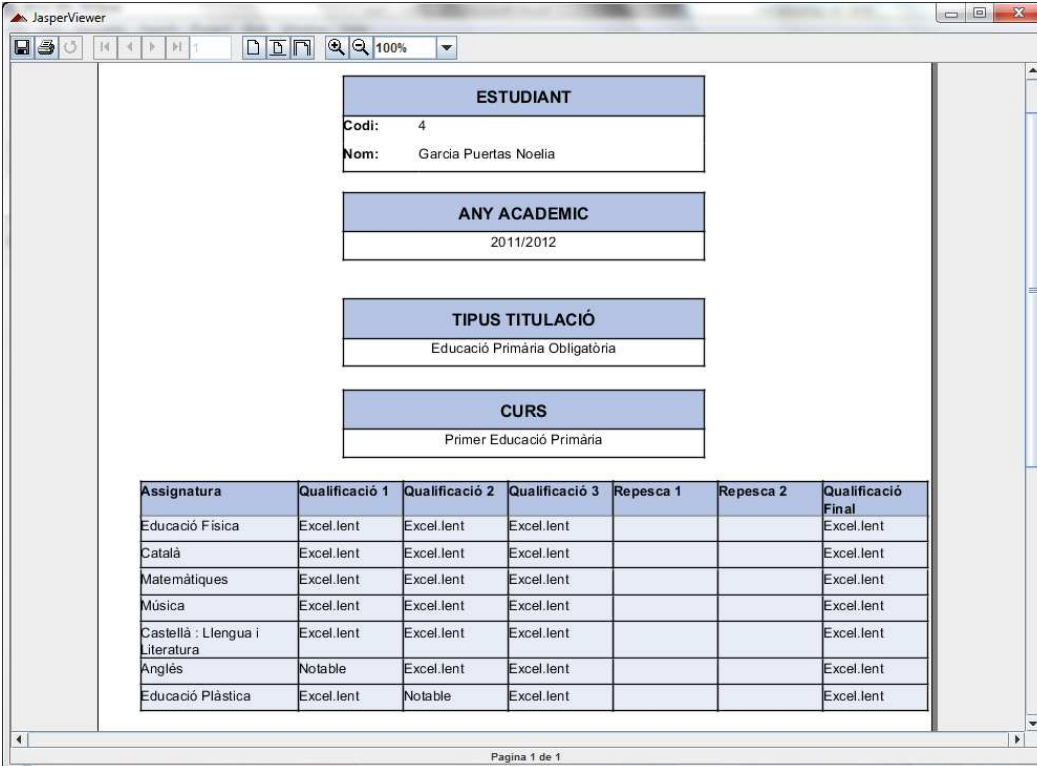
- Codi: 4
- Any acadèmic: 2011/2012
- Primer Cognom: Noelia
- Segon Cognom: Garcia
- Nom: Puertas
- Aula: Primer A Educació Prim
- Assignatura: Anglès
- Qualificació 1: Notable
- Qualificació 2: Excel.lent
- Qualificació 3: Excel.lent
- Repesca 1: Sense qualificar
- Repesca 2: Sense qualificar
- Qualificació Final: Excel.lent

II-lustració 47. Manteniment expedient acadèmic

Aquest manteniment només el podrà mantenir el professor que imparteix l'assignatura.



### 3.6.3 Llistat expedient acadèmic



ESTUDIANT						
Codi:	4					
Nom:	Garcia Puertas Noelia					
ANY ACADEMIC						
2011/2012						
TIPUS TITULACIÓ						
Educació Primària Obligatòria						
CURS						
Primer Educació Primària						
Assignatura	Qualificació 1	Qualificació 2	Qualificació 3	Repesca 1	Repesca 2	Qualificació Final
Educació Física	Excel.lent	Excel.lent	Excel.lent			Excel.lent
Català	Excel.lent	Excel.lent	Excel.lent			Excel.lent
Matemàtiques	Excel.lent	Excel.lent	Excel.lent			Excel.lent
Música	Excel.lent	Excel.lent	Excel.lent			Excel.lent
Castellà : Llengua i Literatura	Excel.lent	Excel.lent	Excel.lent			Excel.lent
Anglès	Notable	Excel.lent	Excel.lent			Excel.lent
Educació Plàstica	Excel.lent	Notable	Excel.lent			Excel.lent

Il·lustració 48. Llistat expedient acadèmic

Des del cercador es podrà imprimir l'expedient acadèmic de l'alumne.

### 3.7 Llegenda d'ícones

A continuació s'exposa la llegenda d'ícones que s'ha utilitzat en el desenvolupament de tot el projecte.

Icona	Descripció	Icona	Descripció
	Afegir		Crear alumne
	Cancel·lar		Crear pare/tutor legal
	Consultar		Crear notificació escola
	Modificar		Crear notificació curs
	Esborrar		Crear notificació aula
	Cercar		Crear notificació assignatura
	Netejar		Accedir al sistema

	Actualitzar		Imprimir expedient
	Crear personal administratiu		Actualitzar estat alta/baixa
	Crear professor		

## 4 Valoració econòmica

Mitjançant aquest projecte hem pogut desenvolupar aplicacions professionals amb eines de codi obert i llicència pública general, per tant no s'ha realitzat cap valoració econòmica.

## 5 Conclusions i línies de futur

### 5.1 Conclusions

Un cop realitzat i verificat el correcte funcionament del treball fi de carrera podem extreure diferents conclusions.

Ha estat una **bona decisió** el haver desenvolupat el projecte utilitzant el cicle de vida clàssic. Això ha permès estalviar molt de temps en l'etapa de codificació, ja que mitjançant l'etapa d'anàlisi i disseny va fer que molta de la feina estigués pensada, evitant divagacions i improvisacions en l'etapa de programació.

S'ha pogut veure tot el **potencial** que hi ha darrera de **J2EE**, i Java com a llenguatge de programació, per obtenir molts beneficis amb el seu ús. Dos dels majors avantatges és que és de codi obert permetent tenir solucions lliures, i la segona, que en el cas de desenvolupar el projecte ha estat de gran ajuda el que hi hagi darrera una àmplia comunitat de desenvolupadors per poder resoldre els problemes que varen aparèixer.

S'ha pogut verificar que utilitzar el patró d'arquitectura de programari **MVC**, Model Vista Controlador, ajuda a **millorar el desenvolupament** i manteniment, ja que separa la capa d'interfície d'usuari de la lògica de negoci i de les dades. Sota aquest patró MVC està l'eina de suport **Struts**, que ha permès reduir el temps de desenvolupament. Fent un estudi previ de l'eina s'ha pogut comprovar que és una eina que s'utilitza àmpliament en el món real, i té una gran solidesa.

En la capa de dades, gràcies a **Hibernate**, ha ajudat a modelar la base de dades i configurar les relacions existents en el model de dades. S'ha verificat que és molt **flexible** en la seva configuració, i ha permès una facilitat en la programació ja que la orientació a objectes facilita moltíssim el passar d'un diagrama a codi. Amb HQL, s'ha permès realitzar els criteris de cerca fàcilment treballant amb objectes facilitant així l'ús de l'herència, polimorfisme i associació.

Una avantatge més en l'ús d'**Hibernate**, és la total **independència** amb el motor de **base de dades** ja que tenim totalment independent la capa de dades amb la capa de negoci. Si en un futur volguéssim canviar de base de dades, bastaria amb canviar una línia del fitxer de connexió i ja tindríem l'aplicació funcionant sota un altre motor de base de dades.

Altres aspectes rellevants en el desenvolupament és haver après nocions de **CSS**. La fulla d'estil en cascada, ha ajudat molt a **millorar la presentació**. Mitjançant un arxiu, on s'emmagatzema les propietats del HTML, ha fet que les regles de configuració per la pàgina HTML i l'estructura bàsica de la pàgina estiguessin totalment separades.

A títol personal he de dir que crec que he encertat en escollir l'**àrea J2EE** ja que m'ha servit tenir una visió global d'un projecte informàtic i he posat en pràctica molts dels coneixements adquirits al llarg de la carrera. Encara que la corba d'aprenentatge ha estat molt alta, i he dedicat moltes hores en la investigació i en el desenvolupament del treball per tenir un resultat òptim del projecte, la sensació ha estat que m'he divertit i que m'ha **enriquit molt**.



## 5.2 Línies de futur

Per al futur es podria fer una sèrie d'extensibilitats en el projecte.

### 5.2.1 Mòdul de facturació

Encara que inicialment no estava previst, es podria desenvolupar el mòdul de facturació. Des d'aquest mòdul l'administratiu podria mantenir els diferents tipus de conceptes (xandall, excursió, piscina,...) i poder generar els rebuts, associats a cada alumne matriculat en el centre. És per aquesta raó que en la fitxa de l'alumnat es pot introduir les dades bancàries associades.

### 5.2.2 Versió en diferents idiomes

La aplicació web només està en català, però es podria treballar en el multi idioma de l'aplicació, de tal manera, que escollint un idioma, el text de les pàgines aparegués traduït al idioma seleccionat.

### 5.2.3 Configuració número d'avaluacions

Un alumne tindrà 3 avaluacions al llarg de l'any acadèmic (i dues possibles repesques). En properes versions s'hauria de permetre configurar el número d'avaluacions.

### 5.2.4 Configuració notes exàmens

Una possible millora és que el professor pogués anar publicant les notes dels exàmens a part de les notes finals de cada avaluació.

### 5.2.5 Calendari

Poder configurar el calendari, de tal manera que en el moment un alumne es matriculés en un nou any acadèmic i fos assignat a una aula es pogués imprimir el calendari escolar.

## 6 Glossari

**Administratiu:** Persona encarregada de gestionar els tràmits que es realitzen en el centre docent. S'encarregarà del manteniment del programari perquè es puguin realitzar les gestions acadèmiques i matriculacions dintre del centre.

**Alumne:** Persona que cursa estudis en el centre.

**Any acadèmic:** Període lectiu on es realitza el curs escolar. Comprèn el curs escolar, les vacances i les festes.

**Assignatura:** Lliçons que imparteix un professor i que l'estudiant ha de cursar obligatòriament per obtenir el títol.

**Aula:** Cadascuna de les sales destinades a fer-hi classes.

**Casos d'ús:** Tècnica per la captura de requisits potencials d'un sistema o actualització de programari

**Classe:** es una construcció que s'utilitza com un model (o plantilla) per crear objecte d'aquest tipus.

**CSS:** Fulles d'estil en cascada. És un llenguatge utilitzat per definir la presentació d'un document estructurat escrit en HTML o XML.

**Curs:** El temps assenyalat dins de cada any acadèmic per a assistir a les lliçons.

**Desplegament:** Publicació d'un programari.

**Diagrama Entitat Relació:** Eina pel modelatge de dades d'un sistema de la informació.

**Diagrama de Gantt:** Popular eina gràfica per mostrar el temps de dedicació previst per diferents tasques o activitats al llarg d'un temps total determinat.

**Ear:** *An Enterprise Archive*, és un fitxer utilitzats en Java EE per empaquetar un o més mòduls en el desplegament d'un programari en un servidor d'aplicacions.

**Eclipse:** És un entorn de desenvolupament integrat (IDE) de codi obert. Aquest entorn facilita molt la creació de projectes informàtics desenvolupats amb Java.

**Escola:** Lloc, edifici, etc, on es desenvolupa l'activitat escolar i docent. Estarà formada per un conjunt d'alumnes i professors.

**Estudiant:** Veure *alumne*

**FrameWork:** Estructura conceptual i tecnològica de suport definit, normalment amb artefactes o mòduls de programari concrets, i per la qual altre projecte de programari pot ser organitzat i desenvolupat.

**Hibernate:** És una eina de mapeig objecte relacional (ORM) per la plataforma Java que facilita el mapeig d'atributs entre una base de dades relacional tradicional i el model d'objectes d'una aplicació.

**HTML:** Llenguatge de Marcat d'Hipertext, es el llenguatge predominat per l'elaboració de pàgines web.

**HQL:** *Hibernate Query Language*, llenguatge de consulta de dades que utilitza Hibernate.

**IDE:** Entorn de desenvolupament integrat.

**iReport:** Eina de programari lliure que permet dissenyar informes de JasperReport escrit en Java. Podem crear diferents llistats que existeixin en les diferents opcions del portal.

**Instància:** Es produeix amb la creació d'un objecte que pertany a una classe (es diu s'instancia la classe). L'objecte que es crea té els atributs, propietats i mètodes de la classe a la que pertany.

**J2EE:** *Java Platform, Enterprise Edition* o *Java EE* (anteriorment conegut com *Java 2 Platform, Enterprise Edition* o *J2EE* fins a la versió 1.4), és una plataforma de programació per desenvolupar i executar programari d'aplicacions en llenguatge Java amb arquitectura de N capes distribuïdes.

**Jar:** Tipus d'arxiu que permet executar aplicacions escrites en llenguatge Java.

**JasperReport:** Eina de creació d'informes que te l'habilitat crear contingut enriquit al monitor, a la impressora o a fitxers PDF, HTML, XLS, CSV i XML.

**Javadoc:** Utilitat per la generació de documentació format HTML partir de codi font Java

**JSP:** es una tecnologia Java que permet generar contingut dinàmic per la web, en forma de documents HTML, XML o d'altres tipus.

**JBoss:** Servidor d'aplicacions de codi obert . Està totalment implementat en Java pur. Ofereix una plataforma d'alt rendiment per aplicacions de negoci electrònic.

**Matrícula:** Procés on s'inscriu un alumne en un any acadèmic dintre d'un centre docent.

**MD5:** *Message-Digest Algorithm 5*, és un algorisme de reducció criptogràfic de 128 bits. És una mesura eficient per encriptar les contrasenyes.

**MySQL:** És una sistema de gestió de base de dades. En aplicacions de baixa concurrència en la modificació de dades és ideal. Aquesta base de dades està sota la llicència pública general de GNU.

**MVC:** és un patró d'arquitectura de programari que separa les dades d'una aplicació, la presentació d'usuari i la lògica de negoci en tres components diferents..

**Notes:** Veure *qualificacions*.

**Objecte:** Es defineix com la instància d'una classe.

**ORM:** Mapeig objecte-relacional es una tècnica de programació per convertir dades entre el sistema de tipus utilitzat en un llenguatge orientat a objecte i l'utilitza en una base de dades relacional, utilitzant un motor de persistència.

**Package:** Contenedor de classes que permet agrupar les diferents parts d'un programa on la seva funcionalitat té elements comuns.

**Pare, mare:** Les persones que han engendrat i que és el responsable legal dels fills.

**Patró de disseny:** És una eina de suport pel desenvolupament d'eines Web. Està sota el patró MVC (Model Vista Controlador) .

**Personal administratiu:** Veure *Administratiu*.

**Professor:** Persona que es dedica a l'ensenyament d'un art, una ciència o una llengua, etc. Serà l'encarregat de la introducció de les qualificacions a cada alumne.

**Qualificacions:** Nota que s'utilitza per valorar una assignatura.

**RM-ODP:** Model de referència per al processament obert i distribuït

**Script:** Arxiu de processament per lots.

**Servlet:** Programa que s'executa en un servidor i el seu ús és generar pàgines web de forma dinàmica a partir dels paràmetres de la petició que s'envia el navegador web.

**Struts:** Eina de suport pel desenvolupament d'aplicacions Web sota el patró MVC sota la plataforma Java EE.

**Tutor:** Persona que encara que no és el pare de l'alumne és el responsable legal.

**Usuari:** Persona que farà servir el programari a desenvolupar.

**UML:** Llenguatge Unificat de Modelatge. És el llenguatge de modelatge d'un sistema de programari

**War:** És un arxiu Jar utilitzat per distribuir una col·lecció de *JavaServer Pages (JSP)*, servlets, classes, Java, arxius XML, llibreries d'etiquetes i pàgines web estàtiques (HTML i arxius relacionats) que junts constitueixen una aplicació web.

## 7 Bibliografia.

### 7.1 Struts

Mike Robinson i Ellen Finkelstein (2004). **Jakarta Struts For Dummies®**. Wiley Publishing, Inc.

Chuck Cavaness (2004). **Programming Jakarta Struts, 2nd Edition**. O'Reilly

Chuck Cavaness (2002). **Jakarta Struts**. O'Reilly

Laliluna Java Tutorials and Development <http://www.laliluna.de/>

Rose India <http://www.roseindia.net/>

### 7.2 Hibernate

Dave Minter, Jeff Linwood (2010). **Beginning Hibernate**. Apress.

Christian Bauer i Gavin King (2007). **Java Persistence with Hibernate**. Manning Publications Co.

Christian Bauer i Gavin King (2005). **Hibernate in action**. Manning Publications Co.

Tim Downey (2007). **Web Development with Java, Using Hibernate, JSPs and Servlets**. Springer-Verlag London.

James Elliott, Timothy M. O'Brien, Ryan Fowler (2008). **Harnessing Hibernate**. O'Reilly.

**JBoss Community Hibernate** <http://www.hibernate.org/>

**Introducció a Hibernate** <http://www.jtech.ua.es/j2ee/2006-2007/doc/sesion03-apuntes.pdf>

### 7.3 Jboss

*Sam Griffith, Norman Richards* (2005). **JBoss: A Developer's Notebook**. O'Reilly

*Javid Jamae, Peter Johnson* (2009). **JBoss in Action, Configuring the JBoss Application**.

Manning Publications Co

### 7.4 JasperReport

*Giulo Toffoli* (2007). **The definitive guide to iReport**. Apress

### 7.5 Altres

**Obtenció de definicions i glossaris** <http://ww.wikipedia.org>

**Obtenció de fotos utilitzades en la presentació i portal (amb llicència Creative Commons)**

<http://flickrcc.bluemountains.net/flickrCC/index.php>

Apunts d'assignatures de la carrera Enginyeria Tècnica en Informàtica de Gestió (UOC):

*Jordi Cabot Sagrera, Josep Maria Camps Riba, Jordi Ceballos Villach, Francisco Javier Durán Muñoz, Nathalie Moreno Vergara, José Raúl Romero Salguero, Antonio Vallecillo Moreno*. **Enginyeria del programari de components i de sistemes distribuïts**

*Benet Campderrich Falgueras*. **Enginyeria del programari**

*Fatos Xhafa*. **Tècniques de Desenvolupament de Programari**

*Miquel Barceló García i Joan Antoni Pastor i Collado*. **Gestió organitzacions i projectes informàtics**

## 8 Annex: Creació del projecte des d'Eclipse

Tenint el *JBoss Developer* configurat en l'Eclipse permet que la creació d'un projecte d'un aplicatiu Struts sigui molt senzill, ja que et genera l'estructura de fitxers i totes les llibreries necessàries per desenvolupar el projecte (incloent les de Struts i Hibernate).

### 8.1 Variables d'entorn

Per treballar correctament amb el projecte des d'Eclipse hem de definir les següents variables d'entorn:

- **JAVA\_HOME**: Apunta al directori a on esta instal·lat Java (per exemple JAVA\_HOME=C:\programes\Java\jdk1.6.0\_23).
- **ANT\_HOME**: Apunta al directori a on esta instal·lat Ant (per exemple ANT\_HOME=C:\programes\apache-ant-1.8.2)
- **JBOSS\_HOME**: Apunta al directori de JBoss (per exemple JBOSS\_HOME=C:\programes\jboss-5.1.0.GA)

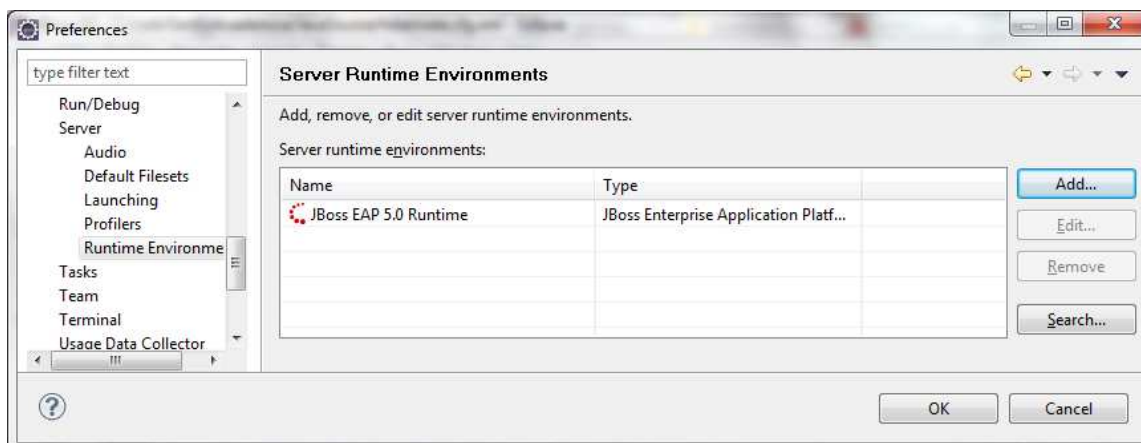
### 8.2 Instal·lació del plugin de JBoss

El plugin *JBoss AS Tools*, que es necessitava en el desenvolupament del projecte, utilitza la funcionalitat de control d'actualitzacions interna d'Eclipse que fa molt senzilla la instal·lació inicial i les seves actualitzacions posteriors. Per realitzar aquesta instal·lació simplement hem d'anar al

menú principal *Help* > *Software Updates*, escriure la direcció <http://download.jboss.org/jbosstools/updates/stable/> i cercar una versió estable del *JBoss Tools*.

### 8.3 Instal·lar JBoss com a Runtime i Servidor

Per poder crear, depurar i executar aplicacions J2EE des de Eclipse necessitem crear instàncies del Runtime i Servidor JBoss, i això ens ho proporciona el plug-in *JBoss AS*. Hem d'anar al menú principal d'Eclipse *Windows* > *Preferences* > *Server* > *RuntimesEnvironment* > *Add* i seleccionar *JBoss Enterprise Middleware* -> *JBoss Enterprise Application Platform 5.0 Runtime*.

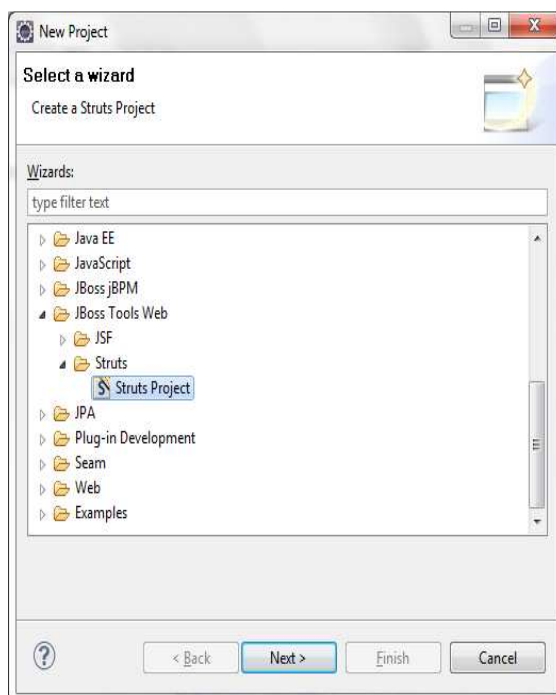


Il·lustració 49. Creació instància runtime

### 8.4 Creació del projecte

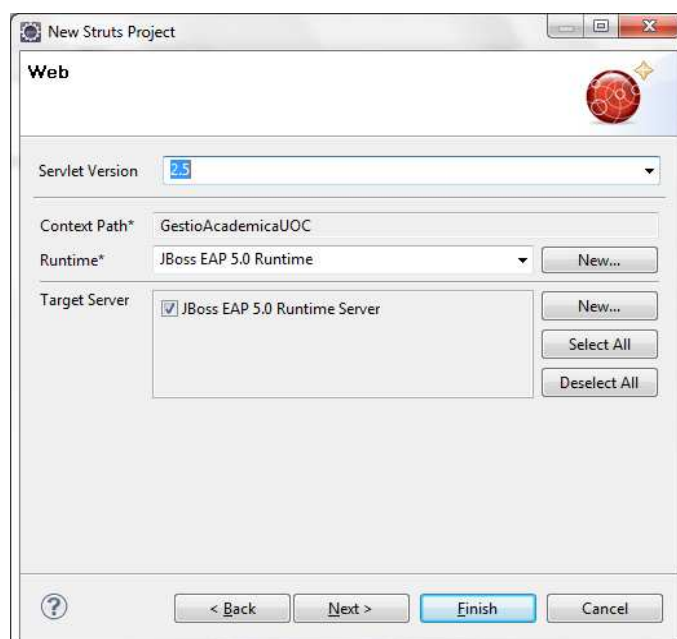
Els passos a realitzar són:

- Anem a la barra de menú i seleccionem *File* > *New* > *Project* ...
- Seleccionem *JBoss Tools Web* > *Struts* > *Struts Project* en la pantalla modal



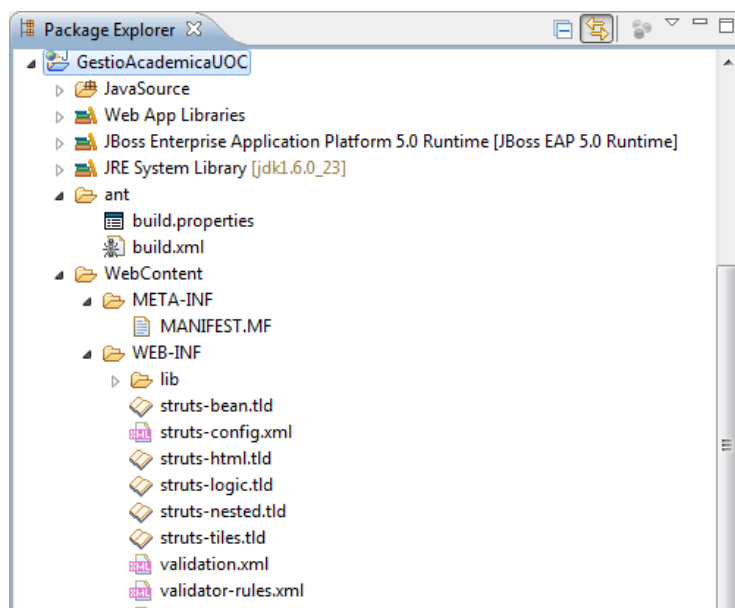
II-lustració 50. Creació projecte Struts

- Seleccionem Next i introduïm el nom del projecte, versió de Struts, versió del servlet i el servidor d'aplicacions JBoss



II-lustració 51. Selecció versió servlet

El *package explorer* queda:



II-lustració 52. Visió genèrica de com queda el projecte

## 8.5 JasperReport

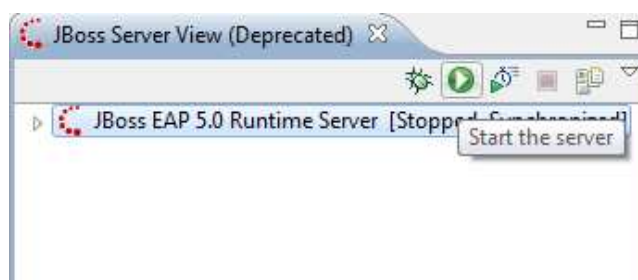
Per poder generar informes dintre de l'aplicació amb JasperReport, simplement s'han de copiar tres llibreries dintre del projecte en el la ruta del projecte\WebContent\WEB-INF\lib.

- jcommon-1.0.16
- jasperreports-3.5.3
- iText-2.1.7

La creació i compilació dels informes el realitzarem mitjançant el programari iReport.

## 8.6 Arrancar servidor d'aplicacions JBoss

Canviem de perspectiva a JBoss AS i arranquem el servidor:



II-lustració 53. Arrancar el servidor JBoss

## 8.7 Instal·lació del connector JDBC de MySQL

Perquè els components que s'executen dins del servidor d'aplicacions puguin accedir al connector JDBC per a MySQL s'ha de copiar la llibreria .jar del connector en el directori `JBOSS_HOME\server\default\lib` i en `JBOSS_HOME\lib`, on `JBOSS_HOME` és la ruta on està instal·lat el servidor d'aplicacions JBoss.



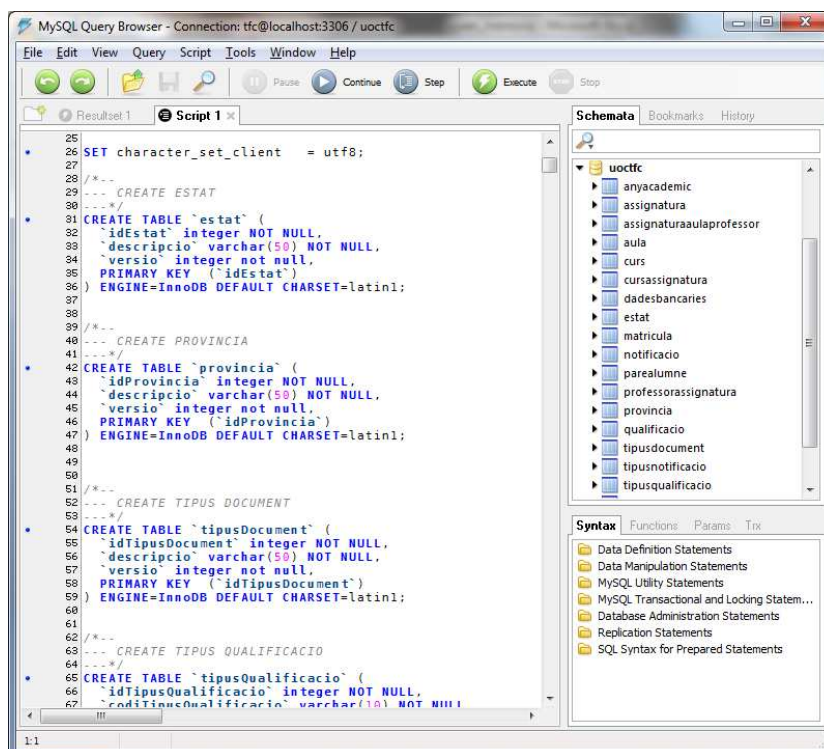
## 8.8 Configuració de JBoss per usar MySQL

Perquè JBoss utilitzi per defecte la base de dades MySQL i treballi amb l'esquema del projecte per defecte, s'han de modificar un parell de fitxers:

- **login-config.xml** (*server/default/conf*), s'haurà de posar l'usuari i contrasenya de la base de dades.
- **mysql-ds.xml** (*server/default/deploy*), s'ha d'especificar el *driver* de MySQL

## 8.9 Crear taules a MySQL

Crearem les diferents taules i els seus registres, mitjançant les ordres SQL que es troben al fitxer *dataBase.sql*, utilitzant l'eina *MySQL Query Browser*:



Il·lustració 54. Creació de taules a MySQL

## 8.10 Hibernate

Des del fitxer de configuració d'Hibernate *hibernate.cfg.xml* s'haurà d'especificar l'esquema, usuari i contrasenya de la base de dades i el *driver* de connexió amb MySQL:

```

<property name="connection.url">jdbc:mysql://localhost/uocffc</property>
<property name="connection.username">tfc</property>
<property name="connection.password">uoc</property>
<property name="connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="dialect">org.hibernate.dialect.MySQLDialect</property>

```

## 8.11 Generar amb l'Ant

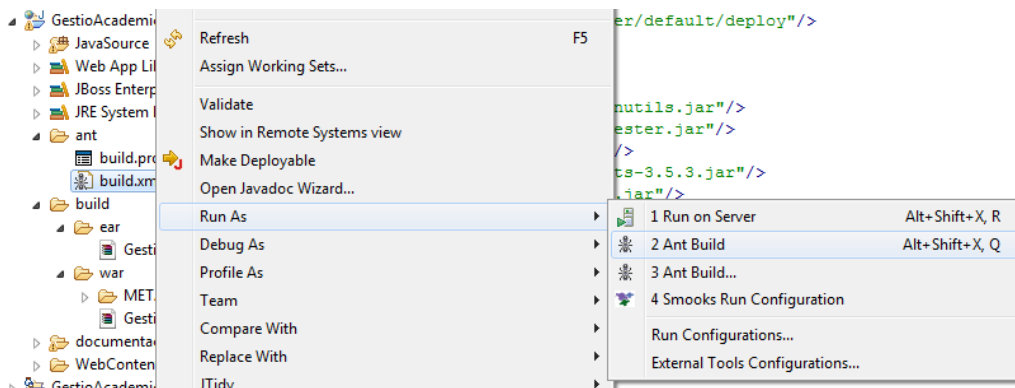
Des del fitxer *build.xml* l'hem de dir el lloc on està instal·lat el JBoss:

```

<property name="jboss.home" value="C:\programes\JBoss-5.1.0.GA" />

```

Després seleccionarem el fitxer build.xml i amb el botó dret, indicarem que volem executar l'Ant:

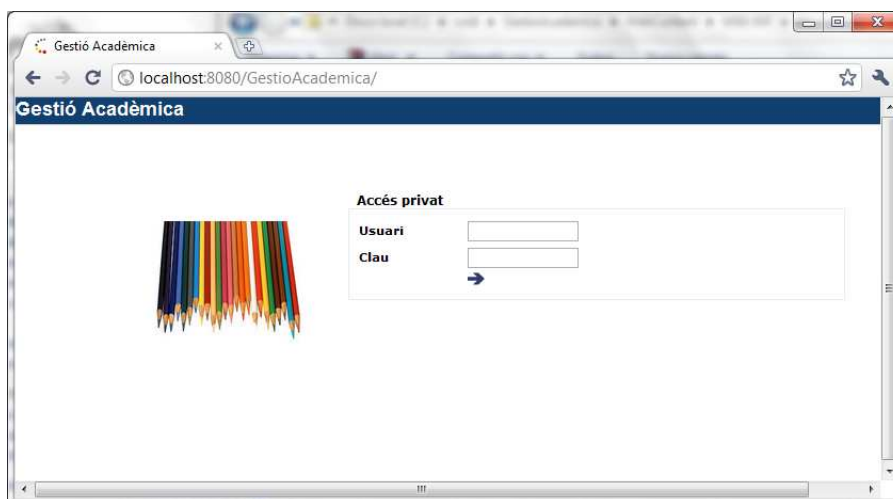


Il·lustració 55. Generar amb l'Ant

Això crearà el fitxer *GestioAcademica.war* i *GestioAcademica.ear* en el servidor d'aplicacions.

## 8.12 Execució des del navegador

Tenint el JBoss arrencat, cridem l'enllaç <http://localhost:8080/GestioAcademica/>:



Il·lustració 56. Executar aplicació

## 8.13 Accés al sistema

Mitjançant el fitxer *dataBase.sql* s'han creat diferents usuaris amb diferents rols. De totes maneres en la següent taula es posa un exemple de cada tipus de rol.

Rol	Login	Contrasenya	Observacions
Administrador	admin	admin	
Personal Administratiu	garc_lau	Sense contrasenya. El primer cop força a canviar la contrasenya.	
Professor	aceb_pab	Sense contrasenya. El primer cop força a canviar la contrasenya.	

Alumne	sanc_agu	Sense contrasenya. El primer cop força a canviar la contrasenya.	
Pare/Tutor legal	garc_ain	Sense contrasenya. El primer cop força a canviar la contrasenya.	És la mare de l'alumne amb codi d'accés sanc_agu.

### 8.14 Programari utilitzat durant el projecte

Programari	Versió	Pàgina web
JDK 6.23	6.23	<a href="http://www.oracle.com/technetwork/java/javase/downloads/index.html">http://www.oracle.com/technetwork/java/javase/downloads/index.html</a>
Ant	1.8.2	<a href="http://ant.apache.org/bindownload.cgi">http://ant.apache.org/bindownload.cgi</a>
MySQL	5.5.9	<a href="http://dev.mysql.com/downloads/mysql/">http://dev.mysql.com/downloads/mysql/</a>
MySQL GuiTools	5.2	<a href="http://dev.mysql.com/downloads/workbench/5.2.html">http://dev.mysql.com/downloads/workbench/5.2.html</a>
Connector JDBC MySQL	5.1	<a href="http://dev.mysql.com/downloads/connector/j/5.1.html">http://dev.mysql.com/downloads/connector/j/5.1.html</a>
JBoss	5.1.9	<a href="http://downloads.sourceforge.net/jboss/jboss-5.1.0.GA-jdk6.zip">http://downloads.sourceforge.net/jboss/jboss-5.1.0.GA-jdk6.zip</a>
Astah Commuty	6.3	<a href="http://astah.change-vision.com/en/product/astah-community.html">http://astah.change-vision.com/en/product/astah-community.html</a>
Dia	0.97.1.2	<a href="http://sourceforge.net/projects/dia-installer/files/dia-win32-installer/0.97.1/dia-setup-0.97.1-2.exe/download">http://sourceforge.net/projects/dia-installer/files/dia-win32-installer/0.97.1/dia-setup-0.97.1-2.exe/download</a>
iReport	3.5.3	<a href="http://jasperforge.org/website/ireportwebsite/IR%20Website/ir_download.html?header=project&amp;target=ireport">http://jasperforge.org/website/ireportwebsite/IR%20Website/ir_download.html?header=project&amp;target=ireport</a>
Eclipse	Galileo	<a href="http://www.eclipse.org/downloads/">http://www.eclipse.org/downloads/</a>